

Separation und Rekonstruktion funktionaler Elemente im Zentral-Nervensystem

- Dissertation -
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim
Fachbereich Biologie und Informatik der
Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von

Oliver Bossert¹
aus Frankfurt am Main

Frankfurt 2003
D F 1

¹e-mail: olf@olfsworld.de

vom Fachbereich Biologie und Informatik der
Johann Wolfgang Goethe-Universität als Dissertation angenommen

Dekan:

Gutachter:

Datum der Disputation:

*»It is a capital mistake to theorize
in advance of the facts. . . «*
Arthur Conan Doyle
(Sherlock Holmes)

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung	4
1.2	Gliederung	7
2	Material und Methode	9
2.1	Histologische Methode	9
2.2	Mikroskopische Aufnahmen	11
2.2.1	Einstellungen am Mikroskop	12
2.2.2	Anpassen der Aufnahmetechnik	13
2.2.3	Einfluss der Mikroskopeinstellung auf die Mustererkennung	28
2.3	Verwendete Hard- und Software	30
2.3.1	Software	30
2.3.2	Hardware	31
2.4	Fazit	36
3	Image Mosaicing	37
3.1	Die Image Mosaicing-Verfahren	38
3.1.1	Image Mosaicing mittels Phasenkorrelation	38
3.1.2	Image Mosaicing mittels Kreuzkorrelation	43
3.2	Implementierung des Image Mosaicings	47
3.3	Kontrollmechanismen	47
3.4	Überblenden der Bilder	48
3.5	Interface und Grafische Benutzeroberfläche	51
3.6	Validierung des Verfahrens	51
3.7	Fazit	55
4	Mustererkennung	57
4.1	Auswahl der Landmarken	58
4.2	Farbkonstanz	60
4.2.1	Klassische Algorithmen zur Farbkonstanz	61
4.2.2	Farbkorrektur bei histologischen Präparaten	70
4.2.3	Ergebnisse der Farbkorrektur	75
4.3	Maskierung der Kapillaren	78
4.4	Vorfilterung der Zellen durch „Color Sensitive Edgedetection“	79
4.4.1	Color-Sensitive Edgedetection	81

4.5	Lokale Kontrasterhöhung	85
4.6	Lokalisieren der Zellkerne	87
4.6.1	Validierung der Zellkern-Mustererkennung	88
4.7	Fazit	89
5	Ausrichten	91
5.1	Alignments auf Basis der Phasenkorrelation	93
5.2	Archimedes-Alignment-Algorithmus	95
5.2.1	Methode	96
5.2.2	Test und Korrektur der Parameter	97
5.3	Validierung der Ergebnisse	99
5.4	Fazit	101
6	Rekonstruktion	103
6.1	Filtern der Segmente im dreidimensionalen Raum	104
6.2	CellInspector	105
6.3	Ergebnis	107
6.4	Verschiedene Möglichkeiten der Rekonstruktion	111
6.5	Fazit	117
7	Diskussion	119
7.1	Verfahrensschritte	119
7.2	Erweiterungen der Software	124
7.2.1	Bestimmen der Zellkern-Kontur	124
7.3	Detektion der Somatagrenzen	127
7.3.1	Entwicklung einer grafischen Bedienoberfläche	129
7.4	Apparativer Ausbau und Erweiterung der Histologie	130
7.4.1	Integration von weiterem Datenmaterial	131
7.4.2	Erweiterung der Mikroskopie	131
7.4.3	Berechnung im Cluster	133
7.5	Ausblick	133
7.5.1	Möglichkeiten zur Modellvalidierung	133
7.5.2	Andere Anwendungsgebiete	142
	Literaturverzeichnis	153
	A Puffer und Lösungen zur Histologie	163
	B Protokoll zur Mikroskopie	165

Einleitung

Das Gehirn ist das komplexeste und trotzdem das am wenigsten verstandene Organ. Schon seit langer Zeit fasziniert die Menschen die Leistungsfähigkeit und die Anpassungsfähigkeit des Gehirns, die bis heute von keinem Computer erreicht werden. Diese Faszination hat zusammen mit der Suche nach Behandlungsmöglichkeiten von Störungen oder Ausfällen von Gehirnarealen dazu geführt, dass in den vergangenen Jahrzehnten eine große Anzahl von Methoden zur Untersuchung des Gehirns entwickelt wurde [58].

So können wir bereits bestimmte Aufgaben wie Sprache, Motorik oder andere Tätigkeiten einem Gehirnareal zuordnen. Auch auf der Ebene einzelner Zellen ist die Forschung sehr weit fortgeschritten. So kann man etwa die Anatomie einer Einzelzelle, ebenso wie Membraneigenschaften, biochemische Charakteristika und sogar die Funktionsweise einzelner Ionenkanäle sehr exakt bestimmen – zwischen diesen Organisationsebenen klafft allerdings eine große Lücke, für die es wenig oder gar keine Daten gibt.

Gerade was die Anordnung und Verknüpfung von Nervenzellen betrifft, gibt es nur sehr wenige Untersuchungen. Diese Ebene der „Microcircuits“ ist aber von größtem Interesse, da es nur mit diesen Daten gelingen kann, die Arbeitsweise des Gehirns zu verstehen. In seinem Buch „Descartes’ Irrtum“ [22] vergleicht Damasio die „umfassende Beschreibung der Schaltkreise und Systeme [des Gehirns] auf mikro- wie auf makrostruktureller Ebene“ mit einem Wert, der der „Entschlüsselung der DNA-Struktur durchaus vergleichbar ist“.

Um eine solch umfassende Beschreibung für zumindest einen Nukleus zu erreichen, ist die präzise anatomische Analyse von Anzahl und Anordnung der Neurone innerhalb des Nukleus notwendig. Nur auf dieser Grundlage wird es möglich sein, den Aufbau bei unterschiedlichen Individuen zu vergleichen und eine abstrakte Karte der Neuronen zu erstellen. Mit einer solchen Karte, welche die Organisation des Nukleus beschreibt, könnte man die Ergebnisse unterschiedlicher anatomischer, physiologischer und biochemischer Methoden einem Neuron zuzuordnen und somit den Aufbau und die Funktionsweise des Nukleus stückweise im Modell nachbilden.

Um einen Nukleus auf zellulärer Ebene zu analysieren, ist einerseits eine ausreichende Auflösung für die Detektion einzelner Zellen und andererseits ein Abbildungsbereich mit der Größe des Nukleus notwendig.

Wie in Abbildung 1.1 schematisch dargestellt, sind für jede Rekonstruktion zwei Achsen entscheidend: Einmal die laterale Ebene (auch als x/y-Ebene bezeichnet), die beim Mikroskop einer Aufnahme entspricht und zum anderen die axiale oder z-Achse, die die dritte Dimension darstellt und beim Mikroskop der Schnittdicke oder dem nächsten Schnitt zugeordnet werden kann. Die Zuordnung der Achsen ist prinzipiell willkürlich, jedoch werden sie häufig so definiert, dass die x- und die y-Achse eine identische Auflösung aufweisen und diese meist besser ist als diejenige entlang der z-Achse.

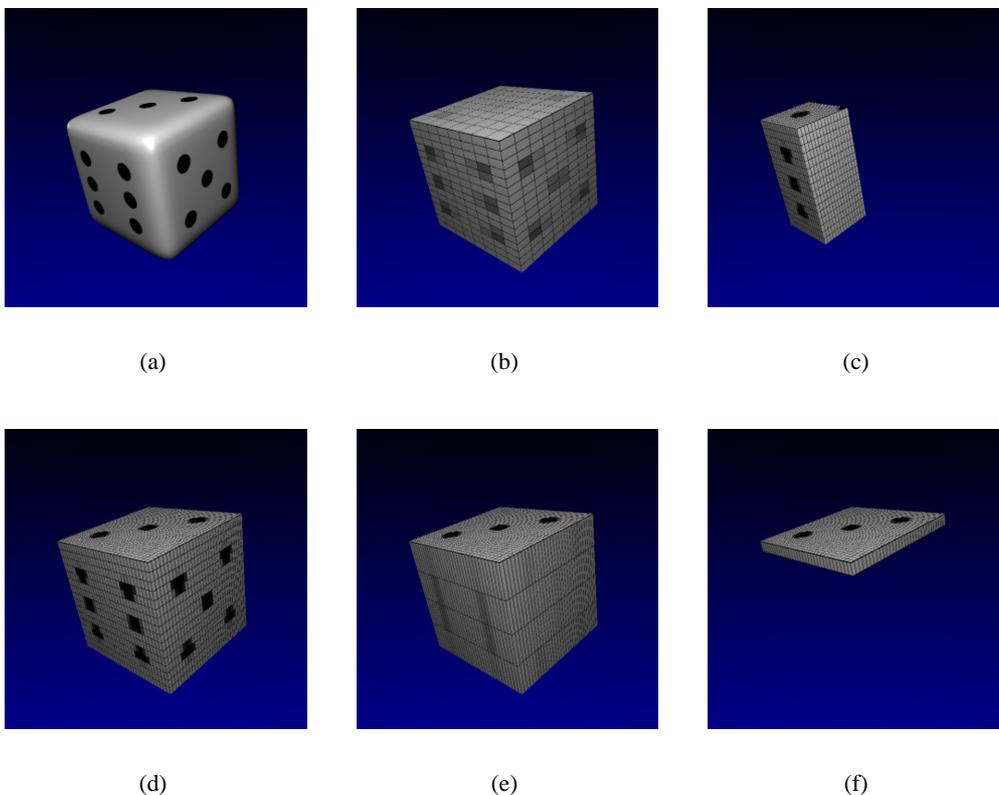


Abbildung 1.1: Schematische Darstellung der methodischen Grenzen: Bei der Rekonstruktion des in (a) abgebildeten Würfels muss zum einen in der hier in der Aufsicht zu sehenden lateralen Ebene eine ausreichend hohe Auflösung und ein genügend großer Abbildungsbereich gegeben sein. In (b) ist die laterale Auflösung zu gering, in (c) hingegen ist der Abbildungsbereich zu klein. Gleiches gilt für die axiale Richtung: In (e) ist die axiale Auflösung, in (f) der axiale Abbildungsbereich nicht ausreichend. Bei der in (d) abgebildeten Rekonstruktion sind beide Parameter richtig dimensioniert.

In Abbildung 1.2 auf der nächsten Seite sind die Auflösungen und Abbildungsbereiche unterschiedlicher Verfahren zusammen mit der in Rot dargestellten Größe zweier auditorischer Nuklei der Wüstenrennmaus [86], dem Inferioren Colliculus (IC) und

dem Komplex der Superioren Olive (SOC), gezeigt. Nur wenn der Kasten eines Verfahrens den roten Bereich ganz einschließt, ist es in der Lage, diesen zu rekonstruieren. Die vier begrenzenden Parameter sind in Abbildung 1.1 am Beispiel der Rekonstruktion eines Würfels dargestellt.

Bei den bildgebenden Verfahren wie der funktionalen Magnet-Resonanztomographie (fMRI) oder der Positronen-Emissions-Tomographie (PET) ist es zwar möglich, das gesamte Gehirn aufzunehmen, allerdings ist die Auflösung für die Abbildung einzelner Zellen nicht ausreichend. Dies entspricht den in Abbildung 1.1(b) und (e) dargestellten Problemen. Sowohl die laterale als auch die axiale Auflösung sind nicht ausreichend, um damit einzelne Strukturen zu erkennen.

Bei Verfahren wie der Konfokalen Laser-Scanning-Mikroskopie (CLSM) ist die Auflösung zwar für die Abbildung einzelner Zellen ausreichend, allerdings ist weder die laterale noch die axiale Ausdehnung des Abbildungsbereiches genügend groß, um eine Rekonstruktion vorzunehmen. Dies ist schematisch in den Abbildungen 1.1(c) und (f) gezeigt. Die Auflösung ist zwar ausreichend, um die einzelnen Strukturen voneinander zu isolieren, allerdings ist es nicht möglich den gesamten Würfel abzubilden.

Dies gilt natürlich ebenso für die Transmissions-Elektronenmikroskopie (TEM) und die Lichtmikroskopie. Bei diesen Verfahren ist vor allem die axiale Auflösung praktisch Null bzw. auf die Schnittdicke begrenzt (entsprechend Abbildung 1.1(c)).

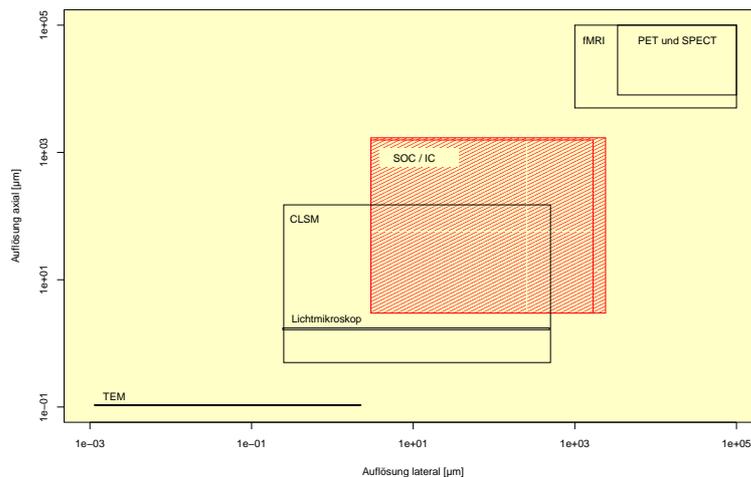


Abbildung 1.2: Auflösungs- und Abbildungsbereiche der unterschiedlichen bildgebenden Verfahren: Die untere linke Ecke eines Rechtecks kennzeichnet die maximale Auflösung und die obere rechte Ecke die Größen des maximalen Abbildungsbereiches des jeweiligen Verfahrens. ([2, 47, 58, 84, 86, 94]).

Da mit keiner der bekannten Methoden die Analyse eines gesamten Nukleus möglich ist, bestand das Ziel dieser Arbeit darin, eine Methode zu entwickeln, deren Auflösung die Betrachtung einzelner Zellen erlaubt und mit ihrem Abbildungsbereich die Größe eines Nukleus abdeckt.

Im Vordergrund stand dabei die Entwicklung von computergestützten Verfahren und Methoden, die zwar in der vorliegenden Arbeit auf ein spezielles Beispiel angewendet werden, allerdings nicht darauf beschränkt sind. Das Ziel bestand vielmehr darin,

ein universell anwendbares Verfahren zu entwickeln, mit dem praktisch jeder andere Nukleus zu analysieren ist. Neben der vollständigen Automatisierung des Verfahrens stand die universelle Anwendbarkeit im Mittelpunkt, so dass viele Verfahrensschritte ohne weiteres auch auf andere Gewebe oder andere Methoden, wie z.B. die Elektronenmikroskopie übertragbar sind.

