

Dreidimensionale virtuelle Organismen

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim Fachbereich Informatik und Mathematik
der Johann Wolfgang Goethe – Universität
in Frankfurt am Main

von
Tobias Christopher Breiner
aus Frankenthal in der Pfalz

Frankfurt 2005
(D 30)

vom Fachbereich Informatik und Mathematik der
Johann Wolfgang Goethe – Universität als Dissertation angenommen.

Dekan:

Gutachter:

Datum der Disputation:

Inhaltsverzeichnis

Inhaltsverzeichnis	V
Abbildungsverzeichnis	XIII
Tabellenverzeichnis	XVI
Kurzfassung	XVII
Danksagung.....	XIX
1 Einleitung.....	1
1.1 Motivation.....	1
1.1.1 Generierungsmethoden ohne Computerhilfe	4
1.1.1.1 Animatronik.....	4
1.1.1.2 Man in a Suit	7
1.1.1.3 Masken	8
1.1.1.4 Puppen, Marionetten und Stop-Motion-Armaturen.....	8
1.1.2 Aufzeichnungscharakter bei CGI-Organismen	11
1.1.3 Aufwand und Nutzen	16
1.1.4 Qualitätsaspekte	19
1.1.5 Fehlende Interaktivität	23
1.1.6 Zusammenfassendes zur Motivation	24
1.2 Ziele und Durchführung der Arbeit.....	25
1.3 Gliederung der Arbeit.....	28
2 Grundlagen und State-of-the-Art-Analyse	31
2.1 Geometrie-Repräsentationen	32
2.1.1 Polygonnetze.....	32
2.1.2 Parametrische Flächenbeschreibungen	35
2.1.3 Volumendaten	36
2.1.4 Billboards.....	37
2.1.5 Punkte und Linien	37
2.1.6 L-Systeme	38
2.1.7 Blobby Objects.....	39
2.1.8 Sonstige Geometrie-Repräsentationen	42
2.1.9 Bewertung der Geometrie-Repräsentationen	43

2.2	Formerfassungsmethoden	47
2.2.1	Mechanooptische Stiftverfahren	47
2.2.2	3D-Scanning	48
2.2.2.1	Stereoskopische Triangulationsverfahren.....	53
2.2.2.2	Punktbasierte Triangulationsverfahren.....	55
2.2.2.3	Streifenbasierte Triangulationsverfahren.....	56
2.2.2.4	Graycode-Triangulationsverfahren.....	57
2.2.2.5	Phasenschiftverfahren.....	58
2.2.2.6	Konoskopische Holographie.....	60
2.2.2.7	Pulslaufzeitverfahren.....	61
2.2.2.8	Kombinationen	62
2.2.3	Bewertung der Formerfassungsmethoden.....	63
2.3	Animationstechniken und Bewegungserfassungsmethoden.....	68
2.3.1	Parametrisches Keyframing	68
2.3.1.1	Linear Interpolation Keyframing.....	69
2.3.1.2	Curved Interpolation Keyframing	69
2.3.1.3	Bewertung des parametrischen Keyframings	70
2.3.2	Pfadanimation	70
2.3.2.1	Double Interpolant Method	71
2.3.2.2	Erweiterte Pfadanimation	71
2.3.2.3	Bewertung der Pfadanimation	71
2.3.3	Kinematik.....	72
2.3.3.1	Vorwärtskinematik	73
2.3.3.2	Inverse Kinematik	73
2.3.3.3	Bewertung der Kinematik.....	73
2.3.4	Motion Capturing.....	74
2.3.4.1	Satellitennavigationsgestützte Motion Capturing Systeme	77
2.3.4.2	Mechanische Motion Capturing Systeme.....	79
2.3.4.2.1	Verwendete Sensoren	79
2.3.4.2.2	Datenübermittlung	79
2.3.4.2.3	Erfasste Körperteile	79
2.3.4.3	Optisch photogrammetrische Motion Capturing Systeme.....	81
2.3.4.3.1	Face Tracking	83
2.3.4.4	Optisch stereoskopische Motion Capturing Systeme	84
2.3.4.5	Optisch videobasierte Motion Capturing Systeme	84
2.3.4.6	Magnetische Motion Capturing Systeme.....	86
2.3.4.7	Akustische Motion Capturing Systeme	86
2.3.4.8	Bewertung des Motion Capturings	87
2.3.5	Stop Motion Input Devices	92
2.3.5.1	Das Dinosaur Input Device	92

2.3.5.2	Modulare DIDs.....	93
2.3.5.3	Bewertung von Stop Motion Input Devices	94
2.4	Basistechniken zur Formveränderung.....	95
2.4.1	Geometrische Methoden zur Objektdeformation.....	95
2.4.2	Simulationsmethoden zur Objektdeformation	96
2.4.2.1	Feder-Masse-Dämpfer-Systeme	96
2.4.2.2	Finite Elemente Methoden	97
2.4.2.3	Randelementmethoden	98
2.4.2.4	Weitere Simulationsmethoden zur Objektdeformation	99
2.4.3	Interpolationsmethoden zur Formveränderung	100
2.4.4	Bewertung der Basistechniken zur Formveränderung	100
2.5	Anatomische Simulationsmodelle	104
2.5.1	Schichtmodelle.....	104
2.5.1.1	Skelettmodelle.....	105
2.5.1.2	3D-Skinning	106
2.5.1.3	Ellipsoidal Sweeping.....	108
2.5.1.4	Das Dreischichtenmodell von Chadwick et al.....	109
2.5.1.5	Das implizite Primitivenmodell von Scheepers et al.....	110
2.5.1.6	Das implizite Primitivenmodell von Thalmann et al.....	110
2.5.1.7	Das Elastic Surface Layer Model	111
2.5.1.8	Bewertung der Schichtmodelle.....	112
2.5.2	Dynamische Muskelmodelle.....	113
2.5.2.1	Dynamisches Muskelmodell von Terzopoulos.....	113
2.5.2.2	Das Modell von Miller	115
2.5.2.3	Das Modell von Nedel und Thalmann.....	116
2.5.2.4	Bewertung der dynamischen Muskelmodelle.....	120
3	Defizit- und Anforderungsanalyse	121
3.1	Holistischer Generierungsprozess	121
3.2	Verwendung intuitiver Metaphern.....	124
3.3	Abkehr vom Aufzeichnungscharakter	127
3.4	Verwendung evolutionärer Methoden	128
3.5	Reusability von anatomischen Strukturen.....	129
3.6	Verwendung medizinischer Termini.....	130
3.7	Integration biologischer Constraints.....	132
3.8	Kanalisierte Modellierung und Animation.....	134

3.9	Echtzeitfähigkeit der Animation	135
3.10	Hohe Bandbreite der Abstraktionsstufen	137
3.11	Hoher Grad des Realismus	140
4	Das Quaoaring-Konzept	143
4.1	Grundidee	144
4.1.1	Allgemeines zum Verständnis des Quaoarings	146
4.1.2	Kurzübersicht über das Gesamtverfahren	149
4.2	Das biologische Koordinatensystem	152
4.2.1	Definitionsproblematik bei kartesischen Koordinatensystemen	152
4.2.2	Umwandlung biologischer in kartesische Koordinaten.....	153
4.2.3	Biologisches versus zylindrisches Koordinatensystem.....	155
4.2.4	Beziehung zwischen anatomischen Termini und Biovektoren	157
4.2.5	Definition von Flächen und Räumen im BKS	161
4.2.6	Kreuzungen im BKS	161
4.3	Deformation hydropneumatischer Einheiten	163
4.3.1	Erster Ansatzversuch mit einem Feder-Masse-Dämpfer-Modell.....	163
4.3.2	Zweiter realisierter Ansatz mit morphischen Feldern	164
4.3.3	Die Definition und Berechnung von morphischen Feldern.....	166
4.4	Die Abschätzung des Volumens	168
4.4.1	Die punktkonzentrische Volumenberechnung mit Pyramiden.....	169
4.4.2	Die Volumenabschätzung mit Kugelschichten	172
4.4.3	Die Volumenberechnung mit Pyramiden entlang der Zentrallinie.....	173
4.4.4	Die Volumenabschätzung mit Prismen entlang der Zentrallinie.....	176
4.4.5	Vergleich der Volumenabschätzungsmethoden	179
4.5	Die Feinmodellierung der Haut	180
4.6	Verbinden zweier Pneus	185
4.7	Animation	187
4.7.1	Kopplung von Kontraktionselementen mit Zentrallinien.....	187
4.7.2	Kontraktionssysteme.....	191
4.7.3	Steuerungsmodus	191
4.7.4	Animation von Wachstums- und Evolutionsprozessen.....	192
4.8	Quaoaring Design Patterns	196
4.8.1	Objekthierarchie der Quaoaring Design Patterns	197
4.8.2	Main.....	198
4.8.3	Bulge.....	199

4.8.4	Canal	201
4.8.5	Organ	204
4.8.6	Squeeze	205
4.8.7	Constriction.....	207
4.8.8	Conics	209
4.8.9	Contractile.....	210
4.8.10	Relief.....	212
4.8.11	Skin	214
4.8.12	Hair.....	216
4.8.13	Material	217
4.8.14	Particlessource	219
4.9	Modellierung mit den Quaoaring Design Patterns	223
4.9.1	Modellierungsbefehle.....	223
4.9.2	Visualisierungsbefehle.....	224
4.9.3	Beispiel für einen Modellierungsvorgang eines Organismus.....	227
5	Implementierungsaspekte	231
5.1	Auswahl der Hilfssoftware	231
5.2	Auswahl der Hardware und der Betriebssysteme.....	234
5.3	Benennungskonventionen.....	235
5.4	Softwareschichten und –module	236
5.4.1	Aufbau und Funktionalitäten der Sun-Engine.....	238
5.4.2	Aufbau und Funktionalitäten der TOP-Softwareschicht	238
5.4.3	Aufbau und Funktionalitäten der MOB-Softwareschicht	239
5.5	Wichtige Softwareschnittstellen.....	240
5.5.1	Die Schnittstelle zwischen MOB- und TOP-Schicht	241
5.5.2	Die Schnittstelle zwischen TOP-Schicht und SUN-Engine	248
5.6	Ordnerstruktur	249
6	Anwendungsbeispiele und Ergebnisse.....	253
6.1	Begründung der Auswahl von Anwendungsbeispielen.....	254
6.1.1	Anforderungskriterien für die Selektion	254
6.1.2	Unterschiede der visualisierten Holarchiestufe.....	254
6.1.3	Unterschiede bei der visualisierten Komplexität.....	255
6.1.4	Unterschiede der analytischen Fähigkeiten.....	256
6.1.5	Unterschiede beim Interaktivitätsgrad	257
6.1.6	Unterschiede der Stabilitätsanforderungen	257

6.1.7	Unterschiede beim erforderlichen Realismus	258
6.1.8	Übersicht der Anforderungsspektren	259
6.2	e-Go	260
6.3	e-Voluzzer	262
6.4	e-VoLuzie	265
6.5	e-VoLutz.....	268
6.6	e-Rektus.....	274
6.7	e-Munkulus	277
6.8	e-Rythrozyt	284
6.9	Bewertung	293
7	Ausblick und Abschluss	295
7.1	Defizite und Erweiterungsmöglichkeiten des Quaoarings.....	295
7.1.1	Kollisionserkennung	295
7.1.2	Kombination mit Feder-Masse-Dämpfer-Systemen	296
7.1.3	Vesikel als weiteres Quaoaring Design Pattern	297
7.1.4	Level-of-Detail des Skin-Meshs	297
7.1.5	Level-of-Detail der Quaoaring Design Patterns	299
7.1.6	Import von Volumendaten	299
7.2	Mögliche Einsatzgebiete des Quaoarings.....	301
7.2.1	Einsatzgebiete in Edutainment.....	301
7.2.2	Einsatzgebiete im Entertainmentbereich.....	302
7.2.3	Einsatzgebiete im e-Learning-Bereich.....	303
7.2.4	Einsatzgebiete in der Kommunikation.....	303
7.2.5	Einsatzgebiete in der Diagnostik.....	304
7.2.6	Einsatzgebiete als Planungshilfe	304
7.2.7	Einsatzgebiete in der Forschung	304
7.2.8	Einsatzgebiete in Design und Produktion	305
7.2.9	Einsatzgebiete als Komprimierungshilfe	305
7.3	Evaluation	306
7.4	Zusammenfassung und Abschluss	309
8	Literaturverzeichnis	311
9	Anhang.....	369

9.1	Zusatzinformationen zum Inhalt	369
9.1.1	Herleitung des Namens "Quaoaring"	369
9.1.2	Tabelle für 3D-Scanning	371
9.1.3	Vergleich von Szenegraphen	375
9.1.3.1	Grundsätzliche Eigenschaften und Kompatibilitäten	375
9.1.3.2	Unterstützte Objekte	376
9.1.3.3	Features der unterstützten Objekte	378
9.1.3.4	Angebotene I/O-Schnittstellen	381
9.1.3.5	Netzwerkfähigkeit	384
9.2	Vorarbeiten des Verfassers	385
9.2.1	Schriftenverzeichnis	385
9.2.1.1	Wissenschaftliche Proceedings	385
9.2.1.2	Populärwissenschaftliche Publikationen	386
9.2.1.3	Interne Schriften	386
9.2.1.4	Nichtwissenschaftliche Artikel	388
9.2.2	Vorträge	391
9.2.3	Organisierte Workshops	394
9.2.4	Erstellte Software	395
9.2.5	Ausstellungen und Messepräsentationen	402
9.2.6	Betreute Studien- und Diplomarbeiten	405
9.2.7	Mitwirkung in Lehrveranstaltungen	408
9.3	Lebenslauf	411

Abbildungsverzeichnis

Abbildung 1: Beispiele für tatsächlich/vermeintlich computergenerierte Organismen.....	3
Abbildung 2: Filmstandbild mit einer animatronischen Einheit.....	4
Abbildung 3: Der dreistufige Herstellungsprozess der animatronischen Skulptur.....	5
Abbildung 4: Beispiel für das Innenleben von Animatronik.....	6
Abbildung 5: Flussdiagramm des Herstellungsprozesses für Animatronik.....	6
Abbildung 6: Beispiel für Man in a Suit-Systeme.....	7
Abbildung 7: Beispiel für den Herstellungsprozess einer mechanischen Maske.....	8
Abbildung 8: Beispiel für eine Stop-Motion-Armatur.....	9
Abbildung 9: Beispiel für individuelle versus modulare Stop-Motion-Armaturen.....	10
Abbildung 10: Standardprozess von CGI-Szenen mit Organismen.....	13
Abbildung 11: Beispiel für den Aufzeichnungscharakter von CGI-Organismen.....	14
Abbildung 12: Stationen des Sculpting-Prozesses des Hulk.....	15
Abbildung 13: Beispiele für nicht überzeugende Resultate bei CGI-Organismen.....	21
Abbildung 14: Beispiel für computergenerierte Tiere in Close-Ups.....	23
Abbildung 15: Beispiel für Dot Product 3 Bump Mapping.....	34
Abbildung 16: Beispiel für die Fusion zweier Blobby Objects.....	39
Abbildung 17: Graphen der Feldfunktionen von Blobby Objects.....	41
Abbildung 18: Beispiel für die Erstellung von Organismen mit Blobby Objects.....	42
Abbildung 19: Beispiel für die Formerfassung mittels Articulation.....	48
Abbildung 20: Beispiele von Close Range-, Surface- und Mid Range-3D-Scannern.....	51
Abbildung 21: Beispiel eines Whole Body 3D-Scanners.....	52
Abbildung 22: Skizze des punktbasierten Triangulationsverfahrens.....	56
Abbildung 23: Skizze des streifenbasierten Triangulationsverfahrens.....	57
Abbildung 24: Skizze zum Graycode-Triangulationsverfahren.....	58
Abbildung 25: Skizze zur Konoskopischen Holographie.....	61
Abbildung 26: Beispielbilder für die Resultate von 3D-Scanning.....	64
Abbildung 27: Beispiel für 3D-Scanning von organismischen Skulpturen.....	65
Abbildung 28: Beispiele für Problemzonen beim 3D-Scanning.....	67
Abbildung 29: Beispiel für Motion Capturing zur Animation von Fabelwesen.....	75
Abbildung 30: Beispiel für satellitennavigationsgestütztes Motion Capturing.....	78
Abbildung 31: Beispiel eines drahtlosen Datenhandschuhs.....	80
Abbildung 32: Beispiel eines drahtlosen mechanischen Ganzkörpertrackers.....	81

Abbildung 33: Gebräuchliche Sensorpositionen	82
Abbildung 34: Beispiel für optisch photogrammetrisches Motion Capturing bei Tieren ...	83
Abbildung 35: Beispiel für Face Tracking	83
Abbildung 36: Beispiel für optisch videobasiertes Motion Capturing	85
Abbildung 37: Beispiel für akustisches Motion Tracking	87
Abbildung 38: Beispiel für ein Dinosaur Input Device	93
Abbildung 39: Beispiel für modulare DID-Systeme	93
Abbildung 40: Beispiel für die Anwendung modularer DID-Systeme.....	94
Abbildung 41: Schema einer Feder-Masse-Dämpfer-Funktionseinheit	97
Abbildung 42: Beispiel für Mesh Skinning	106
Abbildung 43: Skizze und Beispiele für das Syndrom des kollabierenden Ellenbogens ..	108
Abbildung 44: Vier Stationen des Ellipsoidal Sweepings	109
Abbildung 45: Beispiel für das implizite Primitivenmodell von Thalmann et al.....	111
Abbildung 46: Skizze des Vierschichtenmodells	112
Abbildung 47: Skizze des dynamischen Fischmodells von Terzopoulos.....	114
Abbildung 48: Skizze der Wahrnehmung des Fischmodells von Terzopoulos	115
Abbildung 49: Verlauf verschiedener Action Lines	116
Abbildung 50: Schema eines Partikels mit vier Nachbarpartikeln	117
Abbildung 51: Adaptierte Hüllenform von Oberarmmuskeln.....	117
Abbildung 52: Skizze der Positionen der Angular Springs	119
Abbildung 53: Skizze des konventionellen Generierungsprozesses für Organismen	122
Abbildung 54: Skizze des holistischen Generierungskonzepts von Organismen.....	123
Abbildung 55: Beispiel für ein “Luftballonmännchen”	126
Abbildung 56: Drahtgittermodell eines Pneus mit seiner Zentrallinie	153
Abbildung 57: Skizze der Komponenten eines Biovektors	155
Abbildung 58: Beispiel für die einfache Erzeugung von Mischwesens mit dem BKS.....	162
Abbildung 59: Auswirkung des Parameters d bei morphischen Feldern.....	166
Abbildung 60: Skizze der Polygonverteilung im Skin-Mesh	170
Abbildung 61: Beispiel für hydrostatisches Bump Mapping.....	181
Abbildung 62: Realistisches Verhalten von hydrostatischen Bump Maps.....	184
Abbildung 63: Wachstumsprozesse mit konkurrierenden hydrostatischen Bump Maps ..	184
Abbildung 64: Beispiel einer Verbindung zwischen zwei Pneus	186
Abbildung 65: Beispielskizze für die Erstellung von Gelenken, Knochen und Muskeln .	189
Abbildung 66: Sechs Beispielstationen einer Organismenanimation.....	190

Abbildung 67: Wachstumsanimation eines Phantasietieres	195
Abbildung 68: Beispiel von Bulges	200
Abbildung 69: Beispiel von Canals	202
Abbildung 70: Beispiel von Squeezes	206
Abbildung 71: Beispiel für Constrictions	208
Abbildung 72: Beispiel eines Conics.....	209
Abbildung 73: Beispiel für die Beziehung zwischen Texturen, Skins und Reliefs	213
Abbildung 74: Beispiel eines Skins mit seinem Verhalten.....	215
Abbildung 75: Modellierung eines Protoplesiosauriers.....	230
Abbildung 76: Aufbau des Quaoaring-Software-Frameworks mit Fremdbibliotheken	237
Abbildung 77: Gantt-Diagramm der Zeiten der auf Quaoaring basierenden Projekte	253
Abbildung 78: Zusammenhang verschiedener auf Quaoaring basierender Projekte.....	260
Abbildung 79: Visualisierungen von Würmern, Quappen und Egel	262
Abbildung 80: Animation peristaltischer Bewegungen.....	263
Abbildung 81: Visualisierung der Evolution der Enteropneusten	264
Abbildung 82: Drei Bewegungsstationen von e-VoLuzie	267
Abbildung 83: Das Exponat e-VoLutz im Cybernarium	269
Abbildung 84: Screenshot des Elasmosaurus	270
Abbildung 85: Visualisierung einer Nothosaurier-Evolutionslinie	271
Abbildung 86: Beispiel für die Kohärenz von Zwischenstadien	272
Abbildung 87: Flossenschlag des Nothosaurus	273
Abbildung 88: Screenshots des k-MED Pharmakologie-Piloten.....	276
Abbildung 89: Symbolische Anzeige von Organstadien.....	277
Abbildung 90: Visualisierung eines ausgewachsenen Neurons	279
Abbildung 91: Nahaufnahme eines wachsenden Neurons	280
Abbildung 92: Animation der Depolarisation eines Dendriten	282
Abbildung 93: Visualisierung eines Neuronalen Netzwerkes	283
Abbildung 94: Visualisierung eines Discozyten mit Falschfarbencodierung.....	286
Abbildung 95: Visualisierung eines Erythrozyten in drei Zoom-Stufen	290
Abbildung 96: Visualisierung eines Echinozyten.....	291
Abbildung 97: Visualisierung eines Discozyten.....	292

Tabellenverzeichnis

Tabelle 1: Anwendungsmöglichkeiten von Motion Capturing bei Organismen	91
Tabelle 2: Vor- und Nachteile der Methoden für die Volumenabschätzung eines Pneus .	180
Tabelle 3: Vergleich des hydrostatischen Bump Mappings mit anderen Verfahren	182
Tabelle 4: Mögliche Eltern-Kind-Beziehungen der Quaoaring Design Patterns.....	198
Tabelle 5: Modellierungsparameter von Main.....	199
Tabelle 6: Modellierungsparameter von Bulge	201
Tabelle 7: Modellierungsparameter von Canal.....	203
Tabelle 8: Modellierungsparameter von Organ	205
Tabelle 9: Modellierungsparameter von Squeeze.....	206
Tabelle 10: Modellierungsparameter von Constriction	208
Tabelle 11: Modellierungsparameter von Conic	210
Tabelle 12: Modellierungsparameter eines Contractiles	211
Tabelle 13: Modellierungsparameter eines Reliefs.....	214
Tabelle 14: Modellierungsparameter eines Skins	216
Tabelle 15: Modellierungsparameter von Hair	217
Tabelle 16: Modellierungsparameter eines Materials.....	218
Tabelle 17: Modellierungsparameter von Particlesource	221
Tabelle 18: Anforderungen an die auf Quaoaring basierenden Projekte	259
Tabelle 19: Herstellerfirmen von 3D-Scannern.....	374
Tabelle 20: Vergleich der Eigenschaften/Kompabilitäten von sechs Szenengraphen.....	376
Tabelle 21: Unterstützte Objekte von sechs Szenegraphen	377
Tabelle 22: Object Features von sechs Szenegraphen	380
Tabelle 23: Angebotene I/O-Schnittstellen von sechs Szenegraphen.....	383
Tabelle 24: Netzwerkfähigkeit von sechs Szenegraphen.....	384

Kurzfassung

Die vorliegende Arbeit befasst sich mit der Generierung virtueller Organismen respektive mit der dreidimensionalen Nachbildung anatomischer Strukturen von Pflanzen, Tieren, Menschen und imaginärer Wesen per Computer. Berücksichtigt werden dabei sowohl die verschiedenen Aspekte der Visualisierung, der Modellierung, der Animation sowie der Wachstums-, Deformations- und Bewegungssimulation.

Dazu wird zuerst eine umfassende State-of-the-Art-Analyse konventioneller Methoden zur Organismengenerierung durchgeführt. Im Laufe dieser Analyse werden die Defizite herkömmlicher Verfahren aufgezeigt und damit eine gezielte Anforderungsanalyse für neue Verfahren erstellt.

Mit Hilfe dieser Anforderungsanalyse wurde nach neuen Lösungsansätzen gesucht. Besonders hilfreich hat sich in diesem Zusammenhang die Frankfurter Organismus- und Evolutionstheorie erwiesen. Gemäß dieser Theorie stellen Organismen aus biomechanischer Sicht komplexe hydropneumatische Konstruktionen dar. Ihre Körperformen und Bewegungen werden weitgehend durch stabilisierende, kräfteerzeugende und kräfteübertragende Strukturen generiert, die den Gesetzen der klassischen Hydropneumatik folgen. So entstand die Idee, Organismen auf der anatomischen Ebene als eine komplexe Hierarchie unterschiedlicher hydropneumatischer Einheiten anzusehen, welche mechanisch miteinander interagieren.

Diese Sichtweise liefert die Grundlage für ein neues biologisches Simulationsmodell. Es erlaubt der Computergraphik, sowohl die Form eines Organismus zu beschreiben als auch sein Verhalten bezüglich seiner Bewegungsabläufe, seiner evolutionären Formveränderungen, seiner Wachstumsprozesse und seiner Reaktion auf externe mechanische Krafteinwirkungen numerisch zu simulieren.

Aufbauend auf diesem biologischen Simulationsmodell wurde ein neues Verfahren (Quaoaring) entwickelt und implementiert, das es erlaubt, beliebige organische Einheiten interaktiv in Echtzeit zu modellieren. Gleichzeitig ermöglicht dieses Verfahren die Animation von Bewegungen, Wachstumsprozessen und sogar evolutionären Entwicklungen. Die Animation verhält sich dabei im Wesentlichen biologisch stringent, z.B. wird das interne Volumen während komplexer Bewegungsabläufe konstant gehalten.

Die größte Stärke der neuen Modellierungs- und Animationstechnik ist die holistische Verschmelzung des biologischen Simulationsmodells mit einem computergraphischen Geometriemodell. Dieses erlaubt dem Modellierer, biologische

Konzepte für die Beschreibung der Form und anderer Attribute einer organischen Einheit zu verwenden. Darüber hinaus ermöglicht es die Animation des geometrischen Modells durch einfache Parameterspezifikation auf einer hohen Abstraktionsebene. Dazu wird ein Autorenprozess beschrieben, wie Quaoaring für Modellierungs- und Animationszwecke verwendet werden kann.

Es werden Aspekte der prototypischen Implementierung der Quaoaringtechnologie behandelt und über die Ergebnisse berichtet, die bei der Implementierung und der Anwendung dieses Softwareframeworks gewonnen wurden.

Schließlich wird die Quaoaringtechnologie in ihrem technologischen Kontext beleuchtet, um ihr Zukunftspotential einzuschätzen.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter an der Johann Wolfgang Goethe-Universität Frankfurt am Main in der Zeit von November 2002 bis Oktober 2005. An dieser Stelle möchte ich mich bei all jenen bedanken, die am Gelingen dieser Arbeit beteiligt waren:

Mein erster und besonderer Dank gilt *Professor Dr. Detlef Krömker* für die Überlassung des Themas sowie für die vorzügliche Betreuung, welche durch eine herausragende fachliche Kompetenz gepaart mit einem freundlich hilfreichen Umgang geprägt war.

Bedanken möchte ich mich auch für die kompetente Beratung aus biologischer Sicht und die gute Zusammenarbeit bei *Dr. Ingolf Bernhardt*, *PD Dr. Eberhard Frey* alias *Dino*, *Dr. Michael Gudo*, *Dr. Tareq Syed*, *Dr. Oliver Bossert*, *Dr. med. Mathias Wagner* und *Dr. Claus Weiland*.

Zu großem Dank bin ich auch meinen Kolleginnen und Kollegen verpflichtet, die mir bei mancher Frage weitergeholfen haben. Besonders hervorheben möchte ich hierbei *Daniel Abawi*, *Wolfgang Baier (Dennis)*, *Ashraf Baker*, *Jens Barthelmes*, *Joachim Bienwald*, *Jörg Demmer*, *Prof. Dr. Ralf Dörner*, *Roy Goldberg*, *Prof. Dr. Paul Grimm*, *Michael Krauß*, *Christopher Leipold*, *Laurent Letourneux*, *Michael Mandaglio*, *Marianne Mast*, *Michaela Mümpfer*, *Bojja Muralidhar*, *Rüdiger Nöhte*, *Markus Raab*, *Silvan Reinhold* und *Christian Seiler*.

Des Weiteren danke ich allen Studentinnen und Studenten, die im Rahmen von Diplomarbeiten oder als wissenschaftliche Hilfskräfte meine Arbeit unterstützt haben, vor allem *André Bialojahn*, *Armin Burger*, *Daniel Groh*, *Jörg Homann*, *Lidong Huang*, *Katarina Kinder*, *Toni Nemeč*, *Toni Perkovic* und *Carsten Stocklöv*.

Ein herzlicher Dank geht auch an meine Eltern *Dr. Herbert Breiner* und *Ursula Breiner*, sowie an meine drei Geschwister *Anja Breiner*, *Dr. Kerstin Dominique Ahrend* und *Manuel Breiner* für die kraftvolle familiäre Verwurzelung und den daraus resultierenden motivatorischen Rückhalt.

Der größte Dank gebührt aber meiner Familie, allen voran meiner Frau *Nicole Jansen*, ohne deren Rückendeckung und Unterstützung die vorliegende Arbeit nie beendet worden wäre, sowie meinen Kindern *Felix Shjako Breiner*, *Jonas Kasho Breiner* und *Sina Tsaraja Breiner*.

1 Einleitung

*“If you go completely CGI, you end up with
a movie more like Toy Story”*

Michael Lantieri (Special Effects
Supervisor of Jurassic Park I-III) 2003

1.1 Motivation

Organismen sind hochkomplexe Gebilde. Ihre anatomischen Strukturen bestehen aus unzähligen mechanischen Unterobjekten, welche nach einem komplizierten Regelwerk miteinander interagieren, ohne dass dabei eine festgelegte Hierarchiereihenfolge gegeben wäre. Die Formen dieser interdependenten Unterobjekte sind bis auf wenige Ausnahmen nicht starr und fix, sondern sie können sich durch interne und externe Kräfte nach sehr komplizierten Regeln verändern. Darüber hinaus sind wir als Mensch selbst ein Organismus und wir haben viel visuelle Erfahrung im Umgang mit anderen Organismen erworben, sodass wir schon kleine Form- und Bewegungsabweichungen von der Realität als unnatürlich entlarven.

Aus diesen Gründen zählen Organismen in der Computergraphik zu den am schwierigsten zu modellierenden und zu animierenden Objekten. [Parent 2002, 271ff] Die Generierung dreidimensionaler Organismen stellt daher eine besondere Herausforderung an die Computergraphik dar und sie ist – wie in der vorliegenden Arbeit aufgezeigt wird – noch längst nicht zufrieden stellend gelöst worden. Ja mehr noch, besonders in den Bereichen der interaktiven Simulation und Visualisierung von Organismen stehen die entscheidenden Schlüsseltechnologien und algorithmischen Methoden noch aus.

Diesen Tatsachen steht ein unrealistisches Bild gegenüber, das sich fachfremde Personen machen, wenn sie an virtuelle Organismen denken. Es existiert bei informationstechnischen Laien und sogar bei Informatikern ohne fundierte Ausbildung in Graphischer Datenverarbeitung die Vorstellung, dreidimensionale Organismen seien mühelos einfach mit photorealistischer Perfektion per Computer zu generieren. Die Computergraphik wird hier nahezu als omnipotent angesehen und somit maßlos überschätzt, was im diametralen Missverhältnis zur nüchternen Realität einer in ihren Möglichkeiten stark begrenzten Computergraphik steht.

Dies ist eher ein Fluch als ein Segen für Computergraphiker, denen von fachfremder Seite eine übertrieben hohe Erwartungshaltung entgegengebracht wird: Sätze wie „Könnten Sie mal schnell eine Erythrozytenverformung visualisieren?“, „Wie wäre es, wenn ein Rechnercluster die Evolution dreidimensional nachsimuliert?“ oder „Ich hätte gerne bis morgen einen richtig gefährlich aussehenden 3D-T-Rex!“ sind in diesem Zusammenhang keine Seltenheit.

Die Erstellung der vorliegenden Arbeit erfolgte daher unter einem ständigen Rechtfertigungsdruck. So war eine häufig gestellte Frage, ob es in diesem Bereich überhaupt noch etwas zu verbessern gäbe oder ob nicht schon alles ausgereizt sei.

Diese Überschätzung der Möglichkeiten der Computergraphik ist nicht verwunderlich, denn beim Stichwort „Dreidimensionale Virtuelle Organismen“ werden zumeist die frappierend lebensecht aussehenden Wesen aus millionenschweren Hollywood-Filmen evoziert. Die Dinosaurier aus der „Jurassic Park“-Serie, das Monster „Gollum“ aus der „Herr der Ringe“-Trilogie, das extraterrestrische Monster „Alien“ aus den gleichnamigen Horrorfilmen oder der Trickster „Jarjar“ mit Konsorten aus den StarWars-Episoden können hier exemplarisch genannt werden. Auch die Tiere aus den BBC-Serien „Walking with Dinosaurs“, „Walking with Beasts“ und „The Future is Wild“, geben diesem Gesamteindruck Nahrung. Doch diese zugegebenermaßen beeindruckende Bilderwelt aus Film und Fernsehen täuscht über folgende fünf Punkte hinweg:

1. Viele vermeintlich computergenerierte Organismen sind gar nicht computergeneriert, sondern mit anderweitigen konventionellen Methoden erstellt (Diese Tatsache wird in Abschnitt 1.1.1 „Generierungsmethoden ohne Computerhilfe“ behandelt).
2. Die meisten anderen Organismen sind – bei Verwendung einer genauen Definition – auch nicht vom Computer generiert, sondern allenfalls per Computer aufgezeichnet (Behandelt in Abschnitt 1.1.2 „Aufzeichnungscharakter bei CGI-Organismen“).
3. Sowohl die Modellierung als auch die Animation von Organismen ist nicht intuitiv, sondern sehr aufwändig und damit teuer in ihrer Herstellung (Behandelt in Abschnitt 1.1.3 „Aufwand und Nutzen“).

4. Die Form und die Bewegungsabläufe der computergenerierten Organismen sind bei genauer Betrachtung biologisch inkorrekt. Aus diesem Grund können nur Organismen gezeigt werden, bei denen der Betrachter keine Erfahrungswerte mitbringt, wie z.B. paläontologische Organismen, extraterrestrische Wesen oder Fabeltiere (Behandelt in Abschnitt 1.1.4 „Qualitätsaspekte“).

5. Viele Methoden zur Generierung von Organismen sind aufgrund ihrer schlechten Performanz nicht interaktiv verwendbar. Für lebensecht wirkende Organismen existieren zurzeit überhaupt keine interaktiven Verfahren. Somit sind die Einsatzmöglichkeiten begrenzt (Behandelt in Abschnitt 1.1.5 „Fehlende Interaktivität“).



Abbildung 1: Beispiele für tatsächlich/vermeintlich computergenerierte Organismen

Fünf verschiedene Beispiele für tatsächlich oder vermeintlich computergenerierte Organismen aus Film und Fernsehen. Links oben: zwei mit speziellen hydrodynamischen Puppen (Animatronik) generierte Dinosaurier aus „Jurassic Park III“ [Tribute 2004][Cinema 2004a]; rechts oben: der computergenerierte Trickster „Jarjar“ aus „Star Wars“ [Foxwell 2004]; links unten: das computergenerierte Monster „Gollum“ aus der „Herr der Ringe“-Trilogie [Tuckborough 2004]; Mitte unten: ein Puppen-Alien aus „Alien 3“ [Outnow 2004]; rechts unten: computergenerierter Ankylosaurus aus der BBC-Serie „Walking with Dinosaurs“ [Dyneath 2004]

1.1.1 Generierungsmethoden ohne Computerhilfe

Vielfach ist nicht bekannt, dass die meisten Szenen, in denen Organismen vorkommen, keinesfalls computergeneriert sind, sondern mit anderweitigen, konventionellen Methoden der Filmindustrie erzeugt wurden. Dies gilt sogar für diejenigen Filme, die gemeinhin als Musterbeispiel für *Computer Generated Images*, kurz *CGI*, gelten, wie beispielsweise „Walking with Dinosaurs“ oder „Jurassic Park I bis III“. Nur ein verschwindend kleiner Bruchteil der Szenen sind tatsächlich CGI, die große Mehrzahl ist mittels konventioneller Methoden erzeugt. [Walking 2004] [Laniteri 2004] [Universal 2004] So existiert beispielsweise im Film „Jurassic Park I“ – entgegen der weit verbreiteten Ansicht – noch nicht einmal 6 Minuten Filmmaterial, in denen computergenerierte Organismen mitwirken. Manche neueren Filme wie z.B. „Godzilla 2000“ verzichten sogar vollständig auf computergenerierte Organismen. [Columbia 2000b,c] [Hubbard 2003]

1.1.1.1 Animatronik

Bei der Erstellung von Filmen, in denen Dinosaurier und andere paläontologische Tiere eine maßgebliche Rolle spielen, wird aufgrund ihrer Vielseitigkeit und ihrer frappierend lebensechten Anmutung besonders gerne *Animatronik* (englisch: *animatronics*) verwendet. *Animatronische Einheiten* sind dabei hochkomplexe hydropneumatische Puppen, die entweder autonom agieren oder ferngelenkt werden.



Abbildung 2: Filmstandbild mit einer animatronischen Einheit

Bild aus dem Film „Jurassic Park - The Lost World“, welches mit Animatronik hergestellt wurde. Szenen des Herstellungsprozesses derselben animatronischen Einheit befinden sich auf den beiden nächsten Abbildungen. [Universal 2004]

Die Herstellung von Animatronik geht dabei in mehreren aufeinander aufbauenden Schritten vonstatten. Nach einer bzw. mehrerer *Skizzen* wird zuerst eine *Zeichnung* erstellt, die als Vorlage für ein grobes Miniaturmodell dient, die *Maquette*. Die grobe Maquette besteht dabei aus einem leicht formbaren Material wie beispielsweise Gips, Wachs oder Knete und wird in einem Größenverhältnis von ca. 1/16 zur finalen Version geschnitzt. Mit dieser Maquette kann nun grob der Eindruck abgeschätzt werden, den die Animatronik beim Betrachter hinterlassen wird. Falls die erste Maquette-Version nicht zufrieden stellend ist, wird sie verworfen und eine neue Version erstellt. Dies wird solange wiederholt, bis die Maquette von der Grundform, der Mimik und der Farbe her zufrieden stellend erscheint. Danach wird eine größere 1/5-Maquetteversion erstellt, in der kleinere Details wie z.B. Hautfalten eingearbeitet werden. Erst dann wird die finale *Skulptur* in voller Größe erstellt. [Mejias 2004e] [BBC 2004c]



Abbildung 3: Der dreistufige Herstellungsprozess der animatronischen Skulptur

Beispiel für den dreistufigen Herstellungsprozess der Gussform einer animatronischen Einheit des Tyrannosaurus Rex aus den Film „Jurassic Park – The Lost World“. Vorne links ist die Maquette im Maßstab 1/16 zu sehen, in mittlerer Entfernung steht das 1/5-Modell und hinten in voller Größe die finale Skulptur, von der dann eine Gussform erstellt wird. [Mejias 2004e]

Von der finalen Skulptur wird eine *Gussform* erstellt, in die Latexschaum eingegossen wird. Für weniger viskose Körperteile kann auch mit Glasfaser verstärktes Epoxydharz verwendet werden. Parallel dazu wird das ausgeklügelte hydropneumatische Innenleben konstruiert.



Abbildung 4: Beispiel für das Innenleben von Animatronik

Beispiel für das Innenleben verschiedener animatronischer Systeme, die in den Stan Winston Studios für den Film „Jurassic Park - The Lost World“ hergestellt wurden. Zu sehen sind die technischen Innereien eines Velociraptors, eines Tyrannosaurus Rex (siehe auch vorheriges Bild der Hüllform), eines Stegosaurus, eines Pachycephalosaurus und eines Compsognathus. [Mejias 2004, a]

Ist die *Latexhülle* getrocknet und das *hydropneumatische Innenleben* fertig, werden die beiden Komponenten zusammengefügt und die entstehende Animatronik auf Bugs untersucht. Aufgrund dieses aufwändigen Verfahrens dauert der Herstellungsprozess einer einzigen animatronischen Einheit zwischen 10 und 24 Monaten. [Cinema 2004c] [Hsw 2004,2]

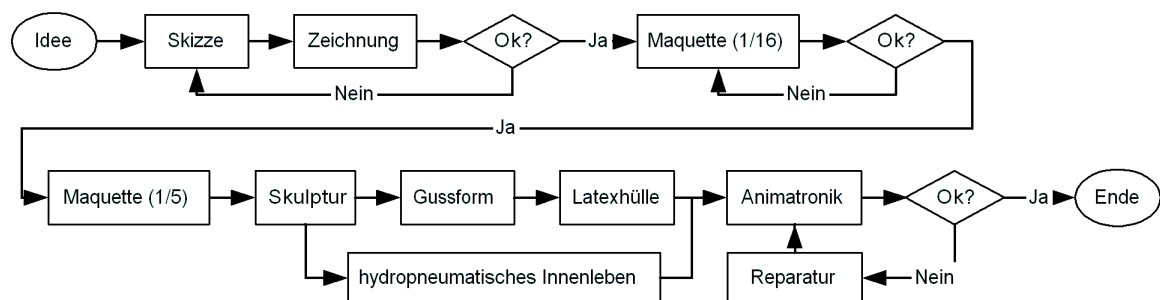


Abbildung 5: Flussdiagramm des Herstellungsprozesses für Animatronik

Horizontales Flussdiagramm für den Standardherstellungsprozess einer animatronischen Einheit. Ausgehend von der Idee wird Schritt für Schritt die Animatronik erstellt. Der gesamte Herstellungsprozess kann bis zu 2 Jahren dauern.

Animatronische Einheiten können dabei riesige Dimensionen annehmen, so wiegt der Tyrannosaurus Rex aus „Jurassic Park III“ 9 Tonnen und besitzt eine Motorenkraft von umgerechnet 300 Pferdestärken. [Cinema 2004b] Der Spinosaurus aus demselben Film wiegt sogar 12 Tonnen, hat dabei eine Länge von 13 Metern, die Gesamtlänge seiner intern verbauten hydraulischen Schläuche beträgt 671 Meter und er besitzt eine Kraft von 1000 PS [Hsw 2004,1]

1.1.1.2 Man in a Suit

Eine spezielle Animatronikversion, die bei größeren Tieren verwendet werden kann, ist die Integration eines oder mehrerer Menschen in die Hülle der animatronischen Einheit. Auf diese Weise gleicht die Animatronik einem hydropneumatischen High-Tech-Anzug und muss nicht mehr ferngesteuert werden. Der integrierte Mensch kann einige oder alle Bewegungen direkt mit seiner Muskelkraft ausführen. Eine solche Methode wird *Man in a Suit* genannt. Oft werden für solche Anzüge von der Körperstatur besonders geeignete Menschen gesucht, z.B. besonders große oder besonders zierliche Menschen. Es existieren auch *Men in a Suit* - Systeme, in denen mehrere Menschen Platz finden. [Columbia 2000c] Ein Beispiel für eine Man in a Suit-Produktion ist „Godzilla 2000“. [Henkel 2000] [Columbia 2000b] [Hubbard 2003]



Abbildung 6: Beispiel für Man in a Suit-Systeme

Herstellung der Man in a Suit-Version eines Pteranodons für den Film „Jurassic Park III“. [Mejias 2004, b]

1.1.1.3 Masken

Eine weitere Methode zur nicht-computerbasierten Generierung von Wesen sind *Masken*. Sie bestehen in der Regel aus einer optionalen *mechanischen Untermaske*, über die eine *Überhülle* aus Latexschaum und Silikon gestülpt wird. Die Hüllen werden zumeist durch einen Gipsabdruck des Trägers exakt von den Proportionen auf diesen abgestimmt. Schmutz, eingeflochtene Yak-Haare und künstliches Blut geben der Maske dann den letzten realistischen Schliff. [Filmlexikon 2004] Masken sind recht kostengünstige, schnelle und vielseitige Utensilien in der Filmindustrie und werden daher oft verwendet. Allerdings sind sie auf Organismen beschränkt, die in ihrer Grundform dem Menschen ähneln, wie z.B. die Affen aus „Planet der Affen“, die Orks aus der „Herr der Ringe“-Trilogie oder extraterrestrische Humanoide wie die Klingonen aus „Star Trek“.



Abbildung 7: Beispiel für den Herstellungsprozess einer mechanischen Maske

Ein Beispiel für drei Stationen des Herstellungsprozesses einer mechanischen Maske. Es handelt sich um die Maske des „Wildschweinmannes“ aus dem Film „The Island of Dr. Moreau“. Links: Untermaske mit beweglichem Unterkiefer. Durch die Kieferbewegungen des Trägers kann somit das Maul der Wildschweinemann-Maske geöffnet und geschlossen werden. Mitte: Übermaske aus Latex; Rechts: Fertige Maske in Aktion. Details wie geschickte Airbrush-Arbeiten und in die Übermaske eingeflochtene Yak-Haare ergeben den gewünschten Realismus. [Mejias 2004, c-e]

1.1.1.4 Puppen, Marionetten und Stop-Motion-Armaturen

Weitere Methoden der Filmindustrie, um Organismen auf die Leinwand zu bannen, sind *Marionetten* und andere *Puppen* sowie *Stop-Motion-Armaturen*.

Einfache Puppen können immer dann verwendet werden, wenn sich das entsprechende Wesen in der Filmsequenz nicht aus eigener Kraft bewegt, wie

beispielsweise schlafende, tote oder fallende Tiere. Beispielsweise wurde die Szene mit einem von der Klippe fallenden Mammut aus „Walking With Beasts“ mit einer einfachen wollenen Riesenpuppe erzeugt. Um den Fall realistisch aussehen zu lassen, muss die Puppe eine interne Gewichtsverteilung haben, wie sie das echte Tier besitzt.

Mit Marionetten können einfache Bewegungen erzielt werden, die allerdings nur dann überzeugend wirken, wenn der Bewegungsablauf einfacher Natur ist, wie beispielsweise leichtes Kopfheben. Sichtbare Nylonfäden werden nach dem Set aus den Aufnahmen herausretuschiert.

Stop-Motion-Armaturen sind Puppen mit einer deformierbaren Außenhaut und einem beweglichen Innenskelett. Die Hülle besteht meistens aus geschäumtem Latex, in seltenen Fällen auch aus Stoff, Kunststoff oder Knete. Das Innenskelett besteht aus Stahl und besitzt schleifende Spezialgelenke, die dazu dienen, die Gliedmaßen der Puppe in jede gewünschte Position zu bringen, in der sie dann stehen bleiben. Auf diese Weise kann man die Gliedmaßenstellungen schrittweise ändern und jeweils aufnehmen, wodurch Bild für Bild die Illusion einer flüssigen Bewegung entsteht. Mit der *Blue-Box-Technik* oder der Technik der *Dynamation* lassen sich auch andere Schauspieler in die Szene integrieren. [Brierton 1997] [Brierton 2004] [Skifi 2004]



Abbildung 8: Beispiel für eine Stop-Motion-Armatur

Dieses Beispiel einer Stop-Motion-Armatur für eine Gottesanbeterin besitzt ein stählernes Innenskelett (links) mit einer Hülle aus bemaltem Latexschaum (rechts). Photo courtesy of and © Tom Brierton. [Brierton 1997]

Die bekannteste Stop-Motion-Armatur dürfte „King Kong“ aus dem gleichnamigen Filmklassiker sein, aber auch andere Filme wie z.B. „The Lost World“, „The Black Scorpion“, „Might Joe Young“, „When Dinosaurs Ruled The Earth“, „The Monster From

20000 Fathoms“, sowie „Jason and the Argonauts“ und „Star Wars Episode V – The Empire Strikes Back“ benutzten diese Methode. [Skifi 2004] [Lukas 2000] Stop-Motion ist eine einfache und kostengünstige Technik. Allerdings ist sie zeitaufwändig und die resultierenden Bewegungsabläufe sind infolge des iterativen Eingabeprozesses nicht immer überzeugend. [Jackson 1997]

In den letzten zwei Jahrzehnten wurden Stop-Motion-Armaturen daher schrittweise aus den professionellen Studios verbannt und durch Animatronik und CGI-Organismen verdrängt. Bei Amateurfilmen werden sie allerdings auch heute noch verwendet. Für diese Zielgruppe existieren mittlerweile kostengünstige modulare Stahlbauteile, die zu beliebigen Armatureskeletten zusammengesetzt werden können. [Armaverse 2004]

In einer veränderten Form erleben Stop-Motion-Armaturen heute eine Renaissance, denn mit der Erfindung der *Direct Input Devices*, die ausführlich in Kapitel 2.3.5 behandelt werden, wurde die klassische Stop-Motion-Technik zu einem Computereingabegerät weiterentwickelt. [Bennett 1995] [Knep 2003]

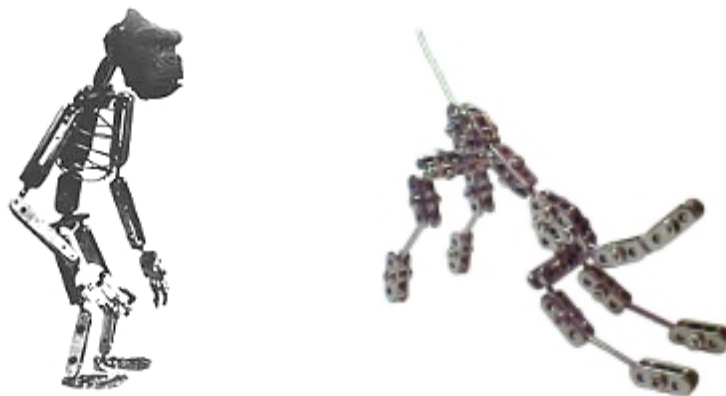


Abbildung 9: Beispiel für individuelle versus modulare Stop-Motion-Armaturen

Linkes Bild: Beispiel für ein individuell hergestelltes Innenskelett einer Stop-Motion-Armatur. Es handelt sich dabei um „King Kong“ aus dem gleichnamigen Filmklassiker. [Skifi 2004] Rechtes Bild: Beispiel für eine modular aufgebaute Stop-Motion-Armatur. Aus genormten Einzelteilen ist das Innenleben einer Hundepuppe erstellt worden. [Armaverse 2004]

1.1.2 Aufzeichnungscharakter bei CGI-Organismen

Bei näherer Betrachtung der wenigen Sequenzen mit CGI-Organismen, denen der Nimbus voraussetzt, angeblich vollständig computergeneriert zu sein, fällt auf, dass sie in der Regel weder am Computer modelliert noch per Computer animiert wurden, sondern dass Form und Bewegung allenfalls vom Computer aufgezeichnet wurden.

Um dies zu verstehen, wird im Folgenden der CGI-Standardprozess näher beleuchtet, der in allen bekannten Filmen wie beispielsweise „Jurassic Park I bis III“, den BBC-Serien „Terminator 2“, „Matrix Reloaded“ oder „Herr der Ringe“ verwendet wurde und noch immer die erste Wahl bei der Erstellung täuschend lebensechter Wesen ist: [Cyberware 1999] [BBC 2004b] [Silberman 2003, 1-4]

Anfangs wird ähnlich wie bei der Erstellung von Animatronik vorgegangen: Ausgangspunkt ist zuerst die *Idee* eines zu erstellenden Wesens. Die Charaktereigenschaften, herausragende körperliche Attribute und sonstige Besonderheiten werden im Rahmen dieser Idee detailliert beschrieben. Ein Künstler lässt daraufhin seiner Phantasie freien Lauf und erstellt mehrere, möglichst unterschiedliche *Skizzen* von Organismen, die dieser Idee gerecht werden könnten. Die beste Skizze wird – meist in einem kollektiven Entscheidungsprozess zwischen Regisseur, Artdirector, Designern und Künstlern – ausgewählt und davon eine detaillierte *Zeichnung* angefertigt, die als Vorlage für eine oder mehrere Versionen von *groben Maquettes* dient. Eine grobe Maquette wird als Vorlage für eine verfeinerte Maquetteversion verwendet. Es kann hier ebenfalls mehrere Durchgänge geben, bis die Details dieser *feinen Maquette* für alle Beteiligten zufrieden stellend sind. Die finale Maquette wird dann allerdings nicht weiter vergrößert, sondern es wird lediglich eine *detaillierte Skulptur* erstellt. Der gesamte Prozess der Maquette- und Skulpturherstellung wird dabei *Sculpting* genannt. Die Form dieser Skulptur wird per *3D-Scanning* in einen Computer eingelesen. (für eine ausführlichere Beschreibung von 3D-Scanning siehe Kapitel 2.2.2). Das Scannen der Form liefert in der Regel *Volumendaten*. (beschrieben in Kapitel 2.1.3) Die Volumendaten werden mit geeigneten Algorithmen in ein *Polygonnetz* umgewandelt. (beschrieben in Kapitel 2.1.1) Die Polygonanzahl wird mit geeigneten Verfahren *reduziert*, Scan-Fehler werden dann *korrigiert* und danach in einem *Texturierungsprozess* mit *Image-* und *Bump Mapping-Texturen* versehen. (beschrieben in Kapitel 2.1.1)

Das Polygonnetz kann temporär noch in *parametrische Flächenbeschreibungen* (beschrieben in Kapitel 2.1.2) umgewandelt werden, die helfen, die Oberfläche während der anschließenden *Animation* (beschrieben in Kapitel 2.3) abzurunden.

Für die Animation wird eine geeignete *Skeletthierarchie* erstellt bzw. eine existierende Skeletthierarchie ausgewählt und an den gewählten Organismus adaptiert, und mit der *3D-Skinning-Methode* (siehe Kapitel 2.5.1.2) werden die einzelnen Eckpunkte des Polygonnetzes mit verschiedenen virtuellen Knochen der Skeletthierarchie verknüpft. (siehe Kapitel 2.5.1.1)

Parallel zum 3D-Scanning erfolgt die Animation. Diese kann auf unterschiedliche Weise erfolgen, wie beispielsweise *Keyframing* (siehe Kapitel 2.3.1), *Pfadanimation* (siehe Kapitel 2.3.2), *Vorwärtskinematik* (siehe Kapitel 2.3.3.1) oder *Inverse Kinematik* (siehe Kapitel 2.3.3.2). In der High-Quality-Filmproduktion werden allerdings fast ausschließlich *Motion Capturing* (siehe Kapitel 2.3.4) oder *Direct Input Devices* (siehe Kapitel 2.3.5) verwendet, da sie die einzigen Techniken sind, die überzeugende Resultate liefern. Es erfolgt dann eine optionale *Korrektur und Modifikation der Bewegungsdaten* und eine *Übertragung der Bewegungsdaten auf die Gelenkpunkte des Skelettes*.

Nun wird die Sequenz mit dem virtuellen Organismus mit Raytracingverfahren *gerendert*. Da für jede Filmminute 1.440 Animationsframes berechnet werden müssen und Organismen in High-Quality-Produktionen durchaus weit über 100 Millionen Polygone haben können, kann sich das Rendering trotz Einsatz von teurer, performanter Hardware über Wochen hinziehen. Für den Renderingprozess müssen eventuelle parametrische Flächenbeschreibungen bei jedem Animationsframe wieder in ein Polygonnetz zurückverwandelt werden.

Sollte die Szenerie, in der sich der virtuelle Organismus bewegt, ebenfalls per Computer erstellt sein, so kann sie zusammen mit dem Organismus gerendert werden. Meistens werden jedoch gefilmte, reale Hintergründe verwendet und der Organismus nach dem Renderingprozess in die Szenerie eingefügt. Für diese *Integration des Organismus in die Szenerie* müssen manuell – Frame für Frame – Teile des Organismus herausretuschiert werden, falls Objektverdeckungen auftreten und Teile der Szenerie abgedunkelt werden, um Schattenwürfe zu simulieren, die der Organismus hervorruft.

Alternativ kann auch die reale Szenerie maßstabsgetreu *nachmodelliert* werden und der Organismus in die virtuelle Szenerie eingebunden werden. Nach dem Rendering wird die virtuelle Szenerie durch einen gleichmäßigen blauen Hintergrund ersetzt, um Filmaufnahmen der realen Szenerie mit dem Blue-Box-Verfahren einzufügen.

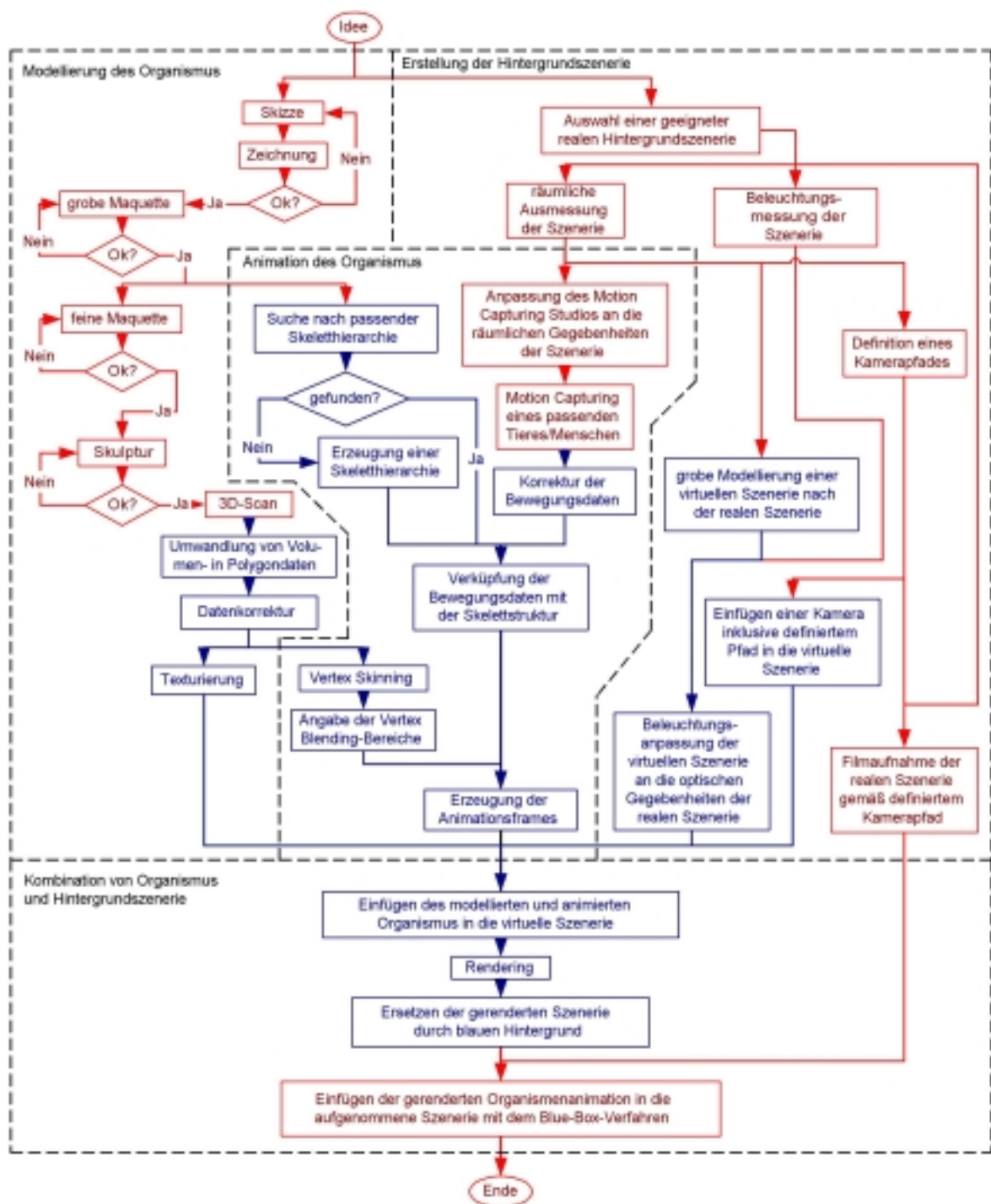


Abbildung 10: Standardprozess von CGI-Szenen mit Organismen

Flussdiagramm für den Standardprozess der angeblich computergenerierten High-Quality-Szenen mit Organismen in der Film- und Fernsehindustrie. Blaue Bereiche zeigen Arbeiten am Computer an, rote Bereiche dagegen nicht-computerunterstützte Arbeiten. Es ist ersichtlich, dass die Hauptarbeit bei der Modellierung und der Animation des Organismus (mit gestrichelten Linien umrandete Bereiche links oben und Mitte) außerhalb des Computers stattfindet. Daher ist die Bezeichnung CGI irreführend.

Wichtig ist, dass vor dem Rendering eine *Anpassung der Beleuchtungsparameter* an die Lichtgegebenheiten der realen Szenerie vorgenommen wird. Auch sollte die Unschärfe des virtuellen Objektes an die Fokussierung der Szenenkamera adaptiert werden, damit der Organismus nicht als Fremdkörper in der realen Szenerie wahrgenommen wird. Für diese Adaption der Renderingparameter sind speziell geschulte Beleuchtungsexperten verantwortlich. Theoretisch könnten die Beleuchtungsgegebenheiten und die Fokussierung in der aufgenommenen Szene gemessen werden, in der Praxis sind jedoch in einem langwierigen Trial-and-Error-Verfahren mehrere Rendering-Testläufe notwendig, bis der virtuelle Organismus sich optisch sinnfällig in die Szenerie einfügt. [BBC 2004b] [BBC 2004a] [Foley et al. 1996] [Uip 2004] [Watt 2000] [Rapido3D 2004c]

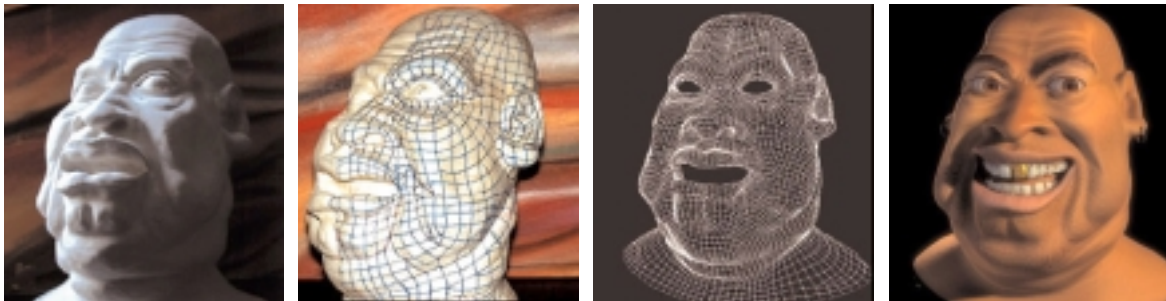


Abbildung 11: Beispiel für den Aufzeichnungscharakter von CGI-Organismen

Hier sind v.l.n.r. vier aufeinander folgende Stationen der Herstellung von Mitch Gates „Bubba“ zu sehen: 1: Die traditionell erstellte Gipskulptur; 2: Aufzeichnung der Scanlinien auf die linke Kopfhemisphäre der Skulptur; 3: Die linke Kopfhemisphäre wurde eingescannt und symmetrisch gespiegelt. Hier ist das Drahtgittermodell abgebildet. 4: Texturierter, animierter und gerenderter Kopf. [Immersion 2004b]

Es gibt bei manchen Filmen kleinere Abweichungen des oben skizzierten Prozessablaufes. Ein Beispiel ist z.B. das Muskelmonster „Hulk“ aus dem gleichnamigen Film, für das in manchen Szenen lebende Menschen statt einer Skulptur gescannt wurden, respektive zwei Bodybuilder für den Körper und den Schauspieler „Eric Bana“ für das Gesicht. Die drei gescannten Menschen wurden nach der Umwandlung in Polygondaten manuell zu einer Einheit zusammengefügt. Die meisten der Hulk-Szenen wurden allerdings ebenfalls per Maquette erstellt, wobei verschiedene Maquetten gescannt wurden, um den unterschiedlichen Posen des Hulks gerecht zu werden. [Animationarena 2004] [Gamma 2003] [Shepter 2004]

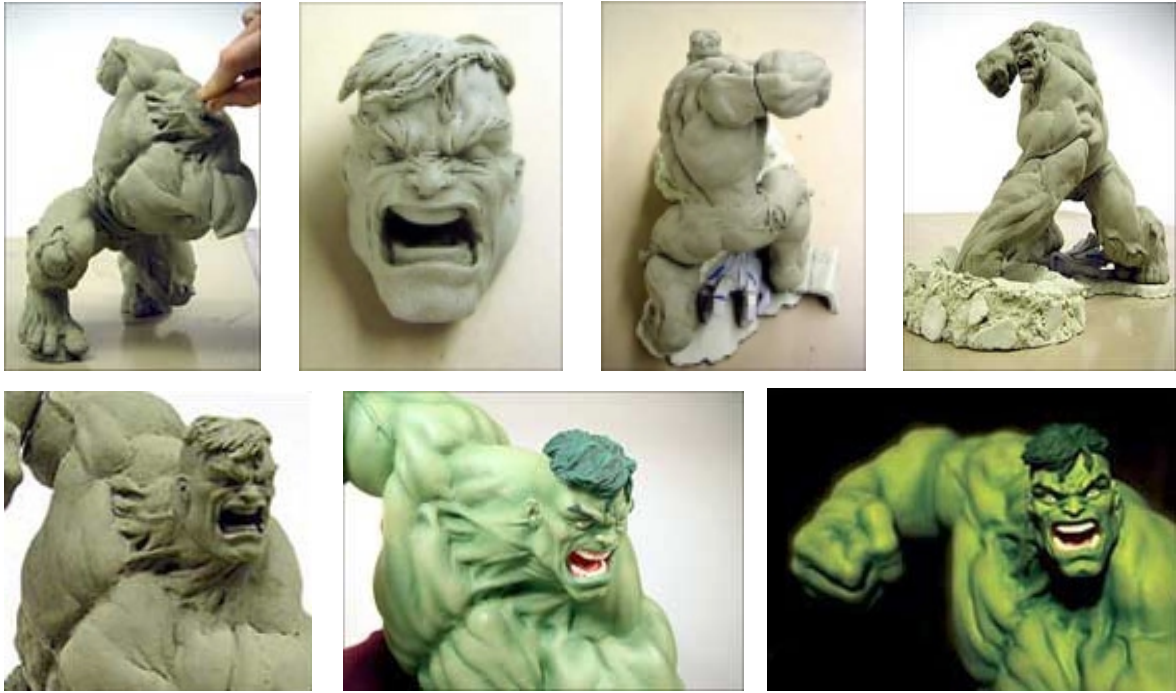


Abbildung 12: Stationen des Sculpting-Prozesses des Hulk

Einige Stationen des Sculpting-Prozesses für die Figur des Hulk aus dem gleichnamigen Film: Links oben: Sculpting des Körpers, wobei zwei Bodybuilder, die für den Künstler in Pose stehen mussten, als Vorlage dienten. Zweites Bild von links oben: Sculpting des Gesichts des Hulk, wobei versucht wurde, die Gesichtszüge dem Schauspieler „Eric Bana“ anzugleichen. Rechte Bilder oben: Kopf und Körper werden zusammengefügt, es entsteht die erste Version einer Maquette. Links unten: Einige Details werden noch hinzugefügt. Mitte unten: Der Hulk wird angemalt, als Vorlage für den späteren Texturierungsprozess. Rechts unten: Die Form des Hulk wird per 3D-Scanning in den Computer übertragen, die Volumendaten in Polygondaten umgewandelt und texturiert. [Shepter 2004]

Wie ersichtlich, ist die Herstellung von virtuellen Organismen in High-Quality-Filmen ein langwieriger und komplexer Prozess, der wiederum auf viele Unterverfahren zurückgreift, die detailliert in Kapitel 2 erklärt werden.

Wenn dieser Prozess analysiert wird, fällt auf, dass die entscheidenden formgebenden Schritte nicht am Computer erfolgen, sondern während der Arbeit an einer Maquette, also einem „echten“ und „handfesten“ Modell aus Knete oder Gips. Am Computer erfolgen lediglich kleinere Korrekturen von Scan- und Konversionsfehlern sowie die Texturierung.

Was für die Form der Organismen gilt, ist auch auf die Bewegung übertragbar: Auch hier stammt die Bewegungserzeugung direkt von einem Akteur, der entweder ein Schauspieler oder ein Tier bei Motion-Capturing-Systemen bzw. ein Puppenspieler bei

Direct Input Devices ist. Die aufgezeichnete Bewegung wird danach allenfalls am Computer modifiziert und dann auf das virtuelle Objekt übertragen.

Sowohl die Modellierung als auch die Animation sind daher im Grunde „aufgezeichnet“ und der Wirklichkeit „entliehen“ und können daher – wenn eine enge Definition benutzt wird – gar nicht als „computergeneriert“ angesehen werden.

Der allseits verwendete Begriff „Computer Generated Images“ ist daher irreführend, da implizit suggeriert wird, nicht nur die Bilder, sondern auch die auf den Bildern gezeigten Organismen seien per Computer generiert. Dies ist jedoch – wie gezeigt wurde – nicht der Fall. Es sollte daher eher von „Computer Recorded Organisms“, oder etwas genauer von „Computer Recorded and Modified Shape and Movement Data of Organisms for Image Generation“ gesprochen werden.

1.1.3 Aufwand und Nutzen

Ein weiterer Punkt, der gerne übersehen wird, ist der große Aufwand, der notwendig ist, um realistisch anmutende Organismen zu erschaffen, denn bis jetzt existieren noch keine intuitiven Methoden für die Organismengenerierung, die es Laien oder zumindest im Eiltempo ausgebildeten Graphikern erlauben würden, CGI-Organismen zu generieren. (siehe auch Kapitel 2)

Schon für die Modellierung der computergenerierten Organismen wird daher ein Heer von Spezialisten benötigt. Dieses Team umfasst Biologen, Bildhauer für die Erstellung der Maquetten, an komplexen 3D-Modellierungsprogrammen speziell geschulte Graphiker und Informatiker für die Spezialeffekte. Auch die darauf folgende Integration der Wesen in die Szene, die Ausleuchtung und die Animation erfordern viel Fachpersonal. Die schiere Menge an benötigten Spezialisten bindet in der Folge weitere Mitarbeiter für die Koordinationsarbeit, wie Administratoren und Projektmanager.

Schon für kostengünstig produzierte CGI-Filme mit Fantasiewesen, wie beispielsweise „Back to Gaya“ aus dem Jahr 2004 von Lenard Krawinkel aus Hannover, arbeitete schon ein Team von zehn Spezialisten, die während der gesamten Produktionszeit ausschließlich für die Beleuchtung von Szenen zuständig waren. Weitere acht Leute haben nur modelliert und 22 Leute nur animiert. [Jasper 2004]

Für die BBC-Fernsehserie „Walking with Dinosaurs“ wurden 13-15 Spezialisten alleine für die Computeranimationen benötigt. Zur Anzahl der anderen Spezialisten wird keine Angabe gemacht, aber sie dürfte weit mehr als 50 betragen. [BBC 2004b] Dies ist

allerdings sehr wenig im Vergleich zu Hollywood-Produktionen. Hier kann die Zahl derjenigen Mitarbeiter, die direkt in die Generierung computergestützter Organismen involviert sind, dreistellige Höhen erreichen. Beispielsweise waren für den Film „Matrix Reloaded“ weit über 500 Computergraphikspezialisten beschäftigt, allerdings nicht alle für die Organismengenerierung. [Silberman 2003, 1]

Zu diesem exorbitanten Arbeitsaufwand gesellen sich noch horrenden Anforderungen an die Hardware, da performante Methoden für Modellierung und Rendering noch nicht existieren:

Die massiven Anforderungen des Filmes „Ice Age“ aus dem Jahre 2002 wurden beispielsweise mit insgesamt 512 Compaq Alpha Servern DS10L mit dem Betriebssystem Compaq Tru64 UNIX geleistet, die in 13 Render Walls konfiguriert waren. Dazu kamen mehrere Compaq Alpha Server ES40 Systeme im Cluster-Verbund zum Einsatz, die mehr als drei Terabyte Speicherkapazität bereitstellten. [Zowack 2002]

Auch die BBC-Serie „Walking with Dinosaurs“, wo bei der Hardware an allen Ecken und Enden gespart wurde, benötigte immerhin mehrere „Silicon Graphics“ und ein großer Clusterverbund von Hochleistungs-PCs. [BBC 2004b]

Die „Herr der Ringe“-Filmtrilogie wurde mit einem ungleich höheren Aufwand an Hardware erstellt. So erforderten die CGI-Szenen mit den Orks, den Hobbits und den anderen Fabelwesen 150 SGI Oktane Workstations, 2 SGI Origin 2000 Workstations, 80 SGI Dual Prozessor-320er-Systeme und 230 Linux Workstations.

Aber nicht nur die Hardware-, sondern auch die Software-Anforderungen der Filmtrilogie waren exorbitant:

So wurden 51 GUI-Lizenzen und 100 Renderlizenzen der „Nothing Real Shake“-Software sowie 60 Voll-Lizenzen von Alias Wavefront Maya benötigt. Dazu kam eine unlimitierte Anzahl der „Massive“-Software für die kämpfenden Massen. [Lehane 2001]

Der Personalaufwand an Spezialisten zusammen mit den extremen Hardwareanforderungen erklärt auch die hohen Produktionskosten, die bei der Generierung von Organismen per Computer entstehen:

So kostete beispielsweise „Jurassic Park I“ über 63 Millionen Dollar, wobei die sechs Minuten Filmmaterial, in denen computergenerierte Organismen mitwirken, mit 45 Millionen Dollar am meisten zu Buche schlugen. [Zelluloid 1993]

Und der Film „Dinosaurs“ von Disney Enterprises verschlang sogar ca. 350 Millionen Dollar. Damit machten ihn die vielen computergenerierten Dinosaurier zum teuersten Kinderfilm aller Zeiten. [Windhorst 2000]

Zum Vergleich kann ein Film aus dem gleichen Jahr herangezogen werden, der aus Kostengründen vollständig auf konventionelle Methoden wie Man in a Suit und Stop-Motion-Devices gesetzt hat. Obwohl „Godzilla 2000“ mit 3,5 Milliarden Yen (umgerechnet ca. 38,5 Millionen Dollar) einer der teuersten Produktionen Japans ist, kostete er lediglich ein Bruchteil von „Dinosaurs“ – und dies bei ungefähr gleich langen Sequenzen mit Monstern und weitaus realistischeren Resultaten. [Hubbard 2003]

Interessant ist in diesem Zusammenhang, dass „Twister“-Regisseur „Jan de Bont“ in Hollywood ebenfalls einen entsprechenden „Godzilla“-Film plante, allerdings im Gegensatz zu den Japanern hauptsächlich mit CGI-Methoden. Er besorgte sich dazu von der Firma „Industrial Light and Magic Inc.“ einen Kostenvoranschlag. Doch nach einer ersten Kalkulation von ILM hätten alleine die CGI-Bilder des Godzillas mindestens 150 Millionen Dollar gekostet. Die weitere Filmplanung wurde daraufhin aufgegeben. [Hoffmann 1997]

Somit können von der Konzeption bis zur Modellierung eines computergenerierten Organismus als Faustregel 5-7 Millionen Euro veranschlagt werden. Für jede Minute Animation dieses Organismus kann mit weiteren 120.000-150.000 Euro gerechnet werden. Low-Budget-Filmproduktionen werden somit verhindert. An eine Spiele- und Filmproduktion am PC in „Heimarbeit“ ist gar nicht zu denken.

Bezeichnend für den Stand von computergenerierten Organismen ist, dass die Nachfolgef়ilme wieder verstärkt auf Animatronik, Man in a Suit und andere konventionelle mechanische Methoden setzen. Während beispielsweise die Gesamtlänge derjenigen Szenen, in denen CGI-Dinosaurier ihr Unwesen treiben, im Film „Jurassic Park I“ noch immerhin ganze 6 Minuten betrug, schrumpfte die Länge bei „Jurassic Park III“ auf 3,2 Minuten, das entspricht umgerechnet nur ca. 2,2% der gesamten Filmlänge und 14% aller Dinosauriersequenzen. [Ball 2002]

1.1.4 Qualitätsaspekte

Alle Filme, in denen realistisch anmutende CGI-Organismen vorkommen, zeigen entweder ausgestorbene Tiere oder imaginäre Wesen – also stets Organismen, bei denen der Betrachter keine eigenen Erfahrungswerte mitbringen kann: In „Jurassic Park I-III“, „Walking with Dinosaurs“ und „Walking with Beasts“ werden beispielsweise stets paläontologische Tiere präsentiert. Bei „Alien 3“, den „Star Wars“ – Episoden und „The Future is Wild“ sind es dagegen zukünftige Wesen, die einer blühenden Phantasie entspringen.

Bei dieser ausschließlichen Versteifung auf die ferne Vergangenheit und die imaginäre Zukunft liegt der Verdacht nahe, dass die Aussparung der gegenwärtigen Fauna der Intention der Produzenten entspringt. Der Betrachter würde vermutlich ein computergeneriertes Tier, mit dessen Vorbild er Erfahrung sammeln konnte, sofort als künstlich entlarven.

Um diesen Verdacht zu erhärten, wird im Folgenden exemplarisch „Jurassic Park I“ von Steven Spielberg näher betrachtet. Dieser Film ist als Referenzfilm geradezu zwingend, da er gemeinhin als Musterbeispiel für photorealistische Organismen via CGI gilt. [Watt 2002] [MMM 2004] Er hat 1993 mit der Präsentation der ersten realistisch anmutenden computergenerierten Organismen einen Paradigmenwechsel in der Filmbranche herbeigeführt. Die damalige Presse überschlug sich mit Lobeshymnen und Experten sehen auch heute noch „Jurassic Park I“ als einen Meilenstein der Computergraphik an, dazu einige Zitate:

„Jurassic Park von Spielberg läutet einen historischen Wendepunkt in der Geschichte der Spezialeffekte ein. Die Dinosaurier scheinen wahrhaft realistisch.“
Académie de Versailles 2004 [Versailles 2004]

„Man meint fast, Steven Spielberg konnte echte Dinos für diesen Film unter Vertrag nehmen.“
Filmzeitschrift „Zelluloid“, 1.10.1993 [Zelluloid 1993]

„Jurassic Park“ ist wahrscheinlich mit seinen inzwischen drei Folgen der bekannteste Film, der durch computergenerierte 3D-Animation hervorsteicht. Die Dinosaurier, vor ca. 50 Millionen Jahren ausgestorben, wurden in realistischer

Weise zum Leben erweckt. Besonders überzeugend waren, neben der großen Detailgenauigkeit, die Bewegungen der Figuren“

Online-Zeitschrift „Wissenswandel“ 30.01.2002 [Steiner 2002]

„Mit Jurassic Park ist eine Wende in der Filmindustrie und der Produktion von Tricksequenzen zu verzeichnen.“

Allen Watt im Buch „3D-Computergraphik“ 2002 [Watt 2002]

„Nach all den Action-Filmen wurde mit Steven Spielbergs Jurassic Park eine neue Qualität erreicht ... der Erfolg des Films und die Ununterscheidbarkeit der künstlichen von „echten“ Bildern läuteten einen Siegeszug (der CGI) ein.“

Heise.de- Telepolis 27.02.1997 [Hoffmann 1997]

“Obwohl die Produktion von Jurassic Park schon über ein Dutzend Jahre her ist, gab es seitdem kein Film mehr, in dem überzeugendere computergenerierte Lebewesen präsentiert wurden. Noch heute wird jedwede CGI an diesem Meisterwerk gemessen.”

Miami Movie Magazine (M.M.M.) 21.09.2004 [Warner 2004]

“None of us expected that ILM would make the next quantum leap in computer graphics, at least not in time for this picture. (Erste Testbilder für Spielberg von der Firma Industrial Light and Magic) But there it was: a living, breathing dinosaur, more real than anything Harryhausen or Phil Tippett (zwei Stop-Motion und Animatronikexperten) had ever done in their careers. At the showing, Phil groaned and pretty much declared himself extinct.”

Steven Spielberg in einem Interview vom 20.02.1995 [Bennet 1995]

Der erste computergenerierte Dinosaurier, mit dem der Zuschauer in „Jurassic Park I“ konfrontiert wird, ist ein Brachiosaurus, der zugegebenermaßen auf den ersten Blick durch seinen Photorealismus beeindruckt. Einen modifizierten „Turingtest für Dinosaurier“ hätte er möglicherweise bestanden.

Etwas ernüchternd ist allerdings der Eindruck, wenn ein Standbild während der Bewegung des Brachiosaurus geschossen wird. (siehe Abbildung 13 links) Nun fällt auf, dass sich im Bereich des Brustkorbes hinter dem rechten Vorderbein eine unnatürliche

konkave Deformation befindet, die anatomisch nicht korrekt ist. In Kapitel 2.5.1.2 wird erklärt, wodurch dieses hässliche Phänomen entsteht, welches unter Computergraphikern als „Syndrom des kollabierenden Ellenbogens“ bekannt ist.

Damit dem Betrachter diese unschöne Delle nicht auffällt, wurde dem Brachiosaurus eine seltsame Gewandtheit verpasst. Er stellt sich innerhalb eines Bruchteils einer Sekunde galant auf seine Hinterpfoten und verhält sich dabei eher wie eine grazile Gazelle als das 50-Tonnen schwere Monstrum, welches er tatsächlich war.

Es soll als zweites diejenige Szene aus „Jurassic Park I“ betrachtet werden, in der die Gallimimus-Herde über die Hügel rennt: Die Szene wirkt auf den Betrachter frappierend realistisch und beeindruckend. Wenn jedoch diese Szene kritisch „unter die Zeitlupe genommen“ wird, wird erkenntlich, dass den Tieren ein übertriebenes Motion Blur verpasst wurde, so dass deren Form gleichsam bis zur Schemenhaftigkeit verwischt wurde. (siehe Abbildung 13 rechts) Somit wird vermieden, dass der Betrachter mit störenden Details wie z.B. mit dem fehlenden Muskelspiel oder mit der biologisch inkorrekten „Halssteifigkeit“ konfrontiert wird.



Abbildung 13: Beispiele für nicht überzeugende Resultate bei CGI-Organismen

Screenshots zweier berühmter Szenen aus „Jurassic Park I“ mit CGI-Organismen, welche biologisch inkorrekte Formen bzw. Deformationen haben (rote Pfeile): Links: Die Form des Tetrapoden ist biologisch inkorrekt. Besonders deutlich wird dies hinter dem rechten Vorderbein, wenn beim Laufen eine unnatürliche „Brustdelle“ entsteht. [Spielberg 2004] Rechts: Die Gallimimusherde wurde mit einem übertriebenen Motion Blur versehen, so dass die eigenartige Steifheit der Hälse dem Beobachter nicht auffällt. [Finkelstein 2002]

Was hier als realistisch empfunden wird, kann also keinesfalls die biologische Stringenz oder Detailreichtum einzelner Gallimimus-Tiere sein, sondern der frappierende Realismus entsteht lediglich durch das sinnfällige Zusammenspiel von lauten Trampelgeräuschen und Gerenne der Herdentiere. Beide sind jedoch der Natur gleichsam entliehen, da sie jeweils von einer galoppierenden Straußenherde aufgenommen wurden. [Uip 2004]

Auch in den anderen Szenen von „Jurassic Park I“ werden die Details der CGI-Organismen kaschiert. Es gibt an keiner Stelle eine Nahaufnahme (die Close-Ups sind durchweg mit Animatronik oder Puppen hergestellt), die Tiere bewegen sich entweder im nächtlichen Dunkel und dazu im heftigen Regen (T-Rex-Szenen mit Jeep), ihre Gliedmaßen sind durch Werkbänke aus Kantinen verdeckt (Szenen mit dem Velociraptoren und den Kindern in der Kantine) oder sie bewegen sich derartig schnell, dass nichts Genaues erkennbar ist (springender Velociraptor).

Diese Kaschierung der Unzulänglichkeiten von computergenerierten Organismen ist nicht nur auf den ersten Film der „Jurassic Park“-Serie beschränkt, so existiert z.B. weder in „Jurassic Park II oder III“ noch in den paläontologischen BBC-Serien irgendeine Szene, in der ein Close-Up eines computergenerierten Dinosauriers gezeigt wird, da ansonsten der Betrachter sofort den Dinosaurier als künstlich entlarven würde. [Hsw 2004, 8] [BBC 2004a] [BBC 2004c]

Es existieren zwar durchaus Filme, in denen CGI-Dinosaurier und andere computergenerierte Tiere in Nahaufnahmen gezeigt werden, z.B. Disneys „Dinosaurs“, „Findet Nemo“, „Toy Story 1+2“, „Ice Age“, „Shark Tale“ und „Garfield – The Movie“, aber hier zeigt die Computergraphik ihr wahres Gesicht. Die präsentierten Tiere sind zweifelsohne beeindruckend, aber sie sind eben weit davon entfernt, biologisch korrekt oder gar lebendig zu wirken. (siehe Abbildung 14) [Leong 2000]

Ergo lässt sich konstatieren, dass es noch nicht möglich ist, lebensechte virtuelle Lebewesen per Computer herzustellen. Rolf Krämer, Festival-Programmchef des 7. Europäischen Festivals für digitale Film-Produktion (eDIT/VES) in Frankfurt am Main, brachte es dort im September 2004 auf den Punkt. Er resümierte: “Die Dynamik des Körpers kriegt er (der Computer) noch nicht perfekt hin.“ [Maier 2004a+b]



Abbildung 14: Beispiel für computergenerierte Tiere in Close-Ups

Zwei Beispielbilder aus dem Film „Dinosaur“ von „Disney Enterprises Inc.“ aus dem Jahre 2000, welche vollständig computergenerierte Tiere in Nahaufnahmen zeigen. Die Tiere wirken dabei weder realistisch noch lebendig. (© Disney Enterprises Inc. – All Rights Reserved) [Leong 2000]

1.1.5 Fehlende Interaktivität

Die photorealistischen Bilder von CGI-Organismen sind in ihrer Verwendung alleine auf teure Filmproduktionen beschränkt. Mit ihrer ausschließlichen Fokussierung auf High-Quality-Movies decken realistische CGI-Organismen damit nur einen Bruchteil der möglichen und notwendigen Anwendungen ab. Unberücksichtigt bleiben nicht nur der Bereich der Low- und Mid-Budget-Filme, sondern auch insbesondere alle Anwendungen, die interaktive Methoden erfordern. Dazu zählen beispielsweise spielerische Anwendungen, e-Learning-Tools, Trainingssimulatoren, interaktive Museumsexponate, Augmented-Reality- und Mixed-Reality-Applikationen für die Medizin, Chirurgische Planungshilfen, Software für die Optimierung von Rehabilitierungsmaßnahmen sowie biologische Analyseprogramme.

Zwar gibt es durchaus auch für diese Bereiche verschiedene interaktive Methoden für dreidimensionale virtuelle Organismen (siehe Kapitel 2 im Allgemeinen und Subkapitel 2.5 im Speziellen), aber diese sind in ihren Resultaten weit davon entfernt, realistisch zu wirken. Darüber hinaus sparen sie die meisten anatomischen Aspekte von Organismen aus, beispielsweise ein biologisch stringentes Wachstum oder korrekte Muskelflexionen.

Diese Beschränktheit wird bei der Forderung nach neuen alternativen Methoden der Generierung von Organismen oft vergessen.

1.1.6 Zusammenfassendes zur Motivation

So beeindruckend und frappierend lebensecht die Organismen aus Film und Fernsehen auch scheinen mögen, so ernüchternd ist doch die Realität der Computergraphik.

Denn wird einmal hinter die Kulissen von Hollywood geschaut, bleibt von den Organismen nicht mehr viel übrig, außer einem Haufen von Knetmasse, vertuschten Brustdellen und eine „gallimimosenhafte“ Halssteifigkeit, die an die Leinwand gebunden ist. Obendrein verschlingen diese fragwürdigen Produkte noch Zigmillionen Euros.

Bis heute ist die Generierung von Organismen ein vernachlässigtes Stiefkind der Computergraphik geblieben, welches immer noch in den viel zu engen Kinderschuhen steckt. Die vorliegende Arbeit hat das Ziel, zur Verbesserung dieser mangelhaften Situation beizutragen.

Die stiefmütterliche Behandlung von Tieren und anderen Organismen in der Computergraphik liegt sicherlich an ihrem komplizierten Wesen. Die Natur lässt sich eben nur schwer bändigen, und dies gilt nicht nur für die Velociraptoren aus „Jurassic Park“ sondern für alle Sorten von Organismen, von der gemeinen Stubenfliege angefangen bis hin zum mächtigen Blauwal. Sie alle haben gemeinsam, dass ihre anatomische Strukturen aus unzähligen miteinander interagierenden Einzelkomponenten bestehen, die von ihrer Interdependenz, ihrer Dynamik und ihrer Topologie her die Komplexität eines Computers bei weitem übertreffen.

Bei den eklatanten Mängeln von computergenerierten Organismen nimmt es kein Wunder, dass Maskenbildner und Experten für Animatronik ihren Job nicht gefährdet sehen. [BBC 2004] [Bennet 1994]

Die Motivation zur vorliegenden Arbeit ist, einen Beitrag zur Computergraphik zu leisten, der gezielt die angesprochenen Probleme aufgreift und die bisherigen Möglichkeiten in wesentlichen Strategien unter Beachtung ganzheitlicher Strukturen verbessert.

1.2 Ziele und Durchführung der Arbeit

Das Hauptziel dieser Arbeit ist die Entwicklung eines effektiven und intuitiven neuen Softwareverfahrens zur Generierung von organischen Objekten. Dabei soll die Generierung möglichst viele Teilaspekte als Gesamtkonzept umfassen, vornehmlich die Modellierung, die Animation, die Bewegungssimulation und die Visualisierung, wobei durch diesen holistischen Ansatz indirekt die Forderung nach Interaktivität und somit Echtzeitfähigkeit impliziert wird.

Mit organischen Objekten ist die gesamte Bandbreite von Organellen über Zellen, Zellgeflechten, Organteilen, gesamten Organen, anatomischen Einheiten bis hin zu Gesamtorganismen und Gruppen derselben gemeint. Der Fokus soll dabei auf der Generierung von Tieren liegen. Pflanzen sollen in der vorliegenden Arbeit nicht behandelt werden, da die Flora gegenüber der Fauna ein anderes Anforderungsspektrum beinhaltet. So ist ihr Verzweigungsgrad und ihre Formkomplexität in der Regel höher, jedoch die interdependente Formdynamik geringer. Dies schließt natürlich nicht aus, dass mittels des Softwareverfahrens auch Einzelobjekte von Pflanzen realisiert werden können.

Somit kommen als Gesamtorganismen entweder Tiere oder bedingt auch Menschen in Frage. Menschen stellen dabei besondere Herausforderungen an den Realismus der Computergraphik, da unser visuelles System naturgemäß besonders auf diese Spezies getrimmt ist. Die Generierung von Gesamtorganismen soll nicht nur auf rezent existierende organische Objekte beschränkt sein, sondern sowohl paläontologische Organismen, wie z.B. Dinosaurier, Ammoniten etc. als auch Fantasietiere wie beispielsweise Drachen, Wolpertinger oder Elwetritsche umfassen. (wenn in der vorliegenden Arbeit von „Organismen“ gesprochen wird, sind diese organischen Objekte vereinfachend mit eingeschlossen)

Diese Vielseitigkeit des zu entwickelnden Verfahrens hinsichtlich der Anwendbarkeit auf mannigfaltige Wesen soll ein breites Applikationsspektrum eröffnen, während mit der gleichzeitigen Forderung nach Echtzeitfähigkeit die Praxistauglichkeit garantiert wird.

Der erste Schritt bei der Entwicklung neuer Software-Verfahren zur Generierung von organischen Objekten ist eine umfangreiche State-of-the-Art-Analyse, bei der bestehende Verfahren nicht nur auf ihre grundsätzliche Tauglichkeit geprüft werden, sondern

zusätzlich ihre jeweiligen Potentiale für Verbesserungen, respektive Erweiterungen, Kombinationen mit anderen Verfahren oder Modifizierungen evaluiert werden.

Zuallererst müssen dabei die Standardverfahren zur Generierung von Organismen in der Filmindustrie analysiert werden, da sie die aktuelle Messlatte für virtuelle Organismen abstecken, aber daneben soll auch eine umfassende Recherche unterschiedlicher Verfahren zur interaktiven Organismengenerierung durchgeführt werden, wobei naturgemäß die vorliegende Arbeit trotz aller Bemühungen keine Gewähr auf hundertprozentige Vollständigkeit der Rechercheergebnisse bieten kann und die mit der Recherchepräsentation verbundene Analyse immer auch etwas subjektiv geprägt sein wird.

Ein wichtiger Teilaspekt sowohl innerhalb der Modellierung als auch bei der Simulation von Organismen ist die geeignete Spezifikation von Positionen innerhalb eines Organismus. Dies scheint auf den ersten Blick nebensächlich und trivial, bei näherer Betrachtung erweist sich diese Positionsdefinition jedoch als grundsätzliches und komplexes Problem. Schließlich sind die „üblichen“ Positionsdefinitionen in lokalen kartesischen Koordinatensystemen weder invariant zum Entwicklungsstatus eines Organismus, wie z.B. dem Wachstums- oder dem Evolutionsstadium, noch zum Bewegungsstatus wie der Gelenkstellung, dem mimischen Ausdruck oder der aktuellen Konstellation der Muskelkontraktionen. Somit kann schon ein einziges Augenzwinkern oder ein Zucken des kleinen Fingers unter Umständen eine Positionsdefinition unbrauchbar machen.

Die Übertragbarkeit von Modellierungsarbeiten von einem Organismus auf einen anderen wird dadurch erschwert oder sogar verhindert, die Simulation eines Organismus kann verfälscht werden.

Eine wichtige Aufgabe besteht daher darin, Positionsdefinitionen zu entwickeln, die diese Nachteile umgehen, sich also möglichst invariant und robust gegenüber Statusveränderungen verhalten. Hier sind neue Ansätze gefragt. Eng verknüpft mit der Lösung des Positionsdefinitionsproblems ist naturgemäß die für biologische Objekte geeignete Definition von Arealen, Volumina, Distanzen, Strecken und Feldern.

Geeignete Positionsdefinitionen bilden die Grundvoraussetzung für die Behandlung von Organismen, welche Wachstumsprozessen oder evolutionären Veränderungsprozessen unterworfen sind, denn die „funktionalen morphischen Veränderungen“ eines Organismus, die durch Wachstumsprozesse oder gedachte evolutionäre Topologiemodifikationen hervorgerufen werden, stellen eine besondere Herausforderung an die Computergraphik dar, da sie nicht mit einfachen Freiformdeformationen oder optischem Morphing

biologisch stringent visualisiert werden können. Die Ontogenese und die Phylogenese eines Organismus spannen dabei zusammen ein komplexes zweidimensionales Aktionsfeld auf, welches wiederum mit den vier Dimensionen der Raumzeit interagiert.

Eine weitere Herausforderung ist die Beherrschung der interdependenten Komplexität von Organismen, die durch die Forderung nach Interaktivität und Echtzeitfähigkeit quasi einer Quadratur des Kreises gleichkommt.

Das zu entwickelnde Softwareverfahren soll schließlich als Proof of Concept prototypisch implementiert werden, um seine Praxistauglichkeit zu beweisen.

Zu dieser Arbeit gehört auch ein kurzer Blick über den Tellerrand der Computergraphik hinaus, um konventionelle sowie potentielle Softwareverfahren zur Generierung von Organismen in ihrem erweiterten Kontext zu betrachten. Dies schließt vor allem eine allgemeine Abschätzung von zukünftigen Computerentwicklungen, vornehmlich mittels Moore's Law und seiner Verallgemeinerungen ein. Dieser metatheoretische Blick gestattet uns, die grundsätzliche zukünftige Verwertbarkeit von Verfahren zur Organismengenerierung zu evaluieren, potentielle Verwertungsmöglichkeiten in der Zukunft zu extrapolieren und somit die Wichtigkeit neu entwickelter Verfahren abzuschätzen.

1.3 Gliederung der Arbeit

Im **Kapitel 1** werden neben einer allgemeinen Motivation auch die Ziele der vorliegenden Arbeit umrissen. Die Motivation wird dabei mit Hilfe einer State-of-the-Art-Analyse der Organismengenerierung in der Filmindustrie durchgeführt, um das Interesse der Leserin bzw. des Lesers zu wecken.

Kapitel 2 befasst sich mit den informationstechnischen Grundlagen, die notwendig sind, um diese Arbeit zu verstehen. Zusätzlich werden konventionelle Verfahren zur Generierung von Organismen beschrieben. Dieses Kapitel beinhaltet daher den Hauptteil der State-of-the-Art-Analyse. Es werden dabei verschiedene Repräsentationen der 3D-Geometrie vorgestellt, die für biologische Strukturen geeignet sind und konventionelle Modellierungsmethoden und Formerfassungsmethoden erläutert, die für die Konstruktion von Organismen nützlich sind. Dann werden Animations- und Bewegungserfassungsmethoden wie unter anderem Keyframing, Pfadanimation Kinematik, Motion Capturing und Direct Input Devices in ihren verschiedenen Variationen behandelt.

Auch werden numerische Methoden zur Simulation biomechanischer Prozesse behandelt, denen ein biologisches oder zumindest biologienahes Modell zugrunde liegt.

In **Kapitel 3** werden die Defizite der in Kapitel 3 analysierten und bewerteten Verfahren zusammengefasst. Daraus wird eine Anforderungsanalyse erstellt, die helfen soll, ein neues Softwareverfahren für Organismen zu generieren.

Kapitel 4 stellt den konzeptionellen Kern dieser Arbeit dar. Hier wird ein Gesamtkonzept für die Generierung dreidimensionaler virtueller Organismen entwickelt. Aufbauend auf der neuen Definition eines Koordinatensystems, welches für Organismen geeignet ist, wird ein holistisches Verfahren vorgestellt, welches Visualisierung, Modellierung, Animation und Simulation von Organismen vereint.

Kapitel 5 befasst sich mit Implikationsaspekten, welche bei der konkreten Realisierung relevant sind. Dazu gehört auch die Wahl eines geeigneten Szenengraphen bzw. Szenenmanagers mitsamt Basis-Graphik-API. Auch die Wahl einer geeigneten Programmiersprache mit hilfreichen Programmierertools wird behandelt. Die Grobarchitektur und die wichtigsten Schnittstellen des realisierten Software-Frameworks werden ebenfalls erläutert.

In **Kapitel 6** werden konkrete Anwendungsbeispiele der Konzepte auf der Basis der im Kapitel 5 beschriebenen Implementierung vorgestellt. Dafür wird die Nutzung des neuen Softwareverfahrens in sieben Projekten exemplarisch aufgezeigt, die von ihrem Anforderungsspektrum eine hohe Variationsbreite aufweisen. Die Projekte werden jeweils kurz mit ihren Hintergründen beschrieben. Auf der Grundlage der Anwendungsbeispiele erfolgt eine Bewertung der in dieser Arbeit vorgestellten Konzepte.

Kapitel 7 fasst die Arbeit zusammen und gibt einen Ausblick auf die Zukunft. Es werden zudem weitere Verbesserungsmöglichkeiten für das in Kapitel 4 vorgestellte Konzept als auch für die in Kapitel 5 und 6 vorgestellte konkrete Realisierung diskutiert und es wird ein Ausblick auf mögliche zukünftige Anwendungen der vorgestellten Konzepte gewagt. Darüber hinaus wird die vorliegende Arbeit kurz in ihrem technologischen Kontext beleuchtet, um ihre derzeitigen und zukünftigen Anwendungsmöglichkeiten und somit ihre zukünftige Bedeutung besser einzuordnen. Beendet wird die Arbeit mit einer kurzen Evaluation auf Basis der Anforderungsanalyse und einer kurzen Zusammenfassung.

Kapitel 8 enthält das Literaturverzeichnis.

Im **Anhang** stehen neben diversen Zusatzinformationen der Überblick über eigene Publikationen, Vorträge, organisierte Workshops, Softwareprojekte, Exponate, betreute Studien- und Diplomarbeiten, eine Darstellung der Lehrbeteiligungen sowie der Lebenslauf.

2 Grundlagen und State-of-the-Art-Analyse

*I have always wished that my computer
would be as easy to use as my telephone.*

My wish has come true.

I no longer know how to use my telephone.

Bjarne Stroustrup, Entwickler der
Programmiersprache C++

In diesem Kapitel werden alle Verfahren behandelt, die im Zusammenhang mit der Modellierung, der Animation oder der Simulation von Organismen stehen.

Informationstechnische und mathematische Grundlagen, die zum Standardrepertoire eines Informatikers gehören, werden jedoch als bekannt vorausgesetzt. Hierzu gehören beispielsweise die verschiedenen Interpolationsverfahren, das Rechnen mit Transformationsmatrizen oder auch die Arbeitsweisen der Renderingalgorithmen. Sollten hier Wissensdefizite bestehen, wird auf folgende Literatur verwiesen: [Foley et al. 1996] [Watt 2000]

Zunächst werden in Kapitel 2.1 die wichtigsten Grundarten der Geometrie-Repräsentationen erklärt, die in der Computergraphik verwendet werden. Zwar stellen sie einen Grenzfall zwischen informationstechnischem Grundlagenwissen und computergraphischem Spezialwissen dar; da sie jedoch letzten Endes als visuelle Basis für alle Modelle von computergenerierten Organismen fungieren, ist ihre Behandlung unumgänglich, denn unabhängig davon, ob es sich um eine reine Visualisierung, ein Modellierungsverfahren oder eine komplexe Simulation eines Organismus handelt, muss stets auf die implizit gewählte Geometrie-Repräsentation Rücksicht genommen werden.

In den Kapiteln 2.2 und 2.3 werden verschiedene Form- und Bewegungserfassungs- als auch Animationstechniken behandelt, die für die Generierung von Organismen grundsätzlich geeignet sind. Modellierungs- und Animationstechniken, die ausschließlich für den technischen Bereich konzipiert wurden und nur für starre Körper geeignet sind, bleiben unberücksichtigt.

Die Basistechniken für die Simulation der Anatomie stehen in Kapitel 2.4. Sie dienen der Berechnung von Bewegungen, Wachstumsprozessen sowie evolutionären

Formveränderungen. Es werden in diesem Zusammenhang auch geläufige hierarchische Repräsentationen von Organismen beleuchtet.

Auf diesen Basismethoden bauen die anatomischen Simulationsmodelle auf, um die es im Kapitel 2.5 geht.

Die einzelnen Beschreibungen der verschiedenen computergraphischen Verfahren schließen jeweils mit einer analytischen Bewertung hinsichtlich ihrer Einsatzfähigkeit zur Generierung und Behandlung von dreidimensionalen virtuellen Organismen ab. Die implizite State-of-the-Art-Analyse mündet im darauf folgenden Kapitel 3 in einer gezielten Defizit- und Anforderungsanalyse.

2.1 Geometrie-Repräsentationen

Durch *Geometrie-Repräsentationen* werden die grundsätzlichen Möglichkeiten und Restriktionen einer dreidimensionalen Visualisierung abgesteckt. Sie stellen daher einen Eckpunkt aller computergraphischen Methoden dar. Oftmals steht im Zusammenhang mit einer Geometrie-Repräsentation auch eine ihr eigene Modellierungstechnik, die in diesem Fall ebenfalls zu besprechen ist.

Im Folgenden wird sich auf Repräsentationen beschränkt, die für biologische Strukturen grundsätzlich sinnvoll erscheinen, wie *Polygonnetze*, *Parametrische Flächen*, *Billbaords*, *Punkt- und Linienrepräsentationen*, *Volumendaten*, *L-Systeme* und *Blobby Objects*. Andere Geometrie-Repräsentationen, wie z.B. die *Constructive Solid Geometry*, *Sweep-Körper*, *implizite Körperbeschreibungen*, *fraktale Beschreibungen*, *IFS* oder *deformierbare geometrische Primitive*, die eher für den technischen oder den künstlerischen Bereich geeignet sind, bleiben dagegen unberücksichtigt oder werden nur am Rande gestreift.

2.1.1 Polygonnetze

Gängige *3D-Basis-APIs*, wie *Direct3D*, *Glide*, *Merlin*, *OpenGL* oder *RenderWare* arbeiten mit *Polygonen*. [Engel 2002] [Datapath 2001] [OpenGL 2004]

Die Eckpunkte der Polygone werden *Vertices* genannt. (Singular: *Vertex*). Berühren sich mehrere Polygone dergestalt an den Kanten, dass sie dadurch eine netzartige Oberflächenstruktur bilden, wird auch von *Polygonnetzen*, *Meshs*, *Polygone Meshs* oder *Skin-Meshs* gesprochen.

Theoretisch könnten Polygonnetze aus Polygonen mit einer beliebigen Eckpunkt- und Kantenanzahl $n > 2$ bestehen. In der Praxis haben sich aber – vor allem in den neueren 3D-APIs – Polygonnetze aus Dreiecken durchgesetzt, da sich jedes beliebige Polygon in Dreiecke zerlegen lässt und bei Dreiecken nicht explizit darauf geachtet werden muss, dass alle Vertices in einer Ebene liegen. Viele Graphikkarten sind von ihrer Hardwarestruktur darauf spezialisiert, mit Polygonen zu arbeiten. Polygonrepräsentationen werden daher zurzeit häufig bei dreidimensionalen Echtzeit-Anwendungen genutzt. [OpenGL 2004]

Zumeist werden n Oberflächen-Normalenvektoren zu den n Vertices hinzugefügt, die für ein *weiches Shading (smooth shading)*, wie beispielsweise *Gouraud-* oder *Phong-Shading*, benötigt werden. Die Polygone sehen damit vom Helligkeitsverlauf abgerundet aus, auch wenn sie geometrisch gesehen noch immer planar sind.

Für das *Backface Culling*, eine Technik, die das Rendering beschleunigt, indem sie nur diejenigen Polygone in den Renderprozess weitergibt, deren Oberfläche in Richtung der Kamera zeigt, wird ein weiterer Vektor benötigt, um die äußere Oberfläche zu markieren.

Ein Polygon kann auch mit *Texturen* versehen werden. Dazu wird dem Polygon mindestens ein Texturpointer und jedem Vertex jeweils eine zweidimensionale Texturkoordinate als zusätzliche Polygoninformation beigefügt. Das Aufspannen einer Textur auf die Polygonfläche wird dabei *Mapping* genannt. [Watt 2000] Beim einfachen *Texture Mapping* wird eine „normale“ zweidimensionale Graphik über das Polygon gezogen. Beim *Environment Mapping* wird ein Abbild der Umgebung als Textur verwendet, um metallische Spiegelungen zu simulieren. Beim *Reflection Mapping* steuert die Textur die spekularen Highlights. Beim *Mip Mapping* wird eine Textur gleich mehrfach in verschiedenen Auflösungsstufen abgespeichert. Je nach Entfernung wird dabei eine andere Auflösungsstufe dargestellt, um Aliasing-Effekte bei hochfrequenten Mustern zu verhindern. Um einen harschen Übergang beim Wechsel zwischen zwei Auflösungsstufen zu verhindern, können dabei verschiedene Arten von *Mip Map Interpolationen* angewandt werden.

Eine besondere Art der Texturierung ist *Bump Mapping*. Die Werte der Texturpixel repräsentieren hier Höhenwerte, die auf dem Polygon eine Art 3D-Relief erzeugen. Es existieren verschiedene Arten von Bump Mapping.

Das *Emboss Bump Mapping*, auch *Fake Bump Mapping* oder *Perturbed Blend Bump Mapping* genannt, ist die einfachste Art des Bump Mappings. Hier wird der Eindruck von Höhenunterschieden durch einfache Verschiebung der Bump Map in Richtung der

Lichtquelle erreicht. Die verschobene Bump Map wird von der originalen Bump Map pixelweise subtrahiert, um somit eine *Lightmap* zu erhalten, welche die Helligkeit der Oberfläche modifiziert. Da keine Rücksicht auf Beleuchtungsmodelle gelegt wird sind die Resultate dementsprechend bescheiden.

Dot Product 3 Bump Mapping, auch *Perturbed Normal Bump Mapping* genannt, verändert die Normalenvektoren der Oberfläche für die Beleuchtungsberechnung. Dadurch wirkt die Tiefenstruktur sehr realistisch. Lediglich an den Rändern eines Objektes wird deutlich, dass die Tiefenstruktur aufgemappt ist, da die Höhen und Tiefen keine Silhouetten erzeugen. Auch bei sich verdeckenden Strukturen, was häufig bei seitlicher Betrachtung tiefer Oberflächenstrukturen auftritt, versagt das Dot Produkt 3 Bump Mapping, es findet keine optische Abdeckung (*Okklusion*) von dahinterliegenden Oberflächenstrukturen statt. [Blinn 1978] [Saar et al. 2000]

Environment Mapped Bump Mapping (EMBM) ist eine Chimäre aus Dot Product 3 Bump Mapping und Environment Mapping. Hier werden die Oberflächenpixel einer Umgebungstextur mittels Normalenvektoren zur Oberfläche ausgewählt. Die Oberflächennormalen werden dabei durch die Pixelwerte einer Bump Map Textur verbogen, so dass die Illusion feiner reflektierender Oberflächenstrukturen entsteht. [Matrox 2004] [Blitzbase 2004]

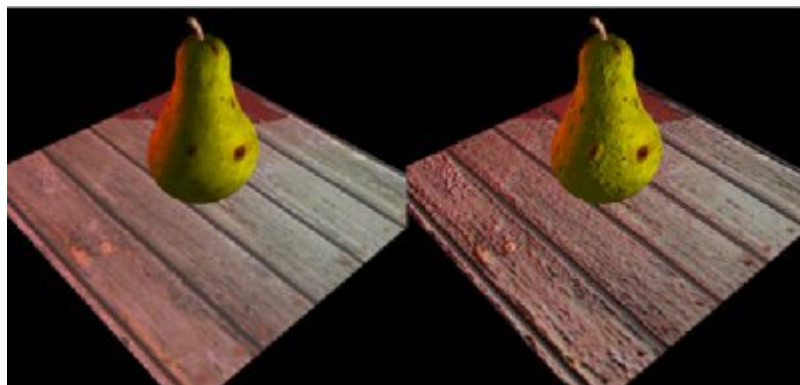


Abbildung 15: Beispiel für Dot Product 3 Bump Mapping

Die Birne und das Holzbrett wurden links mit normalem Textur Mapping versehen, rechts dagegen wurde zusätzlich Dot Product 3 Bump Mapping verwendet. [Papst et al. 2000]

Bilineares Filtering vermeidet die genaue Sichtbarkeit der Pixelgrenzen auf dem Polygon durch Tiefpassfilterung oder Interpolation benachbarter horizontaler und vertikaler Pixel. *Trilineares Filtering* stellt eine Symbiose zwischen bilinearem Filtering und Mip Map Interpolation dar. [Foley et al. 1996] [Watt 2000]

2.1.2 Parametrische Flächenbeschreibungen

Bei *parametrischen Flächenbeschreibungen* ist – im Gegensatz zu Polygonnetzen, bei welchen nur die Eckpunkte definiert sind – jeder Punkt der Fläche genau definiert. Dies impliziert, dass parametrische Flächen eine exakte analytische Repräsentation der Oberfläche zu liefern vermögen.

Kubische parametrische Flächenbeschreibungen werden durch *Kontrollpunkte* bzw. *Stützpunkte* und eine Menge von impliziten *Basis-* oder *Blending-Funktionen* definiert, welche dazu geeignet sind, die Oberfläche stetig zu interpolieren, so dass runde Oberflächenbeschreibungen ermöglicht werden.

Die Blending-Funktionen sind dabei von dem zugrunde liegenden Modell abhängig. Es gibt verschiedene Typen, von denen *Bezier-Kurven* und diverse *Spline-Repräsentationen* (*B-Splines*, *Beta-Splines* u.a.) die gängigsten sind. In manchen Fällen werden auch noch weitere Parameter verwendet, wie etwa Knoten-Vektoren oder Bias- und Tension-Werte. [Watt 2000] [Foley et al. 1996]

In der letzten Zeit setzt sich vermehrt die Technik der *nonuniformen rationalen B-Splines* durch. Nonuniforme rationale B-Splines, kurz *NURBS*, sind parametrische Flächenbeschreibungen, welche mit wenigen Grundparametern beliebige Kegelschnitte und sonstige stückweise konische Oberflächen beschreiben können. [Piegl & Tiller 1997] Sie werden heute häufig gegenüber verwandten Spline-Methoden zur Modellierung biologisch anmutender Objekte vorgezogen. [Parent 2002, S. 354] Viele gängige *3D-Modellierungsprogramme*, wie *Rhino*, *3D Studio Max*, *Maya* und *Softimage*, bieten daher umfangreiche NURBS-Modellierungswerkzeuge an. [Alias Wavefront 2003] [Avid 2004] [Becker et al. 1999]

Aktuelle 3D-Basis-APIs können in der Regel parametrische Flächenbeschreibungen nicht direkt verarbeiten. [Hawkins 2003] [SGI 2004] Sie müssen daher für Echtzeitanwendungen durch eine *Triangulierung* in Polygonnetze umgerechnet werden. Bei dieser Umrechnung lässt sich eine *Tesselierung* der Parameterebene anwenden. Dies hat den Vorteil, dass sich die Granularität des Polygonnetzes dynamisch an die

Erfordernisse anpassen kann. [Piegl & Tiller 1997] Cani und Hornus verwenden in Zusammenhang mit impliziten Flächen dynamische Tessellierungen für eine interaktive Modellierungstechnik, welche runde, organische Formen erzeugen kann. [Cani & Hornus 2001]

2.1.3 Volumendaten

Volumendaten sind eine Menge von skalaren Daten, welche an diskreten Punkten im dreidimensionalen Raum definiert sind. In der Regel sind die einzelnen skalaren Daten zueinander orthogonal-äquidistant angeordnet. Solche rechtwinklig gleichabständigen Skalarwerten werden auch *Volumenelemente* oder *Voxel* genannt. n^3 Voxel spannen dabei als Eckpunkte $(n-1)^3$ Zellen auf.

Diese 3D-Skalarfelder bzw. Zellen können sowohl indirekt als auch direkt visualisiert werden. Zu den *indirekten Visualisierungsmethoden* gehören Oberflächendarstellungen wie beispielsweise zweidimensionale *Schnittebenen* oder dreidimensionale *polygonale Isoflächen*, die meist mit Hilfe des *Marching Tetraeder*- oder des *Marching Cube-Algorithmus* gewonnen werden. [Lorensen & Cline 1987] Die indirekten Methoden können naturgemäß nur einen Teilbereich des gesamten Datensatzes zeigen.

Hingegen kommen die *direkten Visualisierungsmethoden* ohne geometrische Zwischenrepräsentationen aus. Sie stellen beispielsweise die Volumendaten in semitransparenter Weise (*Semitransparenzvisualisierung*) unmittelbar dar. [SGI 2004] [Westermann & Ertl 1998]

Volumendaten können direkt aus den Daten von 3D-Diagnoseverfahren, wie z.B. der *3D-Ultraschalldiagnose*, der *3D-Röntgenbildanalyse* oder der *Kernspintomographie* gewonnen werden. Sie vermögen darüber hinaus auch innere Gewebestrukturen zu zeigen. Sie werden daher oft in der diagnostischen Medizin benutzt. [Volume 2004] Ob direkt oder indirekt visualisiert, stellen jedoch alle Verfahren sehr hohe Anforderungen an die Computerkapazität bezüglich Speicherplatz und Performanz, was sie für Echtzeitvisualisierungen von Organismen nur sehr eingeschränkt benutzbar macht. [Westermann et al. 1999]

2.1.4 Billboards

Ein *Billboard* ist ein einzelnes Polygon bzw. eine Ansammlung von parallelen Polygonen, welche sich stets dem Betrachter zuwenden. In der Regel erfolgt dies, indem die Polygonflächen orthogonal rechtwinklig zur Kamerarichtung stehen. Ein Billboard kann sich aber auch um eine bestimmte Achse drehen und dabei der Betrachtungsrichtung folgen. Haben Billboards nicht nur eine Fläche, sondern bestehen sie aus mehreren hintereinander liegenden Schichten, so wird von *Multibillboards* gesprochen. *Monobillboards* bezeichnen dementsprechend Billboards, die nur aus einer einzigen Schicht bestehen.