Die hier vorgestellten Techniken bieten die Möglichkeit eines Brückenschlags zwischen der theoretischen Lösungen in Form von Modellen einerseits und der realen Topografie andererseits. Sie ermöglichen es erstmals, Gesamtmodelle zu konstruieren, deren Ausgangspunkt ein Modell *von* den realen Gegebenheiten ist und nicht nur ein Modell *für* die Funktionsweise, das eine mögliche Erklärung unter vielen liefert.

Die MSO (Mediale Superiore Olive) ist ein gutes Beispiel für diese Problematik. Schon seit Jeffress 1948 [52] existiert ein Modell für die Lokalisierung von Schallquellen. Es ist sehr wahrscheinlich, dass dieser Verarbeitungsprozess in der MSO lokalisiert ist. Diese wurde in den letzten Jahrzehnten mit den unterschiedlichsten Methoden untersucht. So existieren umfangreiche anatomische Arbeiten [59, 88, 89, 91], eine große Anzahl von elektrophysiologischen Arbeiten [33, 41, 42], Untersuchungen der Synapsenverteilung [59] sowie Untersuchungen mit vielen anderen Methoden [1, 82].

Einige der Experimente unterstützen die Sichtweise des Jeffress-Modells [54]; andere Befunde sprechen eher dagegen [30, 38, 40]. Es ist zur Zeit nicht möglich, ein einheitliches Modell der MSO zu konstruieren, da anatomische und elektrophysiologische Befunde einander nicht zugeordnet werden können und auch die einzelnen Arbeiten untereinander nur bedingt vergleichbar sind, da nicht klar ist, in wie weit einander entsprechende Zellen untersucht wurden. (In Kapitel 7.5.1 werden einige Möglichkeiten der Modellvalidierung noch ausführlich diskutiert.)

Diese Problematik ist nicht durch das weitere Verfolgen eines reinen *top-down*-Ansatzes zu lösen. Solange Modelle nicht an den realen Umständen gemessen und Messungen über eine eindeutige Identifikation der Zellen miteinander verglichen werden können, ist eine systematische Validierung der Modelle nicht möglich.

Nur mit einem *bottom-up*-Ansatz, der auf der Nervenzelle, deren anatomischen und physiologischen Eigenschaften basiert, ist es nach Koch [61] langfristig möglich, Gedächtnis, Wahrnehmung und Bewusstsein zu erklären und diese Aussagen zu testen.

Auch der Nobelpreisträger Gerald Edelman kritisiert den funktionalistischen Ansatz und behauptet, „dass sich geistige Funktionen durch theoretische Konstrukte, die allein auf kognitiver Psychologie und Computermodellierung basieren, nicht angemessen untersuchen lassen – statt dessen müssten sie in der realen Biologie des Gehirns und in der Gehirnevolution wurzeln“ (zitiert nach [94]).

Ich kann mich dieser Meinung nur anschließen und hoffe, dass es mir gelungen ist, mit dieser Arbeit einen Teil zu einer solchen Strukturanalyse beizutragen.

1.1 Problemstellung

Die Zielsetzung dieser Arbeit besteht darin, ein Computerprogramm zu entwickeln, das auf der Grundlage von einer Schnittserie in der Lage ist, vollautomatisch und ohne einen manuellen Eingriff eine funktionale Rekonstruktion (siehe S. 6) von einem Gehirnbereich zu erstellen.

Als Basis für die Entwicklung meiner eigenen Methode dient mir die klassische Lichtmikroskopie. Lichtmikroskopische Aufnahmen haben viele Vorteile. Neben der hohen Auflösung, die unabdingbar ist, erlaubt es dieses Verfahren, unterschiedliche histologische Techniken anzuwenden. Dadurch ist die Lichtmikroskopie flexibler und kostengünstiger als z.B. die auf fluoreszierende Farbstoffe angewiesene CLSM.

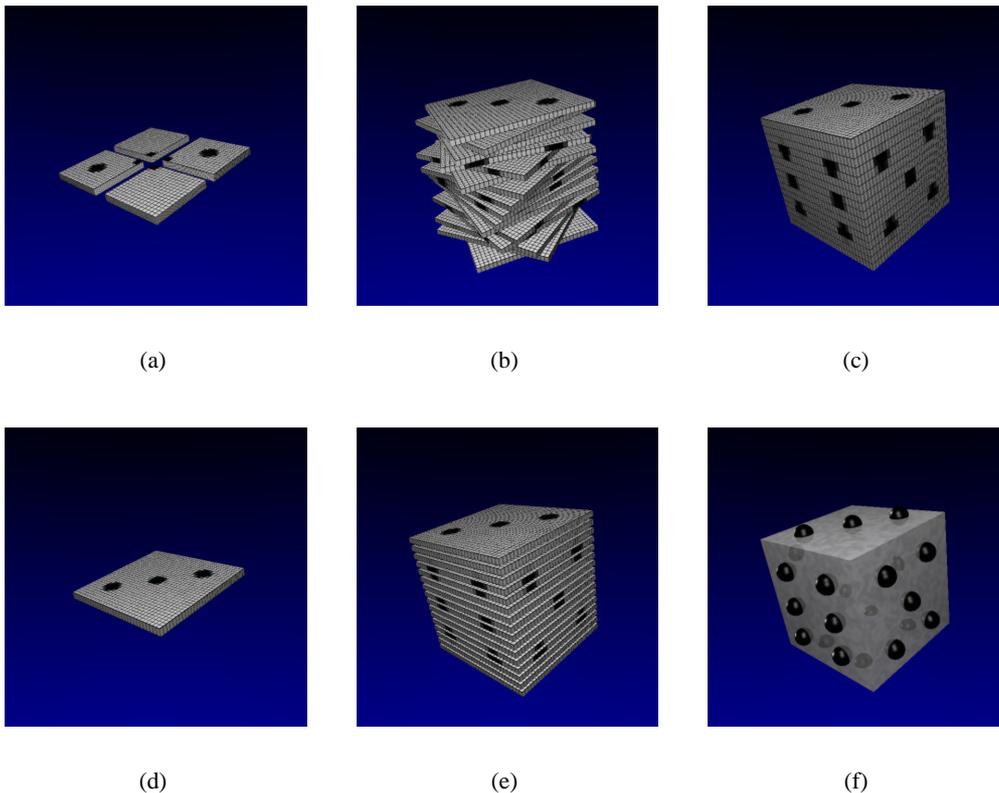


Abbildung 1.3: In der oberen Reihe sind die drei Ausgangssituationen dargestellt. In der unteren Reihe befinden sich die entsprechenden Lösungsansätze. In (a) ist eine Ebene aus vier Einzelbildern zu sehen. Diese sind in (d) zu einem Image Mosaic zusammengesetzt. Ein Stapel solcher Bilder (b) muss in nächsten Schritt korrekt ausgerichtet werden (e). Der daraus entstandene Voxelblock (c) wird zuletzt in ein funktionales Modell (f) überführt.

Um von einer normalen lichtmikroskopischen Aufnahme zur Rekonstruktion eines Nukleus zu gelangen, müssen drei grundlegende Probleme gelöst werden:

Der laterale Abbildungsbereich ist – wie in Abbildung 1.2 auf Seite 3 gezeigt – bei der Lichtmikroskopie zu klein, um einen Nukleus komplett abzubilden. Um den lateralen Abbildungsbereich zu vergrößern, könnte man prinzipiell eine kleinere Vergrößerung wählen, würde damit jedoch die Auflösung so weit reduzieren, dass wiederum das Erkennen einzelner Zellen unmöglich wird.

Dieses Problem habe ich durch den Prozess des Image Mosaicings, d.h. durch patchwork-artiges Zusammenfügen mehrerer Bilder gelöst. Dabei werden wie in Abbildung 1.3(a) und (d) dargestellt, mehrere Einzelbilder zu einem großen Bild mit hoher Auflösung kombiniert.

Nachdem der Bildausschnitt auf diese Weise vergrößert wurde, muss der axiale Abbildungsbereich ebenfalls vergrößert werden. Dies ist bei der Lichtmikroskopie prinzipiell durch zwei unterschiedliche Verfahrensweisen möglich. Entweder man erhöht die Schnittdicke und zerlegt die resultierenden Schnitte optisch, wie es beim CLSM der Fall ist oder man verringert die Schnittdicke so, dass hierdurch ein geringerer Abstand der physikalischen Schnittebenen und damit eine erhöhte Auflösung entlang der z-Achse erreicht wird.

Das erste Verfahren ist prinzipiell sehr attraktiv, hat jedoch zwei gravierende Nachteile: Zum einen ist man beim optischen Schneiden auf fluoreszierende Farbstoffe angewiesen. Dies ist nicht nur kostenintensiv, sondern macht auch die Anwendung vieler histologischer Techniken unmöglich. Zum anderen ist auch hier die axiale Auflösung begrenzt, so dass dies keine prinzipielle Problemlösung ist.

Ich habe in dieser Arbeit den zweiten Ansatz gewählt und sehr dünne Schnitte mit einer Dicke von 2 μm angefertigt, deren Aufnahmen dann jedoch am Computer ausgerichtet werden müssen, damit in axialer Richtung ein Bezugssystem hergestellt wird. Wie in Abbildung 1.3(b) und (e) dargestellt, entsteht nach der Aufnahme ein ungeordneter Stapel von Einzelbildern, die erst noch zueinander in Bezug gesetzt werden müssen.

Die danach vorliegende Rekonstruktion werde ich im folgenden als optische Rekonstruktion bezeichnen. Diese besteht aus Voxeln, d.h. Bildpunkten im dreidimensionalen Raum, wie in Abbildung 1.3(c) schematisch dargestellt. Dort besteht aber noch keine Möglichkeit, einzelne Strukturen zu identifizieren. Die optische Rekonstruktion ist dazu geeignet, den Nukleus dreidimensional darzustellen und zu betrachten. Man kann den dargestellten Raum durchwandern und sich die einzelnen Neuronen ansehen, sie zählen und könnte sie auch zeichnen. Die quantitative Erfassung der Anzahl und der Verteilung erfolgt im nächsten Schritt.

Der dritte logische Schritt auf dem Weg zu einer vollständigen Rekonstruktion ist die Überführung dieser optischen in eine funktionale Rekonstruktion wie in Abbildung 1.3(f) gezeigt. Ich werde in dieser Arbeit den Begriff funktionale Rekonstruktion für eine Rekonstruktion verwenden, bei der einzelne funktionale Elemente wie die Zellkerne oder Somata als Datensätze vorliegen und gezielt angesprochen werden können. Mit einer funktionalen Rekonstruktion ist es z.B. möglich, Zellen zu zählen, deren Größe zu bestimmen o.ä..

Als konkretes Beispiel für das Verfahren wurde ein Teil der MSO von *Meriones unguiculatus* untersucht. Alle Programme wurden jedoch so ausgelegt, dass sie auch auf beliebige andere Bereiche anwendbar sind.

Prinzipiell wäre auch die Rekonstruktion der gesamten MSO möglich gewesen, dies war lediglich mit der in unserer Arbeitsgruppe zur Verfügung stehenden Rechenkapazität in einem angemessenen zeitlichen Rahmen nicht umzusetzen (siehe Kapitel 2.3.2 auf Seite 31).

Die MSO eignet sich aus zwei Gründen für das hier vorgestellte Verfahren: Ein Grund ist die einfache Handhabbarkeit der MSO. So lassen sich Ergebnisse von automatischen Verfahren auf Grund der geringen Größe des Nukleus leicht mit manuellen Methoden vergleichen.

Der Hauptgrund für die Auswahl war aber die Aussicht, die oben geschilderten biologischen Fragestellungen einer Lösung näher zu bringen und die Konstruktion eines

neuen Funktionsmodells für die Verarbeitung von tieffrequentem Schall beim Säuger zu ermöglichen.

1.2 Gliederung

Wie oben dargestellt, ist der gesamte Rekonstruktionsprozess in mehrere Teilschritte untergliedert. Jeder dieser Teilschritte wird in einem eigenen Kapitel vorgestellt und diskutiert.

Zunächst werde ich in Kapitel 2 die histologischen und mikroskopischen Methoden beschreiben, die die Ausgangsbasis für die weitere Untersuchung darstellen.

Danach folgen mehrere technische Kapitel, in denen die Grundlagen der einzelnen Verfahrensschritte erläutert werden. In der Reihenfolge der Bearbeitung wird in Kapitel 3 zunächst das Image Mosaicing und die daran vorgenommenen Verbesserungen erläutert.

In Kapitel 4 werden die zur Erkennung der Zellkerne verwendeten Methoden erläutert. Dieser Prozess beginnt mit einer Korrektur von Farbabweichungen und gliedert sich dann in mehrere Schritte, die schließlich zu einer Extraktion der Zellkerne führen.

In Kapitel 5 beschreibe ich den von mir entwickelten Archimedes-Alignment-Algorithmus zum Ausrichten der Schnitte.

Die in Kapitel 6 erläuterte Rekonstruktion der funktionalen Elemente schließt den Rekonstruktionsprozess ab.

Im letzten Kapitel wird das Ergebnis des gesamten Verfahrens nochmals dargestellt und diskutiert. Außerdem diskutiere ich in diesem Kapitel ausführlich die möglichen Anwendungsgebiete und Erweiterungen dieses Verfahrens.

Material und Methode

Viele Arbeiten im Bereich der Bildbearbeitung sind sehr auf die Erhöhung der Effizienz bei der Auswertung fixiert und ignorieren dabei die – oftmals wesentlich wirkungsvolleren – Möglichkeiten, die eine Optimierung der histologischen Methoden, sowie der mikroskopischen Aufnahmetechnik, bieten.

In diesem Kapitel soll gezeigt werden, wie die Aufnahme so beeinflusst werden kann, dass die anschließende Mustererkennung deutlich erleichtert wird.

Dazu findet zunächst eine histologische Aufarbeitung der Präparate statt. Es kann gezeigt werden, dass die Kombination von Semidünnschnitten mit einer Nissl-Osmium-Färbung optimal geeignet ist, ein Ergebnis zu erzielen, das die spätere Mustererkennung vereinfacht. Diese Prozedur wird im ersten Teil des Kapitels kurz vorgestellt.

Auch beim Mikroskop – vor allem aber bei der digitalen Kamera – bieten sich mannigfaltige Möglichkeiten der Optimierung. Wenn man die Möglichkeiten moderner Kameras effektiv nutzt, erleichtert dies die darauf folgende Mustererkennung zusätzlich. Im zweiten Teil des Kapitels werden mehrere Möglichkeiten beschrieben, wie die Aufnahme so beeinflusst werden kann, dass Zellkerne und Somata möglichst stark hervorgehoben werden.