Texturierte Billboards erlauben eine Polygonreduktion in großen Szenen, indem bestimmte weit entfernte Unterszenen vorgerendert auf die Billboards appliziert werden und somit eine höhere Polygonkomplexität vortäuschen. Normalerweise werden dazu *Alphatexturen* verwendet, bei denen der Bereich außerhalb der Silhouette als transparent und innerhalb als opak definiert wird. Beispielsweise könnten in der Organismengenerierung weit entfernte komplexe Korallen mit Billboards realisiert werden, indem die Bilder einzelner Korallenriffe als Textur dienen.

Werden Billboards zusammen mit *Videotexturen* verwendet, können auch Organismen mit komplexen Bewegungsabläufen vorgetäuscht werden. Zum Beispiel könnte ein Vogelschwarm mit einer Menge solcher *Videobillboards* erzeugt werden.

2.1.5 Punkte und Linien

Punkte und *Linien* – im Sinne der Computergraphik – sind 0-dimensionale bzw. 1-dimensionale Strukturen der Computergraphik, welche die logische Ergänzung zu Polygonen als 2-dimensionale und Volumendaten als 3-dimensionale Strukturen im 3-dimensionalen Raum darstellen.

Point based surface element methods definieren die Oberfläche eines Objektes annäherungsweise durch eine dichte Punktwolke. Die Punkte der Oberfläche werden *Surfels* (von surface elements) genannt. Um die dimensionslosen Surfels auch auf dem Bildschirm darzustellen, müssen verschiedene entfernungsabhängige Filter angewandt werden. [Alexa et al. 2001] [Ren et al. 2002]

Die Darstellung von Linien in der Computergraphik kann durch *Drahtgittermodelle* erfolgen.

Punkte und Linien können von einigen speziellen Graphikkarten direkt dargestellt werden. Auch mit Hilfe von *Vertex Shadern* können begrenzte linien- und punktartige Gebilde erzeugt werden. Da viele Graphikkarten jedoch auf die Verarbeitung von Polygonen spezialisiert sind und dabei Vertex Shader nicht unbedingt beinhalten, werden in der Praxis Linien- oder Punktwolken oft durch mehrschichtige *Billboards* (siehe Kapitel 2.1.4) generiert. Die Billboards sind für die Linien- oder Punktdarstellung mit einer Alphatextur versehen. Diejenigen Pixel der Alphatextur, die einen Punkt bzw. eine Linie markieren, werden opak markiert, die anderen transparent. [Colditz et al. 2005]

2.1.6 L-Systeme

Lindenmayer-Systeme bzw. *L-Systeme* bezeichnen einen mathematischen Formalismus, der 1968 von dem Botaniker *Aristid Lindenmayer* als Grundlage einer axiomatischen Theorie biologischer Wachstumsprozesse propagiert wurde.

L-Systeme verwenden dabei eine Menge von streng definierten *Produktionsregeln*, um mittels sukzessiver, in der Regel rekursiver Ersetzung von Einzelteilen eines einfachen biologischen Objektes komplexe biologische Strukturen zu generieren. Ganz im Gegensatz zur sequentiellen Vorgehensweise bei den Ersetzungsschritten in formalen Phrasenstrukturgrammatiken, wie beispielsweise der Chomsky-Grammatik, werden die Regeln des Formalismus bei L-Systemen parallel ausgeführt. Der dabei verwendete biomathematische Formalismus orientiert sich hauptsächlich an empirischen Beobachtungen der Flora. [Deussen 2003a][Deussen & Lintermann 1997a, 1997b, 2001 & 2005] [Rozenberg & Salomaa 1980] [Prusinkiewicz & Lindenmayer 1990]

Beschrieb Aristid Lindenmayer anfangs lediglich das Wachstum von fadenförmigen Algen mit Hilfe von L-Systemen, so wurden sie mittlerweile auf allgemeine Vegetationsmodelle erweitert. [Rozenberg & Salomaa 1992]

Zusätzlich zur Formbeschreibung erlauben L-Systeme eine bis zu einem gewissen Grad biologisch korrekte Animation von pflanzlichen Wachstumsprozessen.

Deussen und Lintermann erweitern die klassischen L-Systeme zu einer interaktiven Modellierungsmethode. Der Modellierer kann mit Hilfe eines Graphischen User Interface 11 verschiedene *Components* hierarchisch kombinieren, um somit eine Vielzahl von Pflanzenstrukturen zu erzeugen. [Lintermann & Deussen 1998] [Deussen & Lintermann 1997b, 2001 & 2005]

L-Systeme sind damit hervorragend geeignet, mit wenigen Parametern die verzweigten Formen von Pflanzen – vom informationsquantitativen Standpunkt aus betrachtet – ökonomisch zu beschreiben. Trotz dieser ökonomischen Beschreibung kann die komplexe Form von Pflanzen frappierend realistisch beschrieben werden.

2.1.7 Blobby Objects

Ein Modellierungsansatz, der schon per se eine biologisch anmutende Repräsentation der 3D-Geometrie erzeugt, ist das *Blobby Modelling*. Blobby Modelling verwendet imaginäre punktförmige Partikel, die mit einem radial abnehmenden Dichtefeld umgeben sind, die *Blobby Objects* oder kurz *Blobbies*. Die Feldstärken der Dichtefelder addieren sich dabei in jedem Punkt im euklidischen Raum. Wenn die Summe einen festgelegten Grenzwert überschreitet, so ist der Raumpunkt als „innen“ definiert, ansonsten als „außen“. Durch Isoflächengenerierung, die meist mittels des Marching Cube-Algorithmus erfolgt, werden weiche, organisch anmutende Oberflächen erzeugt, die als *implicit surfaces* bezeichnet werden. [Blinn 1982] [Menon 1996] [Shen & Thalmann 1995] [Wyvill et al. 2003] [Wyvill 2004].

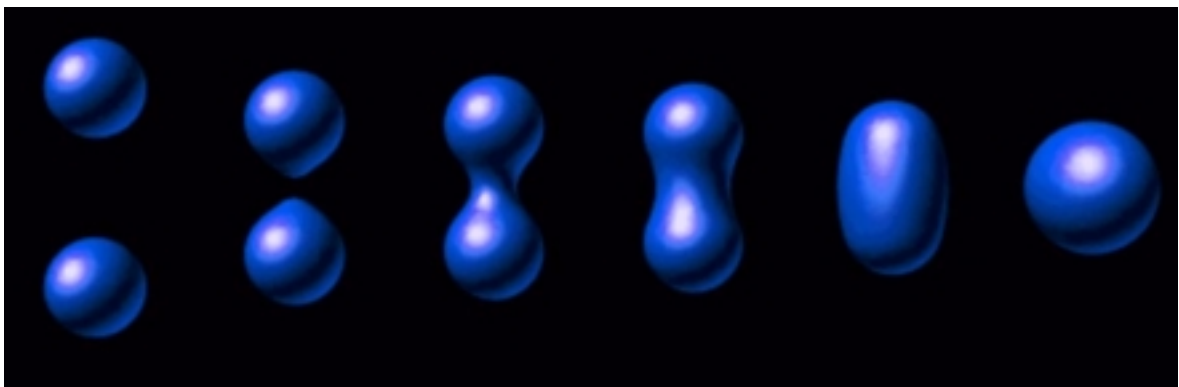


Abbildung 16: Beispiel für die Fusion zweier Blobby Objects

Beispiel für den Verschmelzungsprozess zweier gleich großer Blobby Objects. Wie zu ersehen ist, bilden die generierten Implicit Surfaces automatisch rundliche Formen, die unter anderem auch für die Generierung von Organismen verwendet werden können. [Wyvill 2004]

Es existieren verschiedene Funktionen für die Erzeugung des Dichtefeldes. Formel F1 beschreibt eine einfache Feldfunktion, die in analog modifizierter Form in der Natur häufig vorkommt, z.B. beim Newton'schen Gravitationsgesetz, wo der Parameter a lediglich durch das Produkt der zwei beteiligten Massen durch die Gravitationskonstante ersetzt ist. Sie kann aber auch für die Erzeugung von Bloby Objects verwendet werden. Der Parameter r ist der Radius vom Zentrum des Bloby Object und a ist ein Skalierungsfaktor der Feldstärke:

$$\mathbf{F1} \quad F(r) = \frac{a}{r^2}$$

Funktion F2 ist dagegen die klassische Dichtefunktion, die von Blinn zuerst für die Berechnung von Elektronenwolken verwendet wurde. Die Bloby Objects werden in diesem Fall daher auch als *Bloby Molecules* bezeichnet. Im Gegensatz zu F1 besitzt diese Funktion zwei Parameter: Neben dem Parameter a für die Skalierung der Feldstärke existiert auch der Parameter b , mit dem der Wirkradius verändert werden kann. [Blinn 1982]

$$\mathbf{F2} \quad F(r) = ae^{-br^2}$$

Bei der Funktion F3 wird die Dichtefunktion – im Gegensatz zu F2 – an einem Raumpunkt, der weiter als die Länge b vom Zentrum des Bloby Object entfernt ist, gleich Null. Mit Hilfe von *Octrees* lassen sich daher komplette Raumbereiche von der Berechnung ausklammern, so dass die Berechnung beschleunigt wird. Funktion F3 wird aufgrund dieser algorithmischen Performanz auch am häufigsten in kommerziellen Modellierungsprogrammen für Bloby Objects verwendet, die dort als *Metaballs* bezeichnet werden. [Graves 1993] [Digital 2003]

$$\mathbf{F3} \quad F(r) = \begin{cases} a(1-3r^2)/b^2 & , \quad 0 \leq r < b/3 \\ (3a/2)/(1-r/b)^2 & , \quad b/3 \leq r < b \\ 0 & , \quad b \leq r \end{cases}$$

Zu erwähnen ist auch die Feldfunktion F4 der Gebrüder Wyvill. Hier werden die Blobby Objects auch als *Soft Objects* bezeichnet: [Wyvill & Wyvill 1989]

$$\mathbf{F4} \quad F(r) = a \left(1 - \frac{4r^6}{9b^6} + \frac{17r^4}{9b^4} - \frac{22r^2}{9b^2} \right)$$

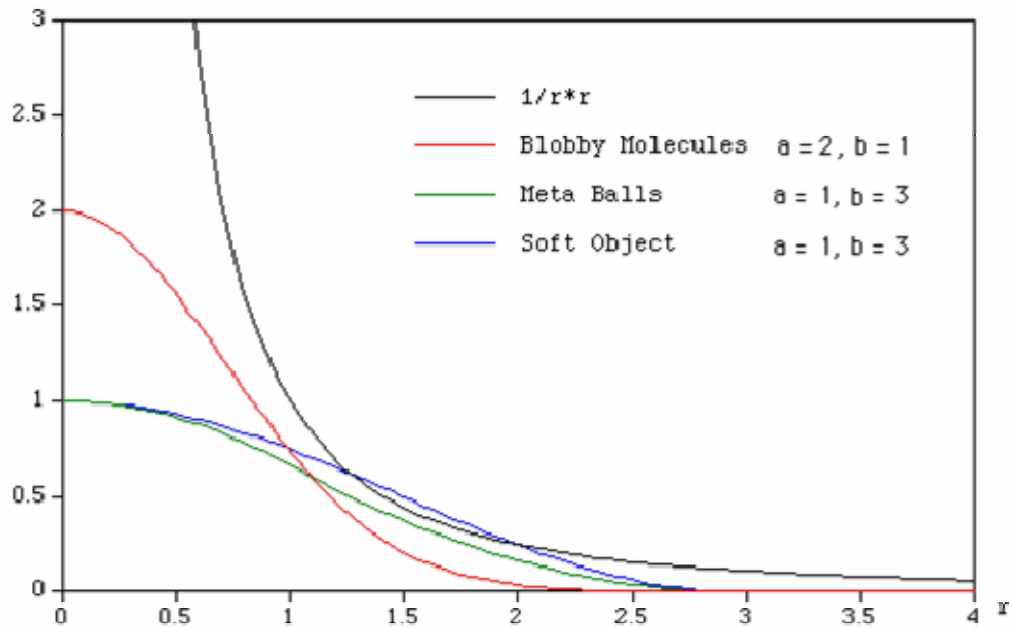


Abbildung 17: Graphen der Feldfunktionen von Blobby Objects

Gegenüberstellung von vier Graphen verschiedener Feldfunktionen von Blobby Objects, die für die Generierung von impliziten Oberflächen verwendet werden. [Bourke 1997]

Blobby Objects werden in Form der Metaballs von einer zunehmenden Zahl an 3D-Modellierungsprogrammen angeboten. [Avid 2004] [Digital 2003] [Kinetix 2003]. [McPheeters 1990]



Abbildung 18: Beispiel für die Erstellung von Organismen mit Blobby Objects

Hier wird ein Beispiel gezeigt, wie mittels Blobby Objects (hier Metaballs) Organismen mit rundlichen Formen modelliert werden können. Links: Skizze der Anordnung (horizontaler Querschnitt durch den Brustbereich) von Blobby Objects, die einen weiblichen Torso bilden. Mitte: Mit Metaballs gebildeter weiblicher Torso - insbesondere im Bereich des Schlüsselbeines fällt auf, dass der Torso nicht biologisch sinnfälliger ist. [Tomoyuki 2004] Rechts: Mittels Metaballs fertig modellierte, texturierte und per Raytracing gerenderte Marilyn Monroe. [Thalmann 1996]

Ein Nachteil von Metaballs ist allerdings, dass sie für interaktive Modellierung und andere Echtzeitanwendungen in der Regel zu rechenintensiv sind, auch wenn durch Weiterentwicklungen, wie z.B. den *Octree Blobbies* oder der *Subdivision Surface Method for Fast Approximate Implicit Polygonization* eine Beschleunigung erzielt wurde. [Bloomenthal 1988] [Menon 1996] [Wyvill et al. 2003]

2.1.8 Sonstige Geometrie-Repräsentationen

Die *Constructive Solid Geometry*, kurz *CSG* ist sowohl eine Geometrie-Repräsentation als auch eine Modellierungstechnik, die vor allem in CAD-Programmen zur Generierung von technisch anmutenden Körpern verwendet wird. Die *Constructive Solid Geometry* verwendet dazu einfache geometrische Primitive, die mit Booleschen Operatoren verbunden werden. [Wiegand 1996]

Sweep-Körper oder kurz *Sweeps* werden erzeugt, indem eine vom Modellierer festgelegte Fläche entlang eines Pfades geführt wird. Das Volumen, das dabei durchfahren wird, definiert den Körper. Mit Sweeps lassen sich nur vergleichsweise einfache Geometrien definieren. Die komplexen deformierbaren Formen von Organismen lassen sich damit nur sehr bedingt erstellen.

Implizite Körperbeschreibungen verwenden einfache Formeln mit drei Unbekannten, um die geometrischen Körper zu beschreiben. Sie erlauben damit zwar kompakte und somit ökonomische Repräsentationen, sind aber auf die Darstellung von vergleichsweise einfachen geometrischen Körpern beschränkt.

Fraktale Geometrie-Repräsentationen benutzen einfache rekursive Generierungsvorschriften, um komplexe selbstähnliche Geometrien zu definieren. Die komplexen skaleninvarianten Formen haben einen speziellen ästhetischen Reiz, so dass sie vor allem in der digitalen Kunst eine große Rolle spielen. In der 3D-Computergraphik werden sie beispielsweise bei der Landschaftsgenerierung und der Planetengenerierung eingesetzt. Die in Kapitel 2.1.6 separat behandelten L-Systeme können dabei als Sonderform der fraktalen Geometrie-Repräsentationen angesehen werden.

Ebenfalls eng mit den fraktalen Geometrie-Repräsentationen verwandt sind *Iterierte Funktionen-Systeme* oder kurz *IFS*. Iterierte Funktionen-Systeme bezeichnen eine Menge von Funktionen, die denselben Raum sowohl als Definitions- als auch als Wertebereich haben und unter Verknüpfung abgeschlossen sind. Mit ihnen lassen sich selbstähnliche Graphiken und Körper erstellen. Bekannte Beispiele für Iterierte Funktionen-Systeme sind z.B. das *Sierpinsky-Dreieck*, die *Sierpinsky-Würfel*, die *Koch-Kurve* oder *Cantormengen-Visualisierungen*. Ein Beispiel, wie Iterierte Funktionen-Systeme bei der Pflanzengenerierung eingesetzt werden, sind *fraktale Farne*. [Mandelbrot 1982][Peitgen & Saupe 1988]

2.1.9 Bewertung der Geometrie-Repräsentationen

Organismen haben in der Regel nicht kantige, sondern eher abgerundete weiche Formen. Kurvige Strukturen können allerdings mit einem engmaschigen Polygonnetz lediglich approximiert, jedoch nie vollständig beschrieben werden. Für optisch zufrieden stellende Beschreibungen einer komplexen anatomischen Oberfläche werden daher mehrere tausend Polygone benötigt. [Parent 2002, 354]

Dieses Manko impliziert, dass Polygonetze in Hinblick auf biologische Strukturen unökonomisch sind. Die umständliche Beschreibung von Körperformen durch Polygonetze ist sehr ineffizient bezüglich Speicherkapazität und wenig flexibel, wenn Deformationen dargestellt werden sollen, wie sie bei Bewegungen, Wachstumsprozessen und evolutionären Formveränderungen auftreten.

Parametrische Flächenbeschreibungen sind dagegen gut geeignet, mit wenigen Parametern die runden, weichen Formen von Organismen zu beschreiben. Die gekrümmte Oberfläche von Organismen kann so – vom informationsquantitativen Standpunkt aus betrachtet – ökonomisch dargestellt werden.

Auch können Formveränderungen, wie sie bei Bewegungsflexionen, Wachstumsprozessen und evolutionären Veränderungen auftreten, elegant durch die Änderung der Stützpunkte erzeugt werden. Die Formen lassen sich dabei editieren, ohne dabei die Glattheit (*smoothness*) der Flächen zu verlieren.

Allerdings müssen parametrische Flächenbeschreibungen für das Rendering meist in Polygone umgewandelt werden, so dass viele dieser Vorteile nur theoretisch bestehen. In der Praxis sind parametrische Flächenbeschreibungen durch den notwendigen Triangulierungsprozess meist langsamer als direkte Polygonnetzbeschreibungen und dadurch nur bedingt für Echtzeitanwendungen geeignet.

Auch sind parametrische Flächenbeschreibungen als rein mathematisches Konstrukt nicht biologisch begründet und sind daher für Biomediziner unintuitiv bezüglich der Modellierung von Organismen.

Volumendaten sind bei diagnostischen Visualisierungen in der Medizin weit verbreitet, da sie einerseits direkt aus den Daten von 3D-Diagnoseverfahren, wie z.B. der *3D-Ultraschalldiagnose*, der *3D-Röntgenbildanalyse* oder der *Kernspintomographie* gewonnen werden können und andererseits innere Gewebestrukturen relativ unverfälscht darstellen können.

Allerdings bedingt die fehlende logische Interdependenz der Volumenelemente und die daraus resultierende mangelnde biomorphe Flexibilität, dass Volumendaten sich vornehmlich zur Visualisierung von 3D-Scans eignen, jedoch nicht zur Neumodellierung von Organismen und nur bedingt zur Simulation derselben. Die Modellierung oder Simulation komplexer dynamisch agierender dreidimensionaler Organismen ist mit Volumendaten unmöglich.

Billboards sind sehr ökonomisch bezüglich der Speicherkapazität und können für Schwärme und weit entfernte Organismen verwendet werden. Sie täuschen in Kombination mit anderen Polygondaten eine hohe Komplexität der Szenerie vor. Allerdings sind der Organismendarstellung mit Hilfe von Billboards enge Grenzen gesetzt.

Punkte und Linien stellen eine Alternative zu Polygondarstellungen dar. Insbesondere wenn hochauflösende Daten vorhanden sind, bei denen ein Polygon höchstens durch ein Bildschirmpixel auf dem gerenderten Bild repräsentiert wird, machen Polygondaten nur noch wenig Sinn. Allerdings ist eine solche Granularität der Daten in der Organismengenerierung nur selten vorhanden. Punkt- und Linienrepräsentationen können jedoch durchaus als zusätzliche Repräsentationsform bei filigranen Strukturen die Polygondarstellung ergänzen.

L-Systeme sind besonders zur eleganten Beschreibung der Form und des Wachstums von filigran-verzweigten biologischen Strukturen geeignet, welche vor allem bei Pflanzen auftreten. Zur Beschreibung der Form, der Bewegungen und des Wachstums von tierischen Organismen – um die es ja in der vorliegenden Arbeit hauptsächlich geht – sind sie dagegen nur bedingt geeignet. Dies liegt an verschiedenen Faktoren:

- Tierische Organismen haben – von verschiedenen Bryozoen-, Seelilien-, Korallen- und Medusenarten abgesehen – in der Regel nicht so filigran-verzweigte Strukturen wie Pflanzen.
- Das Wachstum von tierischen Organismen wird mehr als bei Pflanzen von hydropneumatischen Deformationsprozessen der internen Körperschichten bestimmt.
- Tierische Organismen haben komplexe Bewegungsabläufe, die mit L-Systemen nicht beschrieben werden können.
- Die Form und das Wachstum anatomischer Strukturen von tierischen Organismen können infolge der mechanischen Interdependenz und der Formkomplexität nicht auf einfache regelbasierte Systeme reduziert werden.

Trotz dieser Übertragungsschwierigkeiten der L-Systeme von der Flora auf die Fauna können Teile ihres zugrunde liegenden Paradigmas, eine spezialisierte, biologisch korrekte Regelbeschreibung zur Bildung einer komplexen Objekthierarchie zu verwenden,

modifiziert auch zur Generierung von tierischen Organismen und ihrer anatomischen Strukturen verwendet werden.

Blobby Objects stellen ein intuitives Modellierungswerkzeug dar, da sie mit der Metapher von miteinander verschmelzenden Lehmkugeln arbeiten. [Graves 1993] Die damit erzeugten Implicit Surfaces haben durch ihre rundliche Form per se eine biologische Anmutung. Da jedoch das Metaphermodell nicht biologisch motiviert ist, sind die damit modellierten Organismen biologisch nicht stringent. Dies äußert sich unter anderem dadurch, dass die Organismen etwas „ausgebeult“ aussehen. (siehe Abbildung 18) Nachteilig ist außerdem zu bewerten, dass Blobby Objects – auch unter Berücksichtigung von Octree Blobbies und anderer Beschleunigungsmethoden – nicht völlig echtzeitfähig sind und somit für interaktive Modellierungen oder Simulationen komplexerer Organismen ausscheiden.

Die Constructive Solid Geometry, Sweep-Körper und implizite Körperbeschreibungen erzeugen zu technisch anmutende Geometrien, als dass sie sinnvoll für die Organismenorganisation verwendet werden könnten.

Mit fraktalen Beschreibungen und iterierten Funktionen-Systemen können zwar einige selbstähnliche Aspekte von Organismen ökonomisch erzeugt werden, die Repräsentation ganzer Organismen ist jedoch mit diesen Techniken nicht möglich.

2.2 Formerfassungsmethoden

Bei *Formerfassungsmethoden* wird die Form eines realen Objektes aufgezeichnet und in eine geeignete Geometrie-Repräsentation umgewandelt. Der Modellierungsprozess findet daher – wenn überhaupt – außerhalb des Computers ab. Im Folgenden sollen nur Formerfassungsmethoden behandelt werden, die für die Organismengenerierung grundsätzlich sinnvoll erscheinen. Formerfassungsmethoden, die zu speziell auf die medizinische Diagnostik (*3D-Ultraschall, Computertomographie, etc.*) oder auf die technische Materialprüfung (*CGT-Methoden*) ausgelegt sind, werden in diesem Kapitel nicht behandelt.

2.2.1 Mechanooptische Stiftverfahren

Beim *mechanooptischen Stiftverfahren*, auch *Stylus Tip* oder *Articulation* genannt, bekommt der Benutzer einen Sensorstift als Werkzeug, welcher an einem mehrachsigen Gelenkarm befestigt ist. Mit diesem Stift gleitet er über das zu scannende Objekt.

Aus den Winkelstellungen des Gelenkarmes werden entsprechende Rotationsmatrizen errechnet. Die Länge der Gelenkarme liefert angepasste Translationsmatrizen. Mit Hilfe von konkatenierten Multiplikationen dieser Matrizen können die relativen Positionen der Oberflächenpunkte des Objektes zum Standpunkt des Gerätes zurückberechnet werden. [Immersion 2004c]

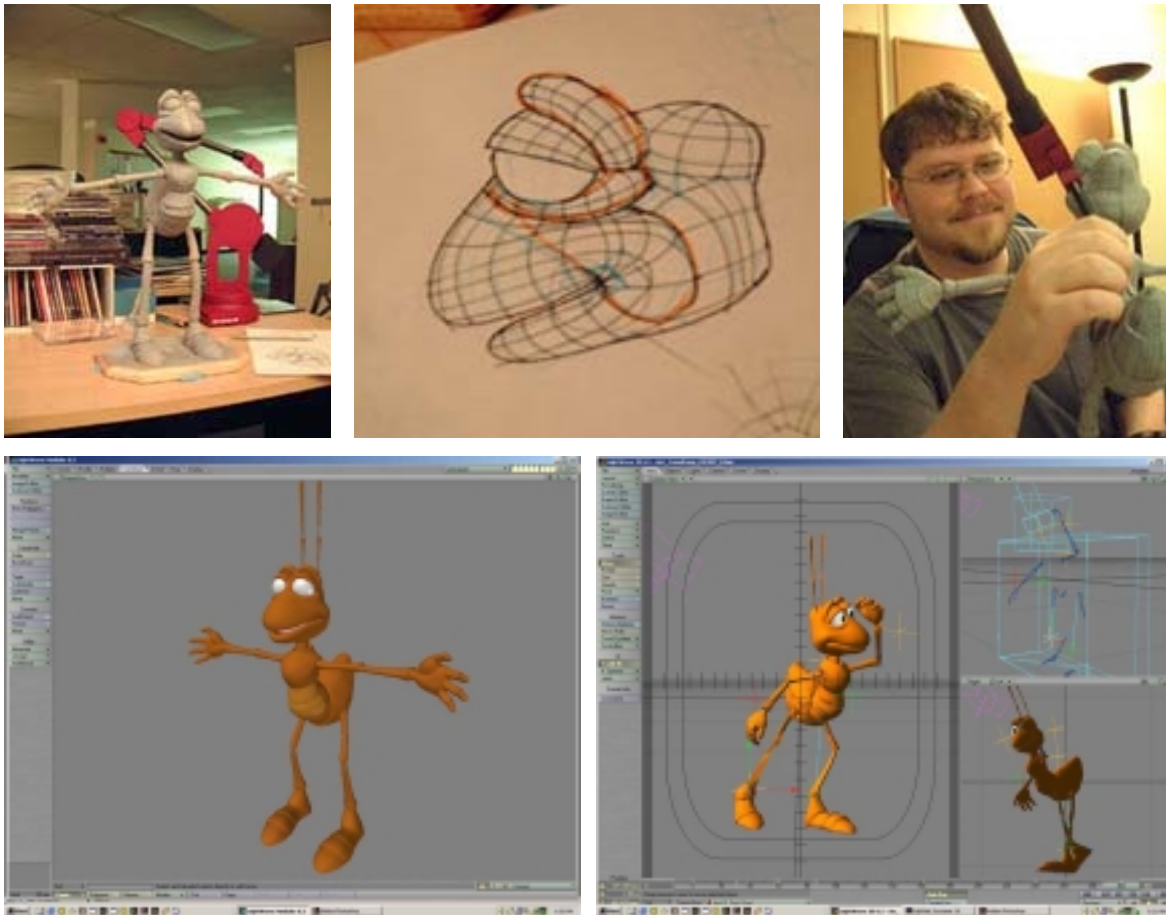


Abbildung 19: Beispiel für die Formerfassung mittels Articulation

Formerfassung von Organismen mittels Articulation am Beispiel der Ameisen des Werbespots „Tax Ants“. Links oben: im Vordergrund ist die Maquette der Ameise, im Hintergrund ist der Immersion MicroStripe 3D-Scanner mit 6 Freiheitsgraden zu sehen. Mitte oben: Die Führungslinien für den Scanning-Stift werden für eine Gesichtshälfte skizziert. Rechts oben: Die Maquette wird eingelesen. Der User fährt dafür die Führungslinien mit dem Griffel des Stylus Tips nach. Links unten: Die Form ist in den Computer eingelesen und kann in einem geeigneten Modellierungsprogramm, hier Lightwave 5.6, nachbearbeitet und texturiert werden. Rechts Unten: Die Ameise wird animiert. [Immersion 2004c]

2.2.2 3D-Scanning

3D-Scanning, auch *Digitizing* genannt, ist ein Verfahren zur berührungslosen Erfassung der Form dreidimensionaler realer Körper. Das Ergebnis des 3D-Scannings ist eine Punktwolke, die mit einer entsprechenden Software entweder in Volumendaten oder direkt in ein Polygonnetz umgewandelt werden kann.

Der Vorteil des 3D-Scannings im Vergleich zur Photogrammetrie besteht in der großen Menge dreidimensionaler Punktdaten, die in kurzer Zeit geliefert werden.

3D-Scanning ist eine recht junge Technik aus den neunziger Jahren. Die Marktlage der Systeme gestaltet sich daher noch sehr dynamisch, mit einer teuren, aber stark abfallenden Preisentwicklung, ständigen Verbesserungen alter Systeme und Entwicklungen neuer Systeme, einem sich ständig erweiternden Anwendungsspektrum sowie einem expansiven Bedeutungszuwachs in der Graphischen Datenverarbeitung. Die für Neuentwicklungen typische expansiv-volatile Marktdynamik ist auch daran ersichtlich, dass der Selektionsprozess bei den Herstellern noch nicht begonnen hat. [Keferstein 2003] (Siehe dafür Tabelle 19 im Anhang auf Seite 374, welche eine Übersicht über 3D-Scanning-Herstellerfirmen mitsamt ihrer Produkte verschafft)

In der akademischen Literatur scheint 3D-Scanning noch nicht umfassend exploriert zu sein, so dass in der vorliegenden Beschreibung vermehrt auf kommerzielle Literaturreferenzen zurückgegriffen werden musste.

Es existieren verschiedene Einteilungsmöglichkeiten von 3D-Scanning. Eine gängige Einteilungsweise in der Industrie ist es, 3D-Scanner nach ihren jeweiligen Einsatzgebieten einzuteilen: [Brückmann 2004] [Böhler & Marbs 2004] [Cyberware 2004] [Daanen & Jeroen 1998] [Steinbichler 2004]

- *Mikro 3D-Scanner*: Zum Scannen mikroskopischer Strukturen; Auflösung 10µm-100µm, Messdauer 10-2000ms; Einsatzgebiete vorwiegend in der biologischen und medizinischen Forschung
- *Derma 3D-Scanner*: Zum Scannen der Hautstruktur; Auflösung 50µm-200µm, Messdauer 100-200ms, Einsatzgebiet im medizinischen Bereich
- *Close Range 3D-Scanner*: Zum Scannen kleiner Objekte von 3cm bis ca. 1 Meter, Auflösung 10µm-1mm, Messdauer 1-5000s, Einsatzgebiet vorwiegend zum Scannen von Skulpturen und Maquettes in der Werbe-, Spiele- und Filmindustrie sowie zum Rapid Prototyping, zur Fräsbahngenerierung, zur computergestützten Qualitätssicherung und zum Reverse Engineering
- *Surface 3D-Scanner*: Zum Scannen von Oberflächenstrukturen; Auflösung 10µm-1mm, Messdauer 1-5000s, Einsatzgebiete in der Fräsbahngenerierung, in der computergestützten Qualitätssicherung und beim Reverse Engineering

- *Face 3D-Scanner*: Zum Scannen von Gesichtern; Auflösung 100µm-2mm, Messdauer 100ms-17s; Einsatzgebiete in der Werbe-, Spiele- und Filmindustrie und der plastischen cranofacialen Chirurgie
- *Head/Limb 3D-Scanner*: Zum Scannen von ganzen Köpfen bzw. ganzen Körperteilen; Auflösung 100µm-2mm, Messdauer ca. 17s; Einsatzgebiete in der plastischen Chirurgie, in der Werbe-, Film- und Spieleindustrie sowie in der Pathologie und der forensischen Kriminalistik
- *Anatomic 3D-Scanner*: Zum Scannen anatomischer Innenstrukturen, Auflösung 0.001mm-2cm, Messdauer 0.001s bis 1h, hierzu gehören verschiedene medizinische 3D-Diagnoseverfahren wie z.B. die 3D-Ultraschalldiagnose, die 3D-Röntgenbildanalyse und die Kernspintomographie. Anatomic 3D-Scanner werden aus historischen Gründen oft nicht dem 3D-Scanning zugeordnet.
- *Whole Body 3D-Scanner*: Zum Scannen ganzer menschlicher Körper; Auflösung 0,1-5mm, Messdauer ca. 17s; Einsatzgebiete im militärischen Bereich sowie in der Bekleidungs-, Werbe-, Spiele- und Filmindustrie und in der Forschung für Restraint Systems
- *Near Range 3D-Scanner*: Zum Scannen großer Objekte über einem Meter; Auflösung 0,1mm-5cm, Messdauer 2-20s; Einsatzgebiete vorwiegend in Kunst und Denkmalpflege, aber auch in der Spiele- und Filmindustrie sowie zum Rapid Prototyping, zur computergestützten Qualitätssicherung und zum Reverse Engineering
- *Mid Range 3D-Scanner*: Zum Scannen von Mauerwerk und Gebäudeteilen, Auflösung ca. 5mm; Messdauer 30s-3h; Einsatzgebiete in der Denkmalpflege, der Architektur, der Forstinventur und im Hoch- und Tiefbau

- *Landscape 3D-Scanner*: Zum großflächigen Scannen von Landschaften, Auflösung 0.1m-1km; Messdauer 1 Stunde bis 20 Tage; Einsatzgebiete im militärischen Bereich, bei der Robotik, sowie in der Architektur und der Städte- und Landschaftsplanung

Die obige Einteilung ist allerdings nur bedingt sinnvoll, da die Klassifizierung von Autor zu Autor bzw. von Anbieter zu Anbieter stark variiert. Darüber hinaus existieren Geräte, die eine Chimäre von zwei Klassifizierungstypen darstellen, z.B. das Modell CT900 der Firma Callidus GmbH. Es könnte von seinen Daten her sowohl als Close Range- wie auch als Surface-3D-Scanner bezeichnet werden, da es einerseits detailliert die Oberflächenstruktur, aber auch durch Drehung des Objektes die grundsätzliche Form aufnimmt. (siehe Abbildung 20 Mitte) [Callidus 2004c]



Abbildung 20: Beispiele von Close Range-, Surface- und Mid Range-3D-Scannern

Links: Beispiel eines Close Range 3D-Scanners, hier das Modell CT180 zum Scannen kleinerer Objekte. Das Gerät besitzt eine Auflösung von 25-75 μm , die Messgenauigkeit beträgt $<10\mu\text{m}$ und die Scangeschwindigkeit 4000 Messpunkte pro Sekunde.

Mitte: Beispiel eines Surface 3D-Scanners. Hier das Modell CT900 zum Scannen der Oberfläche von größeren Objekten bis über einem Meter Höhe. Da es durch Drehung der Scheibe auch die Oberflächenstruktur der Objektrückseite erfasst, könnte es wahlweise auch in der Kategorie der Close Range 3D-Scanner fallen. Das Gerät besitzt eine Auflösung von 10-70 μm , und die Scangeschwindigkeit beträgt 4000 Messpunkte pro Sekunde.

Rechts: Beispiel eines Mid Range 3D-Scanners. Hier das Modell CP 3200. Das Gerät hat eine maximale Reichweite von 80 Metern und eine Messgeschwindigkeit von 1750 Punkten/Sekunde; Die Auflösung kann zwischen 1mm, 1cm und 1dm frei gewählt werden. [Callidus 2004c]

Die Einsatzgebiete, die dieser Klassifizierung zugrunde liegen, sind zudem nicht immer klar und scharf umrissen, z.B. können Whole Body 3D-Scanner auch für das Scannen von Maquetten und anderen Objekten benutzt werden. Darüber hinaus sind einige Whole Body 3D-Scanner nicht in der Lage, größere Menschen bzw. Tiere in einem Scandurchgang als Ganzes einzulesen, manche Geräte sind z.B. auf ein Scanning von Menschen bis zu einer Maximalgröße von 2 Metern beschränkt. (siehe Abbildung 21) [Cyberware 2004]



Abbildung 21: Beispiel eines Whole Body 3D-Scanners

Beispiel eines Color Whole Body 3D-Scanners: Die Auflösung des 3D-Scanners beträgt 0,5mm (Tiefe), ca. 2mm (vertikal) und 5mm (horizontal). Die Dauer des Scanvorganges beträgt 17 Sekunden, in denen sich das zu scannende Objekt, in diesem Fall die Person, nicht bewegen darf. [Cyberware 2004]

Die vorliegende Arbeit befasst sich nur mit 3D-Scannern, die für die Erfassung der äußeren Form von Organismen und Organismenmodellen, vornehmlich Maquetten und Skulpturen, taugen. Dies sind Whole Body-, Close Range, Face-, Head/Limb- und Mid Range-3D-Scanner; alle anderen Scannertypen wie Mikro-, Derma-, Anatomic-, Surface- und Landscape-3D-Scanner werden nicht behandelt.

Weitaus logischer und auch gebräuchlicher als die obige Einteilung der 3D-Scanner nach ihren Einsatzgebieten ist es, die 3D-Scanning-Methoden nach ihren jeweiligen Messmethoden einzuteilen. Dabei existieren zurzeit folgende gängige Messmethoden, die in den Unterkapiteln 2.2.2.1 bis 2.2.2.8 beschrieben werden: [ABW 2003b] [Brückmann 2004] [Wulf & Wagner 2003] [Schroth 2004]

- Passives Triangulationsverfahren (Stereoskopische Photogrammetrie)
- Punktbasiertes Triangulationsverfahren (Punkt-3D-Scanning)
- Streifenbasiertes Triangulationsverfahren (Streifen-3D-Scanning)
- Craycode-Triangulationsverfahren (Balken-3D-Scanning)
- Phasenschiftverfahren (Phasendifferenzverfahren)
- Konoskopische Holographie
- Pulslaufzeitverfahren
- Kombinationen der obigen Verfahren (wie z.B. das MPT-Verfahren)

Dabei gehören das punktbasierte, das streifenbasierte und das Craycode Triangulationsverfahren sowie das Pulslaufzeit- und das Phasendifferenzverfahren zu den *aktiven Verfahren*, da sie jeweils einen *Sender-Empfänger-Aufbau* besitzen. Alle anderen Verfahren kommen dagegen ohne Sender aus und werden daher zu den *passiven Verfahren* gezählt. [Strobl et al. 2004]

Es existieren zahlreiche weitere Verfahren, die weder in der Vergangenheit nennenswerte Bedeutung erlangt haben noch in der Zukunft aufgrund ihrer verfahrensinhärenten Defizite nennenswerten Bedeutungszuwachs erfahren werden, z.B. das *Silhouetten-Verfahren*, bei dem die Formerfassung mittels der Schattenprojektion eines sich drehenden Objektes erfolgt, das einfache und kostengünstige *Schatten-* oder *Shadow-Scanning*, in dem ein Schattenbalken über das still stehende Objekt wandert und anhand der seitlichen Verzerrung der Schattenränder die Form trigonometrisch rekonstruiert wird, *topometrische Gitterstruktur-Verfahren*, bei denen die Kreuzungspunkte eines projizierten Lichtgitters für die Triangulation verwendet werden, oder die Methode der *dynamischen Fokussierung*, welche die Entfernungen der Oberflächenpunkte durch mechanische Scharfeinstellung einer Kamera ermittelt. [Bouget und Perona 2000] [Optimet 2004]

2.2.2.1 Stereoskopische Triangulationsverfahren

Das *stereoskopische Triangulationsverfahren*, auch als *passives Triangulationsverfahren* bezeichnet, verzichtet auf eine separate aktive Lichtquelle und ermittelt die Tiefenstruktur anhand zweier, in einem gewissen Abstand aufgenommener Referenzbilder des gleichen zu scannenden Objektes. [Siebert & Ju 2000] Die Schwierigkeit des stereoskopischen

Triangulationsverfahrens besteht in der Bestimmung homologer Bildpunkte respektive der Identifikation gleicher Objektpunkte in den zwei projektiven Bildern. Dieser Identifikationsprozess wird als *Stereopsie* bzw. *Stereo Matching* bezeichnet und gliedert sich in zwei Grundverfahren, erstens dem *intensitätsbasierten Stereo Matching* bzw. *intensity based stereo matching* und zweitens dem *merkmalsbasierten Stereo Matching* bzw. *feature based stereo matching*.

Dabei werden meist Stereobilder von Kameras verwendet, die zueinander parallel ausgerichtet sind, da dann die epipolaren Linien mit den horizontalen Scanlinien übereinstimmen und sich somit das Stereo Matching von zwei Dimensionen (dem gesamten Bildausschnitt) auf eine Dimension (der jeweiligen horizontalen Linie) reduziert. [Petrov 2002] [Koschan 2004]

Beim intensitätsbasierten Stereo Matching erfolgt die Punktidentifikation anhand ähnlicher Helligkeiten und eventuell auch ähnlicher RGB-Werte. Da sich die diesbezüglichen Werte aufgrund des unterschiedlichen Blickwinkels leicht voneinander unterscheiden können, müssen probabilistische Methoden verwendet werden. Barnard verwendet dazu folgende zu minimierende Energiefunktion F_5 , die hier zur leichteren Verständlichkeit etwas gegenüber dem Originalartikel verändert wurde:

$$F_5 \quad E(x, y) = |I_L(x, y) - I_R(x + D(x, y), y)| + \lambda |\nabla D(x, y)|$$

Dabei bedeuten $I_L(x, y)$ die Intensität (Helligkeitswert oder Farbanteil) in der x . Spalte und der y . Zeile des linken Bildes, $I_R(x, y)$ dementsprechend die Intensität des rechten Bildes an der gleichen Pixelposition. $D(x, y)$ ist der *Disparitätswert* oder in anderen Worten die horizontale Verschiebung und λ ist der *Disparitätslängenparameter*.

Die Identifizierung gleicher Objektpunkte geschieht durch Suchen des minimalen Wertes für $E(x, y)$ für alle Werte von $D(x, y)$, wobei der Term „ $\lambda |\nabla D(x, y)|$ “ als *Continuity Constraint* hilft, kleine Wechsel des Disparitätswertes vor großen zu bevorzugen. [Barnard 1989]

Fensterbasierte Methoden bzw. *window based methods* beschleunigen und verbessern dieses Verfahren, indem sie nur diejenigen Regionen verwenden, die für die Punktidentifikation von Interesse sind, z.B. nur solche, die eine große Variation von Pixel

mit unterschiedlichen Farb- und Helligkeitswerten in horizontaler Richtung aufweisen. [Hannah 1985]

Beim *merkmalsbasierten Stereo Matching* wird das Bildpaar dagegen zuerst vorverarbeitet, um gleiche Bildmerkmale, wie z.B. Ecken, Kanten, Liniensegmente, etc. des Bildes zu extrahieren, die Punktidentifikation wird dann anhand dieser Attribute vorgenommen. [Hannah 1985] [Grimson 1981] [Ohta & Kanade 1985] [Marr & Poggio 1979]

Fehlinterpretationen können dabei jedoch nicht ausgeschlossen werden, die beiden häufigsten Fehlerquellen sind dabei die *Okklusion* und die *Repetitivität*. Okklusion bedeutet in diesem Kontext, dass ein Merkmal, welches in einem Bild vorhanden ist, in dem anderen aufgrund des unterschiedlichen Blickwinkels fehlen kann und somit eine Zuordnung der Pixelwerte verhindert wird. Bei der Repetitivität wiederholen sich in horizontaler Richtung die Bildmerkmale bzw. Intensitätsmuster. Dadurch verliert die Pixelzuordnung ihre Eindeutigkeit.

2.2.2.2 Punktbasierte Triangulationsverfahren

Punktbasierte Triangulationsverfahren, oft auch ungenau als *Triangulationsscanning-* oder *Punktscanning-Verfahren* bezeichnet, beheben die Schwierigkeit des passiven Triangulationsverfahrens, die Oberflächenpunkte eindeutig zu identifizieren, indem sie auf das Objekt einen hellen punktförmigen Lichtstrahl werfen. Eine oder mehrere Kameras (in der Regel CCDs) nehmen das Objekt auf. Dabei haben die Kameras einen gewissen Abstand zum Projektor. Aufgrund seiner Helligkeit wird nun ein Pixel der Kamera extrem stark beleuchtet und der beleuchtete Oberflächenpunkt kann somit eindeutig identifiziert werden. Durch einfache Trigonometrie kann die Raumposition des Punktes ermittelt werden. Der Lichtstrahl gleitet nun zeilenweise (ähnlich dem Elektronenstrahl beim CRT-Fernsehen) über das Objekt, wobei in regelmäßigen Abständen die Raumkoordinaten des Punktes aufgezeichnet werden. Somit kann ein Punktnetz der Objektoberfläche erfasst werden. Meist wird dieser Lichtstrahl durch einen Laser generiert, da nur er ausreichende Helligkeiten erreicht. Es wird dann auch vom *Laserpunktscanning* oder *laser point scanning* gesprochen. [DLR 2004b]

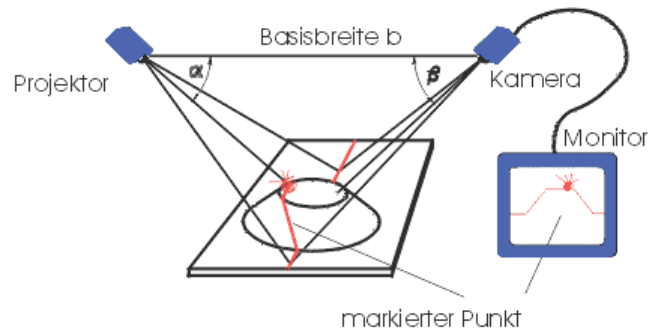


Abbildung 22: Skizze des punktbasierten Triangulationsverfahrens

Ein Projektor wirft einen Lichtpunkt auf ein Objekt. Eine Kamera nimmt den Streifen um die Strecke b versetzt auf. Durch die Basisbreite b sowie den Projektionswinkel α und den Aufnahmewinkel β lässt sich nun durch einfache Triangulation der Punkt auf dem Streifen eindeutig ermitteln. [ABW 2003b]

Wird das Objekt auf einer Scheibe gedreht, damit auch die Rückseite erfasst wird, wird dies *turn table point scanning* genannt.

2.2.2.3 Streifenbasierte Triangulationsverfahren

Das *Streifenbasierte Triangulationsverfahren*, auch *Lichtschnittverfahren*, *Streifenverfahren*, *3D-Streifenscanning*, im englischen *stripe scanning* genannt, ist die gängigste 3D-Scanning-Methode. Wenn der Lichtstreifen durch einen Laser generiert wurde, wird auch von *Laserstreifenverfahren* oder *laser stripe scanning* gesprochen, ansonsten von *Weißlichtstreifenprojektion* bzw. *white light projection*.

Streifenbasierte Verfahren sind eine Weiterentwicklung von punktbasierten Verfahren. Der Hauptunterschied besteht darin, dass anstatt eines einzelnen Lichtpunktes, der über das Objekt gleitet, gleich eine ganze Linie verwendet wird, wodurch sich die Scangeschwindigkeit erhöht.

Bei diesem Verfahren wird Licht in eine feine Linie aufgeteilt und auf das Objekt projiziert. Eine bzw. mehrere Kameras nehmen die reflektierte Linie aus verschiedenen Positionen auf. Aus der um den Triangulationswinkel α gekippten Blickrichtung der Kameras erscheinen die Streifen durch die Oberflächentopographie des Objektes verbogen.

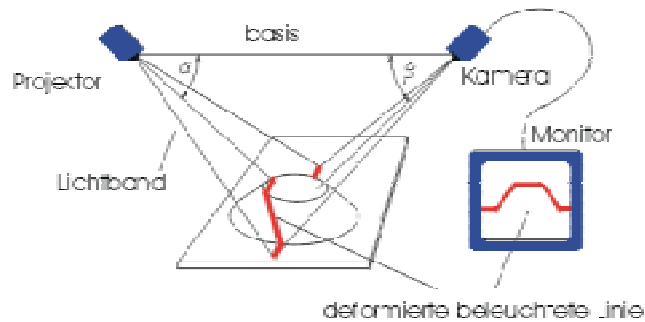


Abbildung 23: Skizze des streifenbasierten Triangulationsverfahrens

Ein Projektor wirft einen Lichtstreifen auf ein Objekt. Eine Kamera nimmt den Streifen um die Strecke b versetzt auf. Die Linie, die vom Projektor aus gerade projiziert wird, erscheint von der Kamera aus deformiert. Durch die Basisbreite b sowie den Projektionswinkel α und den Aufnahmewinkel β lässt sich nun durch einfache sequenzielle Triangulation jeder einzelne Punkt auf dem gesamten Streifen eindeutig ermitteln. [ABW 2003b]

Durch einfache trigonometrische Verfahren werden nun die kartesischen Koordinaten der Oberflächenpunkte ermittelt. Danach wird der Streifen jeweils orthogonal verschoben und die Aufnahmen sowie die Triangulationsberechnungen wiederholt, dies geschieht ca. 25-30 Mal in der Sekunde. So wird Schicht für Schicht die Oberflächenform des Objektes erfasst und in eine Punktwolke verwandelt. [Rapido3D 2004d] [Strobl et al. 2004]

Analog zum *turn table point scanning*, wird vom *turn table stripe scanning* gesprochen, wenn das Objekt auf einer Scheibe gedreht wird, um die Rückseite zu erfassen.

2.2.2.4 Graycode-Triangulationsverfahren

Graycode-Triangulationsverfahren, welche auch als *balkenbasierte Triangulationsverfahren*, *Verfahren mit codiertem Lichtansatz* oder ungenau *gitterbasierte Triangulationsverfahren* bezeichnet werden, stellen eine Weiterentwicklung der streifenbasierten Triangulationsverfahren dar. Anstatt einen Streifen sequenziell versetzt auf das Objekt zu werfen, werden nacheinander von einem Projektor ganze breitflächige Lichtbalken auf das Objekt projiziert. Die Lichtbalken werden in der Regel durch einen LCD-Filter erzeugt, der hinter einer hellen punktförmigen Lichtquelle positioniert wird und durch Aktivierung bzw. Nichtaktivierung verschiedener Pixelbereiche balkenförmige Schatten auf das Objekt wirft. Die Lichtbalken entsprechen dabei einem zeitlich sukzessiv

ausgeführten Graycode, so dass jeder Streifen eine eindeutige Beleuchtungssequenz erfährt. Der Hauptvorteil des Graycode-Triangulationsverfahrens gegenüber dem streifenbasierten Triangulationsverfahren ist die Verkürzung der Scandauer. [ABW 2003b] [Hall-Holt & Rusinkiewicz 2001] [Horn & Hiriyati 1999]

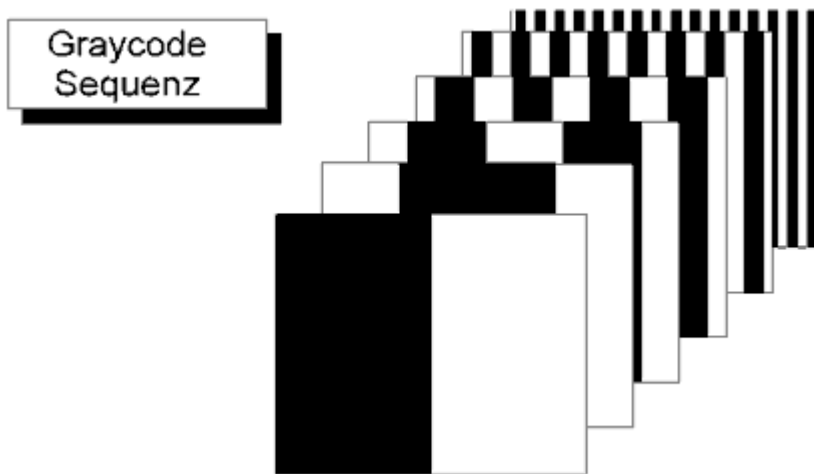


Abbildung 24: Skizze zum Graycode-Triangulationsverfahren

Beim Graycode-Triangulationsverfahren werden nacheinander sequentiell ganze Lichtbereiche, die dem Graycode entsprechen, auf das Objekt geworfen. Somit kann nachträglich jeder Schnittstreifen reidentifiziert werden. Das Graycode-Verfahren ist schneller als das Streifenverfahren. [ABW 2003b]

2.2.2.5 Phasenshiftverfahren

Beim *Phasenshiftverfahren*, auch *Phasendifferenzverfahren*, *Phasen-Messungs-Profilometrie (PMP)* (oder manchmal fälschlicherweise auch *Moiré Interferometrie*) genannt, wird Licht mit einem cosinusförmigen intensitätsmodulierten Linienraster auf das zu scannende Objekt projiziert. Aus einem bestimmten Winkel des Projektors sieht dieses Streifenmuster verzerrt aus. Dabei ist die Verzerrung abhängig von dem Höhenprofil des Objektes. Mit Hilfe gewöhnlicher Triangulationstechniken können Rückschlüsse auf die Oberflächenhöhe gezogen werden. Dazu muss aber erst die Phasenverzerrung auf dem Objekt eindeutig ermittelt werden. Dazu wird – entweder durch eine motorische Verschiebung oder eine elektronisch steuerbare LCD-Einheit – das Linienraster in definierten Schritten jeweils um den Phasenwinkel α verschoben.

Die lokale Lichtintensität $I(x,y)$, die an einem Oberflächenpunkt auf dem zu scannenden Objekt gemessen wird, kann dabei mit Hilfe der Formel F6 aus der

Hintergrundintensität $H(x,y)$, der Streifenmodulation $M(x,y)$ und der gesuchten Phasenlage $\varphi(x,y) + \alpha$ berechnet werden.

$$\text{F6} \quad I(x, y) = H(x, y) \cdot \left[1 + \frac{4}{\pi} M(x, y) \cdot \cos(\varphi(x, y) + \alpha) \right]$$

Es werden drei Messungen I_0 , I_{120} und I_{240} vorgenommen, die zueinander jeweils um 120° phasenverschoben sind, also mit α gleich 0° , 120° und 240° . Durch Einsetzen der jeweils gemessenen Helligkeitswerte $I(x,y)$ zusammen mit dem entsprechenden Phasenverschiebungswinkel α in die Formel F6 kann das resultierende Gleichungssystem nach den drei Unbekannten $H(x,y)$, $M(x,y)$ und $\varphi(x,y)$ aufgelöst werden.

$$\text{F7} \quad H(x, y) = \frac{1}{3} (I_0(x, y) + I_{120}(x, y) + I_{240}(x, y))$$

$$\text{F8} \quad M(x, y) = 2 \frac{\sqrt{[I_0(x, y)]^2 + [I_{120}(x, y)]^2 + [I_{240}(x, y)]^2 - I_0(x, y) \cdot [I_{120}(x, y) + I_{240}(x, y)] - I_{120}(x, y) I_{240}(x, y)}}{2I_0(x, y) - I_{120}(x, y) - I_{240}(x, y)}$$

$$\text{F9} \quad \varphi(x, y) = \arctan \left(\sqrt{3} \cdot \frac{I_{240}(x, y) - I_{120}(x, y)}{2I_0(x, y) - I_{120}(x, y) - I_{240}(x, y)} \right)$$

Wie aus Funktion F9 ersichtlich ist, ist $\varphi(x,y)$ aufgrund des Arcus Tangens eine unstetige Funktion, die unter Berücksichtigung der Vorzeichen in Zähler und Nenner alle 2π eine repetitive Unstetigkeit aufweist. Daher können große Höhengsprünge in der Oberfläche, die eine Phase überspringen, nicht erfasst werden. Dies ist der Hauptnachteil des Phasenschnittverfahrens. Für kleine Höhenänderungen jedoch, bei denen sich eine Hell-Dunkel-Kante im Kamerabild nur um Bruchteile einer Gitterperiode verschiebt, ist das Phasenschnittverfahren sehr präzise, da – wie ebenfalls aus Funktion F9 ersichtlich ist – die Phasenverschiebung unabhängig von der Hintergrundintensität $H(x,y)$ und der Streifenmodulation $M(x,y)$ gemessen werden kann und somit unempfindlich gegen additive und multiplikative Helligkeitsstörungen des Rasters ist. Ein weiterer Vorteil ist die Schnelligkeit der Messung, da nur drei Projektionsschritte für die Erfassung der kompletten Oberflächenform notwendig sind. [ABW 2003b] [Harding & Qian 2004]

2.2.2.6 Konoskopische Holographie

Konoskopische Holographie, im englischen *conoscopic holography*, ist eng mit dem Phasenschriftverfahren verwandt. Der Name ist irreführend, da das Verfahren nur wenig mit der klassischen Holographie zu tun hat.

Bei der konoskopischen Holographie wird der Strahl einer punktförmigen Lichtquelle optisch in zwei Unterstrahlen geteilt, zueinander orthogonal polarisiert und zeilenweise auf ein Objekt geworfen. Der Doppelstrahl wird normalerweise bei diffusen Objekten konzentrisch in verschiedene Richtungen reflektiert. Diese reflektierten Strahlen werden gebündelt und gemeinsam durch ein spezielles einachsiges Prismenkristall geschickt, in dem sie den gleichen Weg nehmen. Die Geschwindigkeit des einen Strahles ist dabei isotropisch und wird *ordentlicher Anteil* genannt, die Geschwindigkeit des anderen ist anisotropisch und wird als *außerordentlicher Anteil* bezeichnet. Die zwei Lichtstrahlen treten mit einer Phasenverschiebung aus dem Kristall aus. Nun werden die Polarisierungen der beiden Strahlen mit einem *Analyser* wieder zueinander angeglichen. Diese Angleichung der Polarisierungsebenen ist für die nachfolgende Interferenz notwendig. Abhängig von dem Austrittswinkel aus dem Kristall wird jeder Doppelstrahl eine verschiedene Phasenverschiebung zwischen ordentlichen und außerordentlichen Anteilen aufweisen. Die zwei zueinander phasenverschobenen Strahlen treffen nun auf eine Projektionsfläche auf, wo sie ein radiales Interferenz-Muster bilden. Die Helligkeitsverteilung des Interferenzmusters ist dabei unter anderem von der Entfernung des Punktes zum Kristall abhängig. Somit kann zeilenweise die Oberflächenstruktur ermittelt werden. [Optimet 2004] [Schiller 2003]

Die Vorteile dieses Verfahrens liegen in der Messgenauigkeit, die bis in den Nanometerbereich reichen kann, der weiten Skalierbarkeit des Messbereiches von $\sim 1\mu\text{m}$ bis $\sim 1\text{m}$ durch einfaches Austauschen einer vorgeschalteten Linse und der Robustheit gegenüber optischen Störungen. [Schroth 2004]

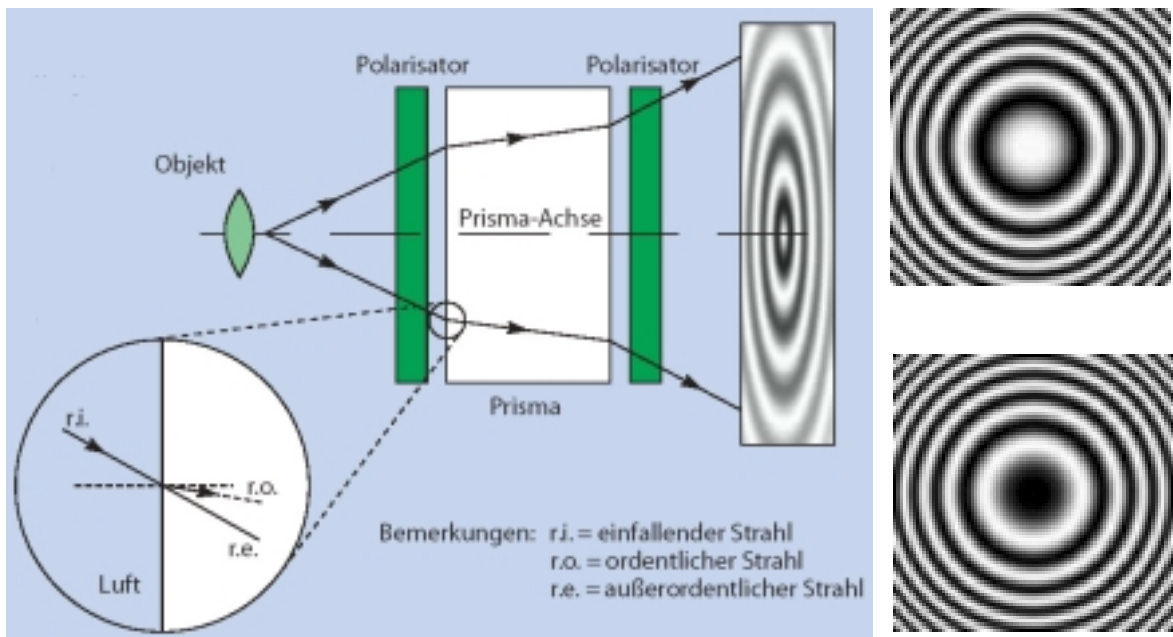


Abbildung 25: Skizze zur Konoskopischen Holographie

Links: Das vom zu scannenden Objekt (hellgrün) reflektierte Licht wird von einem ersten Polarisator (dunkelgrün) in einen ordentlichen-isotropischen und einen außerordentlichen-anisotropischen Strahl aufgeteilt und danach durch ein Prisma (weiß) geschickt. Danach werden die Polarisationsebenen der zwei Strahlen wieder von einem speziellen Polarisator, dem Analyser (dunkelgrün), zueinander angeglichen. Die beiden nun zueinander phasenverschobenen Strahlen bilden auf einer Projektionsscheibe ein konzentrisches Interferenzmuster. [Isel 2004, 3] Rechts oben und unten: Dieses Interferenzmuster sieht je nach Abstand des Objektpunktes zum Kristall unterschiedlich aus. [Optimet 2004]

2.2.2.7 Pulslaufzeitverfahren

Beim *Pulslaufzeitverfahren*, auch *Time of Flight Method* genannt, wird ein Puls erzeugt, der wahlweise aus folgenden Schallwellen bzw. elektromagnetischen Schwingungen oder physikalischen Medien bestehen kann:

- Ultraschallwellen
- Monochromatische Infrarot-Strahlung
- ultraviolette Strahlung
- inkohärentes Weißlicht
- monochromatisches Laserlicht im sichtbaren Frequenzspektrum
- Radar-Strahlung
- Röntgenstrahlung

Bei den 3D-Scannern, die für die Formerfassung von Organismen interessant sind (Whole Body, Face, Head/Limb, Close, Near), besteht der Puls in der Regel aus monochromatischem Laserlicht im sichtbaren Frequenzspektrum. Der Laserpuls wird dabei in gleichmäßigen Abständen in die Umgebung geschossen, in der das zu scannende Objekt steht, und wandert dabei zeilenweise vor und zurück. Er wird vom zu scannenden Objekt bzw. von anderen Objekten der Umgebung reflektiert und dann von einem Sensor erfasst.

Da die Geschwindigkeit von elektromagnetischen Wellen mit ca. 300.000km/h praktisch konstant ist, kann die Zeitspanne zwischen dem Aussenden und der Erfassung des Laserpulses leicht zur Bestimmung des Abstandes zwischen Scanner und Reflexionspunkt verwendet werden. [Archaeoptix 2003] [Wulf & Wagner 2003]

Die Messgenauigkeit des Pulslaufzeitverfahrens ist stark limitiert, sie beträgt typischerweise um 3 bis 6mm. Die Reichweite ist dagegen mit bis zu ca. 80 Metern recht hoch. [Archaeoptix 2003] [Iqsun 2004] [Zöllner 2004]

2.2.2.8 Kombinationen

Neben den verschiedenen 3D-Scannern, die ein einzelnes Verfahren verwenden, existieren auch Geräte, die eine Kombination aus zwei verschiedenen Scanning-Methoden verwenden.

Die häufigste Variante ist dabei die Kombination aus Graycode-Technik und Phasenschiftverfahren, da sich beide synergistisch-komplementär ergänzen. Das Phasenschiftverfahren besticht durch seine Genauigkeit, die bis in den Nanometer-Bereich reichen kann, hat aber Schwierigkeiten mit abrupten Höhenunterschieden, welche die halbe Wellenlänge des verwendeten Lichtes überschreiten. Das Graycode-Verfahren kann dieses Defizit ausgleichen, indem es die ungefähre Höhe schnell ermittelt, so dass der mit dem Phasenschiftverfahren ermittelte Phasenabschnitt dem richtigen Phasenzyklus zugeordnet werden kann.

Es existieren dabei verschiedene Varianten von Kombinationen aus Graycode mit dem Phasenschiftverfahren, welche sich lediglich in Nuancen unterscheiden, aber aus marketingtechnischen Gründen jeweils andere Namen tragen, z.B. die *miniaturisierte Projektionstechnik (MPT-Technik)* oder die *GP-Scanning Technik*. [Brückmann 2004] [ABW 2003]

Vielfach wird auch eine Kombination aus streifenbasiertem Laser-Triangulationsscanning und mechanischem Tracking verwendet. Der Messkopf wird bei dieser Kombination über die markanten Kanten des Objektes gefahren und misst nur noch diese Partien des Objektes detailliert aus. Somit können große Objekte mit vielen planaren Flächen gescannt werden. Eine besonders originelle Variante dieser speziellen Kombination ist die Verwendung eines Ringscanners, der einen rotationssymmetrischen Aufbau des Sensors bezogen auf die Sendestrahlachse besitzt. Dies hat zum einen den Vorteil, dass weniger Abschattungseffekte auftreten, da die Kanten von allen Seiten gleichmäßig betrachtet werden, zum anderen muss keine Sensorrotation bei einer Richtungsänderung erfolgen, da auf Grund der Punktsymmetrie die Drehlage nicht wie bei einfachen Triangulationssensoren einen Einfluss auf das Messergebnis haben kann. [Wolf 2004] Da Organismen jedoch in der Regel keine planaren Flächen haben, ist diese Kombination mit ihren Varianten für die Formerfassung bei Organismen ohne Belang.

2.2.3 Bewertung der Formerfassungsmethoden

Mechanooptische Stiftverfahren und 3D-Scanning sind einfache und schnelle Methoden, um eine realistische Punktwolke der Oberfläche bzw. Volumendaten der inneren und äußeren Strukturen eines Organismus bzw. von Teilen desselben zu erhalten. Mit Hilfe der Punktwolke bzw. der Voxel-Elemente kann das entsprechende Polygonnetz der Oberflächen ermittelt werden.

Insbesondere 3D-Scanning führt – im Gegensatz zu Modellierungsverfahren am Computer – zu detailgetreuen Formen, die nur noch von Experten von der Realität unterscheidbar sind. (siehe Abbildung 26)



Abbildung 26: Beispielbilder für die Resultate von 3D-Scanning

Vier Beispiele für die überzeugenden Resultate, die mit 3D-Scanning erzielt werden können. Zu sehen sind: Ein Unterkieferknochen eines Hundes, ein vertrockneter Käfer, ein Zahnfossil eines Tyrannosaurus Rex und ein Pavianschädel. Die Objekte wurden mit dem Arius3D System gescannt, welches eine Auflösung von ca. 25µm besitzt. Es wurden auch die Farbwerte der Objekte aufgenommen. Copyright: Arius3D [Arius3D 2004] Das System von Arius3D wurde aufgrund seiner hohen Auflösung auch für die Produktion des Films „Matrix Reloaded“ verwendet. [Silberman 2003, 2]

Infolge dieser Vorteile wird 3D-Scanning seit 1993 zunehmend in der Werbe-, Spiel-, Film- und Fernsehindustrie angewandt und ist heute das Hauptstandardverfahren zur Formgebung bei Organismen. [Siebert & Ju 2000] Dazu werden in einem komplexen Prozess, der in Kapitel 1.1.2 beschrieben ist, Maquetten und Skulpturen geformt, die dann eingescannt werden. (siehe dazu auch Abbildung 27) Der Modellierungsprozess findet somit – von Verbesserungsarbeiten abgesehen – komplett außerhalb des Computers statt. [Rapido3D 2004c] [Cyberware 1999] [BBC 2004b] [Silberman 2003, 1-4]



Abbildung 27: Beispiel für 3D-Scanning von organismischen Skulpturen

Beispiel für den Einsatz des 3D-Scannings zur Organismengenerierung in Film und Fernsehen. Hier sind drei Folgestufen des 3D-Scannings einer Diprotadonta-Skulptur für den BBC-Film „Monsters we met“ zu sehen. Links: eingelezene Diprotadontaform; Mitte: Umwandlung der Oberfläche in parametrische Flächenbeschreibungen; rechts: Der Diprotadonta wird texturiert und in eine reale Szenerie integriert. [Rapido3D 2004b]

Alternativ zum Scanning von Skulpturen bzw. Maquetten können auch direkt lebende Organismen eingelezen werden. Allerdings gibt es bei der Formerfassung von Organismen verschiedene Problemzonen, die zu berücksichtigen sind. Diese Problemzonen sind im Einzelnen:

- *Feingliedrige Körperstrukturen:* z.B. Haare, Fell, Federn, Brauen und Wimpern
- *Lichtempfindliche Körperteile:* Augen und evtl. Haut
- *Körperteile mit stark konkaven Formen:* z.B. Zehenzwischenräume, Kniebeugen, Achselhöhlen, Klauen- und Fingerzwischenräume, Nabel, Nasenlöcher bzw. Nüstern und Ohrmuscheln.

Feingliedrige Körperstrukturen, wie z.B. Haare, Fell, Brauen und Wimpern haben zu feine Strukturen, um optimal gescannt zu werden. Da einige Strahlen von den Haaren reflektiert werden und andere Strahlen das Haar durchdringen, können störende Effekte entstehen, die sich durch eine chaotische Verteilung der Vertices in der erfassten Punktwolke äußern. Beim Haupthaar lässt sich dieses Phänomen aber durch einen einfachen Trick beheben. Die Haare werden mit einer opaken, reflektierenden Füllsubstanz versehen, normalerweise weißer wasserlöslicher Haarlack, und werden so als Ganzes problemlos erfasst. [Rapido3D 2004c & 2004d] [Callidus 2004b]

Die gleiche Problematik wie für das menschliche Haupthaar gilt allerdings auch für andere feine Körperstrukturen von Tieren wie z.B. Fell oder Federn, bei der die Applikation von Haarlack das Tier stören würde. Hier können die Positionen der Vertices in der Punktwolke nachträglich mit einem Gaußschen Positionsfilter geglättet werden. Brauen und Wimpern müssen allerdings normalerweise nachträglich per Hand nachmodelliert werden. [Hawkins et al. 2001]

Lichtempfindliche Körperteile stellen – zumindest bei den am weitesten verbreiteten laserbasierten Scanning-Methoden – die zweite Problemzone dar. Die meisten Laser gehören zwar – zumindest bei den Face-, den Head/Limb- und den Whole Body-3D-Scannern – standardmäßig der Laserklasse 1 (DIN EN 60825) oder 1M an und sind daher von ihrer Intensität her (in der Regel ca. 3.5 mW) zu schwach, um dem Körper zu schaden. [Arius3D 2004] Die Laserklassen 1 und 1M werden von der Deutschen Industrienorm in der Fassung vom 11.1.2001 als „augensicher“ eingestuft. Da bisher allerdings keine Erfahrungen über Langzeitexpositionen auf die Retina vorliegen, wird aus Sicherheitsgründen trotzdem empfohlen, die Augen beim Scannen zu schließen. Auch ist darauf zu achten, dass sich keine optisch sammelnden Objekte, wie beispielsweise Trinkgläser, Lupen, Flaschen oder Brillen, im Scanning-Bereich befinden, da diese den Strahl unbeabsichtigt bündeln können und dabei zu einer massiven Gefährdung des Augenlichtes (vergleichbar der Laserklasse 3B) führen könnten. [BGFE 2002]

Bei Mid-, Near- und Landscape-Scannern, die mit dem Pulslaufzeitverfahren arbeiten, ist die Laserklasse typischerweise 3R (DIN EN 60825) und kann bis zu 32mW betragen. [Zöller 2004] Die Laserklasse 3R, deren Ausgangsstrahlung im Wellenlängenbereich von 302,5nm bis 106nm liegt, ist allerdings für das Auge als „potentiell gefährlich“ eingestuft, auch wenn das Risiko eines Augenschadens dadurch verringert wird, dass der Grenzwert der zugänglichen Strahlung im sichtbaren Wellenlängenbereich auf das Fünffache der zugänglichen Strahlung für Klasse 2, in den übrigen Wellenlängenbereichen auf das Fünffache des Grenzwertes der zugänglichen Strahlung für Klasse 1 begrenzt bleibt. Daher sollte das 3D-Scanning von Organismen auf keinen Fall erfolgen, wenn diese die Augen geöffnet haben. Außerdem sind starke Sicherheitsvorkehrungen wie beispielsweise Schutzkleidung und die Anwesenheit eines Laserschutzbeauftragten zu beachten. [BGFE 2002]

Das 3D-Scanning der Augenpartie ist daher je nach verwendetem Scanner mehr oder minder problematisch. Es lassen sich nur erwachsene Personen mit geschlossenen

Augen bzw. lichtundurchlässigen Kontaktlinsen darstellen. Die für den Gesamteindruck der Person wichtige Augenpartie muss daher nach dem Scanning oft mühsam per Hand nachretuschiert werden, was zu einem unnatürlich starren und „toten“ mimischen Ausdruck führt.

Das Scannen von Tieren und Kindern verbietet sich von selbst, da nicht gewährleistet ist, dass sie die Augen während des gesamten Scanvorganges tatsächlich geschlossen halten.



Abbildung 28: Beispiele für Problemzonen beim 3D-Scanning

Beispiele für die drei Problembereiche beim 3D-Scanning von Organismen: Links: Problembereich „feine Körperstrukturen“ – obwohl hier eine Statue (Michelangelos „David“) gescannt wurde, die keine einzelnen Haarstrukturen aufweist, entsteht trotzdem eine chaotische Oberflächenstruktur. [Verdera et al. 2003] Mitte: Problembereich „lichtempfindliche Organe“ – Normalerweise gehören die Laser für offene 3D-Scanner, wie Head/Limb-, Face- oder Whole Body-Scanner, standardmäßig der Laserklasse 1 bzw. 1M an und sind daher für den Körper unschädlich. Aus Sicherheitsgründen wird jedoch trotzdem empfohlen, die Augen während des Scanvorganges zu schließen, so dass Aufnahmen von geöffneten Augen nicht möglich sind. Hier ein Beispiel für das 3D-Scanning-Ergebnis eines Auges. [Arius 3D 2004] Rechts: Problembereich „konkave Körperformen“ – da der Laser die Zehenzwischenräume nicht komplett ausleuchten kann, klafft zwischen dem großen Zeh und seinem Nachbarzeh ein unschönes Loch im Skin-Mesh. Die anderen Zehen sind dagegen gleichsam „zusammengewachsen“. [Verdera et al. 2003]

Die letzte Problemzone sind „stark konkave Körperteile“, wie z.B. die Kniebeugen, die Leistengegend, die Zwischenräume der Zehen, Klauen, Krallen und Finger, der Nabel, die Achselhöhlen, die Nasenlöcher bzw. Nüstern und die Ohrmuscheln. Sie können durch den geradlinigen Laserstrahl nicht komplett ausgeleuchtet werden. [Daanen & Jeroen 1998] [Nist 2003]

2.3 Animationstechniken und Bewegungserfassungsmethoden

Eine *Animation* ist eine Sequenz von Einzelbildern (*Frames*), die beim Betrachter die Illusion flüssiger Bewegung erzeugt. [Linke et al. 2000, 41] Im Folgenden werden diejenigen *Animationstechniken* und *Bewegungserfassungsmethoden zur Animation* vorgestellt, die bei Organismen Verwendung finden. Die Methoden der *simulativen Organismenanimation*, die ein Spezialfall innerhalb der Animationstechniken sind, da sie eine Chimäre zwischen Simulation und Animation darstellen, werden dagegen getrennt in Kapitel 2.5.2 behandelt.

2.3.1 Parametrisches Keyframing

Die Technik des *parametrischen Keyframings* basiert aus historischer Sicht auf der Entwicklung des Trickfilms im letzten Jahrhundert, als übergegangen wurde, einen talentierten Zeichner nur noch bestimmte Schlüsselszenen (*Keyframes*) als Vorlagen anfertigen zu lassen, anhand derer untergeordnete Hilfszeichner die fehlenden Sequenzen, die *In-Betweens*, zeichneten. Analog zu diesem Vorgang aus der Filmindustrie übernimmt beim parametrischen Keyframing der Benutzer die Rolle des Hauptzeichners, während der Computer in der Rolle des Hilfszeichners automatisch sinnfällige Zwischensequenzen generiert.

Im Gegensatz zur historischen Arbeitsweise in der Trickfilmproduktion wird beim parametrischen Keyframing eine parametrische Repräsentation für jedes Objekt des Keyframes benötigt, welche die Position, die Orientierung, die Skalierung und eventuell weitere Stadien des jeweiligen Objektes beschreibt.

Die *In-Betweens* werden nun durch Interpolation dieser Parameter gewonnen, wobei verschiedene Interpolationsverfahren angewandt werden, z.B. die lineare Interpolation als einfachste Variante oder höher polynomiale Interpolationsverfahren als komplexere Varianten, wie beispielsweise die stückweise kubische Interpolation mit Splines. Nach ihren Interpolationsverfahren werden demnach Keyframing-Methoden auch in *Linear Interpolation Keyframing* und *Curved Interpolation Keyframing* eingeteilt. [Watt 2000]

2.3.1.1 Linear Interpolation Keyframing

Beim Linear Interpolation Keyframing werden Geraden zwischen den zu interpolierenden Punkten aufgespannt. Die In-Betweens werden zeitlich gesehen in äquidistanten Abständen zwischen den Keyframes positioniert.

Es können daher an Keyframes abrupte Geschwindigkeits- und Richtungsänderungen auftreten.

2.3.1.2 Curved Interpolation Keyframing

Beim Curved Interpolation Keyframing können verschiedene Funktionen angewandt werden, bei denen zumindest die Stammfunktion und die erste Ableitung stetig sind. Da die erste Ableitung beim Curved Interpolation Keyframing zwischen zwei Keyframes nicht mehr konstant ist, wie beim Linear Interpolation Keyframing, werden die In-Betweens normalerweise, falls keine konstante Frame Rate gefordert ist, in unregelmäßigen Abständen erzeugt.

Als Interpolationsverfahren für Curved Interpolation Keyframing werden in der Regel stückweise kubische Splines, B-Splines oder Bézier-Kurven verwendet. Stückweise kubische Interpolation hat dabei den Vorteil, dass der Graph durch die Kontrollpunkte hindurchgehen muss, was für den Benutzer intuitiv richtig ist, wobei als Nachteil allerdings auch unerwünschte abrupte Geschwindigkeitsänderungen resultieren können, falls die Parameter nicht mit Bedacht gewählt werden.

Animationen, die von B-Splines oder Bézier-Kurven erzeugt werden, weisen sanfte Übergänge auf, haben allerdings den Nachteil, dass die Parameter der Keyframes verfälscht werden, da der erzeugte Graph nicht unbedingt auf den Kontrollpunkten liegen muss. Bézier-Kurven haben gegenüber den B-Splines den zusätzlichen Vorteil, dass sie nicht alleine durch die Position der Kontrollpunkte definiert werden, sondern zusätzlich von Tangentenpunkten, die die Steigung der Kurve toggeln. [Foley et al. 1997] [Pocock & Rosebuch 2002] [Weiss et al. 1999]

Beim *Ease-In* nimmt die erste Ableitung der interpolierten Kurve konstant zu. Mit einem *Ease-In* lassen sich daher gut Objekte mit konstanter Beschleunigung aus dem Ruhezustand darstellen, wie beispielsweise herunterfallende Gegenstände oder anfahrende Fahrzeuge. [Kerlow 1996]

Genau konträr zum Ease-In nimmt die erste Ableitung der interpolierten Kurve beim *Ease-Out* konstant ab. Mit einem Ease-Out lässt sich daher gut ein Objekt mit konstanter negativer Beschleunigung bis zum Stillstand animieren, wie z.B. ausrollende Bälle oder abbremsende Fahrzeuge. [Kerlow 1996]

2.3.1.3 Bewertung des parametrischen Keyframings

Das parametrische Keyframing ist für den Benutzer intuitiv leicht verständlich, da er als Metapher die Trickfilmproduktion benutzen kann. Die Algorithmen zur Interpolation von Parametern sind einfach zu implementieren.

Die Vorteile des parametrischen Keyframings für die Animation von Organismen liegen daher vornehmlich in ihrer metaphorischen und algorithmischen Einfachheit. In der Praxis kann es sich für den Benutzer allerdings aufgrund der Quantität der Parameter als langwierig erweisen, geeignete Keyframes zu definieren. Zusätzlich gibt es bei der Interpolation von Orientierungen Eindeutigkeitsprobleme. Die Effektivität des parametrischen Keyframings ist daher nur bei einfachen Bewegungsabläufen gegeben, wozu sicherlich nicht die komplexen Bewegungsabläufe von Organismen gehören.

Ganz besonders ungeeignet für die Organismenanimation dürfte die Technik des Linear Interpolation Keyframings sein, da hier an den Keyframes abrupte Geschwindigkeits- und Richtungsänderungen auftreten können, die bei massenbehafteten Körpern, wie sie u.a. auch Organismen darstellen, physikalisch unmöglich sind.

Mit Hilfe des Curved Interpolation Keyframings lassen sich zwar sanftere Geschwindigkeitsübergänge realisieren, trotzdem lassen sich mit dieser Technik höchstens trickfilmartige Bewegungen generieren, die mit den komplexen natürlichen Bewegungen von Organismen wenig gemein haben.

Auch Spezialeffekte des Curved Interpolation Keyframings wie z.B. das Ease-In und das Ease-Out sind eher für starre inflexible Körper oder technische Geräte geeignet, auf die ein Kraftvektor mit konstantem Betrag und Richtung einwirkt. Konstante Beschleunigungen sind jedoch bei Organismen nur in Sonderfällen gegeben.

2.3.2 Pfadanimation

Die *Pfadanimation* ist eng verwandt mit dem parametrischen Keyframing. Auch hier werden parametrische Kurven verwendet. Während beim parametrischen Keyframing jedoch einzelne zweidimensionale Graphen für einzelne Parameter angegeben werden, benutzt die Pfadanimation dreidimensionale Bewegungsgraphen, an der die Objekte

entlanggleiten. Die Bewegungsgraphen werden dabei durch stückweise kubische Interpolation erzeugt.

Von einem Bewegungsgraphen können mehrere translierte und eventuell rotierte und skalierte Instanzen erzeugt werden, die wiederum beliebig vielen Objektinstanzen zugewiesen werden können. Somit können z.B. Fischschwärme leicht animiert werden. Nach der Zuweisung eines Bewegungsgraphen auf ein Objekt wird die zeitliche Bewegungscharakteristik definiert.

2.3.2.1 Double Interpolant Method

Wird sowohl der Bewegungsgraph als auch die Bewegungscharakteristik voneinander durch zwei Interpolationsfunktionen isoliert interpoliert, wird von der *Double Interpolant Method* gesprochen. Die Double Interpolant Method stellt somit einen Spezialfall der einfachen Pfadanimation dar.

2.3.2.2 Erweiterte Pfadanimation

Die Pfadanimation kann zusätzlich weitere Parameter interpolieren, wobei die Interpolation der Orientierung sowie der Skalierung die wichtigsten und am häufigsten implementierten sind. [Pocock et al. 2002] Bei der Orientierung kann der Benutzer zwischen 3 Modi wählen:

1. die Orientierung des Objektes bleibt stets tangential zum Pfad
2. die Orientierung des Objektes wird im absoluten Koordinatensystem durch eine zweite isolierte Interpolationsfunktion berechnet
3. das Pfadobjekt ist stets auf ein anderes Zielobjekt gerichtet

2.3.2.3 Bewertung der Pfadanimation

Die verschiedenen Methoden der Pfadanimation sind algorithmisch einfach zu implementieren. Für die Generierung von Organismen besteht zudem der Vorteil, dass durch Instantiierungen schnell Fischschwärme oder Herden von Landtieren erzeugt werden können.

Allerdings sind komplexe Bewegungsabläufe, wie sie bei Organismen die Regel sind, nur unzureichend bzw. mit hohem Aufwand animierbar. Muskelflexionen und hierarchisch interdependente Bewegungsrückkopplungen sind mit Pfadanimationen nicht realisierbar. Für Organismen lassen sich Pfadanimationen daher nur anwenden, wenn es

ausschließlich auf die grobe Gesamtbewegung ankommt, wie beispielsweise bei der Erzeugung von weit entfernten Insektenschwärmen, bei denen der Betrachter nicht die Einzelbewegungen der Gliedmaßen erkennen kann.

2.3.3 Kinematik

Die Grundlage der *Kinematik* bildet ein hierarchisches Modell, welches aus verschiedenen *kinematischen Ketten* besteht. Eine kinematische Kette besteht dabei aus einer *Basis*, an die verschiedene *Glieder* über *Gelenke* angehängt sind. Das Ende des letzten Gliedes wird als *Endeffektor* bezeichnet. Wenn z.B. der menschliche Arm mittels Kinematik implementiert würde, so wäre normalerweise der Rumpf die Basis mit der Schulter als Basisgelenk, an die in hierarchischer Weise Oberarm als erstes kinematisches Glied, Ellenbogen als Zweitgelenk, Unterarm als zweites kinematisches Glied, Handbeuge als Drittgelenk und Hand als drittes kinematisches Glied angeheftet wären, wobei die Hand gleichzeitig als Endeffektor fungieren würde.

Es wird zwischen *offenen* und *geschlossenen kinematischen Ketten* unterschieden. Bei offenen Ketten ist der Endeffektor frei beweglich und nicht direkt abhängig von der Position der Basis, bei geschlossenen kinematischen Ketten ist der Endeffektor mit der Basis dagegen miteinander räumlich verbunden.

Das oben genannte Modell des Oberarms ist ein Beispiel für eine offene kinematische Kette, sofern die Hand nicht auf irgendeine Weise am Körper fixiert wäre.

Die Beine eines Snowboarders würden dagegen vernünftigerweise mittels einer kinematischen Kette animiert, die sich vom linken Fuß (Basis), linken Knöchel (1. Gelenk) über den linken Unterschenkel (1. kinematisches Glied), das linke Knie (2. Gelenk), den linken Oberschenkel (2. kinematisches Glied), das linke Hüftgelenk (3. Gelenk), das Becken (3. kinematisches Glied), das rechte Hüftgelenk (4. Gelenk), den rechten Oberschenkel (4. kinematisches Glied), das rechte Knie (5. Gelenk), den rechten Unterschenkel (5. kinematisches Glied), den rechten Knöchel (6. Gelenk) und zuletzt den rechten Fuß (6. kinematisches Glied und Endeffektor) erstreckt. Da die Füße des Unterleibes, also Basis und Endeffektor, durch das Snowboard räumlich miteinander verbunden sind, bildet sich ein mechanischer Zirkelschluss und es wird von einer *geschlossenen kinematischen Kette* gesprochen.

Der Animator hat bei der Kinematik die Möglichkeit, die Bewegungsfreiheit einer kinematischen Kette einzuschränken. Für das obige Beispiel des Oberarmes würde sich z.B. anbieten, das Gelenk, welches den Ellenbogen repräsentiert, derart einzuschränken,

dass Oberarm und Unterarm sich nur noch innerhalb einer einzigen Rotationsebene zwischen ca. 0 und 180 Grad bewegen können, um somit die Bewegungsmöglichkeiten an die Realität anzupassen. [Boulic et al. 1990] [Boulic & Thalmann 1992] [Parent 2002]

2.3.3.1 Vorwärtskinematik

Bei der *Vorwärtskinematik* werden ausgehend von der Basis entlang der kinematischen Kette sukzessive die Positionen der kinematischen Glieder berechnet. Als Eingangswerte der Berechnung dienen dabei die absoluten Koordinaten des jeweils vorherigen Kettengliedes, sowie die vom Animator gegebenen Winkelstellungen des vorherigen Gelenkes. Da die gesamte Hierarchie der kinematischen Kette nacheinander abgearbeitet wird, wird auch die Position des Endeffektors eindeutig festgelegt.

2.3.3.2 Inverse Kinematik

Die Berechnungsreihenfolge innerhalb der Hierarchie der kinematischen Kette ist bei der *Inversen Kinematik* genau entgegengesetzt zur Vorwärtskinematik. Bei der Inversen Kinematik gibt der Animator lediglich die Position des Endeffektors an. Der Computer berechnet dann sukzessive in Richtung der Basis die Positionen der Kettenglieder, mittels eines iterativen numerischen Verfahrens.

Die Inverse Kinematik bietet einen hohen Komfort für den Animator, da er sich voll und ganz auf die Positionierung der Endeffektoren und die Nebenbedingungen konzentrieren kann, was die Interaktion mit anderen Objekten wesentlich erleichtert. [Watt et al. 1992] [Parent 2002]

2.3.3.3 Bewertung der Kinematik

Die Vorwärtskinematik hat für den Programmierer den unbestreitbaren Vorteil der algorithmischen und für den Benutzer der metaphorischen Einfachheit. Sie gewährt zudem dem Animator die explizite Kontrolle über jedes einzelne Gelenk. Für die Organismengenerierung ist außerdem von Vorteil, dass miteinander verbundene Körperteile bei der Animation nicht versehentlich „auseinander gerissen“ werden können.

Die Definition von geeigneten Winkelstellungen bei der Vorwärtskinematik kann sich für den Animator allerdings sehr langwierig gestalten, da er sich in einem langwierigen Trial-and-Error-Prozess an die Winkelpositionen herantasten muss. Daher ist die Vorwärtskinematik für die Organismengenerierung nur bei einfachsten Bewegungsabläufen bedingt praxistauglich.

Zwar besteht dieses Problem bei der Inversen Kinematik nicht, aber auch hier muss der Animator Zweideutigkeiten durch eine geeignete Festlegung von Nebenbedingungen vermeiden. Darüber hinaus ähnelt die Arbeit mit der Inversen Kinematik sehr der Steuerung einer Marionette und damit muten auch die Resultate für den Betrachter „marionettenhaft“ an und haben wenig mit den feinen Bewegungsabläufen von Organismen gemein.

Die kinematischen Techniken wurden ursprünglich für Bewegungssteuerungen in der Robotik entwickelt und sind daher für Maschinen mit steifen inflexiblen Gliedern gedacht. Dieser technikorienteerte Ansatz hat implizit zur Folge, dass die bei biologischen Organismen auftretenden Muskelaufbauschungen, Faltenlegungen und andere Körperdeformationen bei den Gelenkflexionen weder bei der Vorwärtskinematik noch bei der Inversen Kinematik berücksichtigt werden. Realistische Bewegungsabläufe können daher mit kinematischen Techniken nicht generiert werden.

2.3.4 Motion Capturing

Motion Capturing, auch *Mocap* oder *Movement Tracking* genannt, ist ein Sammelbegriff für alle Verfahren, bei denen die Bewegungen eines realen Objektes, meist eines Sportlers, Schauspielers oder Tieres, digital aufgenommen und auf ein virtuelles Objekt, den *Avataren*, übertragen werden. [Campusprogramm 2004] Sturmman verwendet folgende Definition: [Sturmman 2004]

“Motion capture is the recording of human body movement (or other movement) for immediate or delayed analysis and playback. The information captured can be as general as the simple position of the body in space or as complex as the deformations of the face and muscle masses. Motion capture for computer character animation involves the mapping of human motion onto the motion of a computer character. The mapping can be direct, such as human arm motion controlling a character’s arm motion, or indirect, such as human hand and finger patterns controlling a character’s skin color or emotional state.”

Wichtig ist hierbei zu erwähnen, dass nur der interne hierarchische Körperaufbau, aber nicht die genaue Formausprägung des realen Objektes mit dem Avataren übereinstimmen muss, was es z.B. bei der Herstellung der Filmtrilogie „Herr der Ringe“ ermöglichte, den

durchaus normal gebauten Schauspieler „Andy Serkis“ auf das Monster „Gollum“ zu mappen. (siehe Abbildung 29) [FFMedia 2004]



Abbildung 29: Beispiel für Motion Capturing zur Animation von Fabelwesen

Der Schauspieler „Andy Serkis“ in einem optisch photogrammetrischen Motion Capturing System am Set von „Herr der Ringe: Die zwei Türme“. Die Bewegungen, sowie die Mimik von Serkis (links) werden auf das Monster „Gollum“ (rechts) übertragen. [FFMedia 2004]

Es existieren mehrere Möglichkeiten der Klassifikation für Motion Capturing Systeme. Mulder klassifiziert die Motion Capturing Systeme in drei Kategorien:

- inside-in,
- inside-out
- outside-in

Die Semantik dahinter ist ungewöhnlich, aber einfach: Der erste Parameter bezieht sich auf die Position des Markers bzw. Sensors und der zweite Parameter bezieht sich auf die Position der Emissionsquelle jeweils relativ zum Akteur. [Mulder 1994]

Motion Capturing Systeme lassen sich auch nach erfassten Körperbewegungen einteilen, wobei sich folgende fünf Grundkonstellationen etabliert haben: [Nodna 2004]

- Mimik (Gesichtstracker bzw. Face Tracker)
- Feingestik (Datenhandschuh bzw. Data Gloves)
- grobmotorische Körperbewegungen (Körpertracker bzw. Body Tracker)

- Bewegungen aller Gliedmaßen, Mimik und Gestik (Ganzkörpertracker bzw. Full Body Tracker)
- Bewegungen einzelner Gliedmaßen (Partielle Tracker bzw. Partial Trackers)

Im Folgenden werden dagegen die Motion Capturing Systeme nach ihrer zugrunde liegenden Technologie eingeteilt:

- Satellitennavigationsgestützt
- Mechanisch
- Magnetisch
- Akustisch
- Optisch photometrisch
- Optisch videobasiert
- Optisch stereoskopisch
- Elektromuskular

Dabei werden nur die ersten sechs Verfahren etwas ausführlicher behandelt, da sie zum Zeitpunkt der vorliegenden Arbeit technisch ausgereift sind oder in absehbarer Zukunft sich zu einem interessanten System entwickeln könnten. [Velho 1999] [TheoFou 2004] [Willier 2002] [Bevilacqua 2003]

Optisch stereoskopische Verfahren bieten kaum Vorteile gegenüber optisch photometrischen und optisch videobasierten Verfahren. Sie benötigen darüber hinaus zwei Kameras.

Elektromuskuläre Verfahren erfassen mit auf der Haut aufgebrachten Elektroden die Muskelströme des Objektes und versuchen daraus die Bewegungen zu rekonstruieren. Sie haben aber das Versuchsstadium aufgrund technischer Schwierigkeiten nicht überschritten.

Auch existieren verschiedene Kombinationen der oben genannten Verfahren. [Granum & Moeslund 2001] [Hipp 2002] [Semancik 2003] [Sturman 2004]

2.3.4.1 Satellitennavigationsgestützte Motion Capturing Systeme

Satellitennavigationsgestützte Motion Capturing Systeme verwenden ein Satellitensystem, welches zur geodätischen Navigationshilfe installiert wurde. Jeder Satellit sendet in gleichmäßigen Abständen ein gepulstes Radiosignal aus, welches von Sensoren auf der Erde empfangen werden kann. Durch den Vergleich der Laufzeitunterschiede können mittels trigonometrischen Verfahren die Empfänger ihre geodätische Position durch drei Satelliten eindeutig bestimmen. Ist ein vierter Satellit erreichbar, kann darüber hinaus die Höhe über N.N. bestimmt werden. Zum Zeitpunkt dieser Arbeit sind zwei Satellitennavigationssysteme installiert:

- Das US-amerikanische *Navstar-GPS* (NAVigation System with Time And Ranging – Global Positioning System mit 27 aktiven Satelliten. [Rothacher & Zebhauser 2000]
- Das russische *GLONASS* bzw. genauer *ГЛОНАСС* (ГЛОбальная НАвигационная СПутниковая Система) mit 10 aktiven Satelliten. [Glonass 2004]

Die Auflösung von GPS beträgt durchschnittlich ~15 Meter, diejenige der freien zivilen Frequenzen von ГЛОНАСС 57-70 Meter horizontal und 70 Meter vertikal (jeweils 99.7% Wahrscheinlichkeit). Durch den Einsatz von Differenzialsystemem, speziellen Messverfahren wie z.B. *Carrier Phase* sowie speziellen ortsgebundenen Korrektursystemen wie z.B. *ESA-EGNOS* (Europa), *WAAS* (Vereinigte Staaten), *CWAAS* (Kanada) und *MSAS* (Japan) kann diese Auflösung bis auf unter zwei Meter gesteigert werden. [Löfken 2002] [Glonass 2004] [Tews 2004] [ESA 2004]

Trotzdem hat die Auflösung eine zu geringe Granularität, um für das Motion Capturing von Organismen sinnvoll einsetzbar zu sein. Es kann höchstens zur Bestimmung von langfristigen weiten Bewegungen verwendet werden, z.B. um die Reiserouten von Zugvögeln für Forschungszwecke zu ermitteln. Kurzfristige Körperbewegungen oder gar einzelne Gliedmaßen können dagegen nicht getrackt werden. [Terrier 2000]



Abbildung 30: Beispiel für satellitennavigationsgestütztes Motion Capturing

Anwendungsbeispiel für satellitennavigationsgestütztes Motion Capturing mit Korrektursystem: Skihelm mit GPS-System und zusätzlicher geodätischer Zweifrequenzantenne zur Erfassung und Analyse der Bewegungsrouten eines Skifahrers für die Sportwissenschaft. [Blumenbach et al. 2001, S.27]

Dies kann sich allerdings schlagartig nach dem Jahre 2008 ändern, wenn das europäische Satellitennavigationssystem GALILEO mit 30 Satelliten installiert ist, da es eine absolute Auflösung von 5 bis 10m für den Basisdienst bis hin zu 10-100cm für den kommerziellen Dienst haben wird. [DLR 2004] Mit zusätzlichen Korrektursystemen sind damit absolute Genauigkeiten von ca. 2cm und relative Genauigkeiten von 0,5cm erreichbar, so dass zumindest das Tracking der Gliedmaßen von größeren Tieren möglich sein wird.

Für eine detaillierte Einführung in die Technologie der Satellitennavigation im Allgemeinen und GPS und ГЛЮHACC im Speziellen wird auf Rothacher und Zebhauser [Rothacher & Zebhauser 2000] verwiesen.

2.3.4.2 Mechanische Motion Capturing Systeme

Mechanische Motion Capturing Systeme lesen die Gelenkstellungen direkt anhand eines Exoskeletts ab, welches sich das reale Objekt anziehen muss, wobei es jeweils verschiedene Ausführungstypen hinsichtlich der verwendeten Sensoren, der Art der Datenübermittlung und der Menge und Art der erfassten Körperteile gibt:

2.3.4.2.1 Verwendete Sensoren

Die Sensoren werden in den allermeisten mechanischen Motion Capturing Systemen durch Potentiometer realisiert, die von zwei Stäben des Exoskeletts getriggert werden. Die am Potentiometer entstehenden Variationen des elektrischen Widerstandswertes werden mittels einer elektronischen Schaltung in Potentialänderungen umgewandelt, durch einen AD-Wandler digitalisiert und die entstehenden Binärwerte mit Hilfe einer entsprechenden Software interpretiert.

Es existieren aber auch mechanische Motion Capturing Aufnahmesysteme mit Kreiselsensoren, die aber zumeist nur zur Aufnahme der Hüftdrehungen benutzt werden. Auch existieren Systeme mit optoelektronischen Gelenkstellungssensoren, die eine feinere Granularität der Winkelstellungen erfassen können, als dies Potentiometer vermögen, und darüber hinaus einen langsameren Verschleiß aufweisen. [Knep 1995]

2.3.4.2.2 Datenübermittlung

Hinsichtlich der Datenübermittlung wird zwischen *drahtlosen Systemen*, welche die Daten per Funk oder Infrarot an einen Computer übermitteln, und *verdrahteten Systemen*, welche via Kabelstrang verbunden sind, unterschieden. Drahtlose Mechanische Funksysteme sind dabei als einzige Motion Capturing Variante über größere räumliche Distanzen einsetzbar.

2.3.4.2.3 Erfasste Körperteile

Mechanische Motion Capturing Systeme sind immer auf einen Bereich eines bestimmten Organismus zugeschnitten, daher muss hier – mehr als bei anderen Systemen – unterschieden werden, welche Körperteile erfasst werden können.

Mechanische Motion Capturing Systeme, die nur die Bewegungen der Hände und Finger erkennen, werden als *Datenhandschuh* bzw. *Data Glove* bezeichnet und sind weit verbreitet. [Sturman 2004]



Abbildung 31: Beispiel eines drahtlosen Datenhandschuhs

Beispiel eines drahtlosen Datenhandschuhs mit 14 Sensoren [Metamotion 2004a]

Mechanische Systeme, die nur die Bewegungsabläufe der großen Gliedmaßen aufnehmen, heißen *Körperexoskelett*, *mechanische Körpertracker* oder *Mechanical Body Tracker*.

Es existieren auch sich ergänzende synergistische Kombinationen aus Datenhandschuh und Körperexoskelett, welche nicht nur die Bewegungsabläufe der großen Gliedmaßen, sondern darüber hinaus diejenigen der einzelnen Finger erfassen und oft auch die Gestik berücksichtigen. Ein solches umfassendes mechanisches Motion Capturing System heißt *Ganzkörpertracker* bzw. *Full Body Tracker*. Ein Beispiel für einen Ganzkörpertracker ist in Abbildung 32 zu sehen. [Nodna 2004]



Abbildung 32: Beispiel eines drahtlosen mechanischen Ganzkörpertrackers

Beispiel eines drahtlosen mechanischen Ganzkörpertrackers, welcher nicht nur die Bewegungen aller Gliedmaßen, sondern auch gleichzeitig – mit optisch photogrammetrischen Methoden – die Mimik aufzeichnen kann. Copyright by NoDNA GmbH, Köln. Fotografie: Buss & Ehricke www.digital-studio.de [Nodna 2004]

2.3.4.3 Optisch photogrammetrische Motion Capturing Systeme

Optisch photogrammetrisches Motion Capturing benutzt mehrere Kameras, die in strategisch günstigen Positionen rund um das aufzunehmende Objekt aufgestellt sind. Das aufzunehmende Objekt wird durch Anbringung von *Reflektoren* oder *Leuchten* als *Positionsmarker* an möglichst vielen repräsentativen Körperstellen präpariert. Beim Menschen bieten sich hierfür Stirn, Kinn, Nacken, Schultergelenke, Schulterblätter, Ellenbogen, Handrücken, Rücken, Hüfte, Knie, Knöchel und Fußrücken als Anbringungspunkte an, damit die Bewegungen mit Ausnahme von Mimik und Feingestik annähernd vollständig erfasst werden können. Sollen Mimik und Feingestik ebenfalls aufgenommen werden, so müssen natürlich weitere Bewegungssensoren im Gesicht und an den Fingergliedern befestigt werden, wodurch die Zahl der Bewegungssensoren bei professionellen Bewegungsscans dreistellige Höhen erreichen kann. [Sturman 2004]

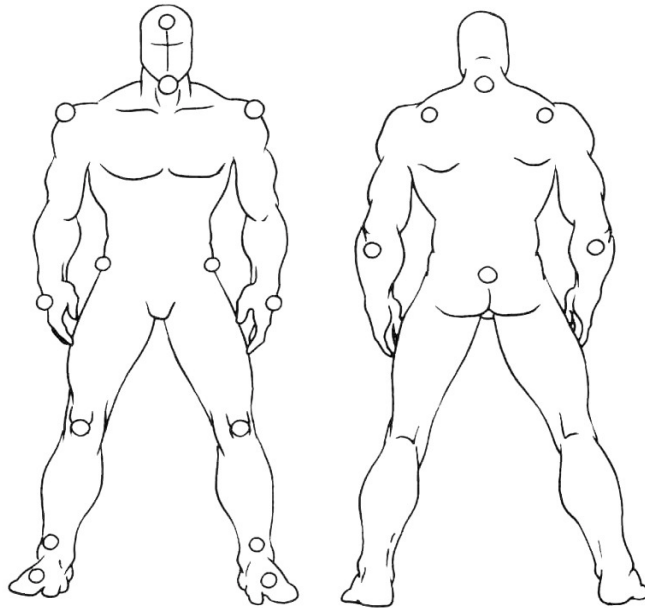


Abbildung 33: Gebräuchliche Sensorpositionen

Gebräuchliche Sensorpositionen beim Menschen für die Erfassung grobmotorischer Bewegungsabläufen [Koris 2005, S.37]

In den allermeisten Fällen zieht sich der Akteur dafür einen speziell präparierten Anzug an, auf den die Reflektoren bzw. Leuchten an den wichtigen Gelenkstellungen aufgenäht sind, so dass die mühselige Einzelanbringung der Positionsmarker entfällt. Nur für die Erfassung der Mimik müssen weiterhin die Positionsmarker einzeln aufgeklebt werden.

Bei Tieren können die möglichen Applikationspositionen der Bewegungssensoren je nach verwendeter Spezies stark variieren, in der Regel werden aber auch hier analog zum menschlichen Motion Capturing die Bewegungssensoren vornehmlich an den Gelenkstellen aufgebracht.

Zu jedem Bewegungssensor definiert der Animator explizit einen Punkt des Avataren. Die aufgenommenen Bewegungsdaten der einzelnen Bewegungssensoren werden entweder direkt oder nach einer optionalen Nachbearbeitungsphase auf die jeweils korrespondierende Punkte des virtuellen Objektes übertragen. [Silva 1999]



Abbildung 34: Beispiel für optisch photogrammetrisches Motion Capturing bei Tieren

Optisch photogrammetrisches Motion Capturing wird auch bei zahmen Tieren angewandt, hier das MoCap eines Pferdes. ©2004 Motion Analysis Corporation [Motionanalysis 2004]

2.3.4.3.1 Face Tracking

Ein Spezialfall des optisch photogrammetrischen Motion Capturing ist das *Face Tracking*. Beim Face Tracking werden kleine optische Marker an verschiedene Stellen des Gesichtes aufgebracht, um die Mimik einer Person zu erfassen.



Abbildung 35: Beispiel für Face Tracking

Beispielfoto für ein optisch photogrammetrisches Face Tracking System: Es ist eine hohe Anzahl an optischen Reflektoren notwendig, um die Mimik einer Person zufrieden stellend zu erfassen. [Carpenter 2004]

2.3.4.4 Optisch stereoskopische Motion Capturing Systeme

Beim *optisch stereoskopischen Motion Capturing* werden zwei oder mehr Kameras in leicht versetzten Positionen verwendet, die in die gleiche Richtung fokussiert sind, um das zu erfassende markerlose Objekt stereoskopisch aufzunehmen. Durch Methoden der Stereopsie werden homologe Objektpunkte in den projektiven Bildern identifiziert und mittels Triangulation die Positionen und Bewegungen vom Animator definierter Objektpunkte extrahiert. [Petrov 2002] [Koschan 2004]

Für diese Form des Motion Trackings gelten im Prinzip gleiche Schwierigkeiten, Lösungsansätze und konkrete Algorithmen wie für das stereoskopische Triangulationsverfahren. (siehe Kapitel 2.2.2.1 auf Seite 53)

Unterschiede ergeben sich allerdings daraus, dass sich das Objekt bewegt und somit Bildpunkte mit einer höheren Wahrscheinlichkeit verdeckt werden können als beim statischen 3D-Scanning. Daher müssen in der Praxis die Bewegungsdaten häufig nachmodelliert werden. [Remondino 2003] [Remondino 2004b]

2.3.4.5 Optisch videobasierte Motion Capturing Systeme

Optisch videobasiertes Motion Capturing versucht, dreidimensionale Bewegungsabläufe aus einer zweidimensionalen Bildsequenz respektive einem Film zu extrahieren, was – wie sich leicht vorstellen lässt – einer Quadratur des Kreises gleichkommt.

Dazu wird zuerst eine *Referenzfigur* benötigt, deren Hierarchie und Tiefenstruktur das zu erfassende Objekt, meist ein Mensch, repräsentiert. Für das Personentracking wird eine Referenzfigur verwendet, welche mit einem geeigneten monokularen Standbild der Videosequenz erst von der Gesamtgröße und dann von den Proportionen größenmäßig her justiert wird. Als Vergleichsobjekte zur Größenbestimmung werden auf dem monokularen Bild rechtwinklig angelegte Objekte zu Rate gezogen, deren Größe bekannt ist, wie z.B. Hochhäuser, Straßen oder Fußballplätze. [Remondino 2004a-c]

Ist die Referenzfigur erstellt, kann die Bewegung der Filmperson erfasst werden. Der Animator gibt dazu mit einem Spezialprogramm durch Mausclick die XY-Positionen aller relevanten Gelenkpositionen manuell ein. Ein Verfolgungsverfahren von Pixelmustern hilft dem Animator dabei, dass er diese Eingaben nicht bei jedem Frame wiederholen muss, sondern nur dann, wenn die Filmperson sich umdreht, nachdem sie von anderen Personen verdeckt wird oder wenn das Verfolgungsverfahren schlichtweg versagt.

Ein probabilistisches Verfahren, welches aus der justierten Referenzfigur die Entfernung der Gelenke zueinander und die möglichen Gelenkstellungen berücksichtigt, versucht nun die wahrscheinlichsten Gelenkpositionen und Winkelstellungen trigonometrisch zu kalkulieren. Diese können dann als Bewegungsdaten abgespeichert werden.

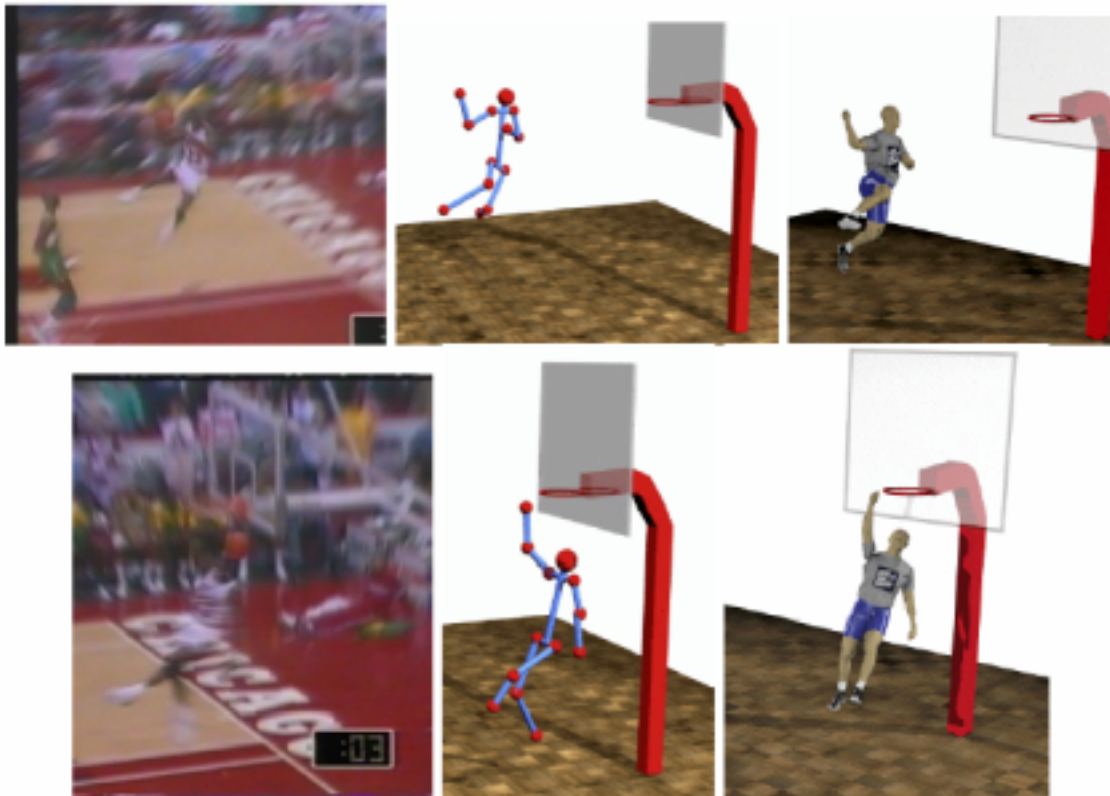


Abbildung 36: Beispiel für optisch videobasiertes Motion Capturing

Beispiel für die Verfahrensweise bei optisch videobasiertem Motion Capturing: Links sind jeweils Frames aus der originalen Videosequenz eines Basketballturniers zu sehen. In der Mitte wurde jeweils daraus ein Skelett mit Gelenkstellungen plus Umfeld extrahiert. Rechts wurden die dazugehörigen Positionsdaten auf einen Avataren gemappt. Das Bild kann nun auch unter anderen Kamerastellungen als im Originalvideo betrachtet werden. [Remondino 2004b, 5]

Es existieren außer dem oben erwähnten auch weitere abweichende videobasierte Motion Capturing Verfahren. Diese sind allerdings stets auf spezielle Mensch-Maschinen-Interaktionen spezialisiert (zum Beispiel zur Erkennung von Zeigegesten der Hand, zum Erfassung der Gesichtspose und Ähnliches) und sind daher für die Animation von Organismen ohne Belang. [Müller 2002]

2.3.4.6 Magnetische Motion Capturing Systeme

Beim *magnetischen Motion Capturing* wird ein starkes magnetisches Feld erzeugt, in dem sich das zu erfassende Objekt bewegt. Analog zum photometrischen Motion Capturing müssen am zu erfassenden Objekt Positionsmarker an repräsentativen Körperstellen angebracht werden. Allerdings werden hier *Hall-Sensoren* (Kabel, die als Antennen dienen) als Positionsmarker verwendet. Bewegt sich das Objekt, wird ein elektrischer Strom in den Hall-Sensoren induziert, welches mit einem DA-Wandler digitalisiert und dann interpretiert werden kann. [Polhemus 2004] [Ascention 2004]

Magnetische Felder können biologische Körper durchdringen, daher können Positionsmarker auch nicht vom Objekt verdeckt werden wie z.B. beim optisch photogrammetrischen Motion Capturing. Die Bewegungsdaten sind dadurch stetig, was ihre Aufbereitung vereinfacht.

Allerdings muss darauf geachtet werden, dass sich keine elektrischen Leiter wie Metallstücke im Einflussbereich des magnetischen Feldes befinden.

2.3.4.7 Akustische Motion Capturing Systeme

Beim *akustischen Motion Capturing* werden *Klicklaut-Sender*, die in regelmäßigen aufeinander folgenden Abständen Klicklaute im Ultraschallbereich aussenden, an repräsentativen Gelenkstellungen des aufzunehmenden Objektes angebracht. Zusätzlich werden mindestens drei *Audioempfänger* an strategisch günstigen Positionen des Raumes verteilt, welche diese akustischen Klicksignale einfangen. Aus der Zeitdifferenz zwischen Aussendung und Empfang kann die jeweilige Entfernung zwischen Sender und Empfänger ermittelt werden. Da mindestens drei Empfänger existieren, können auch drei Distanzen ermittelt werden, aus denen durch einfache trigonometrische Berechnungen der Raumpunkt des Senders eindeutig bestimmt werden kann. Es existieren auch Systeme, die ohne diskrete Sender direkt am Körper auskommen und stattdessen die indirekte Schallreflexion ermitteln. [Zebris 2004] [Schwertner 2001]



Abbildung 37: Beispiel für akustisches Motion Tracking

Drei akustische Motion Tracking Systeme [Zebris 2004]

Diese Methode wird mit gutem Erfolg in der Orthopädie angewandt, vornehmlich der Ganganalyse zur Diagnostik von Wirbelsäulensyndromen und zur Optimierung von Rehabilitationsmaßnahmen und ist dort als *3D-Ultraschallopometrie* bekannt. [Babisch et al. 2001] [Castro et al. 2000] [Dvir und Prushansky 2000]

Werden andere Bereiche angeschaut, außerhalb der Orthopädie und der Sportmedizin, so wird erkennbar, dass diese Technik in anderen Bereichen noch im Experimentierstadium zu sein scheint. [Schwertner 2001] [Vogt und Banzer 1997] [Silva 2000]

Der Grund dafür liegt in technischen Grenzen, die in der Natur des Schalls liegen: Erstens wird durch die sequenzielle Erfassung der Senderpositionen, welche durch die Laufzeitunterschiede entstehen, eine hohe Abtastfrequenz verhindert. Zweitens muss wegen gegenseitiger Interferenzprobleme die Anzahl der Sender sehr gering gehalten werden und sie übersteigt daher selten die kritische Marke von Sieben. Drittens können Schallreflexionen zu fehlerhaften Daten führen. Und viertens kann die Schallübertragung durch Körperteile oder Gegenstände gestört werden, analog wie bei optischen Methoden Störungen durch Verdeckungen entstehen können. [Acclaim 2002]

2.3.4.8 Bewertung des Motion Capturings

Mit Motion Capturing Methoden können – im Gegensatz zu den verschiedenen Methoden des Keyframings und der Pfadanimationstechniken – frappierend realistische Bewegungsabläufe animiert werden, da auch die feinen Interdependenzen bei Bewegungen berücksichtigt werden. Motion Capturing findet daher in der Filmindustrie eine weite Verbreitung und wird spätestens seit 1993 (bei „Jurassic Park I“) auch bei der Organismenanimation angewandt. [JurassicParc 2004] Dieser Umstand hat dazu geführt, dass die Möglichkeiten der Computergraphik beim Laien überschätzt werden, da beim

Stichwort „Virtuelle Organismen“ beim Betrachter von solchen Hollywood-Filmen nicht die Limitierungen und horrenden Kosten, sondern die frappierend realistischen Resultate evoziert werden. Wenige Betrachter erkennen, dass die Anwendungsmöglichkeiten in der Organismenanimation via Motion Capturing eng begrenzt sind:

Mechanische Motion Capturing Anzüge sind bislang nur für Menschen erhältlich. Sie werden in der Praxis häufig eingesetzt, sind aber nur bedingt geeignet, denn durch die klobigen Anzüge können die Bewegungen verfälscht werden, es muss zudem darauf geachtet werden, dass die Bewegungsaufnahme beim Datenhandschuh einige Millimeter und beim Ganzkörperexoskelett einige Zentimeter neben dem eigentlichen Gelenk stattfindet, was ebenfalls zu verfälschten Daten führen kann. Auch hat sich in der Praxis herauskristallisiert, dass sich bei Akteuren mit unterschiedlichen Körpergrößen nicht derselbe Anzug verwenden lässt, so dass ein mechanisches Motion Capturing System immer nur eine bestimmte Untermenge von Menschen (z.B. Kinder von 90cm bis 120cm) umfasst. Gestaltet sich die Erstellung von allgemein verwendbaren Motion Capturing Anzügen schon beim Menschen schwierig, so wird sie bei anderen Organismen unmöglich, da nicht nur für verschiedene Körpergrößen, sondern für jede Spezies ein separater Anzug gefertigt werden müsste. Zudem ist das Anziehen des Anzuges sowie seine Feinjustierung am lebenden Objekt bei den meisten Tieren aus nahe liegenden Gründen nicht praktikabel, Insekten und andere kleinere Tiere hätten nicht die notwendige Körperkraft, um ein Exoskelett zu bewegen, bei größeren und wilden Tieren wäre das Anziehen ohne Betäubungsspritze für die Animatoren lebensgefährlich, mittelgroße zahme Tiere würden sich mit einem Exoskelett nicht natürlich bewegen, ganz davon abgesehen, dass diese Prozedur für Mensch und Tier gleichermaßen eine Quälerei darstellen würde.

Photometrische, akustische und magnetische Motion Capturing Systeme sind für Tieranimationen zwar schon besser geeignet, aber auch hier ist das Capturing mit einem erheblichen Aufwand verbunden, da das Tier in ein Motion Capturing Studio gebracht werden, mit optischen Reflektoren bzw. magnetischen oder akustischen Sensoren versehen und dann zur gegebenen Zeit genau zu den gewünschten Bewegungen „überredet“ werden muss, was, wie sich jeder vorstellen kann, sich für das Tier zur Quälerei und für den Animator zur nervenaufreibenden und langwierigen Prozedur entwickeln kann. Beim akustischen Motion Capturing tritt ein zusätzliches Problem auf. Das auditive Frequenzspektrum von Tieren ist zumeist wesentlich breiter als beim Menschen und kann hoch in den Ultraschallbereich hineinreichen; daher können sich ultraschallsensitive Tiere

wie z.B. Fledermäuse, Hunde, Katzen, Eichhörnchen und Mäuse durch die Ultraschallklicks gestört fühlen und sie bewegen sich dadurch unnatürlich.

Bei Tiefseeorganismen, bei extrem großen Meerestieren wie z.B. Walen oder großen Haien, bei Fabelwesen sowie bei seltenen oder ausgestorbenen Tieren ist sowohl mechanisches, akustisches, magnetisches als auch photometrisches Motion Capturing naturgemäß unmöglich.

Ähnliche Probleme gelten für stereoskopisches Motion Capturing, zwar könnten theoretisch zumindest bei real existierenden Tieren stereoskopische Verfahren angewandt werden, in der Praxis ist es jedoch aufgrund der Unhandlichkeit und komplizierten Bedienbarkeit der Systeme selten möglich, scheue Tiere in freier Wildbahn stereoskopisch zu erfassen.

Bei solchen wilden Tieren kann aber, falls geeignetes Filmmaterial vorhanden ist, auf optisch videobasierte Verfahren zurückgegriffen werden. In der Praxis sind diese Verfahren jedoch nicht ausgereift und gestalten sich für den Animator zur langwierigen, monotonen Fleißarbeit, in der fast jeder Punkt für jedes Frame nachbearbeitet werden muss. Es können vor allem folgende Schwierigkeiten auftreten:

- Nicht immer ist geeignetes Filmmaterial vorhanden. Bei Fantasietieren und ausgestorbenen Tieren kann es naturgemäß keine Filmaufnahmen geben. Aber auch von seltenen Tieren existieren oftmals nur wenige aufgenommene Bewegungssequenzen, die zudem zu kurz, verwackelt, unscharf, über- oder unterbelichtet sein können oder wo die Organismen unerwünschte Bewegungen ausführen bzw. die aus zu großer Entfernung aufgenommen worden sind, so dass ihre Verwendbarkeit nicht gewährleistet ist.
- Tierfilme sind meist in freier Wildbahn aufgenommen, es existieren daher auf dem Videomaterial nur selten rechtwinklige Objekte zur Kalibrierung.
- Nicht immer ist eine Referenzform eines Organismus vorhanden.

- Ein Organismus kann in unterschiedlichen Posen vor verschiedenen Hintergründen aufgenommen worden sein. Es ist daher nicht immer möglich, anhand einer statischen Referenzform die Tiefenstruktur zurückzurechnen.
- Das Erscheinungsbild eines Organismus kann dramatisch zwischen zwei Frames variieren, so dass sich die automatische Verfolgung von Referenzpunkten sehr schwierig bzw. teilweise unmöglich gestaltet.
- Die Organismen können während ihrer Bewegung zeitweise von anderen Objekten partiell oder total verdeckt werden, so dass Referenzpunkte interpoliert werden müssen und Bewegungsabläufe damit unter Umständen verfälscht rekonstruiert werden.

Satellitennavigationsgestütztes Motion Capturing könnte rein theoretisch auch bei frei lebenden wilden Tieren angewandt werden, allerdings ist die Auflösung dafür noch viel zu grobkörnig. Lediglich die Reiserouten und Wanderbewegungen der Tiere können getrackt werden. Für die Erfassung von Bewegungsabläufen einzelner Gliedmaßen ist satellitennavigationsgestütztes Motion Capturing derzeit nicht geeignet.

So bleibt als einzige Zukunftsoption ab 2008 satellitennavigationsgestütztes Motion Capturing mit dem GALILEO-System der europäischen Gemeinschaft. Da es mehrere Vorteile der anderen Systeme, wie Echtzeitfähigkeit, Leichtigkeit, Genauigkeit, Okklusionsvermeidung und Ortsungebundenheit vereint, ist zu erwarten, dass dieses System zukünftig vermehrt eingesetzt und langfristig die anderen Systeme verdrängen wird.

Allerdings wird selbst die Positionsauflösung des GALILEO-Systems mit wenigen Zentimetern noch viel zu grobkörnig sein, um für die Animation von kleineren Organismen oder Pflanzenbewegungen, wie z.B. das Blätterrauschen eines Baumes oder das Wiegen der Algen durch Meereswellen, verwendet zu werden, und auch hier gilt, dass logischerweise nichts getrackt werden kann, was nicht existiert bzw. nicht erhältlich ist. Fabelwesen, paläontologische Tiere und seltene Tiere sind höchstens indirekt durch den Einsatz von Drittorganismen erfassbar; ob die Bewegungen dann biologisch korrekt sind, steht auf einem anderen Blatt.

Es bieten somit alle Verfahren nur wenige Anwendungsmöglichkeiten bei Organismen. Wenn das bisher Gesagte tabellarisch zusammengefasst wird, wie in Tabelle

1 auf Seite 91, in der spaltenweise die Motion Capturing Technologien stehen und zeilenweise Organismenklassen mit verschiedenen Aufnahmeproblematiken aufgetragen sind, wird diese Begrenztheit besonders anschaulich, denn wird vom Menschen abgesehen, stehen in fast allen Zellen „nein“ oder „bedingt“.

	mechanisch	magnetisch	akustisch	satellitennavi- gationsgestützt	photometrisch	videobasiert	stereoskopisch
Menschen	ja	ja	ja	zukünftig	ja	ja	ja
Haustiere	nein	ja	nein	zukünftig	ja	bedingt	bedingt
große Tiere	nein	nein	nein	zukünftig	bedingt	bedingt	nein
kleine Tiere	nein	nein	nein	nein	bedingt	bedingt	bedingt
wilde Tiere	nein	nein	nein	zukünftig	nein	bedingt	nein
seltene Tiere	nein	nein	nein	nein	nein	nein	nein
ausgestorbene Tiere	nein	nein	nein	nein	nein	nein	nein
imaginäre Tiere	nein	nein	nein	nein	nein	nein	nein
Pflanzen (Wind- und Wellenbewegungen)	nein	nein	nein	nein	nein	nein	nein

Tabelle 1: Anwendungsmöglichkeiten von Motion Capturing bei Organismen

Anwendungsmöglichkeiten der verschiedenen Technologien bei Motion Capturing Systemen (Spalten) auf verschiedene Organismenklassen mit verschiedenen Aufnahmeproblematiken (Zeilen). Es wird deutlich, dass Motion Capturing nur in Spezialfällen anwendbar ist.

Abgesehen von den bisher aufgezeigten technischen Schwierigkeiten existieren bei Motion Capturing finanzielle Hürden. So können die Kosten für diese Motion Capturing Systeme mehrere Zehntausend Euro betragen, so dass sich diese Methoden nicht für den Low-Budget-Bereich eignen. Aufgrund der Komplexität und der rein physikalischen Größe der Systeme, die eine kostengünstige Produktion nicht zulassen, ist auch nicht zu erwarten, dass dies sich in der nahen Zukunft entscheidend ändern wird. Einen „Volkstracker“ für den Low-Budget-Bereich wird es jedenfalls nicht geben. [Campusprogramm 2004] [Metamotion 2004c]

2.3.5 Stop Motion Input Devices

Stop Motion Input Devices bzw. *Direct Input Devices* sind bewegliche Metallskelette (*Armaturen*), mit Positions- und/oder Winkelsensoren an den Gelenken. Der Animator kann nun Frame für Frame das Skelett in die gewünschte Stellung und Position bewegen und die Sensordaten inkrementell in einen Computer übertragen lassen. [Knep 2003] [Bennett 1995]

Stop Motion Input Devices verbinden die Technologie des Motion Capturing mit dem traditionellen Stop Motion Verfahren in der Filmindustrie. Sie bieten die intuitive Metapher der Stop-Motion-Armatur und lassen sich daher von traditionellen Filmamateuren auch ohne Computererfahrung und Umschulungsmaßnahmen sofort bedienen. [Knep 1995] [Robertson 1997]

Mittlerweile sind auch Input Devices auf dem Markt, welche sich in Echtzeit bewegen lassen und somit eine Chimäre zwischen Motion Capturing und Stop Motion Input Devices darstellen. [Vaz & Duingan 1996] Sie werden aufgrund der historischen Entwicklungslinie trotzdem in diesem Kapitel behandelt.

2.3.5.1 Das Dinosaur Input Device

Das historisch erste Stop Motion Input Device, das *Dinosaur Input Device* oder kurz *DID*, wurde 1993 speziell für den Film „Jurassic Park I“ von Steven Spielberg entwickelt und besteht aus einem dinosaurierähnlichen Metallskelett, an dessen Scharnieren optoelektronische Winkelsensoren angebracht sind, sowie einer Kontrollbox inklusive Software.

Für „Jurassic Park I“ wurden vier DIDs gebaut, zwei Velociraptor-Tracker und weitere zwei Tyrannosaurus-Rex-Tracker. [Shay & Duncan 1993]

15 der 52 computeranimierten Shots in „Jurassic Park I“ wurden mit diesen DIDs erstellt, darunter die Attacke eines Touristen-Jeeps durch einen Tyrannosaurus Rex sowie die Jagdsequenzen mit den Velociraptoren und den Kindern in der Küche. [Robertson 1997]

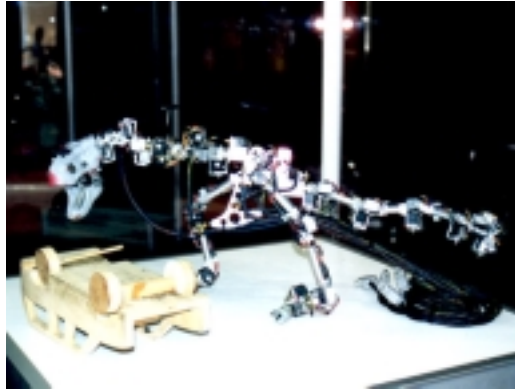


Abbildung 38: Beispiel für ein Dinosaur Input Device

Ein Dinosaur Input Device für einen Tyrannosaurus Rex. [Vaz & Duingan 1996] [Siggraph 2004]

2.3.5.2 Modulare DIDs

Das Dinosaur Input Device kann – in veränderter Form – nicht nur für Dinosaurier, sondern für beliebige Arten von Tieren angewandt werden, trotzdem hat sich der Name Dinosaur Input Device auch bei diesen Anwendungen gehalten. [Knep 1995] Manchmal wird allerdings der Tiername speziell genannt (*Human Figure Input Device, Spider Input Device, etc.*). Monkey Inc. bietet hierfür *modulare DID-Systeme* an, mit denen sich beliebige DID-Tiere aus einem Set von *DID-Basiskomponenten* zusammenstecken lassen. Im Gegensatz zum ursprünglichen DID werden magnetische Sensoren verwendet. [Monkey 2004]

Magnetische Systeme bieten zwar nicht die Genauigkeit mechanischer DIDs, arbeiten jedoch in Echtzeit und können daher nicht nur via Stop Motion Technik, sondern alternativ von speziell ausgebildeten Puppenspielern bedient werden. [Knep 1995]



Abbildung 39: Beispiel für modulare DID-Systeme

Eine DID-Spinne, aufgebaut aus modularen DID-Basiskomponenten. [Monkey 2004]

Modulare DID-Systeme werden zunehmend bei einigen Organismenanimationen in der Filmindustrie angewendet, z.B. bei den Filmen „The Lost World“ oder „Starship Troopers“ [Starship 1997]

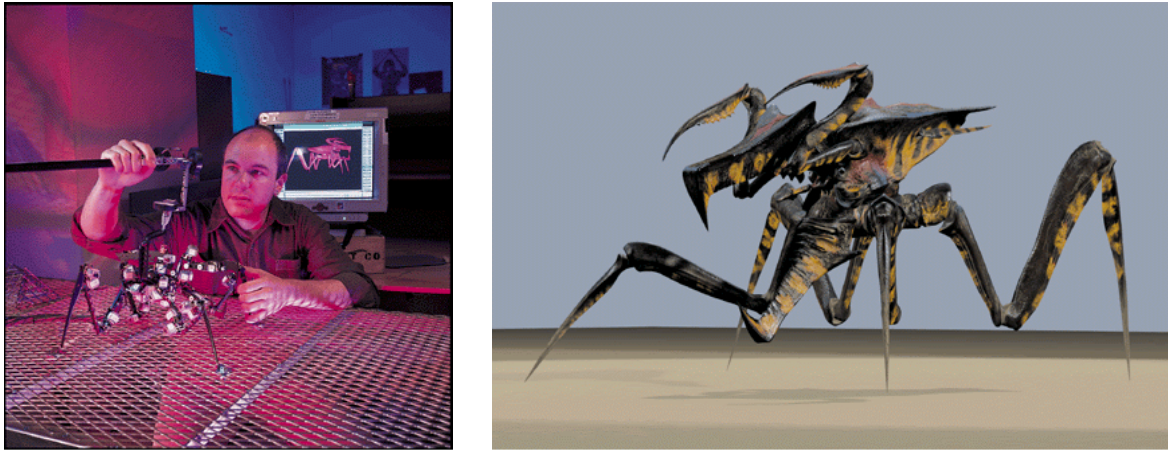


Abbildung 40: Beispiel für die Anwendung modularer DID-Systeme

Die krabbenartigen Aliens (rechts) aus dem Film „Starship Troopers“ von Paul Verhoeven sind hauptsächlich mit Hilfe modularer DID-Systeme (links) animiert. [Starship 1997] [Robertson 1997]

2.3.5.3 Bewertung von Stop Motion Input Devices

Die Technologie der Stop Motion Input Devices bietet eine indirekte Bewegungserfassungsmethode, die auch die Animation von komplexen Bewegungsabläufen von Tieren erlaubt.

Für die Filmindustrie ist diese Methode von Interesse, da sie vorhandene traditionelle Stop-Motion-Fachkräfte auch ohne spezielle Umschulung zu computergraphischen Verfahren direkt einzusetzen vermag. Die Entwicklung des Stop Motion Input Devices erlaubte in den neunziger Jahren einen nahtlosen Übergang zwischen armaturbasierter und computerbasierter Filmproduktion.

Durch die Modularisierung sind Stop Motion Input Devices zumindest theoretisch flexibel auf alle möglichen Arten von Tieren anwendbar, die ein internes Skelett bzw. einen externen stabilisierenden Panzer besitzen. Die Animation von Enteropneusten, Schnecken, Pogonophoren, Cestoden, Turbellarien, Nematoden, Cephalopoden, Oligochaeten, Hirudineen, Mollusken, Seelilien, Quallen und sonstigen „weichen“ Organismen ist dagegen nicht bzw. nur sehr bedingt möglich.

Diese Flexibilität hinsichtlich der Anwendbarkeit auf fast beliebige Tierkonstruktionen gilt allerdings nur theoretisch, denn es ist bezeichnend, dass Stop Motion Input Devices in der Filmindustrie in der Regel für die Animation von extraterrestrischen Wesen und sonstigen Fantasietieren bzw. Dinosauriern verwendet werden, also Organismen, die real nicht existent und deren Bewegungen daher dem Betrachter unbekannt sind – vermutlich würde dem Betrachter bei rezenten Tieren sofort die Unnatürlichkeit der Bewegungen auffallen, wie es auch beim traditionellen Stop Motion mit Tierpuppen der Fall ist.

2.4 Basistechniken zur Formveränderung

2.4.1 Geometrische Methoden zur Objektdeformation

Die *Geometrischen Methoden zur Objektdeformation*, engl. *geometrical shape changing*, werden häufig für biologisch anmutende Animationen benutzt.

Bei der gewöhnlichen *Scherung* oder *Skalierung* wird eine Transformationsmatrix gleichmäßig auf jeden Eckpunkt eines Objektes angewandt. Die Innovation bei der *Nichtlinearen Globalen Deformation* nach Barr ist es, die Transformationsmatrix für jeden Punkt von der Position desselben abhängig zu machen, womit *Tapering-*, *Twisting-* und *Bending-*Effekte erzielt werden. [Barr 1984]

Bei der *Freiform-Deformation (FFD)* nach Sederburg und Parry wird das Objekt in ein umgebendes Raumgitter eingefasst, welches durch Modellingtransformationen deformiert wird. Die Deformationen des Raumgitters werden dann auf das Objekt übertragen. [Sederburg & Parry 1986]

Coquillard führt bei der *erweiterten Freiform-Deformation (EFFD)* neben der Rechteckstruktur der FFD-Gitter noch weitere EFFD-Gitterstrukturen ein. [Coquillard 1990]

Bei den genannten Gitterdeformationen kann jeweils zwischen *polygonalen Modellen*, bei denen die Eckpunkte, und *parametrischen Modellen*, bei denen die Kontrollpunkte verändert werden, unterschieden werden.

2.4.2 Simulationsmethoden zur Objektdeformation

Simulationsmethoden zur Objektdeformation versuchen, die Verformung des Objektes mit numerischen Methoden diskretisiert zu berechnen. Dazu wird das Objekt in eine endliche (*finite*) Anzahl von Untereinheiten aufgeteilt, und jede Untereinheit isoliert berechnet. Als Berechnungsgrundlage dienen mechanische Gesetze wie die z.B. von *Green*, *Gauss*, *Navier*, *Stroke* oder *Hooke*.

Es existieren unzählige Methoden zur Simulation von Objektdeformierungen. Aufgrund ihrer Menge und der Ähnlichkeit ihres zugrunde liegenden diskretisierenden Simulationsparadigmas, werden im Folgenden nur einige exemplarische Methoden kurz gestreift:

2.4.2.1 Feder-Masse-Dämpfer-Systeme

Die verschiedenen *Feder-Masse-Dämpfer-Systeme*, welche im Englischen als *spring mass systems*, *spring mass damper systems* oder *SMD systems* bezeichnet werden, verteilen die Gesamtmasse m_{ges} des zu simulierenden Objektes auf n verschiedene Massepunkte bzw. Partikel. Die Masse eines Partikels erhält dadurch den konstanten Wert $m_i = m_{ges} / n$. [Sarni et al. 2004] Benachbarte Massepunkte sind dabei durch imaginäre masselose elastische Elemente, den Federn, miteinander verbunden, deren Steifheit durch die Federkonstante k angegeben wird. Durch Variation der Federkonstante lassen sich die Elastizität, Flexibilität und Dehnungsfestigkeit des Objektes modifizieren.

Parallel zu den Federn existieren imaginäre masselose Dämpfungszylinder, die notwendig sind, damit sich wie in der Realität eine einmal angeregte Schwingung im Feder-Masse-Dämpfer-System auspendelt und schließlich zur Ruhe kommt. (siehe Abbildung 41) Die Stärke der Dämpfungszylinder wird durch die Dämpfungskonstante c_v repräsentiert.

Je höher die Anzahl der Partikel in einem Feder-Masse-Dämpfer-System ist, desto genauer und realistischer ist die Simulation.

Es existieren verschiedene Feder-Masse-Dämpfer-Modelle, die sich einerseits durch die Vernetzungstopologie der einzelnen Partikel untereinander und andererseits durch die genauen Feineinstellungen von Federkonstanten, Dämpfungskonstanten und Partikelmassen unterscheiden. [Efunda 2004b]

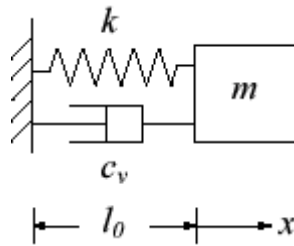


Abbildung 41: Schema einer Feder-Masse-Dämpfer-Funktionseinheit

Schema einer Funktionseinheit mit Feder, Masse und Dämpfungszyylinder sowie den Größen k (Federkonstante), c_v (Dämpfungskonstante), m (Masse des Partikels), l_0 (Ruhelänge der Feder) und x (Ruheposition des Partikels) [Efunda 2004a]

Feder-Masse-Dämpfer-Modelle werden im zunehmenden Maße erfolgreich für die Simulation von weichen, deformierbaren Einzelobjekten angewendet. Die derzeitigen Anwendungen reichen von Textilsimulationen für die Kleidungsindustrie über Mimiksimulation bei Avataren bis hin zur Gewebesimulation bei der cranofacialen Schönheitschirurgie. [Keeve et al. 1997] [Sarni et al. 2004]

2.4.2.2 Finite Elemente Methoden

Auch *Finite Elemente Methoden*, kurz *FEM*, werden für die Simulation von biologischen Strukturen verwendet. Dafür wird das zu simulierende Gewebe in eine endliche Anzahl von Teilgebieten, die *Finiten Elemente*, eingeteilt. Normalerweise wird zur Einteilung des Gewebes in Finite Elemente ein dreidimensionales äquidistantes orthogonales Gitter verwendet. Jedes Finite Element interagiert dann über numerisch-mechanische Differenzialgleichungen mit seinen sechs Nachbarelementen. Die Finiten Elemente korrespondieren bei einer äquidistant orthogonalen Gewebeeinteilung mit entsprechenden Voxelparametern eines passenden Volumendatensatzes. Finite Elemente Methoden eignen sich daher besonders gut in Kombination mit Volumendaten.

Bei der *Finite Elements Stress Analysis* bzw. *FESA* werden rekursiv die wirkenden mechanischen Kräfte Element für Element betrachtet. Die Vektoren extern wirkender Kompressionskräfte werden für die Berechnung in ihre orthogonalen Kraftbestandteile aufgespaltet. Diejenigen Finiten Elemente, bei denen die Kräfte ansetzen, werden mit diesen Werten initialisiert. Infolge der gitterartigen Verknüpfungstopologie werden die Kräfte nach der Kraftableitung durch numerisch-mechanische Formeln im Gitter bei jedem Rekursionsschritt orthogonal an die benachbarten Finiten Elemente weitergegeben, bis sich

im System ein Kräftegleichgewicht einstellt. Aus den resultierenden Kraftverteilungen im Gleichgewichtszustand lässt sich die mechanische Beanspruchung verschiedener Regionen eines Objektes (in der Regel eines Knochens) ableiten.

So verwenden z.B. Chen und Zeltzer einen solchen Ansatz zur Berechnung der mechanischen Kräfte in den Knochen und Knorpeln des menschlichen Knies. [Chen & Zeltzer 1992] Auch für paläobiologische Fragestellungen wurde FESA eingesetzt, um wirkende Kräfte in Schädel-, Kiefer- und Extremitätenknochen zu simulieren. [Fastnacht et al. 2002] [Jenkins et al. 2002] [Preuschoft & Witzel 2002] [Preuschoft & Witzel 2004] [Snively & Russell 2002] [Stokstad 2001]

Zu beachten ist, dass in Zusammenhang mit Finite Elemente Methoden eine Begriffsverwirrung herrscht und andere, lediglich verwandte Methoden (Desbrun, Cani, Wu, etc.) fälschlicherweise oft dieser Gruppe zurechnet werden.

2.4.2.3 Randelementmethoden

Bei den *Randelementmethoden* bzw. englisch *Boundary Element Methods* oder kurz *BEM* werden – im Gegensatz zu Finiten Elementen – nur Massepunkte an den Oberflächen des Objektes berücksichtigt und diskretisiert. Dazu verwendet BEM die Integralsätze nach Green, Gauss und Stroke. Diese Integralsätze werden in Differenzialoperationen transponiert und zu einem numerischen Gleichungssystem zusammengesetzt. Da somit die Berechnung der inneren Massepunkte entfällt, hat die Methode eine lineare Komplexität bezüglich der Anzahl der Oberflächenmassepunkte. Da in der Computergraphik die Massepunkte aus pragmatischen Gründen meist identisch mit den Mesh Vertices sind, verhält sich die Geschwindigkeit auch umgekehrt proportional zu den Mesh Vertices, so dass Deformationen kleinerer Objekte auch in Echtzeit möglich sind. Nachteilig ist, dass stärkere und komplexe Deformationen, bei der auch die Innenbereiche des Objektes mitdeformiert werden, nicht realistisch berechnet werden können. [Wrobel & Aliabadi 2002]

2.4.2.4 Weitere Simulationsmethoden zur Objektdeformation

Finite Volumen Methoden, kurz *FVM* unterteilen das Objekt – analog zu Finite Elemente Methoden – in quaderförmige Untereinheiten. Die Ansatzfunktionen beziehen sich allerdings nicht auf die entsprechenden Gitterpunkte, sondern auf die Volumina der Untereinheiten. [LeVeque 2002]

Finite Differenzen Methoden, kurz *FDM* stellen einen Mittelweg zwischen Feder-Masse-Dämpfer-Systemen und und Finiten Elementen dar. Das Objekt wird in kleine quaderförmige Bereiche zerlegt. Statt Differenzialgleichungen verwenden FDM einfache Differenzengleichungen an den Schnittpunkten, um – im Falle der Organismensimulation – die Deformationen der unterteilten Gewebebereiche zu approximieren.

Eine spezielle FDM stellt das *hybride Methode von Terzopoulos und Witkin* dar. Das der Methode zugrunde liegende Modell teilt ein deformierbares Objekt in einen starre und eine deformierbare Komponente ein. Die starre Referenzkomponente berechnet dabei die Gesamttranslation und -rotation des Körpers, während die deformierbare Komponente die Positionen der Mesh Vertices relativ zum lokalen Bezugssystem der starren Referenzkomponente mittels DDFs berechnet. [Terzopolous & Witkin 1988]

Das *Modell von Desbrun et al.* teilt die Kräfte in dreidimensionalen Feder-Masse-Dämpfer-Systemen in lineare und nichtlineare Kraftkomponenten auf. Die nichtlinearen Kraftkomponenten werden zunächst ignoriert, um eine rasche Annäherung an die tatsächliche Deformation zu erreichen, erst dann werden die nichtlineare Kraftkomponenten berücksichtigt, um die Deformation genau zu berechnen. [Desbrun et al. 1999]

Beim *Modell von Debunne et al.* wird die Größe der Untereinheiten dynamisch an die Stärke der Deformationen angepasst. In Bereichen extremer Deformation wird der Subbereich rekursiv weiter unterteilt, um auf diese Weise eine Art Level-of-Detail der Deformationsberechnung zu erreichen. [Debunne et al. 1999, 2000 & 2001]

Auch das *Modell von Grinspun et al.* ähnelt dem Modell von Debunne et al. Es setzt allerdings hohe Auflösungen an scharfen Kanten und Falten ein. [Grinspun et al. 2002]

Das *Modell von Wu et al.* verwendet dynamische Progressive Meshes, um die Granularität der Deformationssimulation gezielt an die aktuelle interne Deformationstärke anzupassen. [Wu et al 2001]

Eine Kombination zwischen Finiten Elementen und Feder-Masse-Dämpfer-Systemen, welche von linearer Komplexität ist und auch in der Praxis bis zu einem gewissen Grad echtzeitfähig ist, stellt die *Methode von Müller et al.* dar. Das Verfahren ist stabil und Deformationen lassen sich physikalisch korrekt simulieren. [Müller et al. 2002]

2.4.3 Interpolationsmethoden zur Formveränderung

Die *Interpolationsmethoden zur Formveränderung* interpolieren entweder auf indirekte Weise zwischen zwei Referenzbildern oder auf direkte Weise zwischen zwei Referenzformen.

Beim *Blending* handelt es sich um eine einfache Überblendungstechnik zwischen zwei Keyframes, bei der ein Schlüsselbild ausgeblendet und das andere gleichzeitig eingeblendet wird.

2D-Morphing-Techniken interpolieren die Positionen der korrespondierenden Pixel zwischen zwei Schlüsselbildern. Meist wird parallel auch die Farbe der Pixel interpoliert, so dass ein weicher Übergang entsteht.

3D-Morphing-Techniken interpolieren die Positionen korrespondierender Vertices im Skin-Mesh. Hier handelt es sich also um eine Interpolation der Hüllform.

Es gibt viele Varianten der 2D- und 3D-Morphing-Techniken. Die meisten unterscheiden sich in der Verknüpfungsart der korrespondierenden Pixel bzw. Vertices, möglich sind vollautomatisierte, halbautomatisierte und dezidierte Verknüpfungen per Hand durch Bereichsangaben oder durch die Angabe von Referenzpunkten. [Lee et al. 1999]

2.4.4 Bewertung der Basistechniken zur Formveränderung

Die verschiedenen geometrischen Methoden zur Objektdeformation, wie beispielsweise die gewöhnliche Scherung und Skalierung, die Nichtlineare Globale Deformation nach Barr, die Freiform-Deformation nach Sederburg und Parry sowie die erweiterte Freiform-Deformation nach Coquillard, sind in der Organismenanimation häufig verwendete Techniken, die infolge ihrer Einfachheit und algorithmischen Performanz auch für Echtzeitanwendungen praktikabel sind.

Sie sind jedoch mathematisch-geometrisch motiviert und werden bei der Organismenanimation lediglich für biologische Zwecke „missbraucht“. Dies erklärt auch

die Tatsache, dass die mit ihnen erzeugten Körperdeformationen biologisch inkorrekt sind. So bleibt z.B. das Volumen während einer Deformation nicht konstant. Die komplexen Oberflächensubdeformationen, welche bei einer Gelenkflexion durch das interaktive Muskelspiel erzeugt werden, bleiben bei diesen Techniken unberücksichtigt und müssen umständlich mittels anderer Methoden generiert werden.

Feder-Masse-Dämpfer-Systeme vermögen reine Oberflächendehformationen ohne Tiefenwirkung bzw. Flächendehformationen elegant und realistisch zu simulieren. Ein Nachteil von Feder-Masse-Dämpfer-Systemen ist allerdings die mangelnde Robustheit hinsichtlich zyklischer Interdependenzen. Topologien von Feder-Masse-Dämpfer-Systemen, die hochgradig zyklisch sind, können zu unerwünschten Rückkopplungseffekten führen. Bestenfalls werden nur die Simulationsergebnisse durch die Interaktion verfälscht; wenn die Eigenresonanz des Systems jedoch ein Vielfaches der Simulationsfrequenz ist, kann sich die Simulation im schlimmsten Fall asymptotisch aufschaukeln.

Anatomische Strukturen besitzen naturgemäß eine Reihe zyklischer Interdependenzen in ihrem mechanischen Verhalten. Ein einfaches und häufiges Beispiel hierfür ist die mechanische Interaktion zwischen Muskel und darüber liegender Hautschicht. Die Haut deformiert sich durch darunter liegende Muskelflexionen und die Muskelflexion wird durch die Spannung der Haut in einem gewissen Maße beeinflusst. Darüber hinaus interagieren sowohl die Haut als auch der Muskel jeweils mit sich selbst in einer zyklischen Weise.

Einen weiteren Nachteil stellt die algorithmische Performanz von Feder-Masse-Dämpfer-Systemen dar, die bislang nur die Simulation kleinerer organischer Einheiten in Echtzeit erlaubt. So benötigt die Berechnung eines Partikels schätzungsweise ca. 240 Gleitkommaoperationen. Daher können zum Zeitpunkt dieser Arbeit Feder-Masse-Dämpfer-Systeme mit einer Größenordnung von ca. tausend Partikeln in Echtzeit berechnet werden (Annahme: Simulation mit 20 fps auf einem durchschnittlichen PC mit 4,2 MIPS).

Somit werden Feder-Masse-Dämpfer-Modelle zwar langsam für interaktive Echtzeitsimulationen von einzelnen organischen Gewebeschichten interessant. Allerdings reicht die Performanz von PCs zurzeit und auch in der nahen Zukunft noch nicht aus, um die Anatomie komplexer Organismen mit einer Partikelanzahl von weit über einer Million Partikel zu simulieren. Dies wird voraussichtlich erst ab ca. 2020 möglich sein. Die Echtzeitsimulation von ganzen Tierpopulationen wird sogar erst in den dreißiger Jahren

dieses Jahrhunderts und die Simulation der evolutionären Veränderungen von Tierkonstruktionen sogar erst in den vierziger Jahren dieses Jahrhunderts in den Bereich des Machbaren rücken. Für die Zwischenzeit muss nach performanten Alternativen gesucht werden. [BBC 2003][Fraunhofer 1998][Frc 2002][Intel 2003 & 2004][Kish 2002][Kurzweil 2003][Mac 2004][Moore 1965, 2004 & 2003][Moravec 1998a & 1998b][ScienCentral 1999][Thocp 2004]

Mit Finiten Elemente Methoden im Allgemeinen und FESA-Techniken im Speziellen können mechanische Kraftverteilungen recht genau berechnet werden.

Finite Elemente basieren in der Regel auf Volumendaten. Dies ist einerseits ein Vorteil, da die Daten, die aus medizinischen 3D-Visualisierungstechniken (z.B. CT, MRT) gewonnen werden, unmittelbar verwendet werden können, andererseits aber auch ein gravierender Nachteil, da es nicht möglich ist, anatomische Strukturen ohne einen solchen Datensatz zu simulieren.

Ein Ausschlusskriterium für die meisten Anwendungen in der Organismensimulation ist, dass Finite Elemente Methoden gemäß der Recherche bislang nur bei weitestgehend starren und unflexiblen Bereichen von Organismen erfolgreich eingesetzt werden, vornehmlich bei Chitinpanzer, Knochen und Knorpel. Für stark deformierbare Bereiche des Organismus, die normalerweise den größten Teil der Anatomie ausmachen, existieren bislang keine praxistauglichen Ansätze über Finite Elemente Methoden.

Nachteilig bei Finite Elemente Methoden ist weiterhin, dass ihre hohen Anforderungen an die Rechnerperformanz kubisch mit dem zu verwendenden Detailgrad wachsen und somit auf absehbare Zeit keine Echtzeitanwendungen zulassen werden. [Parent 2002, 355]

Für Finite Differenz Methoden und Finite Volumen Methoden gelten prinzipiell die gleichen Vor- und Nachteile wie für Finite Elemente Methoden, sie benötigen Datensätze, die auf Volumendaten basieren, versagen bei starken, globalen Deformationen und sind nur geringfügig performanter als Finite Elemente Methoden.

Randelementmethoden und insbesondere die neueren Simulationsmethoden zur Objektdeformation wie diejenigen von Cani, Debunne, Desbrun, Grinspun, Terzopoulos, Müller, Witkin, Wu und Zhuang et al. stellen Alternativen zu den klassischen Finite Elemente, Finite Differenz und Finite Volumen Methoden dar und sind wesentlich

performanter, so dass sie mittlerweile auch in der Praxis mit Erfolg für interaktive Gewebesimulationen eingesetzt werden. [Debunne et al. 2001, Wu et al. 2001]

Die Simulation komplexerer Gebilde ist dagegen – bei genauerem Hinsehen – zurzeit trotzdem nicht mit diesen neuen Methoden in Echtzeit möglich. Denn um die Echtzeitfähigkeit in der Praxis zu erreichen, verwenden beispielsweise die behandelten Methoden von Debunne, Wu und Zhuang nur sehr kleine Objekte mit lediglich einigen hundert Massepunkten und mit lokal begrenzten Deformationen. Obendrein werden die Demonstrationen bislang auf den besten Hochleistungsrechnern ausgeführt und sind somit nicht für Standard-PCs tauglich. [Müller et al. 2002]

Eine Ausnahme bildet das Verfahren von Müller et al., dass nach eigenen Angaben aufgrund seiner linearen Komplexität zu den performantesten Simulationsmethoden zur Objektdeformation zählt. Allerdings bezieht sich die Linearität bei näherem Hinsehen nicht allein auf die Anzahl der Vertices im Skin-Mesh, sondern zusätzlich auf die verwendeten „strain tensors“. Auch muss eine Vorverarbeitung zur Berechnung einer Steifheitsmatrix erfolgen. Somit wird die Berechnung dynamischer Organismen, die einem Wachstums- oder Evolutionsprozess unterliegen, nicht in Echtzeit unterstützt. [Müller et al. 2002]

Interpolationsmethoden zur Formveränderung werden oft zur Visualisierung von Deformationsanimationen, evolutionären Übergängen oder von Wachstumsvorgängen verwendet. Sie können – im Gegensatz zu den Simulationsmethoden zur Objektdeformation – auch bei komplexen Organismen verwendet werden. Allerdings ist zu beachten, dass es sich hierbei um rein optische oder morphologische Interpolationsverfahren handelt, welche keinerlei funktionalen Constraints beinhalten. Die Zwischenstadien müssen daher biologisch nicht korrekt sein und sehen daher für den Betrachter oft nicht plausibel aus. Dies fällt vor allem beim evolutionären Übergang zwischen genetisch weit entfernten Spezies auf. Weiterhin ist eine Kombination zwischen Interpolationsmethoden zur Objektdeformation und anderen Animationstechniken nur sehr bedingt möglich.

2.5 Anatomische Simulationsmodelle

Von den reinen Animationstechniken, die für anatomische Strukturen geeignet sind, müssen die *anatomischen Simulationstechniken zur Animation* unterschieden werden, die auf *anatomischen Simulationsmodellen* basieren. Im Folgenden werden dabei diejenigen Modelle näher beleuchtet, die für Organismensimulationen besonders interessant sein könnten.

Die Modelle lassen sich dabei in *Schichtmodelle* und *dynamische Muskelmodelle* aufteilen.

Schichtmodelle bedienen sich einer virtuellen Skeletthierarchie, die mit den in Kapitel 2.3 aufgezeigten Techniken animiert werden. Über hierarchisch ineinander geschachtelten Körperschichten werden die zur Skelettbewegung passenden Deformationen der Oberfläche simuliert. Ein Sonderfall der Schichtmodelle sind die *Skelettmodelle*, da sie sich jeweils lediglich einer einzigen Schicht bedienen.

Bei dynamischen Muskelmodellen geschieht die Animation über die Kontraktion virtueller Muskeln oder analoger Gebilde, die in der Folge den Organismus bewegen. Dynamische Muskelmodelle bedienen sich daher einer *simulativen Animation* und sind enger an biologische Vorbilder angelehnt als Schichtmodelle.

2.5.1 Schichtmodelle

Schichtmodelle nutzen die Tatsache aus, dass höhere Organismen in der Regel aus mehreren konzentrischen Schichten bestehen. Primär beeinflussen die inneren die äußeren Schichten, so dass sich eine Implikationshierarchie ergibt, die *Schicht-Pipeline* genannt wird. Folgende Gewebeschichten können bei Schichtmodellen berücksichtigt werden: [Turner & Thalmann 1993]

1. imaginäre Skelettlinien
2. Knochen
3. Muskeln
4. Fettgewebe
5. Haut
6. Pelz bzw. Haar
7. Kleidung (wird hier im erweiterten Sinne als anatomische Schicht angesehen)

Die einzelnen Schichtmodelle lassen sich danach einteilen, wie viele Schichten sie jeweils berücksichtigen:

- einschichtig (Skelettmodelle)
- zweischichtig (3D-Skinning, Ellipsoidal Sweeping)
- dreischichtig (Dreischichtenmodell von Chadwick et al., Implizites Primitivenmodell von Scheepers et al., Implizites Primitivenmodell von Thalman et al.)
- vierschichtig (Elastic Surface Layer Modell)

Schichtmodelle mit gleicher Anzahl von Schichten unterscheiden sich vor allem dadurch, welche Simulationsmethoden sie für die betreffenden Schichten verwenden.

2.5.1.1 Skelettmodelle

Ein *Skelettmodell*, englisch *skeleton armature*, besteht – im Sinne der Computergraphik – aus einer baumartigen Hierarchie von *Knochen*, den *Bones*, die über *Gelenke*, den *Joints*, miteinander verbunden sind.

Jeder Knochen hat seine eigene Transformationsmatrix, die seine Position in kartesischen Koordinaten angibt.

Meist wird diese Position in dem lokalen Koordinatensystem zu der des in der Hierarchie höher stehenden Knochens angegeben, so dass sich eine Hierarchie von lokalen Koordinatensystemen ergibt. Nur die Position desjenigen Knochens, der an der Wurzel der Hierarchie steht, wird in absoluten Koordinaten angegeben.

Es existieren verschiedene Standardisierungen für Skelettmodelle. Das bekannteste davon ist *H-Anim*, ein humanoides Modell, welches von der *VRML Humanoid Working Group* entwickelt wurde. [Zisheng Huang et al. 2002]

Die Skelettmodelle greifen wiederum auf *anatomiebasierte Gelenkmodelle* bzw. englisch *anatomy based joint models*, wie z.B. *H3D (Hierarchy 3D)*, *APJ (Axis Position Joint)* zurück. [Maciel et al. 2002] [Maciel 2000]

Skelettmodelle werden in der gesichteten Literatur nicht den Schichtmodellen zugeordnet sondern getrennt behandelt. Bei näherer Betrachtung entpuppen sich Skelettmodelle allerdings als Spezialfall der Schichtmodelle, bei der lediglich eine einzige Schicht (Skelettlinie) berücksichtigt ist. Daher ist die hier vorliegende unübliche

Zuordnung der Skelettmodelle zu den Schichtmodellen gerechtfertigt. Trotzdem nehmen die Skelettmodelle eine Sonderstellung unter den Schichtmodellen ein, da alle anderen Methoden auf ein Skelettmodell zurückgreifen, und Skelettmodelle somit auch als Basis für andere Schichtmodelle angesehen werden können.

2.5.1.2 3D-Skinning

3D-Skinning, bzw. *Mesh Skinning*, ist eine gängige Technik, um weiche Deformationen der Oberfläche eines Organismus mit einem Skelettmodell zu erzielen. Dazu wird die Skin-Mesh des Organismus in einer Initialstellung, der *Pose*, benötigt. Empirisch hat es sich als nützlich erwiesen, als Pose einen Organismus mit gerade von sich gestreckten Gliedmaßen aufzunehmen, gleichsam einer Pose, die im Bereich zwischen den zwei im Quadrat und Kreis eingefassten menschlichen Stellungen auf dem berühmten Bild „Die menschlichen Proportionen in einer Rekonstruktion nach Vitruv“ von Leonardo da Vinci liegt.

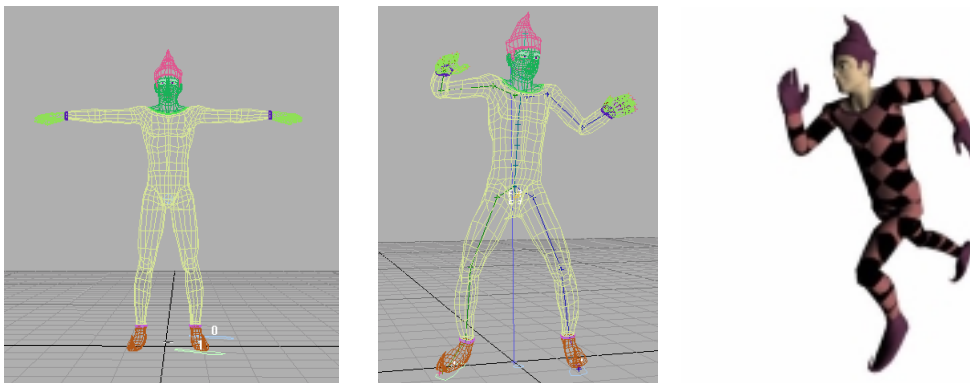


Abbildung 42: Beispiel für Mesh Skinning

Links: Das ursprüngliche Drahtgittermodell eines Narren in „Pose“ mit ausgestreckten Gliedmaßen, Mitte: Dasselbe Drahtgittermodell mit konnektiertem Skelettmodell (blau) [Dominé 2004, 3] Rechts: Das Modell hat den Skinningprozess hinter sich und kann animiert und gerendert werden. [Maughan 2004]

Nach der Erstellung der Pose werden die Vertices des Meshs in einer gewichteten Weise mit den Bones des Skelettmodells verknüpft. Dazu wird jeder Vertex mit n Gewichtungsfaktoren w_1 bis w_n an n verschiedene Bones verbunden, wobei die Summe der Gewichtungsfaktoren gleich 1 sein muss. Die Transformationsmatrizen M_i der verknüpften Bones haben dann gemäß Formel F10 einen Einfluss auf die Positionen \vec{v} der Vertices. [Dominé 2004, 4] [Blender 2004, 64]

$$\mathbf{F10} \quad \vec{v}' = \sum_{i=1}^n w_i M_i \vec{v} \quad , \quad \sum_{i=1}^n w_i = 1$$

In der Regel wird ein Vertex nur mit maximal 2 Bones verknüpft, wobei die Gewichtung w_1 in einer Übergangzone linear von 0 nach 1 zunimmt, während w_2 dementsprechend linear abnimmt. Diese Interpolation wird *Vertex Blending* genannt.

3D-Skinning ist eine performante Methode, bei der die benötigten Operationen unter Annahme einer konstanten Bone-Anzahl gleichmäßig linear mit der Anzahl der Vertices anwachsen. Sie lässt sich daher auch in Echtzeit anwenden.

Diese Vorteile haben dazu geführt, dass 3D-Skinning sowohl bei der Filmindustrie als auch bei der Spieleindustrie weit verbreitet ist. Aufgrund dieser Nachfrage werden von vielen gängigen kommerziellen 3D-Softwareprodukten für CGI, wie z.B. Maya, Rhino, Blender und 3D Studio Max, die Techniken des Mesh Skinings angeboten. [Alias Wavefront 2003] [Avid 2002] [Becker et al. 1999] [Blender 2004] [Kinetix 2003]

Allerdings hat 3D-Skinning auch einige Nachteile. 3D-Skinning orientiert sich nicht bzw. nur sehr ungenau an dem biologischen Vorbild der Hautverformung. Es entstehen dadurch oft unrealistische künstlich wirkende Resultate. Besonders an den Übergangszonen bei Gelenken bemerkt der kritische Betrachter sofort, dass „irgendwas nicht stimmt“. Bei größeren Beugungen und Drehungen können sogar Dellen und Einbuchtungen auf der Haut entstehen. Dieses Phänomen ist unter dem Namen *Syndrom des kollabierenden Ellenbogens* (engl. *collapsing elbow syndrome*) bekannt. (siehe Abbildung 43) [Lewis et al. 2000]

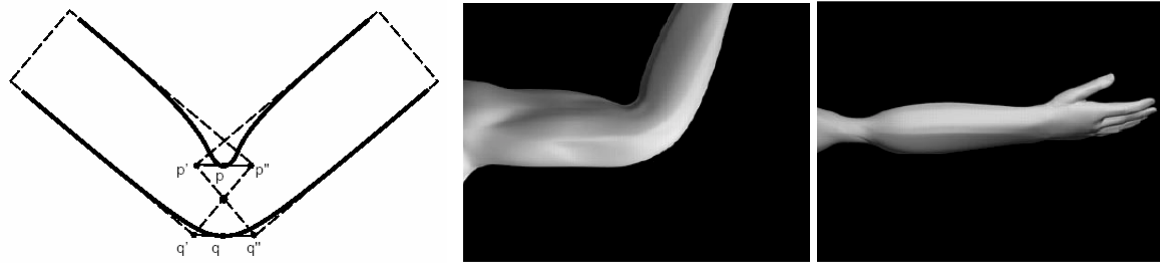


Abbildung 43: Skizze und Beispiele für das Syndrom des kollabierenden Ellenbogens

Linke Skizze: Der Punkt p liegt beim 3D-Skinning mit einer Skelettdeformation auf der Verbindungslinie zwischen p' und p'' , der Punkt q auf der Verbindungslinie zwischen q' und q'' . Bei einer Armbeugung wird zuerst p und dann q in den Arm gleichsam „hineingedrückt“. Dieses Phänomen ist als „Syndrom des kollabierenden Ellenbogens“ (collapsing elbow syndrome) bekannt. Mittleres Bild: Beispiel eines kollabierenden Ellenbogens bei Armbeugung, Rechtes Bild: Beispiel eines kollabierenden Ellenbogens bei einer Armdrehung, wie sie beispielsweise beim Aufschließen einer Tür vorkommt. [Lewis et al. 2000, 166 & 167]

2.5.1.3 Ellipsoidal Sweeping

Eine Technik, welches sich ellipsoider Primitive bedient, ist das *Ellipsoidal Sweeping*. Es beruht wie 3D-Skinning auf einem zweischichtigen Schichtmodell.

Bei dieser Technik werden entlang der Skelettlinien die Zentren von Schlüsselellipsoiden positioniert und so skaliert, dass sie bestmöglich in das Skin-Mesh passen. Wird das Skelett animiert, so verändern sich auch die Positionen und Orientierungen dieser Schlüssel-Ellipsoide. Durch kubische Interpolation der Radien werden bei jedem Frame passende Zwischenellipsoide entlang der Skelettlinie berechnet. Durch rechtwinklig zur Skelettlinie angeordnete Schnittebenen durch die Gesamtheit aller Ellipsoiden wird eine neue Hüllfläche ermittelt. Diese Hüllfläche dient als Vorlage für das Skin-Mesh. [Hyun et al. 2002] [Hyun et al. 2003]

Ellipsoidal Sweeping ist eine einfache Technik mit ausreichender Performanz für interaktive Echtzeitanwendungen. Zusätzlich vermeidet sie das Syndrom des kollabierenden Ellenbogens. (siehe vorheriges Kapitel 2.5.1.2)

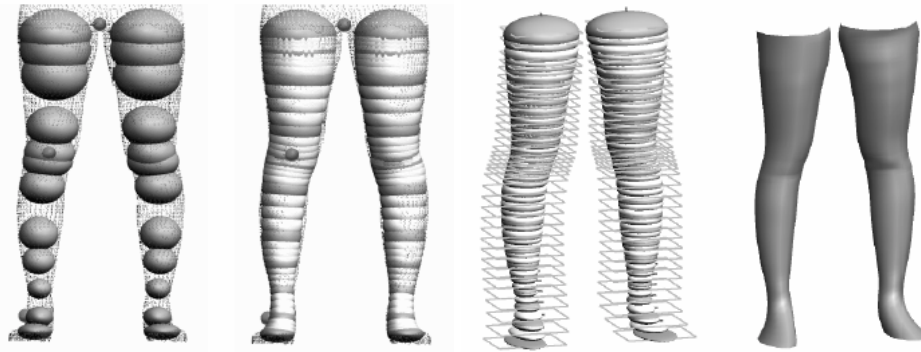


Abbildung 44: Vier Stationen des Ellipsoidal Sweepings

Vier aufeinander folgende Stationen eines Beispiels für die Technik des Ellipsoidal Sweepings. Zu sehen sind v.l.n.r.: 1: Die Anpassung von Schlüsselellipsoiden an das Skin-Mesh in der Pose; 2: Die Generierung von Zwischenellipsoiden durch Interpolation; 3: Die Ermittlung der Hüllfläche durch zur Skelettlinie rechtwinklig angeordnete Schnittebenen durch alle Ellipsoide; 4: Das fertig gerenderte Skin-Mesh. [Hyun et al. 2003]

2.5.1.4 Das Dreischichtenmodell von Chadwick et al.

Das *Dreischichtenmodell von Chadwick et al.* ist ein einfacher Ansatz für Mensch- und Tieranimationen, der auch Deformationen berücksichtigt. Hier wird das Gewebe in die drei Schichten Knochen, Muskeln und Haut aufgeteilt.

Die Muskelschicht wird durch ein grobes Feder-Masse-Dämpfer-System repräsentiert. Manche Enden der Feder-Masse-Einheiten sind dabei mit dem darunter liegenden Skelett verbunden, so dass sich die übrigen Massepunkte sinnfällig mit den darunter liegenden Knochen mitbewegen, wenn es (z.B. mittels Inverser Kinematik o.ä.) animiert wird. Die Massepunkte repräsentieren dabei die Kontrollpunkte für ein Freiform-Deformations-Gitter, in das die Oberflächengeometrie respektive die Hautschicht eingebettet ist. Die beiden Enden des Deformationsgitters stehen stets orthogonal zur jeweiligen Knochenachse, die dazwischen liegenden Bereiche des Gitters verlängern oder verkürzen sich entsprechend der Parallelität der beiden Gitterenden. Auf diese Weise können Muskelwölbungen bei Gelenkbeugungen dargestellt werden.

Das Dreischichtenmodell von Chadwick et al. besticht durch biologisch anmutende Muskelflexionseffekte, die trotz algorithmischer Einfachheit erzielt werden. Allerdings versagt es bei der Visualisierung komplexerer Bewegungen. Hier fällt dem Beobachter sofort die fehlende anatomische Basis auf. Die Bewegungen wirken konstruiert und unnatürlich. [Chadwick et al. 1989] [Turner & Thalmann 1993]

Das Dreischichtenmodell stellt im Grunde genommen keine eigenständige Methode dar, sondern sie ist eine Kombination aus anderen weit verbreiteten Techniken, wie z.B. aus Freiform-Deformations-Gittern (behandelt in Kapitel 2.4.1), aus einem modifizierten Feder-Masse-Modell (behandelt in Kapitel 2.4.2.1) und aus Animationstechniken wie die Inverse Kinematik (behandelt in Kapitel 2.3.3.2) oder Motion Capturing (behandelt in Kapitel 2.3.4).

2.5.1.5 Das implizite Primitivenmodell von Scheepers et al.

Das *implizite Primitivenmodell von Scheepers et al.* bedient sich geometrischer Primitive, um die Form der verschiedenen Gewebeschichten anzunähern. Für Muskeln werden Ellipsoide verwendet, Sehnen werden durch Zylinder repräsentiert, abgeflachte Ellipsoide stehen für Fettgewebe.

Die Muskel-Ellipsoide verändern ihre Radien entsprechend dem Beugewinkel der Gelenke. An Stützpunkten auf der Haut wird die Summe der Radien der darunter liegenden Gewebeteile berechnet. Die Stützpunkte definieren über B-Splines eine parametrische Fläche, welche die Haut repräsentiert. [Scheepers et al. 1996]

Das Modell von Scheepers et al. ermöglicht zwar die Simulation von Organismen mit einer recht detaillierten anatomischen Körperform, so können – im Gegensatz zu den anderen Schichtmodellen – z.B. auch verschiedene übereinander liegende Muskeln berücksichtigt werden, allerdings werden Muskelflexionseffekte nicht berechnet. Da es obendrein in Termini von geometrischen Funktionen arbeitet, ist es auch weder intuitiv handhabbar noch biologisch korrekt. [Scheepers et al. 1997] Darüber hinaus ist es aufgrund seiner mangelnden Performanz nicht für Echtzeitanwendungen geeignet.

2.5.1.6 Das implizite Primitivenmodell von Thalmann et al.

Eine weiteres implizites Primitivenmodell, welches dem von Scheepers propagierten sehr ähnelt, ist das *implizite Primitivenmodell von Thalmann et al.* auch kurz *Bodybuilder-Model* genannt. Als Primitive werden lediglich elliptische Metaballs verwendet, die jeweils an dem am nächsten liegenden Skelettpunkt verankert werden. [Thalmann et al. 1996]

Ein großer Nachteil des impliziten Primitivenmodells von Thalmann et al. ist, das es nicht echtzeitfähig ist. Des weiteren werden elliptische Feldfunktionen anstatt biologischer Termini verwendet werden, so dass sich die Körper nicht korrekt verhalten.

Vor allem an Körperstellen mit übereinander liegenden Muskelpartien fällt die fehlende biologische Stringenz sofort auf, denn hier werden die Radien der Muskeln nicht wie in der Realität addiert, sondern ungefähr das Maximum der beiden verwendet.

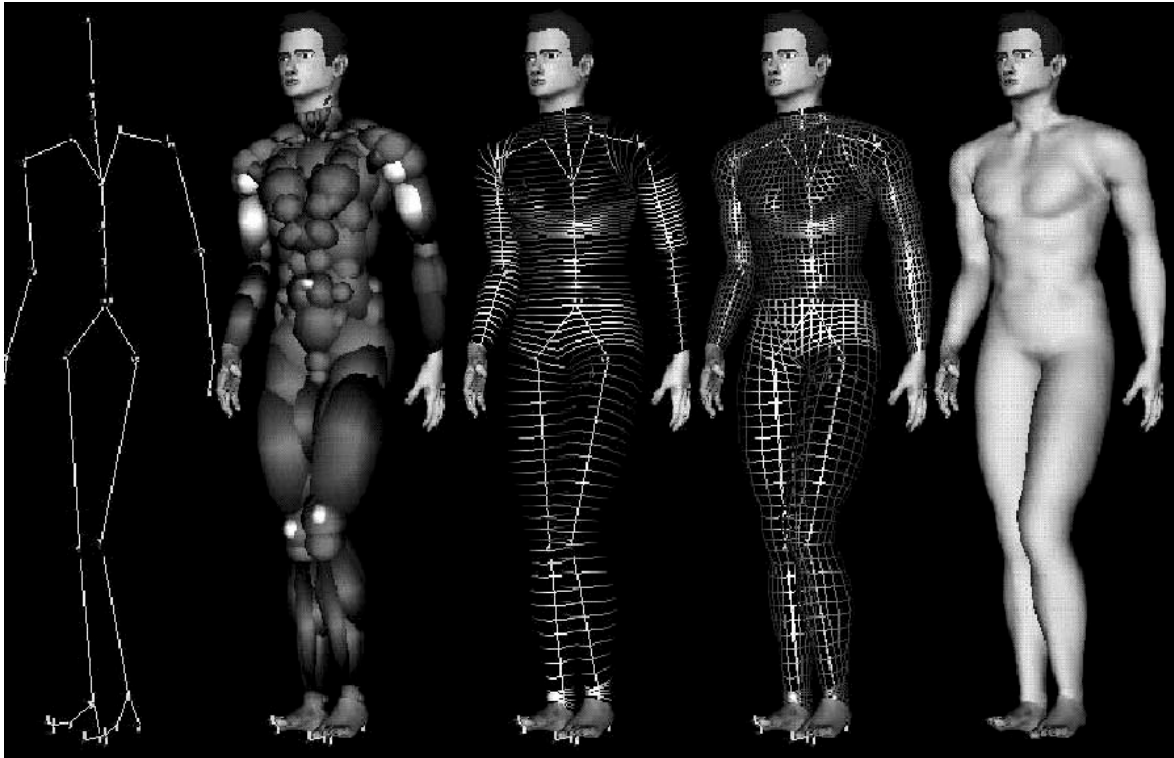


Abbildung 45: Beispiel für das implizite Primitivenmodell von Thalmann et al.

Fünf Stationen des dreischichtigen impliziten Primitivenmodells von Thalmann et al. Zu sehen sind v.l.n.r.:
1. Das Skelett, 2. die Muskelschicht repräsentiert durch elliptische Metaballs, 3. Die Richtungsermittlung für die Texturierung der Implicit Surfaces 4. Das Skin-Mesh als Drahtgittermodell 4. Das gerenderte Skin-Mesh mit geschadeten Polygonen. [Thalmann et al. 1996]

2.5.1.7 Das Elastic Surface Layer Model

Eine Erweiterung des Dreischichtenmodells von Chadwick et al. ist das *Elastic Surface Layer Modell*, auch *Vierschichtenmodell von Turner und Thalmann* genannt, bei dem zusätzlich zur Knochen-, Muskel- und Hautschicht eine Fettschicht eingefügt wurde.

Diese Fettschicht fungiert dabei als Abstandshalter zwischen Haut- und Muskelschicht, Gleichzeitig werden Hooke'sche Federn eingeführt, die als imaginäre Gewebeverbindungen zwischen Haut und Muskelschicht fungieren. Die Haut wird – anders als beim Dreischichtenmodell von Chadwick et al. – als Feder-Masse-Dämpfer-System simuliert, sodass sie beim Elastic Surface Layer Modell über die Fettschicht

hinweggleiten kann. Dies entspricht eher der Realität, als eine Hautschicht, die direkt mit der Muskelschicht verbunden ist. [Turner & Thalmann 1993]

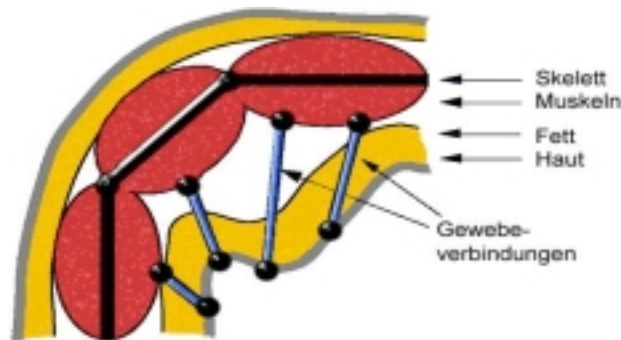


Abbildung 46: Skizze des Vierschichtenmodells

Skizze des Elastic Surface Layer Models von Turner und Thalmann. Zu sehen sind die vier Schichten: 1. Skelettschicht (schwarz) 2. Muskelschicht (rot) 3. Fettschicht (gelb) 4. Hautschicht (grau) sowie die Gewebeverbindungen zwischen Muskeln und Hautschicht (blau mit schwarzen Verbindungspunkten). [Turner & Thalmann 1993]

2.5.1.8 Bewertung der Schichtmodelle

Da Schichtmodelle auf Skeletthierarchien beruhen, sind diese Verfahren ausgezeichnet mit Animationstechniken kombinierbar, die ebenfalls eine Skeletthierarchie verwenden, wie beispielsweise Motion Capturing und Keyframing.

Da sie kein dynamisches Vertexmodell benötigen, sondern von einer Pose ausgehen, lassen sich auch fast alle Modellierungstechniken mit ihr verbinden. Aus diesen pragmatischen Gründen ist es kein Wunder, dass Schichtmodelle eine weite Verbreitung erfahren haben.

Allerdings werden biologisch-anatomische Prozesse, wie z.B. Deformationen bei Muskelflexionen, entweder gar nicht oder nur ungenau simuliert. Die Modelle sind nach biomechanischen Gesichtspunkten allesamt inkorrekt. Komplizierte Deformationen werden durch technozentrische Ansätze nachgebildet. Dies führt dazu, dass die Organismen sich für den kritischen Beobachter unrealistisch bewegen.

2.5.2 Dynamische Muskelmodelle

Dynamische Muskelmodelle sind recht nahe an biologischen Vorbildern angelehnt, denn sie berechnen explizit die Auswirkungen von virtuellen Muskeln bzw. analogen Strukturen auf die Bewegung anatomischer Strukturen. Techniken, die auf dynamischen Muskelmodellen beruhen, werden auch *simulative Organismenanimationen* genannt, da die Erzeugung von flüssigen Bewegungen über eine Simulation des Bewegungsapparates erfolgt. Sie stellen daher eine Chimäre zwischen Animations- und Simulationstechniken dar. Im Folgenden werden exemplarisch zwei sehr unterschiedliche Ansätze vorgestellt:

Das *dynamische Muskelmodell von Terzopoulos* abstrahiert die Anatomie der Organismen – in diesem Falle Fische – sehr stark, so dass die Fische nur noch wenig mit ihren Vorbildern gemein haben. Trotzdem bewegen sich die Fische sinnfällig im Wasser. Neben der anatomischen Bewegungssimulation erfolgt auch eine rudimentäre Verhaltenssimulation.

Dagegen lehnt sich das *Modell von Nedel und Thalmann* sehr stark an biologische Vorbilder an. Es beruht vorwiegend auf einem Feder-Masse-Modell, dessen Topologie speziell an die Erfordernisse der Muskelsimulation adaptiert wurde.

2.5.2.1 Dynamisches Muskelmodell von Terzopoulos

Terzopoulos et al. simulieren die Bewegung von Fischen durch ein einfaches Fischmodell, welches auf 23 Massepunkte und 91 viskoelastische Elementen reduziert ist, wobei 12 der viskoelastischen Elemente aus kontraktile Muskelementen bestehen. Die mechanischen Komponenten sind dergestalt miteinander verbunden, dass sie ein flexibles strukturelles Innengerüst eines Fisches erzeugen, welches durch Muskelkontraktionen gebogen werden kann. (siehe Abbildung 47)

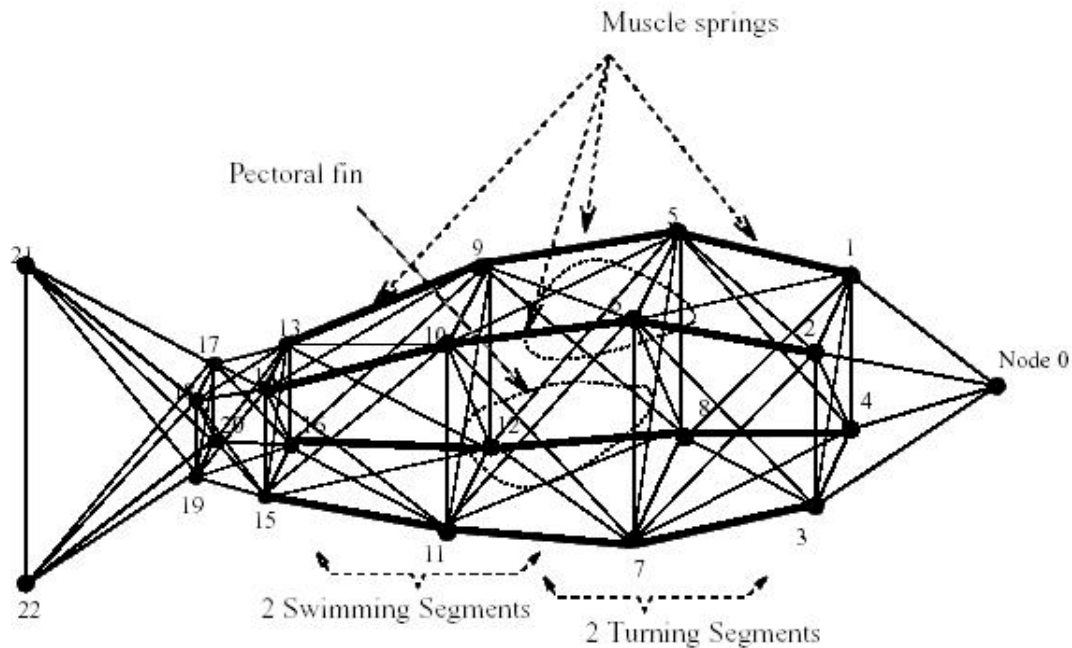


Abbildung 47: Skizze des dynamischen Fischmodells von Terzopoulos

Skizze des dynamisches Fischmodells aus 12 Muskelursprüngen (dicke Linien), 91 viskoelastischen Elementen (dünne Linien) und 23 Massepunkten (Knotenpunkte 0 bis 22) [Terzopoulos et al. 1994] [Terzopoulos and Tu 1997] [Lobb 2002]

Die 12 kontraktile Elemente werden nun durch einen Fließkommawert zwischen 0 und 1 angesteuert, 0 bedeutet dabei die minimale Kontraktionslänge (l_{ij}^{\min}) und 1 die maximale Kontraktionslänge (l_{ij}^{\max}), zwischen diesen beiden Extrema wird linear interpoliert.

Ein sich ständig ändernder Muskel-Aktions-Funktionsvektor mit 6 Elementen spezifiziert den Kontraktionsstatus des gesamten Fisches, da jeweils diejenigen zwei kontraktile Elemente, welche spiegelsymmetrisch zur vertikalen Mittelebene stehen, antagonistisch mit einem Vektorelement angesteuert werden, d.h. während die Länge des linken kontraktile Elementes durch das Vektorelement l angesteuert wird, errechnet sich die Länge des rechten kontraktile Elementes durch $r = 1-l$.

Die Ansteuerung der kontraktile Elemente erfolgt bei der normalen Vorwärtsbewegung sinusförmig, wobei die kontraktile Elemente, welche zueinander antagonistisch stehen, jeweils um 180 Grad phasenverschoben angesteuert werden. Die Bewegung der Schwanzflosse erfolgt indirekt durch die Orientierungsänderung der Schwimmsegmentsursprünge. Die gesamte Positions- und Orientierungsänderung des

Fisches wird nicht physikalisch – wie etwa durch Strömungsberechnungen – ermittelt, sondern erfolgt durch Schätzung.

Eine einfache Implikationslogik ahmt die Verhaltensreaktionen des Fisches auf Objekte nach, die ins radial begrenzte, 300 Grad weite Sichtfeld des Fisches gelangen. (siehe Abbildung 48) Diese Verhaltensreaktionen steuern vordefinierte Bewegungspatterns für den Muskel-Aktions-Funktionsvektor. [Terzopoulos et. Al 1994] [Terzopoulos and Tu 1997] [Lebouthillier 2000]

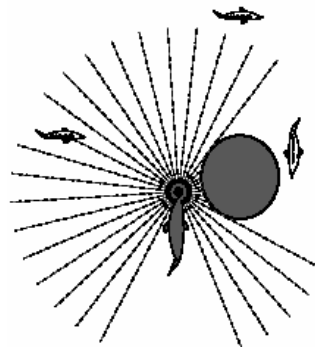


Abbildung 48: Skizze der Wahrnehmung des Fischmodells von Terzopoulos

Skizzenartige Darstellung der sensorischen Wahrnehmung des Fischmodells von Terzopoulos: Nur der linke Fisch wird wahrgenommen, der rechte Fisch ist dagegen durch ein Objekt verdeckt, der vordere Fisch befindet sich außerhalb des vordefinierten Sichtradius. [Terzopoulos and Rabie 1997]

2.5.2.2 Das Modell von Miller

Miller verwendet – ähnlich wie Terzopoulos – ein stark reduziertes Muskelmodell zur Animation von Würmern und Schlangen. Der Wurm bzw. die Schlange wird in Segmente unterteilt. Ähnlich wie bei Terzopoulos werden symmetrische Paare von viskoelastischen Elementen zur Animation verwendet. Zusätzlich berücksichtigt Miller einfache physikalische Regeln bei der Bewegungssimulation, wie z.B. die Haft- und Gleitreibung zwischen der Haut des Organismus und dem Boden. [Miller 1988]

Das System von Miller ist allerdings lediglich für bestimmte Tiergruppen wie Schlangen und bestimmte Würmer geeignet. Hier vermag das System von Miller realistische Bewegungsabläufe zu liefern. Peristaltische Bewegungen sind mit dem System von Miller nicht möglich, so dass nur Würmer mit sinoiden Bewegungen in Frage kommen. Eine Übertragungsmöglichkeit auf andere Tierarten ist nicht möglich.

2.5.2.3 Das Modell von Nedel und Thalmann

Das *Modell von Nedel und Thalmann* stellt ein spezielles Feder-Masse-Dämpfer-System dar, dessen Topologie speziell für die Simulation von Muskeln konstruiert ist. Dafür teilen Nedel und Thalmann den Muskel in zwei Teilbereiche ein, erstens die *Action Lines*, welche imaginäre Muskelmittellinien darstellen, und zweitens die *Muskelhülle*, die aus einem Feder-Masse-Dämpfer-System besteht und radial um die Action Lines aufgebaut wird. Action Lines sind dabei ein sehr reduziertes Teilmodell eines fusiformen Muskels. Es werden nur imaginäre kontraktile Muskelmittellinien inklusive ihrer Ursprungs- und Ansatzpunkte berücksichtigt. Für gewöhnlich wird ein einfacher fusiformer Muskel genau durch eine Action Line repräsentiert, komplexe Muskeln oder Muskeln mit mehreren Ursprungs- oder Ansatzpunkten dagegen durch ein ganzes Bündel von Action Lines. Zwischen dem Ursprungs- und Ansatzpunkt kann eine Action Line optional noch über weitere Führungspunkte, den *Guide Points*, verlaufen, welche einen komplexen Linienzug ermöglichen und somit auch die Simulation von Muskeln ermöglichen, die sich um einen Knochen herumwinden.

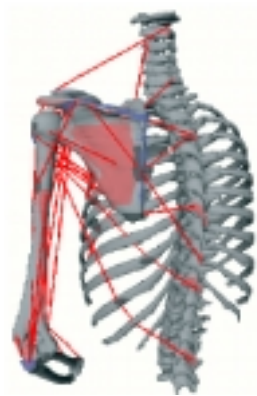


Abbildung 49: Verlauf verschiedener Action Lines

Beispiel für den Verlauf verschiedener Action Lines der Oberarm- und Rückenmuskulatur. [Maurel 1998, 93]

Radial um diese Action Lines werden Partikel positioniert, wobei jeder Partikel jeweils über ein Feder-Masse-Dämpfer-System mit seinen direkten Nachbarn verbunden wird. Da bei jedem radialen Positionierungsschritt immer gleich viele Partikel gesetzt werden, existieren für jeden Massepunkt im Inneren des Feder-Masse-Dämpfer-Systems jeweils zwei vertikale und zwei horizontale Nachbarpartikel (siehe Abbildung 50).

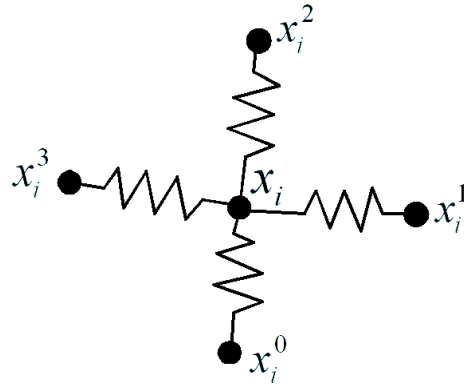


Abbildung 50: Schema eines Partikels mit vier Nachbarpartikeln

Schema eines Partikels x_i , der mit seinen vier direkten Nachbarpartikeln x_j^j mit $0 \leq j < 4$ über Federn und (aus Gründen der Übersichtlichkeit nicht eingezeichneten) parallel verlaufenden Dämpfungselementen verbunden ist.

Somit entstehen die Vertices einer Muskelhülle. Die Form dieser Muskelhülle kann durch radiale Justierung der Massepunkte an die Form echter Muskeln adaptiert werden, die durch CT-Scans ermittelt wurden (siehe Abbildung 51).



Abbildung 51: Adaptierte Hüllenform von Oberarmmuskeln

Gerenderte Hüllenform diverser Oberarmmuskeln, die durch radiale Justierung der Massepunkte an die Form echter Muskeln erzielt wurde [Maurel 1998, 106]

Die Formveränderung der Muskelhülle entsteht durch Krafteinwirkung auf alle Partikel. Dabei werden die *Elastizitätskraft* ($f_{elasticity}$), *Biegungskraft* ($f_{curvature}$) und *geometrische Beschränkungen* (f_{constr}) berücksichtigt [Thalman & Nedel 1998]. Für jeden Partikel x_i kann die resultierende Kraft dann wie folgt berechnet werden:

$$\mathbf{F11} \quad f_{result}(\vec{x}_i) = f_{elasticity}(\vec{x}_i) + f_{curvature}(\vec{x}_i) + f_{constr}(\vec{x}_i)$$

Die Kraft, die durch eine einzelne Feder auf ein Partikel ausgeübt wird, lässt sich dabei durch die Hooke'sche Formel berechnen, wobei k die Federkonstante, \vec{x}_i die aktuelle Auslenkungsposition und \vec{x}_{rest} die Ruheposition des Partikels ist:

$$\mathbf{F12} \quad f_{spring}(\vec{x}_i) = -k(\vec{x}_i - \vec{x}_{rest})$$

Da ein Partikel im Muskelmodell von Nedel und Thalmann immer mit vier Nachbarn verbunden ist, wird die gesamte Elastizitätskraft $f_{elasticity}$ über die entsprechende Summe aller auf diese Weise einwirkenden Kräfte ermittelt:

$$\mathbf{F13} \quad f_{elasticity}(\vec{x}_i, \vec{x}_i^0, \vec{x}_i^1, \vec{x}_i^2, \vec{x}_i^3) = \sum_{j=0}^3 f_{spring}(\vec{x}_i)$$

wobei \vec{x}_i für die Position des Partikels i und \vec{x}_i^j mit $0 \leq j < 4$ für die Position eines der vier Nachbarpartikel steht. [Nedel 1998]

Beim Muskelmodell von Nedel und Thalmann werden zwei verschiedene Federtypen verwendet, die sich lediglich im Wert ihrer Federkonstante unterscheiden, *longitudinale Federn* in Richtung der Action Line mit einer niedrigen Federkonstante und *orthogonale Federn* quer zur Action Line mit einer hohen Federkonstante. Die Kraft, welche die willkürliche Biegung der Oberfläche begrenzt und somit für eine Minimierung der Oberfläche sorgt, wird als Biegunskraft $f_{curvature}$ bezeichnet. Um die Biegunskraft zu berechnen, werden weitere Hooke'sche Federtypen, die *Angular Springs* eingeführt. (siehe Abbildung 52). Im Gegensatz zu den bereits vorhandenen Federn stehen sie rechtwinklig zu den zwei imaginären Verbindungslinien $(\vec{x}_i^0, \vec{x}_i^2)$ und $(\vec{x}_i^1, \vec{x}_i^3)$, die zwischen den jeweils entgegen gesetzten Nachbarpartikeln aufgespannt werden (siehe Abbildung 52).

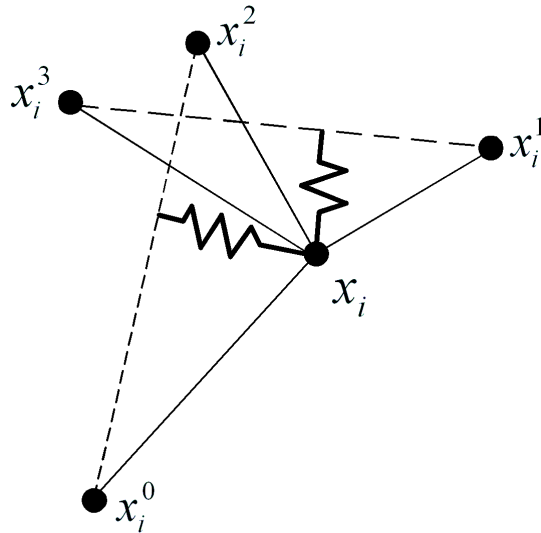


Abbildung 52: Skizze der Positionen der Angular Springs

Skizze der Positionen der zwei Angular Springs eines Partikels x_i , die rechtwinklig zu den zwei imaginären Verbindungsgeraden $(\vec{x}_i^0, \vec{x}_i^1)$ und $(\vec{x}_i^1, \vec{x}_i^2)$ stehen

Die geometrischen Beschränkungen f_{constr} sind Kräfte, die nicht im internen Kräftemodell des Muskels entstehen, sondern durch externe, die Ausdehnung des Muskels begrenzende Faktoren und Objekte. Dies kann beispielsweise ein Knochen sein, auf dem der Muskel aufliegt und der seine Form modifiziert. Die Kraftgröße wird dabei durch inverse Dynamik berechnet.

Die Bewegungssimulation erfolgt durch Anwendung der Lagrange'schen Bewegungsgleichung auf die einzelnen Partikel des Feder-Masse-Dämpfer-Systems, so dass eine Differenzialgleichung zweiter Ordnung entsteht, in der m_i die Masse, \vec{a}_i die Beschleunigung, γ_i die Dämpfungskonstante, \vec{v}_i die Geschwindigkeit, $f_{result}(\vec{x}_i, \vec{x}_i^0, \vec{x}_i^1, \vec{x}_i^2, \vec{x}_i^3)$ die resultierende Kraft aus Gleichung F14 und f_{extern_i} die extern einwirkende Kraft auf den Massepunkt i ist:

$$\mathbf{F14} \quad m_i \vec{a}_i + \gamma_i \vec{v}_i + f_{result}(\vec{x}_i, \vec{x}_i^0, \vec{x}_i^1, \vec{x}_i^2, \vec{x}_i^3) = f_{extern_i}$$

Die Simulation des Feder-Masse-Dämpfer-Systems wird durch Integration über die Zeit dieser Bewegungsgleichung gemäß der Runge-Kutta-Methode vierter Ordnung erzielt. [Nedel & Thalmann 1999] [Mendoza et al. 2002]

2.5.2.4 Bewertung der dynamischen Muskelmodelle

Dynamische Muskelmodelle sind weitaus enger an biologische Vorbilder angelehnt als Schichtmodelle. Insbesondere mit dem Modell von Nedel und Thalmann können Form und Bewegung anatomischer Strukturen recht genau berechnet werden. Trotzdem werden wesentliche biologische Regeln wie beispielsweise die Volumenkonstanz ignoriert.

Nachteilig bei diesen komplexeren dynamischen Muskelmodellen ist die Tatsache, dass sie für die Berechnung großer anatomischer Strukturen weder performant noch stabil genug sind. Somit werden nur sehr kleine anatomische Einheiten simuliert. Interaktive Echtzeitsimulationen von Mehrmuskelsystemen oder anderer größerer Strukturen sind bislang unmöglich.

3 Defizit- und Anforderungsanalyse

*Unseren Gelehrten fällt sogar,
wunderlicherweise,
die allernächste Frage nicht ein:
wozu ihre Arbeit, ihre Hast,
ihr schmerzlicher Taumel nütze sei.*
Friedrich Wilhelm Nietzsche 1873

Als Fazit der State-of-the-Art-Analyse aus Kapitel 2 (eingeleitet durch Kapitel 1.1) lässt sich konstatieren, dass trotz vieler unterschiedlicher Ansätze die Behandlung dreidimensionaler virtueller Organismen bisher nicht zufrieden stellend gelöst worden ist. Neben kleineren, konkreten Defiziten zu einzelnen Verfahren (siehe die entsprechenden Schlussbewertungen in Kapitel 1.1 und 2), haben sich auch allgemeine Problembereiche herauskristallisiert, welche alle Verfahren betreffen. Diese Problembereiche sind so gravierend, dass eine neue Methode zur Organismengenerierung völlig andere Wege beschreiten muss. In den folgenden Unterkapiteln sollen diese Problembereiche nacheinander beleuchtet werden und daraus die entsprechenden Anforderungen an ein neues Konzept erarbeitet werden.

Allgemeine konzeptuelle Defizite und Anforderungen bezüglich des zugrunde liegenden Paradigmas werden dabei in den Unterkapiteln 3.1 bis 3.4 behandelt. Aus dieser grundsätzlichen Anforderungsskizze kann das erwünschte konkrete Fähigkeitsspektrum aufbauen. Die diesbezüglichen konkreten Anforderungen an bestimmte Fähigkeiten, welche das Konzept beinhalten soll, werden in Unterkapitel 3.5 bis 3.11 dargestellt.

3.1 Holistischer Generierungsprozess

Die betrachteten Generierungsmethoden für Organismen sind – sofern sie realistische Resultate zeitigen – bislang in die drei sequentiellen Phasen Modellierung, Animation und Rendering eingeteilt. Bei der Modellierung wird eine Hüllform erzeugt. Dieser Form wird dann im Rahmen eines Animationsprozesses eine Bewegung aufgezwungen. Zuletzt werden die Bilder durch den Renderingprozess erzeugt. (siehe Abbildung 53)

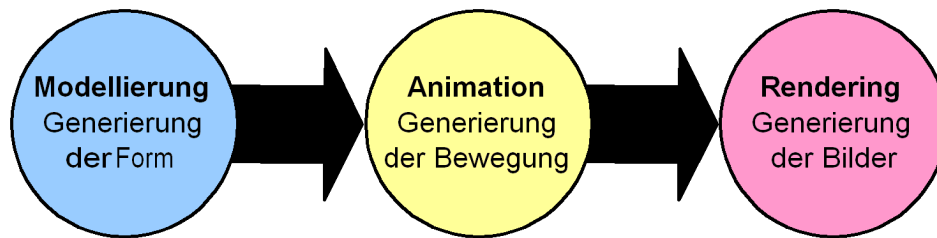


Abbildung 53: Skizze des konventionellen Generierungsprozesses für Organismen

Bei konventionellen Generierungsmethoden von virtuellen 3D-Organismen wird in der Regel erst durch die Modellierung die Form kreiert, dann wird dieser Form eine Bewegung aufgezwungen, zuletzt wird durch das Rendering eine Bildsequenz erzeugt. Somit existiert ein sequentieller dreigeteilter Generierungsprozess.

Wird dieser dreigeteilte Erstellungsprozess mit demjenigen Prozess verglichen, in dem die Natur ihre Geschöpfe erzeugt, so fällt auf, dass er dort keine Entsprechung findet und umständlich ist, denn die Natur kennt keine sequentielle Trennung zwischen Form, Bewegung und Optik. Es ist ja keineswegs so, dass beim Wachstum eines Organismus zuerst die Form und dann die Bewegung erzeugt werden und am Schluss der Organismus ausgewachsen optisch erscheint.

Stattdessen wächst in der Natur ein Organismus. Im Rahmen dieses Wachstumsprozesses verändert sich seine Form und die Form steckt wiederum die Rahmenbedingungen für das Bewegungsspektrum ab. Umgekehrt verändern die Bewegungen von Organismen direkt wieder die aktuelle Form der Organismen. Es besteht also eine Interdependenz zwischen Form und Bewegung. Der Organismus ist während des ganzen Wachstumsvorganges sichtbar, somit ist also auch die Optik an dieses Form-Bewegungs-Kontinuum gekoppelt.

Die Natur verwendet also statt eines umständlich-sequenziellen einen einfach-parallelen Generierungsprozess, der Modellierung (Form), Animation (Bewegung) und Rendering (Bilder) holistisch integriert.

Bei den analysierten Verfahren existiert dagegen durch die sequentielle Behandlung von Modellierung, Animation und Rendering eine künstliche Trennung zwischen Form-Bewegungs- und Bilderzeugung, die einem langwierigen Generierungsprozess zur Folge hat, der darüber hinaus infolge der fehlenden Interdependenz zwischen Form und Bewegung auch meist zu nicht realistischen Resultaten führt.

Die Trennung dieser drei Bereiche, welche ja lediglich technisch oder durch Traditionen bedingt ist, sollte daher zumindest für die Organismengenerierung hinterfragt werden. Für die Generierung virtueller Organismen könnte hier ein Paradigmenwechsel

angebracht sein, indem die künstliche Trennung von Modellierung, Animation und Rendering aufgehoben wird und durch einen holistischen Gesamtprozess ersetzt wird, der sich an natürlichen Wachstumsvorgängen orientiert. (siehe Abbildung 54) In diesem Gesamtprozess sollte die Formgenerierung und die Bewegungsgenerierung interdependent miteinander fusioniert werden und parallel dazu eine Echtzeitvisualisierung erfolgen.

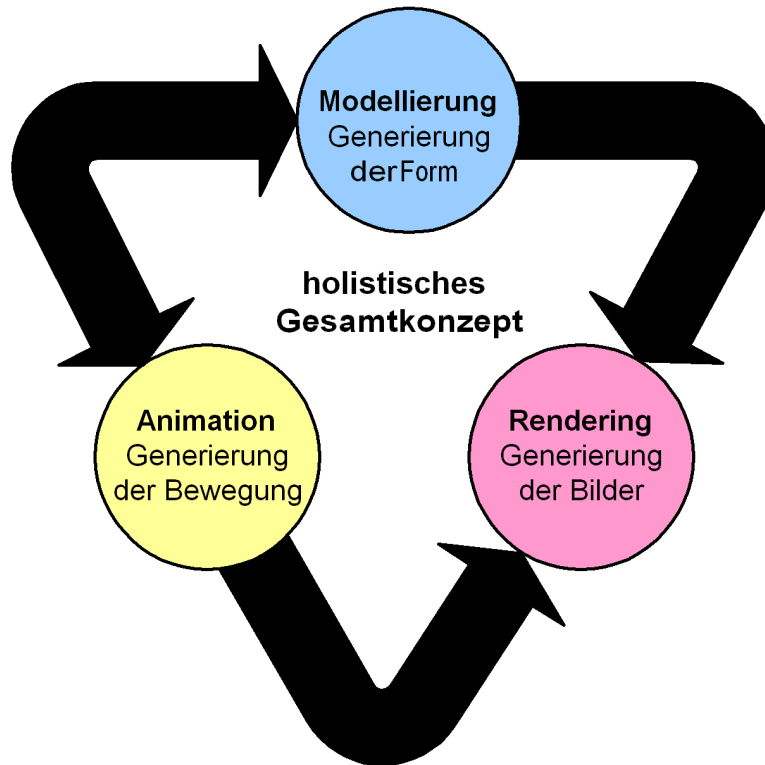


Abbildung 54: Skizze des holistischen Generierungskonzepts von Organismen

Analog zu natürlichen Wachstumsprozessen, bei denen Form, Bewegung und Aussehen jeweils langsam und interdependent gleichzeitig erzeugt werden, sollte auch der Modellierungs-, der Animations- und der Renderingprozess parallelisiert werden. Diese Parallelisierung in Zusammenhang mit der Interdependenz zwischen Modellierung und Animation impliziert die Anforderung an ein völlig neues holistisches Gesamtkonzept.

Ein solches holistisches Gesamtkonzept verspricht, die Modellierung und Animation zu beschleunigen, realistischere Resultate zu liefern und implizit eine Animation von Wachstumsprozessen zu liefern. Sie setzt jedoch einen radikalen Paradigmenwechsel voraus, bei der die Begriffe Modellierung, Animation und Visualisierung durch ihre gegenseitige Verzahnung nicht mehr ohne weiteres voneinander getrennt werden können und einen kompletten Bedeutungswandel erfahren.

3.2 Verwendung intuitiver Metaphern

Die in der State-of-the-Art-Analyse betrachteten Modellierungs- und Animationsansätze für Organismen verwenden – wenn überhaupt – keine überzeugenden Metaphern aus dem Alltag. Eine Ausnahme hiervon stellt das Blobby Modelling dar, welches die einfache Alltagsmetapher von „Lehmklumpen“ als Modellierungsmetapher verwendet. Blobby Modelling hat jedoch andere Defizite, die in Kapitel 2.1.7 besprochen wurden.

Infolge der fehlenden Alltagsmetaphern sind die Methoden für ungeschulte Personen ungeeignet. Sie benötigen vielmehr Fachkräfte, die in den entsprechenden Teilgebieten der Graphischen Datenverarbeitung speziell ausgebildet wurden. Die Generierungsansätze können daher nicht als intuitiv bezeichnet werden.

Der Einsatz hoch qualifizierter Fachkräfte impliziert in der Folge auch hohe Herstellungskosten. Diese Kosten werden zusätzlich durch den horrenden Arbeitsaufwand erhöht, der notwendig ist, um überzeugende Organismen zu generieren. Neben dem in Kapitel 3.1 kritisierten sequentiell geteilten Generierungsprozess führt das Fehlen einer direkten interaktiven Rückkopplung der quantitativen Modellierungs- und Animationsanweisungen zu einem langwierigen Trial-and-Error-Verfahren. Somit sind die betrachteten Verfahren auch nicht effektiv bezüglich ihres Zeit- und Arbeitsaufwandes.

Aus diesem Grund wäre ein Verfahren wünschenswert, welches wirklich einfache Metaphern aus dem Alltag verwendet. Damit eine Metapher wirklich intuitiv und effektiv ist, muss sie folgende Kriterien erfüllen:

1. Die Metapher sollte allen Zielpersonen bekannt sein.
2. Die Zielpersonen sollten schon in einem anderen Kontext mit der Metapher in Berührung gekommen sein und vorzugsweise mit ihnen selbst Organismen erzeugt haben.
3. Die Metapher sollte einfach zu bedienen sein. Ein eindeutiger Garant für die Einfachheit wäre, dass sie unter anderem von Kindern verwendet werden könnte.
4. Die Metapher sollte eine direkte quantitative Rückkopplung der Modellierungsparameter erlauben.
5. Die Metapher sollte kombinierbar sein, so dass sie eine hohe Komplexität der Organismen erlaubt.

6. Die Metapher sollte in die verwendete Methode integriert werden, sie sollte also nicht nur zum Modellieren verwendet werden, sondern gleichzeitig Basis des Modells sein.
7. Die Metapher sollte mit biologischen Constraints kompatibel sein, um auch biologisch stringente Organismen zu erzeugen.
8. Die Metapher sollte nicht nur für den Modellierungsprozess, sondern auch für die Animation der Organismen verwendet werden können.

Um eine Metapher zu finden, welche alle obigen Kriterien erfüllt, wurde eine Brainstorming-Sitzung mit Kindern (siehe Kriterium 3) durchgeführt. Gleichzeitig wurde sich bei Spielen, auf Kinderfesten und in Kindergärten nach geeigneten Metaphern umgeschaut. Danach wurde mit Hilfe der Kriterienliste eine Ausschlussselektion durchgeführt. Letzten Endes blieb die Metapher der so genannten „Luftballonmännchen“ (siehe Abbildung 55) übrig.

Luftballonmännchen sollten allen Menschen in Industriestaaten bekannt sein (Kriterium 1). Die meisten Menschen sind schon mit ihnen in Berührung gekommen (Kriterium 2), meist in Zusammenhang mit Festen oder Marketingevents für Kinder (Kriterium 3). Luftballons erlauben eine haptische Rückkopplung bei ihrer Verformung. Ein direktes, quantitatives Feedback der Modellierungsparameter ist somit gewährleistet (Kriterium 4). Die Luftballons können dabei zusammengebunden oder miteinander verdrillt werden, so dass auch komplexere Gebilde erzeugbar sind (Kriterium 5). Eine Integration der Metapher in die Methode sollte machbar sein, indem eine Ballonsimulation mit einem Geometriemodell kombiniert wird. Da Ballons außer Luft nur aus einer Gummihülle bestehen, eignen sich für letzteres geschlossen-konvexe polygonale Geometriemodelle (Kriterium 6). Indem die ursprüngliche Metapher der Luftballonmännchen leicht abgewandelt wird und wassergefüllte Ballons verwendet werden, ist das Verhalten der Ballons weitgehend kompatibel mit den hydropneumatischen Regeln der Biomechanik (Kriterium 7). Indem das ursprüngliche Konzept durch kontraktile und versteifte Ballons ergänzt wird, können auch biologisch stringente Animationen erzeugt werden (Kriterium 8). Somit erfüllt die Metapher alle acht gestellten Kriterien.

Allerdings muss die einfache Metapher der Luftballonmännchen erweitert werden, um auch realistische Organismen zu erzeugen und vielfältige Animationen zu erlauben. Einige dieser Erweiterungen, wie kontraktile Ballons, um Muskeln zu simulieren, und versteifte

Pneus zur Generierung von Knochen und Knorpel sind schon genannt worden. Weitere Erweiterungen sind:

- Die Ballons sollten in der Lage sein, wie anatomische Strukturen zu wachsen.
- Die Ballons sollten nicht platzen, egal wie sie gequetscht, gestaucht oder anderweitig deformiert werden.
- Die Ballons sollten texturierbar sein.
- Die physikalischen Parameter des Ballons sollten frei modifizierbar sein.
- Es sollten auch Ballons mit gemischten physikalischen Parametern generierbar sein, wobei die physikalischen Parameter von der Oberflächenposition abhängen.



Abbildung 55: Beispiel für ein „Luftballonmännchen“

Als Metapher zur Modellierung von Organismen könnten „Luftballonmännchen“ dienen. [Balloonhq 2004]

3.3 Abkehr vom Aufzeichnungscharakter

Insbesondere diejenigen Methoden und Technologien, welche zu realistisch anmutenden Resultaten führen, wie beispielsweise 3D-Scanning (siehe Kapitel 2.2.2), Motion Capturing (siehe Kapitel 2.3.4) oder Stop Motion Input Devices (siehe Kapitel 2.3.5) haben einen Aufzeichnungscharakter, d.h. der Modellierungs- oder Animationsprozess findet größtenteils außerhalb des Computers statt. Es werden lediglich die Form, die Farbgebung oder die Bewegungen real existierender Organismen bzw. Maquetten aufgezeichnet und auf virtuelle Organismen übertragen. Organismen, die mit diesen Techniken generiert wurden, können daher im Grunde genommen nicht als „computergeneriert“, sondern allenfalls als „vom Computer aufgezeichnet“ bezeichnet werden (siehe dazu auch Kapitel 1.1.2).

Weiterentwicklungen dieser Technologien beziehen sich daher immer nur auf die Verbesserung der jeweiligen Aufzeichnung, jedoch nicht auf den eigentlichen Modellierungs- oder Animationsprozess. Solche Aufzeichnungsmethoden stellen daher langfristig eine Sackgasse innerhalb der Graphischen Datenverarbeitung dar.

Diese technologische Sackgasse ist nicht nur theoretischer Natur: Da die Generierung der Organismen größtenteils außerhalb des Computers stattfindet, kann der Computer logischerweise nur unzureichend eine direkte, unmittelbare Hilfestellung leisten. Dies impliziert in der Praxis einen langwierigen und in der Folge auch kostspieligen Generierungsprozess (siehe Kapitel 1.1.3).

Darüber hinaus wird infolge dieses Aufzeichnungscharakters das Spektrum möglicher Organismen für die direkte Modellierung und Animation erheblich eingeschränkt. Zu den Organismen, die sich in der Regel der direkten Aufzeichnung entziehen, zählen Fabelwesen und andere fiktive Organismen, besonders kleine Organismen (Bakterien, Insekten, Spinnen, Würmer, etc.), extrem große Organismen (z.B. Wale) sowie ausgestorbene Tierarten.

Hier kann höchstens eine indirekte Modellierung oder Animation erfolgen, indem ein Ersatzorganismus herangezogen wird. Ein bekanntes Beispiel für eine indirekte Animation mit Hilfe eines Ersatzorganismus ist die Verwendung einer Straußenherde für die Animation der Gallimimus-Herde in Jurassic Parc 1 (siehe Abbildung 13 auf Seite 21). [Uip 2004] Es ist dabei jedoch nicht immer gewährleistet, dass die Bewegungen des

Ersatzorganismus auf dem virtuellen Zielorganismus überzeugend anmuten. (siehe Kapitel 1.1.4)

Es wäre also angebracht, sich von jenen Methoden abzuwenden, die von einem Aufzeichnungscharakter gekennzeichnet sind, um nach alternativen Konzepten zu suchen, die auf echten computergesteuerten Modellierungsansätzen beruhen, die interdependent mit simulativen Animationsansätzen zusammenarbeiten.

Dabei sollte nicht im Vordergrund stehen, dass diese neuen Konzepte realistisch anmutende Resultate hinsichtlich Form und Bewegung zeitigen, sondern es ist ausreichend, dass neue Methoden verwendet werden, die einen Weg aus der beschriebenen technologischen Sackgasse weisen, so dass zukünftige Weiterentwicklungen von Organismengenerierungsmethoden per Computer überhaupt die Chance bekommen, auch ohne Aufzeichnungscharakter langfristig zur realistischen Resultaten zu kommen.

3.4 Verwendung evolutionärer Methoden

Bei den in der State-of-the-Art-Analyse betrachteten Modellierungs- und Animationsansätzen für Organismen existiert kein einziges Modell, welches es erlaubt, evolutionäre Übergänge auf direkte Weise überzeugend zu visualisieren. Zwar werden verschiedene Überblendungs- und Morphingtechniken (siehe Kapitel 2.4.3) sowie andere Interpolationsmethoden zur Visualisierung von evolutionären Übergängen verwendet, jedoch handelt es sich dabei stets um optische oder parametrische Interpolationen anstatt echter funktional-evolutionärer Veränderungen. Es ist nicht garantiert, dass die Zwischenstadien funktional korrekt und somit lebensfähig sind. Darüber hinaus ist eine Kombination von evolutionären Animationen und Bewegungsanimationen – wenn überhaupt – nur sehr umständlich mit enormen Zeit- und Kostenaufwand möglich, indem Bewegungsanimation und evolutionäre Animationen stückweise alternierend verwendet werden. Es kann mit dieser alternierenden Herangehensweise weder eine überzeugend-realistische Animation noch eine biologische Stringenz der Resultate hinsichtlich Bewegung und Mutation gewährleistet werden.

Keine der betrachteten Methoden erlaubt es zudem, einen einmal modellierten Organismus zu verwenden, um ihn entlang evolutionärer Linien in Richtung verwandter Organismen zu mutieren. Daher besteht keine volle Wiederverwendbarkeit der modellierten und animierten Strukturen.

Die Anlehnung an die natürliche Verfahrensweise der Evolution könnte daher die Herstellung von organismischen Konstruktionen beschleunigen. Es könnte – analog zu Mutationen einer bereits bestehenden Art – auf bereits erstellte Konstruktionen zurückgegriffen werden, um diese zielgerichtet zu modifizieren.

Es bestünde auf diese Weise zudem die Möglichkeit, evolutionäre Übergänge oder zusätzliche Animationen implizit zu visualisieren. Durch die Einbindung biologischer Constraints könnte die funktionale Stringenz des Organismus gewährleistet werden, so dass die evolutionäre Animation und die Bewegungsanimation miteinander kombiniert werden könnten.

Ein solcher Modellierungsansatz entlang evolutionärer Linien mit Hilfe von Deltabeschreibungen würde sowohl die Resultate verbessern als auch die Reusability wesentlich erhöhen. Die Kopie von evolutionären Verfahrensweisen der Natur könnte in diesem Sinne die Graphische Datenverarbeitung bereichern, ähnlich wie die Bionik andere technische Sparten verbessert.

3.5 Reusability von anatomischen Strukturen

Die im Kapitel 2 betrachteten Methoden haben den Nachteil, dass einmal modellierte anatomische Strukturen nur bedingt auf andere Organismen übertragbar sind. Insbesondere wenn Ursprungs- und Zielorganismus jeweils einer anderen Spezies angehören, eine unterschiedliche Größe haben oder hinsichtlich des Alterungsstadiums verschieden sind, gestaltet sich die Übertragung sehr umständlich. Hier müssen die modellierten anatomischen Strukturen reskaliert, die internen Proportionen verändert und die Farbgebung an den Zielorganismus angepasst werden. Hinsichtlich der Animation müssen die Bewegungsabläufe der modellierten anatomischen Strukturen mit denjenigen des Zielorganismus synchronisiert werden. Der gesamte Prozess der Übertragung kann sich so umständlich gestalten, dass eine komplette Neumodellierung der Strukturen einfacher und schneller zu bewerkstelligen ist. Die Wiederverwendbarkeit von modellierten Strukturen ist bei den herkömmlichen Methoden also nicht gewährleistet.

In Anbetracht dieser mangelnden Reusability wäre eine neue Methode wünschenswert, bei welcher die anatomischen Strukturen sich automatisch im Zuge des Übertragungsprozesses an den Zielorganismus adaptieren. Diese Adaption sollte – wenn möglich – Größe, Form,

Anbindung, Farbe und Texturierung umfassen. Im Zuge einer solchen automatischen Reskalierung würde das mühsame Nachskalieren und Nachmodellieren von Hand entfallen. Auch das Ändern der Bewegungsparameter würde obsolet werden. Der Modellierungs- und Animationsprozess würde sich somit beschleunigen. Auch verspricht eine solche automatische Reskalierung nach biomechanischen Regeln biologisch sinnfälligere Resultate zu liefern.

Neben einer verbesserten Wiederverwendbarkeit von einmal modellierten Strukturen, würde eine solche automatische Adaption auch die einfache Kreierung von Mischwesen und Kreuzungen erlauben. Hierzu müssten die modellierten und animierten Strukturen verschiedener Spezies lediglich „zusammenkopiert“ werden

3.6 Verwendung medizinischer Termini

In den in der State-of-the-Art-Analyse betrachteten Modellierungs- und Animationsansätzen für Organismen werden fast ausschließlich rein mathematische bzw. kombiniert mathematisch-physikalische Beschreibungen der Positionen, der Skalierung und der Form von Organen, Körperteilen und anderen anatomischen Strukturen verwendet.

So verwendet Blobby Modelling (siehe Kapitel 2.1.7) die lokalen Summen radialer Kraftfelder um imaginäre Punkte im euklidischen Raum. Die Constructive Solid Geometry verwendet geometrische Primitive in Verbindung mit Boolschen Operatoren (siehe Kapitel 2.1.8). Sweep-Körper verwenden Flächen, die entlang eines Pfades geführt werden. (siehe Kapitel 2.1.8)

Bei der Erstellung des Datensatzes für die Finite Elemente Stress Analysis und anderen Simulationen, die auf Finiten Elementen basieren (siehe Kapitel 2.4.2.2) werden Voxel-Elemente verwendet. Das Gleiche gilt für die meisten anderen Simulationsmethoden zur Objektdeformation (siehe Kapitel 2.4.2.3 und 2.4.2.4) Andere, wie z.B. das System von Müller et al. (siehe Kapitel 2.4.2.4) verwenden ein System dreidimensionaler „Stress Tensors“.

Die Pfadanimation (siehe Kapitel 2.3.2) greift auf ein globales Koordinatensystem zurück. Die verschiedenen Schichtmodelle (siehe Kapitel 2.5.1) inklusive 3D-Skinning (siehe Kapitel 2.5.1.2), kinematische Techniken (siehe Kapitel 2.3.3) und parametrisches Keyframing (siehe Kapitel 2.3.1) verwenden letzten Endes ein ausgehend vom globalen kartesischen Koordinatensystem hierarchisch strukturiertes System lokaler kartesischer

Koordinatensysteme, um die Positionen und Bewegungen der animierten Strukturen zu beschreiben. Selbst diejenigen Methoden, welche ein biologisches Modell verwenden, wie beispielsweise das Modell von Nedel und Thalmann (siehe Kapitel 2.5.2.3), greifen bei der Positionierung, Skalierung und Formbeschreibung der anatomischen Strukturen letzten Endes auf rein mathematische Beschreibungen zurück.

Bei den Bewegungs- und Formerfassungstechniken wie 3D-Scanning (siehe Kapitel 2.2.2), Motion Capturing (siehe Kapitel 2.3.4) und Stop Motion Input Devices (siehe Kapitel 2.3.5) bleiben die internen Bewegungs- und Geometriebeschreibungen dem Benutzer transparent, intern werden aber ebenfalls mathematische Beschreibungen verwendet.

L-Systeme (siehe Kapitel 2.1.6) stellen eine lobenswerte Ausnahme dar, da sie als einzige untersuchte Methode auf biologische Beschreibungsmethoden zurückgreifen. L-Systeme sind jedoch auf die Beschreibung pflanzlicher Strukturen spezialisiert. Für die Beschreibung tierischer Organismen sind sie nur bedingt geeignet.

Kein einziges betrachtetes Modell verwendet also biologische Beschreibungen zur Positionierung, Skalierung und Formdefinition der anatomischen Strukturen von tierischen Organismen, die meisten Verfahren greifen dagegen auf rein mathematische bzw. physikalisch-mathematische Termini zurück.

Rein mathematische oder physikalisch-mathematische Termini sind allerdings nur bedingt dazu geeignet, biologische Strukturen zu beschreiben. Schließlich sind diese Beschreibungen nicht invariant hinsichtlich Statusveränderungen eines Organismus, wie Bewegungen, Deformationen infolge interner oder externer Kräfte, Wachstumsprozesse, Altersdegenerationen oder imaginäre evolutionäre Transitionen. Schon eine kleine diesbezügliche Veränderung kann eine Positions- oder Formdefinition anatomischer Strukturen obsolet machen.

Ein weiterer wichtiger Nachteil von rein mathematischen oder physikalisch-mathematischen Beschreibungssprachen resultiert aus der Tatsache, dass sie in der Regel nicht Bestandteil der medizinischen oder biologischen Ausbildung sind. Sowohl das damit verbundene Vokabular als auch die damit verbundene Denkweise ist Personen, die aus dem medizinischen bzw. biologischen Arbeitsbereich stammen, in der Regel fremd. Eine Methode, welche eine rein mathematische bzw. mathematisch-physikalische Beschreibung zur Positionierung der anatomischen Strukturen verwendet, setzt ihnen daher von vorneherein eine große Hürde für das Erlernen und Begreifen. Infolge dieser

Schwierigkeiten ist die allgemeine Akzeptanz der Methode in biomedizinischen Kreisen nicht gewährleistet.

Somit werden wichtige Zielpersonen nicht angesprochen, die von computergenerierten 3D-Organismen am ehesten betroffen sind und aus neuen Methoden in diesem Bereich große Vorteile ziehen könnten. Vor allem Paläontologen, Evolutionswissenschaftler, Zoologen, Chirurgen, Neurologen, Pathologen und Gynäkologen könnte die Möglichkeit „selbst gestrickter“ 3D-Visualisierungen anatomischer Strukturen neue Perspektiven eröffnen. So würde ihnen beispielsweise ermöglicht, Präsentationen für die Didaktik oder Visualisierungen als Diskussionsgrundlage selbstständig zu erstellen.

Eine neue Methode zur Generierung dreidimensionaler virtueller Organismen sollte daher vorzugsweise auf das tradierte Vokabular der Medizin und Biologie zurückgreifen, um es als Basis für intuitive Positions- und Geometriebeschreibungen zu verwenden.

Eine solche Beschreibung mit Hilfe medizinisch-biologischer Termini verspricht – zusätzlich zur Verbesserung der Intuitivität – eine höhere Invarianz bezüglich Statusveränderungen zu gewährleisten, schließlich wurden diese Beschreibungen empirisch so gewählt, dass ihre Bedeutung – auch während Statusveränderungen – wenn überhaupt nur geringfügig variiert.

3.7 Integration biologischer Constraints

Die in der State-of-the-Art-Analyse betrachteten Ansätze basieren, von den dynamischen Muskelmodellen (siehe Kapitel 2.5.2) und den L-Systemen (siehe Kapitel 2.1.6) abgesehen, durchweg auf rein mathematischen und technozentrischen Konzepten, welche keine biologischen Regeln berücksichtigen: Die räumliche Dynamik des internen Kräftespieles wird – wenn überhaupt versucht wird sie zu berechnen – vereinfachend in konzentrische Körperschichten (siehe Kapitel 2.5.1), in durch viskoelastische Elemente verbundene Massepunkte (siehe Kapitel 2.5.2.1) oder in orthogonal angeordnete Voxel-Elemente (siehe Kapitel 2.1.3) eingeteilt. Diese räumlichen Simulationsstrukturen stammen aus technischen Bereichen wie beispielsweise dem Maschinenbau. Mit der tatsächlichen topologischen Komplexität von anatomischen Strukturen haben sie wenig gemein.

Eine Ausnahme hiervon stellt das Modell von Nedel und Thalman dar, welches zumindest versucht, die Muskeldynamik mit Hilfe von Action Lines recht realistisch zu

simulieren (siehe Kapitel 2.5.2.3). Das Modell von Nedel und Thalman hat jedoch andere Defizite bezüglich der Einhaltung biologischer Constraints, wie z.B. die Instabilität bei größeren anatomischen Strukturen, keine Integrationsmöglichkeit von evolutionären Prozessen und Wachstumsvorgängen oder fehlende Volumenkonstanz bei Körperdeformationen.