Den Abschluss des Kapitels bildet eine Betrachtung der Anforderungen, die diese Arbeit hinsichtlich der Hard- und Software stellt.

2.1 Histologische Methode

Die verwendeten Puffer werden im Anhang A auf Seite 163 beschrieben.

Die Präparation wurde wie folgt durchgeführt: Eine weibliche Wüstenrennmaus (*Meriones unguiculatus*, Milne-Edwards 1867) mit einem Gewicht von 49,0 g und einem Alter von ca. 7 Wochen vom *Charles River Laboratory* wurde perfundiert. Zunächst wurde das Blut durch Ringerlösung [85] und diese im Anschluss durch das Glutaraldehyd-Fixativ ersetzt. Nach der Perfusion wurde das Gehirn freigelegt und in derselben Fixativ-Lösung für 24 h nachfixiert.

Aus diesem Fixativ wurde das Gehirn zunächst in den Phosphat-Waschpuffer überführt und in diesem drei mal für je 20 min gewaschen.

Nach dem Auswaschen des Glutaraldehyds wird das Gehirn mit dem Vibratom in Schnitte von 200 μm Dicke geschnitten. Diese Schnitte werden osmiert und damit nachfixiert, indem die Probe in einer 2%igen Osmiumtetroxid-Lösung zwei Stunden lang verbleibt. Nach der Fixierung wird die Probe drei mal 20 min lang mit dem Waschpuffer nachgespült.

Danach folgt eine Entwässerung mit einer aufsteigenden Ethanolreihe. Hierzu wird die Probe je zwei mal für zehn Minuten in 50%, 70%, 90%, 96% und drei mal für zehn Minuten in 100% Ethanol entwässert. Das unvergällte Ethanol wurde mit Aqua bidest. verdünnt.

Die Proben wurden drei mal für je zwanzig Minuten in Propylenoxid gewaschen, in ein Gemisch aus Propylenoxid und Araldit (50:50) überführt und über Nacht stehen gelassen.

Am nächsten Tag wurden die Proben zwei mal für je drei Stunden in reines, frisches Araldit überführt und evakuiert. Nach der Einbettung in frischem Araldit härteten die Proben für 48 Stunden bei 60°C in einer Flacheinbettung zwischen zwei Overheadfolien aus. Die Präparate wurden mit einer Gelatine kapsel aufgenommen und in 2 μm dicke Schnitte geschnitten.

Diese Schnittdicke ist notwendig, um die Zellkerne auch über mehrere Ebenen verfolgen zu können. Die zum Vergleich angefertigten Paraffin-Einbettung mit einer Schnittdicke von 15 μm ist hierzu nicht geeignet.

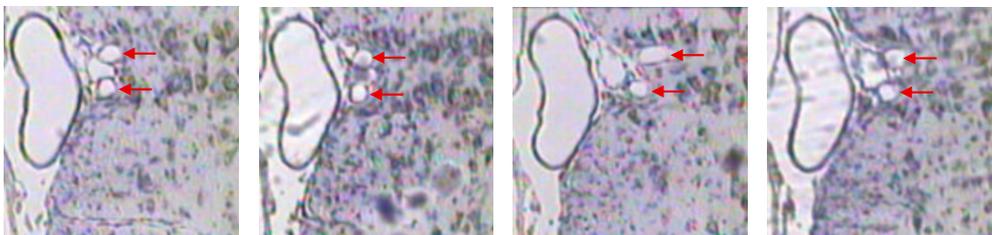


Abbildung 2.1: Die hier gezeigten Paraffin-Schnitte mit einer Dicke von 15 μm weisen eine deutlich schlechtere Qualität auf als vergleichbare Kunstharzschnitte in Abbildung 2.2. Die Pfeile markieren einander entsprechende Punkte in aufeinander folgenden Ebenen. (Maßstab: die Kantenlänge der Bilder beträgt 126 μm)

Wie in in Abbildung 2.1 dargestellt, sind bei aufeinander folgenden Paraffin-Schnitten die Zellen nur mit viel Mühe einander zuzuordnen. Die große Dicke der Schnitte führt zusammen mit den beim Schneiden auftretenden Verformungen dazu, dass selbst manuell nur schwer eine Zuordnung zu treffen ist.

Neben den nur schwer zu beseitigenden Transformationen beim Schneiden besteht außerdem die Gefahr, einen Zellkern nicht zu erfassen, da der Durchmesser eines Zellkerns normalerweise ca. 9 μm beträgt und somit bei einer ungünstigen Lage mitten in einem 15 μm -Schnitt zu liegen käme und damit außerhalb der Fokusebene liegen würde. Bedingt durch die Größe des Zellkerns ist auch die verwendete Schnittdicke von 2 μm sinnvoll. Zum einen wäre es bei einer größeren Schnittdicke, wie z.B. 4 μm

wesentlich schwerer, die Zellen einander zuzuordnen und zu anderen kommt es unweigerlich zum Verlust einiger Zellen, wenn beim Schneiden ein Schnitt verloren geht oder Falten wirft, was selbst bei erfahrenen Histologen häufiger vorkommt. Bei 2 μm Abstand hingegen, sind beim Verlust einer Schnittebene noch sämtliche Zellkerne entweder auf der folgenden oder der vorhergehenden Ebene abgebildet. Erst wenn mehrere aufeinanderfolgende Schnitte verloren gehen, können auch bei diesem Verfahren einige Zellkerne nicht mehr identifiziert werden.

Die Schnitte wurden danach mit einer Nissl-Färbung gefärbt, was zusammen mit der Osmium-Fixierung zu einem hohen Kontrast zwischen den Somata und dem Hintergrund führt und außerdem die Zellkerne ganz besonders kontrastreich hervorhebt. Durch das Kontrastieren der Zellkerne ist es vergleichsweise einfach, diese von Kapillaren und anderen Artefakten zu unterscheiden.

Bei anderen Färbemethoden ist meistens der Kontrast so gering, dass es nur schwer möglich ist, Zellen von zellähnlichen Strukturen zu unterscheiden, denn während Verschiebungen im Farbraum oder ähnliche Fehler rechnerisch zu korrigieren sind, ist ein nicht vorhandener Kontrast niemals in ein Bild hineinzurechnen.

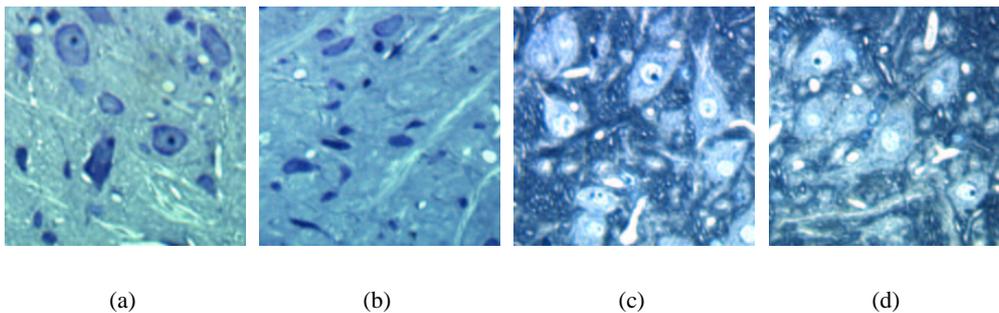


Abbildung 2.2: Vergleich der Färbung mit und ohne Osmium-Nachfixierung: In den Abbildungen (a) und (b) sind zwei Aufnahmen eines Schnittes ohne die Osmiumfixierung abgebildet. In den rechten beiden Bildern ist jeweils eine Aufnahme mit der entsprechenden Nachfixierung zu sehen. (Maßstab: die Kantenlänge der Bilder beträgt 128 μm)

In Abbildung 2.2 sind in den linken beiden Abbildungen Zellen zu sehen, die ohne eine Nachfixierung mit Osmium gefärbt wurden. Man kann deutlich sehen, dass auf der rechten Seite die Zellkerne wesentlich kontrastreicher sind und sich von den restlichen Zellen und dem Hintergrund auf Grund der Nachfixierung besser abheben.

2.2 Mikroskopische Aufnahmen

Die so vorbereiteten Schnitte wurden mit dem Axiophot - Mikroskop der Firma Zeiss aufgenommen. Als Digitalkamera diente das Modell DMC 2 (Digital Microscope Camera 2) der Firma Polaroid. Die Software der Digitalkamera lag in der Version 2.1 vor. Mikroskop und Kamera wurden mir freundlicherweise von Prof. Manfred Kössl (Johann Wolfgang Goethe-Universität Frankfurt am Main) zur Verfügung gestellt.

Das Mikroskopieren stellt einen sehr zeitaufwendigen und arbeitsintensiven Schritt dar. Bei einer vollständigen Rekonstruktion würden in einem typischen Fall über 1.000 Schnitte aufgenommen. Innerhalb jedes Schnittes müssen nochmals mehrere Einzelaufnahmen zu einem Mosaik zusammengefügt werden. Dieser hohe Aufwand macht eine präzise und vor allem reproduzierbare Aufnahmetechnik notwendig.

Um auch bei über einen längeren Zeitraum verteilt liegenden Aufnahmen immer die gleiche Qualität zu gewährleisten, wurde nach einem definierten Protokoll vorgegangen, das in Anhang B wiedergegeben ist. Es ist sehr wichtig, dass alle am Mikroskop und der Software vorgenommenen Einstellungen bei jeder Serie exakt eingehalten werden. Ändert sich zwischen zwei Aufnahmen plötzlich die Lichtintensität, die Farbverteilung, o.ä., so wird eine spätere Auswertung extrem erschwert. Jede Änderung im Strahlengang des Mikroskops führt dazu, dass sich das Farbspektrum des Bildes ändert (bei einer größeren Helligkeit wird z.B. der gesamte Farbraum zu höheren Farbwerten hin verschoben). In einem geringen Maß kann eine solche Veränderung zwar kompensiert werden (siehe Kapitel 4.2); bei größeren Abweichungen wird allerdings eine Erkennung der Zellen unmöglich. In jedem Fall erhöht eine geringere Farbkonstanz die Fehlerrate beim Erkennen der Zellen.

2.2.1 Einstellungen am Mikroskop

Die Einstellungen am Mikroskop wurden wie folgt vorgenommen und jeweils in den oben erwähnten Protokollblättern festgehalten:

Ohne dass Graufilter im Strahlengang eingesetzt waren, wurde die Intensität der Beleuchtung auf einen definierten Wert (3,25) eingestellt. Dann wurde eine 40fache Vergrößerung gewählt.

Das Präparat wurde fokussiert und das Mikroskop auf die Optik der Kamera geköhlet. Die Aperturblende wurde nach der $\frac{2}{3}$ -Regel eingestellt.

Die Auflösung eines einzelnen Bildes wurde auf 1600x1200 Pixel festgelegt; die Farbtiefe betrug 24 Bit. Prinzipiell ist es mit der Kamera auch möglich, mit der doppelten Farbtiefe, d.h. 48 Bit zu arbeiten. Dies ist jedoch unter verschiedenen Gesichtspunkten problematisch: Zunächst würde ein solches Verfahren die Nachvollziehbarkeit der Methoden einschränken, da nicht jede Kamera eine Farbtiefe von 48 Bit erlaubt. Zum anderen sind die meisten Programme nicht darauf ausgerichtet, Bilder mit 16 Bit pro Farbkanal zu verarbeiten. Der entscheidende Grund ist allerdings der immense Platzbedarf der Bilder. Jedes Bild benötigt dann den doppelten Speicherplatz auf der Festplatte. Nach der auf Seite 32 beschriebenen Berechnung würde eine typische Serie somit mehr als 80 GB auf der Festplatte beanspruchen. Dieses Volumen verursacht nicht nur Speicherprobleme, sondern verlangsamt vor allem auch sämtliche Verarbeitungsprozesse, da bei jedem Schritt die doppelte Datenmenge in den Arbeitsspeicher geladen und verarbeitet werden muss. Deshalb habe ich mich für eine Aufnahme mit geringerer Farbtiefe entschieden.

In dem Präparat wurde als letzter Vorbereitungsschritt ein Ausschnitt gewählt, in dem einige Somata lagen – wenn möglich zumindest eines davon vereinzelt. Von dieser Stelle wurde eine Probeaufnahme angefertigt und für die Bestimmung einer geeigneten Kameraeinstellung (siehe unten) verwendet.

2.2.2 Anpassen der Aufnahmetechnik

Im Vergleich zu dem in meiner Diplomarbeit [13] verwendeten „Colour Coolview“-System an der HUB (Berlin) ist die DMC 2 deutlich leistungsfähiger; allerdings teilweise auch wesentlich komplizierter zu bedienen. Neben der fehlenden Skriptsprache zum Automatisieren der Aufnahmebedingungen machte sich die Tatsache, dass bei der DMC 2 die erweiterten Funktionen nur schlecht oder gar nicht dokumentiert sind, negativ bemerkbar. So war es mit großem Aufwand verbunden, durch Tests und Gespräche mit dem technischen Support von Polaroid eine qualitativ zufriedenstellende Aufnahme zu erhalten.

In meiner Diplomarbeit hatte ich über die Skriptsprache der Kamera mehrere Aufnahmen des gleichen Ausschnitts aufgenommen, von diesen den Mittelwert gebildet und so das Signal/Rausch-Verhältnis deutlich verbessert. Diese Funktionalität ist in die Software der DMC 2 bereits integriert, so dass bei jeder Aufnahme automatisch vier Bilder aufgenommen und gemittelt wurden. Dennoch wäre eine Skriptsprache sehr hilfreich gewesen, da es so einfach möglich gewesen wäre, von jedem Ausschnitt mehrere Aufnahmen unter unterschiedlichen Bedingungen anzufertigen.

Eine sehr komplexe Aufgabe war die Anpassung der Farbverteilung. Ziel einer solchen Anpassung ist es, möglichst viele Informationen mittels der drei zur Verfügung stehenden Kanäle abzubilden. Ist ein Kanal zu empfindlich, wird eventuell nur der obere Teil des Histogramms genutzt. Ein Farbunterschied (Kontrast), der im Normalfall zehn Farbwerte beträgt, würde in diesem Fall auf die Hälfte, d.h. auf fünf Farbwerte reduziert. Dieser Fall ist in Abbildung 2.3 links dargestellt.

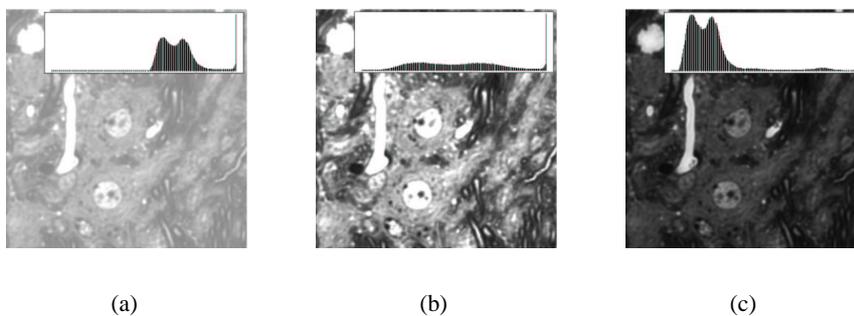


Abbildung 2.3: Vergleich von unterschiedlichen Helligkeits/Kontrast-Einstellungen: In den Bildern ist oben rechts jeweils das entsprechende Histogramm aufgetragen (Auf der x-Achse der Farbwert, auf der y-Achse die Häufigkeit). In Bild (a) ist die Helligkeit zu hoch: Das Histogramm wird nur in der oberen Hälfte genutzt. In Bild (b) ist dagegen der Kontrast zu hoch, so dass Zellkerne und Kapillaren nicht mehr zu unterscheiden sind. Nur in Bild (c) sind beide Parameter korrekt eingestellt.

Ist hingegen der Kontrast zu hoch, kann nicht mehr das gesamte Histogramm abgebildet werden und ein Teil der Informationen geht verloren. Dies ist in Abbildung 2.3 in der Mitte gut zu erkennen. Der Kontrast ist dort so hoch, dass der Farbwert der Zellkerne über 255 liegt und dadurch nicht mehr von den Kapillaren zu unterscheiden ist.

Dieser Schritt wird bei älteren Geräten häufig einfach durch eine Modifikation der Beleuchtung ermöglicht. Dort lässt sich durch eine separate Veränderung der Beleuchtungsintensität für die Kanäle Rot, Grün und Blau eine optimale Ausleuchtung erreichen.

Die DMC 2 bietet umfangreiche Möglichkeiten über die mitgelieferte Software den Farbraum zu manipulieren. Eine Manipulation des Farbspektrums kann auf zwei Arten durchgeführt werden:

- Durch eine *Transformationsmatrix*
- Durch eine *Lookup-Table*

Beide Methoden wurden in dieser Arbeit benutzt und werden im folgenden beschrieben.

Anpassung der Transformations-Matrix

Bei der Transformationsmatrix handelt es sich um eine Datei, wie in Listing 1 auf der gegenüberliegenden Seite gezeigt. Sie kann dazu verwendet werden, die Informationen der einzelnen Kanäle miteinander zu verrechnen. Hier wird die Transformationsmatrix dazu benutzt, die Farbinformationen zu strecken und auf zwei Kanäle aufzuteilen. So kann ein hoher Kontrast erreicht werden, ohne dass wie in Abbildung 2.3(b) das obere Ende des Histogramms verloren geht: Dieses wird in einem anderen Kanal abgebildet.

In dem linken Beispiel in Listing 1 würden sich die Farbwerte des Zielbildes r_o , g_o und b_o (roter, grüner und blauer Kanal) aus den gemessenen Farbwerten r_i , g_i und b_i wie folgt herleiten:

$$r_o = 1.0801 \cdot r_i + -0.0737 \cdot g_i - 0.0064 \cdot b_i \quad (2.1)$$

$$g_o = 0.0148 \cdot r_i + 0.9648 \cdot g_i + 0.0204 \cdot b_i \quad (2.2)$$

$$b_o = -0.0402 \cdot r_i - 1.1533 \cdot g_i + 2.1935 \cdot b_i \quad (2.3)$$

Die Verwendung einer Transformationsmatrix ist dann sinnvoll, wenn man entweder die Farbgebung des Bildes beeinflussen möchte (z.B. um einen Rot- oder Blaustich herauszufiltern) oder aber die Informationen eines Kanals mit denen eines anderen verrechnen will. In diesem Beispiel wird der blaue Kanal z.B. vor allem in den Bereichen besonders betont, in denen der grüne Farbkanal nur einen kleinen Wert besitzt.

In meinem Fall wurde diese Matrix dazu verwendet, den roten und den grünen Farbkanal jeweils durch eine Kombination aller drei Kanäle zu ersetzen. Dies bietet – wie weiter unten beschrieben – die Möglichkeit, unterschiedliche Bereiche des Farbspektrums mit zwei verschiedenen Lookup-Tables hervorzuheben. Möchte man keine Veränderung vornehmen, verwendet man die in Listing 1 rechts gezeigte Transformationsmatrix.

Um die gewünschte KorrekturmatriX zu berechnen, wurde wie folgt vorgegangen: Die drei Kanäle rot, grün und blau sind eng miteinander korreliert (siehe Abbildung 2.4 auf der nächsten Seite) und enthalten somit redundante Informationen. Damit

Listing 1 Die von dem Programm vorgegebenen Dateien CRT.mtx(links) und Identity.mtx (rechts)

```
# DMC 2p Matrix for ESCAPEDE # DMC 1e Matrix for ESCAPEDE
# 12/14/2000 # 2/26/99
# Polaroid Corporation John Francis # Polaroid Corporation Andrew Juenger
# CRT Colors # Identity matrix
MATRIX coefficients 3 3 MATRIX coefficients 3 3
1.080150755 -0.073717373 -0.006433382 1.0000000 0.0000000 0.0000000
0.014805993 0.964831767 0.020362241 0.0000000 1.0000000 0.0000000
-0.040197811 -1.153302189 2.1935 0.0000000 0.0000000 1.0000000
```

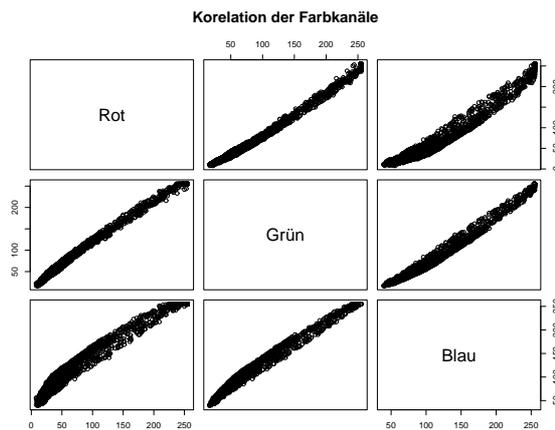


Abbildung 2.4: Korrelation der einzelnen Farbkanäle eines Bildes ($n=10.000$). Der Korrelationskoeffizient (R^2) liegt in allen Fällen über 0.97. Auf den Achsen ist jeweils der Farbwert eines Kanals aufgetragen.

besteht eine Möglichkeit, die Farbsensitivität zu erhöhen, indem man auf einen Teil dieser Informationen verzichtet und dafür einen anderen Bereich des Farbspektrums betont.

Ausgangsmaterial ist eine ausreichend große Punktemenge von Pixeln, die willkürlich aus mehreren Bildern der Serie ausgewählt wurde. Aus Gründen der Handhabbarkeit wurde die Menge der Pixel auf $n = 10.000$ beschränkt. Zunächst werden die Farbkanäle, ähnlich wie bei der Farbkorrektur in Kapitel 4.2 beschrieben, normiert:

$$r'_i = \frac{r_i}{r_i + g_i + b_i} \quad (2.4)$$

$$g'_i = \frac{g_i}{r_i + g_i + b_i} \quad (2.5)$$

$$b'_i = \frac{b_i}{r_i + g_i + b_i} \quad (2.6)$$

wobei r_i , g_i und b_i den Farbwerten (rot, grün und blau) der Pixel $i = 1..n$ entsprechen. Die berechneten Werte stellen den relativen Anteil des jeweiligen Farbkanals an der Gesamtintensität dar. Von den so erhaltenen Werten wird jeweils der Kehrwert des arithmetischen Mittels gebildet:

$$\bar{r}' = \frac{n}{\sum_{i=1}^n r'_i} \quad (2.7)$$

$$\bar{g}' = \frac{n}{\sum_{i=1}^n g'_i} \quad (2.8)$$

$$\bar{b}' = \frac{n}{\sum_{i=1}^n b'_i} \quad (2.9)$$

Diese Mittelwerte werden wiederum in Relation zueinander gesetzt und ergeben so Gewichtungsfaktoren, die dazu verwendet werden können, einen neuen Kanal zusammenzusetzen, den die drei RGB-Kanälen antiproportional ihrer mittleren Intensität beeinflussen:

$$r'' = \frac{\bar{r}'}{\bar{r}' + \bar{g}' + \bar{b}'} \quad (2.10)$$

$$g'' = \frac{\bar{g}'}{\bar{r}' + \bar{g}' + \bar{b}'} \quad (2.11)$$

$$b'' = \frac{\bar{b}'}{\bar{r}' + \bar{g}' + \bar{b}'} \quad (2.12)$$

Der Effekt dieser Korrektur ist in Abbildung 2.5 dargestellt. In Abbildung 2.5(a) ist die ursprüngliche Verteilung der normalisierten Werte r'_i , g'_i und b'_i zu sehen. Daneben ist dieselbe Verteilung gezeigt, wobei die Werte zuvor mit den Faktoren r'' , g'' und b'' multipliziert wurden. Man sieht deutlich, dass sich die Mittelwerte der Verteilungen einander genähert haben. Während zuvor die Mittelwerte mit 0,21, 0,29 und 0,54 deutlich auseinander lagen, liegen nach dem Normalisieren alle Mittelwerte bei 0,10.

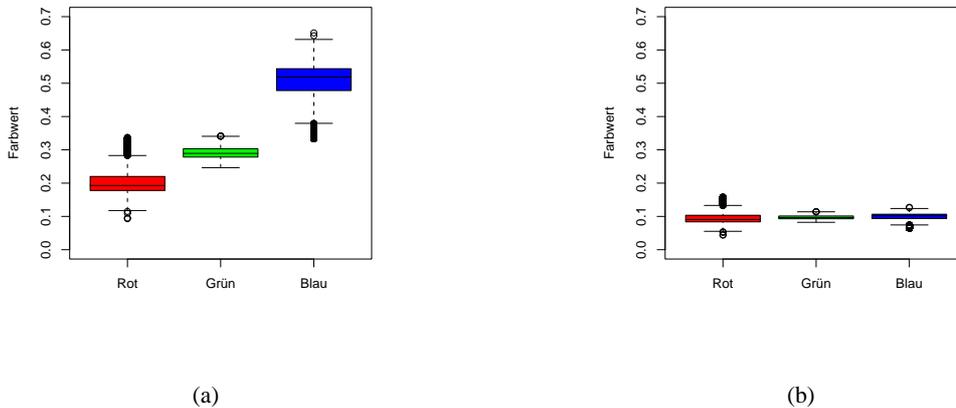


Abbildung 2.5: Boxplot der Farbwerte vor der Transformation (a) und nach Multiplikation mit den Ausgleichsfaktoren (b)

Mittels der Korrekturfaktoren kann nun ein neuer, vierter Farbkanal erzeugt werden. Dieser Kanal, hier als rgb bezeichnet, berechnet sich wie folgt:

$$rgb_i = r_i r'' + g_i g'' + b_i b'' \quad (2.13)$$

Auf Grund der von mir verwendeten Bilder ergaben sich die folgenden Werte:

$$r'' = 0,4700224 \quad (2.14)$$

$$g'' = 0,3340416 \quad (2.15)$$

$$b'' = 0,1959360 \quad (2.16)$$

Damit ist die Korrekturmatrix wie in Listing 2 gezeigt zu gestalten.

Listing 2 Die fertige Korrekturmatrix

#	DMC	ie	Matrix	for	ESCAPADE
#	2/26/99				
#	Oliver				
#	Bossert				
#	Farbausgleichsmatrix				
MATRIX	coefficients	3		3	
0.4700224		0.4700224		0.0000000	
0.3340416		0.3340416		0.0000000	
0.195936		0.195936		1.0000000	

In dieser Matrix sind die Kanäle Rot und Grün durch den neuen *rgb*-Kanal ersetzt. Bis zu diesem Punkt wären die beiden Kanäle identisch, was einen Informationsverlust bedeutete. Erst durch die folgende Behandlung mittels der Lookup-Tables kann ein im Resultat wesentlich verbessertes Ergebnis erreicht werden.

Der Vorteil liegt vor allem darin, dass der Informationsgehalt der Einzelkanäle zwar reduziert, allerdings nicht vollkommen vernichtet ist und dieser neue Kanal nun unterschiedlich gewichtet dargestellt werden kann.

Anpassung der Lookup-Tables

Eine Lookup-Table ist eine Tabelle mit den diskreten Werten einer Funktion, die die Transformation der gemessenen Werte in den Farbraum von 0 bis 255 für die drei Farbkanäle definiert.

Der große Unterschied zwischen einer Beeinflussung des Bildes auf dieser Ebene und einer später erfolgenden Nachbearbeitung liegt darin, dass die Messwerte der Kamera mit einer höheren Präzision (zumindest als 16-Bit-Wert) vorliegen, während alle späteren Transformationen auf der Basis von 8-Bit-Pixeln durchgeführt werden, was zu Ungenauigkeiten, Rundungsfehlern und letztlich zu einem Informationsverlust führt [11].

Ziel der Lookup-Table ist es immer, bestimmte Farbbereiche des Bildes zu betonen und andere dafür zu unterdrücken. Bei dieser Serie war es das Hauptanliegen, den farblichen Kontrast zwischen Soma und Hintergrund einerseits und Soma und Zellkern andererseits möglichst groß zu gestalten – ohne dabei jedoch mit der Farbgebung des Zellkerns in einen solch hohen Bereich zu gelangen, dass sich der Zellkern nicht mehr ausreichend von den Kapillaren unterscheiden lässt.

In diesem Fall wurde der Effekt – wie weiter unten beschrieben – mit zwei einfachen linearen Transformationen der Farbkanäle grün und rot erreicht; beiden wurde

durch die Transformationsmatrix der oben definierte *rgb*-Kanal zugewiesen. Das die Verwendung einer linearen Lookup-Table hier zum gewünschten Effekt führt, ist auf die hohe Farbvariabilität zwischen den einzelnen Bildern zurückzuführen. Wenn die Variabilität der Färbung geringer ausfällt, wird man die Lookup-Table eher durch eine nichtlineare, dynamisch angepasste Funktion gestalten oder sie semi-automatisch an die Daten anpassen.

Welche Kombination von Transformationsmatrix und Lookup-Table für einen bestimmten Datensatz geeignet ist, lässt sich unter Verwendung der weiter unten beschriebenen Zellprofile einfach beurteilen. Eine Kombination ist dann gut geeignet, wenn der Kontrast zwischen den zu trennenden Strukturen maximal ist, ohne dass dabei wichtige Werte außerhalb der Spanne von 0 bis 255 liegen und deshalb nicht mehr korrekt dargestellt werden können.