Eine andere Ausnahme bilden L-Systeme (siehe Kapitel 2.1.6), welche pflanzliche Wachstumsregeln des Botanikers Aristid Lindenmayer als Basis verwenden. Allerdings sind die biologischen Regeln, welche das Wachstum von Pflanzen ausreichend beschreiben, nicht ohne weiteres auf die Tierwelt übertragbar. Insbesondere ist der optische Einfluss der hydropneumatischen und kontraktile Strukturen, welche für die Bewegung und Deformation maßgeblich sind, in der Fauna in der Regel größer als in der Flora. Umgekehrt haben Tiere – von einigen Ausnahmen abgesehen – keine filigran verzweigten Strukturen, die mit L-Systemen hervorragend beschrieben werden können.

Im Großen und Ganzen kann also konstatiert werden, dass Modelle zur Organismengenerierung, welche biologische Constraints integrieren, noch ausstehen.

Es wird in der Regel versucht, biologisch korrektes Verhalten durch biologisch inkorrekte Modelle zu erzeugen, was nur schwer gelingen kann. Biologisch gesehen sind daher die Bewegungen und Deformationen inkorrekt und haben eine unnatürliche Anmutung. Die Nichteinhaltung von biologischen Maßgaben bei herkömmlichen Methoden zur Organismengenerierung impliziert daher, dass durch ausschließlich computerbasierte Methoden keine wirklich überzeugenden tierischen Organismen generiert werden können.

Um hier Abhilfe zu schaffen, sollten die vielfältigen funktionalen und strukturellen Interdependenzen zwischen Gewebeverbänden, also Organen und anatomischen Strukturen, und deren mechanischen Eigenschaften berücksichtigt werden. Dies kann nur dann gelingen, wenn das Modell inhärent biomechanische Regeln als Maßgaben beinhaltet, welche die notwendigen Constraints für ein korrektes biologisches Verhalten des Organismus liefern. Die Fokussierung auf völlig neuartige, biologienahe Modelle wird sich daher für die Organismengenerierung als lohnenswert erweisen.

3.8 Kanalisierte Modellierung und Animation

Die untersuchten Modellierungs- und Animationsansätze erlauben die Erzeugung biologisch abwegiger und/oder nicht überlebensfähiger Organismen. Diese Freiheit der Modellierung erweist sich eher als ein Nachteil als ein Vorteil, denn der Modellierer bzw. Animator bekommt bezüglich der biologischen Stringenz seiner Organismen keine Hilfestellungen geliefert. Die Modellierung kann daher als ungerichtet und nicht-assistiert bezeichnet werden. Insbesondere die Generierung von paläontologischen oder fiktiven Organismen ohne natürliche Vorlage bzw. Maquette kann sich unter Umständen zu einem langwierigen Trial-and-Error-Prozess entwickeln.

Eine Ausnahme von der Regel stellen L-Systeme (siehe Kapitel 2.1.6) dar, bei denen schon eine auf pflanzliche Strukturen gerichtete Restriktion beim Modellierungsprozess erfolgt. Allerdings sind diese Restriktionen der L-Systeme für tierische Organismen wiederum zu eng gefasst.

Hier wäre ein interaktiver Modellierungs- und Animationsprozess von Vorteil, bei dem der Benutzer unmittelbar eine quantitative Rückkopplung während seines kreativen Prozesses erhält. Der kreative Prozess sollte von einer parallel stattfindenden Simulation assistiert werden, welche die Einhaltung biomechanischer Maßgaben überprüft. Somit könnte sowohl der Modellierungs- als auch der Animationsprozess in Chreoden gelenkt und beschleunigt werden. (Eine „Chreode“ ist ein von Waddington geprägter Begriff aus der Evolutionsbiologie, der einem zielgerichteten, sich verzweigenden Flusslaufsystem in einer epigenetischen Landschaft entspricht [Waddington 1957])

In einem solchen Falle könnte man auch von einer biologisch assistierten Organismengenerierung sprechen.

Neben der Beschleunigung des Modellierungs- und Animationsprozesses verspricht die chreodische Assistenz auch realistischere Resultate zu liefern, da selbst biologisch unversierte Menschen nur biologisch sinnfällige Organismen erzeugen können, die infolge ihrer inneren Stringenz auch dem Betrachter realistisch erschienen.

3.9 Echtzeitfähigkeit der Animation

Viele der in der State-of-the-Art-Analyse betrachteten Ansätze sind nicht ausreichend performant, um Echtzeitfähigkeit hinsichtlich der Animation zu gewährleisten. Zwar existieren Methoden, die von linearer Komplexität bezüglich der Anzahl der Vertices im Skin-Mesh sind, wie z.B. das 3D-Skinning (siehe Kapitel 2.5.1.2), das Modell von Nedel und Thalmann (siehe Kapitel 2.5.2.3) und prinzipiell alle Methoden, die auf Feder-Masse-Dämpfer-Modelle basieren (siehe Kapitel 2.4.2.1) und somit zumindest theoretisch echtzeitfähig sein sollten, in der Praxis werden jedoch jeweils pro Vertex und pro Kalkulationsschritt meist mehrere hundert Rechenoperationen benötigt, so dass die angebliche Echtzeitfähigkeit zurzeit entweder nur für einfache Modelle oder unter Einsatz von überdurchschnittlich leistungsfähiger Hardware zum Tragen kommt.

Wenn die erhöhte Rechengeschwindigkeit mit dem erweiterten Gesetz von Moore in Betracht gezogen wird, dürfte zwar – zumindest bei diesen linearen Methoden – die Berechenbarkeit von komplexeren Organismenmodellen auf Standard-PCs schon in naher Zukunft in den Bereich des Machbaren rücken, im gleichen Maße dürften aber auch die Ansprüche an virtuelle Organismen steigen, so dass das Performanzproblem nur verlagert wird. [Moravec 1998a & b][Moore 1997 & 2003]

Es existieren auch Simulationsmethoden zur Objektdeformation, welche von sich behaupten, echtzeitfähig zu sein. Dazu zählen die Methoden Desbrun et al. und DeBunne et al., die Echtzeitfähigkeit gilt allerdings erstens nur für die Verformungssimulation von sehr kleinen Gewebestrukturen und zweitens nur bei lokal eng begrenzten Deformationen. (siehe Kapitel 2.4.2.4).

Auch die Methode von Müller et al. behauptet von sich, infolge einer linearen Komplexität echtzeitfähig zu sein. Die Linearität bezieht sich jedoch bei genauerem Hinsehen nicht auf die Anzahl der Vertices, sondern auf andere interne Strukturen, so dass auch in naher Zukunft die Echtzeitfähigkeit bei der Visualisierung hochkomplexer Organismen nicht gewährleistet ist. (siehe Kapitel 2.4.2.4).

Natürlich existieren auch Animationsmethoden, die sowohl in der Theorie als auch in der Praxis voll echtzeitfähig sind, da sich ihre Linearität auf die animierten Gesamtobjekte bezieht, wie z.B. der Rumpf oder die Gliedmaßen. Zu diesen voll echtzeitfähigen Methoden zählen beispielsweise das parametrische Keyframing (siehe Kapitel 2.3.1), die Pfadanimation (siehe Kapitel 2.3.2) und kinematische Techniken (siehe

Kapitel 2.3.3). Realistisch anmutende Animationen komplexer Bewegungsabläufe sind mit diesen Techniken jedoch nicht möglich.

Es gibt zudem verschiedene Überblendungstechniken sowie diverse 2D- und 3D-Morphingtechnologien, die ebenfalls als voll echtzeitfähig angesehen werden können (siehe Kapitel 2.4.3). Mit diesen Methoden können jedoch lediglich Wachstumsvorgänge oder einfache evolutionäre Übergänge animiert werden, nicht jedoch komplexe Bewegungsabläufe.

Zusammenfassend kann also konstatiert werden, dass unter den Animationsmethoden, die in der State-of-the-Art-Analyse beleuchtet wurden, keine Methode existiert, welche hinsichtlich Echtzeitanwendungen ausreichend performant ist und mit Hilfe derer gleichzeitig überzeugende Bewegungsabläufe der Organismen generiert werden können.

Dieses Manko schränkt das Anwendungsspektrum der betrachteten Methoden stark ein. Alle Echtzeitanwendungen wie beispielsweise Trainingssimulationen, Echtzeit-Spiele, interaktive analytische Forschungstools können damit keine überzeugenden Organismen integrieren.

Die in Kapitel 3.8 geforderte Interaktivität des Modellierungs- und Animationsprozesses inklusive der damit zusammenhängenden parallel integrierten Simulationsassistenz benötigen jedoch performante Methoden, um die Simulation anatomischer Strukturen unter Berücksichtigung ihres mechanisch kohärenten Zusammenhanges bei gleichzeitiger Echtzeitvisualisierung derselben zu ermöglichen.

Hier sind komplett neue Ansätze mit weit performanteren algorithmischen Basisverfahren gefragt als diejenigen, welche in Kapitel 2.4 betrachtet wurden. Diese Ansätze müssen mindestens von linearer Komplexität bezüglich der Anzahl der Vertices – oder noch besser bezüglich größerer anatomische Strukturen – sein.

Zudem sollte auch die Anzahl der Operationen pro Deltatick reduziert werden, um nicht nur in der Theorie, sondern auch in der Praxis eine volle Echtzeitfähigkeit auf Standard-PCs zu gewährleisten. Ein möglicher lohnenswerter Ansatz zur Reduktion der Rechenoperationen wäre es, eine biomechanische Simulation mit einer Geometrie-Repräsentation aus der Computergraphik zu verschmelzen, um die Redundanz bei der Berechnung zu verringern, denn schließlich müssen biologisch-physikalische Simulationen von Organismen auf deren dreidimensionale Daten zurückgreifen, wie beispielsweise die aktuellen Positionen der Oberflächenpunkte, die Abstände zwischen zwei Körpern oder

das interne Körpervolumen. Viele dieser Daten müssen auch für den Aufbau der Geometrie berechnet werden. Es ist also unsinnig, die Simulation unabhängig von der Geometrie-Repräsentation auszuführen. Dieser neue holistische Ansatz setzt natürlich voraus, dass über den Tellerrand der Graphischen Datenverarbeitung geschaut wird, um interdisziplinäre Schnittmengen zwischen biomechanischen Simulationen und Geometrieberechnungen aufzuspüren.

Zusätzlich zur Modellfindung müssen natürlich Implementierungsaspekte berücksichtigt werden, um auch in der Praxis eine Geschwindigkeitsoptimierung zu erzielen. Zu potentiellen Optimierungsmaßnahmen bei der Implementierung für das Speed-Up zählen beispielsweise die Verwendung einer hardwarenahen Programmiersprache, für die ein bezüglich des resultierenden Maschinencodes gut optimierender Compiler verfügbar ist, die konsequente Verwendung von Makroroutinen bei häufig ausgeführten Methoden, die Berücksichtigung der minimalen Cache-Größen bei der Programmierung von zusammenhängenden Codefragmenten, wenn möglich die Ersetzung von Double Precision Floating Point Variablen durch einfache Floating Point Variablen, die Vermeidung von verketteten Listen mit verstreuten Laufzeitallokationen und die Verschiebung möglichst vieler Berechnungen in die Initialisationsroutinen.

3.10 Hohe Bandbreite der Abstraktionsstufen

Viele der in der State-of-the-Art-Analyse betrachteten Methoden vermögen lediglich Strukturen aus dezidierten Granularitätsebenen zu visualisieren. Um die Bandbreite der möglichen Abstraktionsstufen zu quantifizieren, eignet sich besonders das Konzept der *Holons* von Koestler, welches von Sheldrake im Sinne der Biologie weiterentwickelt wurde. [Koestler 1984 & 1989][Sheldrake 1988, S.158ff] Ein Holon ist dabei gemäß Koestler eine Entität, die wiederum aus Subentitäten einer untergeordneten Art aufgebaut ist, bis zu einem gewissen Grad autonom handeln kann und dabei sowohl die Subentitäten morphologisch strukturiert, als auch wiederum von übergeordneten Entitäten strukturiert wird. Auf diese Weise wird eine stabile *Holonhierarchie* oder kurz *Holarchie* aufgebaut. Mit Hilfe dieses Modells können zehn biologisch relevante *Holonhierarchiestufen* bzw. *Holarchiestufen* unterschieden werden:

1. Biochemische Molekülstrukturen
2. Organellen
3. Zellen
4. Zelluläre Netzwerke und Gewebestrukturen
5. Suborgane
6. Organe
7. Anatomische Strukturen
8. Gesamtorganismen
9. Schwärme, Herden und andere Organismengemeinschaften
10. Populationen und Ökosysteme

Um ein großes Anwendungsspektrum zu gewährleisten, müssen aus Sicht der Granularität möglichst viele Holarchiestufen von einer Methode abgedeckt werden. Während die in der State-of-the-Art-Analyse betrachteten Modellierungsansätze diesbezüglich in der Regel eine hohe Bandbreite abdecken, ist dies jedoch bei den analysierten Animationsmethoden nur bedingt der Fall:

Das parametrische Keyframing (siehe Kapitel 2.3.1) und die Pfadanimation (siehe Kapitel 2.3.2) eignen sich primär zur Animation von Schwärmen, Herden und anderen Organismengemeinschaften, und nur sehr bedingt zur Animation von Einzelorganismen, da keine komplexen Bewegungsabläufe animiert werden können. Somit wird im Wesentlichen nur die Holarchiestufe 9 erfasst.

Kinematische Techniken (siehe Kapitel 2.3.3) und Schichtmodelle (siehe Kapitel 2.5.1) eignen sich lediglich zur Animation von Gesamtorganismen und anatomischen Strukturen. Das Gleiche kann von konventionellen Bewegungserfassungsmethoden wie Motion Capturing (siehe Kapitel 2.3.4) oder Stop Motion Input Devices (siehe Kapitel 2.3.5) behauptet werden. Somit erfassen diese Techniken nur Strukturen aus den Holarchiestufen 7 und 8.

Reine Basistechniken zur anatomischen Simulation (siehe Kapitel 2.4), beispielsweise einfache Objektdeformationstechniken wie die Nichtlineare Globale Deformation, FFDs oder EFDs (siehe Kapitel 2.4.1), Feder-Masse-Dämpfer-Systeme (siehe Kapitel 2.4.2.1), Techniken aus dem Bereich der Finiten Elemente (siehe Kapitel 2.4.2.2),

Randelementmethoden (siehe Kapitel 2.4.2.3) und die Methoden von Desbrun, Debunne, Wu und Müller et al. (siehe Kapitel 2.4.2.4), können vornehmlich für die Deformation von homogenen Gewebestrukturen, Suborganen und bestimmten Organen verwendet werden. Strukturen aus dem Bereich der Zytologie oder der Gesamtanatomie können dagegen aufgrund der fehlenden hydropneumatischen Constraints nicht realistisch animiert werden. Somit werden hier nur die Holarchiestufen 4 bis 6 teilweise erfasst.

Mit Hilfe der Modelle von Terzopoulos (siehe Kapitel 2.5.2.1) und Miller (siehe Kapitel 2.5.2.2) können lediglich Einzelorganismen animiert werden, also Strukturen der Holarchiestufe 8.

Das Dynamische Muskelmodell von Nedel und Thalmann (siehe Kapitel 2.5.2.3) ist auf anatomische Strukturen (Knochen, Sehnen und Muskeln) spezialisiert, animiert also hauptsächlich Strukturen der Holarchiestufe 7.

Im Großen und Ganzen kann daher konstatiert werden, dass zwar fast alle der Holarchiestufen 4 bis 9 durch diverse Organismenanimationsmethoden abgedeckt werden, jedoch umgekehrt keine einzige konventionelle Animationsmethoden existiert, die eine Bandbreite von über drei Hierarchiestufen innerhalb der Holarchie aufweist. Insbesondere diejenigen Methoden, welche eine realistische Modellierung und Animation erlauben, weisen in der Regel eine sehr geringe diesbezügliche Bandbreite auf. Somit können Animatoren nicht auf ein auf alle Strukturen allgemein verwendbares Verfahren zurückgreifen. Neben der damit verbundenen Umständlichkeit wird infolge dieses Defizits ein interaktives Zooming zwischen verschiedenen Granularitätsstufen verhindert.

Daher sollte eine wesentliche Anforderung an ein neues Verfahren sein, dass sie eine hohe Bandbreite innerhalb der Holarchie aufweist. Da biochemische Molekülstrukturen sowie Populationen und Ökosysteme von ihren Visualisierungsanforderungen infolge ihrer fehlenden Biomechanik eine Sonderstellung innerhalb der 10 biologisch relevanten Holarchiestufen einnehmen, sollte eine neue Methode vorzugsweise in der Holarchie alle Holarchiestufen zwischen 2 und 9 umspannen. Dies würde neben dem Wechsel zwischen Visualisierungen verschiedener Abstraktionsstufen auch die Visualisierung interholarchischer Zusammenhänge ermöglichen.

3.11 Hoher Grad des Realismus

Sowohl in der Biologie als auch in der Medizin wären für bestimmte Zwecke neue Methoden zur dreidimensionalen Organismengenerierung wünschenswert, die einen abstrahierenden Charakter haben. Beispiele für solche Abstraktionen sind z.B. die Schraffur- und Umrisszeichnungen von anatomischen Objekten, wie sie öfters für didaktische Zwecke verwendet werden. Insofern ist die Frage berechtigt, ob die Fähigkeit einer Methode, realistische Resultate zu zeitigen, die optisch nicht vom Original unterschieden werden können, überhaupt in das Anforderungsspektrum eingehen sollte.

Um diese Frage zu beantworten, muss zunächst einmal betrachtet werden, wie medizinische oder biologische 3D-Abstraktionen von anatomischen Strukturen oder Gesamtorganismen normalerweise erzeugt werden:

In der Regel wird eine computergenerierte 3D-Abstraktion durch Selektion und Modifikation von Form-, Bewegungs- und Farbparametern eines realistisch generierten Organismus oder seines gerenderten Abbildes erfolgen, ein Beispiel hierfür ist die Selektion und Modifikation bestimmter Pixel zur Umrissbildung. (Eine einfache Methode hierfür wäre die Differenzbildung von nebeneinander liegenden z-Bufferwerten und der Einfärbung der entsprechenden Pixel bei Überschreitung eines Grenzwertes). Somit müssen zuerst Methoden existieren, die eine realistische Darstellung erlauben und erst danach kann nach geeigneten Maßnahmen zur Modifikation derselben gesucht werden, so dass daraus abstrahierende Methoden abgeleitet werden.

Modellierungs- und Animationsansätze, welche realistisch anmutende Resultate liefern, sind gemäß der Recherche allerdings nur sehr bedingt vorhanden:

So vermögen mechanooptische Stiftverfahren (siehe Kapitel 2.2.1) und 3D-Scanning-Technologien (siehe Kapitel 2.2.2) zwar realistische Formen zu erzeugen, allerdings sind diese Verfahren keine Modellierungs- sondern reine Formerfassungsmethoden. (siehe Kapitel 2.2) Die damit verbundenen Nachteile werden in Kapitel 3.3 behandelt.

Dagegen sind mit Blobby Modelling (siehe Kapitel 2.1.7) keine realistischen Organismen modellierbar.

Mit L-Systemen (siehe Kapitel 2.1.6) ist zwar eine realistische Modellierung der Flora, nicht jedoch der Fauna möglich.

Mit fraktalen Techniken können höchstens Teilaspekte eines Organismus (z.B. Fell) realistisch erzeugt werden. (siehe Kapitel 2.1.8)

Andere Modellierungsansätze, wie beispielsweise die CSG oder Sweep-Körper (siehe Kapitel 2.1.8), sind nur bedingt dazu geeignet, Organismen zu modellieren, da sie eher kantige oder extrudierte Formen aus dem technischen Bereich erzeugen.

Bewegungserfassungstechnologien wie Motion Capturing (siehe Kapitel 2.3.4) oder Stop Motion Input Devices (siehe Kapitel 2.3.5) vermögen zwar bezüglich der Bewegungsabläufe realistische Resultate zu erzeugen, haben allerdings die gravierende Nachteile von Aufzeichnungstechnologien, welche im Kapitel 3.3 beschrieben sind.

Mit parametrischem Keyframing (siehe Kapitel 2.3.1), Pfadanimation (siehe Kapitel 2.3.2) und kinematischen Techniken (siehe Kapitel 2.3.3) können in der Regel keine realistischen Bewegungen erzeugt werden.

Der Grad des Realismus bei Schichtmodellen (siehe Kapitel 2.5.1) hängt von verschiedenen Parametern wie insbesondere der Anzahl der berechneten Schichten und die zur Berechnung der Schichten verwendeten Basismethoden ab. Generell gilt bei der Schichtmodellen jedoch, dass ihre finale Realistik nicht die Realistik der verwendeten Bewegungsdaten überschreiten kann. Da realistische Bewegungsdaten für Organismen nur von Bewegungserfassungstechnologien erzeugt werden können, übertragen sich ihre Probleme auf die entsprechenden Schichtmodelle.

Dynamische Muskelmodelle (siehe Kapitel 2.5.2) vermögen zwar teilweise realistische Resultate zu erbringen, allerdings sind sie auf Teilaspekte der Organismengenerierung beschränkt, z.B. das Modell von Terzopoulos und Tu (siehe Kapitel 2.5.2.1) nur auf Fische, das Modell von Miller et al. (siehe Kapitel 2.5.2.2) nur auf Würmer und Schlangen oder das Modell von Nedel und Thalmann (siehe Kapitel 2.5.2.3) nur auf Knochen und Muskeln.

Somit existieren keine Methoden zur Organismengenerierung, die sowohl realistische Resultate erbringen und gleichzeitig keine anderen gravierenden Nachteile haben. Dies impliziert auch, dass die primäre Anforderung an eine neue Methode zur Organismengenerierung ein möglichst hoher Grad an Realismus ist. Die zusätzliche Möglichkeit der Generierung abstrahierender Visualisierungen ist zwar wünschenswert, jedoch zurzeit noch sekundär.

4 Das Quaoaring-Konzept

Cogitationis poenam nemo patitur.
Aus dem Corpus iuris civilis, 534 n.Chr.

In diesem Kapitel wird das eigene Konzept vorgestellt. Es besteht aus einem Bündel alter und neuer Methoden, um Organismen zu modellieren und zu animieren. Diese verschiedenen Techniken ergänzen sich synergistisch zu einer neuen Software-Technologie, die es erlaubt, alle möglichen Arten von realen oder imaginären organischen Einheiten zu erzeugen, angefangen von Organellen und zellulären Einheiten bis hin zu Gesamtorganismen und Gruppen derselben (im Folgenden vereinfachend als Organismen bezeichnet). Zugleich erlaubt diese Technologie, Bewegungen, Wachstumsprozesse und sogar evolutionäre Formveränderungen zu animieren.

Die Technologie wird im Folgenden als *Quaoaring* bezeichnet. Die Gründe der Namenswahl stehen in Anhang 9.1.1. auf Seite 369.

Quaoaring integriert eine biologische Simulation in ein neues Geometriemodell der Computergraphik, was es den Modellierern auf einer hohen Abstraktionsebene erlaubt, intuitive Metaphern aus der Biologie zu verwenden, um die Form, die Oberflächenstrukturen und weitere Attribute des Organismus zu beschreiben.

Das biologische Simulationsmodell basiert auf der Frankfurter Organismus- und Evolutionstheorie, welche indirekt Organismen – vom anatomischen Standpunkt her – größtenteils als komplexe hierarchische Strukturen aus miteinander interagierenden pneumatischen Einheiten ansieht. Dazu wird ein neuer Algorithmus vorgestellt, der eine interaktive Berechnung dieser hydropneumatischen Biosimulation in Echtzeit erlaubt und auf einer numerischen Kalkulation von mechanisch miteinander verbundenen Pneus beruht. Unter Verwendung dieses Algorithmus entsteht eine Simulation, die sich biologisch stringent verhält, also beispielsweise Volumenkonstanz während der Bewegung bewahrt.

Am Ende dieses Kapitels wird ein Autorenprozess beschrieben, wie Organismen mit der neuen Technologie modelliert und animiert werden können.

4.1 Grundidee

Die Inspiration für das Modell des Quaoarings stammt von der Frankfurter Organismus- und Evolutionstheorie. Aus ihrer Sicht stellen Organismen – auf der biomechanischen Ebene – komplexe hydropneumatische, energiewandelnde Konstruktionen dar. Ihre Körperformen und Bewegungen werden weitgehend durch verspannende, stabilisierende, kräfteerzeugende und kräfteübertragende Strukturen erzeugt, die den Gesetzen der klassischen Hydropneumatik folgen. [Gutmann & Bonik 1981] [Gutmann 1995] [Gudo 2002] [Gudo & Grasshoff 2002] Diese Sichtweise mag auf den ersten Blick ungewohnt erscheinen, sie ist jedoch vielfach belegt und wird erfolgreich in der Praxis bei der Überprüfung der biomechanischen Kohärenz von Organismen angewandt. [Bonik 1977] [Bonik 1978] [Gudo 2002] [Gudo et. al 2002]

So entstand die Idee, Organismen – von der anatomischen Warte aus betrachtet – als eine komplexe Hierarchie unterschiedlicher hydropneumatischer Einheiten anzusehen. Die verschiedenen hydropneumatischen Einheiten, deren Hüllen wie auch Innenfüllungen mehr oder weniger viskos sein können und die in einem Medium eingebettet sind, können dabei mechanisch untereinander interagieren.

Diese originelle Sichtweise von Organismen liefert die Grundlage für ein Modell, welches aus folgenden Gründen für die virtuelle Organismengenerierung geeignet ist:

- **kanalisierte Modellierung:** Das Modell erlaubt der Computergraphik, die Form eines Organismus zu modellieren. Biologisch nicht stringente Organismen können mit diesem Modell jedoch nicht modelliert werden, so dass der Modellierungsprozess automatisch in die richtige Bahn gelenkt wird.
- **Verwendung einer intuitiven Metapher:** Das Modell basiert auf hydropneumatischen Einheiten, im Folgenden verkürzend als *Pneus* bezeichnet. Ein Pneu als „wassergefüllter Ballon“ liefert eine geeignete und intuitive Metapher für den Modellierer. Das Erlernen des Modellierungsprozesses wird dadurch vereinfacht und beschleunigt.

- **frei wählbare Granularitätsebene in der Holarchie:** Die Ebene in der natürlichen Holarchie, auf der sich arbeiten lässt, ist frei wählbar. Die Bandbreite der Holongranularität reicht von der Ebene der Organellen über ganze Zellen, Zellverbände, Organe, Körperteile bis hin zu ganzen Organismen oder Gruppen derselben. (zur Erklärung der Begriffe „Holon“ und „Holarchie“ siehe Kapitel 3.10)
- **Erzeugung fiktiver oder paläontologischer Organismen:** Durch die Verwendung allgemeiner biologischer Regeln können auch fiktive oder paläontologische Organismen glaubwürdig modelliert und animiert werden.
- **Wiederverwendbarkeit:** Generierte Organismen können durch kleine Delta-Modifikationen in verwandte Tiere überführt werden. Teile der Organismen sind einfach auf andere Organismen übertragbar und adaptieren sich durch den veränderten hydrodynamischen Kontext automatisch auf den entsprechenden Zielorganismus. Dies garantiert eine hohe Wiederverwendbarkeit von einmal modellierten Organismen und Teilen derselben.
- **mechanische Simulationsmöglichkeit:** Die Formveränderung eines Organismus unter Einwirkung von externen Kräften kann simuliert werden. Die mechanische Simulation kann zudem nahtlos in das Geometriemodell integriert werden.
- **hohe Abstraktionsebene:** Es werden hochabstrahierende Verhaltensparameter für den Modellierer und den Animator angeboten. Das biologische Simulationsmodell verschafft die Möglichkeit, Änderungen der Verhaltensparameter sofort in Bewegungs- und Formveränderungen zu übersetzen.
- **hohe Verhaltensbandbreite:** Das Modell erlaubt es, eine große Bandbreite von Verhaltensarten zu generieren. Neben vielen Bewegungsarten sind auch Wachstumsprozesse und sogar imaginäre Evolutionsveränderungen erzeugbar.

- **Holismus:** Modellierung und Animation verschmelzen zu einem holistischen Gesamtkonzept. Die Dichotomie zwischen Form und Bewegung wird dadurch aufgehoben.
- **Relitätsnähe:** Durch Verwendung einer biologisch korrekten Simulationsmethode, die in das Geometriemodell integriert wird, sehen die modellierten Organismen überzeugend realistisch aus, auch die Bewegungen und Wachstumsprozesse erscheinen dem Betrachter glaubwürdig.

4.1.1 Allgemeines zum Verständnis des Quaoarings

Die Quaoaringtechnologie erfordert ein ganzheitliches Verständnis des komplexen Zusammenspiels seiner Unterverfahren, zu denen sowohl konventionelle Verfahren (wie z.B. NURBS, Slerp-Algorithmus, Blob-Funktionen und Objekthierarchien) als auch neue Konzepte und Algorithmen gehören, welche in Kapitel 4.2 bis 4.8 behandelt werden (wie z.B. das biologische Koordinatensystem, die morphischen Felder und die hydrodynamische Biosimulation). Darüber hinaus führt die Zusammensetzung der Unterverfahren zu neuen Effekten, die nicht mittels einer starren analytischen Betrachtungsweise alleine erklärbar sind.

Insbesondere die klassischen Begriffe der Computergraphik wie „Modellierung“, „Simulation“, „Visualisierung“ und „Animation“ haben im Quaoaring infolge der Verwendung eines holistischen Gesamtprozesses eine leicht andere Bedeutung. (siehe dazu auch Kapitel 3.1) Diese Bedeutungsverschiebung wird im Folgenden erklärt:

Wenn in Zusammenhang mit Quaoaring von *Modellierung* gesprochen wird, so hat dies nur wenig mit jenen konventionellen Modellierungsmethoden zu tun, die in Kapitel 2.2 besprochen wurden. Modellierung bedeutet in Quaoaring vielmehr, das Wachstum des Organismus durch richtige Maßgaben und Kraftanwendungen zur rechten Zeit in die gewünschten Bahnen zu lenken. Die morphogenetischen Anweisungen beim Modellierungsprozess werden in zeitlich begrenzt wirkende Restriktions- und Konstriktionseinflüsse übersetzt, die analog einer Entelechie sukzessive die finale Gestalt des Organismus realisieren. Der Modellierer ist somit weniger „Designer“ der Organismusform, sondern vielmehr „Lotse“ des Organismenwachstums.

In der Quaoaringtechnologie wird eine hydropneumatische Biosimulation ausgeführt, welche in eine Geometriepäsentation eingebettet ist. Die Simulation hat Einfluss auf die Ausprägung der Geometriedaten, indem die Oberflächenvertices des Pneus positioniert werden. Rückwirkend beeinflussen die Geometriedaten die Simulation, indem die Vertexpositionen und die für das Shading errechneten Normalenvektoren zur Pnehülle wichtige Daten für die Simulation liefern. Simulation und Visualisierung verschmelzen damit zu einem holistischen Gesamtprozess. Sie können nicht ohne weiteres voneinander getrennt werden. Wenn also im Kontext der Quaoaringtechnologie von *Simulation* oder *Visualisierung* gesprochen wird, sollte stets vor Augen gehalten werden, dass die Dichotomie zwischen Simulation und Visualisierung aufgehoben wurde.

Dieser chimärenhafte Doppelprozess von Simulation und Visualisierung wird jeweils parallel zur Modellierung in Echtzeit ausgeführt. Dies ermöglicht einen interaktiven Modellierungsprozess, in welchem der Modellierer unmittelbar eine visuelle Rückkopplung auf seine Wachstumskorrekturbefehle bekommt und somit direkt sowohl quantitative als auch qualitative Modifikationen vornehmen kann. Die Modellierung stellt somit weniger einen von der Visualisierung abgekoppelten Formerzeugungsprozess dar, wie es bei konventionellen Modellierungsmethoden der Fall ist, sondern sie mutiert vielmehr zu einem sinnlich erfahrbaren Dialog zwischen Mensch und Maschine, der eine autokorrektive intuitive Gestaltung erlaubt. Modellierung und Simulationsvisualisierung stehen damit zueinander in einem engen Zusammenhang.

Können Modellierung, Simulation und Visualisierung beim Quaoaring nicht mehr klar voneinander getrennt werden, so gilt dies auch für die Animation: Da der modellierte Organismus wie in der Natur „wächst“, und meist auch mit automatisch mitwachsenden Organen respektive muskelähnlichen Gebilden ausgestattet ist, die auch die Simulation beeinflussen, wird er automatisch anfangen, sich autonom zu bewegen. Eine explizite Animation des modellierten Organismus wird dadurch obsolet.

Diese automatische und autonome Animation hat den Vorteil, dass keine Extraarbeit für die Bewegungsdefinitionen verlangt wird. Sie hat allerdings auch den Nachteil, dass der Modellierer im Standardmodus keinen direkten Einfluss auf die Bewegungen seines Organismus hat. Sein einmal modellierter Organismus ist frei, diejenigen Bewegungen auszuführen, die gerade für ihn „opportun“ sind.

Diese Unbeeinflussbarkeit der Bewegungen ist vor allem für die Filmproduktion ein Manko, wenn nicht sogar ein Ausschlusskriterium. Hier wird meist gewünscht, dass der Organismus sich streng getreu dem Drehbuch verhält, so dass sinnvolle Filmaufnahmen möglich sind. Aus diesem Grund wurde ein zusätzlicher *Steuerungsmodus* eingeführt. Im Steuerungsmodus wird die Autonomie des virtuellen Wesens unterdrückt. Es muss stattdessen sklavisch Bewegungen vollführen, die aus externen Datenquellen stammen, beispielsweise aus Motion Capturing Devices. Allerdings verschwinden im Steuerungsmodus auch viele der Quaoaringvorteile, wie beispielsweise die biologische Stringenz der Bewegungen oder die schnelle Modellierung mit impliziter Animation.

Ein weiterer Unterschied zu konventionellen Methoden der Organismengenerierung ist, dass die Modellierung des Organismus sich eng an biologische Vorgaben hält. Der Modellierer kann sowohl den Organismus gemäß seiner phylogenetischen Entwicklung erzeugen, er vermag aber auch die Ontogenese mit einzubeziehen. Die Phylogenese und die Ontogenese spannen damit ein zweidimensionales Möglichkeitsfeld auf, auf dem der Modellierer gleichsam navigiert. Da die Erzeugung eines Organismus von Grund auf zu umfangreich wäre, wird die Abspeicherung der Konstruktionsdaten durch baumartige Hierarchien von Deltadateien realisiert. Der Modellierer nimmt zur Realisierung eines Organismus diejenige Tierkonstruktionsdatei, die entweder phylogenetisch oder ontogenetisch der nächste Vorläufer von seinem gewünschten Zielorganismus darstellt.

Möchte er z.B. einen ausgewachsenen Hund modellieren, so könnte er beispielsweise auf die Deltadatei „TomarctusZuWolf“ zurückgreifen und mit einer neuen Deltadatei „WolfZuHund“ zusätzliche Modifikationsanweisungen zur Evolutionslinie hinzufügen, so dass der Wolf zum Hund mutiert. Er könnte aber auch alternativ die Deltadatei „HundeembryoZuWelp“ aufgreifen und ebenfalls in der Datei „WelpZuHund“ geeignete Wachstumsanweisungen geben.

Es ist natürlich auch möglich, Organismen komplett neu mit einer einzigen Deltadatei zu erzeugen oder sich bei den Deltadateien nicht an phylogenetische oder ontogenetische Entwicklungslinien zu halten, doch diese Verfahrensweisen sind nicht anzuraten, da damit die Wiederverwertbarkeit von einmal erstellten Tierkonstruktionen erschwert wird.

4.1.2 Kurzübersicht über das Gesamtverfahren

Im Folgenden soll ein grober Überblick über den konkreten algorithmischen Ablauf des Quaoarings gegeben werden. Viele algorithmische Nebenzweige, die für das Verständnis des Verfahrens nicht essentiell wichtig sind, werden in dieser Grobübersicht nicht behandelt. Dazu gehören u.a. die Buffer-Overflow-Management-Routinen, die Berechnung der Verbindungs-Meshs zwischen zwei adjazenten Pneus, die zeitliche Interpolation zwischen zwei quantitativen Wachstums- oder Evolutionsbefehlen und das Segment-Remeshing.

Zudem ist für das einfachere Verständnis der folgende Algorithmus nur für die Berechnung eines einzelnen Pneus – ohne die Behandlung von eventuell in der Pneu-Hierarchie tiefer stehenden Kinderpneus – skizziert worden. Der tatsächliche Algorithmus ist daher wesentlich größer und vielschichtiger, würde aber an dieser Stelle den Leser mit seiner Komplexität irritieren. Daher soll im Folgenden nur die algorithmische Grundidee behandelt werden:

Zuerst wird ein Ausgangspneu durch ein Polygonnetz mit der Granularität $n*m$ Vertices und dem definierten Zielvolumen erzeugt. Dieser Ausgangspneu ist kugelförmig, da ein idealer undeformierter Pneu immer die Form einer Kugel hat. In einer Quasi-Endlosschleife, die erst terminiert, wenn das Programm beendet wird, werden nun alle weiteren Anweisungsblöcke ausgeführt:

Zuerst erfolgt eine Abfrage, ob Anweisungen für Animations-, Wachstums- und Evolutionsveränderungen vorliegen, falls ja, werden sie in einer Befehlsliste in Form einer Queue gespeichert. Die Anweisungen können dabei entweder aus Tierkonstruktionsdateien stammen, es kann sich aber auch alternativ direkt um interaktive Modellierungsbefehle handeln. Bei Animationsbefehlen können die Anweisungen auch selbstständig von Bewegungspatterns stammen.

Falls nun in der Befehlsliste Instruktionseinträge zu morphischen Veränderungen vorliegen, wird zuerst das Zielvolumen an dasjenige Volumen angepasst, welches dem jeweiligen Wachstumsstadium entspricht. Das Volumen wird dabei über die Zeit linear zwischen dem Volumen der Ausgangstierkonstruktion und dem Volumen der Zieltierkonstruktion interpoliert. Nun werden sequentiell alle Vertices des Skin-Meshs der Pnehülle vorgenommen. Für jeden Vertex werden die lokalen Feldstärken der applizierten „morphischen Felder“ berechnet und aufsummiert. Die morphischen Felder werden später

in Kapitel 4.3.3 erklärt. Der Vertex wird nun in Richtung seiner Normale zur Pnehülle verschoben und zwar mit dem Betrag dieser aufsummierten lokalen Feldstärken.

Danach wird das Volumen des deformierten Pneus neu berechnet. Dafür gibt es 4 verschiedene Verfahren, die alle in Kapitel 4.4 beschrieben werden. Ist nun das ermittelte Volumen um einen nennenswerten Betrag größer als das Zielvolumen, wird der Pneu etwas kleiner reskaliert. Dies geschieht durch relative Verschiebung aller Vertices in Richtung des virtuellen Mittelpunktes, welcher mit dem Mittelpunkt des kugelförmigen Ausgangspneus identisch ist. Ist das ermittelte Volumen um einen nennenswerten Betrag kleiner als das Zielvolumen, wird der Pneu dementsprechend mit einem invers analogen Verfahren etwas größer reskaliert.

Es ist dabei nicht schlimm, wenn nach dieser Reskalierung das Zielvolumen noch nicht mit dem tatsächlichen Volumen übereinstimmt. Schließlich wird der Reskalierungsprozess beim nächsten Schleifendurchlauf wiederholt und das Pneuvolumen wird sich so Durchlauf für Durchlauf asymptotisch dem Zielvolumen nähern. Im schlimmsten Fall hinkt so das Volumen etwas den Anforderungen hinterher, was der Betrachter jedoch aufgrund der Geschwindigkeit in der Praxis nicht bemerkt.

Falls in der Befehlsliste Animationsbefehle vorliegen, werden erst die Positionen der Stützpunkte für eine virtuelle NURBS-Zentrallinie gemäß den Animationsbefehlen neu berechnet. Durch den neuen Verlauf dieser Zentrallinie wird auch indirekt ein spezielles lokales Bezugssystem der Vertices verändert, so dass sie sich ebenfalls mitbewegen. Dieses Bezugssystem wird in Kapitel 4.2 erklärt.

Hier folgt der gesamte Verfahrensablauf nochmals als Pseudo-C-Code:

Kreiere einen kugelförmigen Ausgangspneu durch ein Polygonnetz der Granularität $n*m$

mit $\text{Volumen} = \text{Ausgangsvolumen}$;

Initialisiere $b\text{Volumenkonsistent}$ mit *true* und Zielvolumen mit *Ausgangsvolumen*;

repeat {

Hole Anweisungen und speichere sie in einer Befehlsliste;

// Die Anweisungen stammen entweder indirekt aus einem Modellingskript

// oder direkt vom Modellierungsbefehlen

if (Wachstums- oder Evolutionsveränderungsanweisung in Befehlsliste) {

$\text{Zielvolumen} = \text{Befehlsliste.HoleNeuesZielvolumenDurchInterpolation}()$;

for (int I = 0; I < n; I++)

for (int j = 0; j < m; j++){

```

        bij = 0;
        for(int k=0;k<Anzahl der morphischen Felder bei Vertex[i,j];k++){
            a = berechne Stärke des morphischen Feldes k
                an der Position von Vertex[i,j];
                // für die Feldstärkenberechnung von morphischen
                // Feldern siehe Kapitel 4.3.3
            bij = bij + a;
        }
        Berechne Oberflächennormale an der Position des Vertex[i,j];
        Verschiebe Vertex[i,j] in Richtung dieser Oberflächennormale
            mit dem Betrag bij;
    }
    bVolumenkonsistent = false;
}
if(!bVolumenkonsistent){
    Volumen = Schätze neues Volumen des Pneus ab; // für die Volumenabschätzung
                                                // von Pneus siehe Kapitel 4.4
    if (Volumen > Zielvolumen + Threshold)
        Reskaliere Pneu um den Faktor (Zielvolumen/Volumen)1/3;
    else if (Volumen < Zielvolumen – Threshold)
        Reskaliere Pneu um den Faktor (Volumen/Zielvolumen)1/3;
    else
        bVolumenkonsistent = true;
}
if (Bewegungsanweisung in Befehlsliste){
    Berechne neue Stützpunkte der Zentrallinie gemäß Bewegungsbefehl;
    Berechne neuen Verlauf der Zentrallinie des Pneus anhand der Stützpunkte;
    // Dadurch verändert sich indirekt das biologische Koordinatensystem als lokales
    // Bezugssystem der Pneuvertices, so dass sich der Pneu deformiert und bewegt.
    // Für das Verständnis des biologischen Koordinatensystems
    // und der Zentrallinie siehe Kapitel 4.2
}
    RendereNächstenFrame();
}
until (Programmbeendigungsbefehl in Befehlsliste);

```

4.2 Das biologische Koordinatensystem

4.2.1 Definitionsproblematik bei kartesischen Koordinatensystemen

Geometrisches Modellieren mit der Quaoaringmethode schließt mit ein, dass Punkte, Linien, Flächen oder Räume auf oder in dem Organismus spezifiziert werden müssen. Dies ist eine schwierige Aufgabe, denn im Gegensatz zu starren Objekten verändern Organismen ständig in einer komplexen Weise ihre Form. Diese Formveränderungen können sowohl kurzfristig und vorübergehend sein, z.B. wenn sie atmen, sich krümmen, laufen oder sich in einer anderen Art und Weise bewegen, als auch von langfristiger und irreversibler Natur sein, z.B. wenn sie wachsen oder verletzt werden.

Mit diesen Formveränderungen ändern sich naturgemäß auch die Spezifikationen, zumindest wenn – wie normalerweise üblich – kartesische Koordinaten in einem dreidimensionalen euklidischen Raum benutzt werden. Das Verwenden hierarchisch strukturierter lokaler kartesischer Koordinatensysteme für die Extremitäten kann zwar verhindern, dass die Punkte, Linien, Flächen oder Räume total obsolet werden und schlimmstenfalls an andere Körperstellen wandern, aber auch hier werden die in einem Status des Organismus gemachten Spezifikationen in einer für den Beobachter sichtbaren Weise inkorrekt, wenn der Organismus in einen anderen Bewegungs- bzw. Wachstumsstatus übergeht.

Ein weiterer Nachteil von lokalen oder absoluten kartesischen Koordinatensystemen ist, dass die Spezifikation mathematisch und nicht biologisch motiviert und somit für den Modellierer mit einer rein biomedizinischen Ausbildung nicht gerade effektiv geschweige denn intuitiv ist. Als abschreckend negatives Beispiel für diesen Mangel an Intuitivität könnte z.B. folgende Spezifikation angeführt werden:

„Gesucht wird dasjenige Organ, welches um den Punkt (0,35; 0,10; 0,15) im rechtshändigen Hauptbezugskoordinatensystem der Skalierungseinheit „Meter“ mit dem Ursprung im Körperschwerpunkt des aufrecht gerade stehenden Autors liegt.“

Der Leser wird beim Lesen dieses Definitionsrätsels bestimmt zustimmen, dass es nicht intuitiv ersichtlich ist, dass damit das Herz gemeint sein soll.

Ähnliches gilt auch analog für Spezifikationen, die in anderen, wie z.B. zylindrischen Koordinatensystemen, gemacht worden sind. Auch hier sind Definitionen nicht invariant zum Status des jeweiligen Organismus.

Um die Spezifikation von Punkten, Linien, Flächen oder Räume zu erleichtern und die Berechnungen, welche im Quaoaringansatz angewendet werden, eleganter zu beschreiben, wird ein nonkartesisches Koordinatensystem eingeführt, welches auf biologische Objekte zugeschnitten ist: das *biologische Koordinatensystem* oder kurz *BKS*. Dies beinhaltet, dass eine Definition innerhalb des biologischen Koordinatensystems mit Einschränkungen invariant zum spezifischen Bewegungs- und Wachstumsstatus der hydropneumatischen Körpereinheit ist. Die Einschränkungen beziehen sich darauf, dass es vollkommen ausreichend ist, wenn eventuelle Änderungen sowohl für den Modellierer als auch für den Beobachter hinreichend klein sind, so dass sie sich nicht störend bemerkbar machen.

Die Invarianz wird auch während des Wachstumsprozesses, komplexen Bewegungsabläufen und sogar evolutionären Übergängen zwischen zwei in einer Evolutionslinie liegenden Organismen aufrechterhalten.

4.2.2 Umwandlung biologischer in kartesische Koordinaten

Es wird postuliert, dass jede hydropneumatische Einheit eine fiktive interne Achse besitzt, die *Zentrallinie* bzw. englisch *center line*, die mit einem Start- und Endpunkt ausgestattet ist, welche durch Intersektion der Achse mit der Hülle ermittelt wird (siehe Abbildung 56). Um die Zentrallinie zu definieren, wird NURBS-Interpolation verwendet. Jeder Kontrollpunkt der NURBS-Kurve ist mit einem Normalenvektor ausgestattet, welcher die Richtung des ventralen Meridians ausgehend von der Zentrallinie definiert, dem *Ventralvektor* oder englisch *up vector*. Durch Interpolation dieser Ventralvektoren unter Zuhilfenahme von Quaternionen und dem Slerp-Algorithmus [Shoemake 1985] lassen sich nun beliebig viele intermediäre Ventralvektoren an jeder Stelle der Zentrallinie errechnen.

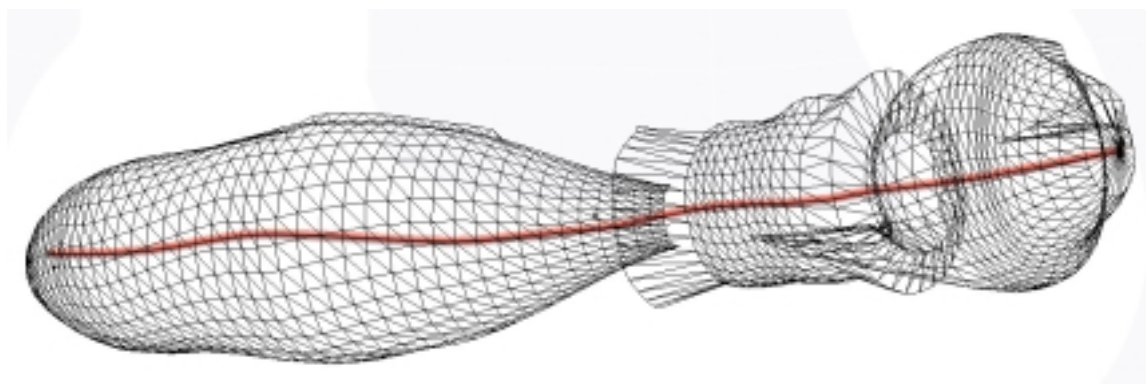


Abbildung 56: Drahtgittermodell eines Pneus mit seiner Zentrallinie

Das Drahtgittermodell des Hauptpneus eines Enteropneusten (wurmähnlicher Organismus, welcher im Meeressediment lebt) zusammen mit seiner Zentrallinie (rot)

Die biologischen Koordinaten eines Punktes T können nun durch einen Dreiertupel $\tau = (f_R; f_A; f_H)$, dem *Biovektor*, beschrieben werden.

Um die kartesischen Koordinaten zu ermitteln, die durch einen Biovektor τ gegeben sind, wird bei Hilfspunkt \vec{q} der Zentrallinie begonnen. Der Parameter f_R gibt dabei die laterale fraktionale Position von \vec{q} entlang der Zentrallinie an, wobei $f_R = 0$ als caudaler Startpunkt definiert ist und $f_R = 1$ dementsprechend als cranialer Endpunkt.

Nun wird ein Hilfsvektor \vec{u} berechnet, in dem der interpolierte Ventralvektor, bei \vec{q} genommen, gegen den Uhrzeigersinn um den Winkel f_A (rad) rotiert und danach in einer Weise skaliert wird, dass seine Länge identisch mit der Entfernung zwischen Zentrallinie und Hülle ist – oder in anderen Worten zwischen Punkt \vec{q} und demjenigen gedachten Punkt, der durch Intersektion von der unendlichen Verlängerung von \vec{u} nach seiner Rotation mit der Hülle entsteht. Die kartesischen Koordinaten \vec{t} des Biovektors τ können nun durch folgende Formel F15 berechnet werden:

$$\text{F15} \quad \vec{t} = \vec{q} + f_H \cdot \vec{u}$$

Der Wert von f_H , gibt nun an, in welchem Bereich des Organismus bezüglich einer Innen-Außen-Skalierung der Punkt liegt:

- Falls $-1 < f_H < 1$ ist, liegt \vec{t} im Innenraum des Pneus und damit auch im Innenraum des Organismus, im Spezialfall $f_H = 0$ sogar direkt auf der Zentrallinie
- Bei $|f_H| = 1$ liegt \vec{t} direkt auf der Hülle.
- Im Falle $|f_H| > 1$ liegt \vec{t} außerhalb des Pneus.

Durch Negation können die lateralen Körperantipoden ermittelt werden, d.h. falls $-f_H = f'_H$ ist, folgt daraus automatisch, dass f_H die longitudinale Körperantipode von f'_H ist und umgekehrt.

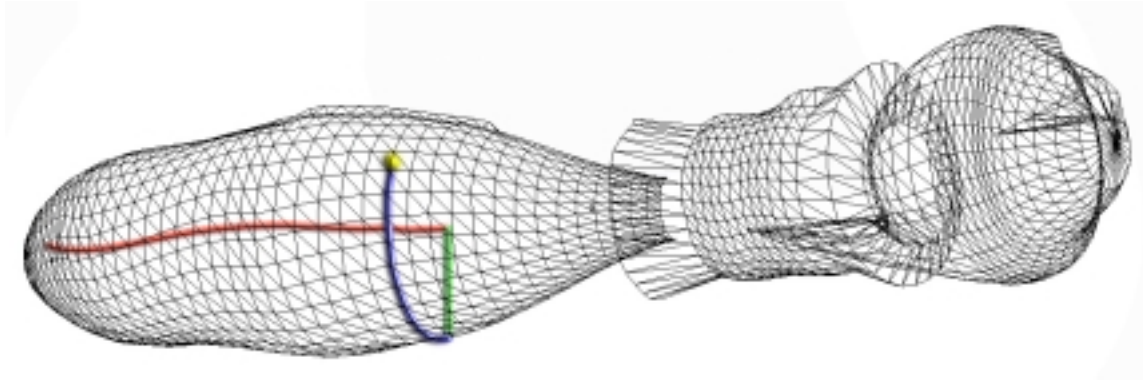


Abbildung 57: Skizze der Komponenten eines Biovektors

Das Drahtgittermodell eines Enteropneusten-Pneus, zusammen mit den drei Komponenten eines Biovektors: $f_R := 1/3$ (rot), $f_A := 2/3\pi$ (blau) und $f_H := 1$ (grün) einer Position (gelb).

Biologische Koordinaten bieten – neben der erwähnten Statusinvarianz – weitere Vorteile für den Modelliervorgang, die Simulationsberechnung und den Einsatz in der Medizin, wie im nächsten Kapitel aufgezeigt wird:

4.2.3 Biologisches versus zylindrisches Koordinatensystem

Es bestehen auffällige Ähnlichkeiten zwischen biologischen Koordinaten auf der einen und zylindrischen Koordinaten auf der anderen Seite, so dass es sich lohnt, einen Vergleich anzustellen:

Beide Koordinaten sind jeweils durch einen Dreier-Tupel gekennzeichnet, von denen ein Tupel die Position entlang einer gekümmten Linie im Raum, ein Tupel den Abstand von dieser Linie und ein Tupel die Position des Punktes radial um diese Linie herum kennzeichnet. Damit erschöpfen sich aber die Gemeinsamkeiten zwischen den beiden Koordinatensystemen. Die Unterschiede zwischen beiden Systemen sind folgende:

Bei zylindrischen Koordinaten existiert keine Ventralvektoren, so dass der Ausgangspunkt der Drehung immer orthogonal zu einer Koordinatenachse des euklidischen Raumes ist.

Auch die Rotationsachse ist – zumindest bei den meisten – zylindrischen Koordinatensystemen nicht orthogonal zur Sweep-Linie, sondern orthogonal zu einer Koordinatenachse des euklidischen Raumes definiert.

Doch der Hauptunterschied zwischen beiden Systemen ist, dass derjenige Tupel, der für den Abstand des definierten Raumpunktes von der Sweep-Linie bzw. Zentrallinie zuständig ist, bei zylindrischen Koordinaten absolut und bei biologischen Koordinaten relativ ist. Das heißt in anderen Worten, während bei zylindrischen Koordinaten einfach der absolute kartesische Abstand von der Sweep-Linie genommen wird, wird bei biologischen Systemen der relative Abstand zwischen Zentrallinie und Skin-Mesh zur Abstandsberechnung herangezogen, was zur Folge hat, dass die Position von der Form des Skin-Meshs abhängig ist – und somit beim Quaoaring indirekt von hydropneumatischen Gesetzmäßigkeiten. Auf diese Weise entsteht ein implikatorischer Zirkelschluss: Die biologischen Koordinaten helfen die Form des Skin-Meshs zu definieren und die Form des Skin-Meshs ist wiederum für die Interpretation der biologischen Koordinaten unabdingbar.

Ähnliches gilt für denjenigen Tupel, der für die Positionierung des Raumpunktes entlang der Aufbau- bzw. Zentrallinie zuständig ist. Es wird ebenfalls bei zylindrischen Koordinaten absolut und bei biologischen Koordinaten relativ zur Hüllform angegeben, wodurch sich letzten Endes analoge Kreisimplikationen ergeben.

Infolge dieser implikatorischen Zirkelschlüsse resultiert beim Quaoaring eine schwer zu verstehende wechselseitige Dynamik zwischen Skin-Mesh und biologischen Koordinaten, bei der sich die Punktdefinition letzten Endes asymptotisch auf einen fixen euklidischen Raumpunkt hin einpendelt.

Eine bildhafte Vergleichsmetapher veranschaulicht den Unterschied zwischen beiden Koordinatensystemen sehr deutlich und einsichtig: Das zylindrische Koordinatensystem verhält sich zum biologischen Koordinatensystem wie ein Rad zu einem Reifen.

So gibt es zwischen einem Rad und einem Reifen auffällige Ähnlichkeiten, so dass sie von Kleinkindern oft verwechselt werden: Beide sind rund, sie verlaufen radial um eine Achse und sie dienen beide zum Fahren. Trotzdem ist ein Rad kein Reifen, denn letzterer unterliegt u.a. den Gesetzen der Aeropneumatik und machte die Entwicklung des modernen Automobils erst möglich.

Auch zwischen dem zylindrischen Koordinatensystem und biologischen Koordinatensystem existieren auffällige Ähnlichkeiten: Beide sind rund, sie verlaufen radial um eine zentral verlaufende Linie und sie dienen beide zum Definieren von Punkten im Raum. Trotzdem ist eine zylindrische Koordinate keine biologische Koordinate, denn letztere unterliegt u.a. den Gesetzen der Hydropneumatik und macht die Quaoaring-Technologie erst möglich.

4.2.4 Beziehung zwischen anatomischen Termini und Biovektoren

Eine Positionsdefinition, die durch einen Biovektor gegeben ist, ist für den Modellierer wesentlich anschaulicher als eine entsprechende Definition mit lokalen kartesischen Koordinaten, da die drei Parameter des Biovektors unmittelbar in gängige biologisch-anatomische Bezeichnungen umgesetzt werden können. Biologen und Mediziner können so direkt und intuitiv mit ihren Fachtermini hantieren, ohne dass sie dafür ein besonderes dreidimensionales Vorstellungsvermögen brauchen und ohne dass sie sich dafür in die mathematische bzw. informationstechnische Terminologie einarbeiten.

Die allgemein übliche Referenzierung von anatomischen Positionen und Bewegungen in der Medizin und in der Biologie wird durch die *Nomina Anatomica Veterinaria* (kurz NAV) des *International Committee on Veterinary Gross Anatomical Nomenclature* standardisiert und benutzt u.a. Begriffe wie inferior, superior, caudal, cranial, proximal, distal, central, peripheral, superficial, profund, dextral, sinistral, median, intern, extern, longitudinal, orthogonal, anterior, posterior, Axis Medialis und Planum Medianum. [ICVGAN 2004] Eine davon nur in Nuancen abweichende Terminologie benutzt die *International Federation on Anatomical Associations*, welche die *Terminologia Anatomica*, kurz TA, herausgibt [IFAA 2003] Diese Begriffe aus der NAV bzw. der TA werden elegant durch die Werte der drei Parameter des Biovektors repräsentiert:

Der Parameter f_R gibt direkt die Position entlang der *logitudinalen* Körperachse an. Beim Rumpf bezeichnen kleine Werte von f_R eine *caudale*, hohe Werte dementsprechend eine *craniale* Lage. Bei den Extremitäten stehen kleine Werte von f_R für eine *proximale*, und hohe Werte für eine *distale* Position. Ein Organ, Organteil oder Körperteil mit einem niedrigen f_R Wert kann mit dem Attribut *inferior*, mit einem hohen f_R -Wert dagegen mit *superior* ausgezeichnet werden.

Der Parameter f_A bezeichnet die Rotation um die *Axis Medialis*. Unter der Voraussetzung, dass $f_H > 0$ ist, ist die Position *dextral*, falls $0 < f_A < \pi$ ist, bei $-\pi < f_A < 0$ dementsprechend *sinistral*. Bei negativen Werten von f_H gilt das Gleiche, jedoch genau invertiert. Ist $f_A = 0$ oder $f_A = \pi$, so liegt der Punkt auf dem *Planum Medianum*.

Der Parameter f_H definiert die Position auf der *central-peripheralen* Skala. Für $|f_H| = 1$ ist die Position *superficial*, für $|f_H| < 1$ *profund*, für $|f_H| \leq 1$ *intern* für $|f_H| > 1$ *extern*. Im Falle

$f_A = 0$ gibt f_H bei negativen Werten die *dorsale* bzw. bei positiven die *ventrale* Position auf dem *Planum Medianum* an. Ein Organ, Organteil oder Körperteil mit einem niedrigen f_H Wert kann mit dem Attribut *posterior*, mit einem hohen f_H –Wert dagegen mit *anterior* ausgezeichnet werden.

Auch die Operationen, die durch die relativen Bewegungsbezeichnungen der anatomischen Terminologie definiert werden, können durch Modifikation eines Parameters des Biovektors einfach und elegant ausgeführt werden, *Adduktion* entspricht einer Erhöhung des f_H -Parameters, *Abduktion* dementsprechend einer Verminderung und *Rotation* – im rein anatomischen Sinne – kann durch eine Werteveränderung von f_A ausgeführt werden.

Diese direkte, streng mathematische Verknüpfung zwischen anatomischen Termini und Biovektoren lässt sich u.a. für boolesche Funktionen benutzen, die für anatomische Entscheidungen und Visualisierungen hilfreich sind und nicht nur auf konkrete Quaoaring-Implementierungen beschränkt sein müssen.

Für den Biovektor $\tau = (f_R; f_A; f_H)$, welcher eine Position definiert, gelten folgende pneuunabhängigen booleschen Bereichsfunktionen $\Xi_B(\tau)$ für das Bereichsattribut B : (Leseart ist „ τ hat das Bereichsattribut B “)

$$\Xi_{\text{Superficial}}(\tau) = \begin{cases} 1 & \forall f_H = 1 \\ 0 & \forall f_H \neq 1 \end{cases}$$

$$\Xi_{\text{Median}}(\tau) = \begin{cases} 1 & \forall f_H = 0 \\ 0 & \forall f_H \neq 0 \end{cases}$$

$$\Xi_{\text{Intern}}(\tau) = \begin{cases} 1 & \forall f_H < 1 \\ 0 & \forall f_H \geq 1 \end{cases}$$

$$\Xi_{\text{Extern}}(\tau) = \begin{cases} 1 & \forall f_H > 1 \\ 0 & \forall f_H \leq 1 \end{cases}$$

Für den Biovektor $\tau = (f_R; f_A; f_H)$, welcher eine Position innerhalb eines Pneus P definiert, gelten folgende pneusensitiven booleschen Bereichsfunktionen $\Xi_B(\tau, P)$ für den Bereich B : (Leseart ist „ τ liegt im Bereich B im Pneu P “)

$$\Xi_{Axis_Medialis}(\tau, P) = \begin{cases} 1 & \forall f_H = 0 \wedge \tau \in P \\ 0 & \forall f_H \neq 0 \vee \tau \notin P \end{cases}$$

$$\Xi_{Planum_Medianum}(\tau, P) = \begin{cases} 1 & \forall (\sin(f_A) = 0 \vee f_H = 0) \wedge \tau \in P \\ 0 & \forall (\sin(f_A) \neq 0 \wedge f_H \neq 0) \vee \tau \notin P \end{cases}$$

Analog zu diesen Bereichsfunktionen können auch boolesche Vergleichsfunktionen aufgestellt werden, welche die anatomischen Positionen zweier Biovektoren miteinander vergleichen. Gegeben sei dazu ein Pneu P , welcher Element der Menge der Rumpfpneus M_R , der Organpneus M_O , der Extremitätenpneus M_E oder der Kanalpneus M_K sein kann. Für die zwei Biovektoren $\tau = (f_R; f_A; f_H)$ und $\tau' = (f_R'; f_A'; f_H')$, welche jeweils eine Position innerhalb von P angeben, gelten dann folgende boolesche Funktionen $\Xi_A(\tau, \tau', P)$ für das anatomische Attribut „ A “:

(Die Lesart ist „Die Position gegeben durch den Biovektor τ hat das anatomische Attribut „ A “ relativ zur Position gegeben durch den Biovektor τ' im Pneu P “)

$$\Xi_{Inferior}(\tau, \tau', P) = \begin{cases} 1 & \forall f_R < f_R' \wedge \tau \in P \wedge \tau' \in P \\ 0 & \forall f_R \geq f_R' \vee \tau \notin P \vee \tau' \notin P \end{cases}$$

$$\Xi_{Superior}(\tau, \tau', P) = \begin{cases} 1 & \forall f_R > f_R' \wedge \tau \in P \wedge \tau' \in P \\ 0 & \forall f_R \leq f_R' \vee \tau \notin P \vee \tau' \notin P \end{cases}$$

$$\Xi_{Caudal}(\tau, \tau', P) = \begin{cases} 1 & \forall f_R < f_R' \wedge \tau \in P \wedge \tau' \in P \wedge P \in M_R \\ 0 & \forall f_R \geq f_R' \vee \tau \notin P \vee \tau' \notin P \vee P \notin M_R \end{cases}$$

$$\Xi_{Cranial}(\tau, \tau', P) = \begin{cases} 1 & \forall f_R > f_R' \wedge \tau \in P \wedge \tau' \in P \wedge P \in M_R \\ 0 & \forall f_R \leq f_R' \vee \tau \notin P \vee \tau' \notin P \vee P \notin M_R \end{cases}$$

$$\Xi_{Proximal}(\tau, \tau', P) = \begin{cases} 1 & \forall f_R < f_R' \wedge \tau \in P \wedge \tau' \in P \wedge P \in M_E \\ 0 & \forall f_R \geq f_R' \vee \tau \notin P \vee \tau' \notin P \vee P \notin M_E \end{cases}$$

$$\Xi_{Distal}(\tau, \tau', P) = \begin{cases} 1 & \forall f_R > f_R' \wedge \tau \in P \wedge \tau' \in P \wedge P \in M_E \\ 0 & \forall f_R \leq f_R' \vee \tau \notin P \vee \tau' \notin P \vee P \notin M_E \end{cases}$$

$$\Xi_{Longitudinal}(\tau, \tau', P) = \begin{cases} 1 & \forall f_R \neq f_R' \wedge f_A = f_A' \wedge f_H = f_H' \wedge \tau \in P \wedge \tau' \in P \\ 0 & \forall f_R = f_R' \vee f_A \neq f_A' \vee f_H \neq f_H' \vee \tau \notin P \vee \tau' \notin P \end{cases}$$

$$\Xi_{Orthogonal}(\tau, \tau', P) = \begin{cases} 1 & \forall f_R = f_R' \wedge (f_A \neq f_A' \vee f_H \neq f_H') \wedge \tau \in P \wedge \tau' \in P \\ 0 & \forall f_R \neq f_R' \vee (f_A = f_A' \wedge f_H = f_H') \vee \tau \notin P \vee \tau' \notin P \end{cases}$$

$$\Xi_{Dextral}(\tau, \tau', P) = \begin{cases} 1 & \forall (f_H \sin(f_A) > f_H' \sin(f_A')) \wedge \tau \in P \wedge \tau' \in P \wedge P \in M_R \\ 0 & \forall (f_H \sin(f_A) \leq f_H' \sin(f_A')) \vee \tau \notin P \vee \tau' \notin P \vee P \notin M_R \end{cases}$$

$$\Xi_{Sinisral}(\tau, \tau', P) = \begin{cases} 1 & \forall (f_H \sin(f_A) < f_H' \sin(f_A')) \wedge \tau \in P \wedge \tau' \in P \wedge P \in M_R \\ 0 & \forall (f_H \sin(f_A) \geq f_H' \sin(f_A')) \vee \tau \notin P \vee \tau' \notin P \vee P \notin M_R \end{cases}$$

$$\Xi_{Anterior}(\tau, \tau', P) = \begin{cases} 1 & \forall (f_H \cos(f_A) > f_H' \cos(f_A')) \wedge \tau \in P \wedge \tau' \in P \\ 0 & \forall (f_H \cos(f_A) \leq f_H' \cos(f_A')) \vee \tau \notin P \vee \tau' \notin P \end{cases}$$

$$\Xi_{Posterior}(\tau, \tau', P) = \begin{cases} 1 & \forall (f_H \cos(f_A) < f_H' \cos(f_A')) \wedge \tau \in P \wedge \tau' \in P \\ 0 & \forall (f_H \cos(f_A) \geq f_H' \cos(f_A')) \vee \tau \notin P \vee \tau' \notin P \end{cases}$$

$$\Xi_{Ventral}(\tau, \tau', P) = \begin{cases} 1 & \forall (f_H \cos(f_A) > f_H' \cos(f_A')) \wedge \tau \in P \wedge \tau' \in P \wedge P \in M_R \\ 0 & \forall (f_H \cos(f_A) \leq f_H' \cos(f_A')) \vee \tau \notin P \vee \tau' \notin P \vee P \notin M_R \end{cases}$$

$$\Xi_{Dorsal}(\tau, \tau', P) = \begin{cases} 1 & \forall (f_H \cos(f_A) < f_H' \cos(f_A')) \wedge \tau \in P \wedge \tau' \in P \wedge P \in M_R \\ 0 & \forall (f_H \cos(f_A) \geq f_H' \cos(f_A')) \vee \tau \notin P \vee \tau' \notin P \vee P \notin M_R \end{cases}$$

$$\Xi_{Central}(\tau, \tau', P) = \begin{cases} 1 & \forall |f_H| < |f_H'| \wedge \tau \in P \wedge \tau' \in P \\ 0 & \forall |f_H| \geq |f_H'| \vee \tau \notin P \vee \tau' \notin P \end{cases}$$

$$\Xi_{Peripheral}(\tau, \tau', P) = \begin{cases} 1 & \forall |f_H| \geq |f_H'| \wedge \tau \in P \wedge \tau' \in P \\ 0 & \forall |f_H| < |f_H'| \vee \tau \notin P \vee \tau' \notin P \end{cases}$$

4.2.5 Definition von Flächen und Räumen im BKS

Analog zur Definition von Räumen mit Hilfe zweier kartesischer Koordinaten können auch Räume mittels zweier biologischer Koordinaten aufgespannt werden. Der Vorteil dieser Räume im BKS ist, dass sie sich ebenfalls invariant zum Bewegungs- und Wachstumsstatus des Organismus verhalten. Wird einem Organismus beispielsweise in einem frühen Entwicklungsstatus ein interner Raum für ein Organ zugewiesen, so wächst und entwickelt sich der Raum – und mit diesem das Organ – ebenfalls in einer sinnfälligen Weise zusammen mit dem Organismus.

Räume können auf mehrere Weisen im biologischen Koordinatensystem aufgespannt werden. In der Praxis hat es sich bewährt, einen Biovektor für die Definition eines Mittelpunktes und den anderen Biovektor als richtungsabhängigen Radiusvektor zu verwenden, da damit automatisch organisch anmutende Strukturen entstehen.

Ein so definierter *Bioraum* in einem Pneu P kann auch als Sechsertupel $S = (c_R; c_A; c_H; r_R; r_A; r_H)$ beschrieben werden, wobei die ersten drei Parameter den Mittelpunkt und die drei letzten Parameter die dreidimensionale Ausdehnung im biologischen Koordinatensystem beschreiben. Auf diese Weise steht r_R für den fraktionierten longitudinalen Radius relativ zur Gesamtlänge des Pneus, r_A für die halbe orthogonale Ausdehnung (in rad) und r_H für die radiale Länge relativ zum Abstand zwischen Zentrallinie und Hülle entlang derjenigen Normalen zur Zentrallinie, welche den Mittelpunkt des Bioraumes durchstößt.

Analog zu Räumen können auch Flächen im BKS definiert werden, wobei diese *Bioflächen* als Spezialfall von Bioräumen angesehen werden können, bei denen einer der drei r -Parameter gleich Null ist und somit eine Dimension des lokalen biologischen Koordinatensystems entfällt. Bioflächen werden hauptsächlich für die Definition von Teilarealen der Pneuhülle gebraucht. Eine solche Biofläche auf der Pneuhülle wird durch einen Bioraum mit $c_H = 1$ und $r_H = 0$ definiert.

4.2.6 Kreuzungen im BKS

Das biologische Koordinatensystem wird bei der Quaoaringtechnologie auf alle Spezies angewandt. Somit verwendet jeder Organismus dasselbe Referenzsystem, was es erlaubt, modellierte Körperteile und Organe durch einfaches Kopieren der entsprechenden Positions-, Flächen-, Raum- und Auslenkungsdefinitionen von einem Organismus auf einen anderen Organismus zu übertragen. Der Transfer dieser biologischen Teilstrukturen

kann auch zwischen stark unterschiedlichen Organismen erfolgen. Zudem braucht der Modellierer sich nicht mehr wie bisher um unterschiedliche Gesamtgrößendimensionen der Organismen, unterschiedliche Körperproportionen und anatomische Verschiebungen zu kümmern.

Da das BKS sowohl der anatomischen Nomenklatur als auch der intuitiven menschlichen Empfindung entspricht (siehe Kapitel 4.2.4), werden die kopierten biologischen Strukturen stets sinnfällig eingefügt und automatisch auf die Dimension des Zielwesens skaliert. Die Strukturen verhalten sich damit zumindest optisch und biomechanisch korrekt.

Die Schöpfung von gekreuzten Mischwesen aus verschiedenen Spezies ist damit auf einfache Weise möglich. Dabei können nicht nur miteinander verwandte Spezies miteinander gekreuzt werden, sondern es sind auch Fabelwesen aus genetisch inkompatiblen Tiergattungen kreierbar. Ein Beispiel einer solchen Chimäre aus Nilpferd, Kaulquappe und Biene zeigt Abbildung 58:



Abbildung 58: Beispiel für die einfache Erzeugung von Mischwesen mit dem BKS

Die biologisch korrekte Parametrisierung im biologischen Koordinatensystem erlaubt es, mit der Quaoaringtechnologie Mischwesen aus verschiedenen Tieren herzustellen. Hier wurde durch einfache Parametermischung unterschiedlicher Spezies eine gekreuzte Chimäre aus Nilpferd (Kopf und Rumpf), Kaulquappe (Flossen, Finne und Schwanz) sowie Biene (Fell) generiert.

4.3 Deformation hydropneumatischer Einheiten

4.3.1 Erster Ansatzversuch mit einem Feder-Masse-Dämpfer-Modell

Die Form einer idealen hydropneumatischen Einheit, auf welche keinerlei Kräfte einwirken, ist eine Kugel. Andere Formen können durch Krafteinwirkungen auf diesen kugelförmigen Pneu erzielt werden. Die Simulation dieser Verformung kann mit Hilfe eines modifizierten Feder-Masse-Dämpfer-Systems geschehen, das die Pnehülle simuliert. Ein solches Verfahren hätte den Vorteil der Vielseitigkeit und der analytischen Realitätsnähe, so dass es nicht nur zur Visualisierung, sondern darüber hinaus auch als analytische Simulationsmethode angewendet werden könnte. Allerdings haben Feder-Masse-Dämpfer-Systeme – zumindest zum Zeitpunkt der vorliegenden Arbeit – noch den Nachteil, dass sie nur kleine Entitäten in Echtzeit simulieren können (siehe Kapitel 2.4.2.1). Zwar wäre die interaktive Echtzeitsimulation eines einfachen Pneus mit ca. 5000 bis 10.000 Massepunkten durchaus im Bereich des Machbaren, die Visualisierung komplexer Organismen mit vielen hierarchisch angeordneten Pneus und einer Gesamtanzahl von weit über einer Million Massepunkten ist jedoch mit Feder-Masse-Dämpfer-Systemen unmöglich. Dies könnte sich allerdings schon in recht naher Zukunft ändern, wobei aber auch vermutlich in fast gleichem Maße die Ansprüche an Organismenvisualisierungen steigen werden.

Ein zweiter weitaus gravierender Nachteil von Feder-Masse-Dämpfer-Systemen ist ihre mangelnde Robustheit hinsichtlich zyklischer Interdependenzen, die zu unerwünschten Rückkopplungseffekten und Aufschaukelungsprozessen führen können. Diese Artefakte sind bei einzelnen Pneus zwar mit zusätzlichen Dämpfungselementen in den Griff zu bekommen. Allerdings verhält sich der Pneu dann nicht mehr realistisch.

Sind diese Rückkopplungseffekte bei einzelnen Pneus noch beherrschbar, so versagen diese Methoden bei großen Pneuanordnungen, die mechanisch zyklisch interdependent miteinander verkettet sind.

Ein einfaches und zugleich häufiges Beispiel für eine solche zyklische Verkettung wäre z.B. die mechanisch bidirektionale Interaktion zwischen dem Pneu, der einen Muskel simuliert, und einer übergeordneten hierarchischen hydropneumatischen Einheit, welche die darüber liegende Hautschicht repräsentiert. In einem solchen System würde eine Flexion des Muskelpneus die Haut deformieren und umgekehrt würde die Spannung des Hautpneus die Deformation des Muskelpneus – in einem kleinen, jedoch für Aufschaukelungsprozesse durchaus ausreichenden Ausmaß – modifizieren. Eine

Pilotstudie mit einem implementierten Prototyp einer Pneusimulation, welcher ein modifiziertes Feder-Masse-Dämpfer-System benutzte, zeigte, dass dies nicht nur theoretische Probleme sind.

4.3.2 Zweiter realisierter Ansatz mit morphischen Feldern

Da sich der Ansatz mit Feder-Masse-Dämpfer-Systemen als nicht praktikabel herausgestellt hat, wurde in der Folge nach alternativen Methoden gesucht, die sich gegenüber Aufschaukelungsprozessen robust verhalten und zugleich für interaktive Echtzeitvisualisierungen ausreichend performant sind.

Diese Performanz und Robustheit kann dabei nur erzielt werden, wenn nicht einzelne Massepunkte berechnet werden, sondern eine komplette Deformation als Gesamtheit berechnet wird. Bei der Recherche wurden allerdings keine Einzelformeln bzw. einzelne Gleichungssysteme gefunden, die eine exakte hydrodynamische Simulation von komplexen Pneuverformungen erlauben würden. (Einerseits fehlen geeignete hydrodynamische Formeln, andererseits würde eine solche Berechnung vermutlich zu unperformant sein, um eine Berechnung in Echtzeit zu erlauben.)

Es ist daher notwendig, eine performante Approximation zu verwenden, die der Realität möglichst nahe kommt. Dazu müssen erst einmal die Eigenschaften der Deformation empirisch näher betrachtet werden und eine Funktion gefunden werden, die möglichst viele dieser Attribute ebenfalls aufweist. Eine Deformation auf einem Pnehüllenbereich, der durch eine Biofläche definiert ist, hat folgende Eigenschaften:

- Die Deformation, die mit dem Zentrum der Biofläche korrespondiert, hat eine bestimmte Auslenkungshöhe ω , die ungleich Null ist.
- An den Rändern der Biofläche ist die Auslenkung gleich Null.
- Der Verlauf jedes beliebigen Querschnittes durch den Deformationswulst ist stetig.
- Die erste Ableitung des Verlaufes jedes beliebigen Querschnittes durch den Deformationswulst ist stetig.
- Die Deformation ist spiegelsymmetrisch entlang derjenigen longitudinalen Achse, die durch das Zentrum der Deformation geht.
- Die Deformation ist ebenfalls spiegelsymmetrisch entlang derjenigen orthogonalen Achse, die durch das Zentrum der Deformation geht.

- Die erste Ableitung des Verlaufes jedes beliebigen central-peripheralen Querschnittes durch den Deformationswulst ist sowohl im Zentrum als auch an den Rändern der Biofläche gleich Null.

Interessanterweise besitzen auch zweidimensionale Blob-Funktionen analoge Eigenschaften, wenn jeweils das Zentrum der Biofläche durch 0, die Ränder der Biofläche durch |1|, die longitudinale Achse durch die X-Achse, die orthogonale Achse durch die Y-Achse und ω durch 1 ersetzt wird.

Somit ist es naheliegend, die Deformationswulste durch zweidimensionale Blobfunktionen anzunähern, die im biologischen Koordinatensystem auf die Biofläche skaliert werden. Es existieren verschiedene Blob-Funktionen. Es wurde die Blob Funktion F16 von Baranoski und Rokne ausgewählt, da sie einerseits weniger Gleitkommaoperationen benötigt als andere Blob-Funktionen und andererseits einen speziellen Toggeling-Parameter d beinhaltet, mit dem die „Spitzheit“ des Deformationswulstes modifiziert werden kann. Wird d klein gewählt, so entstehen stumpfe, dickwulstige Deformationen, bei einem großen Wert von d dagegen spitze, stachelförmige Deformationen. [Baranoski & Rokne 2001]

$$\mathbf{F16} \quad B(r, d) = \begin{cases} (r^2 - 1)^2 / (1 + dr^2) & |r| < 1 \\ 0 & |r| \geq 1 \end{cases}$$

Die zu F16 entsprechende zweidimensionale Blob-Funktion F17 wird durch Einsetzen einer einfachen euklidischen Distanzfunktion erzeugt:

$$\mathbf{F17} \quad B_2(x, y, d) = \begin{cases} (x^2 + y^2 - 1)^2 / (1 + d(x^2 + y^2)) & |x^2 + y^2| < 1 \\ 0 & |x^2 + y^2| \geq 1 \end{cases}$$

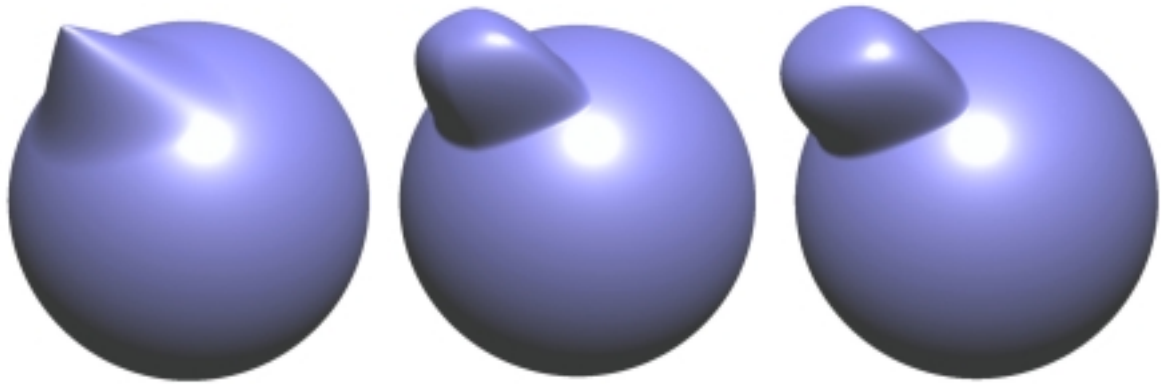


Abbildung 59: Auswirkung des Parameters d bei morphischen Feldern

Auf drei kugelförmigen Pneus wurden gleich große morphische Felder appliziert. Lediglich der Toggling-Parameter d wurde verändert. Links: $d=1$, Mitte: $d=0,1$ Rechts: $d=0,01$

4.3.3 Die Definition und Berechnung von morphischen Feldern

Um eine Pneudefomation, welche durch eine spezifische Kraft hervorgerufen wird, zu simulieren, definiert der Modellierer zuerst eine Biofläche, welche diejenige Region auf der Hülle repräsentiert, auf der die Kraft einen sichtbaren Deformationseffekt haben soll. Dann wird der Kraftbetrag angegeben. Alternativ zu dieser Kraftspezifikation kann der Modellierer auch direkt die maximale Auslenkung des Deformationswulstes ω spezifizieren, welche aus der Kraftanwendung resultieren würde. Die maximale Deformationsauslenkung wird dabei relativ zum aktuellen Abstand zwischen Zentrallinie und Hülle angegeben – gemessen entlang demjenigen Normalenvektor der Zentrallinie, welcher den Mittelpunkt der Biofläche durchstößt.

Würde beispielsweise der Abstand eines hypothetischen Punktes P auf der Hülle zur Zentrallinie 5 Einheiten betragen, dann hätte $\omega = 1$ nach der Kraftanwendung im Punkt P zur Folge, dass der Punkt 10 Einheiten von der Zentrallinie entfernt wäre. Bei $\omega = 0$ würde der Pneu dagegen nicht deformiert werden. Ein negatives ω mit $-1 < \omega < 0$ zöge den Punkt näher an die Zentrallinie heran bis er bei $\omega = -1$ direkt auf der Zentrallinie liegen würde. Bei einem $\omega < -1$ würde die Zentrallinie die Pnehülle durchstoßen.

Die Größe ω kann als Modellierungsparameter angesehen werden und daher direkt durch den Modellierer angegeben werden, sie kann aber auch alternativ in andere Modellierungsparameter aufgebrochen werden, wie den internen hydrostatischen Druck oder die Elastizität des internen Mediums, um Formeln der Hydrodynamik anzuwenden.

[Hart 2003] Gleichung F18 zeigt, wie ω mit Hilfe des internen hydrostatischen Druckes P , der Kraft F und der Oberfläche A , auf welche F appliziert wird, berechnet werden kann.

$$\mathbf{F18} \quad \omega = \frac{F}{P \cdot A} - 1$$

In diesem Fall wären P und F die Modellierungsparameter und A würde aus der Spezifizierung der Biofläche extrahiert. Es ist natürlich auch möglich, die Werte von P und F direkt im Laufe einer Simulation automatisch zu berechnen.

Auf diese Weise kann ein morphisches Feld als 3-Tupel beschrieben werden: $F := (S; \omega; d)$. Um den Parameter r der Blob-Funktion $B(r, d)$ an einem bestimmten Oberflächenpunkt $\tau := (f_R; f_A; f_H)$ zu ermitteln, wird die Γ -Distanzfunktion F19 benutzt, welche eine skalare Entfernung zwischen dem Punkt τ innerhalb des Biofeldes $S := (c_R; c_A; c_H=1; r_R; r_A; r_H=0)$ zum Zentrum des Biofeldes S ermittelt. Das Ergebnis wird mit den relativen Auslenkungen ω gewichtet, die für das entsprechende morphische Feld mittels Formel F18 berechnet werden.

$$\mathbf{F19} \quad \Gamma(\tau, S) = \sqrt{\left(\frac{f_R - c_R}{r_R}\right)^2 + \left(\frac{f_A - c_A}{r_A}\right)^2 + \left(\frac{f_H - c_H}{r_H}\right)^2}$$

Somit kann der Deformationsfaktor $\Theta(\tau, F)$ des Punktes $\tau := (f_R; f_A; f_H)$ für ein gegebenes morphisches Feld $F := (S; \omega; d)$ wie folgt durch die Formel F20 berechnet werden:

$$\mathbf{F20} \quad \Theta(\tau, F) = \begin{cases} \frac{(K-1)^2}{(1+d \cdot K)} \cdot \omega & K < 1 \\ 0 & K \geq 1 \end{cases}, K := \left(\frac{f_R - c_R}{r_R}\right)^2 + \left(\frac{f_A - c_A}{r_A}\right)^2 + \left(\frac{f_H - c_H}{r_H}\right)^2$$

Unter Verwendung der Formel F20 kann der resultierende Deformationsfaktor f_S am Oberflächenpunkt τ für für diejenige Menge von n morphischen Feldern F_i ($i = 1, \dots, n$), die

auf den Punkt einen topologischen Einfluss haben, durch folgende Formel F21 berechnet werden:

$$\text{F21} \quad f_s = \sum_{i=1}^n \Theta(\tau, F_i)$$

4.4 Die Abschätzung des Volumens

Die Volumenkonstanz ist eine der auffälligsten Rahmenbedingungen, sowohl in der Hydrodynamik als auch in der Biologie. Das Volumen jeder abgeschlossenen hydraulischen Einheit bleibt auch nach beliebigen Deformationen und Bewegungen vom Betrag her gleich. [Ullmann 2003]

Es ist dabei zu beachten, dass die Volumenkonstanz nur für kurzfristige Statusänderungen des Organismus gilt, jedoch nicht für Wachstumsveränderungen oder evolutionäre Formveränderungen, da hier die hydropneumatischen Einheiten des Organismus, vorwiegend durch metabolische und osmotische Prozesse, Flüssigkeiten einlagern und die hydropneumatischen Einheiten damit nicht mehr als abgeschlossen angesehen werden können. Trotzdem ist die Abschätzung des aktuellen Volumens auch für solche langfristigen Prozesse relevant, da das Wachstum als Einlagerung und evolutionäre Formveränderungen allgemeiner als Umschichtungsprozess von tatsächlichen oder imaginären Körperflüssigkeiten verstanden werden kann. Somit modifiziert das Volumen der internen Flüssigkeit – zusammen mit anderen Faktoren wie z.B. der Oberflächenviskosität – die Formausprägung der hydropneumatischen Einheit und nicht umgekehrt.

Um Volumenkonstanz eines Organismus zu erreichen, muss während jeder Bewegung oder Deformation kontinuierlich das Volumen abgeschätzt werden, um eventuelle Abweichungen vom Referenzvolumen zu erkennen und Gegenmaßnahmen einzuleiten. Da sich der Organismus bei der Quaoaringtechnologie langsam der Zielform anpasst und somit plötzliche große Volumenänderungen ausgeschlossen sind, reicht als Gegenmaßnahme, den Organismus um einen kleinen inkrementellen Betrag größer (bei negativer Volumenabweichung) oder entsprechend kleiner (bei positiver Volumenabweichung) zu skalieren.

Es gibt verschiedene Möglichkeiten für die Abschätzung des Volumens, wovon folgende vier Verfahren näher unter die Lupe genommen und realisiert wurden:

- Die Volumenberechnung mit punktkonzentrischen Pyramiden
- Die Volumenabschätzung mit Kugelschichten
- Die Volumenberechnung mit Achtfächern entlang der Zentrallinie
- Die Volumenabschätzung mit Prismen entlang der Zentrallinie

4.4.1 Die punktkonzentrische Volumenberechnung mit Pyramiden

Die punktkonzentrische Volumenberechnung mit Pyramiden ist die wohl intuitivste Art zur Bestimmung des Gesamtvolumens einer pneumatischen Einheit. Hier wird iterativ jedes Oberflächenpolygon des Pneus berücksichtigt. Von jeder Kante dieses Oberflächenpolygons wird eine dreieckige Seite zu einem imaginären Zentralpunkt $\vec{p}_{Zentral}$ gezogen, so dass jeweils eine Pyramide entsteht. Da bei der Quaoaringtechnologie nur Dreiecke als Polygone für das Skin-Mesh verwendet werden, entsteht eine dreiseitige Pyramide mit der Spitze $\vec{p}_{Zentral}$. Die Volumina dieser dreiseitigen Pyramiden werden dann aufsummiert, um das Gesamtvolumen der pneumatischen Einheit zu errechnen.

Die Vertices des Skin-Meshs werden – wie an anderer Stelle schon erwähnt – bei der Quaoaringtechnologie radial äquidistant um die Zentrallinie positioniert. Daher gilt, dass m Vertices in lateraler Richtung entlang der Zentrallinie und n Vertices radial um die Zentrallinie herum berücksichtigt werden müssen. Auf jeden Vertex $\vec{v}_{k,j}$ mit $k=1\dots m-1$ und $j=1\dots n$ kommen jeweils zwei dreieckige Polygone im Skin-Mesh, welche die Basis für Pyramiden bilden, die im Folgenden als *PyrA* und *PyrB* bezeichnet werden. Eine Ausnahme bildet die letzte craniale Vertexreihe, so dass insgesamt $2n(m-1)$ Dreiecke bzw. entsprechende imaginäre Pyramiden existieren.

Es gibt im Skin-Mesh zwei gleichwertige Möglichkeiten, die Dreiecke sinnfällig aufzuspannen, eine „absteigende“ und eine „aufsteigende“, sie unterscheiden sich jeweils darin, welche Vertices die Dreiecke als Ecken verwenden. Bei jeder Variante entstehen zwei verschiedene Dreieckstypen, die mit *A* und *B* gekennzeichnet werden.

Bei der *absteigenden Aufspannungsvariante* erstrecken sich die Dreiecke entlang in longitudinaler Richtung im Uhrzeigersinn verlaufender Diagonalkanten. Die Dreieckstypen *A* und *B* sind dabei folgendermaßen aufgespannt:

$$D_{k,j}^{A-ab} = (\vec{v}_{k, \text{mod}(j+1, n)}, \vec{v}_{k,j}, \vec{v}_{k+1, \text{mod}(j+1, n)}) \quad , k=1 \dots m-1, j=1 \dots n$$

$$D_{k,j}^{B-ab} = (\vec{v}_{k+1,j}, \vec{v}_{k+1, \text{mod}(j+1, n)}, \vec{v}_{k,j}) \quad , k=1 \dots m-1, j=1 \dots n$$

Bei der *aufsteigenden Aufspannungsvariante* erstrecken sich die Dreiecke entsprechend entlang in longitudinaler Richtung entgegen dem Uhrzeigersinn verlaufender Diagonalkanten. Die zwei Dreieckstypen A und B besitzen dann folgende Eckvertices:

$$D_{k,j}^{A-auf} = (\vec{v}_{k,j}, \vec{v}_{k+1,j}, \vec{v}_{k, \text{mod}(j+1, n)}) \quad , k=1 \dots m-1, j=1 \dots n$$

$$D_{k,j}^{B-auf} = (\vec{v}_{k+1, \text{mod}(j+1, n)}, \vec{v}_{k, \text{mod}(j+1, n)}, \vec{v}_{k+1,j}) \quad , k=1 \dots m-1, j=1 \dots n$$

Im Folgenden wurde sich für die aufsteigende Aufspannungsvariante entschieden. Ein Beispiel für die Aufspannung eines Dreieckspaares $D_{4,6}^{A-auf}$ und $D_{4,6}^{B-auf}$ ist auf Abbildung 60 zu sehen:

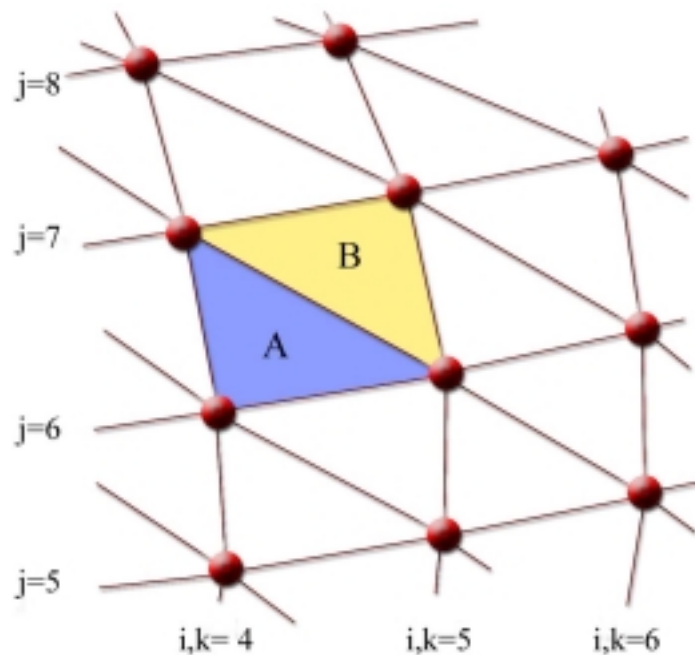


Abbildung 60: Skizze der Polygonverteilung im Skin-Mesh

Skizze der Polygonverteilung im Skin-Mesh mit aufsteigender Aufspannung. Zu sehen sind verschiedene Vertices (rote Kugeln), die Kanten (lila Linien), ein Dreieck A mit den Werten $k=4$ und $j=6$ (blau) und das korrespondierende Dreieck B (gelb).

Das Volumen einer allgemeinen dreiseitigen Pyramide beträgt bekanntermaßen ein Sechstel des von drei Kanten, die gemeinsam eine beliebige Pyramidenecke berühren, aufgespannten Spates gemäß Formel F22:

$$\mathbf{F22} \quad V_{\text{Pyramide}} = \frac{1}{6} V_{\text{Spat}}$$

Das Volumen des entsprechenden Parallelepipeds wird mit dem Spatprodukt derjenigen Kantenvektoren berechnet, die von der Pyramidenspitze \vec{p}_{Zentral} ausgehen. Es ist dabei darauf zu achten, dass das Spatprodukt nicht kommutativ ist und für ein positives Vorzeichen des Spatvolumens die Reihenfolge der Kantenvektoren im Spatprodukt ein rechtshändiges System bilden muss.

Für das Volumen $V_{k,j}^{\text{PyrA}}$ der Pyramide $\text{PyrA}_{k,j}$ gilt dann Formel F23:

$$\mathbf{F23} \quad V_{k,j}^{\text{PyrA}} = \frac{1}{6} \left\langle (\vec{v}_{k,j} - \vec{p}_{\text{Zentral}}), (\vec{v}_{k+1,j} - \vec{p}_{\text{Zentral}}), (\vec{v}_{k, \text{mod}(j+1,n)} - \vec{p}_{\text{Zentral}}) \right\rangle$$

Und entsprechend für das Volumen $V_{k,j}^{\text{PyrB}}$ der Pyramide $\text{PyrB}_{k,j}$ Formel F24:

$$\mathbf{F24} \quad V_{k,j}^{\text{PyrB}} = \frac{1}{6} \left\langle (\vec{v}_{k+1, \text{mod}(j+1,n)} - \vec{p}_{\text{Zentral}}), (\vec{v}_{k, \text{mod}(j+1,n)} - \vec{p}_{\text{Zentral}}), (\vec{v}_{k+1,j} - \vec{p}_{\text{Zentral}}) \right\rangle$$

Um das Gesamtvolumen des Pneus zu berechnen, werden gemäß Formel F25 für jeden nicht-cranialen Vertex die Volumina $V_{k,j}^{\text{PyrA}}$ und $V_{k,j}^{\text{PyrB}}$ der beiden korrespondierenden Pyramiden miteinander addiert und iterativ aufsummiert:

$$\mathbf{F25} \quad V_{\text{Pneu}} = \sum_{k=1}^{m-1} \sum_{j=1}^n (V_{k,j}^{\text{PyrA}} + V_{k,j}^{\text{PyrB}})$$

Werden die Formeln F23 und F24 in Formel F25 eingefügt, so entsteht Formel F26, die es erlaubt, das Volumen einer pneumatischen Einheit (unter bestimmten Voraussetzungen) genau zu berechnen:

$$\mathbf{F26} \quad V_{Pneu} = \frac{1}{6} \sum_{k=1}^{m-1} \sum_{j=1}^n \left(\langle (\vec{v}_{k,j} - \vec{p}_{Zentral}), (\vec{v}_{k+1,j} - \vec{p}_{Zentral}), (\vec{v}_{k, \text{mod}(j+1,n)} - \vec{p}_{Zentral}) \rangle + \right. \\ \left. \langle (\vec{v}_{k+1, \text{mod}(j+1,n)} - \vec{p}_{Zentral}), (\vec{v}_{k, \text{mod}(j+1,n)} - \vec{p}_{Zentral}), (\vec{v}_{k+1,j} - \vec{p}_{Zentral}) \rangle \right)$$

Die angesprochenen Voraussetzungen sind, dass keine einzige der Pyramidenkanten, die von $\vec{p}_{Zentral}$ ausgehen, das Skin-Mesh schneiden darf. Dies ist ein gravierender Nachteil der punktkonzentrischen Volumenberechnung mit Pyramiden: Diese Methode der Volumenberechnung eignet sich nur für hochgradig konvex geformte Pneus wie z.B. fusiforme Muskeln, Augäpfel oder kugelförmige Knorpelstrukturen.

Selbst bei diesen pneumatischen Einheiten ist die Positionierung eines geeigneten Zentralpunktes $\vec{p}_{Zentral}$ nicht einfach. Normalerweise bietet derjenige Punkt, der durch den Biovektor $\tau_{Zentral} = (0,5;0;0)$ beschrieben wird, eine gute Voraussetzung für eine möglichst zentrale Lage.

4.4.2 Die Volumenabschätzung mit Kugelschichten

Für eine andere Abschätzungsmethode des Pneuvolumens wird entlang der Zentrallinie eine Sequenz von Kugelschichten ermittelt, welche die Segmente des Pneus approximieren. Es bieten sich dabei diejenigen Kugelschichten an, welche jeweils den Radius r , Startpunkt s_{Start} und Endpunkt s_{End} desjenigen Ausgangssegmentes haben, auf welchem außer denjenigen radial wirkenden morphischen Feldern, die das Segment bilden, noch keine anderweitigen morphischen Felder appliziert wurden. Das angenäherte Volumen jedes Segmentes berechnet sich dabei nach der allgemeinen Formel F27 für Kreisabschnitte. Die Volumina müssen nur noch addiert werden, um das Gesamtvolumen des Pneus zu ermitteln.

$$\mathbf{F27} \quad V_{Segment} = \frac{1}{3} (s_{End}^2 \cdot (3r - s_{End}) - s_{Start}^2 \cdot (3r - s_{Start}))$$

Die Volumenabschätzung durch Kugelschichten ist die performanteste aller drei Methoden. Allerdings ist sie auch die ungenaueste Abschätzung, da longitudinale und lokale Deformationen des Pneus nicht mitberücksichtigt werden.

Bei einem Test mit 10 Pneuförmern, die in jeweils 12 verschiedenen Stadien aufgenommen wurden, ergab sich eine durchschnittliche Abweichung von 32,8% vom tatsächlichen Wert. Die maximale Abweichung bei den 120 Werten betrug 109,6%. Aus diesem Grund eignet sich diese Methode nur für pneumatische Einheiten mit hauptsächlich segmentbildenden Deformationen, die radial um die Zentrallinie verlaufen, für alle anderen Pneus kann sie lediglich die Größenordnung des Volumens grob abschätzen.

4.4.3 Die Volumenberechnung mit Pyramiden entlang der Zentrallinie

Das Skin-Mesh eines jeden Pneus beinhaltet eine Menge von Vertices $\vec{v}_{i,j}$ mit $i=1 \dots m$ und $j=1 \dots n$. Jeder Vertex $\vec{v}_{i,j}$ steht dabei mit mindestens einem Punkt $\vec{p}_i = (x_i, y_i, z_i)$ der Zentrallinie dergestalt in Verbindung, dass eine Ebene existiert, die orthogonal zur Zentrallinie am Punkt \vec{p}_i steht und gleichzeitig durch $\vec{v}_{i,j}$ verläuft. Diese Ermittlung der Verbindung zwischen \vec{p}_i und $\vec{v}_{i,j}$ ist einfach und benötigt keine zusätzliche Recheninstruktionen, da schon bei der Konvertierung von $\vec{v}_{i,j}$ vom biologischen ins kartesische Koordinatensystem eine Verbindungstabelle angelegt werden kann.

Das Volumen kann nun exakt gemäß Formel F32 durch Summenbildung über die Volumen $V_{k,j}^{\Pi}$ aller achtförmigen geometrischen Körper $\Pi_{k,j}$ mit $k = 1 \dots m-1$ und $j = 1 \dots n$ erfolgen, die entlang dieser Verbindungslinien zwischen den Vertices im Skin-Mesh und der Zentrallinie aufgespannt sind, respektive den achtförmigen Körpern mit folgenden Seiten $S_{k,j}^1$ bis $S_{k,j}^8$:

$$S_{k,j}^1 = (\vec{v}_{k,j}, \vec{v}_{k+1,j}, \vec{v}_{k, \text{mod}(j+1,n)})$$

$$S_{k,j}^2 = (\vec{v}_{k+1, \text{mod}(j+1,n)}, \vec{v}_{k, \text{mod}(j+1,n)}, \vec{v}_{k+1,j})$$

$$S_{k,j}^3 = (\vec{v}_{k,j}, \vec{v}_{k, \text{mod}(j+1,n)}, \vec{p}_k)$$

$$S_{k,j}^4 = (\vec{v}_{k+1, \text{mod}(j+1,n)}, \vec{v}_{k+1,j}, \vec{p}_{k+1})$$

$$S_{k,j}^5 = (\vec{v}_{k, \text{mod}(j+1, n)}, \vec{p}_{k+1}, \vec{p}_k)$$

$$S_{k,j}^6 = (\vec{p}_{k+1}, \vec{v}_{k, \text{mod}(j+1, n)}, \vec{v}_{k+1, \text{mod}(j+1, n)})$$

$$S_{k,j}^7 = (\vec{v}_{k,j}, \vec{p}_k, \vec{p}_{k+1})$$

$$S_{k,j}^8 = (\vec{v}_{k,j}, \vec{p}_{k+1}, \vec{v}_{k+1,j})$$

Die Orientierungen der Flächennormalenvektoren folgen dabei der Rechte-Hand-Regel.

Um die Volumina der achtflächigen Körper zu berechnen, werden sie jeweils in drei dreiseitige Pyramiden unterteilt. Dazu werden zusätzlich zu den Seiten $S_{k,j}^1$ bis $S_{k,j}^8$ folgende weitere Schnittseiten $S_{k,j}^A$, $S_{k,j}^{A'}$, $S_{k,j}^B$ und $S_{k,j}^{B'}$ benötigt:

$$S_{k,j}^A = (\vec{v}_{k,j}, \vec{p}_{k+1}, \vec{v}_{k, \text{mod}(j+1, n)})$$

$$S_{k,j}^{A'} = (\vec{p}_{k+1}, \vec{v}_{k,j}, \vec{v}_{k, \text{mod}(j+1, n)})$$

$$S_{k,j}^B = (\vec{v}_{k+1,j}, \vec{v}_{k, \text{mod}(j+1, n)}, \vec{p}_{k+1})$$

$$S_{k,j}^{B'} = (\vec{v}_{k, \text{mod}(j+1, n)}, \vec{v}_{k+1,j}, \vec{p}_{k+1})$$

Dabei stellen S_A und $S_{A'}$ sowie S_B und $S_{B'}$ im Grunde die gleichen Seiten dar, haben jedoch einen zueinander entgegengesetzten Flächennormalenvektor, so dass die Vor- und Rückseite zueinander invertiert sind.

Es ergeben sich nun folgende drei dreiseitige Pyramiden $P_{k,j}^1$, $P_{k,j}^2$ und $P_{k,j}^3$, deren räumliche Vereinigungsmenge den achtflächigen Körper $\Pi_{k,j}$ bilden:

$$P_{k,j}^1 = (S_{k,j}^A, S_{k,j}^3, S_{k,j}^5, S_{k,j}^7)$$

$$P_{k,j}^2 = (S_{k,j}^1, S_{k,j}^8, S_{k,j}^{A'}, S_{k,j}^{B'})$$

$$P_{k,j}^3 = (S_{k,j}^2, S_{k,j}^4, S_{k,j}^6, S_{k,j}^B)$$

Das Volumen $V_{k,j}^\Pi$ des achtflächigen Körpers $\Pi_{k,j}$ ist die Summe der Volumina $V_{k,j}^{P^1}$, $V_{k,j}^{P^2}$ und $V_{k,j}^{P^3}$ seiner Subpyramiden, daraus ergibt sich Formel F28:

$$\mathbf{F28} \quad V_{k,j}^{\Pi} = V_{k,j}^{P^1} + V_{k,j}^{P^2} + V_{k,j}^{P^3}$$

Das Volumen einer allgemeinen dreiseitigen Pyramide beträgt bekanntermaßen ein Sechstel des von drei Kanten, die gemeinsam eine beliebige Pyramidenecke berühren, aufgespannten Spates gemäß Formel F22 in Kapitel 4.4.1.

Zur Volumenberechnung einer einzelnen dreiseitigen Pyramide eignet sich daher am besten das Spatprodukt derjenigen Kantenvektoren, die ausgehend von einer beliebigen Ecke ein rechtshändiges System bilden. Da jede dreiseitige Pyramide vier Ecken hat, existieren für jede Pyramide auch vier gleichwertige Lösungen für die Volumenberechnung. In den folgenden Lösungsformeln F29 bis F31 für die Volumina der drei Pyramiden wurde sich bei Pyramide 1 und 2 für die gemeinsame Ecke $\vec{v}_{k,j}$ und bei Pyramide 3 für die Ecke \vec{p}_{k+1} entschieden:

$$\mathbf{F29} \quad V_{k,j}^{P^1} = \frac{1}{6} \langle (\vec{p}_k - \vec{v}_{k,j}), (\vec{v}_{k, \text{mod}(j+1, n)} - \vec{v}_{k,j}), (\vec{p}_{k+1} - \vec{v}_{k,j}) \rangle$$

$$\mathbf{F30} \quad V_{k,j}^{P^2} = \frac{1}{6} \langle (\vec{p}_{k+1} - \vec{v}_{k,j}), (\vec{v}_{k, \text{mod}(j+1, n)} - \vec{v}_{k,j}), (\vec{v}_{k+1, j} - \vec{v}_{k,j}) \rangle$$

$$\mathbf{F31} \quad V_{k,j}^{P^3} = \frac{1}{6} \langle (\vec{v}_{k, \text{mod}(j+1, n)} - \vec{p}_{k+1}), (\vec{v}_{k+1, j} - \vec{p}_{k+1}), (\vec{v}_{k+1, \text{mod}(j+1, n)} - \vec{p}_{k+1}) \rangle$$

Um das Gesamtvolumen des Pneu V_{Pneu} zu berechnen, müssen alle achtfächigen Körper berücksichtigt werden, aus denen sich der Pneu zusammensetzt. Daher gilt Formel F32:

$$\mathbf{F32} \quad V_{Pneu} = \sum_{k=1}^{m-1} \sum_{j=1}^n V_{k,j}^{\Pi}$$

Werden die Formeln F29 bis F31 in die Formel F22 eingesetzt und diese wiederum in Formel F32, so entsteht Formel F33, mit der das Gesamtvolumen des mit Polygonen visualisierten Körpers exakt berechnet werden kann. Da dieses Volumen bei hohen

Vertexgranularitäten praktisch identisch mit dem tatsächlichen Pneuvolumen ist, liefert Formel F33 in der Regel eine gute Annäherung:

$$\begin{aligned}
 \text{F33} \quad V_{Pneu} = & \frac{1}{6} \sum_{k=1}^{m-1} \sum_{j=1}^n \left(\left\langle (\vec{p}_k - \vec{v}_{k,j}), (\vec{v}_{k,\text{mod}(j+1,n)} - \vec{v}_{k,j}), (\vec{p}_{k+1} - \vec{v}_{k,j}) \right\rangle + \right. \\
 & \left. \left\langle (\vec{p}_{k+1} - \vec{v}_{k,j}), (\vec{v}_{k,\text{mod}(j+1,n)} - \vec{v}_{k,j}), (\vec{v}_{k+1,j} - \vec{v}_{k,j}) \right\rangle + \right. \\
 & \left. \left\langle (\vec{v}_{k,\text{mod}(j+1,n)} - \vec{p}_{k+1}), (\vec{v}_{k+1,j} - \vec{p}_{k+1}), (\vec{v}_{k+1,\text{mod}(j+1,n)} - \vec{p}_{k+1}) \right\rangle \right)
 \end{aligned}$$

4.4.4 Die Volumenabschätzung mit Prismen entlang der Zentrallinie

Wie im vorherigen Abschnitt erklärt, kann mit Hilfe der aufsummierten Volumina aller achtflächigen geometrischen Körper $\Pi_{k,j}$ entlang der Zentrallinie mit Hilfe von Formel F33 das Volumen eines Pneus sehr gut abgeschätzt werden. Allerdings ist diese Methode sehr rechenintensiv.

Um die Performanz des Quaoarings zu erhöhen, kann das Volumen der achtflächigen Körper aus Kapitel 4.4.3. durch Prismen angenähert werden, die zwischen $\vec{v}_{k,j}, \vec{v}'_{k+1,j} = (\vec{v}_{k,j} + \vec{p}_{k+1} - \vec{p}_k), \vec{v}_{k,\text{mod}(j+1,n)}$ und $\vec{v}'_{k+1,\text{mod}(j+1,n)} = (\vec{v}_{k,\text{mod}(j+1,n)} + \vec{p}_{k+1} - \vec{p}_k)$ als eine Prismenseitenfläche und \vec{p}_k und \vec{p}_{k+1} als zur Seitenfläche entgegengesetzte Prismenkante aufgespannt sind. Es ist dabei zu beachten, dass es sich mathematisch gesehen weder um ein gerades noch um ein gleichseitiges Prisma handelt.

Diese Annäherung ist möglich, da einerseits die Positionierung der Vertices des Skin-Meshs radial äquidistant um die Zentrallinie erfolgt und andererseits die Radien benachbarter Vertices in der Regel nur geringfügig abweichen. Vernachlässigbare Ausnahmen sind der craniale Anfang und das caudale Ende des Pneus sowie spitze lokale Deformationen. Bei lateral-sinoiden Beugungen des Pneus, wie sie z.B. bei schlangenähnlichen Bewegungen auftreten, können sich die Prismen auf der jeweils konkaven Beugungsseite stark durchdringen, so dass hier die Volumenabschätzung zu klein wird, allerdings werden die Prismen dann auf der entgegengesetzten konvexen Seite auseinander klaffen, so dass dieser Effekt fast vollständig aufgehoben wird. Das Volumen $V_{k,j}^{\Pi'}$ des Prismas $\Pi'_{k,j}$ ist also ungefähr gleich dem Volumen $V_{k,j}^{\Pi}$ des entsprechenden achtflächigen Körpers $\Pi_{k,j}$, es gilt also:

$$\mathbf{F34} \quad V_{k,j}^{\Pi'} \approx V_{k,j}^{\Pi}$$

Das Prismenvolumen beträgt bekanntermaßen die Hälfte des Produktes aus einer beliebigen Seitenfläche $A_{k,j}$ und der Höhe $h_{k,j}$ zur entgegengesetzten Seitenflächenkante. Das Volumen $V_{k,j}^{\Pi'}$ des Prismas $\Pi'_{k,j}$ ist daher durch Formel F35 gegeben:

$$\mathbf{F35} \quad V_{k,j}^{\Pi'} = \frac{A_{k,j} \cdot h_{k,j}}{2}$$

Die Höhe $h_{k,j}$ ist ungefähr identisch mit dem euklidischen Abstand zwischen $\vec{v}_{k,j}$ und \vec{p}_k und die Fläche $A_{k,l}$ ist identisch mit dem Produkt von longitudinaler Distanz $d_{longitudinal}^{k,j}$ (parallel zur Zentrallinie) und radialer Distanz $d_{radial}^{k,j}$ (um die Zentrallinie herum). Daraus ergibt sich Formel F36.

$$\mathbf{F36} \quad A_{k,j} = d_{longitudinal}^{k,j} \cdot d_{radial}^{k,j}$$

Die longitudinale Distanz ist dabei die euklidische Entfernung zwischen den zwei Referenzpunkten \vec{p}_k und \vec{p}_{k+1} auf der Zentrallinie, daraus resultiert Formel F37:

$$\mathbf{F37} \quad d_{longitudinal}^{k,j} = \sqrt{(x_k - x_{k+1})^2 + (y_k - y_{k+1})^2 + (z_k - z_{k+1})^2}$$

Die radiale Distanz entspricht der Entfernung eines Vertex $\vec{v}_{k,j}$ zu seinem Nachbarvertex $\vec{v}_{k, \text{mod}(j+1, n)}$ im Skin-Mesh. Sie ist annäherungsweise gleich dem n -ten Teil des Kreisumfangs eines imaginären Kreises mit dem Radius $h_{k,j}$, da alle Vertices $\vec{v}_{k,j}$ mit $k = \text{const}$ radial äquidistant um die Zentrallinie angeordnet sind. Daraus ergibt sich F38:

$$\mathbf{F38} \quad d_{radial}^{k,j} \approx \frac{2\pi \cdot h_{k,j}}{n}$$

Wenn die Formeln F37 und F38 in Formel F36 eingefügt werden und diese wiederum in Formel F35 eingefügt wird, entsteht Formel F39 für ein einzelnes Prismenvolumen:

$$\mathbf{F39} \quad V_{k,j}^{\Pi'} \approx \sqrt{(x_k - x_{k+1})^2 + (y_k - y_{k+1})^2 + (z_k - z_{k+1})^2} \cdot \frac{\pi \cdot h_{k,j}^2}{n}$$

Um das Gesamtvolumen der pneumatischen Einheit zu bestimmen, muss die Summe über alle Volumina $V_{k,j}^{\Pi'}$ der Prismen $\Pi'_{k,j}$ mit $k=1..m-1$ und $j = 1..n$ gebildet werden, daraus ergibt sich Formel F40:

$$\mathbf{F40} \quad V_{Pneu} \approx \sum_{k=1}^{m-1} \sum_{j=1}^n V_{k,j}^{\Pi'}$$

Wird Formel F39 in Formel F40 eingesetzt, ergibt sich die Approximationsformel F41 für die Bestimmung des Gesamtvolumens eines Pneus:

$$\mathbf{F41} \quad V_{Pneu} \approx \sum_{k=1}^{m-1} \sum_{j=1}^n \sqrt{(x_k - x_{k+1})^2 + (y_k - y_{k+1})^2 + (z_k - z_{k+1})^2} \cdot \frac{\pi \cdot h_{k,j}^2}{n}$$

Es ist dabei zu beachten, dass es sich nur um eine grobe Volumenabschätzung handelt. Je höher dabei die Granularitätswerte m und n gewählt werden und je weniger die Zentrallinie deformiert ist, desto genauer wird dabei das Resultat.