Die nächsten beiden Abschnitte gehen auf diese beiden Möglichkeiten der Lookup-Table-Manipulation näher ein; abschließend werde ich dann darlegen, warum in diesem Fall die verwendeten Lookup-Tables eingesetzt wurden.

Vorgehen zum Erstellen einer Lookup-Table

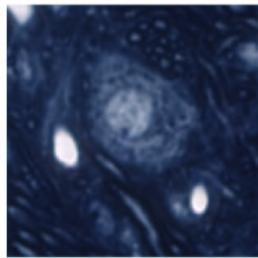
Die von dem Programm vorgegebenen Lookup-Tables sind im wesentlichen darauf ausgerichtet, das Bild für den Betrachter möglichst ansprechend zu gestalten. Dies ist – wie weiter unten gezeigt – eine vollkommen andere Zielsetzung, als diejenige, die für eine Optimierung des Bildes für die Analyse mit dem Computer notwendig ist. Deshalb ist es für jede Serie notwendig, eine eigene Lookup-Table zu erstellen.

Ausgangsmaterial für das Erstellen der Lookup-Table ist ein Probestbild ohne Farbveränderung. Verwendet man eine Transformationsmatrix, so sollte diese eingespielt werden, ansonsten wählt man in dem Programm sowohl für die Transformationsmatrix, als auch für die Lookup-Table die Einstellung `Identity`. Das nun aufgenommene Bild ist die Basis für das Erstellen der Lookup-Table.

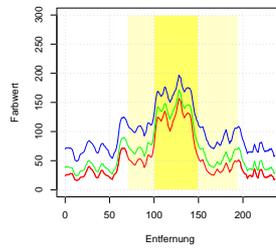
In Abbildung 2.6 sind drei unterschiedliche Lookup-Tables und ihre Auswirkungen dargestellt. In der oberen Reihe links wurde eine lineare Lookup-Table ohne Transformationsmatrix verwendet. In der mittleren Reihe ist die von Polaroid als `Tone` bezeichnete Lookup-Table (die Voreinstellung der Kamera) in Kombination mit der weiter oben dargestellten CRT-Matrix zu sehen. In der unteren Reihe ist eine optimierte Lookup-Table – ebenfalls ohne Transformationsmatrix – dargestellt.

Auf den ersten Blick sieht die Abbildung in der unteren Reihe eher kontrastarm aus und macht einen schlechten Eindruck.

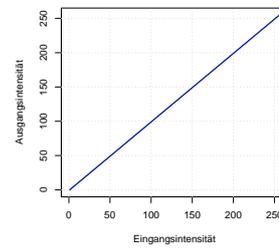
Hier kommt der oben angesprochene Unterschied zwischen der Sichtweise eines Menschen und eines Computers zum Tragen. Für den menschlichen Betrachter mit einem sehr leistungsfähigen Gehirn aber schwachen Augen ist es wichtig, ein möglichst helles Bild zu erhalten. Dabei ist ein Kontrastverlust von einigen wenigen Farbstufen nebensächlich, da wir dank unserer Abstraktionsfähigkeit in der Lage sind, einen Zellkern von einer Kapillare zu unterscheiden. Eine Auswertung im Computers ist zwar Kontraste betreffend sehr sensitiv – so kann ein reines Schwarz mit dem Farbwert 0 leicht von einem etwas weniger dunklen Schwarz mit dem Farbwert 1 unterschieden werden – jedoch ist es relativ schwer, zu entscheiden, ob ein weißes, rundes Objekt eine Kapillare oder ein Zellkern ist.



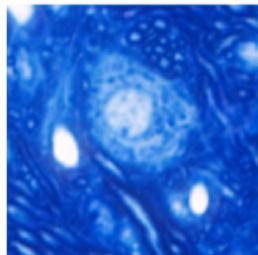
(a)



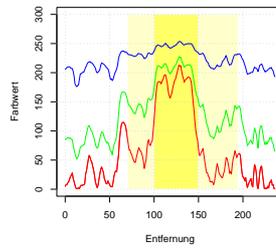
(b)



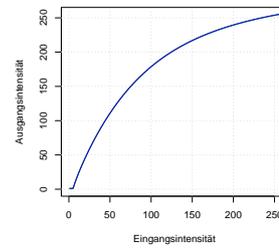
(c)



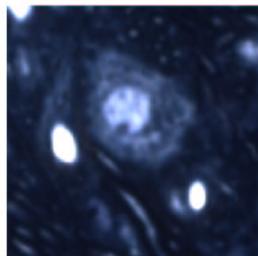
(d)



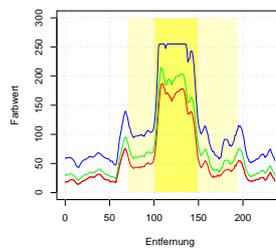
(e)



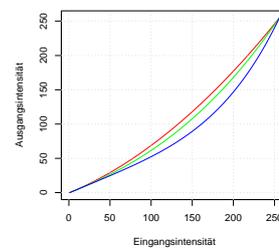
(f)



(g)



(h)



(i)

Abbildung 2.6: Bildausschnitte, Lookup-Tables und das jeweilige Profil einer Zelle: Die linke Spalte zeigt jeweils eine mikroskopische Aufnahme, die durch die rechts dargestellte Lookup-Table entstanden ist. Die mittlere Spalte zeigt das Profil der jeweiligen Zelle mit der Entfernung in Pixeln auf der x- und dem Farbwert der drei Kanäle auf der y-Achse. In (a) bis (c) ist eine lineare Lookup-Table, in (d) bis (f) die Standardeinstellung der Kamera und in der unteren Zeile eine optimierte Lookup-Table zu sehen. Die x-Achse repräsentiert dabei den zu transformierenden, die y-Achse den transformierten Wert.

Deshalb ist die „Mensch-optimierte“ Lookup-Table in der Mitte eher an eine logarithmische Kurve angenähert, bei der vor allem die dunklen Stellen des Bildes heller werden, während bei der „Computer-optimierten“ Kurve im unteren Bereich der Abbildung der genau gegenteilige Effekt erzielt wird.

Die Entscheidung, welche Lookup-Table Verwendung findet, wird auf Grundlage des Profil-Bildes einer oder mehrerer Zellen getroffen. Das Erstellen der Profile wird auf Seite 23 im übernächsten Abschnitt über das semi-automatische Erstellen von Lookup-Tables näher beschrieben.

Die in der mittleren Spalte von Abbildung 2.6 gezeigten Profil-Bilder werden erzeugt, indem eine Linie durch die Zelle gezogen wird und die Intensitäten der drei Farbbänder im Verlauf dieser Linie aufgetragen werden. Die hellgelbe Markierung im Hintergrund zeigt die Ausdehnung des Somas, die etwas dunklere die Abgrenzung des Zellkerns.

Das so erstellte Profil kann nun entweder durch die manuelle Auswahl einer Funktion oder durch ein semi-automatisches Kalibrieren ausgewertet werden. Die semi-automatische Kalibrierung wird im übernächsten Abschnitt erläutert.

In Abbildung 2.6 sind den einzelnen Lookup-Tables unterschiedliche Funktionen zugewiesen und deren Auswirkung auf den Kontrast entlang des Profils dargestellt.

In der oberen Reihe wurde eine lineare Lookup-Table verwendet. Hier ist zwar bereits gut eine Abgrenzung zwischen Zellkern und Soma und im Ansatz auch zwischen Soma und Umgebung zu erkennen, jedoch sind die Kontraste relativ schwach ausgeprägt, da nur die untere Hälfte des Farbspektrums ausgenutzt wird.

Durch die in der mittleren Zeile gezeigte werksseitig vordefinierte Transformation werden zwar die Kontraste zwischen Soma und Kern für den grünen und vor allem den roten Kanal verbessert; allerdings ist der Kontrast zwischen Soma und Umgebung immer noch schwach. Der blaue Kanal ist durch die Transformationsmatrix praktisch jeglicher Kontrastinformation beraubt.

Mit der optimierten Lookup-Table unten ist eine sehr gute Abtrennung an beiden Grenzen erreicht. Diese Verbesserung entsteht, wenn der helle Bereich, in dem sich die beiden Grenzen befinden, selektiv gestreckt wird. Die Berechnung der entsprechenden Funktionen wird im nächsten Kapitel erläutert.

Berechnung einfacher Lookup-Tables

Bei der Berechnung der Lookup-Tables müssen vier Fälle unterschieden werden. Im einfachsten Fall ist der Kanal in dem gegebenen Zustand zufriedenstellend und man behält die lineare Standard-Lookup-Table bei.

Die drei anderen Fälle unterscheiden sich nach der Position des Bereiches, der gestreckt werden soll. Befindet sich der zu streckende Bereich am Rand der Skala, bietet sich eine logarithmische oder eine polynome Transferfunktion an; befindet sich der interessante Bereich in der Mitte des Farbspektrums, ist eine modifizierte sigmoide Funktion geeignet.

Die Polynom-Funktionen sind sehr einfach zu berechnen, indem die entsprechende Funktion (k ist einer beliebige Konstante; in Abbildung 2.7(a) sind für k die Werte 2,

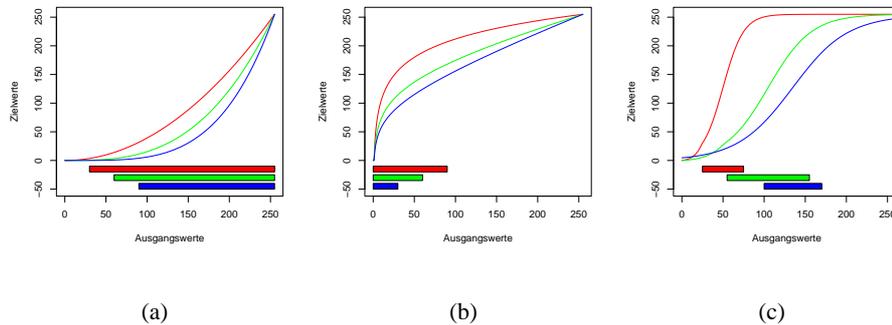


Abbildung 2.7: Darstellung der drei unterschiedlichen Formen von Lookup-Tables: In Abb.(a) ist eine Lookup-Table mit unterschiedlichen Polynom-Funktionen zu sehen. In Abb. (b) ist die Transferfunktion logarithmisch und in Abb. (c) basiert sie auf der sigmoiden Funktion. Die Balken unterhalb der Funktionskurve geben die Bereiche an, die im Ausgabebild besonders betont werden sollen. Die genaue Parameterisierung der Kurven ist im Text erläutert.

3 und 4 eingesetzt) auf den Bereich von 0 bis 255 normiert wird:

$$f_1(x) = \frac{x^k}{255^{k-1}} \quad (2.17)$$

Ganz ähnlich kann man mit der logarithmischen Funktion verfahren:

$$f_2(x) = \frac{\log x \cdot 255}{\log 255} \quad (2.18)$$

In der Praxis wirken sich jedoch sowohl die Funktionen der Form x^k als auch die logarithmischen Funktionen zu extrem auf das Bild aus. Deshalb bietet es sich an, mit den Funktionen einen – gegebenenfalls gewichteten – Mittelwert mit der Geraden $x = y$ zu bilden:

$$f_3(x) = \frac{f_2(x) + x \cdot (l - 1)}{l} \quad (2.19)$$

Dabei ist l eine beliebige Konstante. Durch Einsetzen von 1, 1.5 und 2.5 in diese Gleichung sind die Kurven in Abbildung 2.7(b) entstanden.

Selbstverständlich lassen sich auch logarithmische Funktionen für ein Strecken des oberen und die Polynom-Funktionen für eine Betonung des unteren Randes einsetzen. Dazu müssen lediglich die beiden Spalten der Lookup-Table wie folgt umdefiniert werden:

$$x' = |255 - x| \quad (2.20)$$

$$f'(x) = |255 - f(x)| \quad (2.21)$$

Etwas komplizierter ist das Erstellen der Lookup-Table, wenn sich der interessante Bereich in der Mitte des Farbspektrums befindet. Ein gebräuchlicher Weg besteht darin, eine einfache lineare Funktion über den interessanten Bereich zu definieren und die Randbereiche zu ignorieren (für den Bereich von 50 bis 150 wäre die entsprechende Gerade $f(x) = 2,55x - 127,5$, die man durch Einsetzen der Punkte (50/0) und (150/255) in die Punktsteigungsform erhält).

Bei dieser Transformation gehen jedoch die Randbereiche vollständig verloren. Besser geeignet ist hier die sigmoide Funktion, die in ihrer Mitte linear verläuft; in den Randbereichen die Werte jedoch trotzdem nicht abschneidet, sondern lediglich staucht.

Die normale sigmoide Funktion hat die Form

$$s(x) = \frac{255}{1 + e^{-\frac{m-x}{s}}} \quad (2.22)$$

Dabei ist m der mittlere Wert der Funktion und s entspricht der Steigung. Der Mittelwert zwischen den beiden Grenzwerten entspricht m , während die Steigung s so zu wählen ist, dass der Bereich gut wiedergegeben wird. In Abbildung 2.7(c) auf der vorherigen Seite ist die Steigung als $\frac{1}{8}$ der Differenz zwischen den beiden Grenzwerten definiert.

Bei der Verwendung von Gleichung 2.22 entsteht das Problem, dass die Kurve aus dem Definitionsbereich „herauswandert“ und relevante Bereiche abgeschnitten werden. Um dies zu vermeiden, werden die Funktionswerte in einem der Randbereiche wie folgt transformiert:

$$t(x) = \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1) + y_1 \quad (2.23)$$

$$s_2(x) = s(t(x)) \quad (2.24)$$

$$s_3(x) = \frac{s(x) \cdot x + s_2(x) \cdot (max - x)}{max} \quad (2.25)$$

Dabei sind die Variablen x_1 , y_1 , x_2 , y_2 und max wie folgt belegt: Wenn sich der Mittelwert der beiden Grenzwerte unterhalb der Mitte der Farbskala (127.5) befindet, sind die Werte definiert als:

$$x_1 = 0 \quad (2.26)$$

$$y_1 = \text{Mittelwert} - 255 \quad (2.27)$$

$$x_2 = \text{Untergrenze} \quad (2.28)$$

$$y_2 = \text{Untergrenze} \quad (2.29)$$

$$max = \text{Untergrenze} \quad (2.30)$$

Befindet sich der Mittelwert von Unter- und Obergrenze hingegen oberhalb von 127.5, so gilt die folgende Definition:

$$x1 = \text{Obergrenze} \quad (2.31)$$

$$y1 = \text{Obergrenze} \quad (2.32)$$

$$x2 = 255 \quad (2.33)$$

$$y2 = 2 \cdot \text{Mittelwert} \quad (2.34)$$

$$\text{max} = \text{Obergrenze} \quad (2.35)$$

Die Eigentliche sigmoide Funktion, wie in Abbildung 2.7(c) gezeigt, ist dann eine stückweise definierte Funktion:

$$f(x) = \begin{cases} \text{Mittelwert} < 127,5 \wedge x < \text{Untergrenze} & s_3(x) \\ \text{Mittelwert} > 127,5 \wedge x > \text{Obergrenze} & s_3(x) \\ \text{sonst} & s(x) \end{cases} \quad (2.36)$$

Dabei gilt zu beachten, dass sich die Definition der Variablen der Funktion $s_3(x)$ entsprechend den beiden zuvor definieren Gleichungsblöcken ändert.