Bei einem Test mit 10 verschiedenen Pneuförmern, die in jeweils 12 unterschiedlichen Stadien aufgenommen wurden, ergab sich eine durchschnittliche Abweichung von 2,48% vom tatsächlichen Wert. Die maximale Abweichung bei den 120 Werten betrug 10,2%. Die Volumenabschätzung mit Prismen entlang der Zentrallinie liefert also optisch durchaus zufrieden stellende Ergebnisse, für analytische Zwecke ist die Genauigkeit allerdings nicht ausreichend.

4.4.5 Vergleich der Volumenabschätzungsmethoden

Die vier behandelten Verfahren zur Volumenabschätzung haben ihre jeweiligen Besonderheiten:

Am genauesten ist die Volumenabschätzung mit achtfächigen Körpern entlang der Zentrallinie. Da das Resultat dieser Methode exakt dem Volumen des mit Polygonen visualisierten Körpers entspricht, wächst die Genauigkeit mit der Anzahl der Vertices im Skin-Mesh. Da im Regelfall für die Vertexgranularität des Skin-Meshs eine 4- bis 5-stellige Größe gewählt wird und somit das visualisierte Körpervolumen fast dem tatsächlichen Pnevolumen entspricht, sind die Abweichungen vernachlässigbar.

Die Volumenabschätzung mit Kugelsschichten ist nur unter der Bedingung genau, dass entweder keine oder nur einschnürende Deformationen verwendet werden. Dies ist jedoch nur bei einigen anatomischen Strukturen möglich, wie beispielsweise bei persistaltischen Ringmuskelschläuchen oder bei Augäpfeln. Da die Volumenabschätzung mit Kugelschichten die performanteste der vier Methoden ist, bietet sie sich für solche Strukturen an. Je mehr allerdings andere Deformationsformen angewendet werden, desto größer ist die resultierende Ungenauigkeit. Bei stark lokal deformierten Strukturen ist diese Methode unbrauchbar.

Die Volumenabschätzung mit punktkonzentrischen Pyramiden ist nur bei konkaven Strukturen brauchbar und liefert hier nahezu exakte Ergebnisse. Da die Performanz kaum höher ist als bei der Volumenabschätzung mit achtfächigen Körpern entlang der Zentrallinie (beide Methoden haben eine lineare Komplexität bezüglich der Anzahl der Vertices im Skin-Mesh), ist der Vorteil dieser Methode als gering zu erachten.

Auch die Komplexität der Volumenabschätzung mit Prismen entlang der Zentrallinie ist linear bezüglich der Anzahl der Vertices. Die Ergebnisse hängen hier vor allem von der Deformation der Zentrallinie ab. Je weniger die Zentrallinie deformiert ist und je weniger kleinflächige Deformationen angewandt werden, desto besser sind in der Regel die Ergebnisse. Der Performanzvorteil gegenüber der Volumenabschätzung mit achtfächigen Körpern entlang der Zentrallinie ist zwar minimal, jedoch reicht in der Regel die Genauigkeit dieser Methode aus, um eine für den Betrachter plausible Volumenkonstanz zu erreichen.

Eine Übersicht über die jeweiligen spezifischen Vor- und Nachteile bei Genauigkeit und Performanz liefert folgende Tabelle 2:

Methoden	Genauigkeit	Performanz
Die Volumenberechnung mit punktkonzentrischen Pyramiden	nahezu exakt unter bestimmten Voraussetzungen	langsam
Die Volumenabschätzung mit Kugelsschichten	genau bei Pneus mit ausschließlich radial einschnürenden Deformationen, ansonsten sehr ungenau	sehr hoch
Die Volumenberechnung mit achtflächigen Körpern entlang der Zentrallinie	nahezu exakt	sehr langsam
Die Volumenabschätzung mit Prismen entlang der Zentrallinie	ungenau	langsam

Tabelle 2: Vor- und Nachteile der Methoden für die Volumenabschätzung eines Pneus

In dieser Tabelle sind die Vor- und Nachteile hinsichtlich Genauigkeit kurz beschrieben. Allein die Volumenberechnung mit achtflächigen Körpern vermag das Volumen einer pneumatischen Einheit immer exakt zu berechnen, allerdings ist sie mit $210n(m-1)$ Gleitkommaoperationen (n und m = Granularitätswerte des Skin-Meshes) auch eine der rechenintensivsten.

4.5 Die Feinmodellierung der Haut

Um der Oberfläche eines Pneus mit Feinstrukturen zu versehen, wurde die Technik des *hydrostatischen Bump Mappings* eingeführt. Das hydrostatische Bump Mapping ist eine Weiterentwicklung und Spezialisierung der morphischen Felder, welche im Kapitel 4.3.3 behandelt wurden.

Anstatt jedoch – wie bei einem einfachen morphischen Feld – eine einzige große Kraft auf die pneumatische Einheit einwirken zu lassen, werden beim hydrostatischen Bump Mapping viele kleinere Kräfte auf ein Punktfeld $T_{i,j}$ der Oberfläche appliziert, wobei der Betrag der jeweiligen Kräfte $F_{i,j}$ unter Verwendung von Texturen modifiziert wird.

Durch diese Krafteinwirkungen werden die jeweiligen Deformationsfaktoren f_S an den Oberflächenpunkten $T_{i,j}$ verändert. Dies geschieht allerdings nicht direkt, sondern

indirekt über die Wechselwirkung von $F_{i,j}$ mit dem inneren hydrostatischen Druck P aufgrund von Formel F18.

Dadurch können auf einfache Weise Details auf dem Skin-Mesh wie Venen und Arterien, Runzeln und Fältchen, Pickel und Pusteln, Dornen und Stacheln, Blasen und Beulen, leder- und hautartige Oberflächen sowie jede andere feingranulare Tiefenstruktur modelliert werden.



Abbildung 61: Beispiel für hydrostatisches Bump Mapping

Mit hydrostatischem Bump Mapping lassen sich auf elegante Weise organische Strukturen modellieren. Der Salatkopf entstand durch einfache Applikation eines hydrostatischen Bump Maps auf einen kugelförmigen Pneu. Durch die hydrodynamische Pneuverformung entstehen automatisch biologisch stringente Strukturen.

Es gibt mehrere Vorteile dieser Technik gegenüber einer einfachen Verschiebung der Vertices entlang der Oberflächennormalen oder konventionellen Bump Mapping Verfahren wie beispielsweise das Emboss oder das Dot Product 3 Bump Mapping:

So verhält sich die Deformation beim hydrostatischen Bump Mapping weitgehend biologisch sinnfällig. Es besteht beispielsweise Volumenkonstanz, und bei Einbringung von Normalendeformationen ist auch Vesikelbildung möglich.

Beim hydrostatischen Bump Mapping werden die Vertexpositionen des Skin-Meshs direkt neu platziert. Dies vermeidet die klassischen Inkonsistenzen von konventionellen Bump Mapping Methoden: So wird die Silhouette des Objektes richtig dargestellt, Deformationen im Vordergrund verdecken sinnfällig dahinter liegende Deformationen und das Shading der Deformationen hebt sich nicht von den anderen Shadingweisen ab, d.h. es

wird das gleiche Beleuchtungsmodell angewandt, wie es auch bei anderweitig generierten Deformationen der Fall ist.

	hydrostatisches Bump Mapping	Verschiebung der Vertices entlang Oberfl.-normalen	Emboss Bump Mapping	Dot Product 3 Bump Mapping	Environment Mapped Bump Mapping
korrektes Wachstum	ja	nein	nein	nein	nein
Vesikelbildung	ja	nein	nein	nein	nein
Strukturen haben Einfluss auf die Silhouette	ja	ja	nein	nein	nein
Deformations-Okklusion	ja	ja	nein	nein	nein
stringentes Shading	ja	ja	nein	ja	nein
Illusion spiegelnder Oberflächen	nein	nein	nein	nein	ja

Tabelle 3: Vergleich des hydrostatischen Bump Mappings mit anderen Verfahren

Vergleich des hydrostatischen Bump Mappings gegenüber anderen Bump Mapping Verfahren bzw. einer einfacher Verschiebung der Oberflächenvertices mit Hilfe von Bump Map Texturen.

Zur Berechnung von hydrostatischen Bump Maps definiert der User zuerst auf der Oberfläche des Pneus eine Biofläche $S := (c_R, c_A, c_H=1, r_R, r_A, r_H=0)$, auf der die Bump Map einen Effekt erzielen soll. Dann spezifiziert der User m Texturen. Für jede Textur wird ein maximaler Deformationsfaktor ω_j mit $j=1..m$ angegeben, der dieselbe Funktion erfüllt wie der Parameter ω in morphischen Feldern. Die Farbwerte der einzelnen Pixel seien im RGB-Format gegeben.

Es kann nun eine Funktion $r_j(u,v)$ angegeben werden, die den Rotanteil eines Pixels mit den Koordinaten (u,v) der Textur j ermittelt. Die Koordinaten u und v sind dabei fraktional umgerechnet, d.h. sie sind jeweils im Bereich $[0,1]$ definiert. Analog dazu kann eine Funktion $g_j(u,v)$ für den Grünanteil und $b_j(u,v)$ für den Blauanteil dieses Pixels erstellt werden.

Ein Punkt T , spezifiziert als Biovektor τ , wird auf die Texturkoordinaten (u,v) projiziert, indem die Funktion Φ aus Formel F42 verwendet wird, welche auf der Γ -Distanzfunktion F19 beruht:

$$\mathbf{F42} \quad (u,v) = \Phi(\tau, S) = \begin{cases} (\Gamma(\tau), a/\pi) & a \geq 0 \\ (\Gamma(\tau), -a/\pi) & a < 0 \end{cases} \quad \text{mit} \quad a := \arctan\left(\frac{f_R - c_R}{f_A - c_A}\right)$$

Auf diese Weise wird die Textur radial um das Zentrum der Biofläche S herum gemappt, wobei die Pixelspalte mit $u=0$ auf die Ränder der Biofläche und die Pixelspalte mit $u=1$ auf das Zentrum der Biofläche und die intermediären Pixelspalten mit $0 < u < 1$ entsprechend in konzentrischen Ringen um das Zentrum projiziert werden. Der Parameter v wird dabei in einen Projektionswinkel übersetzt.

Um den Wert des Auslenkungsparameters f_S an einem Punkt T , spezifiziert durch den Biovektor τ , zu ermitteln, muss über alle Bump Map Texturen das arithmetische Mittel über Rot-, Grün- und Blauwert des zu T entsprechenden Farbanteiles gebildet werden. Zusätzlich müssen alle n morphischen Felder F_i berücksichtigt werden, die in diesem Punkt T wirksam sind. Dies führt zu Formel F43:

$$\mathbf{F43} \quad f_S = \sum_{i=1}^n \Theta(\tau, F_i) + \sum_{j=1}^m \left(\frac{r_j(\Phi(\tau, S)) + g_j(\Phi(\tau, S)) + b_j(\Phi(\tau, S))}{3} \cdot \omega_j \right)$$

Anstelle einer Addition aller f_S -Parameter, welche mit Hilfe der hydrostatischen Bump Map Texturen berechnet werden, kann auch einfach der Maximalwert von f_S genommen werden. Daraus resultiert Formel F44:

$$\mathbf{F44} \quad f_S = \sum_{i=1}^n \Theta(\tau, F_i) + \max_{j=1}^m \left(\frac{r_j(\Phi(\tau, S)) + g_j(\Phi(\tau, S)) + b_j(\Phi(\tau, S))}{3} \cdot \omega_j \right)$$

Diese Art der Interaktion wurde *konkurrierende Relieftexturen*, bzw. engl. *competitive bump maps* genannt. Konkurrierende Relieftexturen haben den Vorteil, dass sie sich gegenseitig verdrängen, womit evolutionäre Mutationsprozesse der Organismenoberfläche

mit weichen Übergängen zwischen zwei Evolutionsstufen visualisiert werden können. Abbildung 63 zeigt ein Beispiel für solche Verdrängungsprozesse.



Abbildung 62: Realistisches Verhalten von hydrostatischen Bump Maps

Beispiel für das realistische Verhalten von hydrostatischen Bump Maps. Links: Eine Textur, die sowohl als Image als auch als hydrostatische Bump Map dient. Mitte: Die Textur wurde auf eine Kugel gemappt. Es ist erkennbar, dass sich die Oberflächenstrukturen sinnfällig verdecken, auch verändern sich die Ränder der Silhouette gemäß der Oberflächentopologie. Rechts: Eine zusätzliche Kraftanwendung deformiert die Kugel, es ist ersichtlich, dass die hydrostatische Bump Map Textur sich ebenfalls sinnfällig mitdeformiert.

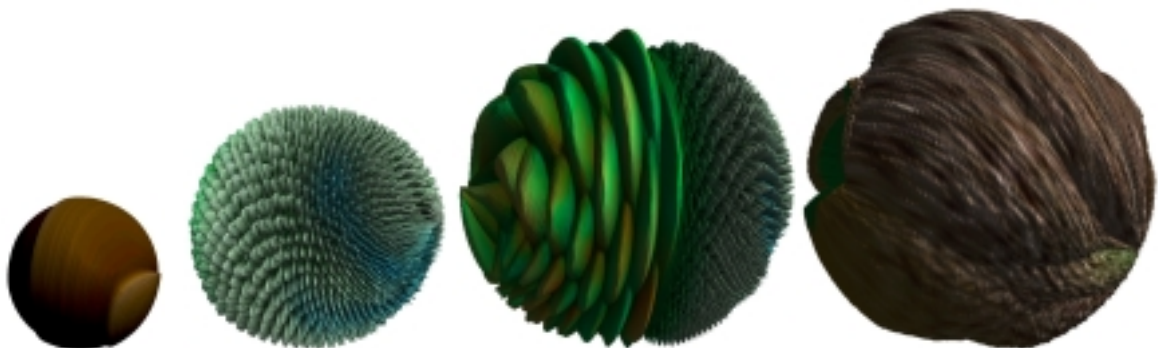


Abbildung 63: Wachstumsprozesse mit konkurrierenden hydrostatischen Bump Maps

Mit konkurrierenden hydrostatischen Bump Maps lassen sich Wachstumsprozesse schnell und einfach modellieren. Hier ein Beispiel eines transformativen Wachstumsprozesses mit vier Momentaufnahmen. Zu sehen sind v.l.n.r. ein Samenkorn, ein Keim mit grünen gummiartigen Dornen, eine exotische Frucht und eine Nuss. Diese Animation wurde lediglich durch kontinuierliche Erhöhung des Volumens eines kugelförmigen Pneus erzielt, auf den vier konkurrierende hydrostatische Bump Maps gelegt wurden. Die Bioflächenparameter der Bump Maps wurden dabei über die Zeit verändert. Bei der dritten Momentaufnahme sind gleichzeitig zwei konkurrierende hydrostatische Bump Maps zu sehen.

4.6 Verbinden zweier Pneus

Theoretisch wäre es möglich, einen kugelförmigen Pneu durch eine sukzessive Anwendung morphischer Felder zu verzweigen, um Gliedmaßen oder baumartige Strukturen zu erzeugen. In der Praxis reicht allerdings die Granularität des Skin-Meshs nicht aus, um auf diese Weise solche verzweigten Strukturen zu erzeugen. Auch summieren sich die Approximationsfehler der morphischen Felder auf, so dass spätestens nach der zweiten Verzweigung eine sinnvolle Pneuberechnung nicht mehr garantiert wäre.

Um dennoch komplexe Pneuförmungen zu erzeugen, muss die Möglichkeit geschaffen werden, verschiedene Pneus miteinander zu einem komplexen *Dendropneu* zu verbinden.

Da die Pneuförmungen bei Applikation vieler morphischer Felder sehr komplex sein können und die Pneus sich in Echtzeit dynamisch zueinander bewegen, stellt das Verbinden zweier verschiedener Pneus eine besondere Herausforderung der Quaoaringtechnologie dar.

Da die Pneus jeweils in sich abgeschlossene Gebilde darstellen, müssen die beiden Pneuhüllen zuerst sinnfällig „geöffnet“ oder „aufgebrochen“ werden, um zwischen den Löchern in der Pneuhülle ein spezielles Verbindungsstück einfügen zu können. In der Quaoaringtechnologie geschieht die Erzeugung einer solchen Perforation durch die Eliminierung geeigneter Polygone aus dem Skin-Mesh. Eine recht elegante Art, die betreffenden Polygone für die Eliminationsliste auszuwählen, ist die Anwendung eines morphischen Feldes in Verbindung mit einem Threshold. Dazu wird für alle Vertices der Pneuhülle die jeweilige lokale Stärke des morphischen Feldes berechnet. In einem zweiten Schritt wird für jedes Polygon das arithmetische Mittel der drei Feldstärken seiner Polygonecken ermittelt und mit dem Threshold verglichen; falls es den Threshold übersteigt, wird ein Eliminationsflag eingeschaltet. Nun werden für alle Vertices in der Pneuhülle die Eliminationsflags seiner korrespondierenden Polygone betrachtet. Hat ein Vertex sowohl angrenzende Polygone mit angeschaltetem als auch mit ausgeschaltetem Eliminationsflag, wird die Vertexposition in eine Liste eingetragen. Diese Positionen werden danach gemäß ihrer Reihenfolge um das Zentrum der Biofläche des morphischen Feldes zyklisch sortiert. Mit Hilfe der Einträge in der sortierten Liste werden nun neue Vertices erzeugt, die als Randvertices für das zu erstellende Verbindungsstück fungieren. Dieser Prozess wird bei beiden Pneus durchgeführt, so dass zwei Listen mit Randvertices generiert werden. Um das Verbindungsstück zu erzeugen, werden zwischen den Randvertices dreieckige Polygone eingefügt, bei denen jeweils zwei Ecken zur Menge der

ersten Randvertices und eine Ecke zur Menge der gegenüberliegenden Randvertices gehören. Zuletzt werden alle Polygone mit eingeschaltetem Eliminationsflag aus den Pneuometrien entfernt.

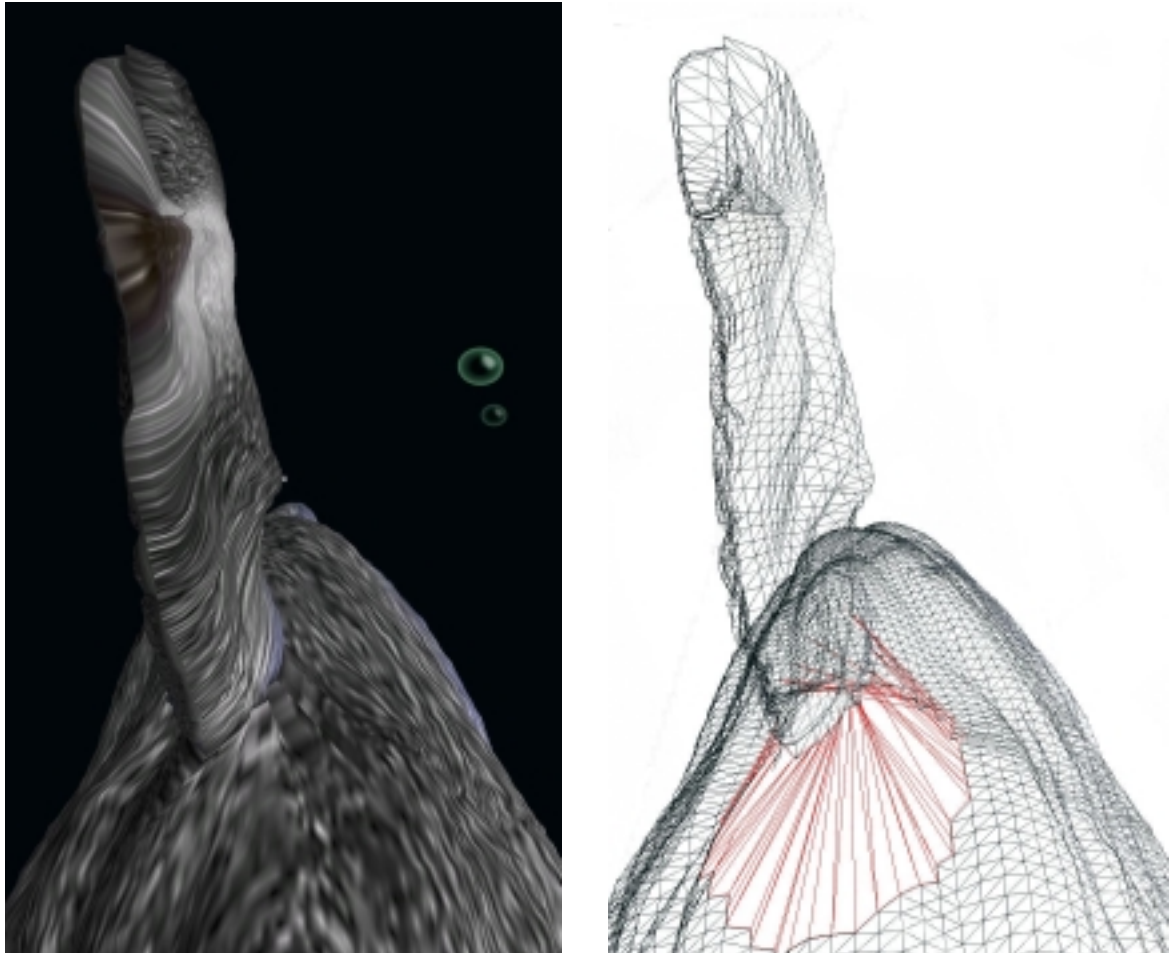


Abbildung 64: Beispiel einer Verbindung zwischen zwei Pneus

Links: Eine gerenderte Szene mit Rumpfpneu, Verbindungsstück und Flossenpneu. Lediglich anhand der Texturausrichtung lässt sich die Nahtstelle zwischen Verbindungsstück und Flossenpneu erahnen. Rechts: Dieselbe Szene als Drahtgittermodell, die Polygonkanten des Verbindungspneus sind rot dargestellt. Deutlich sind die fehlenden Polygone im Drahtgitter des Rumpfes zu sehen. In das entstandene Loch passt sich das Verbindungsstück nahtlos ein und erzeugt einen optisch zufrieden stellenden Übergang zum einseitig ausgeweiteten Flossenpneu.

4.7 Animation

Die Animation ist beim Quaoaring kein von der Modellierung und der Simulation getrenntes Verfahren, sie ist vielmehr holistisch in das Gesamtverfahren integriert. Die Animation erfolgt im Normalfall – analog zu den Muskelfasern bei realen Tieren – über ein System von *Kontraktionselementen*, die mit Hilfe von komplexen *Stimuli-Patterns* angesprochen werden. Durch den rhythmischen Stimulus üben die Kontraktionselemente zwei vom Betrag her gleiche Kräfte an ihren beiden Enden aus. Die Übersetzung von der Größe des Stimulus zur Kraftstärke des Kontraktionselementes erfolgt dabei zurzeit mit Hilfe einer einfachen linearen Relation. Für zukünftige Verbesserungen sind jedoch Modelle angedacht, die eher dem biologischen Vorbild entsprechen, analog der Kopplung zwischen dem Aktionspotential eines Nervs und der Kraftwirkung seiner verbundenen Muskelfaser.

4.7.1 Kopplung von Kontraktionselementen mit Zentrallinien

Der häufigste Fall bei der Anbindung der Kontraktionselemente ist ihre direkte Kopplung an zwei unterschiedliche Stellen der Zentrallinie. Mit einer solchen Konstruktion wird die Zentrallinie durch das Kontraktionselement gekrümmt. Ziehen sich die Kontraktionselemente zusammen, so wird sich die Zentrallinie analog zu einem Expanderstab deformieren. Da sich zusammen mit der Zentrallinie auch die lokalen Biokoordinaten des Pneu verändern, biegt sich der Pneu automatisch sinnfällig mit der Zentrallinie mit.

Um die Biegung der Zentrallinie physikalisch zu simulieren, wird Abschnitten der Zentrallinie ein virtuelles Elastizitätsmodul verliehen. Dieses ist ein physikalisches Maß für die Elastizität des Materials dieses Zentrallinienabschnittes, also für seine Widerstandsfähigkeit gegenüber Verformungen durch externe Kräfte, in diesem speziellen Falle Kräfte, die durch die Kontraktionselemente erzeugt werden. Zusammen mit diesem Elastizitätsmodul wird der Zentrallinie ein zusätzliches virtuelles Flächenträgheitsmoment gegeben.

Da sich die Zentrallinie zwischen zwei Stützstellen in der Regel nie über 120 Grad krümmen wird, kann zur Berechnung der Zentrallinienkrümmung zwischen den Angriffspunkten die Bernoulli-Euler-Näherung herangezogen werden, nach der das

angreifende Biegemoment M und die Krümmung, also die Ableitung des Biegewinkels Θ , zueinander proportional sind. Der Proportionalitätsfaktor α ist dabei gleich dem Produkt von axialem Flächenträgheitsmoment I und Elastizitätsmodul E . [Mayer 2003, 5] Daraus folgt Formel F45 für den Verlauf der Krümmungsveränderung des Zentrallinienabschnittes über die Strecke s :

$$\text{F45} \quad \frac{M(s)}{E(s)I(s)} = \frac{d\Theta}{ds}(s)$$

Für die Umwandlung der Kräfte der Kontraktionselemente in die resultierende Biegekräft werden die klassischen Regeln der Newtonschen Mechanik angewandt: Dazu werden erst alle angreifenden Kräfte durch Anwendung der Hebelgesetze und anschließender Vektoraddition in einen resultierenden Kraftvektor im nächsten Stützpunkt der Zentrallinie umgewandelt. Dann wird das Skalarprodukt zwischen demjenigen orthonormalen Basisvektor im Stützpunkt der Zentrallinie, welcher in der Biegeebene liegt, und dem resultierenden Kraftvektor genommen, um denjenigen Kraftbetrag zu ermitteln, der tangential zur Biegemomentverschiebung wirkt. Mit Hilfe des klassischen Momentensatzes wird dieser Kraftbetrag in das Biegemoment umgewandelt.

Das Elastizitätsmodul kann entlang der Zentrallinie auch lokal variieren. Dies erlaubt mit Hilfe einer Zentrallinie – die sich ja normalerweise eher wie eine Chorda oder bestenfalls wie eine Wirbelsäule verhält – auch Gelenke zu simulieren. Dazu werden einfach vier Stützpunkte eng aneinander positioniert und der Zentrallinie zwischen den mittleren zwei Punkten ein hohes Elastizitätsmodul gegeben. Außerhalb dieser Strecke wird das Elastizitätsmodul auf Null gesetzt, um einen internen Knochen zu simulieren.

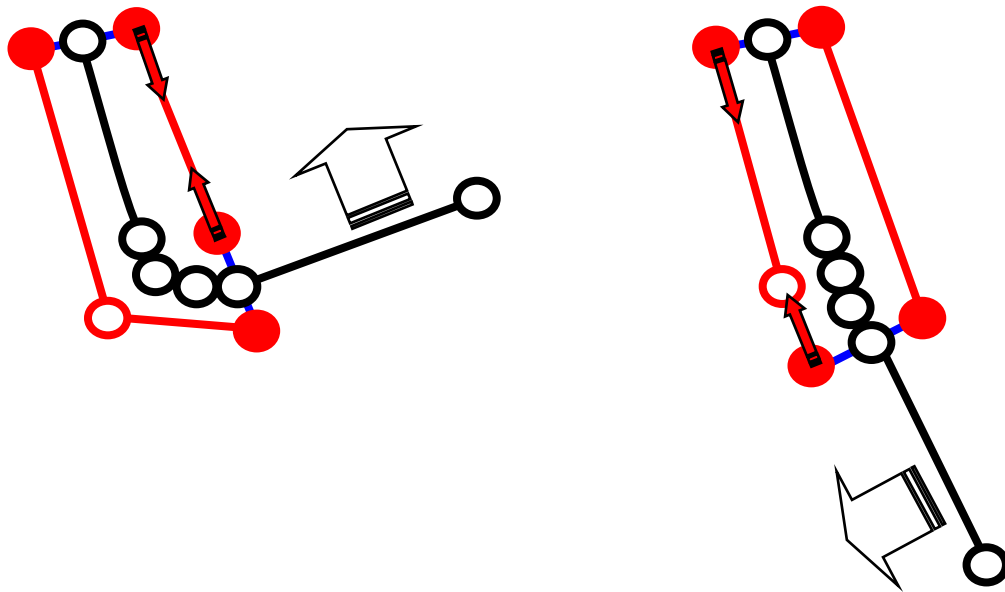


Abbildung 65: Beispielskizze für die Erstellung von Gelenken, Knochen und Muskeln

Beispielskizze für die Modellierung und Animation eines oberarmähnliches Gebildes mit Hilfe der Zentrallinie (schwarze Linie) und antagonistischen Kontraktionselementen (rote Linien): Um ein Gelenk herzustellen, werden vier Stützpunkte der Zentrallinie (schwarze Kreise) nahe hintereinander positioniert. Die Zentrallinie bekommt zwischen den mittleren Stützpunkten ein hohes Elastitätsmodul zur Simulation der Gelenkflexibilität, in den anderen Bereichen wird das Elastitätsmodul gleich Null gesetzt, um einen harten unbiegsamen Knochen zu simulieren. Die Kontraktionselemente werden – analog zweier antagonistischer Muskeln – via Hebel (blaue Linien) mit Stützpunkten der Zentrallinie verknüpft. Die Kontraktionselemente können dabei – wie hier im Falle des Kontraktionselementes links der Zentrallinie – über Umlenkungspunkte geleitet werden, die in biologischem Koordinatensystem definiert werden. Links: Durch Stimulierung des Kontraktionselementes rechts der Zentrallinie wird das Gelenksystem gebeugt. Rechts: Durch Stimulierung des Kontraktionselementes links der Zentrallinie wird das Gelenksystem gestreckt.

Es ist auch möglich, die Enden der Kontraktionselemente mit zwei verschiedenen Zentrallinien zu verknüpfen, die adjazenten Pneus angehören. Mit einer solchen Verbindung zwischen zwei Pneus können z.B. Flossenbewegungen erzeugt werden. Wie bei echten Tierkonstruktionen ist dabei anzuraten, für jeden Agonisten einen Antagonisten als Kontraktionselement anzubringen.

Eine weitere Möglichkeit ist es, Kontraktionselemente mit morphischen Feldern zu verbinden. Die Kraft, die durch die Kontraktionselemente erzeugt wird, modifiziert zusammen mit dem internen hydrostatischen Druck über Formel F18 direkt den Auslenkungsparameter ω . Dadurch wird sich die Pnehülle im Bereich der Bioflächen der morphischen Felder beim Stimulieren des Kontraktionselementes gemäß hydrostatischer Regeln zusammenziehen. Mit solchen Verknüpfungen über morphische Felder ist es u.a.

möglich, Ringmuskeln zu simulieren, die sich wie ein Band um den Pneu herumwinden. Ringmuskeln sind vor allem für peristaltische Bewegungen unabdingbar.

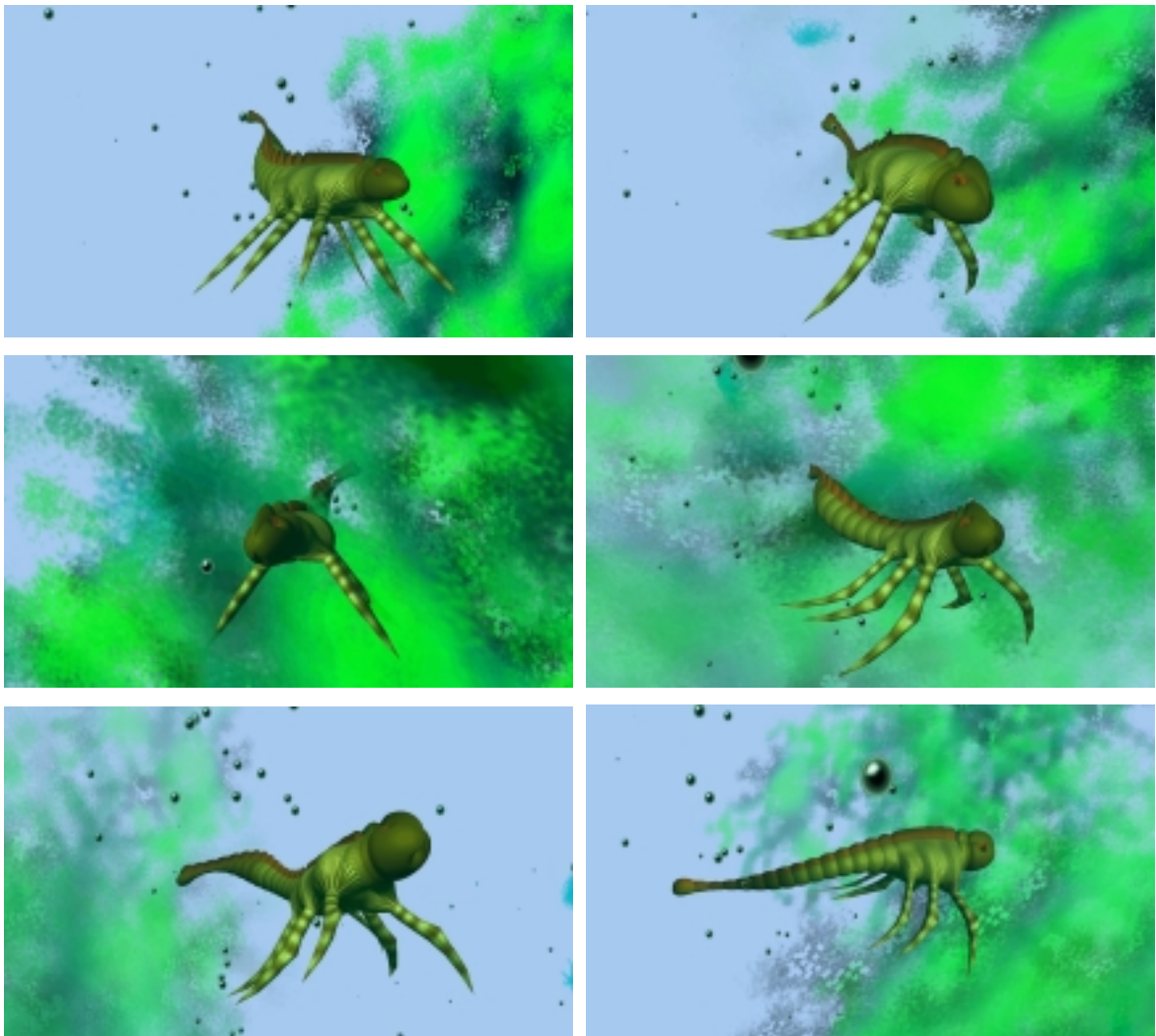


Abbildung 66: Sechs Beispielstationen einer Organismenanimation

Es sind sechs aufeinander folgende Stationen einer Animation eines insektenähnlichen Organismus zu sehen. Die sinoiden Schlängelbewegungen wurden durch 2 mal 18 antagonistische Kontraktionselemente erzeugt, die entlang der Zentrallinie des Rumpfpneus angebracht wurden. Die Bewegungen der Beine wurden durch 12 Kontraktionselemente erzeugt, die zwischen der Zentrallinie des Rumpfpneus und der Zentrallinie der 6 Beinpneus aufgespannt sind. Zusätzlich existieren noch 19 Kontraktionselemente, welche die Stärke der radial geformten morphischen Felder verändern, welche das Tier segmentieren. Durch sequentielle Stimulierung dieser 19 Kontraktionselemente entstehen zusätzliche peristaltische Bewegungen. Das Plankton, die Luftblasen und die Schwebeteilchen, durch die das Tier hindurch schwimmt, wurden jeweils mit speziellen Partikelsystemen erzeugt, deren Emissionsort an das Tier gebunden ist.

4.7.2 Kontraktionssysteme

Um den Modellierer nicht mit anatomischen Detailfragen zu überfordern, kann er für eine Tierkonstruktion vorgefertigte komplette *Kontraktionssysteme* einfügen. Wählt er z.B. das Kontraktionssystem „maritimer Saurier“, wird in das gesamte Pneugefüge automatisch ein gesamtes kontraktiles Innenleben eingebaut und der modellierte Organismus verhält sich dann von der Bewegung her wie ein Meeressaurier. Da sich das Kontraktionssystem durch die Verwendung des biologischen Koordinatensystems automatisch an den Zielorganismus adaptiert, werden verhältnismäßig wenig vorgefertigte Kontraktionssysteme benötigt, um eine ganze Klasse von der Bewegungsdynamik her gleichartige Tierkonstruktionen zu simulieren. Die Auswahl vorgefertigter Kontraktionssysteme macht eine eigene Animation überflüssig und reduziert die Arbeit auf einen Mausklick. Die Auswahl des Kontraktionssystems kann schon vor dem eigentlichen Modellierungsvorgang erfolgen. Der Modellierer kann dann schon während der Modellierung die Bewegungen des Organismus betrachten.

4.7.3 Steuerungsmodus

Bei der vorgestellten Animation durch Kontraktionselemente handelt der Organismus autonom, d.h. er führt Kontraktionsbewegungen durch, die von unbeeinflussbaren Stimuli-Patterns vorgegeben werden. Dies hat einige Vorteile, da der Modellierer sich nicht dezidiert um die Animation kümmern muss und durch das biologenahe Modell der Bewegungserzeugung die Bewegungen trotzdem korrekt sind. Auch mutet der Organismus durch seine eigenständig ausgeführten Bewegungen auf den ersten Blick lebendig an, auch wenn er es natürlich nicht ist. Diese originelle Form der Animation hat aber auch den Nachteil, dass der Modellierer keinen Einfluss auf die genauen Bewegungsabläufe hat. Dies kann für bestimmte Anwendungen ein Ausschlusskriterium sein. Vor allem bei der Generierung von Filmszenen sind genau festgelegte Bewegungsabläufe erwünscht.

Damit Quaoaring auch bei diesen Anwendungsfällen eingesetzt werden kann, wurde ein *Steuerungsmodus* eingeführt. Der Steuerungsmodus ist immer dann angebracht, wenn der Organismus nicht autonom agieren soll, sondern sklavisch bestimmte vorbestimmte Bewegungsabläufe abspulen soll. In diesem Modus werden die Stützpunkte direkt durch Bewegungsdaten positioniert. Die Kraftwirkungen der Kontraktionselemente werden dagegen ignoriert. So kann der Organismus auf konventionelle Art deterministisch animiert werden. Die Bewegungsdaten können dabei von klassischen Animationssystemen

bzw. Animationstechniken stammen, wie beispielsweise Motion Capturing, DIDs, Inverser Kinematik oder Vorwärtskinematik.

4.7.4 Animation von Wachstums- und Evolutionsprozessen

Eine besondere Stärke des Quaoarings ist die einfache Animationsmöglichkeit von Wachstums- oder Evolutionsprozessen, welche sich biologisch korrekt nach hydropneumatischen Regeln verhalten. Diese Animation geschieht auf gänzlich andere Weise als die Animation von Bewegungen:

Wie in Kapitel 4.1.1 bereits erwähnt, wird ein Organismus im Quaoaring normalerweise nicht von Grund auf neu erschaffen, sondern der Modellierer nimmt einen bereits erstellten Organismus, der seinem gewünschten Organismus am ehesten entspricht und modifiziert ihn danach lediglich. Die Modifikationen werden in Form einer speziellen Organismenkonstruktionsskriptsprache (wahlweise im XML- oder INI-Format) zusammen mit einer frei wählbaren *Mutationsdauer* danach in einer Deltadatei abgespeichert. Normalerweise wird der Modellierer dafür einen Organismus wählen, welcher vom Bauplan her der Vorläufer seiner gewünschten Tierkonstruktion sein könnte.

Dieses Modellierungsverfahren entspricht den Schöpfungsvorgängen in der Natur. Auch dort werden – entgegen kreationistischer Annahmen – neue Arten von Organismen nicht plötzlich aus dem Nichts erzeugt, sondern entstehen durch evolutionäre Veränderungen bestehender Organismen.

Dadurch, dass die Tierkonstruktionen aufeinander aufbauen und jede Tierkonstruktion einen Vorläufer und eventuell mehrere Nachfolger hat, entsteht eine baumartige Hierarchie, analog einem idealisierten evolutionären Stammbaum. Die Kanten des so entstehenden Tierkonstruktionsgraphen werden durch den Inhalt der Deltadateien beschrieben. Die Wurzel ist stets ein undeformierter und damit kugelförmiger Pneu. Der Sinn jedes anderen Knoten dieses Graphen erschließt sich infolge der Deltabeschreibung erst durch die Traversierung der Wurzel des Graphen in Richtung des betreffenden Knotens. Er ist dementsprechend sequentiell kontextsensitiv.

Diesen Tierkonstruktionsgraphen kann der Benutzer in beliebiger Reihenfolge traversieren. Wechselt er von einer Tierkonstruktion zur vorherigen bzw. zur nachfolgenden Tierkonstruktion, so geschieht der Wechsel nicht unmittelbar, sondern immer mit der im Deltafile angegebenen Mutationsdauer. Während dieser Zeit der

Mutation wird ein modifiziertes Parameterkeyframing angewandt, um eine evolutionäre Animation zwischen den aufeinander folgenden Konstruktionsstadien zu erzeugen. Dies geschieht durch lineare Interpolation der jeweiligen Parameterwerte über die Zeit. Es werden dabei alle Gleitkommaparameter berücksichtigt. Bei den Werten, die im biologischen Koordinatensystem angegeben sind, werden die einzelnen Tupelwerte und Subtupelwerte getrennt interpoliert: Bei einem morphischen Feld $F = (S; \omega; d)$ werden beispielsweise der Parameter ω und d berücksichtigt, aber zusätzlich auch die dynamischen Subtupelwerte über das Biofeld $S := (c_R; c_A; c_H=I; r_R; r_A; r_H=0)$, also c_R , c_A , r_R und r_A . Bei Farbwerten werden zurzeit einfach noch Rot-, Grün- und Blauanteil getrennt interpoliert, es ist jedoch angedacht, vor der Interpolation das RGB-System in ein wahrnehmungsmäßig gleichabständiges Farbsystem umzuwandeln und den Helligkeits-, Sättigungs- und Farbwert getrennt zu interpolieren, um den Farbübergang subjektiv zu verbessern.

Wird in einer Deltabeschreibung eine Komponente zur Tierkonstruktion hinzugefügt, z.B. ein morphisches Feld F , so wird für die Interpolation vorübergehend bei der vorhergehenden Tierkonstruktion ein Scheinfeld $F' := (S'; \omega'; d')$ mit gleichen Positions- und Modifikationswerten und auf Null gesetzten Stärkewerten erzeugt, also $S'=S$, $d'=d$ und $\omega'=0$. Dies ist deswegen notwendig, da ansonsten eine Interpolation unmöglich wäre und das neue Objekt ohne weichen Übergang plötzlich erscheinen würde.

Diese Art der Animation kann nicht nur zur Darstellung evolutionärer Übergänge verwendet werden, sondern gleichermaßen zur Darstellung von Wachstumsprozessen. Dazu werden die Vorläuferkonstruktionen in den Deltadateien nicht phylogenetisch, sondern ontogenetisch ausgewählt. Im Falle einer Tierkonstruktion nimmt der Modellierer einen kugelförmigen Ausgangspneu, den er als befruchtete Eizelle ansieht, lässt ihn durch Erhöhung seines Volumen wachsen und modifiziert ihn dabei interaktiv gemäß den Regeln der Gastrulation und anderer Gestaltveränderungen während der Embryonalentwicklung, so dass ein Embryo seiner gewünschten Tierkonstruktion entsteht. Während er das Tier weiter wachsen lässt, werden weitere hydropneumatische Modifikationen vorgenommen, die das Tier in die Adoleszenz reifen lassen. Die verschiedenen Stationen kann er in Deltadateien abspeichern (z.B. EizelleZuMorula.xml, MorulaZuBlastula.xml, BlastulaZuGastrula.xml etc.) Analog kann auch das Wachstum von Pflanzen animiert werden, dazu wird der kugelförmige Ausgangspneu als Samenkeim angesehen.

Diese Animation von Evolutions- und Wachstumsvorgängen zeigt einige positive Besonderheiten gegenüber anderen Möglichkeiten, Evolutions- und Wachstumsübergänge darzustellen, wie z.B. Überblendung, parametrisches Keyframing, partial Rescaling oder Morphing:

- Sie entsteht implizit beim Modellierungsvorgang. Es wird keine zusätzliche, umständliche Animationsarbeit benötigt, sondern sie entsteht quasi automatisch als „Nebenprodukt“ der Modellierung.
- Sie hält in jedem Stadium die dem Quaoaring zugrunde liegenden hydropneumatischen Regeln ein. Es wird also zu keinem Zeitpunkt der Animation ein inkonsistenter „nicht-lebensfähiger“ Organismus präsentiert.
- Sie basiert auf funktionalen anstatt geometrischen oder optischen Regeln (wie z.B. beim Morphing). Die Übergänge sehen daher besonders stringent aus.
- Sie erlaubt, evolutionäre Stammbäume frei zu traversieren.
- Sie kann parallel zur Bewegungsanimation stattfinden. Auch die intermediären Organismenformen wirken daher dynamisch und lebendig.

Der Modellierer ist trotz des biologienahen Konzeptes bei der Animation nicht unbedingt an biologische Vorlagen gebunden, er kann somit auch unrealistische Wachstums- und Evolutionsvorgänge von fiktiven Organismen erzeugen. Abbildung 67 zeigt ein solches Wachstum eines Phantasietieres. Abbildung 81 auf Seite 264 dagegen zeigt die biologisch korrekte Evolutionslinien-Animation der Enteropneustenentwicklung.

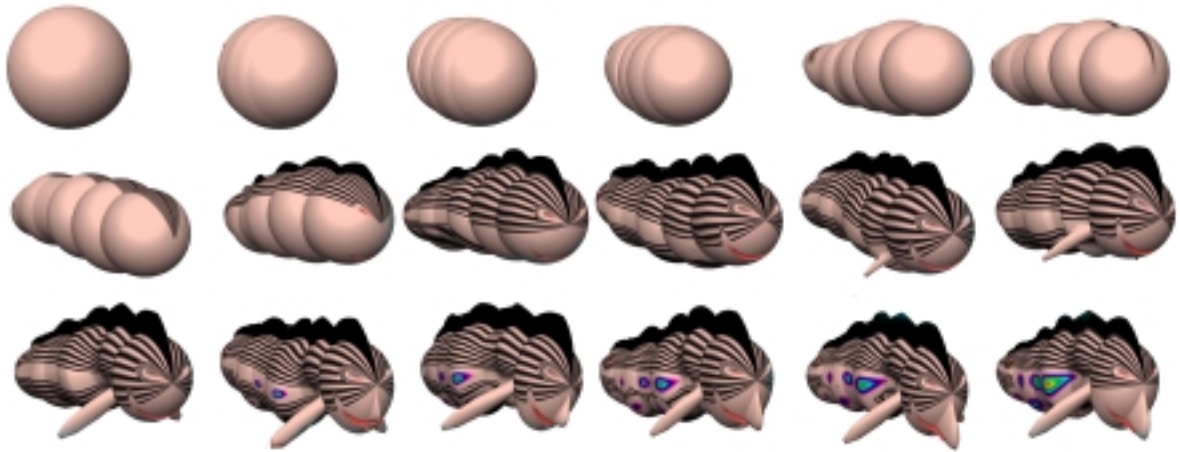


Abbildung 67: Wachstumsanimation eines Phantasietieres

Die Wachstums- und Evolutionsanimation ist nicht unbedingt an biologische Vorlagen gebunden, es können auch fiktive Wachstumsverläufe animiert werden. Hier sind 18 aufeinander folgende Wachstumsstationen eines Phantasietieres zu sehen.

4.8 Quaoaring Design Patterns

Quaoaring Design Patterns sind Objekte, die dem Modellierer beim Quaoaring zur Verfügung stehen, um Organismen zu konstruieren. Sie können zu einer komplexen *Objekthierarchie* zusammengefügt werden. In der Praxis (bei der Arbeit mit prototypischen auf Quaoaring basierenden Konstruktionsprogrammen) haben sich empirisch folgende Quaoaring Design Patterns als unbedingt notwendig erwiesen:

- Main (Hauptpneu)
- Bulge (Seitenpneu)
- Canal (Kanalpneu)
- Organ (Organpneu)
- Constriction (Schnürung)
- Conic (schlauchartige Quetschung)
- Squeeze (beulenartige Deformation)
- Contractile (kontraktilen Element)
- Relief (Tiefenstruktur der Oberfläche)
- Skin (spezielle Biotextur)
- Hair (Pattern für feine haarartige Strukturen)
- Material (optische und mechanische Oberflächeneigenschaften)
- Particlesource (Emitter)

Die Quaoaring Design Patterns lassen sich folgendermaßen unterteilen: *Main*, *Bulge*, *Canal* und *Organ* definieren jeweils eine pneumatische Einheit, bzw. einen Teil einer pneumatischen Einheit. Sie werden als *Pneurobjekte* bezeichnet.

Andere Quaoaring Design Patterns, wie *Constriction*, *Conic*, *Squeeze*, *Contractile* und *Relief* definieren ein Bündel von Kräften auf die Pnehülle, welche in einer Formveränderung des betreffenden Pneuroobjektes resultieren. Diese werden als *Deformationsobjekte* bezeichnet.

Es existieren auch Quaoaring Design Patterns, welche sich ausschließlich auf die Struktur und das Aussehen der Pnehülle auswirken, ohne dabei direkt deren Form zu ändern, diese sind im Einzelnen *Skin*, *Hair* und *Material*. Sie werden als *Oberflächenobjekte* bezeichnet.

Zuletzt gibt es auch noch ein spezielles *Emissionsobjekt*, welches sich nicht den obigen Pneu-, Deformations- und Oberflächenobjekte zuordnen lässt, den *Particlesource*.

4.8.1 Objekthierarchie der Quaoaring Design Patterns

Quaoaring Design Patterns können hierarchisch miteinander verknüpft werden. Mit Hilfe der entstehenden Objekthierarchie können komplexe Organismen erstellt werden. Dabei sind allerdings zwischen den verschiedenen Arten der Quaoaring Design Patterns nur sinnvolle Eltern-Kind-Beziehungen möglich. Es gelten folgende Hierarchieregeln:

- Jeder Pneu außer Main kann als Kind des anderen fungieren. Derjenige Pneu, der in einer solchen *Pneuhierarchie* direkt über dem Pneu *P* liegt, wird dabei *Elternpneu* von *P* genannt und diejenigen Pneus, die eine Hierarchiestufe unter *P* liegen, werden als *Kinderpneus* von *P* bezeichnet. Canals können als einzige Pneuobjekte auch zwei Elternpneus besitzen.
- Sowohl die Deformationsobjekte als auch die Oberflächenmodifikatoren können auf jedes beliebige Pneuobjekt appliziert werden. Eine Ausnahme davon bilden die Constrictions, Conics und Contractiles, die nicht als Kinder von Canals fungieren können. Contractiles können als besondere Ausnahme auch zwei verschiedene Elternpneus und maximal ein Organ als Kindpneu besitzen.
- Reliefs, Hairs und Ciliars können die Kinder von Skins sein, wobei Skins allerdings von jeder Sorte maximal ein Kind haben können.

An der Wurzel jeder Objekthierarchie steht jeweils ein Main-Objekt, der meist den Rumpf des betreffenden Organismus repräsentiert. Bei jeder Verknüpfung achtet eine Kontrollroutine darauf, dass keine zyklischen Hierarchien entstehen.

Folgende Tabelle 4 gibt einen Überblick über die möglichen hierarchischen Verknüpfungen:

Kind- objekt \ Eltern- objekt	Main	Bulge	Organ	Canal	Constriction	Conic	Squeeze	Contractile	Relief	Skin	Hair	Material	Particresource
Main	-	i	x	½i	x	x	x	½	x	x	x	x	x
Bulge	-	i	x	½i	x	x	x	½	x	x	x	x	x
Organ	-	i	x	½i	x	x	x	½	x	x	x	x	x
Canal	-	i	x	½i	-	-	x	-	x	x	x	x	x
Constriction	-	-	-	-	-	-	-	-	-	-	-	-	-
Conic	-	-	-	-	-	-	-	-	-	-	-	-	-
Squeeze	-	-	-	-	-	-	-	-	-	-	-	-	-
Contractile	-	-	x	-	-	-	-	-	-	-	-	-	-
Relief	-	-	-	-	-	-	-	-	-	-	-	-	-
Skin	-	-	-	-	-	-	-	-	1	-	1	1	-
Hair	-	-	-	-	-	-	-	-	-	-	-	-	-
Material	-	-	-	-	-	-	-	-	-	-	-	-	-
Particles.	-	-	-	-	-	-	-	-	-	-	-	-	-

Tabelle 4: Mögliche Eltern-Kind-Beziehungen der Quaoaring Design Patterns

Zu sehen ist eine Verbindungstabelle, welche mögliche Eltern-Kind-Beziehungen in der Objekthierarchie zwischen verschiedenen Arten von Quaoaring Design Patterns anzeigt. Zeilenweise stehen dabei die Elternobjekte und spaltenweise die Kindobjekte. Ein Strich in einer Zelle bedeutet, dass eine Eltern-Kind-Beziehung der entsprechenden Objekte nicht möglich ist, bei einem „x“ ist eine direkte hierarchische Verknüpfung dagegen erlaubt, bei „i“ ist sie indirekt über ein Verbindungsstück möglich (siehe Kapitel 4.6), bei „½“ hat das Kindobjekt zwei verschiedene Verknüpfungen, die eine Hierarchiestufe höher stehen, so dass das Quaoaring Design Pattern eventuell nicht nur eine, sondern zwei Elternobjekte besitzt, bei „1“ ist pro Elternobjekt maximal ein Kindobjekt pro Objektsorte gestattet.

4.8.2 Main

Mit Main wird derjenige Hauptpneu bezeichnet, der an der Wurzel einer Pneuhierarchie steht – und damit natürlich auch die Wurzel der betreffenden Objekthierarchie bildet. Pro Organismuskonstruktion kann jeweils nur ein Main existieren. Meist repräsentiert dieser Pneu den Rumpf des betreffenden Organismus.

Die wichtigsten modifizierbaren Parameter dieses Quaoaring Design Patterns sind das Innenvolumen des Pneus in m^3 sowie die longitudinale und die radiale Granularität des Skin-Meshs. Die Granularitätswerte geben dabei die Anzahl der Vertices an, die das Skin-Mesh jeweils longitudinal und radial aufweist. Je höher die Granularitätswerte sind, desto besser ist naturgemäß die Darstellungsqualität des Pneus, aber umso tiefer liegt auch die Frame Rate. Bei leistungsstarken Computern und wenig verzweigten Tierkonstruktionen können diese Werte hoch gesetzt werden.

Parameter	Maßeinheit	Typ	Variablenname	Default
Innenvolumen	m^3	float	fVolume	0
longitudinale Granularität	Anzahl Vertices	unsigned int	iGranLat	500
radiale Granularität	Anzahl Vertices	unsigned int	iGranRad	360

Tabelle 5: Modellierungsparameter von Main

Modellierungsparameter des Quaoaring Design Patterns „Main“ mit jeweiliger Maßeinheit, Typ, internen Variablenamen und Defaultwert.

4.8.3 Bulge

Ein *Bulge* ist ein Pneu, der an einen anderen Pneu via Interface Junction (Cut) angehängt wird und somit zu diesem eine physikalische Verbindung aufweist. Bulges werden meist an den Hauptpneu (Main) angebracht, um Extremitäten wie beispielsweise Beine, Arme, Rüssel oder große Tentakeln zu modellieren. Sie können aber auch mit Organen oder Canals verbunden werden, um komplexe interne Strukturen zu erzeugen, wie z.B. die Herzkranzgefäße oder den Wurmfortsatz. Ein Bulge lässt sich auch wiederum an einen anderen Bulge anhängen. Mit Hilfe von solchen hierarchisch verketteten Bulges lassen sich hochgradig verzweigte biologische Strukturen erstellen, z.B. Aststrukturen, Gliedmaßen mit Fingern und Zehen oder verzweigte Medusententakel.

Wie bei Mains lassen sich beim Bulge das interne Volumen in m^3 sowie die longitudinale und die radiale Granularität in Vertices definieren.

Ein Bulge benötigt als zusätzliche Parameterangabe diejenige Biofläche seines Elternpneus (gegeben in biologischen Koordinaten), wo er via Interface Junction angehängt wird. Das Zentrum der Biofläche definiert zusammen mit seinem

Normalenvektor zur Elternpneuhülle die Position und Orientierung des Bulges. Somit bewegt sich der Bulge immer sinnfällig mit seinem Superpneu.

Die Ausdehnung der Biofläche gibt dabei denjenigen Teil auf dem Superpneu an, der für die Interface Junction „ausgeschnitten“ werden soll.

Die Oberflächeneigenschaften erbt der Bulge dagegen von seinem Elternpneu. Wenn daher sowohl ein Bulge als auch ein Material ein Kind eines Mains ist, so besitzt die Oberfläche des Bulges ebenfalls die optischen und mechanischen Eigenschaften des Mains. Für spezielle Fälle lässt sich jedoch auch ein eigenes Material an einen Bulge anhängen. In diesem Fall wird der Bulge andere Oberflächeneigenschaften als sein Elternpneu aufweisen. Um die Übergänge dann nicht zu harsch erscheinen zu lassen, kann der Modellierer einen Interpolationsbereich definieren. Dafür setzt er das Interpolationsflag und gibt den longitudinalen Endpunkt des Interpolationsbereiches längs der Zentrallinie des Bulges an. Der Interpolationsbereich beginnt stets am Ansatzpunkt des Bulges. Alle optischen und mechanischen Parameter von Elternpneu und Bulge werden in diesem Interpolationsbereich linear interpoliert.

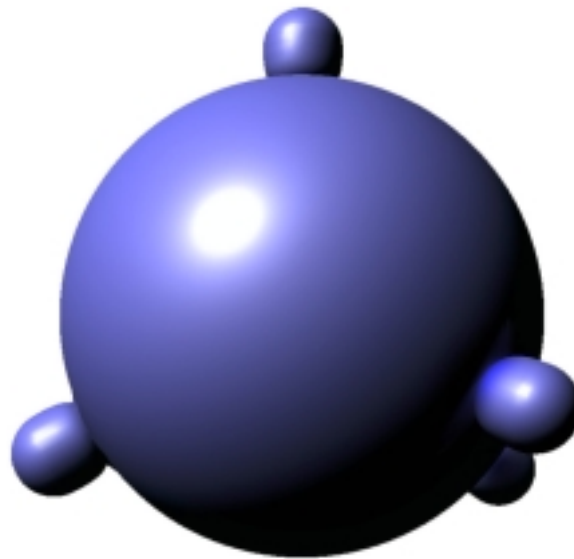


Abbildung 68: Beispiel von Bulges

An einen kugelförmigen Hauptpneu wurden hier vier Bulges als Kinderpneus angehängt. Die Polygonanzahl der Bulges kann einzeln angegeben werden.

Parameter	Maßeinheit	Typ	Variablenname	Default
Innenvolumen	m ³	float	fVolume	0
longitudinale Granularität	Anzahl Vertices	unsigned int	iGranLat	500
radiale Granularität	Anzahl Vertices	unsigned int	iGranRad	360
Interpolationsflag	-	bool	bInterpol	0
longitudinaler Endpunkt des Interpolationsbereiches	Bruchteil der Zentrallinie (BKS)	fractional float	frEndInterpol	0
Ansatzbereich an Elternpneu	Bioflächen-einheiten (BKS)	bioarea	baConnection	-

Tabelle 6: Modellierungsparameter von Bulge

Modellierungsparameter des Quaoaring Design Patterns „Bulge“ mit jeweiliger Maßeinheit, Typ, internen Variablenamen und Defaultwert.

4.8.4 Canal

Ein Canal ist ein schlauchförmiger Pneu, welcher über zwei Interface Junctions (Cuts) mit anderen Oberflächenpunkten von Pneus verbunden wird. Die zwei Verbindungspunkte können zwei verschiedene Oberflächenpunkte eines einzigen Pneus sein. Canals können aber auch zwei verschiedene Pneus miteinander verbinden. Eine zyklische Verbindung mit sich selbst ist jedoch nicht möglich. Auch eine Verbindung zwischen zwei verschiedenen Mains ist nicht vorgesehen, da sie biologisch keinen Sinn machen würde. Mögliche Superpneu-Verbindungen sind daher Main-Bulge, Main-Canal, Main-Organ, Bulge-Bulge, Bulge-Canal, Bulge-Organ, Canal-Canal, Canal-Organ und Organ-Organ.

Mit Canals lassen sich gut Organe des Intestinaltraktes wie Mundhöhle, Speiseröhre, Magen, Dünndarm, Dickdarm und After modellieren, aber auch alle anderen anatomischen Röhren wie z.B. der Gehörgang, die Luftröhre oder die eustachische Röhre sind damit modellierbar.

Wie bei allen anderen Pneus lassen sich beim Canal die longitudinale und die radiale Granularität seines Skin-Meshs einstellen.

Ein Canal benötigt als zusätzliche Parameter zwei Bioflächen seines bzw. seiner Superpneus (gegeben in biologischen Koordinaten), wo er via Interface Junction angehängt wird. Das Zentrum der Bioflächen definiert dabei die Start- und Endposition des Canals.

Anhand der Ausdehnung der Bioflächen wird ein initialer Start- und Endradius abgeschätzt. Dazwischen wird der Radius des Canals linear zwischen Startpunkt und Endpunkt entlang der Zentrallinie interpoliert. Ausbuchtungen und Einschnürungen des Canals werden durch die anschließende Applikation von morphischen Feldern modelliert. Somit kann im Intestinaltrakt (Canal) z.B. ein Magen (Ausbuchtung = morphisches Feld mit einem positiven Deformationsauslenkungsparameter ω) mit seinem Pförtner (Einschnürung = morphisches Feld mit $-1 < \omega < 0$) erzeugt werden.

Um den Verlaufsweg eines Canals (der identisch mit seiner Zentrallinie ist) anzugeben, gibt der Modellierer n Stützstellen in Form von Biovektoren an. Falls der Canal zwei verschiedene Pneus miteinander verbindet, können auch pro Stützstelle zwei verschieden gewichtete Biovektoren angegeben werden (jeweils ein Biovektor für den Ausgangspneu und ein Biovektor für den Zielpneu). Aus den zwei gewichteten Biovektoren wird dann jeweils ein Biovektor als resultierende Stützstelle berechnet. Zwischen den Stützstellen wird der intermediäre Weg kubisch (via NURBS) interpoliert.

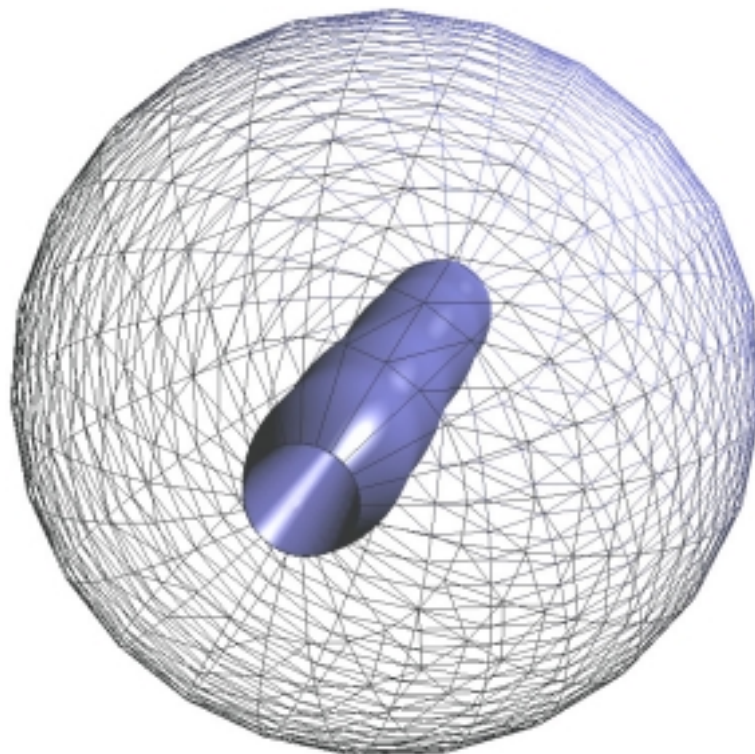


Abbildung 69: Beispiel von Canals

In einem kugelförmigen Pneu (als Drahtgittermodell zur Sichtbarmachung des Innenlebens) wird ein Canal mit vier Stützpunkten aufgespannt.

Die Definition der Position und Ausdehnung eines Canals in biologischen Koordinaten bewirkt, dass der Canal sich immer sinnfällig mit seinem Superpneu bzw. seinen Superpneus bewegt.

Analog zum Bulge können zu Canals Materials angefügt werden. Um in diesem Fall harsche Übergänge der Oberflächenstruktur zwischen dem Canal und seinen Superpneu am jeweiligen Verbindungspunkt zu vermeiden, kann der Modellierer zusätzlich pro Verbindungspunkt einen Interpolationsbereich auf der Zentrallinie angeben, in dem die damit verbundenen Oberflächeneigenschaften zwischen Canal und jeweiligen Superpneu linear interpoliert werden.

Parameter	Maßeinheit	Typ	Variablenname	Default
Innenvolumen	m ³	float	fVolume	0
longitudinale Granularität	Anzahl Vertices	unsigned int	iGranLat	500
radiale Granularität	Anzahl Vertices	unsigned int	iGranRad	360
Interpolationsflag	-	bool	bInterpol	0
longitudinaler Endpunkt des Interpolationsbereiches Nr.1	Bruchteil der Zentrallinie (BKS)	fractional float	frEndInterpol1	0
Ansatzbereich an Elternpneu Nr. 1	Bioflächen-einheiten (BKS)	bioarea	baConnection1	-
inverser longitudinaler Endpunkt des Interpolationsbereiches Nr2.	Bruchteil der Zentrallinie (BKS)	fractional float	frEndInterpol2	0
Ansatzbereich an Elternpneu Nr. 2	Bioflächen-einheiten (BKS)	bioarea	baConnection2	-
Anzahl n der Stützstellen	-	unsigned int	iPoints	
Stützstellen (n mal)	Biovektor-einheiten (BKS)	biovector	bvPoint[]	-
Gewichtungsrelation der Stützstellen	-	fractional float	frWeight[]	0,5

Tabelle 7: Modellierungsparameter von Canal

Modellierungsparameter des Quaoaring Design Patterns „Canal“ mit jeweiliger Maßeinheit, internen Variablenamen und Defaultwert.

4.8.5 Organ

Ein *Organ* ist ein Pneu, der innerhalb eines anderen Elternpneus (Main, Bulge, Canal oder Organ) liegt und sich sinnfällig mit diesem mitbewegt. Im Gegensatz zu Canals oder Bulges ist die Hülle eines Organs aber nicht direkt mit ihrem Elternpneu verbunden.

Wird ein Organ zum Hauptpneu (Main) hinzugefügt, eignet sich dieses Quaoaring Design Pattern zur Modellierung anatomischer Organe, deren Gewebestrukturen scharf von den Strukturen des Hauptpneus abgegrenzt sind. Darunter fallen beispielsweise das Herz, die Leber, die Blase, die Nieren, die Lunge, die Milz oder das Gehirn. Auch anatomische Strukturen, die im biologischen Sinne nicht als Organe bezeichnet werden, aber dennoch als scharf abgegrenzte Einheit im Körper auftreten, wie z.B. Zähne oder Augäpfel, lassen sich mit diesem Quaoaring Design Pattern leicht herstellen.

Ein Organ kann auch wiederum zu einem anderen Organ hinzugefügt werden. Dies ist z.B. dann sinnvoll, wenn vom Gewebe her scharf abgegrenzte anatomische Strukturen innerhalb anderer isolierter Strukturen auftreten, beispielsweise könnte ein Modellierer zum Organ „Niere“ ein weiteres Organ „Nebenniere“ als Subpneu von „Niere“ einfügen. Auch Unterteile des Augapfels wie Retina, Glaskörper, Iris und Linse sollten vorzugsweise in einer hierarchisch zwei- bis dreistufigen Organ-Pneuhierarchie verwirklicht werden.

Organs können auch als Subpneus in Canals eingefügt werden. Eine mögliche Anwendung wäre z.B. die Gehörknöchelchen Hammer, Amboss und Steigbügel als Organs zu modellieren und sie in den Canal „eustachische Röhre“ als Superpneu einzusetzen.

Die Einfügung von Organs in Bulges könnte z.B. bei Fingernägeln und Klauen geschehen.

Wie bei Mains und Bulges lassen sich auch beim Organ das interne Volumen einstellen. Dieses Volumen addiert sich zum Volumen des Superpneus.

Dazu existieren – analog zu Main, Bulge und Canal – die longitudinalen und radialen Granularitätsparameter.

Die Position und initiale Ausdehnung eines Organs wird in biologischen Koordinaten angegeben. Dazu wird einfach derjenige Bioraum seines Superpneus definiert, der seiner Form am ehesten entspricht. Durch zusätzliche Applikation von morphischen Feldern kann das Organ an die gewünschte Zielform adaptiert werden. Die Definition der Position und Ausdehnung eines Organs in biologischen Koordinaten bewirkt, dass das Organ sich immer sinnfällig mit seinem Elternpneu bewegt.

Parameter	Maßeinheit	Typ	Variablenname	Default
Innenvolumen	m ³	float	fVolume	0
longitudinale Granularität	Anzahl Vertices	unsigned int	iGranLat	500
radiale Granularität	Anzahl Vertices	unsigned int	iGranRad	360
Position und Ausdehnung	Bioraum- einheiten (BKS)	biospace	bsExtention	-

Tabelle 8: Modellierungsparameter von Organ

Modellierungsparameter des Quaoaring Design Patterns „Organ“ mit jeweiliger Maßeinheit, internen Variablennamen und Defaultwert.

4.8.6 Squeeze

Ein Squeeze ist eine lokal eng begrenzte Deformation eines Pneus. Eine Metapher für den Squeeze könnte das Hineindrücken bzw. Herausziehen der Pneuhülle an einer definierten Stelle mit einem Stab oder mit einem Finger sein. Ein Squeeze entspricht einem unmodifizierten morphischen Feld.

Mit Squeezes lassen sich u.a. Beulen, Noppen, Auswuchtungen, Gruben, Hörner, Dornen, Stacheln, Flossen, Rinnen und Kragen modellieren. Sollten mehrere Squeezes übereinander gelagert sein, addieren sich ihre Höhenwerte. Somit lassen sich auch komplexere Formen erzeugen, wie z.B. stachelbesetzte Kragen.

Um die Squeeze zu definieren, wird der Kraftbetrag F angegeben, der die Pneuhülle deformiert. Alternativ zu dieser direkten Kraftspezifikation kann der Modellierer auch die maximale Deformationsauslenkung ω spezifizieren, welche aus der Kraftanwendung resultiert.

Zusätzlich wird als Parameter die Biofläche benötigt, welche diejenige Region auf der Hülle repräsentiert, auf der diese Kraft einen sichtbaren Deformationseffekt hat. Das Zentrum der Biofläche entspricht dem Angriffspunkt von F , die Richtung der Kraft steht dabei immer senkrecht zur Pneuhülle.

Mit dem Schärfeparameter kann die „Spitzheit“ des Deformationswulstes modifiziert werden, kleine Werte erzeugen spitze Deformationen, große Werte dagegen stumpfe Deformationen.

Da Organismenstrukturen oftmals spiegelsymmetrisch (im BKS) entlang dem Planum Medianum aufgebaut sind, hat sich ein zusätzliches Symmetrieflag empirisch als sinnvoll erwiesen. Ist dieses Flag gesetzt, wird automatisch ein zweites Squeeze erzeugt, welches auf der anderen Körperhälfte ein analoges spiegelsymmetrisches morphisches Feld erzeugt.

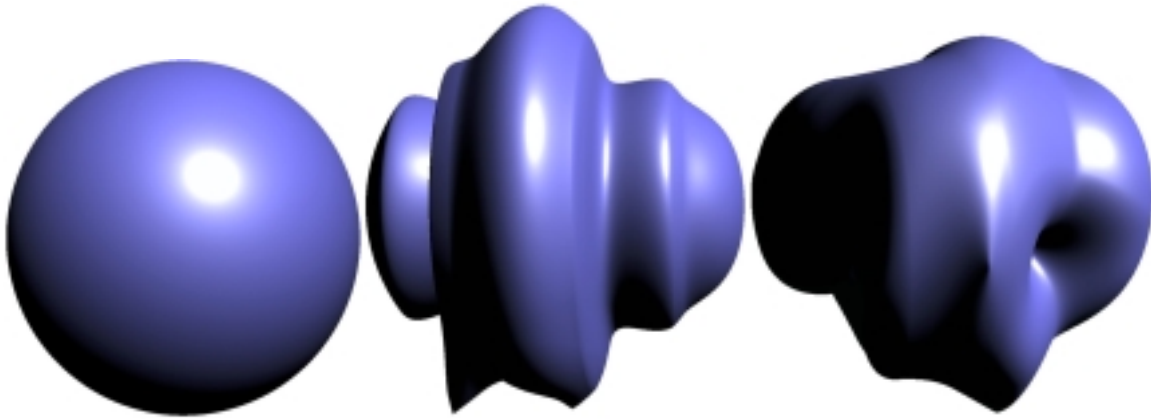


Abbildung 70: Beispiel von Squeezes

Auf einen kugelförmigen Pneu (links) werden zuerst Squeezes mit Bioflächen appliziert, deren r_A – Parameter groß gewählt wurde, so dass ein radial deformierter Pneu entsteht (Mitte), dann werden Squeezes mit eng begrenzten Bioflächen angehängt, so dass Dellen und Beulen entstehen.

Parameter	Maßeinheit	Typ	Variablenname	Default
Ausdehnung des Deformationswulstes	Bioflächen-Einheiten (BKS)	bioarea	baExtention	-
Umschaltflag zwischen direkter Angabe der Auslenkung und indirekter Kraftangabe	-	bool	bIndirect	0
maximale Auslenkung	radiale BKS-Einheiten	fractional float	frExcentricity	0
Auslenkungskraft	Newton	float	fForce	0
Schärfe	-	float	fSharpness	1
Symmetrieflag	-	bool	bSymmetry	0

Tabelle 9: Modellierungsparameter von Squeeze

Modellierungsparameter des Quaoaring Design Patterns „Squeeze“ mit jeweiliger Maßeinheit, internen Variablennamen und Defaultwert.

4.8.7 Constriction

Eine *Constriction* ist eine zur Zentrallinie orthogonale Einschnürung, die auf einen beliebigen Pneu angewendet werden kann. Eine Metapher dafür könnte eine Schlaufe sein, welche einen wassergefüllten Ballon zuschnürt.

Mit Constrictions kann ein Pneu segmentiert werden, z.B. würde die Segmentierung des Main-Pneus einer Biene in die 3 Segmente Kopf, Rumpf und Schwanz durch zwei Constrictions erfolgen. Zu beachten ist dabei, dass die laterale Vertexgranularität gleichmäßig auf die 3 Segmente verteilt wird. Hat der Main-Pneu der Biene beispielsweise eine laterale Granularität von 300 Vertices, so bekäme jedes Segment nach der Applikation der zwei Constrictions jeweils 100 Vertices, auch dann wenn auf den Kopf nur 2 Prozent der lateralen Gesamtlänge entfallen würden. Diese gleichmäßige Neuaufteilung der Granularität hat sich empirisch als sinnvoll erwiesen, da kleine Segmente meist ebensoviel biologische Strukturen aufweisen wie große Segmente.

Die vom Modellierer beeinflussbaren Parameter sind erstens die longitudinale Position der Constriction auf dem Pneu und zweitens die Schnürungstiefe. Beide zusammen können als spezieller Biovektor angegeben werden, bei dem der Parameter f_A ignoriert wird und der Parameter f_H die relative Schnürungstiefe definiert (relativ zum zur Zentrallinie orthogonalen Ausgangsradius des Pneus an der Stelle f_R). Alternativ kann die Schnürungstiefe indirekt durch eine Kraftangabe F berechnet werden.

Negative Werte von f_H bzw. F sind ebenfalls möglich. Damit können Kräfte simuliert werden, welche die Pnehülle an der Stelle f_R radial gleichmäßig nach außen drücken, gleichsam wie ein interner überdimensionierter Gummiring.

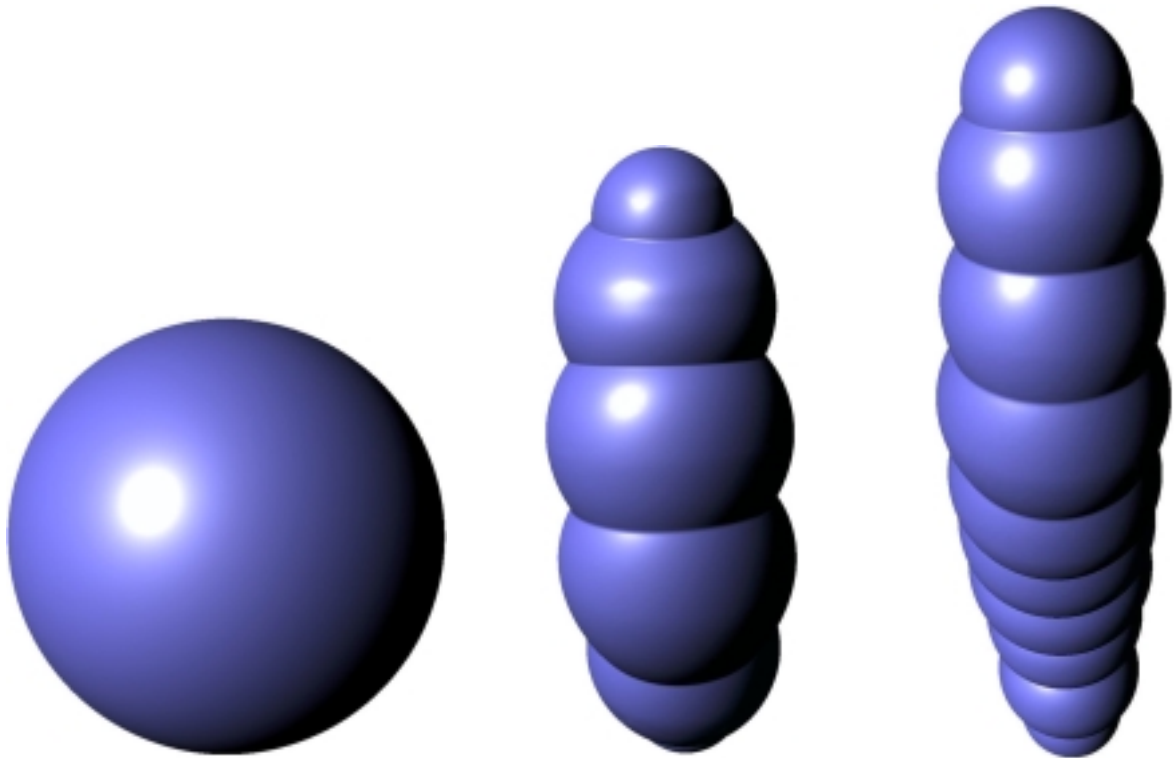


Abbildung 71: Beispiel für Constrictions

Auf einen kugelförmigen Pneu (links) werden erst fünf Constrictions angewandt, die den Pneu in 6 Segmente aufteilen (Mitte). In der Mitte jedes Innensegmentes wird jeweils wieder eine Constriction appliziert, so dass ein Pneu mit 10 Segmenten entsteht (rechts). Das Volumen bleibt währenddessen konstant.

Parameter	Maßeinheit	Typ	Variablenname	Default
longitudinale Position der Schnürung	Bruchteil der Zentrallinie (BKS)	fractional float	frPos	-
Umschaltflag zwischen direkter Angabe der Schnürungstiefe und indirekter Kraftangabe	-	bool	bIndirect	0
relative Schnürungstiefe	Bruchteil des Gesamtradius bei frPos (BKS)	fractional float	frDeepness	0
Schnürungskraft	Newton	float	fForce	0

Tabelle 10: Modellierungsparameter von Constriction

Modellierungsparameter des Quaoaring Design Patterns „Constriction“ mit jeweiliger Maßeinheit, internen Variablennamen und Defaultwert.

4.8.8 Conics

Eine *Conic* ist eine zur Zentrallinie orthogonale radiale Deformation eines beliebigen Pneus, die – im Gegensatz zur Constriction – breitflächig wirkt. Ein metaphorisches Analogon für die *Conic* wäre beispielsweise ein breites Band, welches einen wassergefüllten Ballon zusammenschnürt.

Eine *Conic* segmentiert automatisch den Pneu, analog wie die *Constriction* (siehe Kapitel 4.8.7). Auch bei diesem Quaoaring Design Pattern wird laterale Vertexgranularität des Pneus gleichmäßig auf alle entstehenden Segmente verteilt.

Mit *Conics* kann der Pneu u.a. „in die Länge gequetscht“ werden, um bikonische schlauchförmige Strukturen zu erstellen, wie z.B. Schlangen- und wurmartige Strukturen oder große Tentakeln.

Die vom Modellierer beeinflussbaren Parameter des *Conics* sind weitgehend identisch mit denjenigen von *Constriction*: Zur obligatorischen Angabe der longitudinalen Position der Schnü rungsmitte gesellt sich zusätzlich die relative Breite des *Conics*. Wie bei der *Constriction* benötigt die *Conic* die relative Schnü rungstiefe f_H . Statt der Schnü rungstiefe kann auch alternativ eine Kraftangabe F angegeben werden. Die durch diese Kraftanwendung resultierende Schnü rungstiefe wird dann indirekt berechnet. Negative Werte von f_H bzw. F sind ebenfalls möglich. Damit können Kräfte simuliert werden, welche die Pnehülle an der Stelle f_R radial gleichmäßig nach außen drücken, gleich einem internen überdimensionierten Gummischlauch.

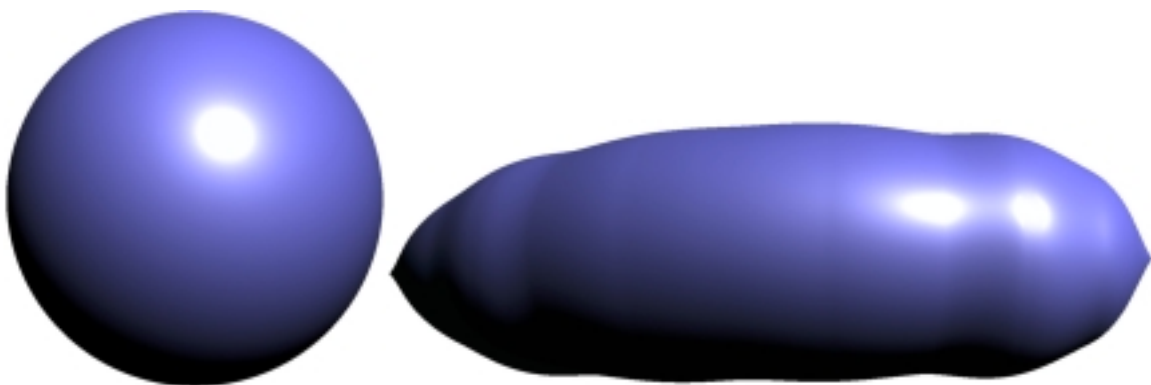


Abbildung 72: Beispiel eines Conics

Ein kugelförmiger Pneu (links) wird mittels fünf *Conics* in die Länge gequetscht. Es entsteht ein länglicher wurmartiger Pneu (rechts) mit dem gleichen Volumen.

Parameter	Maßeinheit	Typ	Variablenname	Default
longitudinale Position der Schnürung	Bruchteil der Zentrallinie (BKS)	fractional float	frPos	-
relative Breite des Schnürungsbandes	Bruchteil der Zentrallinie (BKS)	fractional float	frWidth	-
Umschaltflag zwischen direkter Angabe der Schnürungstiefe und indirekter Kraftangabe	-	bool	bIndirect	0
relative Schnürungstiefe	Bruchteil des Gesamtradius bei frPos (BKS)	fractional float	frDeepness	0
Schnürungskraft	Newton	float	fForce	0

Tabelle 11: Modellierungsparameter von Conic

Modellierungsparameter des Quaoaring Design Patterns „Conic“ mit jeweiliger Maßeinheit, internen Variablennamen und Defaultwert.

4.8.9 Contractile

Eine *Contractile* ist eine komplexe Deformationslinie, welche zwei Pnepunkte miteinander verbindet. Es kann sich dabei um zwei verschiedene Punkte ein und desselben Pneus handeln, es sind aber auch Verbindungen zwischen zwei verschiedenen Pneus möglich. Wird ein Stimulus auf die Contractile gesetzt, zieht die Contractile sich zusammen und lässt mechanische Kräfte an den beiden Ansatzpunkten wirken.

Die Beträge der zwei entstehenden Kräfte gleichen sich und sind umso größer, je höher der aktuelle Stimulus ist. Die Richtung der Kraft liegt jeweils tangential zu den Verläufen der Deformationslinie an den beiden Ansatzpunkten.

Ein biologisches Analogon für die Contractile ist die einzelne Muskelfaser, die je nach Nervenreizung (Stimulus) Kräfte an den beiden Ansatzpunkten ausübt.

Eine Contractile benötigt als Parameter zwei Ansatzflächen, gegeben als Bioflächen. Um den Verlaufsweg einer Contractile zu definieren, gibt der Modellierer n Stützstellen in der Form von Biovektoren an. Zwischen diesen Stützstellen wird der intermediäre Weg kubisch (via NURBS) interpoliert.

Als zusätzlicher Parameter der Contractile existiert die Übersetzungsrelation zwischen Stimulus und Kraft. Die Beträge der Kräfte an den Ansatzpunkten ist gleich der Hälfte des Produktes dieses Kraftumsetzungsparameters mit der aktuellen Stimulusstärke. Für dicke und damit potentiell starke Muskelfasern sollte daher ein hoher Wert für die Übersetzungsrelation gewählt werden, für dünne, potentiell schwache Muskelfasern dementsprechend ein niedriger Wert.

Parameter	Maßeinheit	Typ	Variablenname	Default
Ansatzbereich am Elternpneu Nr. 1	Bioflächen-Einheiten (BKS)	bioarea	baConnection1	-
Ansatzbereich am Elternpneu Nr. 2	Bioflächen-Einheiten (BKS)	bioarea	baConnection2	-
Umschaltflag zwischen direkter Angabe der Deformationstiefen und indirekter Kraftangabe	-	bool	bIndirect	1
Übersetzungsrelation zwischen Stimulus und Kraft	-	float	fTranslation	1
Anzahl n der Stützstellen	-	unsigned int	iPoints	
Stützstellen (n mal)	Biovektor mit BKS-Einheiten	biovector	bvPoint[]	-
Gewichtungsrelation der Stützstellen	-	fractional float	frWeight[]	0,5

Tabelle 12: Modellierungsparameter eines Contractiles

Modellierungsparameter des Quaoaring Design Patterns „Contractile“ mit jeweiliger Maßeinheit, internen Variablenamen und Defaultwert.

4.8.10 Relief

Ein Relief ist ein komplexes Auslenkungs- bzw. Kraftmuster auf der Oberfläche des Pneus, bei dem die jeweilige Auslenkungs- bzw. Kraftstärke durch die Pixel einer speziellen Bump Map Textur variiert werden kann. Es ist dasjenige Quaoaring Design Pattern, welches das Konzept der hydrostatischen Bump Maps verwendet. (siehe auch Kapitel 4.5)

Mit Reliefs lassen sich feingranulare Tiefenstrukturen der Oberfläche modellieren, wie beispielsweise Dornen und Stacheln, Noppen und Panzerungsstrukturen, Venen und Arterien, Runzeln und Fältchen, Pickel und Pusteln, Blasen und Beulen sowie leder- und hautartige Maserungen.

Für die Einstellung eines Reliefs auf einem Pneu wird ein Verweis auf eine Bump Map Textur benötigt. Zusätzlich sind zwei Parameter obligatorisch, einerseits dasjenige Areal, gegeben als Biofläche, auf der das Relief sich ausbreitet, und andererseits die maximale relative Auslenkung f_H (relativ des zur Zentrallinie orthogonalen Ausgangsradius des Pneus am jeweiligen Pixel). Alternativ kann die maximale Auslenkung indirekt durch eine maximale Kraftangabe F berechnet werden.

Negative Werte von f_H bzw. F sind ebenfalls möglich, damit können Kräfte simuliert werden, welche die Pnehülle am jeweiligen Pixel entlang der Oberflächennormale nach außen drücken.

Ist das Konkurrenzflag $f_{Competition}$ eingeschaltet, verdrängen sich die Reliefs gegenseitig auf derjenigen Oberfläche, welche die Schnittmenge ihrer zwei Bioflächen bildet. Sollten mehrere Reliefs übereinander gelagert sein, konkurrieren dann in diesem Bereich die „Feldwerte“ ihrer morphischen Felder. Hierdurch erhält man für gewöhnlich biologisch anmutendere Strukturen als bei einfacher Überlagerung. Bei ausgeschaltetem Konkurrenzflag werden die Auslenkungswerte an jedem Oberflächenpunkt einfach addiert.

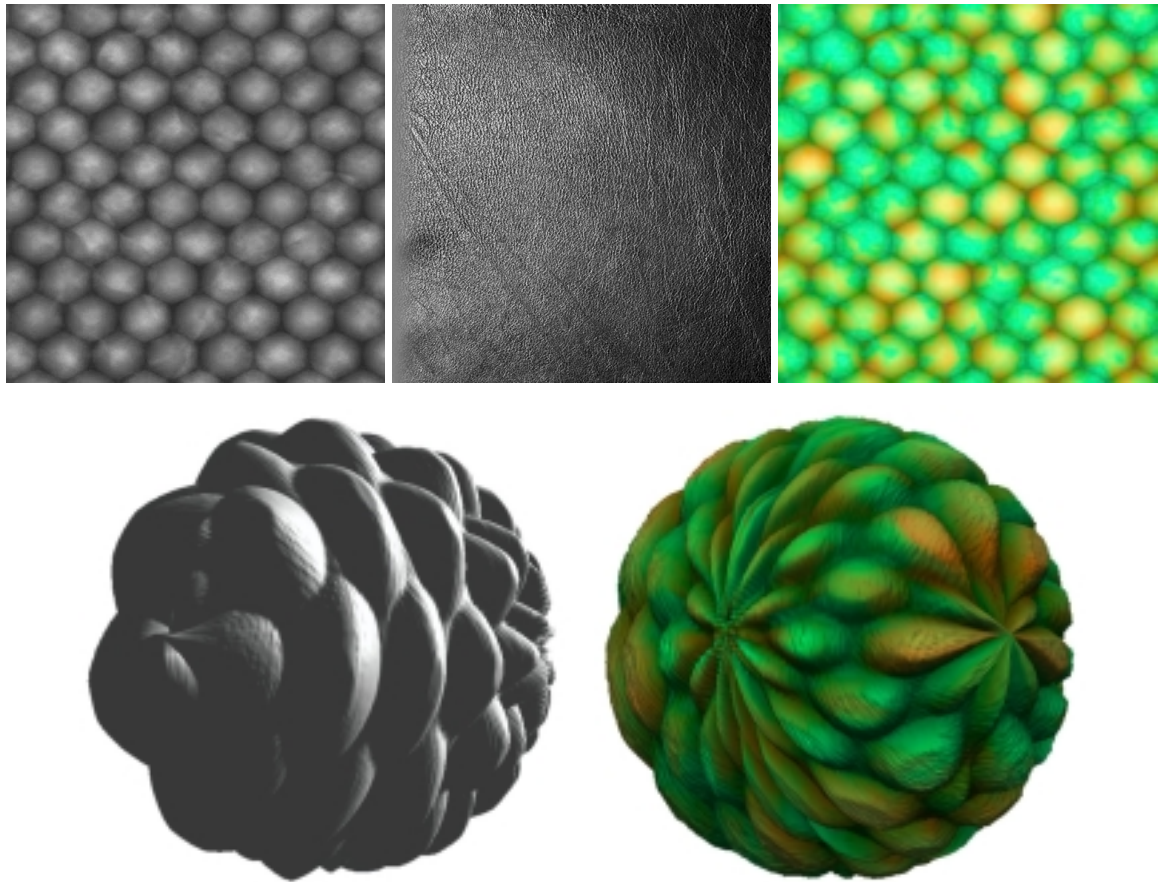


Abbildung 73: Beispiel für die Beziehung zwischen Texturen, Skins und Reliefs

Links oben: Eine wabenförmige Bump Map Textur. Mitte oben: eine lederartige Bump Map Textur. Rechts oben: eine wabenförmige Skin-Textur, deren Farbmuster sich mit dem Höhenmuster der ersten Bump Map Textur deckt. Links unten: kugelförmiger Pneu, auf der die beiden Bump Map Texturen direkt als Reliefs appliziert wurden. Bei der wabenförmigen Bump Map Textur wurde eine große Biofläche und eine hohe maximale Aulenkung eingestellt, so dass große hexagonal angeordnete Noppen auf dem Pneu entstehen, bei der zweiten Bump Map Textur wurden die Parameter für die Biofläche und die maximale Auslenkung dagegen klein gehalten, so dass über das Noppenrelief eine weitere feine „Riffelung“ überlagert wird. Rechts unten: der gleiche Pneu, bei dem zusätzlich ein Skin mit der Textur von rechts oben als Vater des Reliefs von links oben appliziert wurde. Es ist sichtbar, dass die Bioflächenwerte des Skins automatisch auf das Relief übertragen wurden, so dass sich das Farbmuster der Skin-Textur mit dem Höhenmuster der Bump Map Textur deckt. Die Parameter des Lederreliefs bleiben von dieser Hierarchieeinfügung unbeeinflusst.

Parameter	Maßeinheit	Typ	Variablenname	Default
Ausdehnung des Reliefs	Bioflächen- Einheiten (BKS)	bioarea	baExtention	-
Pfadangabe einer Bump Map Texturdatei	-	char [MAX- PATH]	szTexture	NULL
Konkurrenzflag	-	bool	bCompetition	0
Symmetrieflag	-	bool	bSymmetry	0
Umschaltflag zwischen direkter Angabe der Relieftiefe und indirekter Kraftangabe	-	bool	bIndirect	0
maximale relative Relieftiefe	radiale BKS- Einheiten	fractional float	frDeepness	0
maximale Reliefkraft	Newton	float	fForce	0

Tabelle 13: Modellierungsparameter eines Reliefs

Modellierungsparameter des Quaoaring Design Patterns „Relief“ mit jeweiliger Maßeinheit, internen Variablennamen und Defaultwert.

4.8.11 Skin

Ein Skin ist eine spezielle Oberflächentexturierung des Pneus, mit der sich komplexe Oberflächenschattierungen, Einfärbungen und Hautmusterungen erzeugen lassen.

Für die Definition eines Skins wird diejenige Biofläche auf der Pnehülle benötigt, auf der sich die Textur ausbreiten soll.

Zusätzlich bedarf es eines Verweises auf eine Textur. Dies geschieht mit Hilfe einer einfachen Pfadangabe. Es sind Verweise auf JPG-, TGA- und TIF-Dateien möglich. Der Modellierer sollte beim Texturdesign darauf achten, dass sich die Textur stets radial um den Mittelpunkt der Biofläche ausbreitet, wobei alle Pixel der rechten Pixelpalte zusammen auf den Mittelpunkt und die der linken Pixelpalte radial auf die Ränder der Biofläche gemappt werden. Dazu wird eine geeignete organische Texturbibliothek angelegt, bei der diese Besonderheiten des Mappings berücksichtigt worden sind.

Sollten sich die Bioflächen zweier benachbarter Skins überschneiden, kann ihre Interaktion auf der Schnittfläche auf zweierlei Weise erfolgen, einerseits durch einfache Überlagerung

der Texturen und andererseits durch die Konkurrenz der „Feldwerte“ ihrer morphischen Felder. Die Angabe, wie die Interaktion erfolgen soll, erfolgt mittels eines Konkurrenzflags. Per Default ist dieses Konkurrenzflag eingeschaltet, da die gegenseitigen Verdrängungseffekte, die infolge der Konkurrenz der Feldwerte resultieren, für gewöhnlich biologisch anmutendere Strukturen erzeugen als bei einfacher Überlagerung.

Wie bei den anderen Oberflächen- und Modifikationsobjekten existiert auch bei Skins ein Symmetrieflag. Ist es eingeschaltet, breitet sich ein spiegelbildlicher Skin symmetrisch zum Planum Medianum aus.

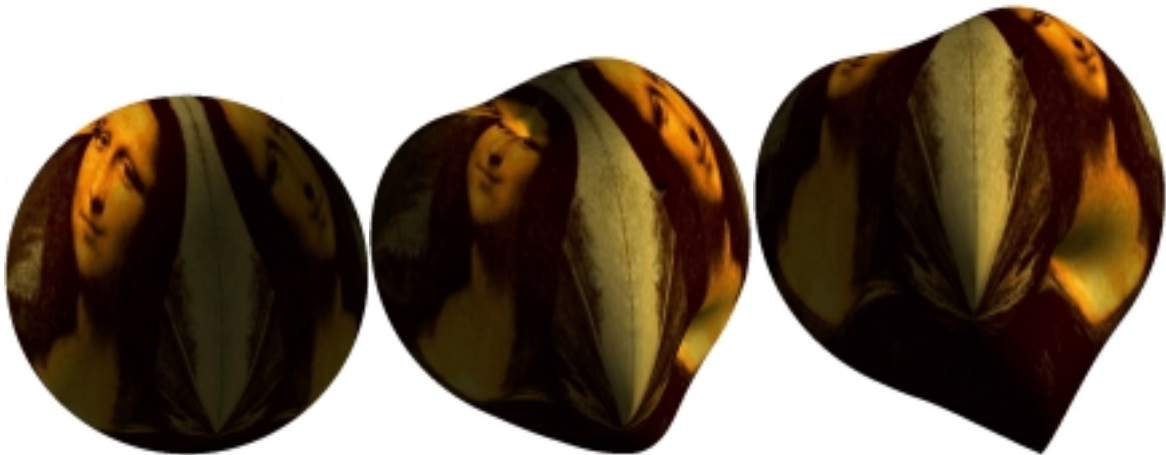


Abbildung 74: Beispiel eines Skins mit seinem Verhalten

Das Bild „Mona Lisa“ (links oben) wird mit Hilfe des Quaoaring Design Patterns „Skin“ auf einen kugelförmigen Pneu gemappt (links unten), wobei die Biofläche des Skins (0,5;60;1;0,7;240;0) beträgt und das Symmetrieflag eingeschaltet ist. Anschließend wird der Pneu mit Squeezes sukzessive zu einer Herzform aufgeblasen. (Mitte und rechts unten) Es ist ersichtlich, dass die Textur dort mitwächst, wo der Pneu expandiert und dort schrumpft, wo der Pneu kollabiert.

Parameter	Maßeinheit	Typ	Variablenname	Default
Ausdehnung des Skins	Bioflächen- Einheiten (BKS)	bioarea	baExtention	-
Symmetrieflag	-	bool	bSymmetry	0
Pfadangabe einer Texturdatei	-	char [MAX- PATH]	szTexture	NULL
Konkurrenzflag	-	bool	bCompetition	0
Stärke des morphischen Feldes	Newton	float	fStrength	1

Tabelle 14: Modellierungsparameter eines Skins

Modellierungsparameter des Quaoaring Design Patterns „Skin“ mit jeweiliger Maßeinheit, internen Variablennamen und Defaultwert.

4.8.12 Hair

Ein Hair ist ein Quaoaring Design Pattern, welches filigrane fraktale Strukturen auf der Pneuoberfläche erzeugt.

Mit Hairs lassen sich Schnurr-, Körper- und Kopfhaare, Pelze, Flaum- und Federkleider sowie Borsten erstellen.

Die Parameter von Hair sind nahezu identisch mit denjenigen von Skin. So ist die Angabe der Biofläche obligatorisch, auf der sich die Strukturen ausbreiten, optional kann dazu noch ein Symmetrieflag gesetzt werden. Es existiert ebenfalls ein Konkurrenzflag, bei dessen Setzen zusätzlich die Feldstärke des morphischen Feldes angegeben werden kann.

Der einzige Unterschied besteht im Dateityp, auf den der Verweis zeigt, denn neben Texturdateien (JPG, TGA, TIF) können auch Vertex Shader Files (VCG) angegeben werden.

(Das Quaoaring Design Pattern „Hair“ ist zurzeit zwar implemetiert, kann zum Zeitpunkt der vorliegenden Arbeit noch nicht voll ausgetestet werden, da die aktuelle RealiMation-Version keine Shader-Sprachen unterstützt. Daher wird in der aktuellen Version mit Haar- und Federtexturen ausgeholfen, die anstatt des Shader-Files angegeben werden.)

Parameter	Maßeinheit	Typ	Variablenname	Default
Ausdehnung des Hairs	Bioflächen- Einheiten (BKS)	bioarea	baExtention	-
Symmetrieflag	-	bool	bSymmetry	0
Pfadangabe einer Texturdatei oder eines Vertex-Shader-Files	-	char [MAX- PATH]	szFile	NULL
Konkurrenzflag	-	bool	bCompetition	0
Stärke des morphischen Feldes	Newton	float	fStrength	1

Tabelle 15: Modellierungsparameter von Hair

Modellierungsparameter des Quaoaring Design Patterns „Hair“ mit jeweiliger Maßeinheit, internen Variablennamen und Defaultwert.

4.8.13 Material

Materials sind Quaoaring Design Patterns, welche sowohl die optischen als auch die mechanischen Eigenschaften der Pneuhülle bzw. des Skins definieren, auf die das Material appliziert wird. Mit Materials lassen sich unterschiedliche organische Materialien und Gewebearten bestimmen, wie z.B. Holz, menschliche Haut, Fettgewebe, Leder, Chitinpanzer, Horn, Knochensubstanz und Zahnschmelz.

Mit dem Transparenzparameter lässt sich die Durchsichtigkeit der Pneuhülle bestimmen. Der Transparenzparameter ist per Default auf 0 gesetzt. Setzt der Modellierer den Transparenzparameter auf einen anderen Wert, so wird automatisch das Backface Culling ausgeschaltet, so dass auch die Innenseite des Pneus sichtbar wird.

Der Modellierer kann auch die Grundfarbe als RGB-Wert und diverse Reflexionseigenschaften der Pneuhülle einstellen, wie die ambiente Reflektivität, die diffuse Reflektivität, die spekulare Reflektivität sowie die „Schärfe“ der spekularen Highlights.

Für spezielle Organismen mit lumineszierenden oder fluoreszierenden Oberflächen, wie z.B. Glühwürmchen oder Neonfische, kann auch eine Leuchtstärke des Materials definiert werden. Per Default ist fEmission gleich Null, die Pneuoberfläche leuchtet also nicht.

Parameter	Maßeinheit	Typ	Variablenname	Default
Transparenzparameter	-	fractional float	frTransparency	0
Farbe	RGB-Anteile	colour	colMaterial	-
ambiente Reflektivität	-	fractional float	frAmbient	0,7
diffuse Reflektivität	-	fractional float	frDiffuse	0,3
spekulare Reflektivität	-	fractional float	frSpecular	0,5
„Schärfe“ der spekularen Highlights	-	fractional float	frSharpness	0,5
Leuchtkraft	-	fractional float	frEmission	0
Viskositätsparameter	-	fractional float	frViscosity	0,5
Haftreibungparameter	-	fractional float	frFrictionH	0,3
Gleitreibungparameter	-	fractional float	frFrictionS	0,3

Tabelle 16: Modellierungsparameter eines Materials

Modellierungsparameter des Quaoaring Design Patterns „Material“ mit jeweiliger Maßeinheit, internen Variablenamen und Defaultwert.

Neben diesen optischen lassen sich auch mechanische Attribute der Pneuohlle einstellen. Der Viskositätsparameter gibt die „Weichheit“ der Pneuohlle an, ein kleiner Viskositätswert (0 bis 0,2) sollte für harte Materialien wie z.B. Knochen, Zähne, Holz und Chitin verwendet werden, ein hoher Viskositätswert (ca. 0,7 bis 0,9) für weiche

Materialien wie z.B. die Unterhaut und das Fettgewebe, mittlere Viskositätswerte für zähe Materialien wie beispielsweise dicke Strukturen der Epidermis und Leder.

Zwei weitere Oberflächenparameter betreffen die Haft- und Gleitreibung des Materials. Kleine Werte dieser Parameter (0 bis 0,2) sollten für glitschige Materialien, mittlere Werte für glatte, hohe Parameter (0,5 bis 0,7) für raue und extrem hohe Werte (0,7 bis 1,0) für klebrige Materialien verwendet werden. Beispielsweise würde die glitschige Haut eines Frosches Werte unter 0,2 und seine klebrige Zunge Werte um 0,8 bekommen. Alle Oberflächenwerte sind vom Typ „fractional float“ und werden in einem Bereich zwischen 0 und 1 angegeben.

4.8.14 Particlesource

Particlesources sind Quaoaring Design Patterns, welche eine Emissionsquelle von Partikelströmen definieren.

Mit Particlesources lässt sich ein Ausstoß biologischen Materials aus dem Organismus erzeugen, wie beispielsweise blutende Wunden, das Spucken, das Transpirieren, das Urinieren und Ejakulieren sowie tierspezifische Effekte wie beispielsweise die Tintenemission von Tintenfischen. Ebenfalls können Ausstöße von Wasser oder Wasserdampf in die Luft generiert werden, dazu gehören die Wasserfontänen von Walen oder der winterliche Nebelhauch beim Ausatmen. Vice versa gilt dies auch für Luftblasen im Wasser bei allen wasserlebenden Organismen. Auch nichtbiologische Spezialeffekte von Fabelwesen wie beispielsweise das Feuerspeien von Drachen, der weiße Glühnebel um Einhörner und die Aura von Engeln sind mit Particlesources erzeugbar.

Particlesources können auch für die Simulation des umgebenden Mediums „missbraucht“ werden, wenn der Bioraum der Emissionsquelle weit größer als der Gesamtbioraum des Organismus gewählt wird. Eine solche Partikelquelle hat den Vorteil, dass sie sich immer mit dem Organismus mitbewegt, und somit keine unnötigen Partikel erzeugt werden, die mit dem Organismus nicht kollidieren können. Beispielsweise wurden für das Protoplesiosaurier-Exponat im Darmstädter Cybernarium (siehe im Anhang Kapitel 9.2.5) solche Particlesources für das wässrige Medium definiert, welche u.a. Luftblasen, Plankton und Algenfelder erzeugen, durch die der Protoplesiosaurier hindurch schwimmt.

Die Partikel werden in einem vom Modellierer zu definierenden Bioraum des Organismus mit initialen Startgeschwindigkeiten und Drehmomenten ausgesandt. Die

Stärke der Geschwindigkeiten bewegt sich dabei in einem Bereich, dessen Werte vom jeweiligen Partikeltyp abhängen. Das Gleiche gilt für die Drehmomente.

Die Vektoren dieser initialen Partikelbewegungen sind im biologischen Koordinatensystemen der Elternpneus definiert, was zu einem biologisch stringenten Auswurf der Partikel führt. Wird z.B. das Spucken eines Lamas mit Hilfe von Particlesources modelliert, so wird das Lama automatisch nach rechts spucken, wenn das Lama seinen Kopf nach rechts wendet, und auch nur dann, wenn die Schnute des Lamas geöffnet ist. Dies vereinfacht und beschleunigt das Modellieren und darin liegt auch der Unterschied zu „normalen“ Partikelemittlern in konventionellen Partikelsystemen.

Die Partikel verhalten sich nach der Emission weitgehend autonom, d.h. sie haben einen eigenen Lebenszyklus, eine Masse, ein spezifisches Gewicht, einen Luft-, Wasser- und Reibungswiderstand, eine gegenseitige Abstoßung bzw. Anziehung, und viele weitere physikalische und temporäre Parameter. Auch verschiedene optische Parameter wie die Leuchtkraft, die Leuchtfarbe und die Transparenz können variabel über die Zeit verändert werden. So kann zum Beispiel die Leuchtkraft von Funken nach ihrer Emission asymptotisch gegen Null sinken. Particlesources können auch gleichzeitig oder temporär versetzt verschiedene Partikelarten aussenden. Die mannigfaltigen Einstellmöglichkeiten machen Particlesources sehr flexibel, sie haben allerdings auch den Nachteil, dass sich das Modellieren sehr langwierig gestalten kann, da mitunter über 150 Parameter einzustellen sind. Um das Modellieren zu beschleunigen, wurde eine Parameterbibliothek für Standardemitter eingerichtet; diese beinhaltet folgende Emissionsquellen:

- Allgemeine organische Tieremissionen: *Urinstrahl, Exkrememente, Spucke, Ejakulat*
- Spezielle organische Tieremissionen: *Tinte, Wasserfontäne, Schleim*
- Ausdünstungen: *Schweiß, Winterhauch*
- Emissionen aus Wunden: *Blut, Eiter*
- Partikel im wässrigem Medium: *Algenfelder, Plankton, Luftblasen*
- Partikel in der Luft: *Sandsturm, Nebel, Wolken*
- Spezialeffekte für Fabelwesen: *Feuer, Rauch, Dampf, Funken, Glühnebel, Aura, Zauberstrahl*

Parameter	Maßeinheit	Typ	Variablenname	Default
Ausdehnung des Particlesources	Bioraum-Einheiten (BKS)	biospace	bsExtention	-
Hauptrichtung des Partikelstromes	Biovektor (BKS-Einheiten)	bioovector	bvDirection	-
Stärke des Partikelstromes	Partikel pro Sekunde	float	fStrength	100
Predefinitionsflag	-		bPredefined	0
je nach Partikelart ca. 150 verschiedene optische, physikalische und temporäre Parameter	siehe Quaoariung Manual S.20ff	siehe Quaoaring Manual S.20ff	siehe Quaoaring Manual S.20ff	siehe Quaoaring Manual S.20ff
ID-Name der vorgefertigten Emitterquelle	-	char []	szEmitter	NULL

Tabelle 17: Modellierungsparameter von Particlesource

Modellierungsparameter des Quaoaring Design Patterns „Particlesource“ mit jeweiliger Maßeinheit, internen Variablennamen und Defaultwert.

4.9 Modellierung mit den Quaoaring Design Patterns

4.9.1 Modellierungsbefehle

Dem Modellierer stehen zur Behandlung der Quaoaring Design Patterns eine Reihe von *Modellierungsbefehlen* zur Verfügung:

- *Make*: Erzeugt ein neues Quaoaring Design Pattern; gleichzeitig werden die Werte des neuen Quaoaring Design Patterns initialisiert. Gibt der Modellierer keine expliziten Initialisierungswerte an, werden Defaultwerte genommen.
- *Kill*: Löscht ein bestehendes Quaoaring Design Pattern
- *Add*: Hängt ein Quaoaring Design Pattern A an ein anderes Quaoaring Design Pattern B in der Objekthierarchie an. A ist danach ein Kindobjekt von B.
- *Sub*: Koppelt ein Quaoaring Design Pattern A von einem anderen Quaoaring Design Pattern B in der Objekthierarchie ab
- *Change*: Verändert einzelne Parameter eines Quaoaring Design Patterns
- *MakeOrg*: Erzeugt eine neue Instanz eines Organismus
- *KillOrg*: Löscht eine neue Instanz eines Organismus
- *MetaOrg*: Fügt Metainformationen zu einer Organismenkonstruktion hinzu, wie z.B. Name, Hauptverbreitungsgebiet, Zeitalter des ersten Auftretens, der maximalen Verbreitung und evtl. des Aussterbens, Ernährungsgewohnheiten, Besonderheiten etc.

- *AddPhylo*: Hängt eine Instanz A eines Organismus in der Evolutionshierarchie zu einer bestehenden Organismeninstanz B an. Der Organismus A ist dann ein evolutionärer Nachfolger von B. Pro evolutionärem Stadium darf jeweils nur eine hierarchische Verbindung zur evolutionären Vorgängerkonstruktion existieren. Wird diese Bedingung nicht eingehalten, wird der Befehl ignoriert, da ansonsten Inkonsistenzen in Zusammenhang mit verschiedenen Wachstumsstadien auftreten könnten.
- *SubPhylo*: Koppelt eine Instanz A eines Organismus in der Evolutionshierarchie von einer bestehenden Organismeninstanz B ab.
- *AddOnto*: Hängt eine Instanz A eines Organismus in der Wachstumskette zu einer bestehenden Organismeninstanz B an. Der Organismus A ist dann Wachstumsnachfolger von B.
- *SubOnto*: Koppelt eine Instanz A eines Organismus in der Wachstumskette von einer bestehenden Organismeninstanz B ab.

4.9.2 Visualisierungsbefehle

Neben den Modellierungsbefehlen steht sowohl dem Modellierer als auch dem Betrachter eine Reihe von *Visualisierungsbefehlen* zur Verfügung. Im Gegensatz zu den Modellierungsbefehlen modifizieren die Visualisierungsbefehle nicht die Konstruktionsdaten, sondern wechseln nur die Sichtweise auf den Organismus oder das aktuelle Evolutions- oder Wachstumsstadium des Organismus:

- *NextPhylo*: Wechselt zur nächsten Organismenkonstruktion in der Evolutionslinie. Falls keine Nachfolgekonzstruktion existiert, wird dieser Befehl ignoriert. Existieren mehrere Nachfolgekonzstruktionen muss sich der Benutzer explizit für einen Nachfolger entscheiden. Mit *NextPhylo* wird die Evolution sozusagen vorangetrieben, um die Entwicklung der Arten zu erleben.

- *PrevPhylo*: Wechselt zur vorherigen Organismenkonstruktion in der Evolutionslinie. Falls keine Vorgängerkonstruktion existiert, wird dieser Befehl ignoriert. Mit *PrevPhylo* wird die Evolution sozusagen zurückgedreht, um paläontologische Tier- oder Pflanzenarten anzuschauen. *PrevPhylo* ist somit der inverse Befehl zu *NextPhylo*.
- *NextOnto*: Wechselt zur nächsten Organismenkonstruktion in der Wachstumskette. Mit diesem Befehl wächst und entwickelt sich der Organismus weiter, bis er schließlich stirbt.
- *PrevOnto*: Wechselt zur vorherigen Organismenkonstruktion in der Wachstums- bzw. Evolutionslinie. Falls keine Vorgängerkonstruktion existiert, wird dieser Befehl ignoriert. *PrevOnto* ist somit der inverse Befehl zu *NextOnto*.

Mit den Befehlen *NextPhylo* und *PrevPhylo* kann der Benutzer frei den „Baum der Evolution“ traversieren. Die Organismen mutieren dabei sinnfällig von einem evolutionären Stadium zum nächsten. Mit *NextOnto* und *PrevOnto* kann er zusätzlich verschiedene Wachstumsstadien des aktuellen Organismus betrachten. Der Wechsel von einem Status zum nächsten erfolgt dabei nicht abrupt, sondern geht in einer vom Modellierer definierten Zeitspanne langsam vor sich, so dass der Betrachter das Wachstum bzw. die evolutionären Veränderungen betrachten kann. Die Veränderungen folgen dabei in jedem Zwischenschritt den Regeln der Hydrodynamik. Der Organismus verhält sich beim Gestaltwechsel also biologisch stringent und sinnfällig. Dieses *hydrodynamische Morphing* unterscheidet sich also grundlegend vom optischen Morphing.

Parallel zu diesen Befehlen für das hydrodynamische Morphing existieren auch *Kamerasteuerungsbefehle*. Mit diesen Befehlen kann der Benutzer die Kamera interaktiv an jede gewünschte Stelle innerhalb und außerhalb des Organismus positionieren und ausrichten. Dies wird durch die Echtzeitfähigkeit des Quaoarings ermöglicht.

Erfahrungsgemäß hat es sich am einfachsten für die Benutzersteuerung erwiesen, die Kamera per Default auf das virtuelle Zentrum des Organismus zu fokussieren, welches mit dem Mittelpunkt des kugelförmigen Ursprungspneus zusammenfällt. Der Benutzer kann jedoch die Position dieses *Fokussierungspunktes* jederzeit innerhalb der räumlichen Grenzen des Organismus ändern. Die Positionierung der Kamera erfolgt radial um diesen

Fokussierungspunkt. Das heißt, die Kamera bewegt sich auf der Oberfläche eines gedachten *Kameraglobus*, dessen Radius mit zwei separaten Befehlen verändert werden kann. Mit dieser Kamertechnik wird das „Lost in Virtual Space“- Syndrom verhindert, da der Benutzer immer ein Teil des Organismus auf dem Bildschirm sieht und somit stets eine Orientierungshilfe hat. Es existieren folgende Kamerasteuerungsbefehle:

- *Nord*: Lässt die Kamera auf dem Kameraglobus hoch gleiten. Hat der Benutzer den Nordpol erreicht, werden weitere Nord-Befehle ignoriert.
- *South*: Lässt die Kamera auf dem Kameraglobus herunter gleiten. Hat der Benutzer den Südpol erreicht, werden weitere Nord-Befehle ignoriert.
- *East*: Lässt die Kamera auf dem Kameraglobus nach rechts gleiten.
- *West*: Lässt die Kamera auf dem Kameraglobus nach links gleiten.
- *Expand*: Erweitert den Radius des Kameraglobus. Ist die Länge der Zentrallinie des Hauptneus des Organismus kleiner als ein Hundertstel des Radius, wird ein weiterer Expand-Befehl ignoriert, um den Organismus nicht aus den Augen zu verlieren.
- *Collaps*: Verkleinert den Radius des Kameraglobus.
- *Caudal*: Bewegt den Kamerafokus in Richtung des Schwanzes, der Wurzel oder des Hinterteils. Dies geschieht parallel zur undeformierten Zentrallinie des Hauptneus.
- *Cranial*: Bewegt den Kamerafokus relativ zum Organismus in Richtung des Kopfes, der Krone oder des Vorderteils. Dies geschieht parallel zur undeformierten Zentrallinie des Hauptneus.
- *Dextral*: Bewegt den Kamerafokus relativ zum Organismus nach rechts.

- *Sinistral*: Bewegt den Kamerafokus relativ zum Organismus nach links.
- *Dorsal*: Bewegt den Kamerafokus relativ zum Organismus in Richtung des Rückens oder der Rückseite.
- *Ventral*: Bewegt den Kamerafokus relativ zum Organismus in Richtung des Bauches oder der Vorderseite.
- *CameraReset*: Setzt Position und Fokussierungspunkt zurück auf die Initialposition.

Um eine richtige Interaktivität zu erreichen, sind die Befehle jeweils durch einen einzigen Tastendruck ansprechbar. Der Bediener drückt einfach solange die gewünschten Tasten, bis seine gewünschte Kameraposition bzw. sein gewünschter Evolutions- oder Wachstumsstatus erreicht ist. Dabei kann er auch mehrere Tasten parallel drücken. Zum Beispiel könnte er die NextOnto-Taste drücken, um das Tier wachsen zu lassen, während er gleichzeitig die Cranial-Taste bedient, um die Kamera in Richtung des Kopfes zu schwenken. Die Befehle sind auch hinsichtlich ihrer Geschwindigkeit quantisierbar: Wird mehrfach direkt hintereinander oder lange auf eine Taste gedrückt, so beschleunigt sich der entsprechende Vorgang.

4.9.3 Beispiel für einen Modellierungsvorgang eines Organismus

Im Folgenden soll ein konkretes Beispiel vorgestellt werden, wie mit Hilfe der Modellierungsbefehle in Verbindung mit den Quaoaring Design Patterns ein Organismus modelliert werden kann. Als Beispiel wurde die Herstellung desjenigen Protoplesiosauriers ausgewählt, welcher mit Hilfe des Quaoarings für eine Dauerausstellung des Darmstädter Cybernariums erstellt wurde und voraussichtlich noch bis Ende 2005 dort zu besichtigen ist:

Im Vorfeld der Modellierung wurden alle für die Modellierung relevanten Informationen über die Protoplesiosauri gesammelt, wie Form, Farbgebung, Strukturierung der Haut und Verwandtschaftsbeziehungen zu anderen Tierarten. Die Verwandtschaftsbeziehungen zu anderen Tierarten sind wichtig, um eventuelle Vorgängertiere zu finden, die schon

modelliert wurden. Diese könnten dann als Ausgangstier für die Modellierung verwendet werden, um Arbeit zu sparen. In unserem konkreten Fall existierte kein evolutionärer Vorgängerorganismus, der schon modelliert gewesen wäre. Aber es gab durchaus ein modelliertes Fabeltier, was dem Protoplesiosaurier von der Grundkonstruktion her recht ähnlich war. Es war in der Datei „WirbeltierZuMeeresmonster.xml“ abgespeichert.

Es galt nun mit Hilfe von geeigneten Modellierungsbefehlen und Quaoaring Design Patterns das Meeresmonster zum Protoplesiosaurier zu mutieren. (siehe Abbildung 75 auf Seite 230)

Dafür wurde auf den Pneu eine zusätzliche negative Conic „Dickrumpf“ appliziert, um das Meeresmonster dicker aussehen zu lassen. Die spitze Schnauze wurde mit einer neuen Squeeze „AufDieSchnauze“ platt gemacht. Der dünne Schwanz, welcher beim Meeresmonster mit der Squeeze „Schlankschwanz“ realisiert war, wurde rückgängig gemacht. Die Pelzhaare „GemaserterPelz“ wurden vom Rumpf und den 4 Flossen entfernt und dem Protoplesiosaurier stattdessen eine neue Haut „Grauhaut“ und ein Relief „Schwimmrillen“ verpasst. Zusammenfassend wurden also folgende Aktionen ausgeführt:

1. Make Conic *Dickrumpf*, *frPos=0.5*, *frWidth=0.5*, *frDeepness=-0.8*
2. Add *Dickrumpf Pneu*
3. Make Squeeze *AufDieSchnauze*, *baExtention:=(c_R=1,0; c_A=0; c_H=1; r_R=0,02; r_A=2π; r_H=0)*, *bSymmetry=true*, *frExcentricity=0.01*
4. Add *AufDieSchnauze Pneu*
5. Sub *Schlankschwanz Rumpf*
6. Kill *Schlankschwanz*
7. Sub *GemaserterPelz Rumpf*
8. Sub *GemaserterPelz FlosseLinksHinten*
9. Sub *GemaserterPelz FlosseRechtsHinten*
10. Sub *GemaserterPelz FlosseLinksVorne*
11. Sub *GemaserterPelz FlosseRechtsVorne*
12. Kill *GemaserterPelz*
13. Make Skin *Grauhaut*, *baExtention:=(c_R=0,5; c_A=π; c_H=1; r_R=0,7; r_A=2π; r_H=0)*, *bSymmetry=true*, *szTexture=Grauhaut.jpg*
14. Add *Grauhaut Rumpf*
15. Add *Grauhaut FlosseLinksHinten*
16. Add *Grauhaut FlosseRechtsHinten*

17. Add *Grauhaut FlosseLinksVorne*
18. Add *Grauhaut FlosseRechtsVorne*
19. Make Relief *Schwimmrillen*, *frDeepness=0.2*, *szTexture=Schwimmrillen.jpg*,
20. Add *Schwimmrillen Grauhaut*
21. MakeOrg *Protoplesiosaurier*

Das Ganze wurde nun abgespeichert, die XML-Datei sieht dann folgendermaßen aus:

```
<?xml version="1.0"?>
<QuaoaringScript>
  <Make      object="Conic"
             name="Dickrumpf"
             frPos="0.5" frWidth="0.5" frDeepness="-0.8" />
  <Add      from="Dickrumpf"
            to="Rumpf"/>
  ...
  <Make      object="Relief"
             name="Schwimmrillen"
             szTexture="Schwimmrillen.jpg" frDeepness="0.2" />
  <Add      from="Schwimmrillen"
            to="Grauhaut"/>
  <MakeOrg   name="Protoplesiosaurier"/>
  <MetaOrg   nameMutation="MeeresmonsterZuProtoplesiosaurier"
             nextMutation="ProtoplesiosaurierZuPlesiosaurier"
             timeMutationSeconds="3.0" .../>
</QuaoaringScript>
```

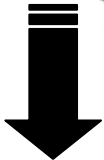
Nun ist der Protoplesiosaurier fertig. Der ganze Modellierungsprozess sollte interaktiv nur einige Minuten dauern. Eine Animation ist nicht erforderlich. Sind die Protoplesiosaurier fertig modelliert, können nun mit Hilfe von MakeOrg verschiedene Instanzen der Tiere gebildet werden. Sie können dann noch optional in einer geeigneten virtuellen Umgebung ausgesetzt werden, wie beispielsweise einem virtuellen Urmeer. (siehe Abbildung 75 unten)

Meeres-
monster



Modifizierung durch
Modellierungsbefehle und
Quoaring Design Patterns

Proto-
plesiosaurier



Instanziierung & Aussetzen
in die virtuelle Umgebung

virtuelles
Urmeer mit
Protoplesio-
sauriern



Abbildung 75: Modellierung eines Protoplesiosauriers

Modellierungsbeispiel eines Protoplesiosauriers. Oben: Zuerst wird ein geeignetes Tier gesucht, welches als Vorläufertier fungieren könnte. In diesem Fall war es das obige Meeresmonster. Mitte: Mit Modellierungsbefehlen kombiniert mit den Quoaring Design Patterns wird das Meeresmonster interaktiv zum Protoplesiosaurier mutiert. Unten: Am Schluss können mehrere Instanzen des Protoplesiosauriers gebildet werden und die Tiere in eine virtuelle Umgebung ausgesetzt werden.

5 Implementierungsaspekte

Electronic Technology is not intuitive.

Mechanical Technology was intuitive.

Gordon Moore

5.1 Auswahl der Hilfssoftware

Vor dem Beginn der Implementierungsarbeiten für das Quaoaring-Software-Framework musste jeweils über eine Programmiersprache inklusive Compiler, eine oder verschiedene Basis-Graphik-API, einen Szenengraphen und eventuell weiteren Programmierertools entschieden werden.

Ein Szenengraph basiert jeweils auf einer Basis-Graphik-API bzw. mehreren davon. Er bietet meist eine Schnittstelle zu einer bestimmten Programmiersprache an. Die Programmiersprache limitiert wiederum die Auswahl an Compilern und Programmierertools. Es ergibt sich demnach folgende Implikationsreihe:

Szenengraph → Graphik-APIs → Programmiersprache → Compiler → Programmierertools

Somit stand die Auswahl des Szenengraphen ganz am Anfang. Ihr musste ein besonderes Augenmerk gewidmet werden. Es wurden 53 Szenengraphen in einem dreistufigen Auswahlverfahren evaluiert. 33 Szenengraphen wurden nach einer ersten Begutachtung wegen offensichtlicher eklatanter Mängel herausgefiltert. Diese Ausschlusskriterien waren zu hohe Kosten, Unverfügbarkeit, mangelnde Unterstützung von Objekthierarchien (so dass im eigentlichen Sinne gar nicht von „Szenengraphen“ gesprochen werden kann), unklare Weiterentwicklung und mangelnde Reliability. Die 20 Szenengraphen, die nach der Erstbegutachtung übrig blieben, waren im Einzelnen:

1. Blitz Engine
2. Coin
3. Crystal Space
4. Experience Engine
5. Gizmo3D

6. Java3D
7. Legus3D
8. Miracle3D
9. Nebula
10. NeMo
11. OpenGVS
12. OpenScenegraph
13. OpenSG
14. PanardVision
15. Performer
16. Pixelbase
17. PowerRenderer
18. RealiMation
19. TechNature Library
20. TitanEngine

Die Zielkriterien bei der zweiten Selektion waren hohe Performanz, Reliability (Eine Testversion des Szenengraphen musste innerhalb einer Woche problemlos installierbar sein), Verfügbarkeit von Manuals, Import-Export-Fähigkeit zu anderen Produkten und Formaten (Maya, Rhino, 3D Studio Max, VRML2 etc.) sowie eine umfassende mathematische Utility-Bibliothek im Software Development Kit, insbesondere für die Behandlung von Vektoren und Transformationsmatrizen. Nach dieser Zweitselektion kamen lediglich folgende 6 Szenengraphen in die engere Auswahl:

1. Java3D
2. NeMo (heutiges Virtools)
3. OpenGVS
4. OpenSG
5. Performer
6. RealiMation

Eine ungenügende Performanz führte bei der dritten Selektion zum Ausschluss von Java3D, NeMo und Performer. Bei Java3D fielen zusätzlich die offensichtlichen Bugs bei wichtigen Methoden (Pickingroutinen und Interpolatoren) negativ ins Gewicht. Bei

OpenGVS und OpenSG war das Angebot an Features und die Import-Export-Bandbreite sehr klein. Letzten Endes blieb lediglich der RealiMation Szenengraph übrig. Für eine detailliere Übersicht der fünf Szenengraphen siehe Kapitel 9.1.3 im Anhang auf Seite 375.

RealiMation ist mit Hilfe der Programmiersprache C programmiert, und bietet im SDK eine umfangreiche C-Funktionenbibliothek an. Infolge einer konsequenten Anwendung von Makros bei zeitkritischen Routinen ist er sehr performant (laut Angaben von Datapath Ltd. der performanteste Szenengraph). [Datapath 2001]

Er bietet u.a. die Möglichkeit des verteilten Renderings. Zusammen mit der SUN-Engine-Software von 3D-Generation ist die schnelle Adaption einer Visualisierung auf viele verschiedene Sichtsysteme möglich (monoskopisch oder stereoskopisch; wahlfreie Auflösung; Monitor, Projektion, Powerwall oder Cave) Dies ist vor allem für die Erstellung von Exponaten und anderen Präsentationen von Vorteil. RealiMation bietet die Auswahl zwischen drei Basis-Graphik-APIs:

- OpenGL Version 1.1 (mittlerweile 1.2)
- Direct3D Version 8.0 (mittlerweile 9.1)
- RenderWare Version 1.0

Zudem ist ein Nullrenderer für Debuggingzwecke vorhanden. Nachteilig ist die Tatsache, dass auf diese Graphik-APIs via DLL-Schnittstellen zugegriffen wird, so dass eine direkte Einfügung von OpenGL, Direkt3D oder RenderWare-Befehlen nicht möglich ist.

Im Nachhinein gesehen war die Entscheidung für RealiMation möglicherweise falsch, da nach dem Jahr 2001 sein SDK nicht mehr weiterentwickelt wurde und RealiMation in diesen Jahren wichtige Entwicklungen verpasst hat. Insbesondere fehlt die Unterstützung der Vertex- und Pixel-Shader-Languages, die insbesondere dem Quaoaring Design Patterns „Hair“ einen höheren Realismus verliehen hätten. Andere Szenengraphen wie z.B. OpenScenegraph haben dagegen in ihrer Entwicklung aufgeholt.

Der Code wurde konsequent objektorientiert in der Programmiersprache C++ implementiert. Als Compiler wurde MS Visual C++ in der Version 6.0 verwendet. Es wurde dabei darauf geachtet, dass der Code für eine spätere eventuelle Portierung auf andere Compilertypen weitgehend ANSI-kompatibel ist.

5.2 Auswahl der Hardware und der Betriebssysteme

In Zusammenhang mit der Auswahl der Hilfssoftware steht die Auswahl der Hardware. Dabei muss zwischen *Entwicklungsrechner*, *Messreferenzrechner* und *Zielrechner* unterschieden werden. Der Entwicklungsrechner, also derjenige Computer, auf dem das Quaoaring Software Framework programmiert wurde, ist beim Quaoaring Software Framework identisch mit dem Messreferenzrechner, also derjenige Computer, auf dem die standardisierten Geschwindigkeitsmessungen der Frame Rate vorgenommen wurden.

Aus Kostengründen wurden diese programmiertechnischen Arbeiten inklusive der Messungen auf einem schon bestehenden handelsüblichen PC mit folgenden Kenndaten vorgenommen:

- CPU: Intel Pentium 4-Prozessor mit MAG-Technologie, 3.0Ghz
- Grafikkarte: GeForce FX Go 5700, 64 MB DDR RAM
- Hauptspeicher: 512 MB DDR-RAM
- Betriebssystem: Windows 2000 bzw. (ab März 2004) Windows XP

Alle in der vorliegenden Arbeit veröffentlichten Daten bezüglich der Frame Rate beziehen sich auf Messungen, die auf diesem PC gewonnen wurden. Die Auflösung bei den Messungen beträgt dabei stets 1440 * 900 Pixel, die ausgewählte Graphik-Basis-API ist OpenGL und das Betriebssystem ist Windows XP.

Die Zielrechner, also diejenigen Computer, für die das Quaoaring Software Framework konzipiert wurde, sind beliebige PCs mit folgenden Kenndaten:

- CPU: mind. Intel Pentium 3-kompatibel, mind. 1.0Ghz
- Grafikkarte: 3D-Beschleunigung, mind. DirectX 9.0-fähig
- Hauptspeicher: mind. 256 MB DDR-RAM
- Betriebssystem: Windows 98, ML, 2000 oder XP.

Dabei wurde die Lauffähigkeit des Quaoaring Software Frameworks stichprobenartig auf 12 verschieden konfigurierten Zielrechnern mit unterschiedlicher Leistungsfähigkeit getestet, um mögliche Inkompatibilitäten im Vorfeld festzustellen.

Bei verteiltem Rendering muss ein Fast Ethernet LAN mit TCP/IP und UDP existieren. Der Zentralcomputer sollte dabei ähnliche Kenndaten wie der

Entwicklungsrechner haben, bei den reinen Rendercomputern sollte lediglich die Graphikkarte eine ähnliche Kapazität aufweisen wie die des Entwicklungscomputers, die Daten der CPU und des Hauptspeichers können dagegen niedriger gewählt werden (mind. Pentium 3 kompatibel, 64MB RAM, 1.0 GHz). Auf den Zielrechnern muss jeweils die MOON-Engine von 3D-Generation installiert sein.

5.3 Benennungskonventionen

Bei der Benennung der Variablen wurde streng auf die Einhaltung der selbst gestellten Benennungskonventionen geachtet, das Gleiche gilt für die Benennung von Methoden, Klassen und Konstanten.

Zusätzlich zu den üblichen Benennungskonventionen, die sich bei C-Programmieren allgemein eingebürgert haben (z.B. das Bekannte „m_“ Präfix für Membervariablen oder das Präfix „sz“ für Charakter Strings) wurde ein komplexes Präfixsystem für die Arbeit mit Szenegraphen eingeführt. So erhielt jeder Szenegraphobjekttyp ein eigenes Identifikationskürzel aus zwei Buchstaben. Auch andere in der Computergraphik oft verwendete Datentypen, wie beispielsweise Farbe, Farbe mit Alphakanal, Vektoren, Quaternionen oder homogene Matrizen, erhielten eindeutige Präfixe. [Breiner 2004] Diese Konsequenz und Strenge bei den Benennungen von Dateien sowie Konstanten, Variablen, Makros, selbst definierten Typen, Methoden und Klassen im Quellcode hat folgende Vorteile:

- Programmierer müssen sich nicht jedes Mal neu Gedanken über die Benennung ihrer Variablen, Methoden, Klassen etc. machen. Die Benennung ist schließlich eindeutig definiert. In der Folge wird die Programmierarbeit beschleunigt.
- Programmierneulinge haben es durch die Namenskonventionen wesentlich leichter, sich in der Software zurechtzufinden.
- Eventuelle Namenskonflikte im Team werden ausgeschlossen
- Die Verständlichkeit des Codes wird verbessert.

- Die Namen sind selbstkommentierend. Daher sind weniger Kommentare im Quellcode erforderlich.
- Das Kopieren von Teilen des Quellcodes ist unproblematisch. Anschließend mühsame Namensersetzungen werden obsolet.
- Das Suchen von Variablen, Methoden und Klassen im Quellcode wird vereinfacht und beschleunigt, da das Spektrum der Benennungsmöglichkeiten reduziert wird.
- C++-Programmierfehler, die durch fehlendes Casting auftreten, werden durch die Typenpräfixierung leichter entdeckt. Auch der Umgang mit Pointern wird erleichtert. Die Fehleranfälligkeit des Codes wird reduziert.

5.4 Softwareschichten und –module

Das Quaoaring-Software-Framework wurde in zwei streng hierarchisch aufgebauten Softwareschichten realisiert. Die untere Schicht heißt *TOP-Schicht*. In ihr ist u.a. die Funktionalität des Quaoarings mitsamt der Quaoaring Design Patterns implementiert. Sie bedient sich wiederum der *SUN-Engine* von 3D-Generation.

Die obere Schicht heißt *MOB-Schicht*. Sie ist projektspezifisch, d.h. ihr Aufbau und ihre konkrete Implementierung sind von Projekt zu Projekt unterschiedlich.

Diese Trennung des Quaoaring-Software-Frameworks in eine MOB- und eine TOP-Schicht ermöglicht, dass verschiedene Anwendungen auf dieselbe Quaoaring-Software zugreifen können. Die einzelnen Anwendungen können unabhängig voneinander an ihre jeweiligen Spezifika angepasst werden, ohne dass dabei die Quaoaring-Software modifiziert werden muss. Umgekehrt kann das Quaoaring anwendungsunabhängig weiterentwickelt werden.

Die Programmierer der MOB-Schicht, in der die anwendungsspezifischen Softwareteile stehen, können zukünftig von den Personen, die in der TOP-Schicht arbeiten verschieden sein. Sie müssen sich beide lediglich an die vereinbarten Schnittstellenfunktionen halten.

Die Einteilung des Quaoaring-Software-Frameworks in eine MOB- und eine TOP-Schicht ist auch in Hinblick auf zukünftige technische Weiterentwicklungen von Vorteil. So sind die Objektmodellierung sowie die damit zusammenhängende Mesh-Generierung in

der TOP-Schicht angesiedelt, während die Aktions- und Objektsteuerung in der MOB-Schicht liegen. So wird z.B. die (derzeit noch nicht implementierte, aber zukünftig angedachte) Verhaltenssteuerung der Organismen klar von der eigentlichen Modellierungs- und Animationstätigkeit getrennt.

In den folgenden Unterkapiteln werden die Funktionalitäten der grobe Aufbau der zwei Schichten inklusive der SUN-Engine beschrieben:

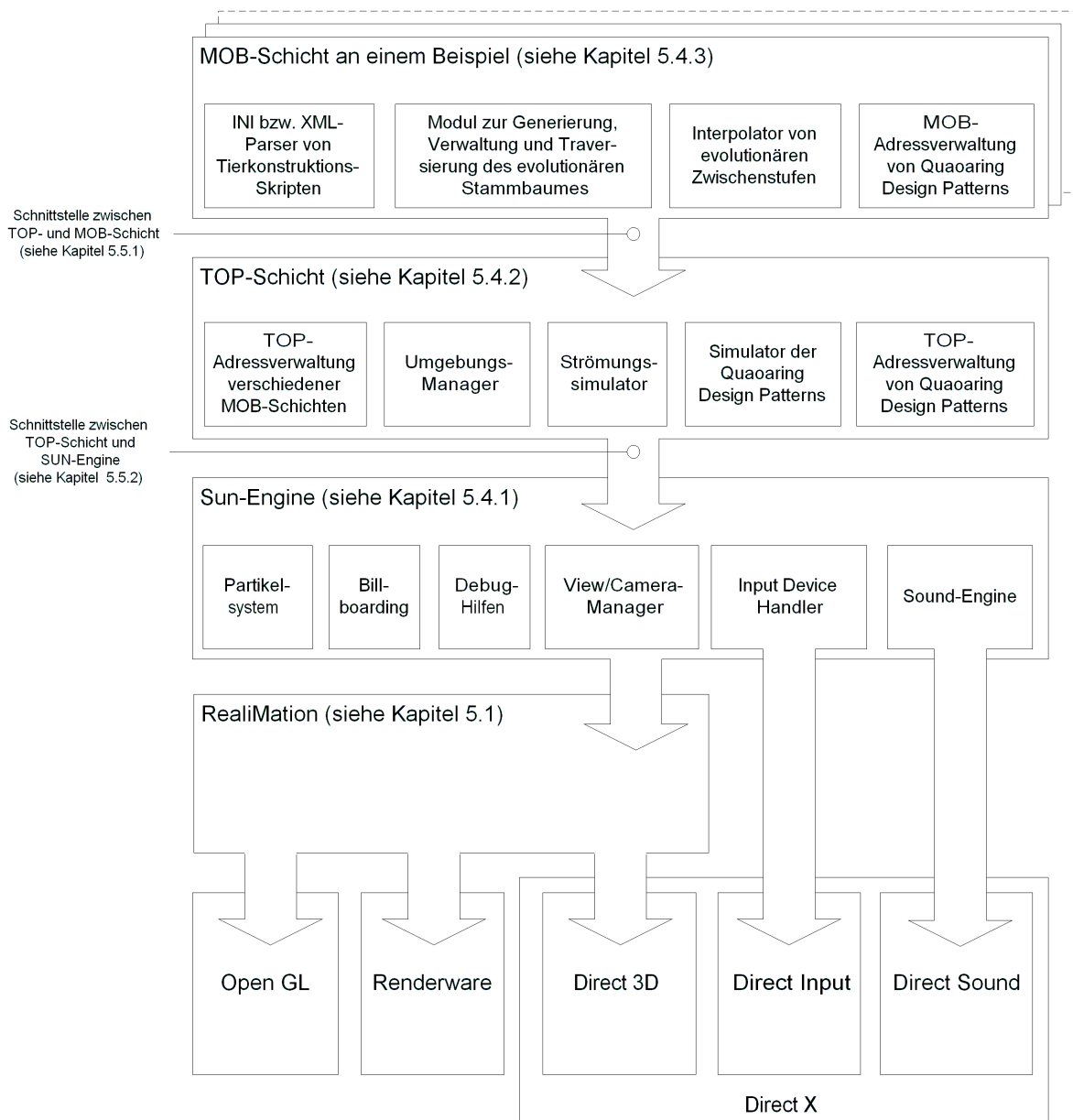


Abbildung 76: Aufbau des Quaoaring-Software-Frameworks mit Fremdbibliotheken

Miller-Diagramm des Softwareaufbaus des Quaoaring-Frameworks (die beiden obersten Schichten) inklusive der wichtigen Fremdbibliotheken und -engines (die drei unteren Schichten).

5.4.1 Aufbau und Funktionalitäten der Sun-Engine

Die SUN-Engine der Firma 3D-Generation bietet u.a. verteiltes Rendering im PC-LAN, Möglichkeit zur Stereoskopie, Billboarding, Partikelsysteme, Debugging-Hilfen, Sound und Input Device Handling an. Sie bedient sich der SDK-Befehle des Szenegraphen „RealiMation“. Somit können sich die mit der SUN-Engine realisierten Topapplikationen sowohl der Funktionalitäten bedienen, welche die SUN-Engine selbst bereitstellt, als auch auf die Features des RealiMation-Szenegraphen zugreifen.

Mittels des von der Sun-Engine angebotenen Zugangs zu DirectSound und DirectInput wird zusätzlich zur 3D-Ausgabe die Möglichkeit zur Soundausgabe und der Ansteuerung von externen Geräten wie beispielsweise Joystick, Tastenfeld, Maus und Lenkrad gegeben.

Die unterste Schicht bildet eine DLL-Schnittstelle zu wahlweise Direct 3D, OpenGL, Renderware oder den Nullrenderer. Darauf aufbauend ist der Szenegraph „RealiMation“ angesiedelt. Sie bietet eine genormte DLL-Schnittstelle an, und ruft damit eine vorher definierte Topapplikation auf, welche projektspezifisch ist, d.h. jedes auf Quaoaring basierende Konzept verwendet eine andere Topapplikation.

5.4.2 Aufbau und Funktionalitäten der TOP-Softwareschicht

In der TOP-Schicht befindet sich die eigentliche Funktionalität des Quaoarings respektive diejenigen Klassen und ihre Methoden, welche für die Quaoaring Design Patterns, ihre gegenseitigen Verknüpfungen und ihr interdependentes Zusammenspiel verantwortlich sind.

Jedes einzelne Quaoaring Design Pattern ist hier durch eine eigene Klasse repräsentiert. *CQdpMain*, *CQdpBulge*, *CQdpCanal* und *CQdpOrgan* sind dabei die Klassen der vier Pneuobjekte. Sie sind alle in der Vererbungshierarchie von der Klasse *CQdpPneu* abgeleitet, von der sie die Methoden zur Berechnung der Zentrallinie als auch die Routinen zur Erhaltung der Volumenkonstanz und für die Generierung des Skin-Meshs erben.

CQdpConstriction, *CQdpSqueeze*, *CQdpContractile* und *CQdpRelief* sind die Klassen der Deformationsobjekte. Sie sind von der Klasse *CQdpDeformation* abgeleitet und erben von ihr verschiedene Methoden zur Berechnung der morphischen Felder und der dazugehörigen Auslenkungen.

CQdpSkin, *CQdpHair* und *CQdpMaterial* sind die Klassen der Oberflächenobjekte. Sie sind von der Klasse *CQdpSurfaceObject* abgeleitet, von der sie die Parameter für ihre Oberflächenausbreitung in biologischen Koordinaten erhalten.

Als letzte Klasse für die Quaoaring Design Pattern existiert *CQdpEmission*. Die Klasse ist zusammen mit *CQdpPneu*, *CQdpDeformation* und *CQdpSurfaceObject* von der Basisklasse *CQdpObjekt* abgeleitet, welche Membervariablen zur Objektidentifikation als auch zur Hierarchiebildung beinhaltet, die für alle Quaoaring Design Pattern von Relevanz sind.

Neben den Klassen für die Quaoaring Design Patterns existieren noch zwei verschiedene Klassen für den Aufbau und die Verwaltung der Objekthierarchie, diese sind *CQdpAddress* und *CQdpAddresses*. Dabei instantiiert *CQdpAddresses* sowohl eine mehrfach verkettete Liste von *CQdpAdress* als auch ein zweidimensionales Array von Enumerationen, welches die erlaubten Eltern-Kind-Beziehungen der Quaoaring Design Patterns enthält.

Die TOP-Schicht vermag mehrere Instanzen von MOB-Schichten zu verwalten, dafür existiert ein *Softwaremodul zur Verwaltung verschiedener MOB-Schichten*. Das Modul enthält die Klassen *CMobAdresses*, *CMobAdress* und *CMob2QdpID*.

Zusätzlich existieren noch verschiedene Klassen für die Interpretation der Schnittstellenbefehle. Diese sind *CIfcTopMobInterpreter*, *CIfcTopMobTransductor*, *CIfcTopSunInterpreter* und *CIfcTopSunTransductor*.

In der TOP-Schicht existieren noch weitere Klassen, die jedoch für das eigentliche Quaoaring irrelevant sind, sie betreffen u.a. die Berechnung der Strömungssimulation für maritime Organismen, die Generierung von Soundeffekten, den Umgebungsaufbau und die Kamerasteuerung.

5.4.3 Aufbau und Funktionalitäten der MOB-Softwareschicht

In der Schichthierarchie des Quaoaring-Software-Frameworks steht die MOB-Schicht an oberster Stelle. Das Akronym MOB ist dabei von *motion object module* abgeleitet. Es können pro TOP-Schicht mehrere MOB-Schichten gleichzeitig instantiiert werden. Diese Aufgabe übernimmt dabei das in der TOP-Schicht liegende Softwaremodul zur Verwaltung verschiedener MOB-Schichten. (siehe Kapitel 5.4.2)

Eine MOB-Schichtinstanz steuert eine logisch zusammenhängende Objekthierarchie der Quaoaring Design Patterns, also in der Regel einen einzelnen

Organismus. Sollen mehrere Organismen gleichzeitig dargestellt werden, müssen dementsprechend mehrere MOB-Schichtinstanzen erzeugt werden. Es können dabei auch verschiedene Arten von MOB-Schichten gleichzeitig instantiiert werden.

Der konkrete Aufbau der MOB-Schicht kann von Projekt zu Projekt variieren. Daher bringt eine genaue Beschreibung der MOB-Schicht keinen Nutzen.

Lediglich die unidirektionale Schnittstelle zur TOP-Schicht ist naturgemäß bei allen Projektschichten identisch. Die entsprechenden Schnittstellenfunktionen stehen in der Klasse *CIfcMobTop* und sind in Kapitel 5.5.1 beschrieben. Damit verbunden ist ein Modul zur *Adressverwaltung der Quaoaring Design Patterns*. Diese Adressverwaltung vergibt jedem Quaoaring Design Pattern eine eindeutige Identifikationsnummer und verwaltet die interne Objekthierarchie der MOB-Schicht.

Darüber hinaus besteht bei den meisten Projekten die MOB-Schicht aus einem *INI- bzw. XML-Parser* zur Interpretation der Organismenkonstruktionsskripte. Dieser steuert ein *Modul zur Generierung, Verwaltung und Traversierung des evolutionären Stammbaumes*. Dieser ist wiederum mit einem *Interpolator von evolutionären Zwischenstufen* verbunden, der aus einem Objektakkumulator mit integriertem Interpolator für das parametrische Keyframing besteht.

5.5 Wichtige Softwareschnittstellen

Innerhalb des Quaoaring-Software-Frameworks existieren zwei wichtige genormte Software-Schnittstellen.

Die eine Schnittstelle verbindet die TOP- mit der MOB-Schicht und sollte damit MOB-seitig für die Projektentwickler und TOP-seitig für die Entwickler des allgemeinen Quaoaring-Kerns bekannt sein. Sie ist im Quaoaring-Schnittstellen-Handbuch beschrieben. [Breiner 2003a]

Die andere Schnittstelle verbindet die TOP-Schicht mit der darunter liegenden SUN-Engine. Sie ist damit nur für die Entwickler des allgemeinen Quaoaring-Software-Kerns von Relevanz. Sie ist ausführlich im 3D-Generation SUN-Engine-Manual beschrieben. [Breiner 2003b]

5.5.1 Die Schnittstelle zwischen MOB- und TOP-Schicht

Die Schnittstelle zwischen der MOB- und der TOP-Softwareschicht abstrahiert die für unterschiedliche Projekte oder Anwendungen relevanten Softwareteile von denjenigen Softwaremodulen, die den eigentlichen Quaoaring-Kern ausmachen.

Die Schnittstelle verläuft unidirektional in Richtung der MOB-Schicht, d.h. die Schnittstellenfunktionen werden ausschließlich von der MOB-Schicht bereitgestellt und die Aufrufe derselben erfolgen ausschließlich von der Seite der TOP-Schicht.

Damit sich die Anzahl der Schnittstellenfunktionen in einem erträglichen Rahmen hält, wurden Abfragefunktionen von verschiedenen Parametern von gleichem Typ miteinander fusioniert und mit einem zusätzlichen Enumeratorparameter *eProp* die verschiedenen Parameter identifizierbar gemacht. Somit existieren keine einzelnen Abfragefunktionen für die vielen unterschiedlichen Parameter der Quaoaring Design Patterns, sondern es gibt nur noch eine Abfragefunktion pro Typ und pro Interpolationsart. Auf diese Weise konnten die Schnittstellenfunktionen für Parameterabfragen von ca. 2300 Funktionen auf 13 Funktionen reduziert werden. Dies hat den positiven Nebeneffekt, dass die Schnittstelle in Hinblick auf zukünftige Weiterentwicklungen flexibel an mögliche neue Quaoaring Design Patterns oder zusätzliche neue Parameter angepasst werden kann. Es müssen dazu nur vor dem Compilierungsvorgang neue Werte für *eProp* in eine Define-Liste eingetragen werden, auf die beidseitig der Schnittstelle zugegriffen werden kann.

Instanzen der Quaoaring Design Patterns werden dabei über eine eindeutige Identifikationsnummer angesprochen, die sie jeweils von der Adressverwaltung der MOB-Schicht bei ihrer Erzeugung zugewiesen bekommen. Der ID-Wert wird dabei jeweils mit dem Parameter *idQdp* (im Falle von Call by Value) bzw. *pidQdp* (im Falle von Call by Reference) der Schnittstellenfunktionen übermittelt.

Bei manchen Quaoaring Design Patterns, bei denen ein Array ausgelesen werden muss (wie beispielsweise bei „Canal“), wird ein zusätzlicher Integer-Parameter *iNr* notwendig. Insgesamt existieren 33 Schnittstellenfunktionen. Im Folgenden werden diese Schnittstellenfunktionen kurz erklärt. Zuerst werden die Schnittstellenfunktionen zur Initialisierung bzw. Deinitialisierung von Modulen der MOB-Softwareschicht behandelt:

1. void Init(char * szIni, char * szdGrf, char * szdSnd, char * asdIni, CWRL_World * pworld);
2. void Fini();
3. void Reset();

Die Schnittstellenfunktionen *Init* und *Fini* sind dabei für die Initialisierung und die Deinitialisierung zuständig. Als Parameter werden Init dabei die Pfade zum ersten Organismus-Skriptfile (*szIni*), welches die Wurzel der Evolutionshierarchie darstellt, sowie zum Ordner der Texturbibliotheken (*szdGrf*), zum Ordner der Soundbibliothek (*szdGrf*) und zum Ordner der Parameterbibliothek (*szdIni*) übergeben. Weiterhin bekommt Init einen Pointer auf die Objektwurzel des Szenegraphen. Der Funktion Fini werden keine Parameter übergeben.

Die Funktion *Reset* reinitialisiert alle Systemparameter der MOB-Schicht mit Ausnahme der Pfadangaben und der Szenegraph-Pointer.

Es existieren weiterhin Funktionen, welche bei jedem Berechnungszyklus der TOP-Schicht einmal aufgerufen werden sollten:

4. void Tick(float fTime);
5. void Input(float fX, float fY, float fZ, int iDelta, bool b);
6. void GetPosHero(rtPosition * ppos);
7. void GetPosTarget(rtPosition * ppos);

Tick ist eine Laufzeitfunktion, die bei jedem Frame von der TOP-Schicht aufgerufen wird. Die Funktion wird jeweils mit dem Parameter *fTime* aufgerufen, welcher die seit dem Programmstart vergangene Zeit in Sekunden angibt.

Mit *Input* können Systembefehle an die MOB-Schicht heraufgereicht werden. Dabei wird in der aktuellen Version von e-Voluzzer lediglich der vierte Parameter *iDelta* verwendet, um die Traversierungsrichtung innerhalb des Evolutionsbaumes zu steuern. Der Wert -1

steht dabei für das Zurückverfolgen der Evolution in Richtung der Wurzel, Null steht für das Fehlen einer Aktion und positive Werte für das Verzweigen zu einen der evolutionären Nachfolger in Richtung der Blätter des Evolutionsbaumes.

Mit Hilfe von *GetPosTarget* kann die Position und Orientierung desjenigen Zielpunktes abgefragt werden, auf welchen die Kamera ausgerichtet ist. Der Kamera-Zielpunkt muss nicht unbedingt identisch mit dem Zielorganismus sein, um den sich die evolutionäre Simulation gerade dreht. (Es kann demnach pro MOB-Schicht nur einen einzigen Zielorganismus geben, sollen mehrere Organismen simuliert und visualisiert werden, müssen von der TOP-Schicht mehrere Instanzen der MOB-Schicht erstellt werden.) Daher existiert die Funktion *GetPosHero*. Mit ihr lässt sich die Position und Orientierung dieses „Helden“ abfragen. Die Positionsangaben werden dabei in absoluten Koordinaten (linkshändige Notation) über den Parameter *ppos* zurückgegeben.

Zu *GetPosHero* und *GetPosTarget* existieren auch inverse Funktionen sowie Initialisierungsfunktionen:

8. void SetPosHero(rtPosition pos);
9. void InitPosHero();
10. void SetPosTarget(rtPosition pos);
11. void InitPosTarget();

Mit Hilfe der Funktion *SetPosHero* kann der Zielorganismus platziert und mit Hilfe von *InitPosHero* initialisiert werden, als Ausgangskordinaten dient dabei der Ursprung des absoluten Koordinatensystems. Analog wird mit Hilfe von *SetPosTarget* der Kamera-Zielpunkt positioniert und mit *InitPosTarget* initialisiert.

Es existieren zwei Funktionen, welche dezidierte Stadien der gesamten MOB-Schicht betreffen:

```
12. int GetMobIProp(int eProp);
```

```
13. void SetMobIProp(int eProp, int eState);
```

GetMobIProp ist eine Schnittstellenfunktion zur Statusabfrage der gesamten MOB-Schicht. Welcher Status dabei abgefragt werden soll, wird mit Hilfe des Enumerationsparameters *eProp* bestimmt. Sowohl die Statusliste als auch die Liste für die Bedeutung der Resultate befinden sich als Auflistung von Define-Präprozessor-Makros in der Datei „DEF_TopMobConstants.h“, auf welche von beiden Seiten der Schnittstelle zugegriffen wird. Zu *GetMobIProp* existiert auch die inverse Funktion *SetMobIProp*, mit welcher der Status der MOB-Schicht verändert werden kann.

Es existieren weiterhin vier Funktionen, die dem Aufbau der Objekthierarchie von Quaoaring Design Patterns dienen:

```
14. bool MakeQdp(unsigned int * pidQdp, int eKind);
```

```
15. bool KillQdp(unsigned int * pidQdp);
```

```
16. bool AddQdp(unsigned int * pidQdpParent, unsigned int *  
pidQdpChild);
```

```
17. bool SubQdp(unsigned int * pidQdpParent, unsigned int *  
pidQdpChild);
```

Mit Hilfe der Funktion *MakeQdp* werden neue Instanzen von Quaoaring Design Patterns in der TOP-Schicht kreiert. Dabei bekommt das neue Quaoaring Design Pattern von der Adressverwaltung der MOB-Schicht mit Hilfe von *pidQdp* eine eindeutige Identifikationsnummer zugewiesen. Der Enumerationsparameter *eKind* gibt an, welche Art Quaoaring Design Pattern erzeugt werden soll. Die Liste für die Codierung der Quaoaring Design Patterns befindet sich dabei in der Datei „DEF_TopMobConstants.h“, auf welche auf beiden Seiten der Schnittstelle zugegriffen wird. Soll kein Quaoaring Design Pattern erzeugt werden, gibt die Funktion den Wert „false“ zurück.

Mit Hilfe der Funktion *KillQdp* können Instanzen der Quaoaring Design Patterns wieder gelöscht werden, der Parameter *pidQdp* verweist dabei auf die

Identifikationsnummer der Instanz des Quaoaring Design Patterns. Sollen keine Instanzen von Quaoaring Design Patterns mehr gelöscht werden, gibt die Funktion den Wert „false“ zurück.

Mit Hilfe von *AddQdp* kann die Objekthierarchie der Quaoaring Design Patterns aufgebaut werden. Die Funktion fügt ein Quaoaring Design Pattern (identifiziert durch den Parameter *pidQdpChild*) als Subobjekt an ein anderes Quaoaring Design Pattern (identifiziert durch *pidQdpParent*) an.

Mittels *SubQdp* können zwei Quaoaring Design Patterns wieder voneinander abgekoppelt werden. Die beiden Koppelfunktionen geben den Wert „false“ zurück, wenn keine weitere Operation gleichen Typs mehr ausgeführt werden soll.

Es existieren weiterhin zwei Funktionen zur Abfrage von kartesischen Werten (Position, Orientierung und Skalierung) im absoluten Koordinatensystem:

```
18. rtPosition GetQdpPos(unsigned int idQdp, int iNr = -1);
```

```
19. rtScale GetQdpScale(unsigned int idQdp);
```

Mit *GetQdpPos* kann über die ID die Position und Orientierung eines Quaoaring Design Patterns abgefragt werden. Die Angaben erfolgen im globalen linkshändigen Koordinatensystem. Mit *GetQdpScale* ist es möglich, die Skalierung eines Quaoaring Design Patterns zu ermitteln. Die beiden Funktionen können logischerweise alleine für das Quaoaring Design Pattern des Typs „Main“ angewandt werden, da es als Wurzel der Objekthierarchie das einzige Quaoaring Design Pattern ist, dessen Position, Orientierung und Skalierung nicht im biologischen Koordinatensystem angegeben wird. Die Datentypen der Ergebnisparameter *rtPosition* und *rtScale* werden vom RealiMation-Szenegraphen SDK bereitgestellt. [Datapath 2000]

Zusätzlich zur Skalierungs- und Positionsabfrage gesellen sich noch eine Vielzahl weiterer Schnittstellenfunktionen für allgemeine Parameterabfragen von Quaoaring Design Patterns:

20. unsigned int GetNrOfSubQdps(unsigned int idQdp);
21. rtID GetQdpRProp(unsigned int idQdp, int eProp, int iNr = -1);
22. int GetQdpIProp(unsigned int idQdp, int eProp, int iNr = -1);
23. char * GetQdpSZProp(unsigned int idQdp, int eProp,
int iNr = -1);
24. float GetQdpFProp(unsigned int idQdp, int eProp, int iNr = -1);
25. rtColour GetQdpColourProp(unsigned int idQdp, int eProp,
int iNr = -1);
26. rtVector GetQdpVProp(unsigned int idQdp, int eProp,
int iNr = -1);
27. float GetQdpFInterpolLinear(unsigned int idQdp, int eProp,
int iNr = -1);
28. float GetQdpColourInterpolLinear(unsigned int idQdp, int eProp,
int iNr = -1);
29. float GetQdpVInterpolLinear(unsigned int idQdp, int eProp,
int iNr = -1);
30. float GetQdpFInterpolCubic(unsigned int idQdp, int eProp,
int iNr = -1);
31. float GetQdpColourInterpolCubic(unsigned int idQdp, int eProp,
int iNr = -1);
32. float GetQdpVInterpolCubic(unsigned int idQdp, int eProp,
int iNr = -1);

Mit Hilfe von *GetNrOfSubQdps* kann die Anzahl der Subobjekte eines Quaoaring Design Patterns ermittelt werden, wie beispielsweise die Anzahl der Stützstellen beim Quaoaring Design Pattern „Contractile“.

Die restlichen Parameterabfrage-Funktionen unterscheiden sich allein durch den Typ ihres Ausgabeparameters. So werden dreidimensionale homogene Vektoren mittels *GetQdpVProp*, Farbwerte (in RGB codiert) mittels *GetQdpColourProp*, Winkel (im Bogenmaß) mittels *GetQdpSegRadius*, Szenegraphen-Ids (RealiMation Ids) mittels *GetQdpRProp*, Integer-Werte und Enumerationen mittels *GetQdpIProp*, Zeichenketten mittels *GetQdpSZProp* und Gleitkommazahlen mittels *GetQdpFProp* abgefragt.

Die Quaoaring Design Patterns werden dabei jeweils mit Hilfe des Parameters *idQdp* über ihre ID angesprochen. Durch einen weiteren optionalen Parameter *iNr* können die Unterobjekte des Quaoaring Design Patterns angesprochen werden. Wird *iNr* gleich -1 gesetzt, so werden Parameter angesprochen, die das gesamte Quaoaring Design Pattern betreffen. Die Art und die Funktion des zu übertragenden Parameters werden mit dem Enumerationsparameter *eProp* übertragen. Die Konstantenliste befindet sich in der Datei „DEF_TopMobConstants.h“, auf welche auf beiden Seiten der Schnittstelle zugegriffen wird.

Um lineare Interpolationen zwischen zwei parametrischen Keyframes zu erhalten, können die Funktionen *GetQdpFInterpolLinear* (für Gleitkommazahlen), *GetQdpVInterpolLinear* (für Vektoren) und *GetQdpColourInterpolLinear* (für Farbinterpolationen) verwendet werden. Für stückweise kubische Interpolationen stehen analog die Funktionen *GetQdpFInterpolCubic*, *GetQdpVInterpolCubic* und *GetQdpColourInterpolLinear* zur Verfügung.

Die Ergebnissparameter-Datentypen *rtID*, *rtColour* und *rtVector* werden vom RealiMation-Szenegraphen SDK bereitgestellt. [Datapath 2000]

Zuletzt existiert noch eine Funktion zur Abfrage von Fehlern, welche bei der Anwendung der Schnittstellenfunktionen aufgetreten sind:

```
33. unsigned int CheckTopMobError();
```

Die Funktion *CheckTopMobError* gibt den Wert „0“ aus, falls die letzte angewandte Schnittstellenfunktion fehlerfrei ausgeführt wurde, ansonsten wird eine Fehlernummer größer 0 ausgegeben. Die dazugehörige Liste der Fehlernummern steht als Auflistung von

Präprozessordefinitionen in der Datei „DEF_TopMobConstants.h“. Zu den möglichen Fehlern zählen u.a. Allokationsfehler durch mangelnden Speicherplatz, Verbindungsversuche miteinander inkompatibler Quaoaring Design Patterns und falsche Eingabeparameterwerte.

5.5.2 Die Schnittstelle zwischen TOP-Schicht und SUN-Engine

Die Schnittstelle zwischen TOP-Schicht und SUN-Engine trennt diejenigen Softwareteile, die für das Quaoaring relevant sind, von der zugrunde liegenden 3D-Engine, welche u.a. verteiltes Rendering im PC-LAN, Möglichkeit zur Stereoskopie, Billboarding, Partikelsysteme, Debugging-Hilfen, Sound und Input Device Handling anbietet. Die DLL-Schnittstelle beinhaltet über 212 Funktionen, dazu zählen u.a.:

- Schaltfunktionen zum Generieren, Löschen, Konfigurieren, Modifizieren, An- und Ausschalten von Viewports, (im Local Area Network auch über PC-Grenzen hinweg)
- Schaltfunktionen zum Remote-Booting, Shutdown und Restarting von PCs im LAN
- Funktionen zum Erzeugen, Löschen, Konfigurieren und Modifizieren von Single- und Multibillboards, dabei sind auch komplexe mehrschichtige Billboards sowie zueinander sternförmig angeordnete Billboards erzeugbar
- Funktionen zum Erzeugen, Löschen, Konfigurieren und Modifizieren von Partikelemittern
- Funktionen zum Erzeugen, Löschen, Konfigurieren und Steuern eines einfachen 3D-GUIs

- Funktionen zum Erzeugen, Löschen, Konfigurieren und Steuern von intelligenten Kamerasystemen, wie z.B. autonome Verfolgungskameras; hierbei sind sowohl stereoskopisch miteinander gekoppelte Kamerasysteme als auch gekoppelte Kamerasysteme zum Handling von verteilten Sichtsystemen möglich
- Funktionen zur Unterstützung des Debuggings

Die entsprechenden Schnittstellenfunktionen sind ausführlich im SUN-Engine-Manual beschrieben. [Breiner 2003b]

5.6 Ordnerstruktur

Die Struktur der Ordner des Quaoaring-Software-Frameworks wurde in einer zweischichtigen Hierarchie angelegt. Die erste Ordnerhierarchieschicht dient zur einfachen Trennung der verschiedenen Softwaremodule, die zweite Schicht differenziert zwischen verschiedenen Dateitypen. Die Dateien stehen alle in den Ordnern der zweiten Schicht. In der ersten Schicht sind dabei folgende Ordner relevant:

- GEN
- DLL
- MOB_Organism
- TOP_Xxx (Xxx = Namenskürzel des Projektes)

GEN ist dabei der generische Hauptordner. In ihm steht das Executable von 3D-Generation (SUN.exe), welches die Topapplikation aufruft. Bei jedem Aufruf wird ein ausführliches Log-File (SUN_Log.txt) zur Fehlersuche generiert. Dazu stehen alle benötigten MFC-Ressourcen sowie alle Icons und Bilder für das Splash-Window.

DLL ist der Ordner für alle Untereinheiten, die sowohl von der SUN-Engine als auch von der Topapplikation gemeinsam verwendet werden. Hier stehen unter anderem alle dafür verwendeten DLLs mitsamt ihren zugehörigen INI-Files, Bilder und Sounds.

MOB_Organism: Hier stehen alle Files, die in Zusammenhang mit der direkten Organismengenerierung stehen, also die Organismenkonstruktionsskriptdateien sowie die Textur- und Sound-Datenbank.

TOP_Xxx: Hier stehen alle Ordner, die im direkten Zusammenhang mit der Topapplikation stehen. Wichtig zu erwähnen sind hier von allem die *App.ini*, in der u.a. wichtige Grundkonfigurationen der TOP-Schicht vorgenommen werden können.

Die zweite Ordnerschicht ist für alle Ordner der ersten Schicht einheitlich und differenziert zwischen verschiedenen Dateitypen:

- *BIN*: Hier stehen jeweils sowohl die Debug- und Release-Executable (als EXE-Files) als auch die dazugehörigen Programmbibliotheken (DLL-Files) sowie das Logfile (als TXT-File)
- *INI*: Hier stehen alle Konfigurationsdateien, seien es XML- oder INI-Files
- *ICO*: Hier stehen alle Icons (ICO-Files), Splash-Window-Graphiken als Bitmaps (BMP-Files) und Microsoft Foundation Class Resources (RC2-Files)
- *GRF*: Hier stehen alle Grafikdateien (u.a. TGA-, GIF- und JPG-Files) sowie Dateien mit 3D-Szenengraphen-Hierarchien (RBS-Files)
- *SND*: Hier stehen alle Sound-Dateien (WAV-Files)
- *DOC*: Hier stehen alle Dokumente wie Bedienungsanleitungen, Kommentardateien für die Software und Ideenskripte (DOC-, TXT- und HTM-Files)
- *SRC*: Hier stehen die C++ Source- und Header-Files inklusive Projekt- und Arbeitsbereichsdateien (CPP-, H-, DSP- und DSW-Files). Auch alle während des Kompilervorgangs erzeugten Zwischendateien (wie z.B. OPT-, PLG-, OBJ- und SRB-Files) werden in diesem Ordner abgelegt.

Diese Dateistruktur hat ihren Sinn: So stehen voneinander verschiedene Softwareschichten auch in unterschiedlichen Ordnern der ersten Hierarchiestufe. Auch verschiedene Projekte sind in dieser Schicht voneinander durch verschieden benannte TOP-Xxx Ordner sauber getrennt.

Die unterschiedlichen Funktionalitäten werden dagegen in der zweiten Hierarchiestufe abgebildet. So sind z.B. für eine Release-Version ohne Sound die Ordner SRC und SND nicht von Belang.

Auf diese Weise wird die Erzeugung verschiedener Projekt- und/oder Auslieferungsversionen durch das einfache Kopieren bzw. Nichtkopieren ganzer Ordnerstrukturen beschleunigt.

6 Anwendungsbeispiele und Ergebnisse

*Die Wissenschaft fängt eigentlich erst da an
interessant zu werden, wo sie aufhört.*

Justus Freiherr von Liebig 1844

In diesem Kapitel wird beschrieben, wie das Quaoaringkonzept in der Praxis umgesetzt und verwendet wird, indem einige konkrete Anwendungen exemplarisch vorgestellt werden.

Als Anwendungsbeispiele dienen die Projekte *e-Go*, *e-Voluzzer*, *e-VoLuzie*, *e-VoLutz*, *e-Rektus*, *e-Munkulus* und *e-Rythrozyt*. Die Begründung dieser exemplarischen Auswahl wird dabei im Unterkapitel 6.1 gegeben. Die einzelnen Projekte werden dann in den Unterkapiteln 6.2 bis 6.8 mit ihren Hintergründen beschrieben.

Das gleichförmige Präfix „e-“ in den Projektnamen soll dabei verdeutlichen, dass die Projekte nicht isoliert voneinander betrachtet werden können, sondern miteinander zusammenhängen. Das Unterkapitel 6.2 dient nicht nur zur Beschreibung des Projektes *e-Go*, sondern gleichzeitig zum Aufzeigen der Art dieser Zusammenhänge.

Eine Bewertung der Projekte findet am Schluss dieses Kapitels im Unterkapitel 6.9 statt.

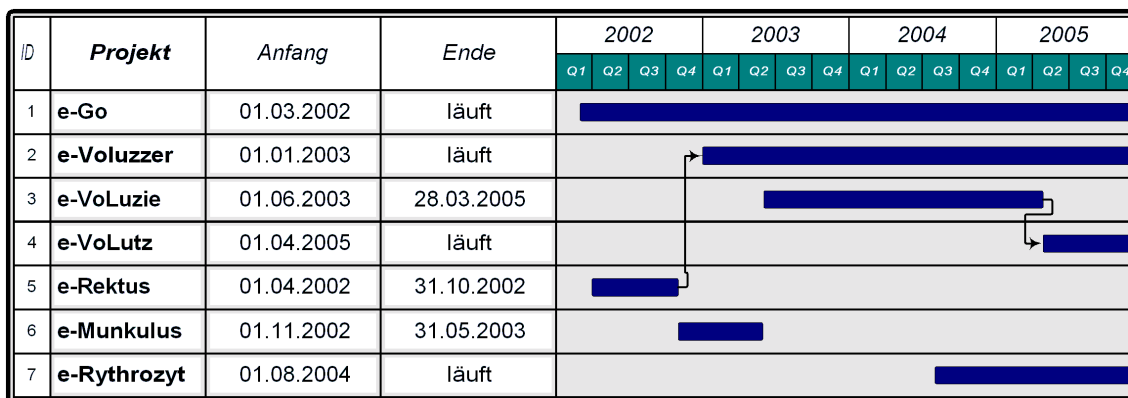


Abbildung 77: Gantt-Diagramm der Zeiten der auf Quaoaring basierenden Projekte

In dieser Abbildung sind die sieben auf Quaoaring basierenden Projekte als Gantt-Diagramm eingetragen. *e-VoLutz* ist ein Folgeprojekt von *e-VoLuzie* und *e-Voluzzer* ein Folgeprojekt von *e-Rektus*. *e-Go* hat als administratives Projekt zwangsläufig die längste Projektdauer.

6.1 Begründung der Auswahl von Anwendungsbeispielen

6.1.1 Anforderungskriterien für die Selektion

Für eine Bewertung der allgemeinen Praxistauglichkeit des Quaoarings anhand der Anwendungsbeispiele müssen Projekte herangezogen werden, die von ihrer Zielsetzung her sehr unterschiedlich sind und deren Anforderungen ein breites Spektrum abdecken. Als Kriterien für die Unterschiedlichkeit der Projekte wurden folgende Parameter bzw. Attribute herangezogen:

1. die Holarchiestufe (siehe Kapitel 3.10) beim zu visualisierenden Zielorganismus
2. der Komplexitätsgrad der zu visualisierenden Strukturen
3. die analytischen Fähigkeiten
4. der Interaktivitätsgrad und die damit verbundene Echtzeitfähigkeit
5. die Anforderungen an die Sicherheit, Stabilität und Robustheit der Software
6. der Grad des erforderlichen Realismus der Visualisierung

Wie in den folgenden Unterkapiteln aufgezeigt wird, erfüllen die sieben exemplarischen Projekte diese Anforderungskriterien bei allen sechs Parametern.

6.1.2 Unterschiede der visualisierten Holarchiestufe

Die exemplarischen Projekte haben vom Aspekt der Abstraktionsebene in der Holarchie (siehe Kapitel 3.10) her sehr unterschiedliche Zielobjekte zur Visualisierung:

- Ein Projekt visualisiert Zellen, inklusive ihrer Substrukturen. (e-Rythrozyt)
- Ein Projekt simuliert und visualisiert zelluläre Netzwerke. (e-Munkulus)
- Ein Projekt visualisiert und simuliert die Wirkung metabolischer Vorgänge auf anatomische Strukturen insbesondere toxikologisch relevante Organe wie Herz, Leber und Niere. (e-Rektus)
- Drei Projekte visualisieren Gesamtorganismen und Schwärme derselben (e-Voluzzer, e-VoLuzie, e-VoLutz)

Somit sind von den zehn biologisch relevanten Holarchiestufen (1. biochemische Molekülstrukturen, 2. Organellen, 3. Zellen, 4. zelluläre Netzwerke und Gewebestrukturen, 5. Suborgane, 6. Organe, 7. anatomische Strukturen, 8. Gesamtorganismen, 9. Schwärme, Herden und andere Organismengemeinschaften sowie 10. Populationen und Ökosysteme) zumindest acht Ebenen erfasst (Holarchiestufen 2 bis 9). Da sowohl eine optimale Visualisierung biochemischer Molekülstrukturen als auch von Populationen und Ökosystemen per Quaoaring sowieso unmöglich ist, umfassen die sieben exemplarischen Projekte alle möglichen Holarchiestufen und bieten somit in dieser Hinsicht das maximal mögliche Projektspektrum.

6.1.3 Unterschiede bei der visualisierten Komplexität

Der Grad der Komplexität hängt eng mit der Art der Holarchiestufe der zu visualisierenden Strukturen zusammen, ist jedoch nicht mit ihr identisch, da auch komplexe Strukturen schematisiert oder unvollständig visualisiert werden können. Er muss somit gesondert behandelt werden.

Die Strukturen der Zielobjekte, welche visualisiert werden, sind in den verschiedenen Projekten von stark unterschiedlicher Komplexität:

- Bei einem Projekt werden Einzelobjekte visualisiert, die maximal 8 Quaoaring Design Patterns mit insgesamt lediglich ~5.000 Polygonen beinhalten (e-Rektus).
- Ein Projekt versucht neuronale Netzwerke inklusive ihrer filigranen Substrukturen (Dendriten, Axone, Synapsen etc.) zu visualisieren und ist damit verständlicherweise besonders anspruchsvoll hinsichtlich der Komplexität. Pro Neuron kann von einer durchschnittlichen Anzahl von ~10.000 Quaoaring Design Patterns und ~120.000 Polygonen ausgegangen werden. Schon bei der Visualisierung eines kleineren Netzwerkes ($N \approx 1000$) kann somit von ~10 Millionen zu berechnenden Quaoaring Design Patterns mit ~120 Millionen Polygonen ausgegangen werden. (e-Munkulus)

- Bei den meisten Projekten werden Gesamtobjekte visualisiert, die aus bis zu 250 Quaoaring Design Patterns bestehen können. Die Gesamtanzahl der Polygone kann bis zu 30.000 betragen (e-Rythrozyt, e-Voluzzer, e-VoLuzie und e-VoLutz).

Somit umspannen die Anforderungen bezüglich der Komplexität beim anspruchsvollsten Projekt das 1.250.000-fache (Anzahl der Quaoaring Design Patterns) bzw. 24.000-fache (Polygonanzahl) des einfachsten Projektes. Eine große Bandbreite bezüglich der Komplexitätsanforderungen ist somit gewährleistet.

6.1.4 Unterschiede der analytischen Fähigkeiten

Die exemplarischen Projekte unterscheiden sich hinsichtlich ihrer analytischen Fähigkeiten zur Überprüfung wissenschaftlicher Theorien:

- Es existieren drei Projekte, bei denen es lediglich auf eine korrekte Visualisierung ankommt. Analytische Fähigkeiten sind hierbei nicht erforderlich (e-Munkulus, e-VoLuzie und e-Rythrozyt).
- Bei zwei Projekten werden lediglich die dem Quaoaring inhärenten Fähigkeiten der Formanalyse zur Verifizierung oder Falsifizierung evolutionswissenschaftlicher bzw. paläontologischer Hypothesen verwendet (e-Voluzzer und e-Go).
- Bei einem Projekt ist eine einfache Strömungssimulation integriert, die u.a. für die Bewegungsanalyse von Flossenschlägen verwendet wird (e-VoLutz).
- Bei einem Projekt ist eine analytisch verwertbare pharmakologisch-toxikologische Simulation integriert (e-Rektus).

Somit umfassen die sieben exemplarischen Projekte ein weites Spektrum hinsichtlich ihrer Analysefähigkeiten, welches alle Nuancen zwischen nicht vorhandener, über qualitativ verwertbare bis hin zu einer geeichten und somit quantitativ verwertbaren Analytik umspannt.

6.1.5 Unterschiede beim Interaktivitätsgrad

Die Anforderungen an die Projekte unterscheiden sich hinsichtlich ihrer Interaktivität und der damit eng verbundene Echtzeitfähigkeit:

- Es existiert ein Projekt, bei dem es lediglich auf eine Visualisierung von Einzelbildern ankommt. Die Aspekte der Interaktivität und der Echtzeitfähigkeit haben hierbei keine Relevanz (e-Rythrozyt).
- Bei drei Projekten sind Interaktivität und Echtzeitfähigkeit zwar wünschenswert, jedoch nicht unbedingt erforderlich, auch wenn diese Gesichtspunkte eine mehr oder weniger hohe Priorität in ihren Projektbeschreibungen haben (e-Go, e-Voluzzer, e-Munkulus).
- Bei weiteren drei Projekten ist der Aspekt der Interaktivität obligatorisch, sie würden somit ohne Echtzeitfähigkeit keinen Sinn ergeben (e-Munkulus, e-VoLuzie und e-VoLutz).

Somit umspannt die Auswahl der Anwendungsbeispiele bezüglich der Erfordernisse der Interaktivität und Echtzeitfähigkeit eine maximale Bandbreite zwischen „nicht vorhanden“ und „unentbehrlich“.

6.1.6 Unterschiede der Stabilitätsanforderungen

Die Anforderungen der sieben exemplarischen Projekte bezüglich ihrer Softwarestabilität und Robustheit unterscheiden sich in folgender Hinsicht:

- Es existieren Projekte, wo der Aspekt der Softwarestabilität von geringer Relevanz ist, da einerseits die Software nur von einzelnen Experten bedient wird und somit die Wahrscheinlichkeit von Komplikationen niedrig ist und andererseits Software-Issues keine gravierenden Implikationen zeitigen würden. (e-Rythrozyt)

- Es existieren Projekte, bei denen der Aspekt der Softwarestabilität zwar eine hohe Priorität eingeräumt wird, an die jedoch in dieser Hinsicht keine besonderen Anforderungen gestellt werden. (e-Go, e-Voluzzer, e-Munkulus und e-Rektus)
- Es gibt Projekte, welche als interaktive Museumsexponate extrem robust gegenüber Falscheingaben und Sabotageversuchen sein müssen, insbesondere da sie unter anderem auch von Kindern und Jugendlichen bedient werden. Zudem könnten Softwareabstürze zu Verdienstausschlägen führen und die Reputation des Museums beschädigen. (e-VoLuzie und e-VoLutz)

Somit variiert der Anforderungsgrad an die Softwarestabilität einen Bereich zwischen „gering“ und „sehr hoch“. Eine hohe Variabilität beim Aspekt der Softwarestabilitätsanforderungen ist somit gewährleistet.

6.1.7 Unterschiede beim erforderlichen Realismus

Die Anforderungen in Bezug auf den optischen Realismus der zu visualisierenden Strukturen unterscheiden sich bei den sieben exemplarischen Projekten in folgender Hinsicht:

- Es existiert ein Projekt, bei welchem abstrahierende Visualisierungen verwendet werden können, da es hier eher um das Verdeutlichen dreidimensionaler Topologien geht. (e-Munkulus)
- Es existieren Projekte, bei denen eine realistische Visualisierung zwar wünschenswert ist, ihr jedoch keine große Priorität eingeräumt wird. (e-Voluzzer und e-Rektus)
- Bei drei Projekten hat der Realismus der Visualisierung eine sehr hohe Priorität (e-Rythrozyt, e-VoLuzie und e-VoLutz)

Somit reicht die Auswahl der Anwendungsbeispiele von abstrahierenden bis hin zu realistischen Visualisierungserfordernissen. Eine große Bandbreite bezüglich der Anforderungen an den Realismus der Visualisierung ist somit sichergestellt.

6.1.8 Übersicht der Anforderungsspektren

In den vorherigen Kapiteln 6.1.3 bis 6.1.7 wurde aufgezeigt, dass die sieben exemplarischen auf Quaoaring basierenden Projekte bezüglich der in Kapitel 6.1.1 gegebenen Anforderungskriterien eine maximale Bandbreite aufweisen. (Eine Übersicht über die Erfüllung der Anforderungskriterien verschafft Tabelle 18.)

Die Projektauswahl ist somit geeignet, möglichst viele Aspekte des Quaoarings zu beleuchten, mögliche unentdeckte Schwachpunkte aufzufinden und die Praxistauglichkeit dieser Technologie aufzuzeigen.

Projektname	Visualisierte Holon-hierarchiestufe	Komplexität der visualisierten Strukturen	Analytische Fähigkeiten	Interaktivitäts-grad und Echtzeit-fähigkeit	Stabilitäts-anforderungen	Anforderungen an den Realismus der Visualisierung
<i>e-Go</i>	5,6,7,8,9	+	+	+	+	+
<i>e-Voluzzer</i>	7,8,9	+	+	+	+	+
<i>e-VoLuzie</i>	8,9	+	-	++	++	++
<i>e-VoLutz</i>	8	+	++	++	++	++
<i>e-Rektus</i>	5,6	-	++	+	+	+
<i>e-Munkulus</i>	3,4	++	-	+	+	-
<i>e-Rythrozyt</i>	2,3	+	-	-	-	++

Tabelle 18: Anforderungen an die auf Quaoaring basierenden Projekte

In dieser Tabelle sind zeilenweise die sieben exemplarischen auf Quaoaring basierenden Projekte und reihenweise ihre spezifischen Anforderungskriterien eingetragen. Ein Minuszeichen bedeutet, dass im Kontext des jeweiligen Anforderungskriteriums keine besonderen Anforderungen an das Projekt gestellt werden, ein Pluszeichen bedeutet leichte bis hohe Anforderungen, zwei Pluszeichen bedeuten sehr hohe Anforderungen. Die Zahlencodierung in der 2. Spalte (Visualisierte Holarchiestufe) wird in Kapitel 6.1.2 erläutert.

6.2 e-Go

Die sieben exemplarischen Projekte, welche auf Quaoaring basieren, sind eng miteinander verwoben, da sie – zumindest hinsichtlich Quaoaring – denselben Software-Code verwenden. Softwaremodifikationen im Quaoaring-relevanten Code, die im Rahmen eines Projektes durchgeführt werden, wirken sich zwangsläufig auf die anderen sechs Projekte aus. Die zentrale koordinierende Rolle in diesen Projekten nimmt dabei stets das Institut für Graphische Datenverarbeitung an der Johann Wolfgang Goethe-Universität in Frankfurt am Main ein.

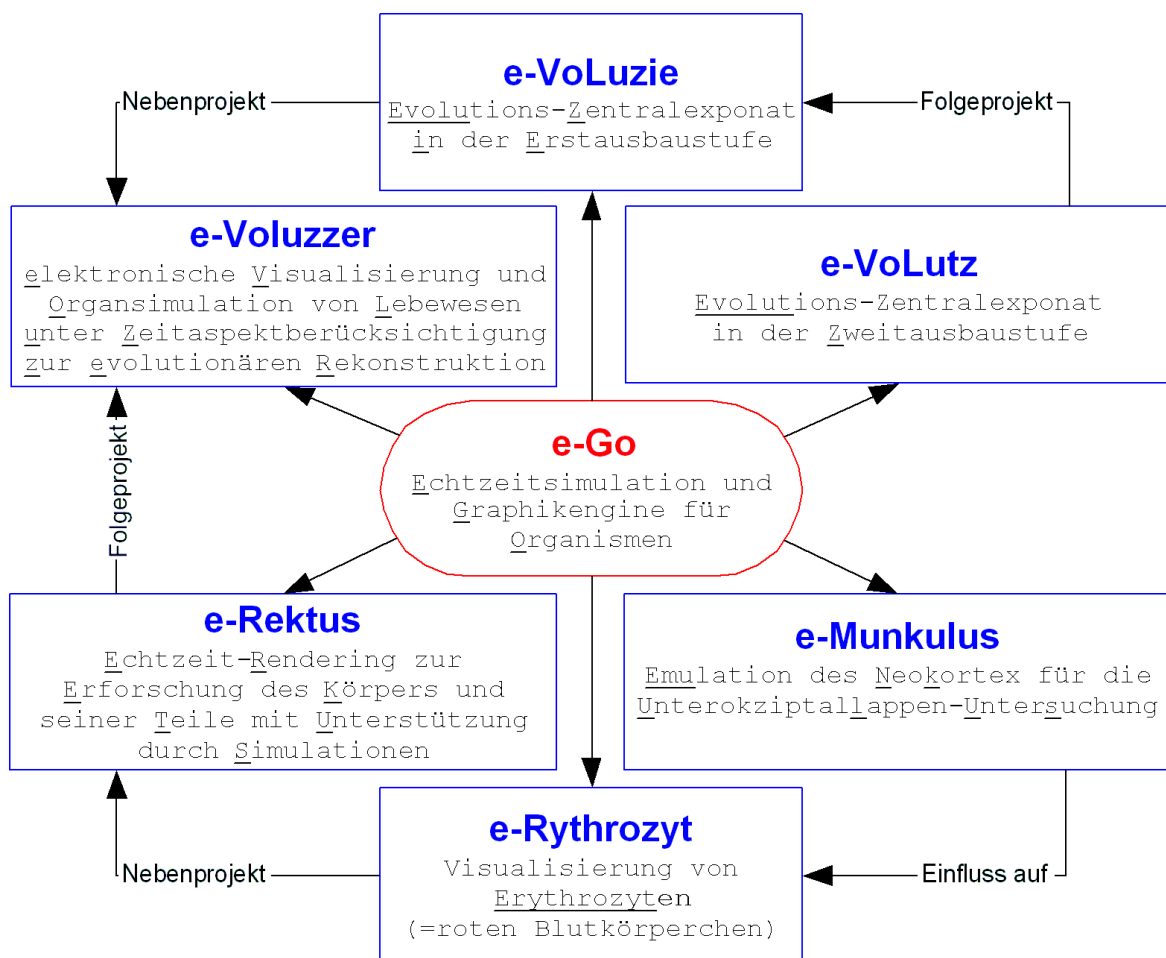


Abbildung 78: Zusammenhang verschiedener auf Quaoaring basierender Projekte

Hier wird in einem Dependenzgraphen aufgezeigt, wie die 7 exemplarischen auf Quaoaring basierenden Projekte e-Go, e-Voluzer, e-VoLuzie, e-VoLutz, e-Rektus, e-Munkulus und e-Rythrozyt miteinander zusammenhängen. Unverkennbar nimmt das Projekt e-Go (rot) eine zentrale Stellung innerhalb der 7 Projekte ein, indem es die restlichen Projekte (blau) koordiniert (unbeschriftete schwarze Pfeile).

Als Schlüsselprojekt dieser Projektzusammenhänge gilt dabei *e-Go*. *e-Go* steht dabei für *Echtzeitsimulation und Graphikengine für Organismen*. Es handelt sich um ein intermediäres Projekt des Instituts für Graphische Datenverarbeitung an der Johann Wolfgang Goethe-Universität in Frankfurt am Main, welches versucht, Synergien zwischen den verschiedenen anderen auf Quaoaring basierenden Projekten wie *e-Voluzzer*, *e-VoLuzie*, *e-VoLutz*, *e-Rektus*, *e-Munkulus* und *e-Rythrozyt* zu generieren. Um einen maximalen Synergieeffekt zu erreichen, werden im Rahmen von *e-Go* in Absprache mit den Teilnehmern anderer Quaoaring-Projekte Normen und Richtlinien erlassen, die zwingend einzuhalten sind. Das Projekt *e-Go* hat damit implizit einen administrativen Charakter. Das Ziel dieser administrativen Tätigkeit ist die Bewahrung der projektübergreifenden Kompatibilität und der Konsistenz der Software und Daten zwischen den verschiedenen Quaoaring-Projekten. So wird beispielsweise auf eine einheitliche Benennung der Instanzen der Quaoaring Design Patterns geachtet, um eine projektübergreifende Austauschbarkeit und Wiederverwendbarkeit von Tierkonstruktionsskripten zu garantieren. Darüber hinaus wird eine Datensammlung von Image- und hydrostatischen Bump Map Texturen angelegt. Dabei wurden strenge Richtlinien und Normen bezüglich Auflösung, Datentyp, Kompressionsart, Farbgebung, Beziehung zwischen Bump-Map- und Image-Textur sowie Benennungskonventionen erlassen, um die Kompatibilität der Texturen über Projektgrenzen hinaus zu wahren. Als Grundlage dieser Maßgaben dient ein Manual, welches von allen Projekten gemeinsam verwendet wird. [Breiner 2005]

Eine weitere zentrale Aufgabe von *e-Go* ist die koordinierte Weiterentwicklung und die Softwarepflege der für Quaoaring relevanten Software-Teile im Code. Dazu gehören u.a. die Analyse und Behebung von Software-Issues, die Erhaltung der internen Software-Kohärenz, die Implementierung neuer Features sowie die Adaption an neue Versionen des verwendeten Szenegraphen und anderer verwendeten Fremdsoftware (wie z.B. neue Versionen der SUN-Engine).

Um die Projektarbeiten von *e-Go* von den anderen Projekten klar zu trennen, wurde eine Schnittstelle zwischen der TOP- und der MOB-Softwareschicht (siehe Kapitel 5.5.1) eingeführt. Die TOP-Schicht, welches die für Quaoaring relevante Software enthält, obliegt dabei dem Projektkreis um *e-Go*, für die verschiedenen Module der MOB-Schicht sind die anderen Projekte zuständig.

6.3 e-Voluzzer

e-Voluzzer ist ein auf der Quaoaring-Technologie basierendes Software-Framework zur Analyse und Visualisierung von Evolutionsverläufen. Der Name „e-Voluzzer“ ist dabei ein Akronym von *elektronische Visualisierung und Organsimulation von Lebewesen unter Zeitaspektberücksichtigung zur evolutionären Rekonstruktion*.

e-Voluzzer ist ein interdisziplinäres Kooperationsprojekt zwischen dem Institut für Graphische Datenverarbeitung der Johann Wolfgang Goethe-Universität sowie dem Forschungsinstitut Senckenberg in Frankfurt am Main. Wie das Akronym schon nahe legt, ist das Ziel von e-Voluzzer, ausgewählte Evolutionsverläufe zu rekonstruieren, respektive auf der Basis der Quaoaring-Technologie die Körperkonstruktionen der in dieser Linie befindlichen Organismen dreidimensional zu modellieren und ihre Bewegungen in Echtzeit zu simulieren. Die Quaoaring-Technologie wird weiterhin dazu genutzt, evolutionäre Transformationen zu visualisieren und diesbezügliche rekonstruierte Zwischenstadien analytisch zu evaluieren. Die von Evolutionswissenschaftlern postulierten Tierkonstruktionen werden dazu mit Hilfe von e-Voluzzer visualisiert und die entsprechenden Organismenkonstruktionskripte in einer Datenbank abgelegt. Durch die dreidimensionale Visualisierung werden mögliche Konstruktionsfehler sofort offensichtlich.

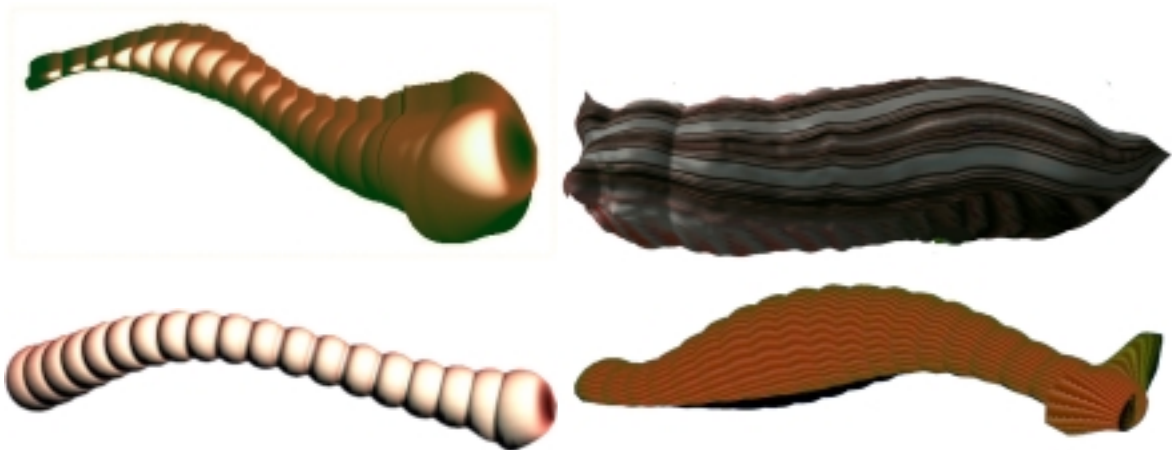


Abbildung 79: Visualisierungen von Würmern, Quappen und Egel

Vier Beispiele für interaktive Visualisierungen von Würmern, Quappen und Egel, welche im Rahmen von e-Voluzzer mit Hilfe von Quaoaring entstanden sind.

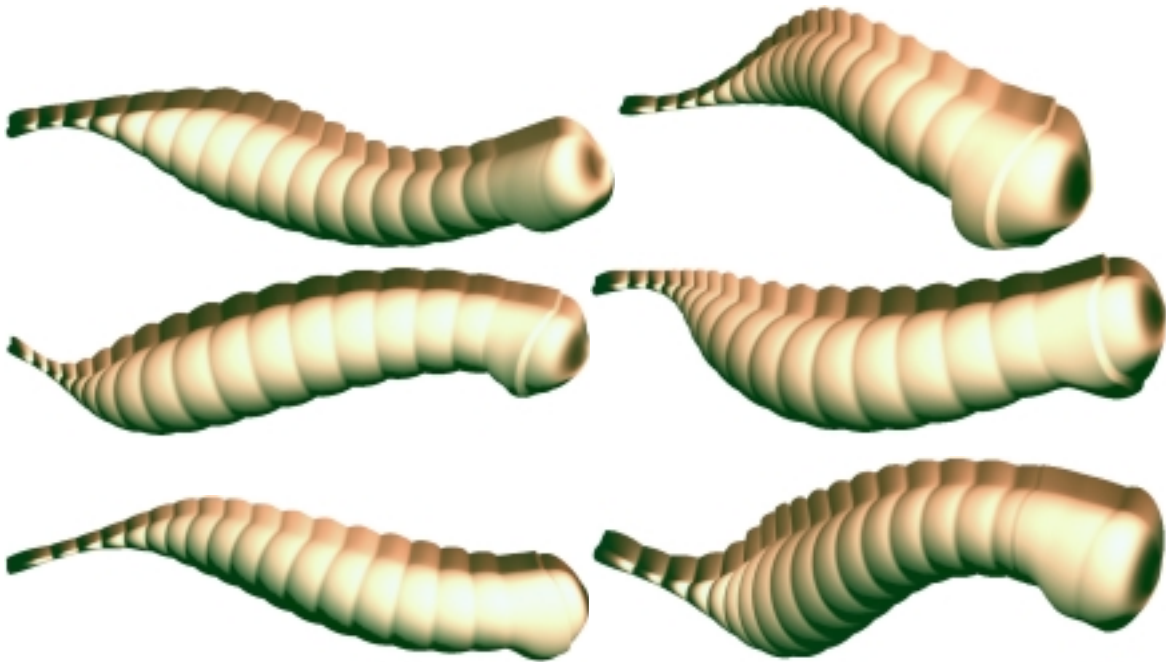


Abbildung 80: Animation peristaltischer Bewegungen

Anbei einige Standbilder eines Wurmes, welcher peristaltische Bewegungen ausführt. Der Wurm wurde in Zusammenarbeit mit dem Forschungsinstitut Senckenberg als didaktische Animation erstellt.

Zudem können durch die in e-Voluzzer integrierte Strömungs- und Muskelkraftsimulation Schwimmbewegungen ausgestorbener Tierarten nach ihrer Effektivität und in der Folge nach ihrer Plausibilität analysiert werden. Somit können evolutionäre Hypothesen besser verifiziert bzw. falsifiziert werden.

Die Erstellung der Tierkonstruktionen erfolgte bislang durch den Autor. Realisiert wurden bislang der Evolutionsverlauf zwischen Achsenstabkonstruktionen und den ersten Enteropneusten. Die Evolutionslinie der Seesterne ist zurzeit in Arbeit.

Die Erstellung wird schrittweise in die Hand der Biologen selbst gelegt. Dazu werden die Biologen in Kursen geschult, um sie in die Lage zu versetzen, selbst Tierkonstruktionen zu erstellen. Neben einer höheren Quantität des Contents hilft die direkte Erstellung der Tierkonstruktionen dazu, die Theorien hinsichtlich der Evolutionsverläufe präzise in Konstruktionsskripte zu übersetzen und in der Folge die biologischen Konzepte zu konkretisieren. Darüber hinaus dienen die Visualisierungen der Organismen als dreidimensionale visuelle Diskussionsgrundlagen und treiben auf diese Weise den dialektischen Diskurs in den Evolutionswissenschaften voran.

Neben den direkten wissenschaftlichen Zielen, können ausgewählte Screenshots und Filme, die während der interaktiven Visualisierung gewonnen wurden, auch zur Erstellung von Postern für die Öffentlichkeitsarbeit oder für didaktische Lehrmaterialien verwendet werden.

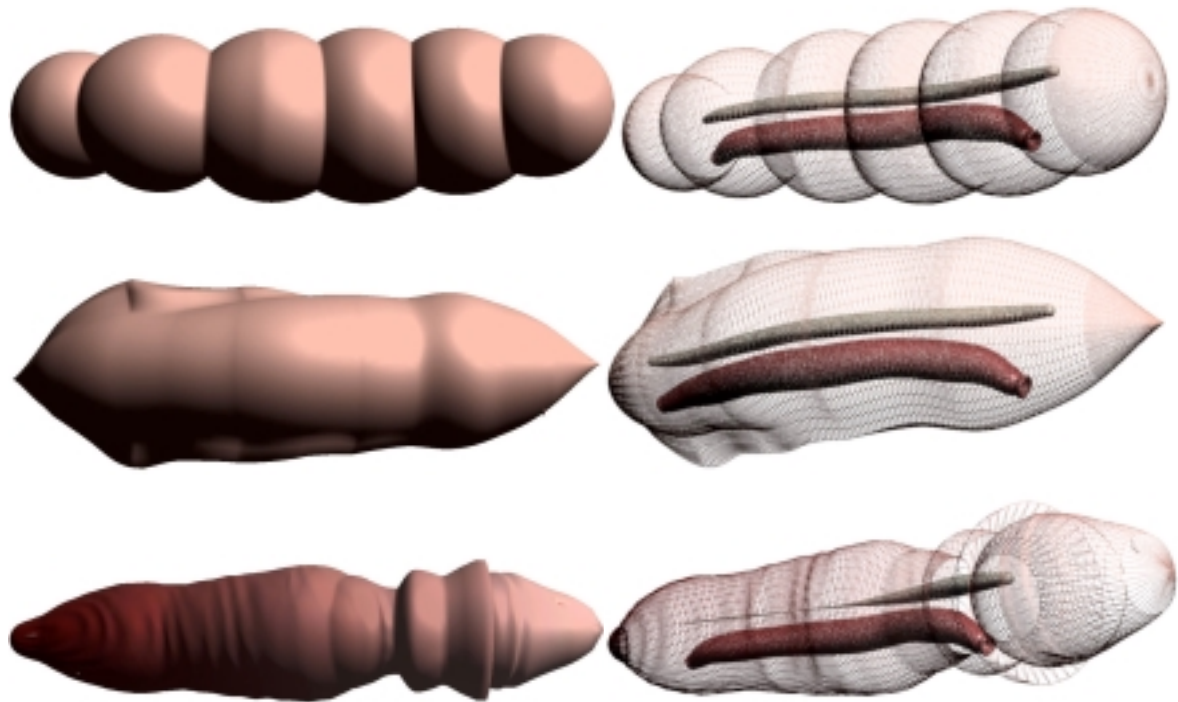


Abbildung 81: Visualisierung der Evolution der Enteropneusten

In Zusammenarbeit mit dem Forschungsinstitut Senckenberg wurde die Evolution der Enteropneusten nachgebildet. Links sind jeweils drei Stadien der Enteropneustenevolutionslinie zu sehen, rechts ist das entsprechende Innenleben mit Chorda (grau) und Intestinaltrakt (weingelb) erkennbar.

Die Chorda, die sich caudal im Laufe der Evolution zurückbildet, ist als Organ modelliert. Der untere Intestinaltrakt wurde mit einem prototypischen Canal realisiert. Es ist erkennbar, dass sich sowohl das Organ als auch der Canal sinnfällig mit den sinoiden und peristaltischen Bewegungen mitbewegt.

6.4 e-VoLuzie

Das *Cybernasium* in Darmstadt ist ein Museum für Computergraphik. Am ehesten wird die Zielrichtung des Cybernasiums durch seine Internet-Selbstdarstellung verständlich:

„Neueste Technologien und faszinierende Wissenschaft interaktiv erleben und dabei Neues erfahren – das ist das Konzept des Cybernasiums. Hier können Sie einzigartige Exponate mit Technologien der Virtuellen und Erweiterten Realität kennen lernen und begreifen, wie digitale Erlebniswelten unser Leben und Lernen in Zukunft beeinflussen werden.

Auf 600m² eignen Sie sich neueste Erkenntnisse intuitiv und spielerisch an. Das Cybernasium schafft dabei einen unbeschwerten Zugang zu moderner Wissenschaft, ermöglicht Hochtechnologie zum Anfassen und baut somit Hemmschwellen gegenüber Wissenschaft und Technik ab.

Erfahren Sie virtuelle Welten als ganzheitliches Erlebnis, das alle Sinne anregt. Sie erwartet eine Mischung aus Forschung und Fun, Lehre und Leidenschaft, die für jedes Alter etwas bietet.“ [Cybernasium 2005]

Im Dezember 2003 fragte das Cybernasium beim Institut für Graphische Datenverarbeitung der Johann Wolfgang Goethe-Universität in Frankfurt am Main nach, ob das Institut ein VR-Exponat für den Museumsbereich „Forschung und Fun“ mit Hilfe der Quaoaringtechnologie entwickeln könne. Eine wichtige Maßgabe dafür war, dass das Exponat auf der zum Zeitpunkt der Anfrage schon vorhandenen Holoscheibenprojektion laufen sollte. Das Sichtsystem „Holoscheibe“ verlangte nach einem hellen Einzelobjekt vor 100% schwarzen Hintergrund.

Es war weiterhin zu beachten, dass das Exponat tagtäglich mehrere Stunden von den Museumsbesuchern benutzt würde, unter anderem auch von Kindern und Jugendlichen. Daher musste es einerseits „kinderleicht“ zu bedienen und andererseits auch extrem robust gegenüber Falscheingaben und Sabotageversuchen sein.

Mit diesen Maßgaben wurde – unter biologischer Beratung des Naturkundemuseums Senckenberg in Frankfurt – entschieden, einen virtuellen Protoplesiosaurus zu entwickeln. Um den Saurier zu personalisieren, wurde er *e-VoLuzie* genannt. *e-VoLuzie* steht dabei für Evolutions-Zentralexponat in der Erstaustufe. Die Entscheidung für dieses Tier erfolgte aus folgenden Gründen:

- Protoplesiosauri haben eine ausreichend „gefährliche“ und „starke“ Anmutung, um Kinder und Jugendliche zu faszinieren.
- Protoplesiosauri sind maritime Saurier, benötigen somit keinen Untergrund zum laufen und können daher glaubwürdig schwimmend vor einem schwarzen Hintergrund dargestellt werden.
- Protoplesiosauri lebten u.a. in einem Meer, welches sich im Bereich der Rheinebene erstreckt hat, und haben daher auch einen direkten Bezug zu Darmstadt.
- Zum Zeitpunkt der Anfrage war ein ähnlicher Phantasieorganismus (ein haariger Drachen) vorhanden, der nur noch mit einem Delta-Skript modifiziert werden musste, um einen Protoplesiosaurus zu erhalten, so dass der Arbeitsaufwand minimal war.
- Über Protoplesiosauri ist bislang aus paläontologischer Sicht nur wenig bekannt, daher konnte der Phantasie (insbesondere bei Hautfarbe und Hautbeschaffenheit) in weiten Teilen freien Lauf gelassen werden.

Um die Stabilität und die Wartbarkeit der Software zu erhöhen, wurden zusätzliche Debugging-Werkzeuge eingeführt. So zum Beispiel ein umfangreiches Log-File, welches die Software bei jedem Start neu generiert, als auch ein automatisches Memory-Leak-Detektor-Tool.

Um Sabotageversuche und Systemabstürze durch Tastaturbuffer-Overflows von vorneherein auszuschalten, wurde die gesamte Eingabe auf Direct Input umgestellt, und nur sinnvolle Tastatureingaben an den Quaoaring-Kern der Software weitergereicht. Zusätzlich wurde auf die Tastatur wurde eine Maske aufgesetzt, welche nur dezidierte Eingaben erlaubt. Die Maske hat dabei den zusätzlichen Nebeneffekt, dass die Bedienung vereinfacht wird. Mit Hilfe dieser Maßnahmen konnte die Stabilität und Robustheit der Software so erhöht werden, dass das Exponat während des anderthalbjährigen Betriebes ununterbrochen fehlerfrei lief und kein einziges Mal neu hochgefahren werden musste.

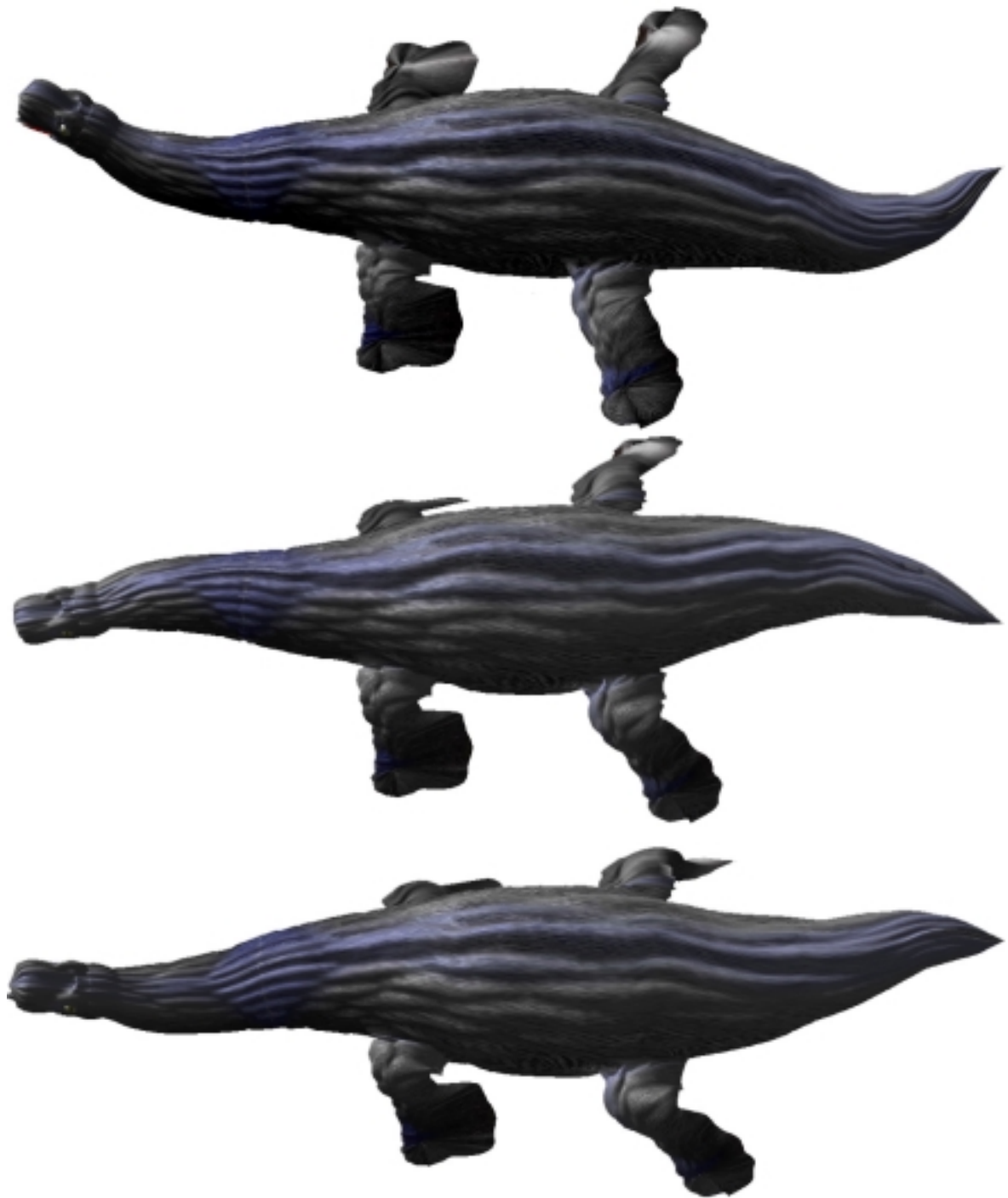


Abbildung 82: Drei Bewegungsstationen von e-VoLuzie

Drei exemplarische Stationen der Bewegungsanimation des Protopleosauiers namens e-VoLuzie, welcher als Exponat für das Cybernarium ausgewählt wurde. Neben den sinoiden Hals- und Schwanzbewegungen vollführt e-VoLuzie mit allen vier Extremitäten gleichzeitig Flossenschläge, um sich vorwärts zu bewegen.

6.5 e-VoLutz

Im Sommer 2005 verlangte die Leitung des Cybernariums nach einem weiteren Exponat, das e-VoLuzie, welche im Kapitel 6.4 behandelt wurde, ergänzt. Die zwei Exponate sollten abwechselnd auf einer Holo­scheibe präsentiert werden. Da der weibliche Name e-VoLuzie zu diesem Zeitpunkt schon feststand, bot sich für das zweite Exponat fast zwingend die männliche Bezeichnung *e-VoLutz* an. Der Name ist eine Abkürzung für *Evolution-Zentral­exponat in der Zweitausbaustufe*. e-VoLutz ist somit das direkte Nachfolgeexponat zu e-VoLuzie.

Diesmal wurde – unter biologischer Beratung von Dr. Michael Gudo von der Morphisto GmbH. in Zusammenarbeit mit PD Dr. Eberhardt Frey vom Staatlichen Museum für Naturkunde Karlsruhe – für einen giftgrünen Nothosaurus Mirabilis als Spezies entschieden. Dabei gab es folgende Gründe zur Wahl dieses Tieres:

- Nothosauri haben – ganz im Gegensatz zu Protoplesiosauri – einen maskulinen aggressiven Gesichtsausdruck, so dass der männliche Name e-VoLutz passt. Sie sehen zudem mit ihrem Revolvergebiss ähnlich „gefährlich“ wie Protoplesiosauri aus, so dass sie aufgrund dieser Attribute Kinder und Jugendliche zu faszinieren vermögen.
- Nothosauri sind wie Protoplesiosauri maritime Saurier und können daher glaubwürdig schwimmend vor schwarzen Hintergrund dargestellt werden, der für eine Holo­scheibenprojektion zwingend ist.
- Nothosauri unterscheiden sich von der Form her stark von Protoplesiosauri, so dass sich e-VoLutz schon auf den ersten Blick rein optisch von e-VoLuzie abhebt, trotzdem ist die Grundtopologie der Gliedmassen ähnlich, so dass wesentliche Teile der Skripte übernommen werden konnten.
- Es existieren (im Gegensatz zu vielen andern Sauriern) konkrete Theorien über die genauen Bewegungsabläufe des Nothosaurus. Dies erlaubte eine realitätsnahe Umsetzung von Benutzereingabeparametern in Schwimmbewegungen. Zusätzlich können die Theorien für Paläontologen verifiziert und für Besucher veranschaulicht werden.



Abbildung 83: Das Exponat e-VoLutz im Cybernarium

Für das Cybernarium wurden mit Hilfe der Quaoaring-Technologie die Exponate e-VoLuzie und e-VoLutz erschaffen. Die Bewegungen von e-VoLutz, der hier zu sehen ist, lassen sich von Museumsbesuchern interaktiv mit Steuerrad und Pedalen lenken. Die Kamerasteuerung sowie die evolutionären Veränderungen geschehen mit Hilfe von 15 zusätzlichen Knöpfen. Als Vorlage für e-VoLutz diente der maritime Saurier „Nothosaurus Mirabilis“.

Da e-VoLutz das Nachfolgeexponat von e-VoLuzie ist, konnten die Erfahrungen, die in den anderthalb Jahren von e-VoLuzie gewonnen wurden, in die Entwicklung von e-VoLutz mit einfließen:

Ein häufig geäußertes Kritikpunkt an e-VoLuzie war, dass sie unrealistisch „fett“ aussah. Die fettige Körpergestalt lag in wesentlichen Teilen daran, dass sich der Funktionsverlauf der bis dahin verwendeten statischen Blob-Funktionen, der für die genaue Ausprägung der Deformationswulste der morphischen Felder maßgebend ist, bei der Modellierung nicht variieren ließ. Dies führte zu der Erkenntnis, dass die Verwendung statischer Blob-Funktionen nicht ausreichend ist, um Wirbeltiere glaubwürdig zu modellieren. Um dieses Manko zu beheben, wurde statt einer statischen eine mit einem d -Parameter toggelbare Blob-Funktion (siehe Kapitel 4.3.2) eingeführt.

Bewährt hatte sich indessen das umfangreiche Sicherheitskonzept von e-VoLuzie zur Erhöhung der Hard- und Software-Robustheit, der Softwarestabilität und der Sabotageabwehr (siehe Kapitel 6.4). Dieses Konzept wurde daher auch bei e-VoLutz übernommen. Allerdings wurde diesmal – zusätzlich zur Tastatureingabe – eine intuitive

Bewegungssteuerung des Sauriers per Lenkrad, Hebel und Pedalen hinzugefügt, einerseits um e-VoLutz von e-VoLuzie abzuheben, andererseits um die Interaktivität und Attraktivität des Exponats zu steigern.

Um die Bewegungssteuerung für die Museumsbesucher möglichst intuitiv zu gestalten, wurde die Metapher der Steuerung eines Schiffes verwendet. Wird das Steuerrad im Uhrzeigersinn gedreht, wendet der Nothosaurier seinen Kopf nach rechts und schwimmt in diese Richtung. Bei einer Drehung des Steuerrads gegen den Uhrzeigersinn schwimmt der Nothosaurier dementsprechend nach links. Drückt man das Gaspedal erhöht sich die Flossenschlagfrequenz und damit beschleunigt der Nothosaurus, drückt man das Bremspedal, stellt der Nothosaurus seine Flossen quer und verlangsamt sich durch seine damit reduzierte Stromlinienform.



Abbildung 84: Screenshot des Elasmosaurus

Die auffälligsten anatomischen Unterschiede des Elasmosaurus gegenüber seinem evolutionären Vorläufer, dem Nothosaurus, sind seine Größe und sein langer schlangenartiger Hals. Der Elasmosaurus wurde mittels einer allgemeinen Volumenerhöhung in Verbindung mit einem zusätzlichen Conic am Hals des Nothosaurus erzeugt. Zusätzlich wurden kleinere Form- und Textur- und Farbmodifikationen vorgenommen, um die neue Spezies zu erzeugen.



Abbildung 85: Visualisierung einer Nothosaurier-Evolutionslinie

Es ist eine Gesamtvisualisierung der Evolutionslinie vom Nothosaurier (grün) über den Postnothosaurier (rot und grün) bis hin zum Elasmosaurier (braun) zu sehen, die quasi als Nebenprodukt des Projektes e-VoLutz entstand. Bei e-VoLutz werden drei Animationsformen frei miteinander gemischt: erstens eine evolutionäre Animation, zweitens eine Wachstumsanimation und drittens eine Bewegungsanimation mit U-Turns, Loopings und Flossenschlägen. Diese einfache Kombinationsmöglichkeit verschiedener Animationsformen ist eine der besonderen Stärken von Quaoaring.



Abbildung 86: Beispiel für die Kohärenz von Zwischenstadien

Evolutionäre Zwischenstadien verhalten sich beim Quaoaring sinnfällig. Dies unterscheidet Quaoaring von reinen Interpolationstechniken (z.B. Keyframing, Blending und Morphing). So bewegt sich der intermediäre Organismus zwischen Nothosaurus und Elasmosaurus aus dem Projekt e-VoLutz plausibel. Hier sind als Beispiele für die intermediäre Kohärenz dieser Zwischenspezies drei aufeinander folgende Bewegungsstadien einer aufwärtsgerichteten „Schnappbewegung“ zu sehen.

Damit diese Bewegungssteuerung glaubwürdig umgesetzt wird, musste eine physikalische Strömungssimulation zusätzlich zur vorhandenen Muskelsimulation in die Software integriert werden. Dies hatte den positiven Nebeneffekt, dass die paläontologischen Theorien der Bewegungsabfolgen des Flossenschlages, die von Herrn PD Dr. Eberhardt Frey entwickelt wurden, im Kern verifiziert werden konnten. Es wurden darüber hinaus neue Erkenntnisse über das präzise Timing des Flossenschlages gewonnen. Die Simulation verschiedener Bewegungsmuster ergab, dass der Nothosaurus für eine energieökonomische Vorwärtsbewegung seine Flossen nach dem Schlag einige Sekunden an den Körper angeschmiegt halten musste (wie auf der Abbildung 83). Die damit verbundene Verbesserung der Aquadynamik erlaubte es dem Nothosaurus, den durch den vorherigen Flossenschlag erzeugten Bewegungsschub lange auszunutzen.

Eine weitere Erfahrung, die im Zusammenhang mit der Farbgebung des Protoplesiosaurus gewonnen wurde, war, dass seine Hautfarbe für die Holo Scheibenprojektion nicht hell und leuchtend genug war, so dass die Beamerintensität für eine ausreichende Sichtbarkeit sehr hoch eingestellt werden musste, was negative Folgen für den schwarzen Außenbereich zur Folge hatte. Aus diesem Grund wurde diesmal statt einer dezenten Farbgebung eine grelle Schattierung zwischen Giftgrün und Hellorange gewählt.

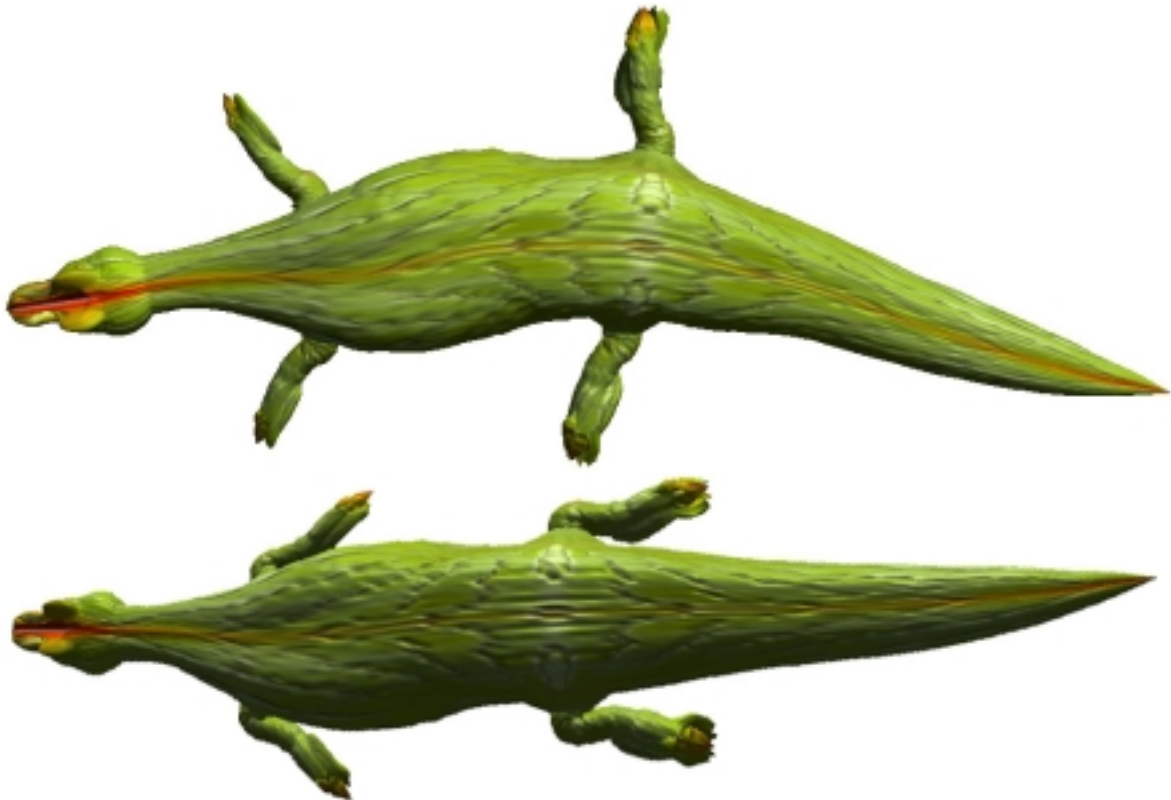


Abbildung 87: Flossenschlag des Nothosaurus

Die paläontologischen Theorien über die konkreten Bewegungsabfolgen des Flossenschlages des Nothosaurus Mirabilis konnten durch die in e-VoLutz integrierte Strömungssimulation im Kern verifiziert werden. Es wurden darüber hinaus neue Erkenntnisse über das präzise Timing des Flossenschlages gewonnen. Hier zwei Stadien des synchronen Flossenschlages.

6.6 e-Rektus

e-Rektus ist ein Teilprojekt des Hauptprojektes *k-Med* im Rahmen einer Pilotstudie. Das Akronym *e-Rektus* steht dabei für *Echtzeit-Rendering zur Erforschung des Körpers und seiner Teile mit Unterstützung durch Simulationen*. Der Name *k-Med* ist eine Abkürzung für *Knowledge-based Multimedia Medical Education*.

Hauptziel des Projektes *k-Med* ist der Aufbau eines virtuellen Fachbereichs Medizin zur Unterstützung der Lehre. Experten aus verschiedenen Teilbereichen der Medizin sollen mit Unterstützung von Instituten aus dem Bereich der Computergraphik und Psychologie in die Lage versetzt werden, eigenständig internetbasierte Lehr- und Lernangebote für die medizinische Aus-, Fort- und Weiterbildung zu generieren.

Das Projekt *k-Med* wurde vom Bundesministerium für Bildung und Forschung von Januar 2001 bis April 2004 sowie vom Hessischen Ministerium für Wissenschaft und Kunst (inklusive Vorgängerprojekten seit 1999) unterstützt. Die beteiligten Projektpartner sind neben der Justus-Liebig-Universität in Gießen als federführende Institution die Johann Wolfgang Goethe-Universität in Frankfurt am Main, die Technische Universität und die Fachhochschule in Darmstadt, die Philipps-Universität in Marburg, die Friedrich Alexander-Universität in Erlangen und die Westfälische Wilhelms-Universität in Münster. [Kmed 2005]

Ziel der Pilotstudie *e-Rektus*, welche vom Institut für Klinische Pharmakologie und Toxikologie in Zusammenarbeit mit dem Institut für Graphische Datenverarbeitung an der Johann Wolfgang Goethe-Universität in Frankfurt am Main betreut wurde, war die Evaluierung der Möglichkeit der Erstellung eines interaktiven virtuellen 3D-Patienten, der vielseitig für die medizinische Ausbildung verwendet werden kann. Der Hauptfokus sollte einerseits auf den Bewegungsapparat gelegt werden, respektive das Skelett, die Muskeln, sowie die Sehnen und Bänder, und andererseits auf die Behandlung metabolischer und regulatorischer Netzwerke.

Zum Zeitpunkt des Starts von *e-Rektus* gab es zwar bereits eine Vielzahl von Projekten, deren Inhalt die 3D-Darstellung der menschlichen Anatomie ist. (z.B. Virtual Human. 3D-Anatom u.a.) Jedoch gingen die meisten Anwendungen nicht über die reine graphische Darstellung und die Animation dieser 3D-Modelle hinaus. Zudem existierten keine Ansätze zur Umsetzung einer „intelligenten“ 3D-Simulations-Anwendung, welche die physikalischen Grundprinzipien berücksichtigt, die hinter solchen Bewegungsabläufen stehen, bzw. biochemische und pharmakologische Aspekte integriert.

Als erster Prototyp für e-Rektus wurden zuerst ausgesuchte Organe eines virtuellen Patienten für die Pharmakologie realisiert, an dem Studierende pharmakologische Experimente vornehmen können. Zu diesen mittels Quaoaring erstellten Organen gehören die Nieren, die Nebennieren, die Leber und das Herz, da sie pharmakologisch relevant sind. Sie wurden jeweils mit Hilfe eines Quaoaring Design Pattern „Main“ und weiteren Quaoaring Design Patterns des Typs „Squeeze“ realisiert, die auf den Main appliziert wurden. Andere Arten der Quaoaring Design Patterns waren nicht notwendig, da nur die grobe Form der Außenhülle inklusive der Farbe visualisiert werden musste. Die Organe sollten sowohl einzeln verfügbar sein, als auch kombiniert als erste realisierte Bausteine für den virtuellen Patienten dienen.

Zu den Experimenten zählen auch solche, die sich an lebenden Personen aus ethischen Gründen verbieten, aber Medizinstudenten ein intuitives Verständnis der pharmakologischen Vorgänge in vivo vermitteln. Die wichtigsten medizinisch relevanten Parameter des virtuellen Patienten (Alter, Gewicht, Größe und Insuffizienzen) sind dabei einstellbar.

Die Studierenden haben die Möglichkeit, dem virtuellen Patienten verschiedene Gifte, Drogen oder Medikamente zu verabreichen. Sie können die Dosis, den Zeitpunkt, die Applikationsform, die Applikationsdauer sowie verschiedene andere Parameter (z.B. Verunreinigungen des Stoffes) frei aussuchen.

Die Studierenden können auf diese Weise spielerisch die Auswirkungen verschiedener Medikamente und Toxine auf menschliche Individuen testen. Die Simulation erfolgt dabei unter Berücksichtigung zeitlicher und quantitativer Aspekte.

Ebenso kann eine Rekonstruktion von pharmakologischen Abläufen unter gezieltem Ausschluss von einzelnen Organkomponenten erfolgen. Auf diese Weise kann pathologisches Verhalten simuliert und veranschaulicht werden.