Mit diesen Funktionstypen ist es möglich, jeden Bereich des Farbspektrums spezifisch zu betonen. Will man jedoch nicht nur einen, sondern mehrere Bereiche des Spektrums mit einem höheren Kontrast versehen, benötigt man eine komplexere Funktion. Das Erstellen einer solchen Funktion wird im nächsten Abschnitt erläutert.

Erstellen einer optimalen Lookup-Table

Das in diesem Abschnitt beschriebene Verfahren erlaubt es, eine für einen bestimmten Zweck optimierte Lookup-Table semi-automatisch zu berechnen. Dies sollte normalerweise die Methode der Wahl sein, da so die Mustererkennung deutlich verbessert wird.

Bei den hier verwendeten Schnitten musste die Lookup-Table jedoch manuell erstellt werden, da die farbliche Abweichung zwischen den einzelnen Schnitten (siehe Abschnitt 4.2 auf Seite 60) bei einer optimierten Lookup-Table dazu geführt hätte, dass einige Bilder zwar optimal abgebildet werden, andere jedoch nicht mehr auszuwerten sind.

Dieses Verfahren ist demnach nur dann anwendbar, wenn entweder die Färbung eine geringe Variationsbreite aufweist oder aber der in Abschnitt 4.2 beschriebene Algorithmus zur Farbkonstanz vor der Anwendung der Lookup-Table angewendet würde. Diese wäre dann optimal, läßt sich allerdings nur dann umsetzen, wenn über ein API (Application Programming Interface) ein direkter Zugriff auf die Hardware der Kamera möglich ist.

Solange nicht entweder die Variationsbreite beim Färben der Schnitte reduziert werden kann oder der Hersteller einer Kamera seine Schnittstellen offen legt, ist es notwendig, die Lookup-Table in geringerem Maße zu optimieren und somit einen höhere Variationsbreite zu berücksichtigen.

Um eine optimierte Lookup-Table zu erhalten, muss zuerst das Ziel, für das diese optimiert werden soll, definiert werden. In diesem Fall ist es das Ziel der späteren Mustererkennung, vier unterschiedliche Bereiche voneinander zu unterscheiden:

- Hintergrund
- Soma
- Nukleus
- Kapillare

Dabei ist es jeweils besonders schwer, die aufeinander folgenden Kompartimente voneinander zu unterscheiden. Deshalb ist es notwendig, dass der Kontrast zwischen Hintergrund und Soma, zwischen Soma und Nukleus und zwischen Nukleus und Kapillare besonders groß ist.

Ausgangsmaterial sind Bilder mit einer linearen Lookup-Table. In diesen Bildern werden Zellen, wie in Abbildung 2.6 auf Seite 19 gezeigt, im Profil analysiert.

Zur Analyse der Zellprofile wurde das Programm ImageJ vom NIH (National Institute of Health) [50] verwendet. Mit diesem Programm wurde das Bild zunächst in seine Farbkanäle zerlegt

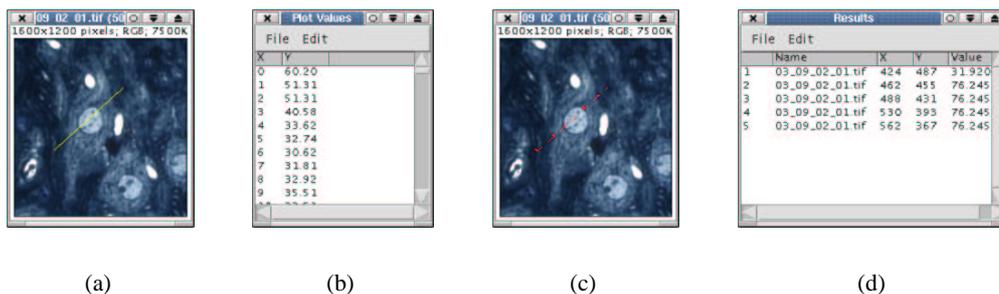


Abbildung 2.8: In dieser Bilderfolge ist dargestellt, wie ein Profil der Zelle mit dem Programm ImageJ erstellt wird. Zunächst wird eine Linie durch die Zelle gezogen (a). Das Farbprofil (b) entlang der Linie wird in einer Datei gespeichert. Dann werden manuell die Übergänge zwischen Hintergrund, Soma und Nukleus markiert (rote Punkte in (c)), so dass den einzelnen Zonen Bereiche des Profils zugeordnet werden können (d).

Dann wird für jeden Farbkanal die in Abbildung 2.8 gezeigte Prozedur durchgeführt. Zunächst wird eine Linie so durch die Zelle gezogen, dass der Anfang und das Ende der Linie im Hintergrund liegen und in der Mitte sowohl Bereiche des Nukleus als auch des Somas geschnitten werden. Dann wird ein Farbprofil, d.h. eine Verteilung der Grauwerte für den entsprechenden Kanal entlang der Linie erstellt und in einer Datei gespeichert.

Als letztes werden manuell die Übergänge zwischen Hintergrund, Soma und Nukleus markiert, so dass den einzelnen Zonen Bereiche des Profils zugeordnet werden können.

Die gespeicherten Dateien werden von einem kleinen Programm ausgewertet. Dieses Programm bestimmt die Länge der einzelnen Bereiche auf Grundlage der manuell

markierten Punkte und berechnet für jeden Bereich die Mittelwerte der einzelnen Farbk채n채le.

Die Analyse geschieht in mehreren Schritten, wie in Abbildung 2.9 gezeigt. Zun채chst wird ein Bild mit einer linearen Lookup-Table aufgenommen und gespeichert. Dann werden – wie in Abbildung 2.8 auf der gegenOberliegenden Seite gezeigt – fOr mehrere Zellen die Profile analysiert.

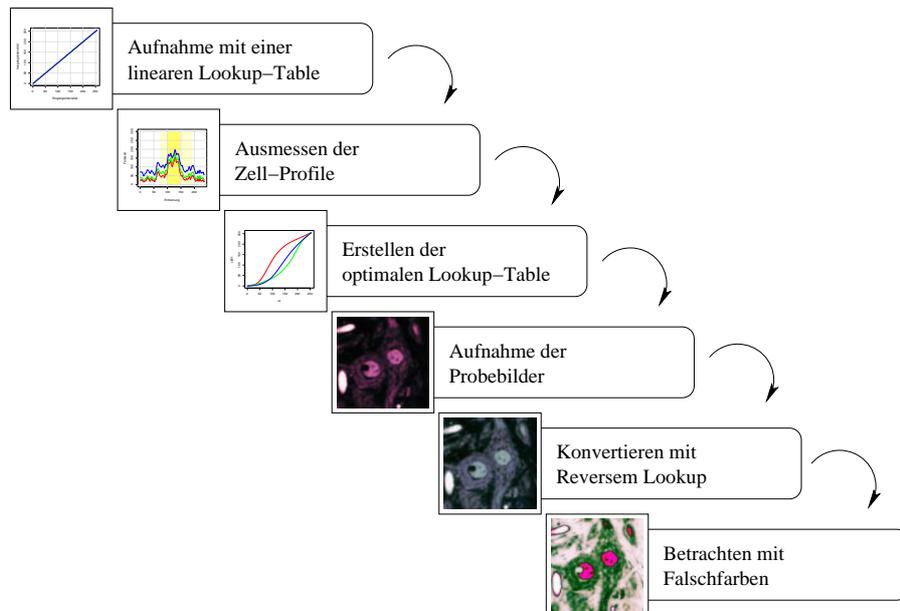


Abbildung 2.9: Flusschema zum Erstellen einer optimalen Lookup-Table: Zun채chst wird ein Bild mit einer linearen Lookup-Table aufgenommen und davon das Profil bestimmt. Auf dieser Basis wird, wie im Text beschrieben, eine optimale Lookup-Table berechnet. Nach deren Anwendung kann das Ergebnis mittels Falschfarben oder einem reversen Lookup beurteilt werden.

Die durch die Profile gewonnenen Daten werden von einem anderen Programm analysiert, welches zun채chst bestimmt, welcher Bereich des Profil-Plots welcher Klasse zugeordnet wird, indem es die Entfernung des markierten Punktes zum Anfang der Linie berechnet. Dann wird von allen Werten in diesem Bereich der Mittelwert gebildet. Dabei sind x_0/y_0 die Koordinaten des ersten Punktes und x_n/y_n sowie x_{n-1}/y_{n-1} die Ober- und Unterschranke der Klasse n . Die Daten des Profils sind in dem Feld `profildata[]` abgelegt.

$$i_0 = \sqrt{(x_{n-1} - x_0)^2 + (y_{n-1} - y_0)^2} \quad (2.37)$$

$$i_1 = \sqrt{(x_n - x_0)^2 + (y_n - y_0)^2} \quad (2.38)$$

$$mean = \frac{\sum_{i=i_0}^{i_1} profildata[i]}{i_1 - i_0} \quad (2.39)$$

Die Mittelwerte der einzelnen Klassen kOnnen dann in einem Diagramm wie in Abbildung 2.10 auf der n채chsten Seite gezeigt, berechnet werden. Bei dieser Darstellung

nutzen alle Kanäle das Spektrum nur sehr schlecht aus. Der Kontrast zwischen Hintergrund und Nukleus liegt für alle Kanäle bei maximal 175 Pixeln. Ziel ist es nun, die Unterschiede zwischen den drei Klassen zu maximieren, wobei darauf zu achten ist, dass die Nukleus-Klasse sich immer noch von den Kapillaren (deren Werte für alle Kanäle bei 255 liegen) unterscheiden lässt.

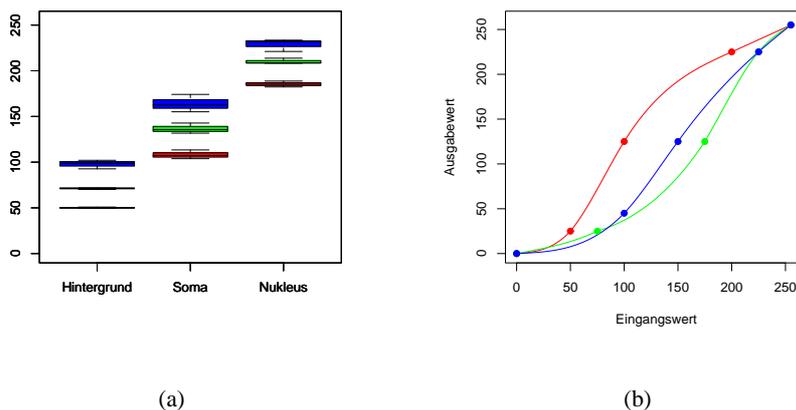


Abbildung 2.10: Darstellung der Mittelwerte einer Profilanalyse (a) und den daraus resultierenden Kurven.

Auf Grundlage dieser Abbildung wird eine Datei mit Transformationswerten erstellt, wie in Listing 3 gezeigt. In der ersten Spalte der Datei steht der Farbkanal, in der zweiten der Ausgangswert und in der dritten Spalte der Wert, in den der Ausgangswert transformiert werden soll. So ist z.B. für den roten Kanal festgelegt, dass Schwarz (Wert 0) und Weiß (Wert 255) nicht transformiert werden sollen. Der Wert für den Hintergrund wird von ca. 50 nach unten auf den Wert 25 korrigiert, das Soma wird von 100 leicht auf 125 angehoben, ebenso wie der Nukleus (von 200 auf 225). Als nächster Schritt wird dann für jeden Kanal eine Kurve berechnet, auf der all diese Punkte liegen.

Listing 3 Beispiel für eine RGB-Datei. Diese Datei bildet die Grundlage für die Berechnung der Kurve in Abbildung 2.10(b). Nähere Erläuterung im Text.

R	0	0	B	0	0
R	50	25	B	100	45
R	100	125	B	150	125
R	200	225	B	225	225
R	255	255	B	255	255
G	0	0			
G	75	25			
G	175	125			
G	225	225			
G	255	255			

Die Berechnung einer Kurve durch die so vorgegebenen Punkte ist prinzipiell durch die Anpassung einer Polynomfunktion möglich. Da jedoch bei mehr Punkten eine Po-

lynomfunktion höherer Ordnung verwendet werden muss, kommt es leicht zu artifiziellen Minima und Maxima, was einen kontinuierlichen, glatten Übergang verhindert.

Deshalb habe ich mich entschieden, für die Anpassung nicht eine einzige Polynomfunktion hoher Ordnung, sondern stückweise definierte Polynomfunktionen niedriger Ordnung zu verwenden. Diese Funktionen nennt man Splines. Eine Spline erster Ordnung entspricht einer linearen Verbindung der einzelnen Punkte. Bei einer Spline zweiter Ordnung werden die Segmente durch Parabeln angenähert. Die am häufigsten verwendeten Splines sind die so genannten *kubischen Splines* dritter Ordnung, die auch hier verwendet wurden.

Bei der Implementierung der Splines habe ich mich an [75] orientiert und wie dort beschrieben, zunächst die zweite Ableitung in jedem Punkt berechnet, um daraus die Koeffizienten für die Splines abzuleiten.

Das Resultat der Berechnung ist in Abbildung 2.10(b) auf der vorherigen Seite zu sehen. In dieser Abbildung sind sowohl die Stützpunkte der Splines (ausgefüllte Kreise), als auch die Splines selbst zu sehen. Indem die Spline-Funktionen für alle Kanäle mit den Werten von 0 bis 255 berechnet werden, kann eine Lookup-Table erstellt werden.

In Abbildung 2.11(a) ist ein Ausschnitt einer Aufnahme mit einer solchen Lookup-Table zu sehen. Die Aufnahme ist dunkel und mit bloßem Auge sind nur schwer die Konturen der Zellen zu erkennen.

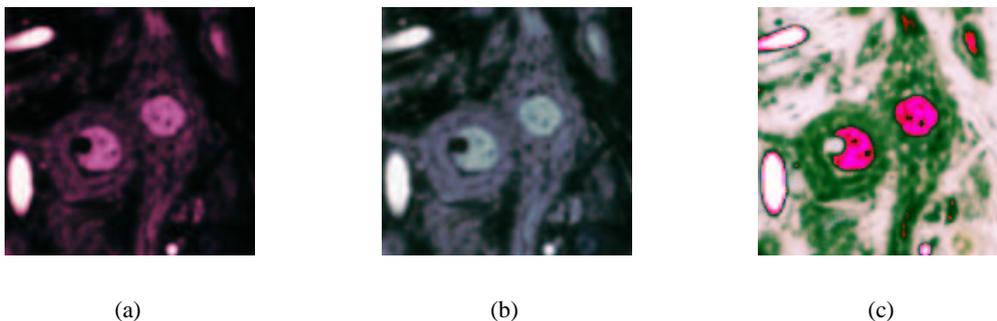


Abbildung 2.11: Drei unterschiedliche Darstellungsformen zweier Zellen: Das Originalbild ist in (a) dargestellt, das Bild mit einem reversen Lookup in (b) und in (c) ist das Bild mit einer Falschfarben-Abbildung gezeigt.

Die Bilder müssen zunächst nachbearbeitet werden, damit die Kontraste für den Betrachter ebenso deutlich werden, wie für den Computer. Dazu gibt es zwei Möglichkeiten:

In Abbildung 2.11(b) ist das Ergebnis eines reversen Lookups zu sehen. Dazu werden einfach die Wertepaare (x/y) dem Programm zur Berechnung der Splines als (y/x) -Paare präsentiert, wodurch die Umkehrfunktion der ursprünglichen Lookup-Table, die reverse Lookup-Table, berechnet wird. Wenn man diese Tabelle verwendet, ist es leicht möglich, mit einem weiteren Programm die Transformation durch die erste Lookup-Table rückgängig zu machen und das Bild wieder in den gewohnten Farbraum zu transformieren. Diese Transformation kann bei allen Lookup-Tables und teilweise auch bei

einer Transformationsmatrix durchgeführt werden. Sie ist sehr hilfreich, um dem Betrachter die gewohnte Ansicht seines Präparates zur Verfügung zu stellen.

In der Abbildung 2.11(c) ist eine manuell erstellte Transformation mit Falschfarben zu sehen. Die Anpassung wurde durch eine manuell angepasste, parabelförmige Falschfarben-Funktion vorgenommen und zeigt sehr deutlich, wie gut sich nach diesen Vorbereitungen am Mikroskop die Farben separieren lassen.

2.2.3 Einfluss der Mikroskopeinstellung auf die Mustererkennung

Jede der vorgestellten Methoden bietet unterschiedliche Vor- und Nachteile für eine spätere Mustererkennung. Generell kann man sagen, dass der Grad der Optimierung umso größer sein sollte, je reproduzierbarer die Farbwiedergabe der Bilder ist. Umgekehrt muss der Grad der Optimierung reduziert werden, wenn die Bilder eine hohe Variabilität aufweisen.

Im Optimalfall verwendet man Bilder mit geringer Variation und kann dann eine optimale Lookup-Table berechnen. Je größer die Variation zwischen den einzelnen Bildern ist, desto weniger sollte man die Aufnahmebedingungen optimieren.

Im vorliegenden Fall wurde eine vergleichsweise einfache Lookup-Table verwendet, indem der rote Kanal (mit dem neu erstellen *rgb*-Kanal belegt) im Bereich von 0 bis 128 linear gespreizt wurde und der grüne Kanal (ebenfalls mit *rgb* belegt) im oberen Bereich des Farbspektrums von 128 bis 256 gespreizt wurde. In Kombination mit der weiter oben berechneten Transformationsmatrix bietet dieses Verfahren eine optimale Auftrennung.

Der Vorteil dieses Verfahrens liegt vor allem darin, dass die zwei Kanäle rot und grün für einen unterschiedlichen Zweck optimiert sind. Dabei wird in Kauf genommen, dass durch eine Verschiebung des Farbraumes einer der beiden Kanäle an eine Grenze stößt, was jedoch vom anderen kompensiert werden kann. Durch die Verteilung der Information auf zwei Kanäle wird so innerhalb des RGB-Raumes indirekt ein 16-Bit-Grauwert-Bild kodiert.

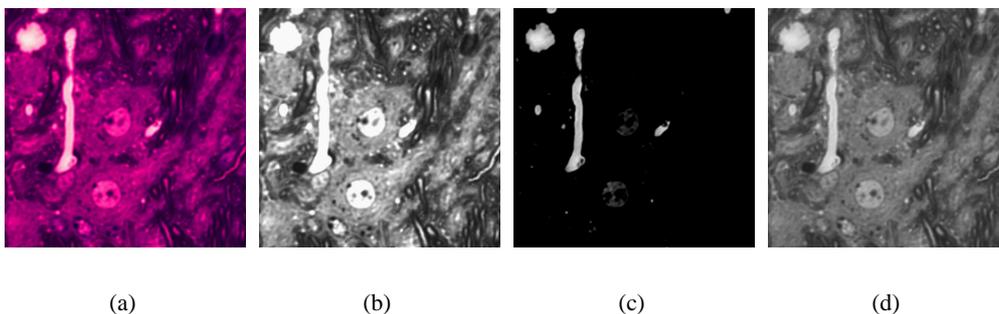
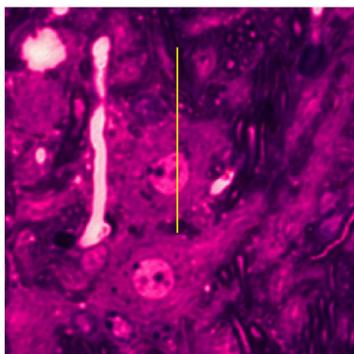


Abbildung 2.12: Effekt der verwendeten Lookup-Table: In Abbildung (a) ist die durch die Lookup-Table modifizierte Abbildung zu sehen. Der rote Kanal (b) trennt die Zellkerne vom Hintergrund, der grüne Kanal (c) trennt Zellkerne und Kapillaren und der blaue Kanal (d) enthält noch die originalen Informationen.

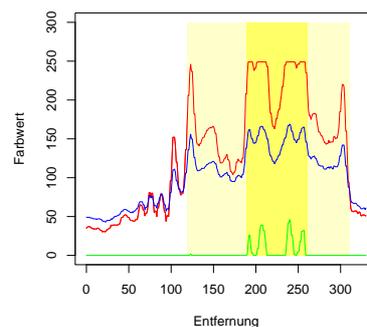
Dies lässt sich am Beispiel der Aufnahme in Abbildung 2.12 auf der vorherigen Seite gut zeigen. Das Ergebnis kann an Hand des Profils in Abbildung 2.13 visualisiert werden. Der in Abbildung 2.12(b) gezeigte rote Kanal spreizt den unteren Bereich des Farbspektrums auf. Dadurch lassen sich einerseits die Somata gut vom Hintergrund unterscheiden und andererseits tritt der Zellkern gegenüber dem Soma in einem reinen Weiß hervor. Mit diesem Bild alleine wäre es jedoch nicht möglich, die Zellkerne von den Kapillaren zu unterscheiden. Hierzu wird der in (c) gezeigte grüne Kanal über die untere Hälfte des Farbspektrums gespreizt.

Der grüne Kanal ist in der Lage, die Zellkerne von den Kapillaren zu unterscheiden. In dem Beispiel sind die Kapillaren sehr hell, während die Zellkerne praktisch gar nicht zu sehen sind. So ist eine einfache Unterscheidung zwischen diesen beiden Klassen gewährleistet.

Der blaue Kanal enthält schließlich die unveränderte Information. Dies ist sowohl für die optische Darstellung als auch für das Zusammenführen der beiden anderen Kanäle nützlich.



(a)



(b)

Abbildung 2.13: Profilbild einer Zelle mit der endgültigen Kombination aus Transformationsmatrix und Lookup-Table: In (b) ist das Profil für die drei Farbkanäle entlang der in (a) gelb gezeichneten Linie zu sehen.

Auch wenn ein Teil der Informationen verloren gegangen ist, so kann dennoch aus dem erstellten Bild die ursprüngliche Farbinformation zurückgewonnen werden. Kombiniert man die Informationen aus dem roten und dem grünen Kanal so, dass die des roten die untere und die des grünen die obere Hälfte des Farbraumes bilden, erhält man den ursprünglich erstellten *rgb*-Kanal. Multipliziert man diesen nun mit $1 - r''$ bzw. $1 - g''$ erhält man einen roten und einen grünen Kanal, die sich mit dem blauen Kanal zu einem Bild kombinieren lassen, das der ursprünglichen Färbung sehr nahe kommt. Dieses Verfahren ist hilfreich für eine spätere visuelle Beurteilung.

Als Fazit lässt sich festhalten, dass eine Optimierung der Einstellungen des Mikroskops redundante Informationen eliminieren und die verbleibenden Informationen besser ausnutzen kann, als es bei einer normalen Aufnahme der Fall wäre.

2.3 Verwendete Hard- und Software

Bei einem Projekt, das hauptsächlich auf der Anwendung von Computern basiert, ist die Auswahl von geeigneter Hard- und Software von großer Bedeutung. Bei der Software wurde eine gezielte Auswahl getroffen, die im nächsten Abschnitt kurz beschrieben und begründet wird.

Bei der Hardware hingegen ist die Situation komplizierter. Eine Beschreibung der verwendeten Hardware wäre nur bedingt aussagekräftig, da sich die verwendete Hardware mehr an den in unserer Arbeitsgruppe vorhandenen Ressourcen als an der eigentlich notwendigen Rechenleistung orientiert hat. Deshalb werde ich im übernächsten Abschnitt eine Betrachtung der im Optimalfall zu verwendenden Hardware anschließen.

2.3.1 Software

Bei der Software galt es vor allem drei Entscheidungen zu treffen:

- Welches Betriebssystem?
- Welche Programmiersprache?
- Welches Bildformat?

Die Aufnahme der Bilder erfolgte unter Windows98, da nur unter diesem System die Software der Kamera lauffähig ist. Nach dem Transfer der Bilder erfolgten alle weiteren Berechnungen unter dem Betriebssystem Linux.

Linux bietet für den praktischen Einsatz viele Vorteile. Das System lässt sich dank der offenen Quellen optimal an die persönlichen Bedürfnisse anpassen. Des Weiteren ist es kostenlos einzusetzen – was bei der Verwendung von mehreren Rechnern gegenüber Windows eine beträchtliche Einsparung ergibt. Im Allgemeinen läuft Linux stabiler als Windows und bietet eine gute Unterstützung von Skripten, d.h. der automatischen Ausführung von mehreren miteinander verknüpften Programmen.

Die Verwendung von Linux ist jedoch keine unabdingbare Voraussetzung für die Anwendung des hier vorgestellten Verfahrens. Es wurde strikt darauf geachtet, dass alle Algorithmen komplett in Java geschrieben wurden und somit plattformunabhängig zu verwenden sind.

Die Verwendung von Java als Programmiersprache hat den Vorteil, dass ein einmal kompiliertes Programm auf jeder Plattform, für die es eine JVM (Java Virtual Machine) gibt, lauffähig ist. Die JVM interpretiert den Byte-Code eines Java-Programms.

Hierdurch ist eine hohe Portabilität und ein einfacher Aufbau des Quelltextes gewährleistet. Durch die Benutzung von Java vermeidet man den großen Zeitaufwand, der bei C/C++-Programmen notwendig ist, um das Programm auf unterschiedliche Plattformen zu portieren und es dort zu testen. In der Vergangenheit haben unterschiedliche Benchmark-Tests gezeigt, dass die Performance der JVM unter günstigen Bedingungen der Ausführungsgeschwindigkeit von C/C++ - Programmen sehr nahe kommt.

Ein weiterer sehr wichtiger Punkt ist die gute Unterstützung von Bildbearbeitungsoperationen durch Java. Durch die JAI (Java Advanced Imaging) - API [51, 95] und die Java2D - API stehen viele leistungsfähige Werkzeuge für die Bildbearbeitung zur Verfügung. Dies erleichtert das Implementieren einfacher Operationen und bildet für komplexere Berechnungen eine gute Basis – ohne dass die Installation von zusätzlichen Bibliotheken und Modulen notwendig wird.

Als Bildformat kommen prinzipiell nur solche Formate in Frage, die drei Farbkanäle und einen Alphakanal unterstützen, die Bilder verlustfrei komprimieren und von Java unterstützt werden. Diese Kriterien treffen auf die Formate TIFF und PNG, sowie bedingt auf PPM zu. Die Bilder des Mikroskops wurden als TIFF bereitgestellt. Alle weiteren Bearbeitungsschritte wurden im Format PNG gespeichert. Eine ausführliche Diskussion der einzelnen Formate unter dem Aspekt des Speicherbedarfs folgt im Abschnitt „Speicherkapazität“.

2.3.2 Hardware

Durch die Verwendung von Java ist die Wahl des Betriebssystems und der Computerarchitektur weitgehend frei. Neben Windows, Linux, Solaris und MacOS werden mittlerweile auch die viele Unixderivate wie AIX, HP-UX, IRIX, OS/390, OS/400, SCO, Tru64 UNIX, UnixWare und VxWorks unterstützt. In den letzten Jahren setzt sich jedoch auch bei High-Performance-Anwendungen Linux immer mehr durch, da die Nachteile von Linux im Bereich der Performance durch die geringen Kosten und die damit mögliche Anschaffung weiterer, zu einem Cluster verbundener Computer leicht kompensiert werden können.

Für die Analyse einer Bildserie sind drei Komponenten der verwendeten Rechner ausschlaggebend:

- Rechenleistung (CPU - Central Processing Unit)
- Größe des Arbeitsspeicher (RAM - Random Access Memory)
- Kapazität Festplatte (HD - Hard Disk)

Dabei ist die Größe der Festplatte ein „hartes“ Limit, d.h. eine gewisse Größe ist unabdingbar für die Durchführung der Analyse.

Der Arbeitsspeicher hat ebenfalls eigentlich eine „harte“ Grenze, da die zu verarbeitenden Bilder darin Platz finden müssen. Diese Grenze kann jedoch etwas verschoben werden, indem virtueller Speicher auf der Festplatte verwendet wird – dies geht jedoch stark zu Lasten der benötigten Rechenzeit, da Zugriffe auf die Festplatte wesentlich langsamer sind, als ein Zugriff auf den Arbeitsspeicher.

Bei der Rechenleistung schließlich gibt es kein absolutes Ausschlusskriterium. Grundsätzlich ist die Analyse mit jeder CPU möglich, allerdings nimmt die benötigte Laufzeit antiproportional der CPU-Leistung ab.