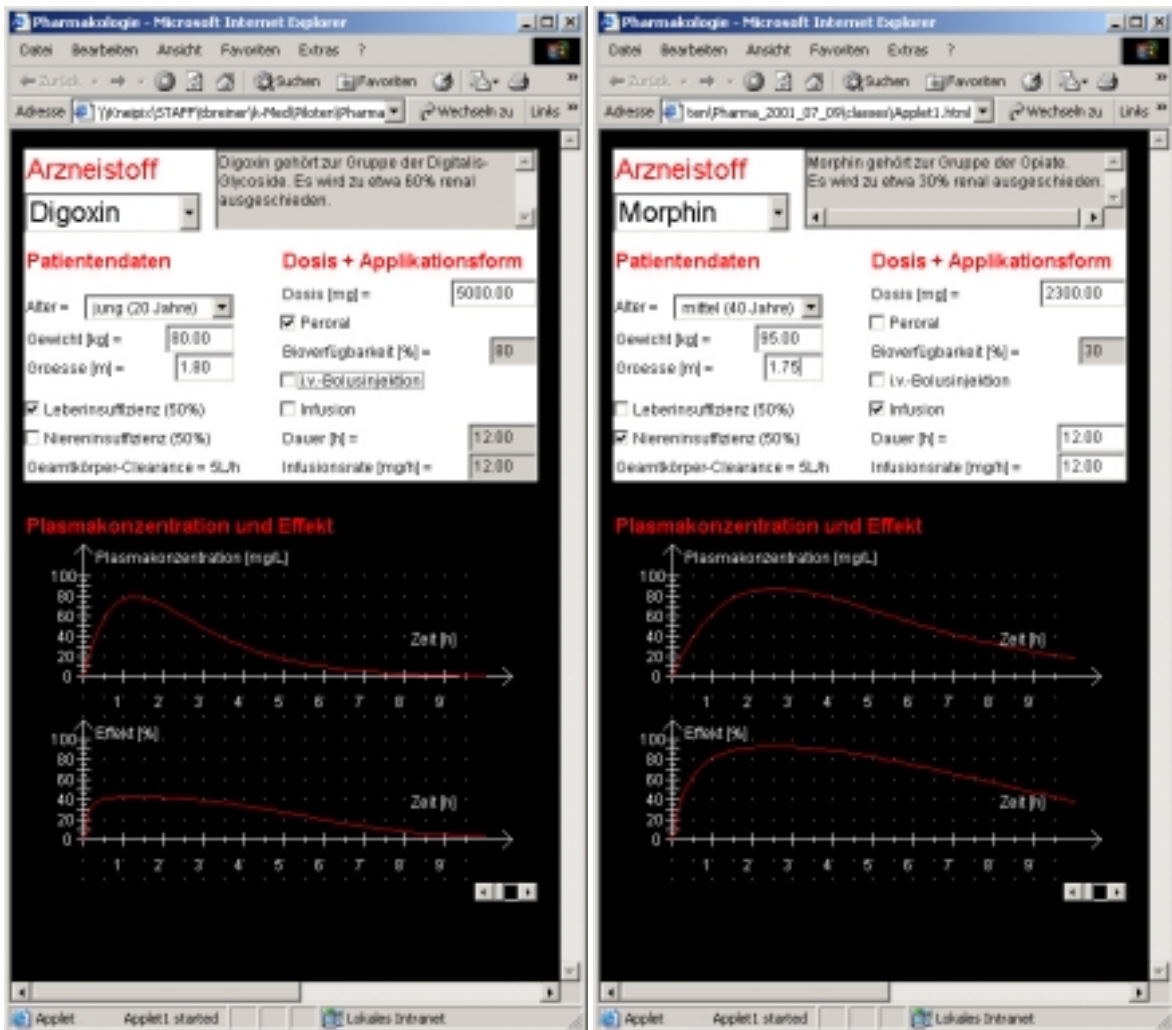


Abbildung 88: Screenshots des k-MED Pharmakologie-Piloten

Zwei Screenshots des Pharmakologie-Piloten. Die vorbereitenden Lehrmodule wurden mit Hilfe von HTML und Macromedia Flash erstellt. Die eigentliche Pharmakologiesimulation ist dagegen in Java geschrieben. Mit ihrer Hilfe können Studierende an virtuellen Patienten Experimente vornehmen. So können sie den virtuellen Patienten Gifte oder Medikamente verabreichen. Art, Dosis, Applikationsform und –dauer sowie verschiedene medizinisch relevante Parameter des virtuellen Patienten (Alter, Gewicht, Größe und Insuffizienzen) sind dabei einstellbar. Die Studierenden können dann die zeitlichen Auswirkungen verschiedener Medikamente und Toxine auf menschliche Individuen testen. Angedacht war die Kopplung der Simulation mit Organen eines 3D-Patienten, die mit Hilfe der Quaoaring-Technologie erstellt sind.

Geplant war eine interaktive Koppelung des pharmakologischen Simulators mit den exemplarischen Organen. Mit einer Software-Schnittstelle konnte der Zustand und in der Folge auch das Aussehen der Organe kontrolliert werden. Die Studierenden sollten so anschaulich die Effekte ihrer Experimente auf die einzelnen Organe studieren.

Nach einer ersten Pilotanwendung für die Pharmakologie (siehe Abbildung 88) wurde jedoch die Quaoaring-Integration verworfen. Dies lag vornehmlich am Zeitpunkt des Projektanfanges von k-Med, bei der Quaoaring noch nicht ausgereift war, so dass die Erstellung eines ganzen Menschen mit Quaoaring unrealistisch erschien.

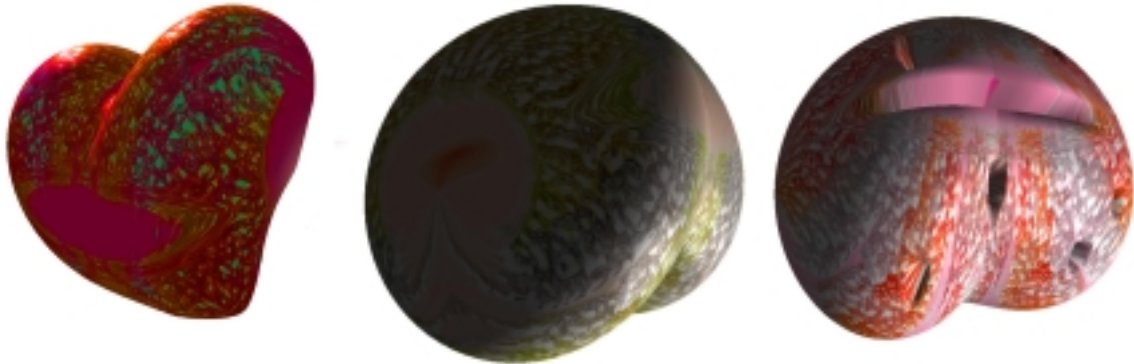


Abbildung 89: Symbolische Anzeige von Organstadien

Es sind drei Organstadien für die Verwendung im Projekt e-Rektus zu sehen. Bei der interaktiven Simulationsvisualisierung kam es eher auf symbolische Prägnanz als auf Realistik an. Links: Herz; Mitte: Gesunde Leber; Rechts: Leber mit Zirrhose

6.7 e-Munkulus

Das Projekt *e-Munkulus* ist ein interdisziplinäres Forschungsvorhaben des Instituts für Graphische Datenverarbeitung des ehemaligen Fachbereiches „Biologie und Informatik“ sowie der Abteilung Computational Neuroscience des Fachbereichs Biologie der Johann Wolfgang Goethe-Universität in Frankfurt am Main, der ATIP Moderne Technologien zur Informationsverarbeitung GmbH in Frankfurt am Main sowie der Abteilung “Auditorial Physiology” des Max-Planck-Instituts für Neurobiologie in Martinsried. Der Name *e-Munkulus* ist dabei eine Abkürzung für *Emulation des Neokortex für die Unterokzipitallappen-Untersuchung* und gleichzeitig eine historische Anspielung auf die Homunkulus-Legende. Das Forschungsvorhaben wurde schon nach ca. sieben Monaten Laufzeit wegen Finanzierungsschwierigkeiten abgebrochen, trotzdem hat es hinsichtlich der Verwendbarkeit des Quaoarings für zelluläre Netzwerke Resultate erzielt.

Das Projekt *e-Munkulus* hatte als Ziel, verschiedene Mechanismen und Strukturen des Gehirns respektive den Primären Visuellen Kortex und die Obere Olive möglichst realitätsnah in Echtzeit unter Zuhilfenahme eines Computernetzwerkes zu simulieren.

Dabei sollten sowohl die exakte räumliche Neuronenanordnung, die synaptische Verschaltung als auch die quantitative und temporäre Interaktionsweise von Neuronenverbänden berücksichtigt werden.

Die Simulation sollte durch eine interaktive Quaoaring-Schnittstelle visualisiert werden, so dass eine optimale Entwicklungsplattform für Informatiker, Neurologen und Biologen entsteht, in der die Emulation mit den biologischen Experimentaldaten verglichen, validiert und feinjustiert werden können. Durch die der Simulation eigenen Besonderheiten wie Interaktivität, Echtzeitfähigkeit, Plastizität, Realitätsnähe sowie eine theoretisch beliebige Komplexität mit freier Skalierbarkeit hob sich die Entwicklungsplattform deutlich von anderen Simulationen neuronaler Netzwerke ab (wie GenSim, NeuroVis, NeoSim etc.) und sollte somit vielmehr eine „Emulation des Gehirns“ darstellen. Aus dieser realitätsgetreuen Gehirnemulation sollten die relevanten Daten extrahiert werden, um ein vereinfachtes Modell eines bioanalogen neuronalen Netzwerkes zu entwickeln, welches die Leistungsfähigkeit biologischer Systeme, insbesondere Robustheit, Fehlertoleranz, Adaptivität und Entwicklungsfähigkeit für die technische Informationsverarbeitung gewinnt.

Darüber hinaus sollen die populärsten Theorien über die Funktionsweise des Gehirns im Allgemeinen und der oben genannten sensorischen Hirnareale im Speziellen wie z.B. das Punktneuronenmodell mit zeitcodierenden Einheiten, das Kompartimentenmodell u.a. verifiziert oder falsifiziert werden, indem diese den Theorien zugrunde liegenden Konzepte auf den Gehirnemulator angewendet und die Ergebnisse mit den bestehenden Experimentaldaten verglichen werden.

Da ein vergleichbares Vorhaben auf diesem Integrationsniveau noch nicht existierte, sollte dieses Projekt eine Pionier- und Katalysatorfunktion in Lehre, Wirtschaft und Wissenschaft einnehmen.

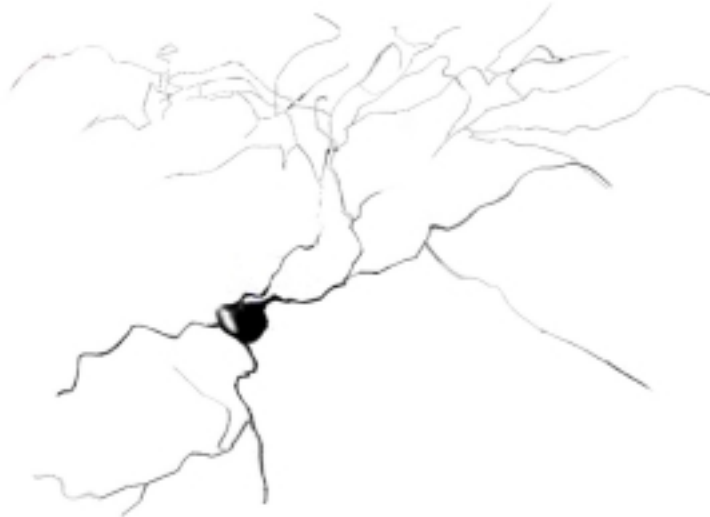


Abbildung 90: Visualisierung eines ausgewachsenen Neurons

Es ist die dreidimensionale Visualisierung eines ausgewachsenen Neurons der Mediale Superiore Olive zu sehen. Quaoaring vermag die Neuronen inklusive ihres Dendritenwachstums besonders realitätsnah zu visualisieren. Allerdings versagt Quaoaring aus Performanzgründen schon bei der interaktiven Echtzeitvisualisierung kleinerer neuronaler Netzwerke mit einer Anzahl von über 10 Neuronen.

Da bei Beginn des Forschungsvorhabens noch keine zellulären Objekte mittels Quaoaring realisiert waren, musste erst die Praxistauglichkeit des neuen Verfahrens anhand konkreter Einzelneuronen und neuronaler Netzwerke demonstriert werden. Dafür wurden drei Pilotvisualisierungen vorgenommen. Das Datenmaterial stammte von der Abteilung Computational Neuroscience des Fachbereichs Biologie der Johann Wolfgang Goethe-Universität in Frankfurt am Main und bestand aus verketteten Einzelpunkten, die durch histologische Schnittanalysen gewonnen wurden. Diese Einzelpunkte wurden für die Positionierung der Stützpunkte der Zentrallinien verwendet. Das Soma wurde mit dem Quaoaring Design Pattern „Main“, die Axone und Dendriten wurden per Quaoaring Design Pattern „Bulge“ in Kombination mit „Conics“ realisiert. Bei jeder dendritischen Verzweigung wird daher ein weiterer Bulge benötigt. Dabei wurden der Einfachheit halber pro angegebene Stützstelle der Zentrallinie genau ein Quaoaring Design Pattern „Conic“ mit negativem Auslenkungsparameter verwendet. Auf diese Weise wurde der schlauchförmige Charakter der Dendriten bzw. Axone erzeugt. Synapsen wurden mit Hilfe verschiedener Quaoaring Design Patterns vom Typ „Squeeze“ generiert. Die Neurotransmitterausschüttung konnte mit modifizierten Particlesources erzeugt werden.



Abbildung 91: Nahaufnahme eines wachsenden Neurons

Zu sehen ist eine mit Quaoaring visualisierte Nervenzelle mit Soma (Kern in der Mitte), Axon (linker großer Ast) und dendritischen Verzweigungen (übrige Äste). Dieses hypothetische Neuron befindet sich noch im Wachstum, so dass sich die Dendriten mit der in Quaoaring integrierten Wachstumssimulation den Weg zu anderen Neuronen bahnen.

Die Pilotvisualisierungen verifizierten die theoretisch vorhergesagten Vor- und Nachteile der Quaoaringtechnologie bei der Visualisierung von Einzelneuronen und dreidimensionalen neuronalen Netzwerken auch in der Praxis, die besagten Vorteile sind dabei:

- Die Form des Somas eines Neurons kann biologisch korrekt visualisiert werden.
- Die Formen der Axone und Dendriten können ebenfalls realistisch visualisiert werden. Darüber hinaus werden fehlende Messwerte des Axon- bzw. Dendritenverlaufes automatisch durch den NURBS-Charakter der Zentrallinie interpoliert.

- Das Wachstum des Axons und der Dendriten ist biologisch korrekt animierbar, darüber hinaus kann die Animation der Wachstumsvorgänge direkt an die Simulation des neuronalen Netzwerkes gekoppelt werden.
- Es sind auch Details der Neuronenformen erzeugbar, wie z.B. Ranviersche Schnürringe oder Synapsen.
- Sowohl das spezifische Aktionspotential innerhalb der Neuronen als auch die Neurotransmitterausschüttung können sehr eindrucksvoll und einsichtig mit Hilfe des Quaoaring Design Patterns „ParticleSystem“ visualisiert werden.

Der Nachteil des Quaoarings zeigt sich bei Visualisierung von neuronalen Netzwerken. Schon ab 10 Neuronen sinkt die durchschnittliche Frame-Rate auf unter 12 fps, so dass nicht mehr von Echtzeitfähigkeit gesprochen werden kann. Ab 50 Neuronen sinkt sie sogar unter 3fps. (Die Daten des betreffenden Referenzrechners, auf denen die Messungen durchgeführt wurden, befinden sich in Kapitel 5.2) Quaoaring ist daher in seiner bisherigen Form nicht für eine Echtzeitvisualisierung komplexer Netzwerke geeignet. Hier besteht Forschungsbedarf in der Zukunft, um Quaoaring auch für die Visualisierung komplexer filigraner Strukturen geeignet zu machen.

Die nächstliegende Möglichkeit zur Beschleunigung des Quaoarings bei filigranen Verzweigungsstrukturen wäre das Einführen von automatisierten Level-of-Detail-Techniken, die speziell auf das Konzept des Quaoarings abgestimmt sind. Zwei unterschiedliche Lösungswege, wie solche automatisierte Level-of-Detail-Techniken im Kontext des Quaoarings realisiert werden könnten, werden im Kapitel 7.1.4 sowie in Kapitel 7.1.5 beschrieben.

Wird eine solche Erweiterung in Betracht gezogen, dürfte auch die Visualisierung komplexerer Strukturen von neuronalen Netzwerken ($n > 1000$) mit Quaoaring schon in naher Zukunft in den Bereich des Machbaren rücken, insbesondere wenn die zusätzliche Beschleunigung der Rechengeschwindigkeit mit dem erweiterten Gesetz von Moore in Betracht gezogen wird. [Moravec 1998a & b][Moore 1997 & 2003]



Abbildung 92: Animation der Depolarisation eines Dendriten

Im Projekt e-Munkulus wurde die Depolarisation mit Hilfe des Quaoaring Design Patterns „Particlesource“ angezeigt. Die nanostrukturellen Oberflächeneigenschaften des Dendriten und der Synapse wurden mit einem Skin erzeugt.



Abbildung 93: Visualisierung eines Neuronales Netzwerkes

Visualisierung eines Ausschnittes mit drei Neuronen eines Neuronales Netzwerkes.

6.8 e-Rythrozyt

Das Projekt *e-Rythrozyt* hat als Ziel, die Formen und Oberflächenstrukturen roter Blutkörperchen (*Erythrozyten*) mitsamt ihren pathologischen Derivaten (Lacrimozyten, Echinozyten, Sphärozyten, Sichelzellen etc.) dreidimensional zu visualisieren.

e-Rythrozyt ist dabei ein interdisziplinäres Kooperationsprojekt zwischen dem Institut für Graphische Datenverarbeitung der Johann Wolfgang Goethe-Universität in Frankfurt am Main, des Pathologischen Instituts der Universität des Saarlandes in Homburg an der Saar (Dr. Mathias Wagner) und dem Universitätsklinikum in Saarbrücken (Prof. Dr. Ingolf Bernhardt).

Die Formen von Zellen resultieren im Wesentlichen aus der Interaktion zwischen der osmotischen Druckdifferenz von intra- zu extrazellulärem Medium, der Phospholipiddoppelmembran sowie dem Zytoskelett und anderen stabilisierenden und verspannenden Strukturen. Daher bietet sich als Visualisierungsbasis bezüglich der Software die Quaoaring-Technologie an, welche diese Interaktionen prinzipiell berücksichtigen kann. Insbesondere das Hydrostatische Bump Mapping des Quaoarings verspricht, durch die Möglichkeit der Visualisierung von Vesikelbildungen die Oberflächenstrukturen realistischer darzustellen als herkömmliche Bump Mapping Techniken.

Als nachteilig erwies sich, dass die Quaoaring-Technologie zum Zeitpunkt des Projektstartes auf anatomische Objekte und Gesamtorganismen spezialisiert war. So existieren bei Zellen im Gegensatz zu Organismen keine caudalen bzw. cranialen Endpunkte, an denen sich das Biologische Koordinatensystem orientieren könnte. Auch das Konzept der Zentrallinie findet bei zellulären Strukturen keine Entsprechung.

Diese Nachteile können allerdings alle umgangen werden, so dass sie kein Ausschlusskriterium darstellen: Das Biologische Koordinatensystem wird einfach willkürlich gewählt; bei der Dimensionierung von morphischen Feldern und Texturen muss die Verzerrung zu den caudalen und cranialen Endpunkten berücksichtigt werden. Die Zentrallinie bleibt undeformiert, was einen leichten Berechnungs-Overhead zu Folge hat. Dies ist jedoch nicht gravierend, denn im Gegensatz zu den anderen auf Quaoaring basierenden Projekten (insbesondere e-VoLuzie, e-VoLutz) steht bei e-Rythrozyt der Aspekt der Echtzeitfähigkeit und Interaktivität nicht im Vordergrund, der Fokus liegt vielmehr auf der Detailgenauigkeit bezüglich des Höhenprofils der

Phospholipiddoppelmembran des Erythrozyten und der Zoombarkeit zwischen verschiedenen Granularitätsstufen.

Nach dem Projektstart wurden am Universitätsklinikum in Saarbrücken von möglichst vielen exemplarischen Erythrozyten mit Hilfe von *Rasterkraftmikroskopen* (engl. *Atomic Force Microscope*, kurz *RKM* bzw. *AFM*) Messungen vorgenommen. Vom Institut für Graphische Datenverarbeitung wurde danach dieses umfangreiche AFM-Datenmaterial (638MB in 266 verschiedenen AFM-Scan-Dateien) gesichtet und auf seine Tauglichkeit für spätere 3D-Visualisierungen bewertet. Die AFM-Höhendaten lagen dabei in unterschiedlichen Formaten vor (TIFF- und JPG-Bilddateien so wie verschiedene ASCII-basierte Tabellen) und waren teilweise redundant. Auch stammten nicht alle Messungen von unverformten Erythrozyten. Dies implizierte, dass zuerst einmal eine geeignete Auswahl innerhalb des Datenmaterials getroffen werden musste. Nach dieser Selektion blieben 78 verwertbare Dateien übrig.

Die resultierenden Dateien wurden zuerst optisch begutachtet, dabei fielen 6 verschiedene Arten von Höhenstrukturen auf. Da sich die Höhenstrukturen über mehrere Dateien verteilten, konnten Mustergeneralisierungen lokal begrenzter Strukturen oder zufälliger Artefakte ausgeschlossen werden. Die sechs Strukturen waren im Einzelnen:

- Die torusähnliche Grundstruktur des Erythrozyten mit ca. 7,92 μm Durchmesser (im Folgenden *Erythrozyten-Primärstruktur* genannt)
- Die Unregelmäßigkeiten der Primärstruktur von ca. 2-4 μm Durchmesser (im Folgenden *Erythrozyten-Sekundärstruktur* genannt)
- Netzartige Falten von 2.3 μm – 4.3 μm Länge (im Folgenden *Erythrozyten-Tertiärstruktur A* genannt)
- Kleine sich gegenseitig überlagernde blasenartige Strukturen zwischen 0.0092 μm Durchmesser / 0.26 nm Höhe und 0,088 μm Durchmesser / 8.2 nm Höhe, welche sich hauptsächlich in den Zwischenräumen der Tertiärstruktur befinden (im Folgenden Tertiärstruktur B genannt)

- Wellenartig repetitive Strukturen von ca. 0.07 μm Länge und 0.3 μm Breite und ca.10 nm Höhe (im Folgenden *Erythrozyten-Tertiärstruktur C* genannt)
- Sehr kleine, streifenartige repetitive Strukturen von ca. 0.012 μm (im Folgenden als *Erythrozyten-Quartiärstruktur* bezeichnet).

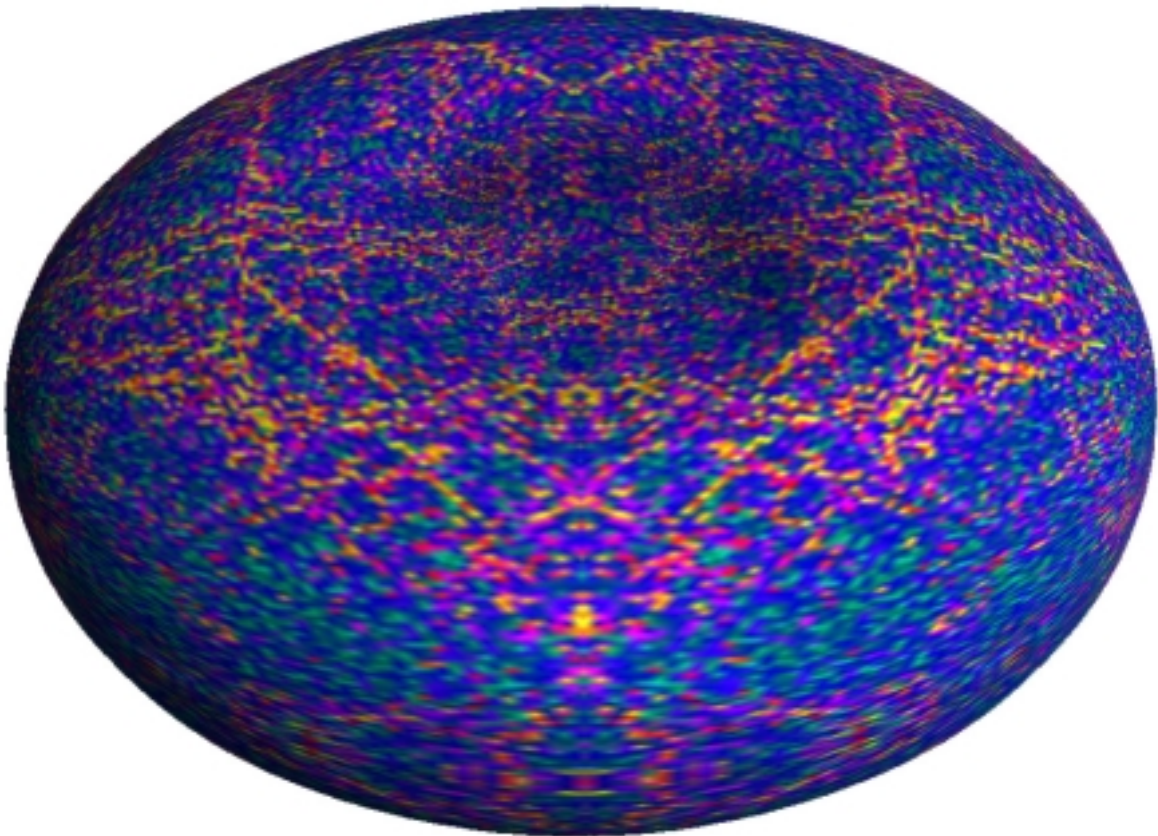


Abbildung 94: Visualisierung eines Discozyten mit Falschfarbencodierung

Quaoaring-Visualisierung eines Discozyten in Falschfarben zur Kennzeichnung der verschiedenen in den AFM-Daten identifizierten Erythrozytenstrukturen. Die Farben haben dabei folgende Bedeutung: *Rot, Orange* und *Gelb*: Erythrozyten-Tertiärstruktur A in unterschiedlichen Stärkegraden (Rot am tiefsten); *Grün*: Erythrozyten-Tertiärstruktur B; *Blau*: Überlagerungen von Erythrozyten-Tertiärstruktur B und C; *Violett*: Überlagerungen von Erythrozyten-Tertiärstruktur B und A.

Die Erythrozyten-Primärstruktur, also die typische „Donut-Form“ des roten Blutkörperchens, wurde durch einen Main-Pneu mit zwei großflächigen morphischen Feldern $F_1 = (S_1; \omega_1; d_1)$ und $F_2 = (S_2; \omega_2; d_2)$ erzeugt. Die Auflösung des Main-Pneus beträgt 1200 Vertices radial und 900 Vertices longitudinal. Als Quaoaring Design Pattern der morphischen Felder wurde „Squeeze“ gewählt. Die Parameter des Biofeldes des ersten

morphischen Feldes liegen bei $S_1 := (c_R=0; c_A=0; c_H=1; r_R=0,572; r_A=4\pi; r_H=0)$, diejenigen des Biofeldes des zweiten morphischen Feldes liegen bei $S_2 := (c_R=1; c_A=0; c_H=1; r_R=0,5,72; r_A=4\pi; r_H=0)$. Als Auslenkungsparameter wurde jeweils $\omega_1 = \omega_2 = -0.723$ gewählt, die Konvexitätsparameter d_1 und d_2 wurden jeweils auf 1,21 gesetzt. Auf diese Weise erzeugen die beiden morphischen Felder zwei diametral entgegengesetzte Einbuchtungen, so dass als Nebeneffekt die Torusform in den restlichen Bezirken entsteht.

Zur Erzeugung der Erythrozyten-Sekundärstruktur wurden weitere 19 morphische Felder $F_3 = (S_3; \omega_3; d_3)$ bis $F_{21} = (S_{21}; \omega_{21}; d_{21})$ verwendet. Die Biofelder dieser morphischen Felder wurden mit einem Zufallsgenerator dabei linear über einen Bereich zwischen $S_A := (c_R=0,1; c_A=0; c_H=1; r_R=5,72; r_A=0,64\pi; r_H=0)$ und $S_B := (c_R=0,9; c_A=2\pi; c_H=1; r_R=5,72; r_A=1,33\pi; r_H=0)$ gestreut. Die Auslenkungsparameter ω_3 bis ω_{21} variierten zwischen 0.13 und 0.21; die Konvexitätsparameter wurden jeweils auf 1 gesetzt. Sowohl bei der Primär- als auch bei der Sekundärstruktur wurden die genauen Parameterwerte empirisch durch einen approximativ-asymptotischen Trial-and-Error-Prozess ermittelt.

Die Erythrozyten-Tertiärstrukturen A bis C wurden jeweils gemeinsam mit Hilfe von hydrostatischem Bump Mapping generiert. Es wurde eine Auflösung von $4048 * 4048$ Pixel mit 8 Bit/Pixel gewählt, damit auch die feinen Strukturen von Tertiärstruktur A, B und C abgebildet werden konnten.

Um die Größenverhältnisse der Strukturtopologien möglichst nahe an die Realität anzulehnen, wurden jeweils 30 exemplarische Messungen des Querschnitts an verschiedenen Stellen der Erythrozyten-Tertiärstrukturen A, B und C durchgeführt. Von diesen Messwerten wurde jeweils der arithmetische Mittelwert errechnet.

Diese Mittelwerte dienten als Parameter für den späteren prozeduralen Generierungsprozess der Bump Map Textur. Die Texturerstellung erfolgte zuerst durch die Technik der Fraktalen Netze, wobei diejenigen Texturpixel, die von Kantenverläufen des Netzmusters berührt wurden, schwarz gefärbt wurden, die restlichen Pixel erhielten die Farbe weiß. Danach wurde die Bump Map Textur mit einem Gaußschen Unschärfefilter (Unschärfedistanz = 4,5 Pixel) behandelt, um das Rinnenrelief des Netzmusters zu glätten. Auf diese Weise entstand das netzartige Höhenrelief für die Tertiärstruktur A. Die Höhenwerte für die Tertiärstruktur B und C wurde mit Hilfe von zweidimensionalen Perlin Noise erzeugt. Korrespondierende Pixelwerte der resultierenden Bump Map Texturen wurden aufeinander addiert und dann zueinander proportional in einem Bereich zwischen 0

und 1 reskaliert. Das Resultat war eine hydrostatische Bump Map, die sowohl Tertiärstruktur A, B als auch C gleichzeitig abbildet.

Um die verschiedenen Strukturen optisch besser erkennbar zu machen, wurde aus dieser Bump Map Textur zusätzlich eine Image Textur mit der gleichen Auflösung erstellt, welche die Strukturen mit verschiedenen Rot-Schattierungen berücksichtigt. Niedrige Höhenwerte der Bump Map Textur korrespondierten dabei mit dunkelroten Bereichen auf der Image-Textur, hohe Höhenwerte dementsprechend mit Hellroten.

Da die topologische Gesamtstruktur der Image-Textur äquivalent zu derjenigen der hydrostatischen Bump Map ist, konnten sie gemeinsam für ein Quaoaring Design Pattern „Skin“ verwendet werden, das an vier verschiedenen Stellen des Erythrozyten mit einem kompetativen Verfahren auf die Primärstruktur appliziert wurde.

Bei einer genaueren Sichtung der Erythrozyten-Quartärstruktur fiel die extreme Genauigkeit und Parallelität der Strukturen auf, die für biologische Strukturen untypisch ist. Es wurde daher vermutet, dass diese Quartärstruktur keine realen, dem Erythrozyten immanenten Höhenstrukturen sind, sondern dass es sich um Abtastungsfehler handelt, die durch das zeilenweise Gleiten des Cantilevers über die Phospholipiddoppelmembran entstehen. Ein Vergleich mit den AFM-Originaldaten bestätigte den Verdacht. Die Zeilenbreite der Struktur entsprach stets einem Vielfachen der Abtastungsauflösung. Die Quartärstruktur wurde daher für die weitere Generierung der Erythrozytenvisualisierung ignoriert.

Parallel zu dieser Quaoaring-Modellierung des Gesamterythrozyten, erfolgte das Rendering. Dabei wurde Gouraud-Shading und der z-Buffer-Algorithmus benutzt. Nachdem sich der Erythrozyt entwickelt hatte, wurden Screenshots erzeugt und als TIFF-Dateien abgespeichert. Diese Bilder wurden nachbearbeitet. Die Nachbearbeitung umfasste Aspekte der Optimierung des Kontrastranges sowie des Bildausschnittes als auch der Retuschierung von Textur- und Hintergrund-Artefakten, insbesondere musste hier die Texturverzerrung an den caudalen und cranialen Endpunkten rückgängig gemacht werden.

Da die Strukturen auf Erythrozyten sehr große Granularitätsunterschiede mit einem Faktor von 860 (von 0,0092 μm bis 7,92 μm) aufweisen, mussten neben der Visualisierung des Gesamterythrozyten, in denen hauptsächlich die Primär- und Sekundärstruktur sowie die Gesamttopologie der Tertiärstruktur A und C sichtbar waren, noch zwei weitere Visualisierungen feinerer Strukturen erfolgen. Es wurden die Zooming-Stufen mit 0,75 μm sowie 130 nm Durchmesser gewählt, da damit die Tertiärstrukturen B und C optimal darstellbar waren und somit alle Strukturen abgedeckt waren.

Ein Zooming mit der Kamera kommt nicht in Frage, da schnell an die Auflösungskapazitäten des Skin-Meshs und der Texturen gestoßen wird. Daher mussten für die Zooming-Stufen zwei weitere undeformierte Main-Pneus genommen werden. Auf einen Teilbereich des Pneus wurde jeweils eine Bump Map appliziert, die direkt und unverfälscht aus den Höhendaten des ausgewählten AFM-Datenmaterials erzeugt wurde. Um zu kaschieren, dass es sich bei dem Übergang zwischen der Visualisierung des Gesamterthrozyten und der zweiten Zoom-Stufe um zwei verschiedene Objekte handelt, wurde im AFM-Datenmaterial nach Oberflächentopologien gesucht, die Teilbereichen des Gesamterthrozyten optisch ähneln.

Analog zu den „normalen“ discozytären Erythrozyten wurden auch ihre pathologischen Verformungen wie Echinozyten, Lacrimozyten, Codozyten, Microstomatozyten, Leptozyten und Reticulozyten erzeugt. Dazu müssen allerdings zusätzliche morphische Feldern für die Deformationen appliziert werden. Bei der Texturgenerierung müssen dabei die mit den morphischen Feldern einhergehenden Verzerrungen berücksichtigt werden.

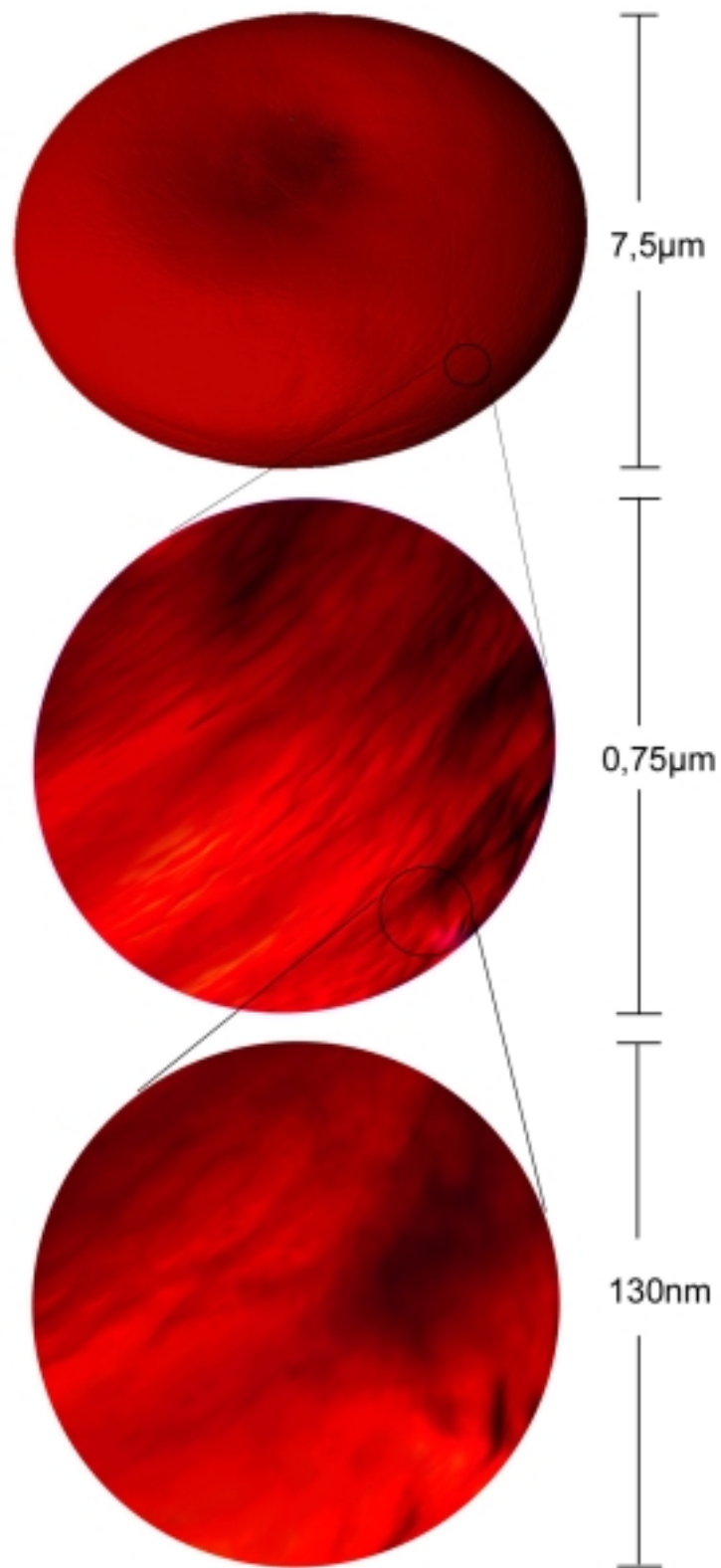


Abbildung 95: Visualisierung eines Erythrozyten in drei Zoom-Stufen

Beispiel einer Erythrozytenvisualisierung mit Hilfe der Quaoaring-Technologie. Aufgrund der stark unterschiedlichen Dimensionen der Strukturen wurden drei verschiedene Zoom-Stufen erstellt (Gesamt-Erythrozyt mit $7,5\mu\text{m}$ Durchmesser und zwei zusätzliche Zoom-Stufen mit 750nm und 130nm). Grundlage für die Erstellung der Bilder war ein umfangreiches AFM-Datenmaterial.

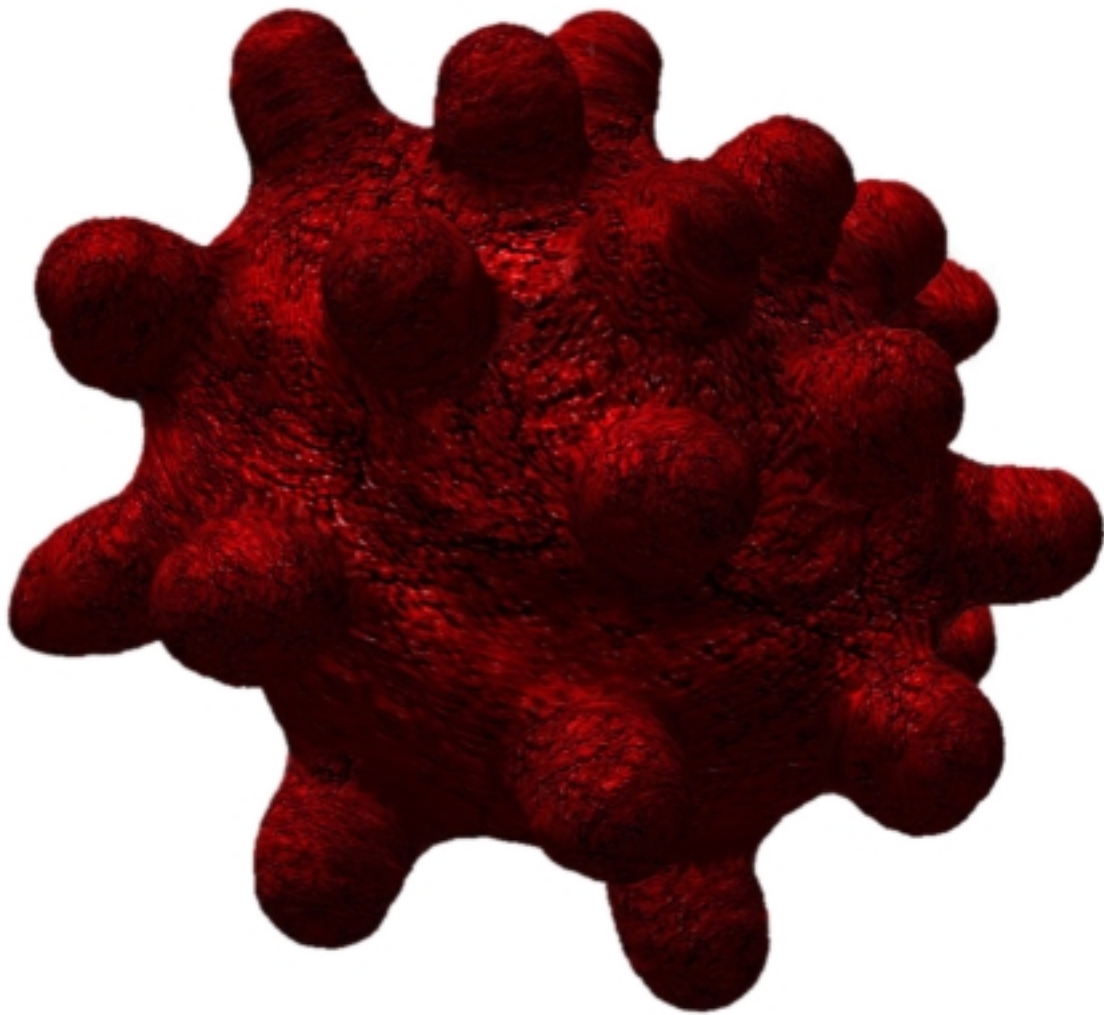


Abbildung 96: Visualisierung eines Echinozyten

Beispiel einer Echinozytenvisualisierung. Die Grundform wurde mit Hilfe des Quaoaring Design Patterns „Main“ erzeugt, auf den verschiedene Quaoaring Design Pattern vom Typ „Squeeze“ appliziert wurden. Die Spiculae wurden mit „Bulges“ realisiert, die nanostrukturelle Feinstruktur mit verschiedenen „Skins“, die auf Texturen zurückgreifen, welche gemäß rasterkraftmikroskopischen AFM-Daten erzeugt wurden. Die Auflösungsverzerrungen, die an den Punkten entstehen, an denen die Zentrallinien das Skin Mesh durchstoßen, wurden im Postprocessing an den betreffenden Stellen retuschiert.

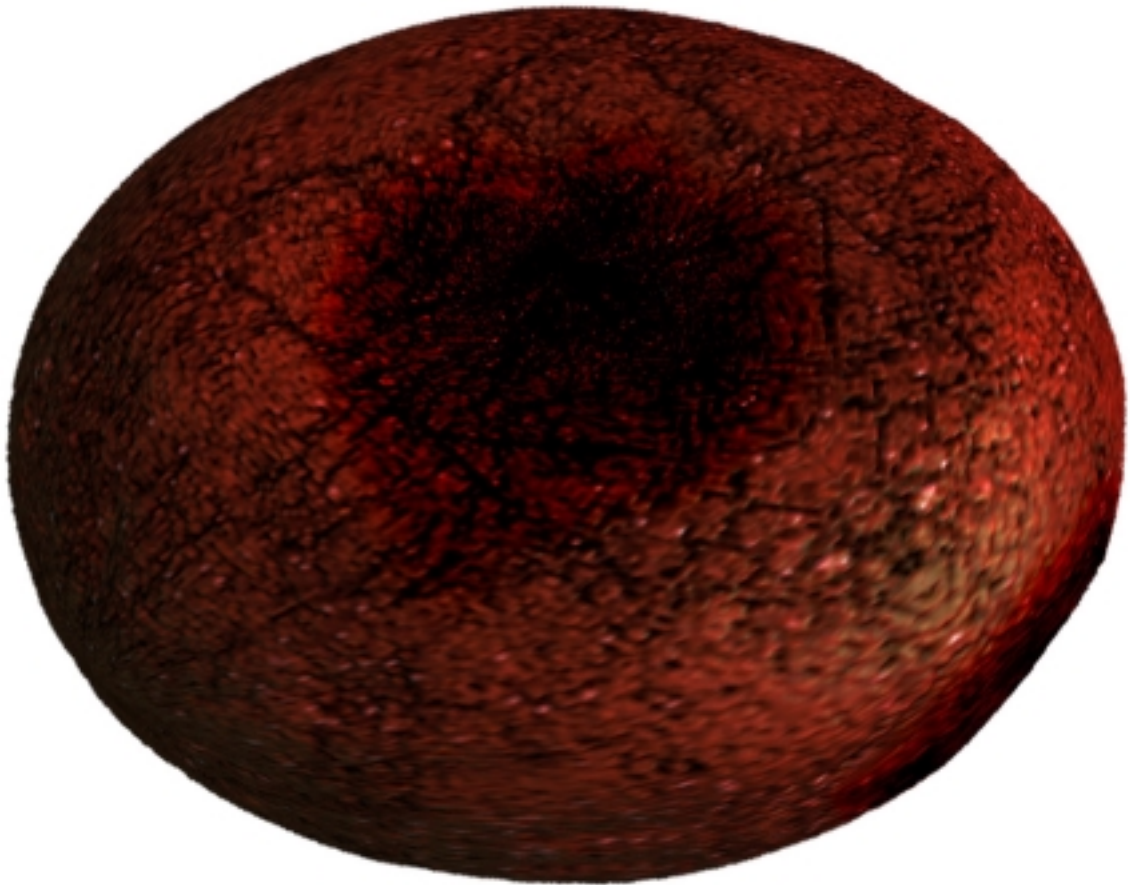


Abbildung 97: Visualisierung eines Discozyten

Beispiel einer Visualisierung eines discozytären Erythrozyten. Die Grundform ist mit Hilfe des Quaoaring Design Patterns „Main“ erzeugt, auf das an den Antipoden zwei großflächige „Squeezes“ appliziert sind, um die typische „Donut“-Grundform des Diskozyten zu erzeugen. Die nanostrukturelle Feinstruktur ist mit einem „Skin“ modelliert, welche auf eine große Textur zurückgreift, die wiederum gemäß rasterkraftmikroskopischer AFM-Daten erzeugt ist. Die Auflösungsverzerrungen, welche an den Antipoden entstehen, an denen die Zentrallinien das Skin Mesh durchstoßen, wurden im Postprocessing retuschiert.

6.9 Bewertung

Die sieben exemplarischen auf Quaoaring basierenden Projekte beweisen die Praxistauglichkeit der neuen Technologie hinsichtlich aller in Kapitel 6.1.1 gestellten Anforderungskriterien für den Einsatz in der Praxis, mit einer einzigen Ausnahme. Die Ausnahme betrifft den Aspekt der Komplexitätsbewältigung. Die Schwierigkeiten traten im Projekt e-Munkulus bei der Echtzeitvisualisierung extrem filigraner Gebilde mit über 100.000 Quaoaring Design Patterns und über 1,2 Millionen Polygonen auf (siehe Kapitel 6.7), dagegen stellt die Echtzeitvisualisierung von höheren Organismen (wie z.B. Saurier, Enteropneusten etc.) mit ~250 Quaoaring Design Patterns und ~30.000 Polygonen kein Problem dar. Hier besteht Handlungsbedarf für weitere Zusatzentwicklungen. Mögliche Lösungswege zur Behebung dieses Defizites werden in Kapitel 7.1 beschrieben.

Zusätzlich zur Erfüllung der Anforderungskriterien wurden allgemeine Erfahrungswerte gesammelt. So zeigen vor allem die Projekte e-Voluzzer, e-VoLuzie und e-VoLutz, dass informationstechnische Laien mit geeigneten User Interfaces das Quaoaring Software Framework sofort intuitiv bedienen können, und nach mehrstündigen Schulungsmaßnahmen sogar in der Lage sind, eigene Organismen zu modellieren.

Im Großen und Ganzen lässt sich konstatieren, dass das Interesse und die hohe Akzeptanz seitens von Biologen, Paläontologen und Medizinern an dem neuen Verfahren unerwartet hoch ist, was zeigt, dass biologisch sinnfällige 3D-Visualisierungstools für Organismen bei Biologen und Paläontologen anscheinend gesuchte Mangelware sind.

7 Ausblick und Abschluss

*Alles Vortreffliche bereichert die Zukunft,
ist aber auch ein Raub an ihr,
denn nie ist ein gleiches möglich.*

Christian Friedrich Hebbel (1851)

7.1 Defizite und Erweiterungsmöglichkeiten des Quaoarings

Quaoaring ist in seiner derzeitigen Form durchaus ein in sich abgeschlossenes und praxistaugliches Verfahren. Es hat seine besondere Stärken wie beispielsweise Echtzeitfähigkeit und damit Interaktivität, einen einfachen und intuitiven Modellierungs- und Animationsprozess, einen recht hohen Realismus der Resultate durch ein biomechanisch stringentes Verhalten der Organismen und die Fähigkeit, evolutionäre Übergänge auf einfache Weise biologisch stringent zu visualisieren. Diese Technologie hat allerdings auch einige Schwächen, die hier nicht verschwiegen werden sollen. In Zusammenhang mit den Defiziten existieren verschiedene Erweiterungs- und Verbesserungsmöglichkeiten für die Zukunft. Diese betreffen vornehmlich die Kollisionserkennung, die Analysefähigkeit, zusätzliche Quaoaring Design Patterns und den Import von Volumendaten. Im Folgenden wird auf diese Erweiterungsmöglichkeiten näher eingegangen.

7.1.1 Kollisionserkennung

Die aktuelle Version des Quaoarings beinhaltet keine eigene Kollisionserkennung. Für alle Organismen, die im Steuerungsmodus animiert werden, ist dies kein Problem, da der Akteur bzw. die Armatur sowieso die reellen Objektgrenzen beachten muss. Ist der Steuerungsmodus dagegen ausgeschaltet, kann der Organismus bislang keine Objektgrenzen detektieren und bewegt sich durch sie hindurch. Daher wurden bislang entweder nur Organismen animiert, die durch das Wasser schwimmen oder durch die Luft fliegen. Für auf dem Land lebende Organismen müsste daher für zukünftige Implementierungen eine Kollisionserkennung mit einer geeigneten Collision Response integriert werden.

Interessant wäre in diesem Zusammenhang eine Kollisionserkennung, die eng mit der Quaoaringtechnologie verwoben ist. So könnte die Auswahl der Polygone, die in den Kollisionserkennungsprozess mit einbezogen werden, anhand von Bioflächen erfolgen. Auf diese Weise würden biologienah diejenigen Körperstellen vorselektiert, die mit einer nennenswerten Wahrscheinlichkeit mit Umgebungsobjekten kollidieren. Beim Menschen könnte eine solche Selektion beispielsweise die Füße, die Knie, das Gesäß, den Rücken, die Nase, den Schädelbereich, die Ellenbogen und die Hände beinhalten. Dagegen dürfte es im Normalfall nicht notwendig sein, die Arm- und Kniebeugen, die Hüfte sowie die inneren Strukturen wie die inneren Organe, die Knochen, die Sehnen, die Muskeln und die Bänder in den Kollisionserkennungsprozess zu integrieren. Auf diese Art und Weise würde die Kollisionserkennung wesentlich beschleunigt.

7.1.2 Kombination mit Feder-Masse-Dämpfer-Systemen

Feder-Masse-Dämpfer-Systeme und Quaoaring ergänzen sich komplementär. Feder-Masse-Dämpfer-Systeme können analytisch genutzt werden, erlauben allerdings nicht die Berechnung komplexer Organismen mit zyklischen Interdependenzen. Quaoaring ist dagegen vorwiegend ein Visualisierungstool, welches mit seiner Performanz auch die Visualisierung komplexer Tiere in Echtzeit erlaubt und sich robust gegenüber zyklischen Interdependenzen verhält. Feder-Masse-Dämpfer-Systeme erlauben die Visualisierung von Gewebeverzerrungen wie sie z.B. bei der Mimik auftreten. Quaoaring spielt dagegen seine Stärken vorwiegend im Rumpfbereich aus, wo eine realistische Deformationsdynamik erzielt wird, hat aber seine Schwächen bei der Visualisierung einer realistisch anmutenden Mimik.

Eine synergistische Kombination der Performanz und der Robustheit des Quaoarings mit der Analytik und Genauigkeit der Feder-Masse-Dämpfer-Systemen könnte daher zu völlig neuen Anwendungen führen und würde insbesondere die Simulation und Visualisierung von höheren Wirbeltieren, insbesondere des Menschen, verbessern.

Ein solches *FMD-Quaoaring* würde das Anwendungsspektrum des Quaoarings stark erweitern.

7.1.3 Vesikel als weiteres Quaoaring Design Pattern

Das Wachstum biologischer Pneus unterscheidet sich von der Formgestaltung technischer Pneus vor allem darin, dass Vesikelbildung möglich ist. Diese Vesikel können sich autonom weiterentwickeln. Vesikel können zu Blasen heranwachsen, sie können durch den Körper bzw. das inter- und extrazelluläre Medium wandern. Darüber hinaus ist in der Natur rekursive Vesikelbildung möglich, d.h. aus größeren Vesikeln können sich wiederum Untervesikel abspalten.

Im gegenwärtigen Quaoaringkonzept wird dieser Tatsache nur bedingt Rechnung getragen. Es ist zwar Vesikelbildung möglich. Die Vesikel können aber nur noch bedingt durch zusätzliche morphische Felder beeinflusst werden. Dies hat drei Gründe:

Der erste Grund ist der, dass die Adressierung von Flächen auf dem Vesikel über den Pneu erfolgt, aus dem sich der Vesikel abgespalten hat, und somit nicht mehr intuitiv einsichtig ist. Somit wird die Modifikation eines Vesikels für den Modellierer zu einem mühsamen Trial-and-Error-Verfahren.

Der zweite weitaus gewichtigere Punkt ist derjenige, dass in der Praxis die Mesh-Granularität bei der Vesikelbildung nicht für eine rekursive Vesikelbildung ausreicht.

Drittens und letztens ist Vesikelbildung bislang nur indirekt über Reliefs und Squeezes möglich. Größere Vesikelabspaltungen und Wanderungen derselben sind daher nicht möglich. Insbesondere der komplexe Einfaltungsmechanismus, der in der Embryonalentwicklung eine große Rolle spielt, ist damit nicht modellierbar.

Abhilfe würde hier ein weiteres Quaoaring Design Pattern schaffen, welches ein funktionelles Morphing zwischen leichten Einfaltungen und einem autonom agierenden Organ schafft.

7.1.4 Level-of-Detail des Skin-Meshs

Das Anwendungsbeispiel aus Kapitel 6.7 offenbarte, dass ein Schwachpunkt des Quaoarings darin liegt, extrem komplexe Strukturen mit vielen filigranen Strukturen in Echtzeit zu visualisieren. Hier könnten algorithmisch automatisierte Level-of-Detail-Techniken Abhilfe schaffen, die speziell auf das Konzept des Quaoarings abgestimmt sind. Eine diesbezügliche Lösung wäre es, in Abhängigkeit von der Kameradistanz bei der Mesh-Generation einzelne Reihen von Vertices einfach zu überspringen. Die Polygone des Skin-Meshs müssten dementsprechend weitläufiger aufspannt werden.

Hierzu existieren verschiedene Möglichkeiten, welche sich hauptsächlich darin unterscheiden, welche Reihen von Vertices übersprungen werden. So wären z.B. das Überspringen von longitudinalen Spalten (entlang der Zentrallinie), von radialen Reihen (um die Zentrallinie herum) sowie Kombinationen der beiden Varianten möglich und sinnvoll. Welche Varianten in der Praxis die beste Beschleunigung bei einer gleichzeitigen optischen Minimierung der Artefakte bei den Übergängen zwischen zwei adjazenten LOD-Stufen erzielen, müssen zukünftige Forschungen ergeben.

Eine algorithmisch einfache und somit pragmatische Lösung wäre es, bei der Vertex-Auflösung des Skin-Meshs sowohl longitudinal als auch radial jeweils nur noch eine Anzahl zuzulassen, die Zweierpotenzen entspricht, um dann longitudinal nur noch jeden x .Vertex mit $x=k*2^{lod}$, $k=1 \dots \frac{m}{2^{lod}} - 1$ und radial nur noch jeden y .Vertex mit $y=j*2^{lod}$, $j=1 \dots \frac{n}{2^{lod}}$ zu beachten, wobei m und n jeweils die longitudinale bzw. radiale Grundauflösung des Skin-Meshs und lod die Level-of-Detail-Stufe darstellt. Bei der aufsteigenden Aufspannungsvariante (siehe Kapitel 4.4.1) würden die Polygone dann folgendermaßen aufgespannt:

$$\text{F46} \quad D_{k,j}^{A_auf} = (\vec{v}_{k \cdot 2^{lod}, j \cdot 2^{lod}}, \vec{v}_{(k+1) \cdot 2^{lod}, j \cdot 2^{lod}}, \vec{v}_{k \cdot 2^{lod}, \text{mod}(j+1, n) \cdot 2^{lod}}), k=1 \dots \frac{m}{2^{lod}} - 1, j=1 \dots \frac{n}{2^{lod}}$$

$$\text{F47} \quad D_{k,j}^{B_auf} = (\vec{v}_{(k+1) \cdot 2^{lod}, \text{mod}(j+1, n) \cdot 2^{lod}}, \vec{v}_{k \cdot 2^{lod}, \text{mod}(j+1, n) \cdot 2^{lod}}, \vec{v}_{(k+1) \cdot 2^{lod}, j \cdot 2^{lod}}), k=1 \dots \frac{m}{2^{lod}} - 1, j=1 \dots \frac{n}{2^{lod}}$$

Die Bestimmung der Vertices sowie der Aufbau des Meshs würde natürlich total ignoriert werden, wenn entweder in radialer oder in longitudinaler Richtung weniger als zwei Vertices existieren, daher gelten die Formeln F46 sowie F47 nur unter folgender Bedingung:

$$\frac{m}{2^{lod}} \geq 2 \wedge \frac{n}{2^{lod}} \geq 2$$

7.1.5 Level-of-Detail der Quaoaring Design Patterns

Statt der im vorigen Kapitel 7.1.4 besprochenen Level-of-Detail-Techniken, welche die Erstellung des Skin-Meshs betreffen, könnten auch Level-of-Detail-Techniken eingeführt werden, welche sich auf die Quaoaring Design Patterns beziehen:

Die Berechnung der Deformationsobjekte – also derjenigen Quaoaring Design Patterns, welche auf morphischen Feldern beruhen – könnte beispielsweise übersprungen werden, wenn der Wert des Auslenkungsparameters ω einen auf jede LOD-Stufe abgestimmten Grenzwert unterschreitet. Damit die Übergänge zwischen den LOD-Stufen nicht zu harsch erscheinen, würde es Sinn machen, den Auslenkungsparameter in einem bezüglich der Kameradistanz festgelegten Übergangsbereich vor dem Ausblenden graduell auf Null zu senken.

Analog wie bei diesen Level-of-Detail-Techniken für Deformationsobjekte, könnten auch LODs für Pneuobjekte eingeführt werden, wobei sich – statt des Auslenkungsparameters ω – das Volumen des jeweiligen Pneus zur Auslösung der Berechnung anbieten würde.

Diese LODs der Quaoaring Design Patterns können durchaus auch parallel zu den LODs des Skin-Meshs ausgeführt werden, um so eine hohe Beschleunigung des Quaoarings bei komplexen Strukturen zu erreichen.

7.1.6 Import von Volumendaten

Vielfach liegen anatomische Daten als Volumendaten vor, vor allem, wenn sie mit Hilfe von Magnetresonanz-Verfahren, 3D-Ultraschall etc. gewonnen wurden. Zurzeit sind diese Daten nicht für das Quaoaring zugänglich. Eine Einbeziehung von Volumendaten hätte folgende praktische Anwendungsbereiche:

- Formerfassung realer anatomischer Strukturen
- Vergleich zwischen realen und modellierten Strukturen
- Verbesserte Visualisierung der Gewebegrenzen.
- Datenkomprimierung

Daher wäre die Entwicklung von Übersetzungsverfahren, die Volumendaten in Quaoaring-Skripte umwandeln, sinnvoll und lohnenswert.

Die intuitivste Art der Umwandlung von Volumendaten in Quaoaring-Skripte ist das „Hineinwachsen“ von Pneus in die gewünschten Strukturen, so dass sich die Pnehüllen langsam an geeignete Isoflächen „heranschmiegen“. Dieses Verfahren hat Ähnlichkeit mit dem Aufwachsen von Bonsai-Bäumchen.

Dazu wird ein unendlich kleiner und somit punktförmiger Ausgangspneu in die Mitte eines Gewebefeldes eingesetzt. Das Volumen des Pneus würde dann schrittweise erhöht, bis seine Oberfläche an eine Gewebegrenze stößt. An der Schnittstelle wird dann ein morphisches Feld mit wachsender negativer Feldstärke appliziert, bis der Pneu sich nicht mehr mit der Isofläche schneidet. Der Pneu wird durch dieses negative morphische Feld nach innen gewölbt, und weicht somit der Gewebegrenze aus. Das Pneuwachstum und die Applikation inverser morphischer Felder werden so lange wiederholt, bis alle Regionen mit der Gewebegrenze kollidiert sind. Der resultierende Pneu wird dann der Gestalt des Gewebes ähneln.

Dieses Verfahren erscheint theoretisch einfach, in der Praxis dürften jedoch viele Probleme auftreten, die ein weites Feld für zukünftige Forschungen und Entwicklungen offenlassen. Die notwendigen Arbeiten betreffen dabei vor allem folgende Problembereiche:

- Es müssen geeignete Thresholds für verschiedene Gewebegrenzen gefunden werden. Die Grenzwerte müssen dabei variabel an die verwendeten diagnostischen Verfahren justiert werden. Die Justierungsarbeiten werden sich voraussichtlich sehr zeitaufwändig gestalten, da sie in der Hauptsache empirische Trial-and-Error-Verfahren beinhalten.
- Messfehler und andere Artefakte des Volumendatensatzes könnten Gewebegrenzen vortäuschen, wo tatsächlich keine existieren oder andersherum Gewebegrenzen kaschieren oder nicht abbilden. Diese fehlerhaften Daten müssen in einer sinnvollen Weise umgangen oder interpolierend ersetzt werden.
- Die Positionen und Ausrichtungen der Ausgangspneus sind essentiell für eine optimale Adaption der Pneus an die tatsächlichen Gewebegrenzen. Die automatische Ermittlung dieser Startwerte anhand der Volumendaten ist aber nicht trivial.

- Die Typenauswahl der Quaoaring Design Patterns ist nicht immer eindeutig. Hier müssen geeignete Entscheidungskriterien gefunden werden. Das Gleiche gilt für die Ausbildung der Objekthierarchien. Insbesondere bei verzweigten Strukturen, wie z.B. bei Kapillaren, muss ein Pneu in eine komplexe Hierarchie von Bulges und Canals übersetzt werden. Die damit zusammenhängende Hierarchiebildung kann in manchen Fällen multidirektional sein.

Die Entwicklung eines Übersetzungsverfahrens von Volumendaten in Quaoaring-Skripte erscheint daher sehr zeitaufwändig. Sie ist jedoch grundsätzlich lösbar.

7.2 Mögliche Einsatzgebiete des Quaoarings

Innovationen in der Informatik führen stets auch zu neuen Anwendungsmöglichkeiten, die zum Zeitpunkt der Entwicklung unvorhersehbar sind. Daher ist es schwierig, Voraussagen zu treffen, welche potentiellen Einsatzgebiete für die Quaoaringtechnologie in Frage kommen. Diese Tatsache zeigte sich konkret auch an einer unerwarteten Anfrage aus der Industrie, Quaoaring für einen auf Technomusik tanzenden Drachen für Diskothekenprojektionen zu verwenden. Trotz dieser Schwierigkeiten ist es durchaus berechtigt, Anwendungsvorschläge zu geben. Dies soll im Folgenden versucht werden:

7.2.1 Einsatzgebiete in Edutainment

Quaoaring wurde bisher am meisten im Edutainmentbereich in die Praxis umgesetzt. So wurden mit Hilfe der Quaoaringtechnologie schon diverse interaktive Museumsexponate verwirklicht, beispielsweise die Saurier e-VoLuzie und e-VoLutz für das Darmstädter Cybernarium (siehe Kapitel 6.4 und 6.5), ein kindgerechtes Dinosaurierbau-Kit für den Einsatz in Kindergärten, eine interaktive Enteropneusten-Evolutionslinie und weitere Exponate für das Frankfurter Senckenbergmuseum (siehe auch Kapitel 9.2.5 im Anhang)

Geplant ist des Weiteren die Verwirklichung eines ganzen Urmeeres mit Schwärmen verschiedener urzeitlicher Tiere. Museumsbesucher werden dann individuell mit virtuellen Zeitreise-U-Booten in das Meer eintauchen und interaktiv die urzeitliche Fauna erforschen. Die Interaktivität kann sowohl räumlich als auch zeitlich erfolgen, d.h. der Museumsbesucher kann an jeden beliebigen Ort des Urmeeres navigieren, er kann aber

auch mit der virtuellen Zeitmaschine in der Zeit vor- und zurückfahren, um die Veränderung der Fauna zwischen Kambrium, Ordovizium, Silur, Devon, Karbon, Perm, Trias und Jura zu erforschen.

Quoaring spielt seine Vorteile dabei immer dann aus, wenn es um biologische Stringenz und Interaktivität geht, speziell um interaktive Wachstums- oder Mutationsprozesse. Alle Exponate, Messepräsentationen oder Lernspiele, die diese Eigenschaften erfordern, werden daher besonders vom Quoaring profitieren.

7.2.2 Einsatzgebiete im Entertainmentbereich

Quoaring kann darüber hinaus im gesamten Entertainmentbereich wirkungsvoll eingesetzt werden. Mit Quoaring werden die Erstellungskosten von überzeugenden Tieranimationen auf einen Bruchteil der heutigen Kosten sinken. Vor allem Film- und Fernsehproduktionen aus dem Mid- und Low-Budget-Bereich werden davon profitieren, da sie mit ihren Special Effects bei Organismen mit den Hollywoodproduktionen gleichziehen können. Aber auch Independent- und Amateurfilme werden virtuelle Organismen einsetzen können, was zu neuen Genres in der Amateurfilmszene führen könnte.

Was für die Filmindustrie gilt, hat auch für die Computerspiele-Industrie Gültigkeit. Mit Quoaring werden interaktive virtuelle Tiere eine bessere Dynamik, einen höheren Detailgrad und eine überzeugendere Anmutung bekommen als bisher. Zudem wird ihre Modellierung und Animation vereinfacht und somit verbilligt.

Damit können herkömmliche Spiele verbessert werden wie z.B. Ego-Shooter gegen sich dynamisch bewegende Monster oder Adventure Games, in denen Organismen vorkommen. Es werden aber auch komplett neue Formen von Computerspielen ermöglicht, z.B. interaktive AR-Kampfturniere gegen realistische Wesen, Evolutionsspiele zur Kreation einer eigenen Fauna und Flora, Schneckenwettrennen, Elwetrutsche-Suchen im Pfälzer Wald, virtuelles Perlentauchen im Haifischbecken, Godzilla-Wolkenkratzer-Wettklettern, Einhornreiten in den schottischen Highlands, Hundewettzuchten, Klapperschlangensammeln in Arizona, selbstgebastelte Tamagotchis oder Dinosaurierausweichmanöver bei Meteoriteneinschlägen. Der Kreativität sind hier kaum Grenzen gesetzt.

7.2.3 Einsatzgebiete im e-Learning-Bereich

Quaoaring kann für interaktive Lernmodule in allen Fachbereichen eingesetzt werden, in denen Organismen als Ganzes oder anatomische Teile derselben eine Rolle spielen, respektive Paläontologie, Anatomie, medizinische Chirurgie, Zoologie und Sportwissenschaften.

Quaoaring lädt dazu ein, interaktive Experimente mit den virtuellen Organismen durchzuführen, um so im wahrsten Sinne des Wortes „spielerisch“ zu lernen.

Die intuitive Modellierungstechnik ermöglicht dabei auch Biologen und Medizinern ohne dezidierte Ausbildung in Computergraphik die Erstellung von überzeugend wirkenden Lernmodulen mit virtuellen Organismen.

7.2.4 Einsatzgebiete in der Kommunikation

Quaoaring eignet sich hervorragend für die Erstellung von ausgefallenen 3D-Avataren, deren Bewegungen interaktiv auf den Gesprächspartner eingehen. Der Vorteil gegenüber herkömmlichen 3D-Avataren, die entweder starr oder bestenfalls mit Skinning-Methoden animiert werden, besteht darin, dass die Bewegungsabläufe der auf Quaoaring basierten Avatare eine höhere Dynamik aufweisen und biologisch sinnfälliger anmuten. Ein weiterer Vorteil dieser Avatare ist, dass der Benutzer seinen Avatar ohne Modellierungsausbildung schnell selbst nach seinen Wünschen modifizieren kann, so dass die Identifizierung mit ihm leichter wird. Auch könnte die Form des Avataren graduell je nach Stimmungslage wechseln. Wäre die Benutzerin bzw. der Benutzer zum Beispiel gerade wütend, könnte sich der Avatar sukzessive in einen Drachen verwandeln, oder bei Furcht in einen ängstlichen Hasen und bei Verliebtheit möglicherweise in ein Einhorn. Besonders interessant ist die Tatsache, dass die Repräsentationsformen für Stimmungen beim Quaoaring frei miteinander mischbar sind, d.h. wenn im obigen Beispiel eine Person verliebt ist und dabei gleichzeitig Verlustängste hat, würde sich die Form des Einhorns mit der Form des Angsthasen sinnfällig verbinden und automatisch ein Mischwesen zwischen Einhorn und Hasen erzeugen (siehe auch Kapitel 4.2.6). Die zusätzlichen multivarianten Dimensionen der Avatarengestalt bereichern die Kommunikationsmöglichkeiten durch wichtige zusätzliche Metainformationen.

Was für Avatare gilt, besitzt natürlich auch für virtuelle 3D-Roboter Gültigkeit. Auch sie können interaktiv und biologisch sinnfällig auf die Eingaben des Benutzers reagieren und ihre Form multivariant gemäß der gerade aktuellen Aufgabe ändern.

7.2.5 Einsatzgebiete in der Diagnostik

Weiterhin kann Quaoaring die Visualisierung bei diagnostischen Verfahren verbessern, insbesondere wenn die Voxeldaten in Pneu hierarchien übersetzt werden, um die Gewebegrenzen besser sichtbar zu machen und darüber hinaus besondere Artefakte (z.B. Tumorgewebe) optisch zu kennzeichnen.

Mit Quaoaring lassen sich insbesondere Wachstumsübergänge zwischen zwei Stadien gut visualisieren. Das Volumen von Organen, Organteilen und biologischen Strukturen wird im Quaoaring automatisch abgeschätzt und muss nicht dezidiert nachberechnet werden. Dies ist insbesondere in der geburtsbegleitenden Gynäkologie und der Krebsfrüherkennung wichtig.

7.2.6 Einsatzgebiete als Planungshilfe

Interaktive chirurgische Planungshilfen, die eine Quaoaringtechnologie einbeziehen, können Operationen optimieren. Insbesondere bei Anwendung von Pneu hierarchien, die an die Daten der Magnetresonanztomographie, an 3D-Ultraschall oder an andere nichtinvasive 3D-Diagnoseverfahren approximiert werden, können millimetergenaue interaktive Simulationen des chirurgischen Eingriffes im Vorfeld der Operation erfolgen.

Die Simulation ist keineswegs auf die Chirurgie beschränkt. Mit analogen Verfahren können Kraftsportler die optischen Auswirkungen von verschiedenen Trainingsmethoden in die Zukunft extrapolieren und so ihren Körper optimal ausdefinieren.

Durch die Simulation der einzelnen Muskelbeanspruchungen können Maßnahmen im Bereich der Rehabilitation, Orthopädie oder Gymnastik optimiert werden.

7.2.7 Einsatzgebiete in der Forschung

Ein weites Feld, welches durch Quaoaring bereichert werden kann, ist die virtuelle Paläontologie. Erste praktische Anwendungen sind mit e-Voluzzer schon demonstriert worden, indem dezidierte Evolutionzweige visualisiert wurden und so Übergangsformen zwischen zwei evolutionären Stadien optisch auf ihre Lebensfähigkeit überprüft wurden.

Allerdings ist Quaoaring in seiner bisherigen Form nur bedingt als Forschungswerkzeug tauglich, da die Technologie primär nicht für die Analytik, sondern für die Visualisierung konzipiert wurde. Quantitative Simulationen sind derzeit nicht möglich.

Ein analytisch erweitertes Quaoaring, welches anhand der Natur verifiziert und geeicht wurde und welches die Kräfte und Deformationen von speziellen Pnuteilen analytisch genau mit Feder-Masse-Modellen simuliert, könnte hier in Zukunft Abhilfe schaffen und Quaoaring für die medizinische, biologische und paläontologische Forschung tauglich machen.

So könnten beispielsweise mit einem solchen analytisch erweiterten Quaoaring gesamte Evolutionzweige nachsimuliert werden, um Hypothesen zur Entstehung der Arten zu verifizieren oder zu falsifizieren.

7.2.8 Einsatzgebiete in Design und Produktion

Eine fachfremde Anwendung des Quaoarings wäre möglicherweise, neue biologisch anmutende Formen für Werkzeuge, Fahrzeuge und Gebäude zu erzeugen. Dies könnte Industriedesign und Architektur um ein neues Werkzeug bereichern.

Auch bei der Entwicklung bionischer Maschinen kann Quaoaring in den Formerzeugungsprozess integriert werden.

7.2.9 Einsatzgebiete als Komprimierungshilfe

Medizinische 3D-Daten, die aus nichtinvasiven medizinischen Diagnoseverfahren gewonnen wurden, respektive Ultraschalldiagnostik, Nuklear- und Computertomographie, Magnetresonanztomographie und Angiographie liegen in der Regel als Volumendaten vor. Der Platzbedarf wächst dabei kubisch mit der Granularität der Einzeldimensionen. Die Speicherung und die Übermittlung großer Datenmengen stellt daher eine enorme und bisher ungelöste Herausforderung dar.

Das biologisch basierte Modell im Quaoaring erlaubt eine sehr einfache und komprimierte Abbildung biologischer Strukturen mit Hilfe von Hierarchien von Quaoaring Design Patterns, wodurch zur Lösung des Problems beigetragen werden könnte.

Eine Übersetzung von Voxeldaten in Quaoaring Design Patterns würde organische Strukturen datentechnisch sehr komprimieren. Medizinische 3D-Daten würden nur einen Bruchteil der bisherigen Größe benötigen. Diese Tatsache ist sowohl bei der Archivierung als auch beim Netztransfer von medizinischen Daten hilfreich.

7.3 Evaluation

Das Ziel dieser Arbeit war die Entwicklung eines neuen Softwareverfahrens zur Generierung von dreidimensionalen virtuellen Organismen, wobei der Fokus auf die Generierung der Fauna gelegt wurde.

In der vorliegenden Arbeit wurde dazu ein neues holistisches Gesamtkonzept und eine darauf aufbauende Softwareimplementation entwickelt, die völlig neue Wege beschreitet und die sich somit mit keinem bekannten Verfahren direkt vergleichen lässt. Aufgrund dieser mangelnden Vergleichsmöglichkeiten macht eine umfangreiche Evaluation mit zahlreichen Messungen, die zur Gegenüberstellung zu Messwerten aus ähnlichen traditionellen Verfahren dienen, an dieser Stelle keinen Sinn. Stattdessen wird im Folgenden die Quaoaring-Technologie an den in der Anforderungsanalyse (siehe Kapitel 3) gestellten 11 Kriterien gemessen:

1. *Erfüllung des Anforderungskriteriums „Holistischer Generierungsprozess“ (siehe Kapitel 3.1):* Die Quaoaringtechnologie erfüllt durch seine Interdependenz zwischen Modellierungs- und Animationsprozess sowie seiner parallel stattfindenden Echtzeitvisualisierung voll und ganz dieses gestellte Anforderungskriterium.
2. *Erfüllung des Anforderungskriteriums „Verwendung intuitiver Metaphern“ (siehe Kapitel 3.2):* Die Quaoaringtechnologie verwendet – wie gefordert – die Metapher von „hierarchisch kombinierbaren wassergefüllten Pneus“ und lehnt sich damit eng an die in der Anforderungsanalyse vorgeschlagene „Luftballonmännchen“-Metapher an.
3. *Erfüllung des Anforderungskriteriums „Abkehr vom Aufzeichnungscharakter“ (siehe Kapitel 3.3):* Auch wenn im Steuerungsmodus per Motion Capturing oder Stop Motion Input Devices aufgezeichnete Bewegungsdaten immer noch zur Animation verwendet werden können, so erfolgt der Modellierungs- und die Animationsprozess im Normalfall komplett am Computer. Somit ist die geforderte Abkehr vom Aufzeichnungscharakter gegeben.

4. *Erfüllung des Anforderungskriteriums „Verwendung evolutionärer Methoden“ (siehe Kapitel 3.4):* Die Quaoaringtechnologie erlaubt es, mit Hilfe seiner Organismenkonstruktionsskriptbefehle einmal erstellte Organismen zu verwenden, um sie entlang evolutionärer Linien in Richtung des gewünschten Zielorganismus zu mutieren. Damit wird die Anforderung zur Verwendung evolutionärer Methoden erfüllt.
5. *Erfüllung des Anforderungskriteriums „Reusability von anatomischen Strukturen“ (siehe Kapitel 3.5):* Durch die Verwendung des BKS adaptieren sich einmal modellierte und animierte anatomische Strukturen automatisch sinnfällig an den Zielorganismus. Somit ist die geforderte Reusability anatomischer Strukturen gegeben.
6. *Erfüllung des Anforderungskriteriums „Verwendung medizinischer Termini“ (siehe Kapitel 3.6):* Das Vokabular der Nomina Anatomica Veterinaria bzw. der Terminologia Anatomica können mit den in Kapitel 4.2.4 präsentierten Gleichungen direkt in Wertebereiche des BKS übersetzt werden und umgekehrt. Somit ist die geforderte Verwendung medizinischer Termini möglich.
7. *Erfüllung des Anforderungskriteriums „Integration biologischer Constraints“ (siehe Kapitel 3.7):* Das Modell der Quaoaringtechnologie basiert auf der Frankfurter Organismen- und Evolutionstheorie und verwendet deren biologische Constraints, wie vornehmlich hydropneumatische Regeln und die damit verbundene Volumenkonstanz. Damit ist die geforderte Integration biologischer Constraints gewährleistet.
8. *Teilweise Erfüllung des Anforderungskriteriums „Kanalisierte Modellierung und Animation“ (siehe Kapitel 3.8):* Durch die Verwendung des BKS und die Einbeziehung biologischer Constraints (siehe Anforderungskriterium 7) wird von vorneherein die Organismengenerierung in sinnvolle Bahnen gelenkt. Völlig absurde Kreationen (beispielsweise Organismen in Form einer Kaffeekanne) sind nicht bzw. nur unter erhöhtem Aufwand erzeugbar. Somit ist diese Anforderung zu einem großen Teil erfüllt.

9. *Erfüllung des Anforderungskriteriums „Echtzeitfähigkeit der Animation“ (siehe Kapitel 3.9):* Quaoaring verhält sich von seiner Komplexität linear zu der Anzahl der Vertices im Skin-Mesh. Gleichzeitig werden pro Vertexberechnung wenige Operationen gebraucht. Dies garantiert, dass die Anforderung an eine volle Echtzeitfähigkeit bis zu einem hohen Komplexitätsgrad auch in der Praxis gewährleistet ist. Beispielsweise läuft der interaktive Nothosaurus Mirabilis (siehe Kapitel 6.5) auf dem Referenz-PC (siehe Kapitel 5.2) mit einer Frame-Rate von 21,6 fps.

10. *Erfüllung des Anforderungskriteriums „Hohe Bandbreite der Abstraktionsstufen“ (siehe Kapitel 3.10):* Die in Kapitel 6 beleuchteten Anwendungsbeispiele zeigen, dass mit Quaoaring die geforderte Bandbreite der Abstraktionsstufen 2 bis 10 in der Holarchie erzeugt werden kann.

11. *Teilweise Erfüllung des Anforderungskriteriums „Hoher Grad des Realismus“ (siehe Kapitel 3.11):* Während mit Quaoaring generierte wirbellose Tiere dem Betrachter sehr plausibel erscheinen, ist dies bei höheren Wirbeltieren wie beispielsweise Reptilien, Säugetiere und Vögel nicht mehr der Fall. Die betreffenden Tiere sehen etwas zu „aufgeblasen“ aus. Dies liegt vermutlich unter anderem daran, dass möglicherweise bei Wirbeltieren außer hydropneumatischen Regeln auch noch weitere biologische Constraints eine große Rolle spielen. Daher ist diese Anforderung nur teilweise erfüllt.

Somit sind 9 von den 11 Kriterien, die in der Anforderungsanalyse gestellt wurden, vollkommen, die beiden anderen zumindest teilweise erfüllt.

Die nicht vollständige Erfüllung der Anforderungskriterien 8 und 11 ist nicht als besonders negativ zu werten, denn die verbleibenden Defizite können vermutlich durch zukünftige Zusatzentwicklungen der Quaoaringtechnologie, respektive die Einbeziehung weiterer biologischer Constraints, behoben werden. Doch selbst ohne diese Zusatzentwicklungen ist Quaoaring – wenn die Technik an den verbleibenden 9 vollständig erfüllten Kriterien gemessen wird – konventionellen Techniken zu Organismengenerierung weit voraus.

Es kann somit konstatiert werden, dass die in Kapitel 1.2 gesteckten Ziele im Großen und Ganzen erreicht wurden. Quaoaring erlaubt eine effiziente Modellierung und Animation von dreidimensionalen virtuellen Organismen und umgeht dabei viele Defizite der in der State-of-the-Art-Analyse (siehe Kapitel 2) behandelten Verfahren. Es erfüllt weitgehend die erarbeiteten 11 Anforderungskriterien und geht in Teilaspekten auch über die Zielsetzungen hinaus.

7.4 Zusammenfassung und Abschluss

In der vorliegenden Arbeit wurde aufbauend auf einer State-of-the-Art-Analyse (siehe Kapitel 1 und 2), die in eine Anforderungsanalyse mündet (siehe Kapitel 3), ein in sich abgeschlossenes, eigenes Konzept entwickelt (siehe Kapitel 4), welches bei der Visualisierung von Organismen vielseitig eingesetzt werden kann und einige Vorteile gegenüber den anderen beschriebenen Verfahren aufweist.

Die Vorteile des Quaoarings liegen vor allem in der Möglichkeit, Körperdeformationen, komplexe Körperbewegungen, Wachstumsvorgänge und evolutionäre Formveränderungen biologisch korrekt und überzeugend zu animieren. Die Vorteile liegen zudem in seinen intuitiven Beschreibungsmetaphern, welche die Modellierung vereinfachen sowie in seiner hohen Performanz, welche es für interaktive Echtzeitanwendungen geeignet macht.

Das Konzept wurde über eine reine prototypische Implementation hinaus mit einem umfangreichen Software-Framework realisiert (siehe Kapitel 5). Dieses Framework wurde für mehrere Projekte verwendet. Aus diesem Projektspektrum wurden einige Anwendungsbeispiele ausgewählt und vorgestellt, welche die Vielseitigkeit, Softwarestabilität und Praxistauglichkeit der Quaoaringtechnologie demonstrieren (siehe Kapitel 6).

Zu guter Letzt wurden Anregungen gegeben, wie sich das Quaoaring in Zukunft weiterentwickeln könnte, um das dieser Technologie inhärente Potential zu beleuchten (siehe dieses Kapitel 7). Es ist in diesem Zusammenhang zu hoffen, dass die Quaoaringtechnologie einen wesentlichen Beitrag zur zukünftigen Entwicklung der Computergraphik leisten wird.

8 Literaturverzeichnis

- [3DDC 2004] 3D Digital Cooperation: *Capturing Concepts in 3D*
<http://www.3ddigitalcorp.com/home.htm>
Zugriff: 23.08.2004
- [3Dlabs 2004] 3Dlabs: *The wildcat technology* (2004)
<http://www.3dlabs.com/product/technology/mooresla.htm>
Zugriff: 01.04.2004
- [3Dscanners 2004] 3Dscanners Inc.: *3D-Scanners*
<http://www.3dscanners.com/>
Zugriff: 23.08.2004
- [3Dshape 2004] 3D Shape GmbH: *3D Scanning*
http://www.3d-shape.com/home/home_d.php
Zugriff: 23.08.2004
- [ABW 2003a] ABW GmbH: *3D-Scanning Projektoren* [a] *Messverfahren* [b]
[ABW 2003b] http://www.abw-3d.de/produkte/projektoren/projektoren_de.php
http://www.abw-3d.de/messverfahren/messverfahren_de.php
Zugriff: 24.08.2004
- [Acclaim 2002] Motion Capture: *Acclaim's Optical system*
http://www.css.tayloru.edu/instrmat/graphics/hypgraph/animation/motion_capture/motion_optical.htm
Zugriff: 23.5.2002
- [Alexa et al. 2001] Alexa, Marc; Behr, Johannes; Cohen-Or., Daniel; Fleishman Shachar; Levin, Daniel; Silva, Claudio T.: *Point Set Surfaces*; IEEE Visualization; S.21-28 (2001)

- [Alias 2003]** Alias Wavefront Inc.: *Maya* (2003)
<http://www.aliaswavefront.com>
Zugriff: 01.11.2003
- [Amak 2004]** AMAK : *Conseil MoCap*
http://www.amak.fr/conseil/mocap/mocap_corporelle.html
Zugriff : 01.01.2004
- [Archeoptix 2003]** Archeoptix Ltd.: *3D Laser Scanning*; (2003)
<http://www.stonehengelaserscan.org/laser.html>
Zugriff: 26.08.2004
- [Arius3D 2004]** Arius 3D: *3D Scanning*
<http://www.arius3d.com/index.html>
Zugriff: 17.08.2004
- [Armaverse 2004]** Armaverse Armatures: *Armaverse Armatures for Stop-Motion Animation and More* (2004)
<http://www.armaverse.com>
Zugriff 12.08.2004
- [Ascension 2004]** Ascension: *Magnetic Motion Capturing Technologies*
<http://www.ascension-tech.com>
Zugriff: 24.06.2004
- [Austrian 1982]** Austrian, Geoffrey: *Herman Hollerith: Forgotten Giant of Information Processing*. Columbia University Press, New York.
(1982)
- [Avid 2004]** Avid Technology Inc.: *Softimage/3D* (2004)
<http://www.softimage.com>
Zugriff: 06.12.2004

- [Babisch et al. 2001]** Babisch, Jürgen; Seidel, Egbert Johannes; Conradi, Steffen: *3D-Ultraschall-Ganganalyse ZEBRIS bei Dysplasiekoxarthrose vor und nach der Hüft-TEP-Implantation – eine Pilotstudie*; Phys Med Rehab Kuror 2001, 11, Heft 4: S.139 (2001)
- [Ball 2002]** Ball, Ryan: *The Real Mayor Of Halloween Town Henry Selick on A Holiday Classic And What's Growing In the Pumpkin Patch*; Animation Magazine, 11/2002, 24 (28.10.2002)
Auch Online:
<http://www.stopmotionanimation.com/>
Zugriff 12.08.2004
- [Balloonhq 2004]** Balloon HQ: *Educating the World About Balloons – Muscle Picture*;
<http://www.balloonhq.com/photos/multipics/muscle.jpg>
Zugriff: 05.08.2004
- [Baranoski & Rokne 2001]** Baranoski, Gladimir V. G.; Rokne, Jon: *An efficient and controllable blob function*; In: Journal of Graphic Tools 2001; Vol.6, Issue 4; S.41–54. (2001)
- [Barnard 1989]** Barnard, Stephen T.: *Stochastic Stereo Matching over Scale*; In: International Journal of Computer Vision 3; S.17-32 (1989)
- [Barr 1984]** Barr, Alan H.: *Global and Local Deformations of Solid Primitives*; In: Computer Graphics – Proceedings of the ACM SIGGRAPH 1984; Vol. 18, Issue 3; S.21-30 (1984)
- [BBC 2003]** Shields, Maggie: *Moore predicts more advances*; BBC-News (11.02.2003)
<http://news.bbc.co.uk/2/hi/technology/2748281.stm>
Zugriff: 28.07.2004

- [BBC 2004a] BBC: *Science and Nature – Prehistoric Life – Dinosaurs – TV Series – Production* [a]
- [BBC 2004b] *Science and Nature – Prehistoric Life – Dinosaurs – TV Series – Computer Graphics* [b]
- [BBC 2004c] *Science and Nature – Prehistoric Life – Dinosaurs – TV Series – Animatronics* [c] (2004)
http://www.bbc.co.uk/dinosaurs/tv_series/production.shtml [a]
http://www.bbc.co.uk/dinosaurs/tv_series/graphics.shtml [b]
http://www.bbc.co.uk/dinosaurs/tv_series/animatronics.shtml [c]
 Zugriff 15.08.2004
- [Becker et al. 1999] Becker, Margaret; Golay, Pascal; McNeel, Robert: *Rhino NURBS 3D Modelling*; McMilan Computer Publishing, Indianapolis (1999)
- [Bennett 1995] Bennett, Rod: *Jurassic Park and the Death of Stop-Motion Animation*; In: *Wonder magazine*, Nr. 9/1995, S.12 (1995)
 Auch Online:
<http://www.cornerstonemag.com/imaginarium/features/jurassic.html>
 Zugriff 12.08.2004
- [Bevilacqua 2003] Bevilacqua, Frédéric ; Dobrian, Christopher ; Ridenour, Jeff : *Mapping sound to human gesture: demos from video-based motion capture systems*; In : *Proceedings of the New Interfaces for Musical Expression 2003 Conference*; Montréal, Quebec, Canada (2003)
 Auch Online unter:
www.humanities.uci.edu/visualstudies/everyday/papers/Ridenour.pdf
 Zugriff: 04.08.2004

- [BGFE 2002]** Berufsgenossenschaft der Feinmechanik und Elektrotechnik
(BGFE): *Neue Laserklassen* (2002)
<http://www.bgfe.de/pages/thema/archiv/april02.htm>
Zugriff: 26.08.2004
- [Blender 2004]** Stichting Blender Foundation: *Blender Dokumentation*
Volume I – User Guide: (08.07.2004)
Auch Online unter:
<http://www.blender.org/modules/documentation/html/x6880.html>
Zugriff: 11.08.2004
- [Blinn 1978]** Blinn, James: *Simulation of Wrinkled Surfaces*; In: Computer Graphics – Proceedings of the ACM SIGGRAPH '78; Vol.12, Nr.3, S.286-292 (1978)
- [Blinn 1982]** Blinn, James F.: *A Generalization of Algebraic Surface Drawing*; In: ACM Transactions on Graphics; Vol. 1, Issue 3; Juli 1982; S.235-256 (1982)
- [Blitzbase 2004]** Blitzbase: *Bump Mapping*
<http://www.blitzbase.de/lexikon/bumpmapping.htm>
Zugriff: 14.12.2004
- [Bloomenthal 1988]** Bloomenthal, Jules: *Polygonization of Implicit Surfaces*; In: Computer Aided Geometric Design Vol.5; 4.Nov.1988 (1988)

- [Blumenbach et al. 2001]** Blumenbach, T.; Henke, T. (Projektleiter); Augath, W.; Heckl H.: *Satellitenavigation in der Sportwissenschaft*; Bundesinstitut für Sportwissenschaft (BISP)-Jahrbücher 2001; Bonn; S.25-35 (2001)
auch Online unter:
http://www.bisp.de/Produkte/Publikationen/BISp_Jahrbuecher/2001_pdf/13BluHenk.pdf
Zugriff: 06.08.2004
- [Böhler & Marbs 2004]** Böhler, Wolfgang; Marbs, Andreas: *Vergleichende Untersuchung zur Genauigkeit und Auflösung verschiedener Scanner*; In: Luhmann, Th., Photogrammetrie, Laserscanning, Optische 3D-Messtechnik. Beiträge der Oldenburger 3D-Tage 2004; Wichmann Verlag Heidelberg, S; 82-89 (2004)
Auch Online unter:
http://www.i3mainz.fh-mainz.de/publicat/oldenbg3d-04/scanner_vergleich.pdf
Zugriff: 17.08.2004
- [Bonik 1977]** Bonik, Klaus: *Quantitative Aspekte hydraulischer Systeme in Metazoen-Konstruktionen. I. Die Statik von Hydroskelett-Konstruktionen*; Courier Forschungsinstitut Senckenberg; Vol. 23: S.1-79; Frankfurt (1977)
- [Bonik 1978]** Bonik, Klaus: *Quantitative Aspekte hydraulischer Systeme in Metazoen-Konstruktionen. II. Die Evolution des Chorda-Myomeren Apparates*; Courier Forschungsinstitut Senckenberg; Vol. 32; S. 1-57; Frankfurt (1978)
- [Bouget & Perona 2000]** Bouguet, Jean-Yves; Perona, Pietro: *3D photography on your desk* (2000)
<http://www.vision.caltech.edu/bouguetj/ICCV98/.index.html>
Zugriff: 02.09.2004

- [Boulic & Thalmann 1992]** Boulic, Ronan, Thalmann, Daniel: *Combined Direct and Inverse Kinematic Control for Articulated Figures Motion Editin*; In: Computer Graphics Forum, October 1992; Vol. 2, No.4, S.189-202 (1992)
- [Boulic et al. 1990]** Boulic, Ronan; Magnenat-Thalmann, N; Thalmann, Daniel: *A Global Human Walking Model with real time Kinematic Personification*; In: The Visual Computer; Vol.6, Nr.6, S.344-358 (1990)
- [Bourke 1997]** Bourke, Paul : *Implicit Surfaces* ; (1997)
[http ://astronomy.swin.edu.au/~pbourke/modelling/implicitsurf/](http://astronomy.swin.edu.au/~pbourke/modelling/implicitsurf/)
 Zugriff: 07.09.2004
- [Breiner 2003a]** Breiner, Tobias: *Erweiterte Schnittstellendefinitionen für Motion DLLs*; In: GDV-Papiere der Johann Wolfgang Goethe Universität Frankfurt am Main; (2003)
- [Breiner 2003b]** Breiner, Tobias: *Interface Structure of the 3DG SUN Framework*; In: 3D-Generation-Papiere, Darmstadt (2003)
- [Breiner 2004]** Breiner, Tobias: *Namenskonventionen für Konstanten, Variablen, Methoden und Klassen in C++, Java und darauf aufbauende Szenegraphen*; In: GDV-Papiere der Johann Wolfgang Goethe Universität Frankfurt am Main, S.1-17 (2004)
- [Breiner 2005]** Breiner, Tobias: *User Manual für das e-Go Framework*; Johann Wolfgang Goethe Universität Frankfurt am Main; 157 Seiten (2005)

- [Breiner et al. 2004]** Breiner, Tobias; Dörner, Ralf; Seiler, Christian; Gudo, Michael:
Visualizing Organisms with Hydraulic Body Parts: A Case Study in Integrating Simulation and Visualization Models; In: Joint IEEE - EUROGRAPHICS Symposium on Visualization 2004; S.97-102 (2004)
- [Brierton 1998]** Brierton, Tom: *At Last, Foam Puppet Fabrication Explained*
Animation World Magazine, Issue 2.11, Februar 1998 (1998)
Auch Online:
<http://www.stopmotionanimation.com/>
Zugriff 12.08.2004
- [Brierton 2004]** Brierton, Tom: *Stop-Motion Armatures and Stop-Motion Animation Workshops*
<http://members.aol.com/tbrierton/armaturepage.htm>
Zugriff 12.08.2004
- [Brückmann 2004]** Brückmann GmbH.: *3D-Scanning*
<http://www.breuckmann.com/flash/breuckmann.html>
Zugriff: 17.08.2004
- [Bruder 2002]** Bruder, Ralph: *Natürlich künstlich – Zum gestalteten Verhältnis von Menschen und neuen Medien* (2002) ESSENER UNIKATE 17/2002, S. 56-61
Auch Online unter:
<http://www.uni-essen.de/unikate/pdf/17-Bruder.pdf>
Zugriff: 01.08.2004
- [Brunet 2003]** Brunet, Olivier : *PanardVision* ;
<http://www.panardvision.com>
Zugriff : 01.08.2003

- [Callidus 2004a] Callidus Precision Systems: *Was ist 3D-Scanning?* [a] *FAQs* [b]
 [Callidus 2004b] *Systeme* [c] (2004)
 [Callidus 2004c] <http://www.callidus.de/de/3dscan.html> [a]
<http://www.callidus.de/de/cp3200/faq.html> [b]
<http://www.callidus.de/de/cp3200/index.html> [c]
 Zugriff: 17.08.2004
- [Campusprogram 2004] Campusprogramm: *Motion capture*
http://www.campusprogram.com/reference/en/wikipedia/m/mo/motion_capture.html
 Zugriff: 2004-06-18
- [Cani & Hornus 2001] Cani, Marie-Paule; Hornus, Samuel: *Subdivision-Curve-Primitives: A New Solution for Interactive Implicit Modelling*; Shape Modelling International, Mai 2001 (2001)
- [Carpenter 2004] Carpenter, Brian: *MoCap Setting*
<http://www.briancarpenter.com/mocapset.html>
 Zugriff: 2004-06-18
- [Castro et al. 2000] Castro, Wiliam; Sautmann, Alfred; Schilgen Markus; Sautmann, M[arkus: *Noninvasive Three-Dimensional Analysis of Cervical Spine Motion in Normal Subjects in Relation to Age and Sex*; In: *Spine* 2000 25(4); S.443-449 (15.02.2000)
- [CGI 2004] Capure Geometry Internally: *First Article Inspection and Reverse Engeneering of small complex parts*
<http://www.reverse-eng.com/index.html>
 Zugriff: 17.08.2004

- [Chadwick et al. 1989] Chadwick, John; Haumann, David; Parent, Richard: *Layered Construction for Deformable Animated Characters*; In: Computer Graphics – Proceedings of the 1989 ACM SIGGRAPH; Vol. 23, Issue 3; S.243-252 (1989)
- [Chapman 1958] Chapman, G.: *The hydrostatic skeleton in the invertebrates*; Biological Reviews of the Cambridge Philosophical Society, Vol. 33; Issue 3; S. 338-360. (1958)
- [Chapman 1962] Chapman, Graham: *Versatility of Hydraulic Systems*; Journal of Experimental Zoology; Vol. 194; S. 249–270; (1962)
- [Chen & Zeltzer 1992] Chen, Tzu-Wei; Zeltzer, David: *Pump It Up: Computer Animation of a Biomechanically Based Model of Muskel Using the Finite Element Method*; In: Computer Graphics – Proceedings of SIGGRAPH'92, Vol. 26, Issue 2; S.89-98. (1992)
- [Cinema 2004a] Cinema: „*Jurassic Park III*”- *Production Notes*
- [Cinema 2004b] <http://www.cinema.com/article/item.phtml?ID=522&Page=6>
- [Cinema 2004c] - Bild mit zwei Animatronic-Dinosaurier [a]
- [Cinema 2004d] <http://www.cinema.com/article/item.phtml?ID=522&Page=7>
- Aussage von Winston [b] und Rosengrant [c]
- <http://www.cinema.com/article/item.phtml?ID=522&Page=10>
- Aussage von Lantieri [d]
- Zugriff: 10.08.2004
- [Cognitens 2004] CogniTens : *3D Vision Systems*
<http://www.cognitens.com/>
Zugriff : 17.08.2004

- [Colditz et al. 2005]** Colditz, Carsten; Coconu, Liviu; Deussen, Oliver; Hege, Christian: *Real-Time-Rendering of Complex Photorealistic Landscapes Using Hybrid Level-of-Detail Approaches*; Real-time visualisation and participation; 6th International Conference for Information Technologies in Landscape Architecture; Dessau 26.-28. Juni (2005)
- [Columbia 2000a]** Columbia Tristar: *Godzilla 2000* – DVD (2000)
- [Columbia 2000b]** Film [a]
- [Columbia 2000c]** Behind-the-scenes Footage [b]
Production Notes [c]
- [ComLogic 2004]** ComLogic Visual Solutions: Mantra4D
http://www.comlogic.de/mantra/mantra_4d.html
Zugriff: 19.11.2004
- [Coquillard 1990]** Coquillard, Sabine: *Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modelling*; In: Computer Graphics – Proceedings of the ACM SIGGRAPH'90, Vol. 24, Issue 4; S.187-196. (1990)
- [Criterion 2002]** Criterion Software Inc.: (2002)
<http://www.mantra4d.com>
Zugriff: 19.11.2004
- [Cuisenaire 1999]** Cuisenaire, Olivier : Distance Transformations : *Fast Algorithms and Applications to Medical Image Processing* ; Thèse présentée en vue de l'obtention du grade de Docteur en Sciences Appliquées (Ph.D. thesis) : Université Catholique de Louvain, Belgien (1999)
Auch Online unter :
<http://www.tele.ucl.ac.be/PEOPLE/OC/these/these.html>
Zugriff: 20.09.2004

- [Cybernarium 2005]** Cybernarium : Cybernarium – Ausstellung
<http://www.cybernarium.de/ausstellung.html>
 Zugriff: 01.08.2005
- [Cyberware 1999]** Cyberware Inc.: *3D Scanner Designed To Scrutinize Works Of Michelangelo*; Press Release; Monterey, California (06.01.1999)
<http://www.cyberware.com/news/pressReleases/pr023.html>
 Zugriff: 17.08.2004
- [Cyberware 2004]** Cyberware Inc.: *Whole Body Color 3D Scanner – Model WB4*
<http://www.cyberware.com/products/wbInfo.html>
 Zugriff: 17.08.2004
- [Daanen & Jeroen 1998]** Daanen, Hein A. M.; Van de Water, Jeroen G.: *Whole body scanners*, In: *Displays*; Vol. 19, S. 111-120. (1998)
- [Datapath 2000]** Datapath Ltd.: *RealIMation SDK Manual – Version 5.20*; Derby, England (2000)
- [Datapath 2001]** Datapath Ltd.: *RealIMation* (2001)
<http://www.realimation.com>
 Zugriff: 02.05.2001
- [Debunne et al. 1999]** Debunne, Gilles; Desbrun, Mathieu; Barr, Alan H.; Cani, Marie-Paule: *Interactive multiresolution animation of deformable objects*; In: *Proceedings of the Eurographics Workshop on Computer Animation and Simulation*; Springer-Verlag (1999)
- [Debunne et al. 2000]** Debunne, Gilles; Desbrun, Mathieu; Cani, Marie-Paule, Barr, Alan H.: *Adaptive simulation of soft-bodies in real-time*; In: *Computer Animation*; Philadelphia, USA S. 133-144 (2000)

- [Debunne et al. 2001]** Debunne, Gilles; Desbrun, Mathieu; Cani, Marie-Paule, Barr, Alan H.: *Dynamic real-time deformation using space & time adaptive sampling*; In: Proceedings of the ACM SIGGRAPH Conference August 2001; Los Angeles; S.31-36 (2001)
- [Delphi 2003]** Hoffknecht, Andreas: *Elektronik der Zukunft, Mini-Delphi-Studie – Technologieanalyse im Rahmen des Vorhabens "Identifikation und Bewertung von Ansätzen Zukünftiger Technologien" (Förderkennzeichen NT 2113) der Abteilung Zukünftige Technologien Consulting des VDI-Technologiezentrums im Auftrag und mit Unterstützung des BMBFs, Düsseldorf; ISSN 1436-5928 (Februar 2003)*
Auch Online unter:
<http://www.zukuenftigetechnologien.de/band46.pdf>
Zugriff: 02.08.2004
- [Deussen 2003a]** Deussen, Oliver: *Computergenerierte Pflanzen – Technik und Design digitaler Pflanzenwelten*; Springer-Verlag, Berlin, Heidelberg, New York (2003)
- [Deussen 2003b]** Deussen, Oliver: *Oliver Deussen's 1-billion triangle "Sunflowers" scene*; 4 Screenshots (2003)
http://www.openrt.de/Gallery/OliverDeussen_Sunflowers
Zugriff: 27.05.2005
- [Deussen et al. 1998]** Deussen, Oliver; Hanrahan, Pat; Lintermann, Bernd; Měch, Radomír; Pharr, Matt; Prusinkiewicz, Przemyslaw: *Realistic modeling and rendering of plant ecosystems*; In: Computer Graphics Proceedings of the ACM SIGGRAPH'98 – 19.-24.Juli 1998; Orlando, Florida; S.257-286, (Juli 1998)

- [Deussen & Lintermann 1997a]** Deussen, Oliver; Lintermann, Bernd: *Erzeugung komplexer botanischer Objekte in der Computergraphik*; Informatik-Spektrum; Nr. 20, Oktober 1997; Springer-Verlag, Berlin, Heidelberg, New York; S.208-215 (1997)
- [Deussen & Lintermann 1997b]** Deussen, Oliver; Lintermann, Bernd: *Interaktiv modellierung av komplekse naturlige objekter*; NORSIGD Info, Band 1/1997; S.5-6 (1997)
Auch Online unter:
<http://www.wisg.cs.uni-magdeburg.de/graphik/pub/files/Deussen.1997.IMK.pdf>
Zugriff: 26.05.2005
- [Deussen & Lintermann 2001]** Deussen, Oliver; Lintermann, Bernd: *Computerpflanzen*; Spektrum der Wissenschaft; Januar 2001 (2001)
- [Deussen & Lintermann 2005]** Deussen, Oliver; Lintermann, Bernd: *Computer Generated Plants and Organics*; Springer, New York (2005)
- [Digibotics 2004]** Digibotics Inc.: *3D Laser Digitizing Systems*
<http://www.digibotics.com>
Zugriff: 23.08.2004
- [Digital 2003]** Digital Pixels Inc.: *Metaballs*; (2003)
<http://www.pixels3d.com/products/metaballs.html>
Zugriff: 04.08.2003
- [DLR 2004]** Deutsches Zentrum für Luft- und Raumfahrt (DLR): *Satelliten als Wegweiser – Galileo* (2004)
http://www.dlr.de/dlr/ila2004/ila2004_hi_R.html
Zugriff: 06.08.2004

- [DLR 2004a]** DLR German Aerospace Center – Institute of Robotics and
[DLR 2004b] Mechatronics: *Laser Range Scanner* [a]; *Laser Range Scanner – Construction* [b] (2004)
<http://www.robotic.dlr.de/MECHATRONICS/SCAN/index.html>
 [a] .../mechatronics/scanner/construction.html [b]
 Zugriff: 26.08.2004
- [Dominé 2004]** Dominé, Sébastien: *Mesh Skinning*; Proceeding-Slides of the nVidia-Developer-Conference 2004 (2004)
 Auch Online unter:
<http://developer.nvidia.com/object/skinning.html>
 Zugriff: 11.08.2004
- [Dvir und Prushansky 2000]** Dvir, Zeevi; Prushansky, Tamara: *Reproducibility and instrument validity of a new ultrasonography-based system for measuring cervical spine kinematics*; *Clinical Biomechanics* 15: S.658-664 (2000)
- [DynEarth 2004]** Dynamic Earth Centre: *Science Information: Ankylosaurus reconstruction from “Walking with Dinosaurs”*
http://www.dynamicearth.co.uk/education/casualtiessurvivors_science.asp
 Zugriff: 09.08.2004
- [Edlinger & Gutmann 1997]** Edlinger, K.; Gutmann, Wolfgang Friedrich: *Molluscs as evolving constructions: necessary aspects for a discussion of their phylogeny*; *Iberus*; Vol. 15, Issue 2; S.51–66 (1997)
- [Efunda 2004a]** Efunda: *Efunda engineering fundamentals: „Damped Vibration of Free SDOF Systems“*
http://www.efunda.com/formulae/vibrations/sdof_free_damped.cfm
 Zugriff: 24.07.2004

- [Efunda 2004b]** Efunda: *Efunda engineering fundamentals: „Free SDOF System Calculator“*
http://www.efunda.com/formulae/vibrations/sdof_cal.cfm
24.07.2004
- [Engel 2002]** Engel, Wolfgang F.: *Direct3D – ShaderX: Fundamentals of Pixel Shaders*; Wordware Publishing, Frankenthal (2002)
Abgespeckte Version auch Online unter:
<http://www.shaderx.com/direct3d.net/tutorials/shader/shader3.html>
Zugriff: 20.09.2004
- [ESA 2004]** ESA: *EGNOS explained*
http://www.sso.admin.ch/Themes/04-Satellite_Navigation-Communication/pdf/EGNOS_fact_sheet_1_egnos.pdf
Zugriff: 06.08.2004
- [Euro 2002]** FNSEA: *Actualités: “1 euro = 1 dollar”* 17.01.2002
[http://www.euro.fnsea.fr/actu/actu_suite.asp?IdArticle=2827;](http://www.euro.fnsea.fr/actu/actu_suite.asp?IdArticle=2827)
Zugriff: 01.08.2004
- [Eyetronics 2004]** Eyetronics: *3D Scanning*
<http://www.eyetronics.com/>
Zugriff: 23.08.2004
- [Faro 2004]** Faro Inc.: *CAM Measurement*
<http://www.faro.com/>
Zugriff: 23.08.2004

- [Fastnacht et al. 2002]** Fastnacht, Michael; Hess, Natascha; Frey, Eberhard; Weiser, Hans-Peter: *Finite Element Analysis in Vertebrate Palaeontology*; Senckenbergiana lethaea; Vol. 82; Nr. 1; S.195-206 (2002)
- [FFMedia 2004]** Film Force Media: *Andy Serkis in his MoCap Suit And Gollum*
<http://ffmedia.ign.com/filmforce/image/andyserkis-suit-gollum15.jpg>
Zugriff: 04.08.2004
- [Filmlexikon 2004]** Filmlexikon: *Hinter den Kulissen: Über Masken Menschen und große Designs* (2004)
http://www.epilog.de/Film/He/Texte/Herr_der_Ringe_Die_Gefahrten_NZ_USA_2001_Produktion3.htm
Zugriff: 11.08.2004
- [Finkelstein 2002]** Finkelstein, Dan: *Dan's JPIII Page: Gallimimus* (2002)
<http://www.dansjp3page.com/hosting/realjp/Galli.html>
Zugriff 12.08.2004
- [Foley et al. 1996]** Foley, James D.; Van Dam, Andries; Feiner, Stephen K.; Hughes, John F.: *Computer Graphics: Principles and Practice* (2nd Edition in C). Addison-Wesley Longman Publishing Co. Inc.; Boston (1996)
- [Foxwell 2004]** Foxwell, Chris: *JarJar picture*
<http://wso.williams.edu/~rfoxwell/starwars/TPM/pics/JarJar.jpg>
Zugriff: 09.08.2004
- [Fraunhofer 1998]** Fraunhofer-ISI Publikationen: *Zukunft Nachgefragt* (1998)
<http://www.fraunhofer.de/german/publications/df/df1998/298-06.htm>
Zugriff: 01.05.2001

- [Frc 2002]** Frc: *Processor List* (2002)
<http://www.frc.ri.cmu.edu/users/hpm/book97/ch3/processor.list>
Zugriff: 28.07.2004
- [Fröhlich 2000]** Fröhlich, Torsten: *The Virtual Oceanarium*; In: Communications of the ACM, Vol. 43, Nr. 7, S.95-101 (2000)
Auch Online unter:
<http://delivery.acm.org/10.1145/350000/341868/p94-frohlich.pdf?key1=341868&key2=3007485901&coll=GUIDE&dl=GUIDE&CFID=27807904&CFTOKEN=20530658>
Zugriff: 22.09.2004
- [Gamma 2003]** Gamma Studio: *The Hulk Movie – FAQs*
<http://www.hulkmovie.com/navigation/faq.htm>
Zugriff 15.08.2004
- [Geisert 2004]** Geisert, Marc: *Die Wissenexplosion*; Freiburger Schriften zum Zeitgeist; Vol. 2, S.30-36 (2004)
- [Glonass 2004]** Russian Federation Misistry of Defence – Coordinational Scientific Information Center: *Glonass*
http://www.glonass-center.ru/intro1_w.html
Zugriff: 06.08.2004
- [GOM 2004]** Gesellschaft für Optische Messtechnik (GOM) mbH.: *3D Measuring Techniques*
<http://www.gom.com/>
Zugriff: 23.08.2004

- [Granum & Moeslund 2001]** Granum, Erik; Moeslund, Thomas Baltzer: *A Survey of Computer Vision Based Human Motion Capture*; In: *Computer Vision and Image Understanding*, 81(3); S.231-268. (2001)
- [Grasshoff & Gudo 2002]** Grasshoff, Manfred; Gudo, Michael: *The Origin of Metazoa and the Main Evolutionary Lineages of the Animal Kingdom – The Gallertoid Hypothesis in the Light of Modern Research*; *Senckenbergiana lethaea*; Vol. 82 Nr.1; S.295-314 (2002)
- [Graves 1993]** Graves, G.: *The Magic of Metaballs*; In: *Computer Graphics World*; S. 27-32 (Mai 1993)
- [Grimson 1981]** Grimson, W. Eric L.: *From Images to Surfaces: A Computational Study of the Human Early Visual System*; MIT Press; Cambridge, Massachusetts (1981)
- [Grinspun 2002]** Grinspun, Eitan; Krysi, Petr; Schröder, Peter: *CHARMS: A Simple Framework for Adaptive Simulation*; In: *Proceedings of the ACM SIGGRAPH*; San Antonio, Texas, S.281-290 (2002)
- [Gudo & Grasshoff Gudo 2002]** Gudo, Michael; Grasshoff, Manfred: *The Origin and Early Evolution of Chordates: The ‘Hydroskelett-Theorie’ and New Insights Towards a Metameric Ancestor*; *Senckenbergiana lethaea*; Vol. 82; Nr.1; S.325-346. (2002)
- [Gudo 2002]** Gudo, Michael: *The development of the critical theory of evolution: The scientific career of Wolfgang F. Gutmann*; *Theoiy Biosci*; S.101-137; Urban & Fischer-Verlag (2002)
Auch Online unter:
http://www.senckenberg.de/files/content/forschung/projekte/km_fet/gudo-2002-gutmann.pdf
Zugriff: 26.11.2004

- [Gudo et al. 2002] Gudo, Michael; Gutmann, Mathias; Scholz, Joachim [ed.]: *Concepts of functional, engineering and constructional morphology: Biomechanical approaches to recent and fossil organisms*; Schweizerbart-Publishing, Frankfurt am Main (2002)
- [Guenther 2004a]
[Guenther 2004b] Günter, Ramona: *Abriß zur Entwicklung der Rechentechnik* (2004)
<http://www.guenther-s.de/rechentechn/comput.html> [a]
<http://www.guenther-s.de/rechentechn/zuse1.html> [b]
Zugriff: 28.07.2004
- [Gutmann & Bonik 1981] Gutmann, Wolfgang Friedrich; Bonik, Klaus: *Kritische Evolutionstheorie – Ein Beitrag zur Überwindung altdarwinistischer Dogmen*; S. 227; Gerster-Verlag, Hildesheim (1981)
- [Guttmann 1995] Guttmann, Wolfgang Friedrich: *Die Evolution hydraulischer Konstruktion – organismische Wandlung statt altdarwinistischer Anpassung*; Kramer-Verlag, Frankfurt am Main (1995)
- [Hall-Holt & Rusinkiewicz 2001] Hall-Holt, Olaf; Rusinkiewicz, Szymon: *Stripe Boundary Codes for Real-Time Structured-Light Range Scanning of Moving Objects*; Proceedings of the Eighth International Conference on Computer Vision – ICCV (2001)
Auch Online:
<http://www.cs.princeton.edu/~smr/papers/realtimerange/realtimerange.pdf>
Zugriff: 25.08.2004
- [Hannah 1985] Hannah, Marsha Jo: *SRI's Baseline Stereo System*; In: Proceedings of the Image Understanding Workshop; S. 149-155 (1985)