Bei modernen Grafikkarten ist es möglich, zur Performancesteigerung Teile von intensiven 3D-Rechnungen direkt von der Grafikkarte durchführen zu lassen. Da eine solche Beschleunigung bei den verwendeten Methoden nicht sinnvoll anzuwenden war, wird die Grafikkarte im folgenden nicht weiter betrachtet.

Speicherkapazität

Die Speicherkapazität der Festplatte ist eine wichtige Größe, die leicht zu bestimmen ist. Zunächst lässt sich die Größe eines Bildes relativ einfach berechnen, da jeder Pixel pro Kanal ein Byte verbraucht. Das bedeutet, dass ein typisches Bild mit 1600x1200 Pixeln einen Speicherplatz von

$$1600 \cdot 1200 \cdot 4 = 7.680.000 \quad (2.40)$$

Byte belegt, da neben den drei RGB-Kanälen zusätzlich ein Alpha-Kanal für die Transparenz benötigt wird. Die meisten Formate speichern die Bilder jedoch komprimiert ab, so dass die Größe beträchtlich reduziert wird. In Tabelle 2.1 werden unterschiedliche Bildformate und Kompressionsalgorithmen verglichen.

Format	tif	ppm	ppm	png	png	png
Kompression	keine	keine	bzip2	c1	c6	c9
Größe [Byte]	262.731	196.623	75.064	92.586	82.821	81.846
Prozentual	100	100	38	35	32	32

Tabelle 2.1: Speicherbedarf unterschiedlicher Bildformate für ein Bild der Größe 256x256

Für diesen Vergleich wurden zehn Bilder der Größe 256x256 in unterschiedlichen Formaten abgespeichert und der Speicherbedarf bestimmt. Der Speicherplatz für das unkomprimierte Bild beträgt 262.144 mit bzw. 196.608 ohne den Alpha-Kanal.

Wie in Abbildung 2.14 auf der gegenüberliegenden Seite durch die blauen Linien gezeigt, entsprechen die beiden kompressionslosen Verfahren tif und ppm (ohne Alpha-Kanal) genau dieser Vorgabe. Der geringfügig höhere Speicherbedarf entsteht durch zusätzliche Informationen wie z.B. die Höhe und die Breite des Bildes, die in der Datei gespeichert werden.

Die rechten vier Verfahren beinhalten alle einen Kompressionsalgorithmus, der die Dateigröße reduziert. Zunächst wurde die ppm-Datei mit dem weit verbreiteten Bzip2-Algorithmus komprimiert, der bereits ein gutes Ergebnis erzielt. Die drei rechten Balken stellen die Kompression mittels des im png-Standard implementierten zlib-Algorithmus dar. Diese kann in Stufen von eins bis neun geregelt werden, wobei eins einer sehr geringen und neun der maximalen Kompression entspricht.

Da das ppm-Format den Alpha-Kanal nicht unterstützt und die bei tif eingesetzte Kompression auf Grund von Patentstreitigkeiten in vielen Softwarelösungen nicht implementiert ist, habe ich mich für die Verwendung der png-Kompression auf Stufe 6 entschieden. Eine stärkere Kompression bringt kaum einen Vorteil und führt zu einer erheblichen Einbuße bei der Geschwindigkeit des Einlesens.

Unter Einbeziehung der Kompression kann der Speicherplatz eines Bildes demnach auf ca. ein Drittel reduziert werden:

$$\frac{1600 \cdot 1200 \cdot 4}{3} = 2.560.000 \quad (2.41)$$

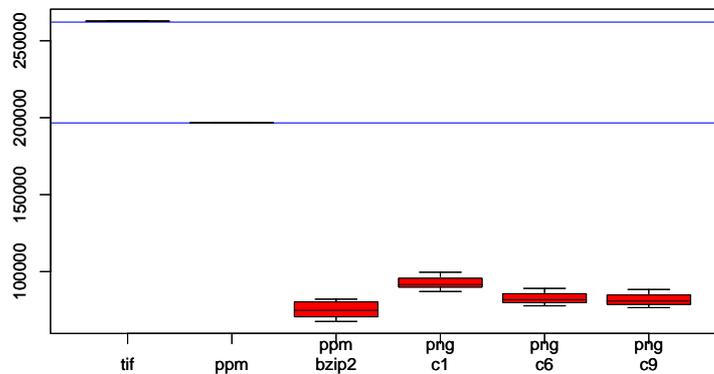


Abbildung 2.14: Speicherbedarf unterschiedlicher Bildformate

Auf dieser Basis kann nun der Gesamtspeicherbedarf eines Projektes berechnet werden: Angenommen, das Projekt umfasst 1.000 Einzelebenen, in denen jeweils drei Bilder aufgenommen werden, so ergibt dies einen Speicherbedarf von 7,680 GB (GigaByte) für die komprimierten png-Bilder. Es gilt jedoch zu beachten, dass für die vom Mikroskop aufgenommenen Originalbilder nochmals 23,040 GB als Zwischenspeicher zu reservieren sind. Diese Bilder können nach der Konvertierung in das png-Format gelöscht werden.

Normalerweise wird man die weiteren Zwischenergebnisse nicht sofort löschen wollen, sondern sie für eine spätere Nachkontrolle archivieren. Dies bedeutet, dass für jeden Zwischenschritt, in dem mit den farbigen Bildern gearbeitet wird ca. 8 GB reserviert werden müssen. Für jeden Schritt, in dem mit grauwert-reduzierten Bildern gearbeitet wird, sind immer noch 2 GB bzw. 4 GB (wenn die Bilder einen Alphakanal enthalten) notwendig.

Momentan beinhaltet das Verfahren fünf farbige und drei graue Zwischenergebnisse, so dass eine vollständige Rekonstruktion ca. 46 GB benötigen würde. Dabei ist zu beachten, dass in Abhängigkeit der Daten und der zu speichernden Zwischenschritte das Volumen erheblich schwanken kann, so dass eine Festplatte mit einem Volumen von 80 GB für eine Rekonstruktion angemessen wäre. Eine Festplatte dieser Größe ist heutzutage vergleichsweise billig und somit stellt der Festplattenspeicher das geringste Problem dar.

Arbeitsspeicher (RAM)

Der Arbeitsspeicher ist bei bildverarbeitenden Algorithmen immer eine kritische Größe, da er in der Lage sein sollte, auch mehrere Kopien eines Bildes aufzunehmen. Der benötigte Arbeitsspeicher lässt sich nicht so einfach wie der benötigte Festplattenplatz abschätzen, da diese Größe stark von den Interna der verwendeten Programmiersprache und der Anzahl von Kopien eines Bildes, die simultan im Speicher gehalten werden

müssen, abhängt.

Bei nicht mehr als drei Bildern bei einem Mosaicing sind alle hier vorgestellten Verfahren in der Lage, mit 512 MB Arbeitsspeicher auszukommen. Einige Verfahren könnten jedoch bei einem größeren Hauptspeicher nochmals deutlich beschleunigt werden, so dass 1-2 GB Hauptspeicher für eine Analyse sinnvoll wären.

Bei der im folgenden Kapitel beschriebenen Abschätzung der Dauer der Berechnung gilt es, die Zugriffszeit des RAM ebenfalls zu berücksichtigen. Bei einem Computer mit einer schnelleren CPU sollte in jedem Fall ein RAM mit möglichst schneller Zugriffszeit verwendet werden.

Rechenleistung

In dieser Arbeit wurden alle Berechnungen auf einem AMD-Athlon mit 700 MHz durchgeführt. Um die Verkürzung der Rechenzeit bei einem schnelleren Prozessor abzuschätzen, wurden zwei Verfahren angewendet. Zum einen wurden die Werte des Linpack-Benchmarks [5] herangezogen. Dieser Benchmark verwendet Systeme von linearen Gleichungen, sowie auf den kleinsten Fehlerquadraten basierende lineare Optimierungsprobleme, um auf diesem Weg die CPU-Leistung zu bewerten. Daraus resultiert ein Wert in der Einheit Mflop/s, der angibt wieviele Millionen Fliesskommaoperationen in einer Sekunde ausgeführt werden können. Dieser – für eine große Anzahl unterschiedlicher Computer verfügbare Wert – wurde mit einem speziellen Bildverarbeitungsbenchmark verglichen.

Die Werte für den in Java ausgeführten Linpack-Benchmark [4] sind in Abbildung 2.15 dargestellt. Man kann aus der Abbildung gut erkennen, dass zwar ein prinzipieller Zusammenhang zwischen der CPU-Geschwindigkeit und der realen Leistung besteht, letztere allerdings sehr breit gestreut ist und demnach noch von anderen Faktoren abhängig sein muss.

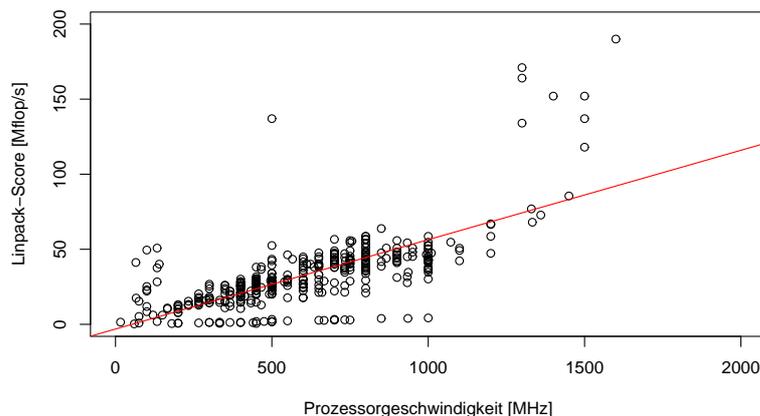


Abbildung 2.15: Korrelation zwischen der Prozessorgeschwindigkeit und dem im Linpack-Benchmark erzielten Wert

Es wurde absichtlich die in Java implementierte Variante des Linpack-Benchmarks verwendet, da dies bereits eine etwas bessere Übertragbarkeit auf die zu lösenden Aufgaben bedeutet. Dadurch haben jedoch auch die Implementation der Virtuellen Maschine, sowie andere Merkmale der Computer Einfluss auf das Ergebnis, was die Bewertung verfälschen kann. Grundsätzlich kann man jedoch sagen, dass eine Verdopplung der CPU-Taktrate eine Verdopplung der Benchmark-Ergebnisse mit sich bringt. Dabei ist zu beachten, dass die modernen Computer mit 1,5 GHz oder einer noch höheren Taktrate deutlich oberhalb dieser Beziehung liegen, daher mit einem noch größeren Zuwachs zu rechnen ist.

Um die Übertragbarkeit der durch Linpack erzielten Ergebnisse auf die Bildbearbeitung zu überprüfen, wurden mit dem in ImageJ [50] enthaltenen Standard-Benchmark unterschiedliche Computer verglichen.

Dieser Benchmark ist in Java geschrieben und testet 72 unterschiedliche Bildverarbeitungsoperationen. Als Ergebnis wird die benötigte Zeit für einen Durchlauf verwendet. In Abbildung 2.16 sind die Ergebnisse des Benchmarks mit dem vierfach vergrößerten Mandrill-Image (ein 256x256-Bild, das häufig als Standard in der Bildverarbeitung dient) dargestellt.

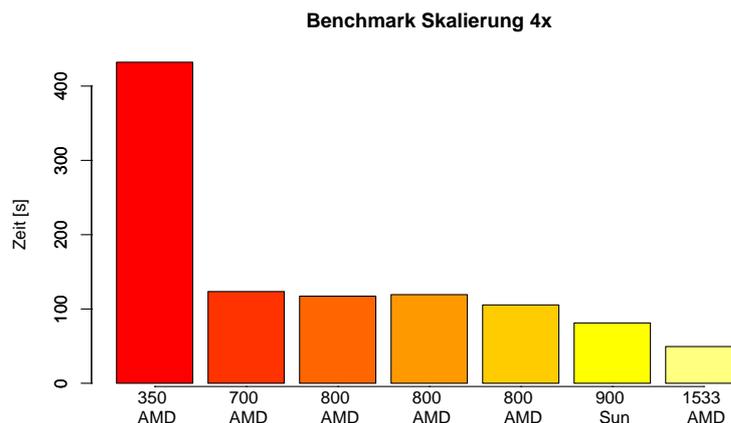


Abbildung 2.16: Benötigte Zeit unterschiedlicher Rechner für die Anwendung des ImageJ-Benchmarks auf das vierfach ergrößerte Mandrill-Image

Diese Ergebnisse bestätigen, dass das ungefähre Verhältnis der Geschwindigkeitszunahme auch für die Bildverarbeitung gilt. Während der in dieser Arbeit verwendete AMD 700MHz für die Berechnung 123,5 s benötigte, war der AMD 1533MHz bereits nach 49,4 s mit der Berechnung fertig.

Demnach kann man bei den bisher getesteten Modellen den Zeitbedarf der Berechnung über einen einfachen Dreisatz gegen die Prozessorgeschwindigkeit ausrechnen. Im Mittel benötigt die Analyse einer Schnittebene ca. 57 min Rechenzeit auf einem AMD 700MHz. Die Analyse von 1.000 Schnitten würde demnach ca. 40 Tage dauern. Durch einen oder mehrere schnellere Computer könnte die Dauer der Analyse beträchtlich gesenkt werden: So würde ein doppelt so schneller Rechner nur 20 Ta-

ge und ein Cluster aus 10 dieser Rechner nur zwei Tage für eine komplette Analyse benötigen.

Zusammenfassung

Die komplette Analyse eines Nukleus war im Rahmen der in dieser Arbeit zur Verfügung stehenden Rechenkapazitäten nicht durchführbar. Mit einer verbesserten Architektur, z.B. einem Cluster von 10 modernen Computern mit jeweils einem schnellen Prozessor und 2 GB RAM, sowie einem zentralen RAID (Redundant Array of Inexpensive Disks) von ca. 1 TB Speicherkapazität wäre das Verfahren jedoch routinemäßig einsetzbar.

2.4 Fazit

In diesem Kapitel konnte gezeigt werden, dass eine Optimierung der Einstellungen an Digitalkamera und Mikroskop einen wesentlichen Beitrag zur Verbesserung der Erkennungsrate leisten kann. Durch die richtige Einstellung von Transformationsmatrix und Lookup-Table lassen sich die gewünschten Strukturen deutlich zuverlässiger extrahieren, als es ohne diese Vorverarbeitung möglich wäre.

Wenn der Hersteller der Kamera einen direkten Zugang zu deren Interface erlauben würde, wäre es möglich, den größten Teil der Mustererkennung bereits auf dieser Ebene zu bewältigen. Solange solch ein Zugang nicht gegeben ist, muss man auf etwas weniger optimierte