- [Harding & Qian 2004]** Harding, Kevin; Qian, Xiaoping: *Phase Shifting Based 3D Systems*; Machine Vision Technical Papers 2004/1, S.14-18; General Electric Global Research Center; Niskayuna, NY (2004)
Auch Online unter:
http://www.machinevisiononline.org/public/articles/Phase_Shifting_3D_Systems.pdf
Zugriff: 30.08.2004
- [Hardware 2004]** Hardware-Net: *Hardware News* (10.02.2004)
www.hardware.net/news_35996.html
(25.03.2004)
- [Hart 2003]** Hart, Karl: *Pressure approximation in complex hydrostatic systems*; In: The European Physical Notes; Vol.1, Issue 03, S.12-17 (2003)
- [Hawkins 2003]** Hawkins, K.: *Direct3D*
<http://www.gamedev.net/reference/programming/features/d3do>
Zugriff:01.09.2003
- [Hawkins et al. 2001]** Hawkins, Tim; Cohen, Jonathan; Debevec, Paul: *A Photometric Approach to Digitizing Cultural Artifacts*; In: 2nd International Symposium on Virtual Reality, Archaeology, and Cultural Heritage; Glyfada, Greece (November 2001)
Auch Online unter:
http://www.debevec.org/Research/LS2/hawkins_debevec_VAST2001.pdf
Zugriff: 27.08.2004
- [Heise 2002]** Heise-News: *Von Star-Wars-Generatoren und Facelets* (18.05.2002)
<http://www.heise.de/newsticker/meldung/27473>
Zugriff: 01.08.2004

- [Henkel 2000]** Henkel, Guido: *Godzilla 2000-Review* (2000)
http://www.dvdreview.com/fullreviews/godzilla_2000.shtml
Zugriff: 15.08.2004
- [Hinze 2003]** Hinze, Sybille; Gaisser, Sibylle: *Nanotechnologie und Life Sciences*; Endbericht im Auftrag des Deutschen Bundestages, vorgelegt dem Büro für Technikfolgenabschätzung. Karlsruhe: Fraunhofer ISI (2003)
- [Hipp 2002]** Hipp, Cornelia: *Motion Capture* (2002)
<http://medien.informatik.uni-ulm.de/lehre/courses/ss02/Computergrafik/CorneliaHipp.pdf>;
Zugriff: 1.3.2004
- [Hitchcock 2004]** Hitchcock-Manthey LLC.: *Products*
<http://www.camsysinc.com>
Zugriff: 17.08.2004
- [Hoffmann 1997]** Hoffmann, Kay: *Vom spektakulären Computereffekt zur unscheinbaren, alltäglichen Manipulation*; Heise-Telepolis: (27.02.1997)
<http://www.heise.de/tp/deutsch/special/film/6119/2.html>
Zugriff 15.08.2004
- [Horn & Hiriyati 1999]** Horn, Eli; Hiriyati, Nani: *Toward Optimal Structured Light Patterns*; In: *Image and Vision Computing*, Vol. 17/1999, Nr.1; S.87-97 (1999)

- [Hsw 2004,1..9]** HowStuffWorks: *How Animatronics Works*
<http://entertainment.howstuffworks.com/animatronic1.htm> [,1]
<http://entertainment.howstuffworks.com/animatronic2.htm> [,2]
 ... /animatronic9.htm [,9]
 Zugriff: 10.08.2004
- [Hubbard 2003]** Hubbard, Michael: *Godzilla 2000 – FAQs about the Production* (2003)
<http://www.gvsdestoroyah.com/FaqProd.htm>
 Zugriff 15.08.2004
- [Hyun et al. 2002]** Hyun, Dae-Eun; Yoon, S. H.; Kim, Myung-Soo: *Human Limbs Reconstruction based on Ellipsoidal Sweeping*; In: Journal of the Korean Computer Graphics Society (KCGS), Dezember'02. (2002)
- [Hyun et al. 2003]** Hyun, Dae-Eun; Yoon, Seung-Hyun; Kim, Myung-Soo; Jüttler, Bert: *Modeling and Deformation of Arms and Legs Based on Ellipsoidal Sweeping*; In: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications (PG 2003); 08.-10. Oktober 2003; Canmore, Kanada; IEEE Press; S.204-213 (2003)
- [ICVGAN 1994]** ICVGAN – International Committee on Veterinary Gross Anatomical Nomenclature: *Nomina Anatomica Veterinaria*; (*Abk. NAV*); 4th Edition; World Association of Veterinary Anatomists Handbook (1994)
- [IFAA 2003]** IFAA – International Federation on Anatomical Associations: *Terminologia Anatomica – International Anatomical Terminology*; 3rd Program (2003)

- [Immersion 2004a]** Immersion Corporation: *Digitizer* [a] *Creating Realistic 3D*
[Immersion 2004b] *Models with the Immersion MicroScribe and Ghost 3D's Scribe-*
[Immersion 2004c] *iT* [b] *An Army of Ants: Digitizing Bugs with MicroScribe and*
Vertisketch [c] (2004)
<http://www.immersion.com/digitizer/> [a]
<http://www.immersion.com/digitizer/docs/bubba.pdf> [a]
http://www.immersion.com/digitizer/case_study_gallery/animation_4.php [b]
Zugriff: 02.09.2004
- [InSpeck 2004]** InSpeck Inc.: *3D Digitizing and modelling solutions*
<http://www.inspeck.com>
Zugriff: 23.08.2004
- [Intel 2003]** Zhirnov, Victor V.; Cavin, Ralph K.; Hutchby, James A.;
Bourianoff, George I.: *Limits to Binary Logic Switch Scaling – A*
Gedanken Model (Intel-Study); In: Proceedings of the IEEE
Conference, Vol. 91, Issue 11; S.1934-1939 (1.11. 2003)
- [Intel 2004]** Intel Research Department: *Moore's Law* (2004)
<http://www.intel.com/research/silicon/mooreslaw.htm>
Zugriff: 28.07.2004
- [Iqsun 2004]** Iqsun GmbH.: *New 3D Laserscanner Generation*
http://www.iqvolution.com/pages_e/3dls_e/index_3dls_e.html
Zugriff: 26.08.2004
- [Isel 2004]** Isel KG.: *Interferometrische Grundeinstellungen* (2004)
[http://www.de.isel.com/wysise/iselwys5.nsf/files/3dmess_d.pdf/\\$FILE/3dmess_d.pdf](http://www.de.isel.com/wysise/iselwys5.nsf/files/3dmess_d.pdf/$FILE/3dmess_d.pdf)
Zugriff: 01.09.2004

- [Jackson 1997]** Jackson, Wendy: *An Interview With Aardman's Peter Lord*
Animation World Magazine, Issue 2.2, May 1997
Auch Online:
<http://www.stopmotionanimation.com/>
Zugriff 12.08.2004
- [Jasper 2004]** Jasper, Dirk: *Filmlexikon: "Back to Gaya" Produktionsnotizen*
<http://www.djfl.de/entertainment/djfl/1115/111664pr.html>
Zugriff 15.08.2004
- [Java3D 2003]** Java3D Community: *Java3D*
<http://www.j3d.org>
Zugriff: 01.09.2003
- [Jenkins et al. 2002]** Jenkins, Ian, Thomason, Jeffrey J.; Norman, David B.: *Primates and Engineering Principles: Applications to Craniodental Mechanisms in Ancient Terrestrial Predators*; *Senckenbergiana lethaea*; Vol. 82, Issue 1, S.223-240 (2002)
- [Keeve et al. 1996]** Keeve, Erwin; Girod, Sabine; Girod, Bernd: *Computergraphik in der Craniofacialen Chirurgieplanung*; In: itti-Sonderheft „Graphische Datenverarbeitung“ (1996)
auch Online unter:
<http://www.caesar.de/uploads/media/S96-1.pdf>
Zugriff: 24.07.2004
- [Keferstein 2003]** Keferstein, Claus P.: *Marktanalyse optischer Sensoren 1D-3D*; NTB Interstaatliche Hochschule für Technik Buchs, Schweiz (2003)
Vorinformation auch Online unter:
http://www.ntb.ch/files/0/2550/Fly21_Marktanalyse03_v03.pdf
Zugriff: 01.09.2004

- [Kerlow 1996] Kerlow, Ian V: *The Art of 3-D Computer Animation and Imaging*; John Wiley & Sons, Inc. (1996)
- [Kinetix 2003] Kinetix: *3DstudioMax*;
<http://www.ktx.com>
Zugriff: 08.09.2003
- [Kish 2002] Kish, Laszlo, B.: *End of Moore's law: thermal (noise) death of integration in micro and nano electronics*, Physics Letters A 305 (2002) S.144–149
Auch Online:
http://ee.tamu.edu/%7EElaszlo/Moore_PLA.pdf
Zugriff: 01.08.2004
- [Kmed 2005] k-MED Community: *Knowledge-Based Multimedia Medical Education* (2005)
<http://www.k-med.org>
Zugriffdatum: 12.08.2005
- [Knep 1995] Knep, Brian; Hayes, Craig; Sayre, Rick; Williams, Tom: *Dinosaur Input Device*; In: Proceedings of CHI'95: ACM Conference on Human Factors in Computing Systems; S. 12-24 (1995)
- [Knep 2003] Knep, Brian: *DID* (2003)
<http://home.comcast.net/~bleppie/blep/jurassicPark/did.htm>
Zugriff: 1.1.1994
- [Koestler 1984] Koestler, Arthur: *Die Wurzeln des Zufalls*; Scherz Verlag, München (1984)
- [Koestler 1989] Koestler, Arthur: *The Gost in the Machine*; Arkana Books (1989)

- [Konica 2004]** Konica Minolta: *3D Digitizers*
<http://www.minoltausa.com/vivid/contact/main.asp>
Zugriff: 23.08.2004
- [Koris 2005]** Koris, Juri: *Animation with optical Motion Capturing*; Los Angeles (2005)
- [Koschan 2004]** Koschan, Andreas: *What is New in Computational Stereo Since 1989: A Survey on Current Stereo Papers*
Auch Online unter:
<http://iristown.engr.utk.edu/~koschan/paper/Stereo-Report2.1.pdf>
Zugriff: 30.08.2004
- [Kréon 2004]** Kréon Technologies: *New 3D Scanning solutions*
<http://www.kreon3d.com/>
Zugriff: 23.08.2004
- [Krömker & Dörner 2003]** Krömker, Detlef; Dörner, Ralf: *Visualisierung in der Bioinformatik*; In: Informatik 2003 – Innovative Informatikanwendungen, Band 1, Beiträge der 33. Jahrestagung der Gesellschaft für Inforematik e.V. (GI), 29.9-2.10.2003; Vol. 34, S.352-353 (2003)
- [Kurzweil 2003]** Kurzweil, Ray: *Kurzweil's Law aka "the law of accelerating returns"* (12.01.2004)
<http://www.kurzweilai.net/meme/frame.html?m=1>
Zugriff: 28.07.2004

- [**Laserdesign 2004a**] Laserdesign Inc.: *3D Laser Scanning and 3D Laser Scanners* [a]
- [**Laserdesign 2004b**] *FAQs* [b]
<http://www.laserdesign.com/> [a]
<http://www.laserdesign.com/faq.htm> [b]
 Zugriff: 23.08.2004
- [**Lebouthillier 2002**] Lebouthillier, Arthur Ed: *Animated Artificial Fishes* RSSC, The Robot Builder, Vol. 12, Nr. 2.; S.34-45 (2002)
- [**Lee 1999**] Lee, Aaron W. F.; Dobkin, David; Sweldens, Wim; Schröder, Peter: *Multiresolution Mesh Morphing*; In: Proceedings of the ACM SIGGRAPH; Los Angeles, S.343-350 (1999)
- [**Legus 2001**] Legus Inc.: *Legus3D*;
<http://www.legus3D.com>
 01.12.2001
- [**Lehan 2001**] Lehan, Scott: *The Making of the Lord of The Rings*; In: *Fim&Video Magazine* 10/2001, 3-8 (Oktober 2001)
 Auch Online unter:
<http://www.cosmo.sgi.com/pdfs/3234.pdf>
 Zugriff: 27.08.2004
- [**Leong 2000**] Leong, Anthony: *Dinosaur Movie Review* (2000)
<http://www.geocities.com/aleong1631/dinosaur.html>
 Zugriff 15.08.2004
- [**LeVeque 2002**] LeVeque, Randall L.: *Finite Volume Methods for Hyperbolic Problems*; Cambridge Texts in Applied Mathematics (2002)

- [Lewis et al. 2000]** Lewis, John P.; Cordner, Matt; Fong, Nickson: *Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation*; In: Proceedings of the ACM Siggraph 2000; (2000)
Zusätzliche Code-Beispiele auch Online unter:
http://www.idiom.com/~zilla/Work/PSD/PSD_WEBNOTES/index.html
Zugriff: 21.09.2004
- [Linke et al. 2000]** Linke, Marcus; Winkler, Peter: *Das M&T-Computer-Lexikon*; Wilhelm Heyne Verlag, München (2000)
- [Lintermann & Deussen 1998]** Lintermann, Bernd; Deussen, Oliver: *A Modelling Method and User Interface for Creating Plants*; Computer Graphics Forum Vol. 20, Nr. 1 (1998)
- [Lobb 2002]** Lobb, Adele: *Virtual Reality Fish*; Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science, Rhodes University (2002)
- [Lobb 2003]** Lobb, Adele; Shaun Bangay: *Realistic Autonomous Fish for Virtual Reality*; In: Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa; Session 1; S.167 – 188
Auch Online unter:
<http://delivery.acm.org/10.1145/610000/602361/p167-lobb.pdf?key1=602361&key2=5236485901&coll=GUIDE&dl=GUIDE&CFID=27806492&CFTOKEN=73293837>
Zugriff: 22.09.2004

- [Löfken 2002]** Löfken, Jan Oliver: *Wo bin ich? ESA-Internetservice verbessert GPS-Auflösung auf zwei Meter*; In: *Bild der Wissenschaft* 9/2002; S.25 (16.09.2002)
auch Online:
<http://acce.fhnon.de/Aktuelles/SchwarzesBrett/GPS.htm>
sowie
<http://www.wissenschaft.de/wissen/news/149588.html>
Zugriff: 06.08.2004
- [Lorensen & Cline 1987]** Lorensen, William E.; Cline, Harvey E.: *Marching Cube: A High Resolution 3-D Surface Construction Algorithm*; In: *Computer Graphics – Proceedings of the 1987 ACM SIGGRAPH*; Vol.21; Issue 3; S.163-169 (1987)
- [Lukas 2000]** Lukas, George: *Star Wars: Episode V – The Empire Strikes Back* (2000)
<http://www.starwars.com/episode-v/classic/2000/06/classic20000626.html>
- [Mac 2004]** Mac and Win: *Aussagen über Computer*
<http://www.mac-and-win.de/>
Zugriff: 28.07.2004
- [Maciel et al. 2000]** Maciel, Anderson; Da Sasso Freitas, Carla Maria; Nedel, Luciana Porcher: *Modelagem de Articulações para Humanos Virtuais Baseada em Anatomia*; In: *Procedimentos do VII. Semana Acadêmica do Programa Pós-Graduação (PPGC) em Computação do Brasil*; Porto Alegre, RS, Brasilien; S.110-112; (2000)
Auch Online unter:
<http://www.inf.ufrgs.br/pos/SemanaAcademica/Semana2000/AndersonMaciel/>
Zugriff: 21.09.2004

- [Maciel et al. 2002]** Maciel, Anderson; Nedel, Luciana Porcher; Da Sasso Freitas, Carla M.: *Anatomy-Based Joint Models for Virtual Humans Skeletons*; In: Proceedings of the International Conference on Computer Animation (CA 2002), Genf, Schweiz; IEEE Computer Society, Los Alamitos; S.220-224 (2002)
Auch Online unter:
<http://vrlab.epfl.ch/~amaciel/docs/CA2002.pdf>
Zugriff: 21.09.2004
- [Maier 2004a]** Maier, Thomas: *Festival für digitalen Film – Computer erobert*
[Maier 2004b] *Leinwand*; In: n-tv news [a]
Kino-Festival in Frankfurt; Der Computer erobert die
Leinwand; In: Rhein-Main.Net - Aktuelle Meldungen [b]
(22.09.2004)
<http://www.n-tv.de/5427996.html> [a]
<http://www.www.fnp.de/sixcms/detail.php/1866267> [b]
Zugriff: 22.09.2004
- [Mandelbrot 1982]** Mandelbrot, Benoit B.: *The Fractal Geometry of Nature*;
Freeman, San Francisco (1982)
- [Marr & Poggio 1979]** Marr, David; Poggio, Tommaso: *A Computational Theory of Human Stereo Vision*; In: Proceedings of the Royal Society of London, Vol. 204 of B; S. 301-328 (1979)
- [Matrox 2004]** Matrox Graphics Inc. : *Le Bump Mapping ; Comparaison de deux techniques de Bump Mapping : la gestion de l'alpha en multi-passes et l'Environment Mapped Bump Mapping ;*
gefunden Online unter :
http://www.matrix.com/mga/francais/products/tech_info/pdf/enbm.pdf
Zugriff: 13.12.2004

- [Matten 2003]** Matten, Friedemann (Hrsg): *Total Vernetzt*, Springer-Verlag (2003)
- [Maughan 2004]** Maughan, Christopher: *NVEffects Sample Descriptions And Effects Browser Framework – Mesh Skinning With Vertex Programs* (2004)
http://cvs1.nvidia.com/MEDIA/html/NVEffectsExplained_files/image004.jpg
 Zugriff: 11.08.2004
- [Mayer 2003]** Mayer, Hubert: *On the stem curve of a tall palm in a strong wind* (2003)
www.fernuni-hagen.de/NUMERIK/kurse/mathepraks0203/mayer.pdf
 Zugriff: 02.09.2003
- [McPheeters 1990]** McPheeters, C. W.: *Isosurface Modelling of Soft Objects in Computer Graphics*; PhD Thesis; National Centre for Computer Animation, Department of Communication and Media, Dorset Institute (1990)
- [Mejas 2004a]** Mejias, Paul: *The Lost World – Animatronics [a]*
- [Mejas 2004b]** *Jurassic Park III – Suit [b]*
- [Mejas 2004c]** *The Island of Dr. Moreau – Boarman Cap [c,d,e]*
- [Mejas 2004d]** <http://paulmejias.heyyoukids.com/scrapbook/studio-shots-image-2.html> [a]
- [Mejas 2004e]** [../scrapbook/studio-shots-image-7.html](#) [b]
[../character-effects/moreau-image-1.html](#) [c]
[../character-effects/moreau-image-2.html](#) [d]
[../character-effects/moreau-image-3.html](#) [e]
 Zugriff: 09.08.2004

- [Mendoza et al. 2002]** Mendoza, Cesar; Sundaraj, Kenneth; Laugier, Christian: *Issues in Deformable Virtual Objects Simulation with Force Feedback*; In: Proceedings of the IARP International Workshop on Human Robot Interfaces; Rom, Italien; November 2002; S.57-61 (2002)
Auch Online unter:
<http://vcg.isti.cnr.it/people/vcgpeople/mendoza/iarp02.pdf>
Zugriff: 24.07.2004
- [Menon 1996]** Menon, Jai: *An Introduction to Implicit Techniques*; In: SIGGRAPH Course Notes on Implicit Surfaces for Geometric Modeling and Computer Graphics; Course Nr.: 11 (1996)
Auch Online unter:
<http://cis.k.hosei.ac.jp/~F-rep/SIG96.html>
Zugriff: 20.09.2004
- [Metamotion 2004a]** MetaMotion: *Motion Capture – Data Gloves*
http://www.metamotion.com/hardware/motion-capture-hardware-gloves-Datagloves.htm#14_sensor 2004
Zugriff: 2004-06-18
- [Metric 2004]** Metric Vision: *Coherent Laser Radar 3D Scanning*
<http://www.metricvision.com/pages/home.htm>
Zugriff: 23.08.2004
- [Miller 1988]** Miller, Gavin S. P.: *The Motion Dynamics of Snakes and Worms*; Computer Graphics, August 1988; Vol. 22, Nr. 4, S. 169-173 (1988)
- [Monkey 2004]** Digital Image Design Inc.: *Monkey Motion Capturing Bones*
<http://www.didi.com/www/areas/products/monkey2/>
Zugriff: 01.01.2004

- [Moore 1965]** Moore, Gordon E.: *Cramming more components onto integrated circuits* (1965)
Electronic Magazine, Vol. 38, Nr. 8. (19.04.1965)
auch Online unter:
<ftp://download.intel.com/research/silicon/moorespaper.pdf>
(28.07.2004)
- [Moore 1997]** Wolfson, Jill; Cervantes, Theo: *An Interview with Gordon Moore*; San Jose Mercury News (26.01.1997)
auch Online unter:
http://www.thetech.org/exhibits/online/revolution/moore/i_c.html
Zugriff: 28.07.2004
- [Moore 2003]** Moore, Gordon E.: *No exponential is forever ... but we can delay forever* (2003)
Presentation Proceedings of the International Solid State Circuits Conference (10.02.2003)
auch Online unter:
ftp://download.intel.com/research/silicon/Gordon_Moore_ISSCC_021003.pdf
Zugriff: 28.07.2004
- [Moravec 1998a]** Moravec, Hans (1998) *When will computer hardware match the human brain?* Journal of Evolution and Technology, März 1998, Vol. 1, S.1-4
Auch Online:
<http://www.transhumanist.com/volume1/moravec.htm>
Zugriff: 28.07.2004
- [Moravec 1998b]** Moravec, Hans: *ROBOT: mere machine to transcendent mind*; Oxford University Press (1998)

- [Motionanalysis 2004]** MotionAnalysis: *International Conference on Equine Locomotion*
http://www.motionanalysis.com/about_mac/equine.html
Zugriff: 01.01.2004
- [Mulder 1994]** Mulder, Alex S.: *Human Movement Tracking Technology*; Report 94-1; Hand Centered Studies of Human Movement Project; Simon Fraser University, School of Kinesiology (Juli 1994)
- [Müller 2004]** Müller, Heinrich: *VTM-Videobasiertes Trackingmodul-Schlussbericht zum gleichnamigen Teilprojekt des Leitprojektes „intuitive Mensch-Maschine-Interaktion für die vernetzte Informationswelt der Zukunft“ – INVITE (2002)*
auch Online unter:
http://ls7-www.cs.uni-dortmund.de/research/projekte/invite/Schlussbericht_Invite.pdf
Zugriff: 05.08.2004
- [Müller et al. 2002]** Müller, Matthias; Dorsey, Julie; McMillan, Leonard; Jagnow, Robert; Cuttler, Barbara: *Stable Real-Time Deformations*; In: Proceedings of the ACM SIGGRAPH – Symposium on Computer Animation (2002)
Auch Online unter:
http://graphics.ethz.ch/Downloads/Publications/Papers/2002/p_Mue02.pdf
Zugriff: 21.09.2005
- [Nasa 2002]** NASA: *Nasa Headlines – Quaoar, a Cold New World*; (07.10.2002)
http://science.nasa.gov/headlines/y2002/07oct_newworld.htm
Zugriff: 18.04.2004

- [Nedel & Thalmann 1999]** Nedel, Luciana Porcher; Thalmann, Daniel: *Animation of Virtual Human Bodies Using Motion Capture Devices*; In: Proceedings of the 2nd Brazilian Workshop on Virtual Reality – WRV’99; Marilia, SP, Brazil; November 1999; S. 139-150; (1999)
Auch Online unter:
<http://www.inf.ufrgs.br/cg/publications/nedel/wrv99.pdf>
Zugriff: 23.07.2004
- [Nedel 1998]** Nedel, Luciana Porcher: *Anatomic modelling of human bodies using physically based muscle simulation*; Ph.D. Thesis, LIG – Computer Graphics Lab, EPFL – École Polytechnique Fédérale de Lausanne; Lausanne, Switzerland (Juni 1998)
Auch Online unter:
<http://www.inf.ufrgs.br/~nedel/texts/Thesis.pdf>
Zugriff: 23.07.2004
- [Nist 2003]** NIST: *NIST*; Proceedings of the LADAR Calibration Facility Workshop, Juni 12-13, 2003. (2003)
National Institute of Standards and Technology, NISTIR Publication 7054, Gaithersburg, USA.
Auch Online unter:
http://www.bfrl.nist.gov/861/CMAG/publications/NISTIR_7054.pdf
Zugriff: 17.08.2004
- [Nodna 2004]** NoDNA GmbH.: *Virtual Character and Motion Capturing*
www.nodna.com/technologie.2.0.html
Zugriff: 01.07.2004
- [Nub3D 2004]** Nub3D: *Soluciones para el digitalizado tridimensional*
<http://www.nub3d.com/Spanish/Index.htm>
Zugriff: 23.08.2004

- [Ohta & Kanade 1985]** Ohta, Yuichi; Kanade, Takeo: *Stereo by Intra- and Inter-scanline Search Using Dynamic Programming*; In: IEEE Transactions On Pattern Analysis and Machine Intelligence, Vol.7, Nr.2; S. 139-154 (1985)
- [OpenGL 2004]** OpenGL Konsortium: *OpenGL*;
<http://www.opengl.org>
Zugriff: 20.09.2004
- [Optimet 2004]** Optimet Ltd.: Optical Metrology
<http://www.optimet.com/>
Zugriff: 31.08.2004
- [Opton 2004]** Opton Ltd. Of Japan: *High Speed 3D Digitizers*
<http://www.eois.com/>
Zugriff: 23.08.2004
- [Otto 1977a]** Otto, Frei: *Das Konstruktionsniveau Pneu*; Mitteilungen des Institutes für leichte Flächentragwerke der Universität Stuttgart (IL), Vol. 9 (Pneus in Natur und Technik): S.23-47. (1977)
- [Otto 1977b]** Otto, Frei: *Wachsende und sich teilende Pneus*; Mitteilungen des Institutes für leichte Flächentragwerke der Universität Stuttgart (IL), Vol. 9 (Pneus in Natur und Technik); S. 22-97. (1977)
- [Otto 1978]** Otto, Frei: *Der Pneu; Bauprinzip des Lebens*; Bild der Wissenschaft; Vol. 15; S.124-135 (1978)
- [Outnow 2004]** Outnow: *OutNow – die andere Seite des Films: Alien Screenshots*
<http://outnow.ch/Media/Img/1979/Alien/movie.1/12.jpg>
Zugriff: 09.08.2004

- [Pantin 1951]** Pantin, Carl Frederick Abel (1951): *Organic design. – Advancement of Science*; Vol. 8; S.138-150.
- [Papst et al. 2000]** Papst, Thomas ; Orozco, Silvino ; Scheffel, Uwe : *Der Radeon von ATI : Bump Mapping* (21.07.2000)
http://www.de.tomshardware.com/graphic/20000721/ati_radeon-16.html
 Zugriff: 14.12.2004
- [Parent 2002]** Parent, Rick: *Computer Animation: Algorithms and Techniques*; Academic Press, San Diego (2002)
- [Peitgen & Saupe 1988]** Peitgen, Heinz-Otto; Saupe, Dietmar: *The Science of Fractal Images*; Springer-Verlag, New-York (1988)
- [Perceptron 2004]** Perceptron Inc.: *The focus on process*
<http://www.perceptron.com>
 Zugriff: 23.08.2004
- [Petrov 2002]** Petrov, Yuri: *Disparity capture by flanking stimuli: a measure for the cooperative mechanism of stereopsis*; In: *Vision Research* 42; S.809–813 (2002)
 Auch Online unter:
http://www.ski.org/SPMcKee_lab/yury/DOCS/VisRes01.pdf
 Zugriff: 30.08.2004
- [Piegl & Tiller 1997]** Piegl, Les A.; Tiller, Wayne: *The NURBS book*; Springer-Verlag, Heidelberg (1997)
- [Pocock & Rosebuch 2002]** Pocock, L., Rosebuch, J.: *The Computer Animation's Technical Handbook*; Academic Press, London (2002)

- [Polhelmus 2004]** Polhelmus technologies: *MoCap Systems and 3D Scanners*
<http://www.polhelmus.com>
 Zugriff : 24.06.2004
- [Pollard et al. 1985]** Pollard, Stephen B.; Mayhew, John E. W.; Frisby, John. P.:
PMF: A Stereo Correspondence Algorithm Using a Disparity Gradient Limit; In: *Perception* 14; S.449-470. (1985)
- [Preuschoft & Witzel 2002]** Preuschoft, Holger; Witzel, Ulrich: *Biomechanical Investigations on the Skulls of Reptiles and Mammals*;
Senckenbergiana lethaea, Vol. 82, Issue 1, S.207-222 (2002)
- [Preuschoft & Witzel 2004]** Preuschoft, Holger; Witzel, Ulrich: *A Biomedical Approach to Craniofacial Shape in Primates, Using FESA*; *Folia Primatol* 2004; Vol.75; S.341-354; Universität Greifswald (2004)
- [Prusinkiewicz & Lindenmayer 1990]** Prusinkiewicz, Przemyslaw; Lindenmayer, Aristid: *The Algorithmic Beauty of Plants*; Springer, Heidelberg (1990)
- [Quaoar 2004]** Uniprotokolle: *Lexikoneintrag Quaoar*
<http://www.uni-protokolle.de/Lexikon/Quaoar.html>
 Zugriff: 16.04.2002

- [Rapido3D 2004a]** Rapido3D Ltd.: *Laser Scanning*;
- [Rapido3D 2004b]** -*Laser Scanning & Rapid Prototyping Bureau* [a]
- [Rapido3D 2004c]** -*Examples* [b]
- [Rapido3D 2004d]** -*FAQs* [c]
 -*Scanning* [d] (2004)
<http://www.rapido3d.co.uk> [a]
<http://www.rapido3d.co.uk/bbc.htm> [b]
<http://www.rapido3d.co.uk/xindex.htm> [c]
<http://www.rapido3d.co.uk/scanning.htm> [d]
 Zugriff: 17.08.2004
- [Rechenwerkzeug 2004]** Atzbach, Reiner: *Die Faszination der Mechanik*
<http://www.rechenwerkzeug.de/default.htm>
 Zugriff: 28.07.2004
- [Reiners 2003]** Reiners, Dirk: *OpenSG*; (2003)
<http://www.opensg.org>
 Zugriff: 20.09.2004
- [Remondino 2003]** Remondino, Fabio: *From point cloud to surface: the modeling and visualization problem*; In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV, Part 5/W10; ISPRS International Workshop on “Visualization and Animation of Reality-based 3D Models”, 24-28 Februar 2003, Tarasp-Vulpera, Schweiz (2003)
 Auch Online unter:
http://www.photogrammetry.ethz.ch/general/persons/fabio/tarasp_remondino.pdf
 Zugriff: 22.09.2004

- [Remondino 2004a]** Remondino, Fabio: *Character Animation and Understanding from Sequences of images (C.A.U.S.E.)*; Poster of: EPSRC/British Machine Vision; Summer School in Computer Vision, University of Surrey, UK (June 2001)
- [Remondino 2004b]** Remondino, Fabio: *Character reconstruction and animation from monocular sequence of images*; In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXV-B5; S.34-37 (2004)
auch Online unter:
<http://www.photogrammetry.ethz.ch/general/persons/fabio/isprs04.pdf>
Zugriff: 05.08.2004
- [Remondino 2004c]** Remondino, Fabio; Roditakis, Andreas: *Human motion reconstruction and animation from video sequences*; In: Proceedings of the 17. International Conference on Computer Animation and Social Agents (CASA2004); 7.-9. Juli 2004; Computer Graphics Society (CGS), Genf, Schweiz, S.347-354 (2004)
Auch Online unter:
<http://www.photogrammetry.ethz.ch/general/persons/fabio/casa04.pdf>
Zugriff: 05.08.2004
- [Ren et al. 2002]** Ren, Liu; Pfister, Hanspeter; Zwicker, Mathias: *Object Space EWA Surface Splatting: A Hardware Accelerated Approach to High Quality Point Rendering*; Computer Graphics Forum; Vol. 21, Issue 3; September 2002 (2002)

- [Renishaw 2004]** Renishaw plc.: *Digitising Systems*
<http://www.renishaw.com/client/product/UKEnglish/PGP-890.shtml>
 Zugriff: 23.08.2004
- [Riegl 2004]** Riegl. GmbH.: *Laser Measurement Systems*
<http://www.riegl.com/>
 Zugriff: 24.08.2004
- [Robertson 1997]** Robertson, Barbara: *Computer Graphics World – Real Troopers* (1.9.1997)
http://cgw.pennnet.com/Articles/Article_Display.cfm?Section=Archives&Subsection=Display&ARTICLE_ID=49306&KEYWORD=tippett
 Zugriff: 5.1.2004
- [Roland 2001]** Roland Digital Group: *Scanning Machines* (2001)
<http://www.rolanddga.com/products/3D/scanners/>
 Zugriff: 23.08.2004
- [Romer 2002]** Romer CimCore: *Portable CMM Products* (2002)
<http://www.romer.com/main/products.php>
 Zugriff: 23.08.2004
- [Rothacher & Zebhauser 2000]** Rothacher, Markus; Zebhauser, Benedikt: *Einführung in GPS, Tutorial zum 3.Sapos-Symposium 2000 in München;*
 Institut für Astronomische un Physikalische Geodäsie/
 Technische Universität München, Forschungseinrichtung
 Satellitengeodäsie, München (2000)
 auch Online unter:
http://step.iapg.verm.tu-muenchen.de/reports/iapg_fesg_rpt_08.pdf
 Zugriff: 06.08.2004

- [Rozenberg & Salomaa 1980]** G. Rozenberg, A. Salomaa: *The Mathematical Theory of Lindenmayer Systems*; Academic Press (1980)
- [Rozenberg & Salomaa 1992]** Rozenberg, G.;Salomaa, A: *Lindenmayer Systems. Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology*; Springer, Berlin (1992)
- [Saar et al. 2004a]** Saar, Victor; Rötzler, Ingmar; Itskov, Costa:
[Saar et al. 2004b] *Direct 3D Tutorial 5: Texture Mapping [a]*
Direct 3D Tutorial 14: Dot3 Bump Mapping [b]
[http://www.riaz.de/tutorials/d3d5/ d3d5.html](http://www.riaz.de/tutorials/d3d5/d3d5.html) [a]
[http://www.riaz.de/tutorials/d3d14/ d3d14.html](http://www.riaz.de/tutorials/d3d14/d3d14.html) [b]
 Zugriff: 14.12.2004
- [Sarni et al. 2004]** Sarni, Sofiane; Maciel, Anderson; Boulic, Ronan; Thalmann, Daniel: *Evaluation and Visualisation of Stress and Strain on Soft Biological Tissues in Contact*; In: International Conference on Shape Modeling and Applications 2004 (SMI'04); 07. – 09. Juni 2004; Genova, Italy; S. 255-262 (2004)
 Auch Online unter:
http://vrlab.epfl.ch/~amaciel/docs/sarni_maciel_boulic_thalmann_smi04.pdf
 Zugriff: 24.07.2004
- [Scheepers 1996]** Scheepers, Coenraad Frederik: *Anatomy-Based Surface Generation for Articulated Models of Human Figures*; Ph.D. Thesis; The Ohio State University, Department of Computer and Information Science (1996)
- [Schiller 2003]** Schiller, Arnulf: *Konoskopische Holographie – Laboraufbau*; CD-Labor für Sens. Messtechnik, c/o Institut für Automation, Montanuniversität, Leoben (2003)

- [Schroth 2004]** Schroth GmbH.: *Conoskopische Holographie* (2004)
<http://www.optimet.de/Technology.htm>
 Zugriff: 01.09.2004
- [Schwertner 2001]** Schwertner, Jaqueline: *Anwendbarkeit und Grenzen der Ultraschalldiagnostik (zebris® - System) und einer Laufband-Belastung zur Ermittlung von Bewegungsmöglichkeiten und –problemen bei Schlaganfallpatienten;* In: Leipziger Sportwissenschaftliche Beiträge (LSB) Sankt Augustin, 42/2001 1: S.159-167 (2001)
- [ScienCentral 1999]** ScienCentral Inc and The American Institute of Physics: *Gordon Moore* (1999)
<http://www.pbs.org/transistor/album1/moore/>
 Zugriff: 28.07.2004
- [Sederburg & Parry 1986]** Sederburg, Thomas W.; Parry, Scott R.: *Free-Form Deformation of Solid Geometric Models;* In: Computer Graphics – Proceedings of the ACM SIGGRAPH'86; Vol. 20, Issue 4; S.151-159 (August 1986)
- [Semancik 2003]** Semancik, Jaroslav: *Motion Capturing and Retargeting* (2003)
<http://cgg.ms.mff.cuni.cz/~semancik/data/motion.pdf>
 Zugriff: 21.6.2004
- [SGI 2004]** SGI: *Volumizer* (2004)
<http://www.sgi.com/software/volumizer>
 Zugriff: 20.09.2004
- [Shay & Duncan 1993]** Shay, Don; Duncan, Jody: *The making of Jurassic Park;* New York, Ballantine Books (1993)

- [Sheepers et al. 1997]** Scheepers, Ferdi; Carlson, Wayne E.; Parent, Richard E.; May, Stephen: *Anatomy-Based Modelling of the Human Musculature*; In: Proceedings of the ACM SIGGRAPH'97; S.163-172. (1997)
Auch Online unter:
<http://accad.osu.edu/~smay/Human/human.pdf>
Zugriff: 20.09.2004
- [Sheldrake 1988]** Sheldrake, Rupert: *The Precence of the Past – Morphic Resonance and the Habits of Nature*; Inner Traditions International, London (1988)
- [Shen & Thalmann 1995]** Shen, Jihanhua; Thalmann, Daniel: *Interactive Shape Design Using Metaballs and Splines*; In: Eurographics Implicit Surfaces Proceedings 1995, Grenoble, Frankreich; Vol. 5; Issue 1; S.40-45 (1995)
- [Shepter 2004]** Shepter, Joe: *Animation Arena – Sculpting the Hulk*
<http://www.animationarena.com/3d-modeling.html>
Zugriff: 17.08.2004
- [Shoemake 1985]** Shoemake, Ken: *Animating rotation with quaternion curves*; In: Computer Graphics – Proceedings of the ACM SIGGRAPH 85; ACM Press, New York; Vol.19; Issue 3, S.245 – 254 (Juli 1985)
- [Siebert & Ju 2000]** Siebert, Paul; Ju, Xiangyang: *Realistic Human Animation Using Scanned Data*; International Conference of Numberisation 3D – Scanning 2000, 24-25 Mai 2000, Paris, Frankreich (2000)
Auch Online unter:
<http://www.dcs.gla.ac.uk/publications/PAPERS/6793/scan2000b.pdf>
Zugriff: 17.08.2004

- [Siggraph 2004]** ILMfan: *Photo of Dinosaur Input Device*; History of Animation – Special Session at Siggraph 2000
http://www.ilmfan.com/articles/inc/photo_display.php?year=2002&article=siggraph_primer&image=s2000_did.jpg
Zugriff: 01.07.2004
- [Silberman2003]** Silberman, Steve: *Matrix²*; In: Wired: Issue 11/5; S. 22-29; Condé Nast Publications Inc. (11.05.2003)
Auch Online unter:
<http://www.wired.com/wired/archive/11.05/matrix2.html?pg=1>
Zugriff: 17.08.2004
- [Silva 1998]** Silva, Fernando W.: *Um Sistema de Animação Baseado em Movimento Capturado (A Motion Capture Based Animation System)*. M.Sc. Thesis, COPPE/Sistemas, Federal University of Rio de Janeiro (UFRJ) Thesis advisor: Luiz Velho, IMPA (März 1998)
- [Silva 1999]** Silva, Fernando W.; Velho, Luis: *Um Sistema de Animação Baseado em Movimento Capturado*; In: Proceedings of SBC'99 – Conference of the Brazilian Computer Society; Vol. 1/1999; S.403-406; Rio de Janeiro, Brazil (July 1999)
- [Skifi 2004]** Skifi2K: *Stop Motion Photography*
http://www.skifi2k.com/misc_html/articles/stop_motion.html
Zugriff 12.08.2004
- [Snellenburg 2004]** Snellenburg, Jonathan: *An Early Walnut cased "Millionaire" Calculator*
http://www.snellenburg.com/gallery2/SciInstr/pgs_instrum/millionaire/millionaire136.htm
Zugriff: 25.03.2004

- [Snively & Russell 2002]** Snively, Eric; Russell, Anthony P.: *The Tyrannosaurid Metatarsus: Bone Strain and Inferred Ligament Function*; Senckenbergiana lethaea; Vol. 82, Issue 1, S. 35-42 (2002)
- [Solutionix 2004]** Solutionix : *3D-Scanning*
<http://www.solutionix.com/solutionix.asp>
 Zugriff: 23.08.2004
- [Sowi 2000]** Henry Sowizral: *Scene graphs in the new millennium*;
 In: IEEE Computer Graphics and Applications, Vol. 20, Issue 1, Jan.-Feb. 2000; S.56 -57 (2000)
- [Spielberg 2004]** Steven Spielbergs Directory: *Jurassic Park*
<http://www.scruffles.net/spielberg/movies/jurassicPark.html>
 Zugriff 12.08.2004
- [Steinbichler 2004]** Steinbichler Optotechnik GmbH.: *3D Scanning*
<http://www.steinbichler.de/>
 Zugriff: 23.08.2004
- [Steiner 2002]** Steiner, Peter M.: *Anleitungen für den Bau zukünftiger „Jurassic Parks“*; Online-Zeitschrift Wissenswandel, Artikel 140 (30.01.2002)
http://www.wissenswandel.de/index.php?session=&article_id=140
 Zugriff 15.08.2004
- [Stokstad 2001]** Stokstad, Erik: *Paleontologists Learn to Shake Up Virtual Bones*; Science; Vol. 291; Issue 5508; S.1475-1476 (23.02.2001)
 Auch Online unter:
www.sciencemag.org/cgi/content/full/291/5508/1475
 Zugriff: 06.12.2004

- [Strobl et al. 2004]** Strobl, Klaus H.; Sepp, Wolfgang; Wahl, Eric; Bodenmueller, Tim; Suppa, Michael; Seara, Javier F.; Hirzinger, Gerd: *The DLR Multisensory Hand-Guided Device: The Laser Stripe Profiler*; ICRA 2004, New Orleans, U.S.A., (2004)
- [Sturman 2004]** Sturman, David: *A Brief History of Motion Capture for Computer Character Animation*; Paris, MEDIALAB (2004)
- [Terrier 2000]** Terrier, Phillippe U. A.: *High-precision satellite positioning system as a new tool to study the biomechanics of human locomotion*; In: Journal of Biomechanics 33/2000, S.1717-1722 (2000)
- [Terzopoulos 2002]** Terzopoulos, Demetri: *Artificial Life: Creatures and Environment*; In: YLEM Journal, Artists Using Sciences & Technology Nr. 12, Vol. 22., S. 78-82 (2002)
- [Terzopoulos et al. 1994]** Terzopoulos, Demetri; Tu, Xiaoyuan; Grzeszczuk, Radek: *Artificial Fishes: Autonomous Locomotion, Perception, Behavior, and Learning in a Simulated Physical World*; In: Artificial Life, Vol.1(4); S.327-351 (1994)
- [Terzopoulos und Rabie 1997]** Terzopoulos, Demetri; Rabie, Tamir F.: *Animate Vision: Active Vision in Artificial Animals*; In: Journal of Computer Vision Research, Vol. 1, Nr. 1; S.45-49; MIT Press, Videre; Cambridge, MA (1997)
- [Terzopoulos und Tu 1994]** Terzopoulos, Demetri; Tu, Xiaoyuan: *Perceptual Modeling for Behavioral Animation of Fishes*; In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques; S.43-50 (1994)
- Auch Online unter:
<http://delivery.acm.org/10.1145/200000/192170/p43-tu.pdf?key1=192170&key2=7214495901&coll=GUIDE&dl=GU>

IDE&CFID=27895358&CFTOKEN=29528828

Zugriff: 22.09.2004

[Tews 2004]

Tews Computer & NetWork Service: *GPS Technology – EGNOS*

<http://www.hr-tews.de/GPS/egnos.htm>

Zugriff: 06.08.2004

[Thalmann & Nedel 1998]

Thalmann, Daniel; Nedel, Luciana Porcher: *Real Time Muscle Deformations Using Mass-Spring Systems*; In: Proceedings of Computer Graphics International 98 (CGI'98); Hannover; S. 156-165 (Juni 1998)

Auch Online unter:

http://ligwww.epfl.ch/Publications/pdf/Nedel_Thalmann_CGI_98.pdf

Zugriff: 22.07.2004

[Thalmann 1996]

Thalmann, Daniel: *The Complexity of Simulating Virtual Humans*; In: École Polytechnique Fédérale de Lausanne – Supercomputing Review (EPFL-SCR) Nr. 8/1996, S. 16-20 (1996)

Auch Online unter:

<http://sawww.epfl.ch/SIC/SA/publications/SCR96/scr8-page16.html>

Zugriff: 07.09.2004

- [Thalmann et al. 1996]** Thalmann, Daniel; Shen, Jihanhua; Chauvineau, Eric: *Fast Realistic Human Body Deformations for Animation and VR Applications*; In: Proceedings of Computer Graphics International IEEE Computer Society Press; Juni 1996; S.166-174 (1996)
Auch Online unter:
http://ligwww.epfl.ch/Publications/pdf/Thalmann_Shen_Chauvineau_CGI_96.pdf
Zugriff: 14.09.2004
- [TheoFou 2004]** Theoretical Foundation: *Theories and Practices of Motion Analysis*
<http://www.kwon3d.com/theories.html>
Zugriff: 08.06.2004
- [Third 2003]** Third Dimension Ltd.: *3dCam-1* (2003)
<http://www.third.com/Products/3dCam-1/>
Zugriff: 23.08.2004
- [Thocp 2004]** Thocp-Federation: *The History of Computing Project*
www.thocp.net
Zugriff: 25.03.2004
- [Tomoyuki 2004]** Tomoyuki, Nishita: *A human body modelled by Metaballs*;
<http://www.eml.hiroshima-u.ac.jp/~nis/EG98/demoEG.html>
Zugriff: 07.09.2004
- [Tongva 2004]** Tribal Home of the Gabrieleno/Tongva of San Gabriel; *About the Tongva*
<http://tongva.com>
Zugriff: 18.04.2004

- [Transhumanist 2004]** Transhumanist: *Processor List*
<http://www.transhumanist.com/volume1/appendix.htm>
 Zugriff: 28.07.2004
- [Tribute 2004]** Tribute Publishing Inc.: *Jurassic Park III – Photo Gallery*
<http://www.tribute.ca/PHOTOS/1801.htm>
 Zugriff: 09.08.2004
- [Trudillo 2004]** Trudillo, Chad: *Frequently Asked Questions about Quaoar*;
 Californian Technology Intitute
<http://www.gps.caltech.edu/~chad/quaoar/>
 Zugriff: 18.04.2004
- [Tu 1996]** Tu, Xiaoyuan: *Artificial Animals for Computer Animation: Biomechanics, Locomotion, Perception, and Behaviour*;
 Springer Verlag, Heidelberg (1996)
- [Tuckborough 2004]** Tuckborough: *The Thain's Book – An encyclopedia of Middle-earth in the Third Age: Smeagold – Gollum* (2004)
<http://www.tuckborough.net/gollum.html>
 Zugriff: 09.08.2004
- [Turner & Thalmann 1993]** Turner, Russel; Thalmann, Daniel: *The Elastic Surface Model for Animated Character Construction*; In: Proceedings of Computer Graphics International'93; Lausanne, Schweiz; Springer-Verlag, Tokyo; S.399-412 (1993)
 Auch Online unter:
<http://ligwww.epfl.ch/~thalmann/papers.dir/CGI93.LEMAN.pdf>
 Zugriff: 14.09.2004

- [Uip 2004]** United International Pictures GmbH.: *Jurassic Park: Hinter den Kulissen* (2004)
http://movies.uip.de/lostworld/Lost_World02/Jurassic_Park.Site/Behind.html
 Zugriff 15.08.2004
- [Ullmann 2003]** Ullmann, Heinz: *The mathematics of balloons*; In: The European Physical Notes; Vol.1, Issue 03; S.4–6 (2003)
- [Universal 2004]** Universal Pictures: *Jurassic Park* (2004)
http://www.lost-world.com/Lost_World02/Jurassic_Park.Site/Jurassic_Park.html
 Zugriff 12.08.2004
- [Vaz & Duingan 1996]** Vaz, Mark Cotta; Duingan, Patricia Rose: *Industrial Light & Magic, Into the Digital Realm*; Del Rey, New York (1996)
- [Verdera et al. 2003]** Verdera, Joan; Caselles, Vincent; Bertalmio, Marcelo; Sapiro, Guillermo: *Impainting Surface Holes*; In: Proceedings of the IEEE International Conference on Surface Processing; Barcelona, Spanien; 14.-17. Sept. 2003; S.12-18 (2003)
 Auch Online unter:
http://www.iaa.upf.es/%7Evcaselles/papers_v/2003/MAIEEE.pdf
 Zugriff: 30.08.2004
- [Versailles 2004]** Académie de Versailles : *Les effets spéciaux*
<http://www.ac-versailles.fr/etabliss/herblay/audiovis/techniq/EFFETSPE/Effetspe.htm>
 Zugriff 15.08.2004

- [Vitana 2004]** Vitana Cooperation: *ShapeGrabber*
<http://www.shapegrabber.com/Products.shtml>
 Zugriff: 23.08.2004
- [Vitronic 2004]** Vitronic GmbH.: *VITUS 3D Ganzkörperscanner*
<http://www.vitus.de/>
 Zugriff: 30.08.2004
- [Vogel 1991]** Vogel, Klaus Peter: *Konstruktionsmorphologie: Ein Schlüssel zum Verständnis der Biologischen Evolution*; Sitzungsberichte der wissenschaftlichen Gesellschaft an der Johann Wolfgang Goethe–Universität Frankfurt am Main; Vol. 28, Nr. 2; S.1-56 (1991)
- [Vogt & Banzer 1997]** Vogt, Lutz; Banzer, Winfried: *Reproduzierbarkeit cervikaler Flexionsbewegungen der Wirbelsäule mit der 3-D-Ultraschalltopometrie*; In: Schmidtbleicher, D.; Bös, K.; Müller, A.F.; (Eds.) *Sport im Lebenslauf*; Czwalina Verlag; S.92-97 (1997)
- [Volume 2004]** Volume Graphics GmbH: *VG Studio Max*; (2004)
<http://www.volumegraphics.com>
 Zugriff: 20.09.2004
- [Waddington 1957]** Waddington, Conrad: *The Strategy of the Genes*; Allen and Unwin, London (1957)
- [Walking 2004]** Cinema: „*Jurassic Park III*” - *Production Notes*
 - Aussage von Walking
<http://www.cinema.com/article/item.phtml?ID=522&Page=8>
 Zugriff: 04.02.2004
- [Warner 2004]** Warner, John: *The History of CGI*; In: *Miami Movie Magazine* (M.M.M.) Nr. 10/2004; S.12-17 (21.09.2004)

- [Watt 2000]** Watt, Allan: *3D Computer Graphics* (3rd Edition). Addison-Wesley (2000)
- [Watt 2002]** Watt, Alan: *3D-Computergrafik*; Pearson Studium (2002)
ISBN 3-8273-7014-0
- [Westermann & Ertl 1998]** Westermann, Rüdiger; Ertl, Thomas: *Efficiently using graphics hardware in volume rendering applications*; In: Computer Graphics – Proceedings of the ACM SIGGRAPH '98; Vol. 32, Issue 4; S.169-177 (1998)
Auch Online unter:
<http://delivery.acm.org/10.1145/290000/280860/p169-westermann.pdf?key1=280860&key2=0696865901&coll=GUIDE&dl=GUIDE&CFID=27837572&CFTOKEN=8684048>
Zugriff: 20.09.2004
- [Westermann et al. 1999]** Westermann, Rüdiger; Kobbelt, Leif; Ertl, Thomas: *Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces*; In: The Visual Computer, Vol.15, Nr.2; S.100-111 (1999)
- [Wiegand 1996]** Wiegand, Tim F.: *Interactive Rendering of CSG models*; Computer Graphics Forum; Vol. 9, Nr. 3, S. 20-28 (1989)

- [Willier 2002]** Willier, Aymeric; Marque, Catherine: *Juggling gestures analysis for music control*; In: Lecture Notes in Computer Science – Vol. 2298/2002; S.296 (2002)
Title: Gesture and Sign Language in Human-Computer Interaction: International Gesture Workshop, GW 2001; London, UK; Revised Papers, Editors: I. Wachsmuth, T. Sowa (Eds.): (18-20 April 2001)
Online Date: July 2003
Publisher: Springer-Verlag Heidelberg
ISSN: 0302-9743
gefunden auf
<http://ame2.asu.edu/faculty/dab/classes/papers/juggling-gestures.pdf>
Zugriff: 04.08.2004
- [Wilson 2004]** Wilson, David: *Knowledge*; ACA Press (2004)
- [Windhorst 2000]** Windhorst, David: *The Dino King: Disney lumbers into summer with expensive, computer-generated Dinosaur* (2000)
<http://www.geocities.com/trisport/dinosaur.html>
Zugriff 15.08.2004
- [Wolf 2004]** Wolf & Beck GmbH: *3D-Ringscanner*
<http://www.wolfbeck.com/>
Zugriff: 30.08.2004
- [Wrobel & Alibadi 2002]** Wrobel, Luiz C.; Aliabadi, Mohammed H.: *The Boundary Element Method – Applications in Solid and Structures*; Vol. 2; Wiley-VCH, New-York (2002)

- [Wu et al. 2001] Wu, Xunlei; Downes, Michael; S.;Goktekin, Tolga; Tendick, Frank: *Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes*; In: EUROGRAPHICS, September 2001; Vol. 20, Nr.3, S.349-358 (2001)
- [Wulf & Wagner 2003] Wulf, Oliver; Wagner, Bernado: *Fast 3D-Scanning Methods for Laser Measurement Systems*; International Conference on Control Systems and Computer Science (CSCS14), Juli 2003, Bukarest (2003)
Auch Online unter:
<http://www.rts.uni-hannover.de/mitarbeiter/wulf/Wulf03-CSCS14.pdf>
Zugriff: 26.08.2004
- [Wyvill & Trotmann 1990] Wyvill, Geoff; Trotman, Andrew: *Ray-Tracing Soft Objects*; In: Proceedings of Computer Graphics International; Springer-Verlag; S.469-475 (1990)
- [Wyvill & Wyvill 1989] Wyvill, Brian; Wyvill, Geoff: *Field Functions for Implicit Surfaces*; In: The Visual Computer; Vol. 5, Nr.4; S.120-126 (1989)
- [Wyvill 2004] Wyvill, Brian: *Blobby*
<http://pages.cpsc.ucalgary.ca/~blob/images/blobby.gif>
Zugriff: 10.09.2004
- [Wyvill et al. 1986] Wyvill, Brian; McPheeters, Craig; Wyvill, Geoff: *Animating Soft Objects*; In: The Visual Computer; Vol. 2, Nr. 4; S.235-242 (1986)

- [Wyvill et al. 2000] Wyvill, Brian; Jepp, Pauline; van Overveld, Kees; Wyvill, Geoff: *Subdivision Surfaces for Fast Approximate Implicit Polygonization*; University of Calgary, Dept. of Computer Science, Research Report 2000-671-23 (2000)
Auch Online unter:
<http://pages.cpsc.ucalgary.ca/~blob/ps/polyg.pdf>
Zugriff: 10.09.2004
- [Wyvill et al. 2003] Wyvill, Geoff, McPhetters, Craig; Wyvill, Brian: *Data Structure for Soft Objects*; In: *The Visual Computer*; Vol. 2, Nr. 4; S. 227-234 (2003)
- [Zebris 2004] zebris Medizintechnik GmbH: *3D-Mess-Systeme*
<http://www.zebris.de/3D-mess-systeme/3d-mess-systeme.html>
Zugriff: 30.06.2004
- [Zelluloid 1993] Hoffmann, Dennis; Jachmann Alexander: *Zelluloid – Jurassic Park* (1993)
<http://www.zelluloid.de/filme/index.php3?id=846>
Zugriff 15.08.2004
- [Zlotnick 2004] Zlotnick, Joel: *An animatronic of the legendary Tyrannosaurus rex (T. rex) being built*
<http://entertainment.howstuffworks.com/animatronic1.htm>
Photo courtesy Stan Winston Studio, photographer Chuck Zlotnick
Zugriff: 10.08.2004
- [Zöllner 2004] Zöllner & Fröhlich GmbH: *Bildgebende Lasermesssysteme (3D)*
http://www.zf-laser.com/d_bildgebende.html
Zugriff: 26.08.2004

[Zowack 2002]

Zowack, Martina: *Compaq Technologie für ICE AGE den neuesten Animationsfilm "ICE AGE – Der Film von Twentieth Century Fox"*; Presstext Austria; Wien (29.03.2002)

Auch Online:

<http://www.presetext.at/pte.mc?pte=020329002>

Zugriff 15.08.2004

9 Anhang

9.1 Zusatzinformationen zum Inhalt

9.1.1 Herleitung des Namens "Quaoaring"

Der Name „Quaoaring“ stammt von der allgegenwärtigen Schöpfungskraft „Quaoar“ des kalifornischen Tongva-Stammes, welcher im Becken von Los Angeles beheimatet war. [Tongva 2004] Für die Aussprache von Quaoar wurden drei verschiedene Aussprachevarianten ([kwa:wa:r] [kwa:owa:] und [q'wa:wa:]) gefunden. [Nasa 2002] [Quaoar 2004] Aus ästhetischen Gründen wurde die letzte Variante gewählt, so dass die Technologie als Ganzes [q'wa:wa:riŋ] ausgesprochen wird. Mark Acuña, ein Schüler, Tänzer und Stammesältester der Tongva berichtet über Quaoar: [Trudillo 2004]

“Quaoar the great force of creation sings and dances the high ones (Deities) into existence. While Quaoar has no form or gender he is usually referred to with the male pronoun. He dances and sings first ‘Weywot’ who becomes Sky Father; they sing and dance ‘Chehooit’ Earth Mother into existence. The trio sing ‘Tमित’ Grandfather Sun to life. As each divine one joins the singing and dancing, the song becomes more complex and the dance more complicated. In turn ‘Moar’, Grandmother Moon (a very complex deity), ‘Pमित’ the Goddess of the sea, ‘Manit’, the Lord of dreams and visions, ‘Manisar’ the bringer of food and harvests, ‘Tukupar Itar’ Sky Coyote (who is also our major hero), ‘Tolmalok’, the Goddess of Shishongna (the underworld) join in the singing, dancing and creating. And finally the great seven giants who hold up the worlds are created. The High Ones in turn are aided by ‘Eagle, Duck, Bear, and Frog’ in a grand earth diving story. Frog brings up soil out of the deep dark sea, and the four animals dance it flat and wide. The ‘Gods and Goddesses’ then furnish the world ‘Tovangar’ with hills, mountains, trees, rivers, etc. ‘Tobohar’ (first man) and ‘Pahavit’ (first woman) are also part of this great ‘Creation song and dance cycle’.”

Nach einer anderen Legende kam Quaoar vom Himmel und, nachdem er das Chaos reduziert und die Welt erschaffen hatte, erzeugte er zuerst die niederen Tiere und danach die höheren Tiere sowie den Menschen. [Nasa 2002]

Der Name „Quaoaring“ mag zuerst etwas ungewohnt und seltsam für europäische Ohren klingen, er ist jedoch sehr passend, da sehr viele Parallelen zwischen der alten Mythologie und der neuen Technologie existieren:

- Quaoar ist eine Kraft, die fast spielerisch die Organismen erschafft. Auch mit der Quaoaringtechnologie können spielerisch-einfach virtuelle Organismen generiert werden.
- Quaoar erzeugt in einer komplizierten aber logischen Abfolge zuerst die niederen, und erst dann die höheren Tiere sowie den Menschen. Die Mythologie von Quaoar ist daher diejenige Stammesmythologie, die noch am ehesten mit dem modernen Evolutionsgedanken in Einklang zu bringen ist.
- Die Schöpfung geschieht sowohl der in der indianischen Mythologie als auch in der Frankfurter Organismus- und Evolutionstheorie zwangsläufig und sie ist irreversibel.
- Quaoar tanzt im Kreis um einen Totempfahl, um die Organismen zu erschaffen. Dies hat – wie später gezeigt wird – Ähnlichkeit mit dem zirkulären Skin-Mesh-Herstellungsverfahren der Quaoaringtechnologie, in der gleichsam auch um eine Zentrallinie „getanzt“ wird, um die Position der Vertices zu ermitteln.

9.1.2 Tabelle für 3D-Scanning

Firma	Land (Stadt)	Produkte	Techno- logien	Applika- tionen	Far- be	Quelle
3D Digital Corp.	USA	Model 100,200,300	Laserstreifen/ CCD	Face, Close	optio- nal	[3DDC 2004]
3D Scanners	GB (London)	Model Maker, Reversa, Replica	Laserstreifen/ CCD, Manuell	Close, Mid	optio- nal	[3D- scanners 2004]
3D Shape GmbH	D (Erlangen)	CAM ^{3D} , SCAN ^{3D}	Laserstreifen/ CCD	Face, Head/Limb, Whole Body	nein	[3Dshape 2004]
ABW GmbH.	D (Fricken- hausen)	KOMBI-640	Graycode/ CCD + Phasenschift	Head/Limb, Whole Body, Mid	ja	[ABW 2003]
Arius3D	Kanada	Arius3D	Laserstreifen/ Double aperture mask	Head/Limb, Close, Mid, Near	ja	[Arius 3D]
Breuckmann GmbH	D (Merseburg)	optoTOP, optoSIS, dermaTOP, mikroTOP, faceSCAN, triTOS	MPT-Technik	Derma, Face, Close, Mikro, Surface, Head/Limb	optio- nal	[Brück- mann 2004]
CGI	USA (Minea- polis)	CSS-100	Automatic Cross- sectional Scanning	Close	nein	[CGI 2004]
CogniTens 3D Vision Systems	Israel	Optigo200, OptiCell	Weißlicht.- punkt- projektion	Close, Mid	ja	[CogniTens 2004]
Cyberware	USA (Monterey, CA)	Face, Desktop, Whole Body	Laserstreifen/ CCD	Face, Head/Limb, WholeBody, Mid	nein	[Cyberware 2004]
Digibotics,Inc	USA (Addison, Texas)	Digibot II, Digibot IIHD, Digibot III	Laserstreifen/ CCD	Close	ja	[Digibotics 2004]

DLR	D	Laser Range Scanner	Laserpunkt/ PSD	Close, Mid	k.A.	[DLR 2004a]
Eyetrionics	Belgien (Heverlee)	Shape Snatcher, Bscan-S, ShapeCam	Weißlicht-Punktprojektion	Close, Mid, Face, Head/Limb, WholeBody	nein	[Eyetrionics 2004]
Faro Technology	USA (Lake Mary, Florida)	FaroArm	Articulation: sechs Freiheitsgrade	Close, Mid	ja	[Faro 2004]
GOM mbH	D (Braunschweig)	ATOS I, ATOS II, ATOS III, ATOS SO, TriTop	Weißlicht-Projektion (Punkt oder Streifen)	Mikro, Close, Surface, Near, Mid	ja	[GOM 2004]
Hitchcock-Manthey LLC.	USA (New York)	ASAME	Silhouette / Turn Table	Surface	nein	[Hitchcock 2004]
Immersion Corp.	USA (San Jose, CA)	MicroScribe 3D	Artikulation / Stylus tip (Punkt)	Close, Near	ja (manuell)	[Immersion 2004]
InSpeck, Inc	Kanada (Montreal)	Capturor, Capturor DF, Capturor II, Mega Capturor II	Moiré	Head/Limb	nein	[InSpeck 2004]
Iqsun GmbH.	D (Ludwigsburg)	Iqsun 880, Iqsun 625	Pulslaufzeit/ Laser	Mid, Landscape	k.A.	[Iqsun 2004]
Kréon Industries	F (Limoges)	KLS 51, KLS 171, Zephyr	Laserstreifen/ CCD	Close, Near	ja	[Kréon 2004]
Laser Design Inc.	USA (Minneapolis)	Surveyor	Laserstreifen/ CCD	Close, Near	ja	[Laser-design 2004a]
MetricVison	USA (Newington, VA)	MV200	Pulslaufzeitverfahren/ Radar-Laser	Close	ja	[Metric 2004]
Minolta Corp	Japan	VIVID300, VIVID900	Laserstreifen/ CCD	Close, Near	nein	[Konica 2004]

Nub3d	Spanien (Barcelona)	triple, voxes	Weißlicht- projektion	Close, Near, Surface	ja	[Nub3D 2004]
Optimet	Israel (Jerusalem)	ConoProbe ConoLine, Micro ConoProbe	kono- skopische Holographie	Mikro, Close, Surface	k.A.	[Optimet 2004]
Opton	Japan (Anada)	Handy, Mini-Moire	Moiré	Close	optio- nal	[Opton 2004]
Perceptron Inc.	USA (Plymouth, MI)	ScanWorks	Laserstreifen/ CCD	Close	ja	[Perceptron 2004]
Polhemus	USA (Colches- ter, VT)	FastScan™ Cobra™	Laserstreifen/ CCD	Close, Face, Head/Limb (Handgerät)	ja	[Polhelmus 2004]
Riegl GmbH.	Österreich (Horn)	LMS-Z420i, LMS-Z360i, LMS-Z210i, LPM-25HA, LPM800HA LPM-2K LPM-VHS	Pulslaufzeit- verfahren/ Radar-Laser	Near, Mid, Landscape	ja	[Riegl 2004]
Renishaw	GB (New Mills)	Cyclone	Laserpunkt/ CCD	Close	ja	[Renishaw 2004]
Roland Digital Group	USA (Los Angeles, CA)	Picza PIX4, Picza PIX30, LPX-250, LPX-1200, MDX-15, MDX-20	Stylus tip (Punkt) / Laserpunkt	Close, Surface	ja	[Roland 2001]
Romer Inc.	USA (Farming- ton Hills, MI)	Romer 1000i, 3000i™, LSI, Stinger II™, 3000iSC, TooLOK™, GridLOK™	Artikulation, sechs Freiheitsgrade	Close	ja	[Romer 2002]
RSI GmbH	D (Oberursel,	DigiScan20 00,	Weißlicht- projektion	Close, Near, Head/Limb,	nein	[RSI 2004]

	Hessen)	3D Body Builder, 3D 1500		Whole Body		
Solutionix	Südkorea (Seoul)	SNX	Weißlichtprojektion	Close, Near	optional	[Solutionix 2004]
Steinbichler	D (Altenbeuern, Bayern)	COMET , T-SCAN	Weißlicht-Projektion & Laserstreifen/ CCD	Close, Near, Mid	ja	[Steinbichler 2004]
Third Dimension Software	GB (Bristol)	Gap-Gun, 3dCam-1	Farbcodiertes strukturiertes Licht & Laserstreifen/ CCD	Close, (Handgerät) Near	ja	[Third 2003]
Vitana Corporation	Kanada (Ottawa)	Shapegrabber	Laserstreifen/ CCD	Close, Near, Surface	ja	[Vitana 2004]
Vitronic	D (Wiesbaden)	PEDUS, VITUS pro, ahead, smart	Laserstreifen/ CCD	Head/Limb, Whole Body	ja	[Vitronic 2004]
Wolf & Beck GmbH.	D (Wangen)	OTP, OTM, ORS	rotationssymmetrisches Laserpunkt/ CCD	Close, Near, Surface	Ja	[Wolf 2004]
Zöller & Fröhlich GmbH.	D	IMAGER 5300	Pulslaufzeitverfahren / Laser	Mid, Near, Landscape	ja	[Zöller 2004]

Tabelle 19: Herstellerfirmen von 3D-Scannern

Übersichtstabelle für 3D-Scanning mit internationalen Herstellerfirmen und ihren jeweiligen Angeboten: Erste Spalte: Firmenname; Zweite Spalte: Land und – falls bekannt in Klammern – Ort des Firmenhauptsitzes; Dritte Spalte: Angebotene 3D-Scanner der betreffenden Firma; Vierte Spalte: Verwendete 3D-Scanning-Verfahren der jeweiligen Produkte; Fünfte Spalte Einsatzgebiete der 3D-Scanner; Sechste Spalte: Farboption (ja, nein, k.A., optional); Siebte Spalte: Quellenreferenz; Die URLs der Firmen stehen im Literaturverzeichnis bei der jeweiligen Quellenangabe.

Die Tabelle erhebt keinen Anspruch auf Vollständigkeit. Es wird jedoch aufgrund der Quantität der Firmen und der qualitativ sehr unterschiedlichen Verfahren ersichtlich, dass sich der Markt für 3D-Scanning noch nicht gesättigt hat und sich immer noch in einer Umbruchphase befindet.

9.1.3 Vergleich von Szenegraphen

9.1.3.1 Grundsätzliche Eigenschaften und Kompatibilitäten

	Features	Real- Mation	Open SG	NeMo	Java3D	Open GVS	Per- former
Betriebs- systeme	Irix	(+) ¹	+	-	+	+	+
	Unix (& Derivate)	-	-	-	+	-	-
	Linux	-	+	~	+	+	-
	Mac-OS	-	-	~	-	-	-
	Windows NT	+	(+) ²	+	+	+	-
	Windows CT	+	-	-	-	-	-
	Windows 95/ 98/ 2000/ ME/ XP	+	-	+	+	+	-
Basis-APIs	Direct 3D	+	-	+	+	+	-
	Direct Full Screen	+	-	+	-	+	-
	3Dfx Glide	+	-	+	-	+	-
	OpenGL	+	+	+	+	+	+
	Merlin	+	-	-	-	-	-
	Real 3D Pro	+	-	-	-	-	-
	Number 9 Imagine 128	+	-	-	-	-	-
	Hitachi Spherics	+	-	-	-	-	-
	Glint	+	-	-	-	-	-
	RenderWare	+	-	+	-	-	-
Null-Renderer	+	-	-	-	-	-	
Programming	objektorientiert	(-) ³	+	+	+	+	+

¹ Mit dem RealMation SDK geschriebene Programme können zwar bis zur Version 4.5 auch auf IRIX portiert werden, viele Spezialfunktionen wie Import/Export als auch die File-Handling-Funktionen funktionieren allerdings nur unter den Windows-Derivaten

² OpenSG ist vornehmlich für IRIX design, läuft bis jetzt nur theoretisch auf Windows NT, wird auch zukünftig nicht für Windows NT optimiert werden

³ Geschwindigkeitsrelevante Codeteile sind prozedural geschrieben, häufige Verwendung von Makros

	multithreading-capability	+	+	-	+		+
	modelling functions	+	+	+	+		(+)

Tabelle 20: Vergleich der Eigenschaften/Kompabilitäten von sechs Szenengraphen

Legende: + realisiert; (+) eingeschränkt realisiert; - nicht realisiert; (-) von einigen Ausnahmen abgesehen nicht realisiert; ~ anvisiert aber nicht realisiert; (~) möglicherweise anvisiert; ? keine Informationen

9.1.3.2 Unterstützte Objekte

	Features	Reali- Mation	Open SG	NeMo	Java 3D	Open JVS	Per- former
unterstützte hierarchische Objekte	machines	+	-	-	-	-	-
	channels/ windows	+	(+)	-	-	(+) ⁴	+
	views/ scenes/ worlds	+	+	+	+	+	+
	instances (transformation)	+	+	+	+	+	+
	instances (scaling)	+	+	+	+	+	+
	instances (billboarding)	+	+	+	+	+	-
	instances (switching)	+	+	+	+	-	+
	instances (rendering switches)	+	-	-	-	-	+
	instances (z-buffer manipulations)	+	-	-	-	-	+
	instances (LODs)	+	~	~	+	+	+
	geometries	+	+	+	+	+	+
	environments/ atmospherics	+	+	+	+	+	+
	lights (directional)	+	+	+	+	+	+
lights (point)	+	~	+	+	+	+	

⁴ OpenGVS trifft keine Unterscheidung zwischen Channels/Windows und Viewports

lights (spot)	+	~	+	+	+	+
lights (ambient)	-	-	+	+	+	+
cameras	+	(+)	+	+	-	+
materials	+	+	+	+	-	-
textures	+	~	-	+	+	-
images	+	~	+	+	-	+
groups	+	+	+	+	-	-
cells	+	-	-	-	-	-
portals	+	-	-	-	-	-
links	+	-	+	-	-	-
actions (interactive behaviours)	-	-	+	+	-	-
actions (movements)	+	-	+	+	-	-
actions (shapescaling)	+	-	+	-	-	-
actions (morphing)	-	-	+	+	-	-
actions (material changings)	+	-	+	+	-	-
actions (texture changings)	+	-	+	+	-	-
actions (switches)	+	-	+	+	-	-

Tabelle 21: Unterstützte Objekte von sechs Szenegraphen

Legende: + realisiert; (+) eingeschränkt realisiert; - nicht realisiert; (-) von einigen Ausnahmen abgesehen nicht realisiert; ~ anvisiert aber nicht realisiert; (~) möglicherweise anvisiert; ? keine Informationen

9.1.3.3 Features der unterstützten Objekte

	Features	Reali- Mation	Open SG	NeMo	Java3D	Open GVS	Performer
Material/ Textur/ Image- Features	perspective	+	~	+	+	+	+
	texture shading						
	per vertex lighting	-	-	-	-	+	-
	diffuse force	+	-	+	+	+	+
	ambient force	+	-	+	+	+	+
	emissive force	+	-	+	+	+	+
	specular force	+	-	+	+	+	+
	shininess force	+	-	+	+	+	+
	diffuse color	+	+	+	+	+	+
	ambient color	+	+	+	+	+	+
	emissive color	+	~	+	+	+	+
	specular color	+	~	+	+	+	-
	transparency	+	~	+	(+)	+	+
	detail mapping	+	~	+	+	+	+
	bump mapping	~	-	+	-	-	-
	multitexturing	+	~	-	+	-	+
	texture combiners	+	~	-	?	-	+
	alpha textures	+	~	+	+	+	+
	blending	+	~	+	-	+	+
	texture antialiasing	+	-	+	+	-	+
mipmapping	+	-	-	+	+	+	
mipmap blending	+	-	-	-	+	+	
optional alphakey	+	~	+	-	-	-	
texture clipping	+	~	-	+	-	-	
videotextures	(+) ⁵	-	+	-	-	?	

⁵ RealiMation bietet Programmierern eine Hilfestellung an, um selbst Videotexturen zu programmieren.

	background- texture	+	-	+	+	-	-
	texture compression	+	-	-	-	-	-
	dynamic texture loading	+	-	-	?	-	+
	procedural textures	-	-	-	-	-	-
	environmental mapping	-	-	+	-	-	-
	metallic surface imitation	-	-	-	-	-	-
	reflection tracing	-	?	-	-	-	-
	Geometry Features	progressive meshes	+	~	-	-	-
ASD mit morphing		-	-	-	-	-	+
LODs		+	~	-	+	+	+
strips/fans		+	~	-	+	-	+
morphing		-	~	(+)	+	-	-
table compression		+	-	-	-	-	-
deformable meshes		-	~	+	-	-	-
vertex normals		+	+	+	+	+	+
plane normals	+	-	-	+	-	+	
Light Features	multiple lights	+	~	+	+	+	+
	coloured lights	+	~	+	+	-	+
	negative lights	+	-	-	+	-	-
	spotlights	+	~	+	+	+	+
	point lights	+	~	+	+	+	+
	parallel lights	+	+	+	+	+	+
	ambient lights	-	-	+	+	+	+
	volume light	-	-	-	-	-	-

	lens flare effects	-	-	+	-	-	-
	coronas	-	-	-	-	-	-
	realtime shadow casting	-	+	-	-	-	-
Atmospheric Features	volumetric fog	+	~	+	-	+	+
	coloured fog	+	~	+	-	+	+
Rendering Features	frustrum culling	+	+	+	+	+	+
	backface culling	+	+	+	+	+	+
	cloud frame	+	+	-	+	+	-
	wireframe	+	+	+	+	+	+
	flat shading	+	+	+	+	+	+
	gourauld shading	+	+	+	+	+	+
	phong shading	-	-	-	-	-	-
	lambert shading	-	-	-	-	-	-
	HW antialiasing	+	-	-	+	-	+
	y-inversion	-	-	-	-	-	-
	dynamic video resolution	-	-	-	-	-	+
Andere Features	mirrors	-	-	+	-	-	-
	particle system	-	-	-	-	-	-
	collision detection	(+)	~	+	(+)	+	+
	physical integration	-	-	-	-	-	-
	sound integration	-	-	+	+	-	-
	motion blur	-	-	-	-	-	-
	3D text	(+)	+	-	+	-	-
	2D overlay functions	+	-	(+)	+	-	-
stereoscopic capability.	+	~	+	+	+	-	

Tabelle 22: Object Features von sechs Szenegraphen

Legende: + realisiert; (+) eingeschränkt realisiert; - nicht realisiert; (-) von einigen Ausnahmen abgesehen nicht realisiert; ~ anvisiert aber nicht realisiert; (~) möglicherweise anvisiert; ? keine Informationen

9.1.3.4 Angebotene I/O-Schnittstellen

	Features	RealiMation	Open SG	NeMo	Java3D	Open GVS	Performer
unterstützte Texturformate	tga	+	+	+	-	+	+
	jpg	+	-	+	+	+	-
	gif	-	-	-	+	+	-
	bmp	+	-	+	-	+	-
	pcx	+	-	+	-	-	-
	tif	-	-	+	-	+	-
	dip	-	-	+	-	-	-
	rle	-	-	-	-	+	-
	rgb	+	+	-	-	+	+
	rgba	+	~	-	-	+	+
	int	+	~	-	-	+	+
inta	+	-	-	-	+	+	
Input Plug-In- Schnittstellen zu anderen 3D- Programmen	3D-Studio Max	+	-	+	(+)	-	-
	Softimage	+	-	+	-	-	-
	AutoCAD	+	-	-	-	-	-
	MicroStation	+	-	-	-	-	-
	Bryce	-	-	-	-	-	-
Output Plugin Schnittstellen zu anderen 3D- Programmen	3D-Studio Max	+	-	-	-	-	-
	Softimage	+	-	-	-	-	-
	AutoCAD	+	-	-	-	-	-
	MicroStation	-	-	-	-	-	-
	Bryce	-	-	-	-	-	-
unterstützte 3D-Input Formate	dwb (Centric)	+	-	-	-	-	+
	3ds (3D Studio)	+	-	+	-	+	-
	obj (WaveFront)	+	-	-	-	-	+
	dxf (AutoCAD)	+	-	-	-	-	+
	iv (Inventor)	-	-	-	-	-	+

	rbs (RealiBase)	+	-	-	-	-	-
	sv (Superviewer)	-	-	-	-	-	+
	x (Direct3D)	+	-	-	-	-	-
	lwo (LightWave)	-	-	-	-	-	-
	raw	+	-	+	-	-	-
	VRML1	+	+	+	-	-	-
	wrl (VRML2)	+	+	+	-	-	-
	flt (MultiGen)	~	-	-	-	+	-
unterstützte 3D-Output Formate	dwb (Centric)	-	-	-	-	-	+
	3ds (3D Studio)	+	-	-	-	-	-
	obj (WaveFront)	+	-	-	-	-	+
	dxf (AutoCAD)	+	-	-	-	-	+
	iv (Inventor)	-	-	-	-	-	+
	rbs (RealiBase)	+	-	-	-	-	-
	sv (Superviewer)	-	-	-	-	-	+
	lwo (LightWave)	-	-	-	-	-	-
	raw	+	-	+	-	-	-
	VRML1	+	+	+	-	-	-
	wrl (VRML2)	+	+	+	-	-	-
	flt (MultiGen)	~	-	-	-	-	-
option files	sko	-	-	+	-	-	-
	opr	-	-	+	-	-	-
unterstützte Output Formate der gerenderten Scene	avi	+	-	+	-	-	-
	wrl	+	-	-	-	-	+
	raf	+	-	-	-	-	-
	tga	+	~	+	-	-	+
	jpg	+	~	+	-	-	-
	gif	-	-	-	-	-	-
	tif	-	-	+	-	-	-
	bmp	+	-	+	-	-	-
	pcx	-	-	+	-	-	-
	rgb	-	-	-	-	-	+
rgba	-	-	-	-	-	+	

	int	-	~	-	-	-	+
	inta	-	-	-	-	-	+
	xlg	-	-	-	-	-	-

Tabelle 23: Angebotene I/O-Schnittstellen von sechs Szenegraphen

Legende: + realisiert; (+) eingeschränkt realisiert; - nicht realisiert; (-) von einigen Ausnahmen abgesehen nicht realisiert; ~ anvisiert aber nicht realisiert; (~) möglicherweise anvisiert; ? keine Informationen

9.1.3.5 Netzwerkfähigkeit

	Features	RealiMation	Open SG	NeMo	Java3D	Performer
LAN	TCP/IP capabilities	+	-	-	+	-
	distributed rendering im lokalen Netzwerk	+	-	-	-	-
	frei definierbare LAN messages	+	-	-	-	-
	LAN error handling	+	-	-	-	-
Internet	Netscape Viewer	+	-	+	?	-
	Explorer Viewer	+	-	+	?	-
	object related hyperlinks	+	-	+	-	-
	interactive link behaviours	-	-	+	-	-

Tabelle 24: Netzwerkfähigkeit von sechs Szenegraphen

Legende: + realisiert; (+) eingeschränkt realisiert; - nicht realisiert; (-) von einigen Ausnahmen abgesehen nicht realisiert; ~ anvisiert aber nicht realisiert; (~) möglicherweise anvisiert; ? keine Informationen

9.2 Vorarbeiten des Verfassers

9.2.1 Schriftenverzeichnis

9.2.1.1 Wissenschaftliche Proceedings

- Breiner, Tobias; Dörner, Ralf; Seiler, Christian; Gudo, Michael: *Visualizing Organisms with Hydraulic Body Parts: A Case Study in Integrating Simulation and Visualization Models*; In: Joint IEEE - EUROGRAPHICS Symposium on Visualization 2004; S.97-102 (2004)
- Gudo, Michael; Syed, Tareq; Breiner, Tobias: *e-Voluzzer: Hydropneumatic biosimulation for computer-based modeling and simulation of evolutionary transitions*; In: Tagungsbandfolien der 97. Jahresversammlung der Deutschen Zoologischen Gesellschaft (2004)
- Breiner, Tobias: *A generalized View of Moore's Law and its Implications on Future Technology*; In: SIPS Proceedings: 1/03; S.3-4 (2003)
- Breiner, Tobias: *Visualisation of Animals in the Movie Industry*; In: SIPS Proceedings; 1/02; S.2-3 (2002)
- Breiner, Tobias: *Konzeptioneller und visueller Aufbau von Autoren- und Lernprogrammen in der Biochemie*; Hand-Out-Proceedings zum Fraunhofer-Seminar "Chancen und Perspektiven modernen Visualisierungstechnologien in Medizin und Pharmazie" – Medica, Düsseldorf, 23.11.2001 (2002)
- Breiner, Tobias: *Freie Konzeption zukünftiger Fahrsimulationen* In: Möller, Reinhard 7. Workshop Sichtsysteme – Visualisierung in der Simulationstechnik: Aachen: Shaker, S. 15-25 (2001)

9.2.1.2 Populärwissenschaftliche Publikationen

- Breiner, Tobias: *Fahrsimulationen – Vergangenheit, Gegenwart und Zukunft*, In: Forumsbericht Interaktive Visuelle Kommunikation des Cybernariums, S.1-2 (2004)
- Breiner, Tobias: *Die herannahende Wirtschaftskrise in der ersten Dekade des 21. Jahrhunderts – Computerprognose als Menetekel*; In: Capital Growth Special; 2.10.1999; S.21-24; Darmstadt (1999)
- Breiner, Tobias: *Formation of Information*; In: G.M. international magazine; November-Dezember 1998, S.56-58; Amsterdam, Niederlande (1998)
- Breiner, Tobias; Jansen, Nicole: *Zensur im Internet*; G.M. Magazin - Deutschland; 3/97; S.12-13; Köln (1997)
- Breiner, Tobias: *Die allgemeine Antigravitationstheorie*; In: Oxmorox; Oktober 1991; Stuttgart (1991)

9.2.1.3 Interne Schriften

- Breiner, Tobias: *User Manual für das e-Go Framework*; Johann Wolfgang Goethe-Universität Frankfurt am Main; 157 Seiten (2005)
- Breiner, Tobias: *Namenskonventionen für Konstanten, Variablen, Methoden und Klassen in C++, Java und darauf aufbauende Szenegraphen*; In: GDV-Papiere der Johann Wolfgang Goethe-Universität Frankfurt am Main, S.1-17 (2004)
- Breiner, Tobias: *„Openscenegraph“, „openSG“, „RealiMation“, „VirTools“ und „Java3D“ im kurzen Vergleich*; Johann Wolfgang Goethe-Universität Frankfurt am Main; 21 Seiten (2004)
- Breiner, Tobias: *Interface Structure of the 3DG SUN Framework*; In: 3D-Generation-Papiere, Darmstadt (2003)

- Breiner, Tobias: *Klassen und Methoden des Szenengraphen „Holotek“*; In: GDV-Papiere der Johann Wolfgang Goethe-Universität Frankfurt am Main; (2003)
- Breiner, Tobias: *Erweiterte Schnittstellendefinitionen für Motion DLLs*; In: GDV-Papiere der Johann Wolfgang Goethe-Universität Frankfurt am Main; (2003)
- Breiner, Tobias: *Bedienungsanleitung 3DG Quaoar*; Johann Wolfgang Goethe-Universität Frankfurt am Main; 7 Seiten (2002)
- Breiner, Tobias: *Erstellung von biochemischen Lernmodulen* In: k-MED Autorenhandbuch; Sept. 2002; S.170-199 (2002)
- Breiner, Tobias; Baier, Wolfgang: *Erstellung des Workflows zur Modulerstellung*; In: k-MED Autorenhandbuch; Sept. 2002; S.120-125 (2002)
- Breiner, Tobias; Mümpfer, Michaela: *Kurzübersicht der Evaluationsergebnisse verschiedener Formeleditoren unter besonderer Berücksichtigung von „Accent“, „MathTyp“, „FormelBaska“ und „MS Word Formel Editor“*; In: k-MED Community Papers; 2002; 10 Seiten (2002)
- Seiler, Christian; Breiner, Tobias; Krömker, Detlef: *Real 3D-2D-Displays – Räumliche Darstellung mit neuer Technologie*; Informationsbroschüre für das gleichnamige Best-Practice-Seminar am 16.Mai 2002 im Fraunhofer Anwendungszentrum Computergraphik in Chemie und Pharmazie (AGC) in Frankfurt am Main; 8 Seiten (2002)
- Breiner, Tobias: *Oberflächengestaltung in k-MED – Knowledge-Based Multimedia Medicine Education*; “; In: k-MED Community Papers; 3 Seiten (2002)
- Breiner, Tobias: *Ergebnisse der Evaluation von 30 Szenegraphen*; Johann Wolfgang Goethe Universität Frankfurt am Main; 73 Seiten (2002)

- Breiner, Tobias; Seiler, Christian; Krömker, Detlef: *Visualisierung in den Life Sciences – Computergraphik zur Visualisierung von Vorgängen und Verfahren in Biologie, Chemie, Pharmazie und Medizin*; Informationsbroschüre für das gleichnamige Best-Practice-Seminar am 25. September 2001 im Fraunhofer Anwendungszentrum Computergraphik in Chemie und Pharmazie (AGC) in Frankfurt am Main; 8 Seiten (2001)
- Breiner, Tobias: *VR-gestützte Wissensmanagement-Plattform für die holistische Analyse regulatorischer und metabolischer Netzwerke*; Fraunhofer Anwendungszentrum Computergraphik in Chemie und Pharmazie (AGC) in Frankfurt am Main; 9 Seiten (2000)
- Breiner, Tobias: *Softwaremodule für das TDL-Framework*; In: GDV-Papiere der Johann Wolfgang Goethe Universität Frankfurt am Main (2000)

9.2.1.4 Nichtwissenschaftliche Artikel

- Breiner, Tobias: *Zwischen Manipulation und Marketing*; 3D-Holotek, S.9-12 Darmstadt (2003)
- Breiner, Tobias: *Kleiner Stilkunde-Leitfaden*; 3D-Holotek, S.12-14; Darmstadt (2000)
- Breiner, Tobias: *Spray or Pray!*; grow music magazine; Nr. 4/98; S.10-16; Darmstadt (1998)
- Breiner, Tobias: *Der Bio-Bluff*; VOICE Magazine; 3/98; S.12 (1998)
- Breiner, Tobias: *Die goldenen Achziger, Die braunen Neunziger, die schwarzen Nullinger – Düstere Prognose des drohenden deutschen Wirtschaftskollaps*; grow music magazine; 4/97; Darmstadt (1996)
- Breiner, Tobias: *Wem gehört die Welt?*; grow music magazine, Nr.4/97; S.16; Darmstadt (1997)

- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie Saturns*; Zone-Zine Nr. 6/96; S.50-51; Dresden (1996)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur – Der mythologische Paradigmenwechsel zur Zeit Alexander des Großen*; Zone-Zine Nr. 5/96; S.50-51; Dresden (1996)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie von Neptun und Poseidon*; Zone-Zine Nr. 4/96; S.50-51; Dresden (1996)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie der Götterboten Hermes und Merkur*; Zone-Zine Nr. 2/96; S.50-51; Dresden (1996)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie der griechischen Unterwelt*; Zone-Zine Nr. 1/96; S.62-63; Dresden (1996)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologien von Pluto, Hades und Cernunnos*; Zone-Zine Nr. 6/95; S.62-63; Dresden (1995)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie von Pluto*; Zone-Zine Nr. 5/95; S.44-47; Dresden (1995)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologien der Liebesgöttinnen Freya, Aphrodite und Venus*; Zone-Zine Nr.4/95; S.62-63; Dresden (1995)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologien von Ziu, Ares und Mars*; Zone-Zine Nr. 3/95; S.62-63; Dresden (1995)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur – Die Mythologie von Uranos*; Zone-Zine Nr. 2/95; S.52-53; Dresden (1995)

- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur – Das Weltbild von Aristarch von Samos*; Zone-Zine Nr. 1/95; S.50-51; Dresden (1995)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie der Kelten*; Zone-Zine Nr. 6/94; S.50-51; Dresden (1994)
- Breiner, Tobias: *Das Liebesleben des gemeinen Zilpzalps*; Leibeygen Magazin, Oktober 1994; S.28-31; Darmstadt (1994)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie von Pholus*; Zone-Zine Nr. 5/94; S.61-62; Dresden (1994)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie Chirons*; Zone-Zine Nr. 4/94; S.61-62; Dresden (1994)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie Jupiters*; Zone-Zine Nr. 3/94; S.50-51; Dresden (1994)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie des Sonnengottes Helios*; Zone-Zine Nr. 2/94; S.70-71; Dresden (1994)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur - Die Mythologie der dreifachen Mondgöttinnen*; Zone-Zine Nr. 1/94; S.70-71; Dresden (1994)
- Breiner, Tobias: *Serie: Antike Mythologie und Moderne Subkultur – Einführung in die griechische Mythologie*; Zone-Zine Nr. 6/93; S.50-51; Dresden (1993)
- Breiner, Tobias: *Waterloo auf dem Fettstuhl*; Aberration; März 1991; Stuttgart; S.27 - 31 (1991)
- Breiner, Tobias: *Panokratie*; 3D-Holotek, Darmstadt; 335 Seiten (1990)

9.2.2 Vorträge

- **Thema des Vortrages:** e-Voluzzer[®]: Hydropneumatische Biosimulation für computergestützte Modellierung und Simulation organischer Konstruktionen und Evolutionsverläufe
Veranstaltung: 74. Jahresveranstaltung der Paläontologischen Gesellschaft
Ort: Georg-August Universität Göttingen
Datum: 05.10.2004
Referenten: Gudo, Michael; Syed, Tareq; Oschmann Wolfgang
Mitwirkende: Prof. Dr. Krömker, Detlef; Breiner, Tobias
- **Thema des Vortrages:** Paradigmenwechsel bei Simulationen
Veranstaltung: Forum Interaktive Visuelle Kommunikation
Ort: Cybernarium in Darmstadt
Datum: 08.07.2004
Referent: Breiner, Tobias
- **Thema des Vortrages:** e-Voluzzer: Hydropneumatic biosimulation for computer-based modeling and simulation of evolutionary transitions
Veranstaltung: 97. Jahresveranstaltung der Deutschen Zoologischen Gesellschaft
Ort: Universität Rostock
Datum: 02.06.2004
Referenten: Dr. Syed, Tareq; Dr. Gudo, Michael
Mitwirkender: Breiner, Tobias
- **Thema des Vortrages:** Aufbau der Solar System 3D Engine
Veranstaltung: Fraunhofer AGC Forum 2002
Ort: Frankfurt am Main
Datum: August 2003
Referent: Breiner, Tobias

- **Thema des Vortrages:** Structure of the Solar System 3D Engine
Veranstaltung: Symposium Visual Solutions
Ort: Atricom, Frankfurt am Main
Datum: Sommer 2003
Referent: Breiner, Tobias
- **Thema des Vortrages:** Cross-Road-Fahrsimulationen
Veranstaltung: Internationale Funkausstellung Berlin (IFA) 2002
Ort: Berlin
Datum: Sommer 2002
Referent: Breiner, Tobias
- **Thema des Vortrages:** Konzeption und visueller Aufbau von Autoren- und Lernprogrammen in der Biochemie
Veranstaltung: Best Practice Seminar „Chancen und Perspektiven moderner Visualisierungstechnologien in Medizin und Pharmazie“ auf der MEDICA 2001
Ort: Messecenter Düsseldorf
Datum: 23.11.2001
Referent: Breiner, Tobias
- **Thema des Vortrages:** Erstellung multimedialer Lernmodule für die Biochemie
Veranstaltung: k-MED Workshop: „Pilot Projects for knowledge-Based Medical Education“
Ort: Homberg/Ohm
Datum: 20.10.2001
Referent: Breiner, Tobias
- **Thema des Vortrages:** Visualisierungen und VR für die holistische Analyse regulatorischer und metabolischer Netzwerke
Veranstaltung: Beilstein-Symposium: „Visual Chemistry“ (GDCh Annual Meeting)
Ort: Universität Würzburg
Datum: 25.09.2001
Referenten: Breiner, Tobias; Prof. Dr. Krömker, Detlef

- **Thema des Vortrages:** Freie Konzeption zukünftiger Fahrsimulationen
Veranstaltung: 7. Workshop Sichtsysteme 2001 – Visualisierung in der Simulationstechnik
Ort: Wuppertal
Datum: 01.03.2001
Referent: Breiner, Tobias

- **Thema des Vortrages:** Interaktive Fahrsimulationen für die Airbagoptimierung
Veranstaltung: Siemens BSRS Restraint Systems - Automotive Airbag Forum;
Ort: Alzenau
Datum: 12.11.1999
Referent: Breiner, Tobias

9.2.3 Organisierte Workshops

- **Workshop:** Best Practice Seminar: Real 3D-2D Displays
Ort: Fraunhofer AGC in Frankfurt
Datum: 16.05.2002
Link: <http://www.agc.fhg.de/agc/events/BestPractice>
Organisator(inn)en: Prof. Dr. Krömker, Detlef; Seiler, Christian; Breiner, Tobias; Quasten, Jenny
- **Workshop:** Best Practice Seminar: „Chancen und Perspektiven moderner Visualisierungstechnologien in Medizin und Pharmazie“
Ort: MEDICA 2001 - Messecenter Düsseldorf
Datum: 23.11.2001
Link: <http://www.agc.fhg.de/agc/events/BestPractice>
Organisator(inn)en: Prof. Dr. Krömker, Detlef; Prof. Dr. Dörner, Ralf; Breiner, Tobias; Quasten, Jenny
- **Workshop:** Best Practice Seminar: Visualisierung in den Life Sciences
Ort: Fraunhofer AGC in Frankfurt
Datum: 15.07.2001
Link: <http://www.agc.fhg.de/agc/events/BestPractice>
Organisator(inn)en: Prof. Dr. Krömker, Detlef; Breiner, Tobias; Seiler, Christian; Quasten, Jenny

9.2.4 Erstellte Software

e-VoLuzie & e-VoLutz

Kunde: Cybernarium GmbH.

Art: Interaktive Saurier als Demonstration der Quaoaring-Technologie

Datum: 2004&2005



e-Voluzzer

Kunde: Forschungsinstitut Senckenberg

Art: Software-Framework zur elektronischen Visualisierung und Organsimulation von Lebewesen unter Zeitaspektberücksichtigung zur evolutionären Rekonstruktion

Datum: 2003-2004



Daimler-Chrysler 3D Research Framework

Kunde: Daimler-Chrysler GmbH.

Art: 3D-Visualisierung der Kräfte, welche auf die ungefederte Masse einwirken, inklusive der daraus resultierenden Deformationen.

Datum: 2002



Virtual Optimizer

Kunde: Canon Inc.

Art: Programm zur automatischen Optimierung von in Mantra4D importierter Auto-CAD-Daten für VR-Anwendungen

Datum: 2002



Daimler Chrysler Werk Tascalousa

Kunde: Daimler-Chrysler GmbH.

Art: Optimierung und Aufbereitung der 3D-CAD-Daten des geplanten Daimler-Chrysler-Werkes in Tascalousa/USA zur interaktiven virtuellen Begehung.

Datum: 2002



Auro

Kunde: RealiCon GmbH.

Art: Objektsuche in CAD-Daten zur automatischen Erstellung einer Szenegraphen-Hierarchie

Datum: 2002



Werksbefahrung Mettingen

Kunde: Daimler-Chrysler GmbH.

Art: Virtuelle Befahrung des Daimler-Chrysler Werksgeländes in Mettingen durch eine stereoskopische Fahrsimulation

Datum: 2002



Stratopolis IFA

Kunde: ComLogic

Art: Interaktive, immersive und physikalisch korrekte Fahrsimulation mit autonomer KI-Kameraführung und stereoskopischer Projektion, u.a. für die Internationale Funkausstellung 2002 in Berlin

Datum: 2002



Daimler-Chrysler Werk Berlin70

Kunde: Daimler-Chrysler GmbH.

Art: Aufbereitung und Optimierung der CAD-Daten des Berliner Daimler-Chrysler-Werkes für virtuelle Echtzeitbegehungen

Datum: 2001-2002



Smart Roadster Simulationsvisualisierung

Kunde: Daimler-Chrysler GmbH.

Art: Achsschenkelsimulation mit interaktiver 3D-Visualisierung für den „Smart Roadster“

Datum: 2001-2002



Stereoviewer

Kunde: Fujitsu-Siemens

Art: Programm zur mobilen stereoskopischen Projektion von 3D-Daten mit Hilfe zweier Laptops und zwei Mini-Beamern

Datum: 2000



Stratopolis

Kunde: ComLogic

Art: Interaktive, physikalisch korrekte Cross-Road-Fahrsimulation

Datum: 2000



Tip-Star

Kunde: Gauselmann GmbH.

Art: Wettautomat für die Nürnberger Automatenmesse

Datum: 1999



RealiMation Testumgebung

Kunde: Datapath Ltd./Criterion Software Inc.

Art: Entwicklung einer Softwaretestumgebung für das RealiMation SDK

Datum: 1999-2001



Merkur Projektstudie

Kunde: Gauselmann GmbH.

Art: 3D-GUI Projektstudie für Arcade Games

Datum: 1999



Scooter

Kunde: Escape GbR.

Art: 3D Scooter Driving Game

Datum: 1999



CAD Furniture Designer

Kunde: Hi-CAD GmbH.

Art: Spezialisiertes CAD-Modellierungsprogramm
für Möbel

Datum: 1999



BSRS-Fahrsimulation

Kunde: Siemens BSRS GmbH.

Art: Interaktiver 3D-BSRS-Fahrsimulation mit Airbagauslösung für die
IAA 1999

Datum: 1999



Smart Steering Wheel

Kunde: TRW GmbH.

Art: Interaktive Force-Feedback-Simulation eines Lenkrades von TRW
für die IAA 1999 & 2001

Datum: 1999



Braking Assistant Simulator

Kunde: TRW GmbH.

Art: 3D-Simulation und Visualisierung des Braking Assistant Systems
„BAS“

Datum: 1999



BMW-Race

Kunde: BMW GmbH.

Art: Physikalisch korrekte interaktive 3D-Fahrsimulation für
Messevents

Datum: 1999



Fahrsimulation Nürburgring

Kunde: BMW GmbH.

Art: BMW-Fahrsimulation für Nürburgring-Marketingevent

Datum: 1999



CAD Furniture Query

Kunde: Hi-CAD GmbH.

Art: Intelligentes Suchprogramm von Möbeln in CAD-Datenbanken

Datum: 1999



CAD-Rendernet-Konverter

Kunde: Hi-CAD GmbH.

Art: Konvertierungsprogramm zwischen CAD-Daten und Rendernet-Format

Datum: 1999



TRW Lenkradsimulations-Framework

Kunde: TRW GmbH.

Art: Simulationsframework mit Force-Feedback für Servolenkungen

Datum: 1999



Crash Test Dummy Demo

Kunde: Siemens BSRS Restraint Systems

Art: Interaktive 3D-Unfalldemonstration für die BSRS-Inhouse-Messe

Datum: 1999



Siemens Smart Airbag Simulator

Kunde: Siemens BSRS Restraint Systems

Art: Verhaltenssimulation des „Smart Airbags“ von Siemens BSRS inklusive dreidimensionaler Visualisierung

Datum: 1999



Rendernet

Kunde: escape GbR.

Art: Programmpaket, welches ein Verteiltes Rendering im LAN ermöglicht

Datum: 1998-2000



Slaughtybartfas

Kunde: Escape GbR.

Art: Fraktale 3D-Landschaftsgenerierung

Datum: 1998



Merkur Backgammon

Kunde: Gauselmann GmbH.

Art: 2D Backgammon Arcade Game

Datum: 1998



Chart Breaker

Kunde: Banco Do Brasil (potentiell, Kontrakt nicht zustande gekommen)

Art: Aktienverwaltungsprogramm (inklusive Chartvisualisierung und Chartanalyse, Prognose, Renditeberechnung, Portfolio, Zyklenstatistik)

Datum: 1989-1994



Infraphon

Kunde: -

Art: Sequenzer zur Ansteuerung von MIDI-fähigen Musikinstrumenten

Datum: 1988



Grund- und Aufbaukurs Englisch

Kunde: Adventure-Soft

Art: Englisch-Lernprogramm

Datum: 1987



Come on Baby, lösche my Fire!

Kunde: Feuerwehr Offenbach

Art: Graphisch-interaktive Feuerwehr-Ausbildung

Datum: 1987



Logoko

Kunde: Adventure-Soft GmbH.

Art: Autorenprogramm für e-Learning Applikationen

Datum: 1986



Tjonisch-Editor

Kunde: Panokratische U-Topienauten

Art: Editor für tjonische Schriftstücke

Datum: 1985



Elektra

Kunde: -

Art: Berechnungsprogramm des Verhaltens elektrischer Schaltungen

Datum: 1985



Astro

Kunde: -

Art: Berechnung von Sonne-, Mond- und Planetenpositionen

Datum: 1985



3D Mathe

Kunde: -

Art: Visualisierungsprogramm für zweidimensionale Mathematikfunktionen

Datum: 1985



9.2.5 Ausstellungen und Messepräsentationen

- **Veranstaltung:** Dauerausstellung des Cybernariums
Datum: ab 3.4.2004
Ort: Darmstadt
Exponat: Erweiterte Interaktive Quaoaring-Modellierung eines Protoplesiosauriers
Sichtsystem: Holoscheibenprojektion
Urheber der Software: Breiner, Tobias
- **Veranstaltung:** Eröffnungsausstellung des Cybernariums
Datum: 2.4.2004
Ort: Darmstadt
Exponat: Interaktive Quaoaring-Modellierung eines Protoplesiosauriers
Sichtsystem: Holoscheibenprojektion
Urheber der Software: Breiner, Tobias
- **Veranstaltung:** Wiedereröffnungsausstellung des Naturkundemuseums Senckenberg
Datum: 1.9.2003
Ort: Frankfurt am Main
Exponat: Interaktive Simulation der Enteropneusten-Evolution
Sichtsystem: Projektion
Urheber der Software: Breiner, Tobias
- **Veranstaltung:** Abschluss-Veranstaltung des Naturkundemuseums Senckenberg
Datum: 1.3.2003
Ort: Frankfurt am Main
Exponat: Insektenähnliches Tier zur Demonstration der Pneu-Modellierung
Sichtsystem: Projektion
Urheber der Software: Breiner, Tobias

- Veranstaltung:** Daimler-Chrysler GmbH. – Virtual Worlds
Datum: 03.11.2003 (genaues Datum unsicher)
Ort: Berlin
Exponat: Virtuelle Fabrikvisualisierung (Werk Mettingen)
Sichtsystem: Gebogene stereoskopische Powerwall (6 Beamer)
Urheber der Software: Roosen, Gottfried; Breiner, Tobias
- Veranstaltung:** Internationale Funkausstellung Berlin (IFA)
Datum: 29.08.2002-03.09.2002
Ort: Berlin
Exponat: Stratopolis-Fahrsimulation
Sichtsystem: Stereoskopische Riesen-Powerwall (25*12 Meter)
Urheber der Software: Breiner, Tobias
- Veranstaltung:** EuroMold 2002
Datum: 28.11.2001-01.12.2001
Ort: Frankfurt am Main
Exponat: 3D- Visualisierung der mechanischen Kraft- und Verbiegungssimulation der Hinterachse des Smart Roadsters
Sichtsystem: Stereoskopische Projektion
Urheber der Software: Breiner, Tobias (Entwicklung & Programmierung); Roosen, Gottfried (Organisation); Zeugmann, Thomas (Datenversorgung)
- Veranstaltung:** MEDICA 2001
Datum: 22.11.2001-25.11.2001
Ort: Düsseldorf
Exponat 1: Visualisierung des Citrat-Cyclus;
Exponat 2: Simulation der Goldmann-Gleichung;
Exponat 3: Toxikologie-Simulation
Sichtsystem: 4 getrennte CRT-Monitore
Urheber der Software: Breiner, Tobias (Exp.1,2 & 3) ; Baier, Wolfgang (Exp.1)

- **Veranstaltung:** Siemens BSRS Restraint Systems – Automotive Airbag Forum

Datum: 03.12.1999

Ort: Alzenau

Exponat: Interaktiver Airbag-Simulator

Sichtsystem: 23''CRT-Monitor

Urheber der Software: Breiner, Tobias; Wombacher, Hubert; Wenzel, Thomas; Trebs
Harald

9.2.6 Betreute Studien- und Diplomarbeiten

- **Thema:** Entwicklung eine lokationssensitiven Puppe
Diplomand: Perkovic, Toni
Art: Laufende Diplomarbeit am Fachbereich Mathematik und Informatik der Johann Wolfgang Goethe-Universität Frankfurt am Main
Betreuer: Breiner, Tobias
- **Thema:** Fraktale Generierung virtueller Planeten durch rekursive Oktaederstrukturen
Diplomand: Homann, Jörg
Art: Laufende Diplomarbeit am Fachbereich Mathematik und Informatik der Johann Wolfgang Goethe-Universität Frankfurt am Main
Betreuer: Breiner, Tobias
- **Thema:** Entwicklung eines Echtzeit-Raytracers
Diplomand: Nemeč, Toni
Eingereicht: September 2005
Art: Diplomarbeit am Fachbereich Mathematik und Informatik der Johann Wolfgang Goethe-Universität Frankfurt am Main
Betreuer: Breiner, Tobias
Prüfer: Prof. Dr. Krömker, Detlef
- **Thema:** Entwicklung eines Klassengraphen
Diplomand: Stocklów, Carsten Rudolf
Art: Diplomarbeit am Fachbereich Biologie und Informatik der Johann Wolfgang Goethe-Universität Frankfurt am Main
Eingereicht: Oktober 2004
Betreuer: Breiner, Tobias
Prüfer: Prof. Dr. Krömker, Detlef
Zweitkorrektor: Prof. Dr. Waldschmitt, Klaus

- **Thema:** Visualisierung und Simulation der Physiologie von Neuronen
Diplomand: Bialojahn, André
Art: Diplomarbeit am Fachbereich Biologie und Informatik der Johann Wolfgang Goethe-Universität Frankfurt am Main
Eingereicht: September 2004
Betreuer: Breiner, Tobias; Dr. Weiland, Claus
Prüfer: Prof. Dr. Krömker, Detlef
Zweitkorrektor: Prof. Dr. Wotschke, Detlef

- **Thema:** Simulation von Muskeln, Sehnen und Bändern
Diplomand: Burger, Armin
Art: Diplomarbeit am Fachbereich Biologie und Informatik der Johann Wolfgang Goethe-Universität Frankfurt am Main
Eingereicht: Juli 2004
Betreuer: Breiner, Tobias
Prüfer: Prof. Dr. Krömker, Detlef
Zweitkorrektor: Prof. Dr. Waldschmitt, Klaus

- **Thema:** Playing and Programming Computer Games: Spiel als kulturelle Praxis
Diplomandin: Kinder, Katharina
Art: Fachübergreifende Diplomarbeit zusammen mit dem Fachbereich Kulturanthropologie der Johann Wolfgang Goethe-Universität Frankfurt am Main
Eingereicht: März 2003
Betreuer: Breiner, Tobias
Prüfer: Prof. Dr. Faßler, Manfred
Zweitkorrektor: Prof. Dr. Krömker, Detlef

- **Thema:** Programme und Skriptsprachen für die Modellierung von Molekülen und die Behandlung ihrer Konformationen

Diplomand: Huang, Lidong

Art: Diplomarbeit am Fachbereich Biologie und Informatik der Johann Wolfgang Goethe-Universität Frankfurt am Main

Eingereicht: Juli 2002

Betreuer: Breiner, Tobias; Seiler, Christian

Prüfer: Prof. Dr. Krömker, Detlef

Zweitkorrektor: Prof. Dr. Brause

9.2.7 Mitwirkung in Lehrveranstaltungen

Wintersemester 2005/2006

- Vorlesung und Übung Grundlagen der Programmierung (für Bioinformatiker)
- Vorlesung und Übung Visualisierung
- Praktikum Graphische Datenverarbeitung

Sommersemester 2005

- Vorlesung und Übung Graphische Datenverarbeitung

Wintersemester 2004/2005

- Programmierpraktikum

Sommersemester 2004

- Proseminar Simulation und Visualisierung in der Biologie
- Seminar Simulation und Visualisierung in der Biologie

Wintersemester 2003/2004

- Proseminar Computergraphische Menschmodelle
- Seminar Computergraphische Menschmodelle
- Übung zu Vorlesung Animation und Multimedia

Sommersemester 2003

- Ergänzungen zu Graphische Datenverarbeitung
- Übung zu Graphische Datenverarbeitung
- Proseminar Visualisierung in der Bioinformatik

Wintersemester 2002/2003

- Übung zu Animation und Multimedia
- Proseminar Computer Graphics and 3D Games

Sommersemester 2002

- Ergänzungen zu Graphische Datenverarbeitung I
- Übung zu Graphische Datenverarbeitung I
- Seminar 3D Games
- Proseminar 3D Games

Wintersemester 2001/2002

- Praktikum 3D-Spiele (Pacman 3D) inklusive VRML- und Java3D-Kurs
- Übung Animation und Multimedia, Wintersemester 2001/2002

Sommersemester 2001

- Praktikum Graphisch Interaktive Systeme
- Ergänzungen zu Graphische Datenverarbeitung I
- Übung zu Graphische Datenverarbeitung I

Wintersemester 2000/2001

- Übung zu Animation und Multimedia

Sommersemester 2000

- Ergänzungen zu Graphische Datenverarbeitung I
- Übung zu Graphische Datenverarbeitung I

Wintersemester 1999/2000

- Proseminar Animation und Visualisierung
- Seminar Animation und Visualisierung
- Blockpraktikum Computergraphik mit VRML und Java-3D

Alle Lehrveranstaltungen fanden im Fachbereich „Biologie und Informatik“ (bis SS 2004) bzw. „Informatik und Mathematik“ (ab WS 2004/05) der Johann Wolfgang Goethe-Universität in Frankfurt am Main statt.

9.3 Lebenslauf

Name	<u>Tobias</u> Christopher Breiner	
Geburtsdaten	10. August 1965 in Frankenthal/Pfalz	
Familienstand	Verheiratet seit 15.04.1999 mit Nicole Jansen, Dipl.-Pädagogin	
Kinder	2 Söhne (geb. 1999 und 2001) 1 Tochter (geb. 2004)	
Schulbesuch	1972-1976	Grundschule in Frankenthal/Pfalz
	1976-1985	Albert-Einstein-Gymnasium Frankenthal
	1985	Abitur
Akademischer Werdegang	1986-1988	Informatikgrundstudium an der TU Darmstadt
	1989	Stipendium an der Universidade Nova de Lisboa in Portugal
	1990-1994	Informatikhauptstudium sowie Studium des Nebenfaches Biologie an der TU Darmstadt
	1994	Diplom in Informatik mit dem Thema: „Generierung einer internationalen Plansprache unter besonderer Berücksichtigung der textuellen und verbalen Interaktion zwischen Mensch und Computer“

2002 Anfang der Promotionsarbeiten;
Thema der schriftlichen Ausarbeitung:
„Dreidimensionale virtuelle Organismen“;
Voraussichtlicher Disputationstermin Ende 2005

Beruflicher

Werdegang

1981-1987 Lehrbeauftragter für Informatik an der Augustin-
Violet-Schule in Frankenthal (Sondergenehmigung)

1985-1989 freiberufliche Programmierung von Börsen-,
Astronomie-, Spiele-, Musik- und Autorenprogrammen
auf Apple 3 und ST unter UCSD-Pascal bzw. Pascal++,
Verkauf an div. Softwarefirmen.

1987-1988 &

1990-1991 wissenschaftlicher Mitarbeiter am Fraunhofer-Institut für
Graphische Datenverarbeitung in Darmstadt
Mitentwicklung des objektorientierten
Datenbanksystems ADS1; Arbeit mit C unter Unix

1992-1995 Projektleiter bei der Firma SBS-Darmstadt GmbH;
Entwicklung von KI-Maschinensteuerungssoftware;
Arbeit mit Turbo Pascal unter OS/2

1995-1997 Koordinierender Softwareingenieur in der
Entwicklungsabteilung der Firma Controlware GmbH
in Dietzenbach; Arbeit mit Internet-Protokollen und
Echtzeit-Targetsystemen. Programmierung von Visual
Basic und Visual C++ unter Windows 95

- 1998-2000 Selbstständiger Softwareingenieur;
 Programmierung von Echtzeit-Fahrsimulatoren u.a.
 für BMW, Daimler-Chrysler, ComLogic, TRW,
 Siemens BSRS Restraint Systems und escape;
 Arbeit mit 3DMax, RealiMation, Mantra4D, DirectX,
 Visual C++ und MFC unter Windows NT/98/2000
- 2000-2005 Wissenschaftlicher Mitarbeiter an der Johann Wolfgang
 Goethe-Universität in Frankfurt am Main,
 Fachbereich Biologie und Informatik,
 Professur Prof. Dr. Krömker – Graphische
 Datenverarbeitung

zusätzliche

Qualifikationen

- 1984 Jugend forscht - Landespreisträger im Bereich
 Informatik
- 1988 Erstes deutsch-portugiesisches Erasmus-Stipendium
 vermittelt durch *Prof. Dr. Encarnação*
 nach Arbeiten in der GDV

Fremdsprachen

Muttersprache Deutsch, Englisch, Französisch, Spanisch,
 Portugiesisch, Neunorwegisch (nynorsk)

Hobbies

- Entwicklung von Sprachen und Schriften (Snupisch, Prototjonisch,
 Tjonisch 1-5)
- Komponieren (Keyboarder & Ex-Sänger der Band „Pampa“)
- Graffitikunst (Graffiti-Ausstellungen)
- Entwicklung von politischen Utopien (Modell der Panokratie)
- Geschichte und Psychologie von Subkulturen (viele Artikel)

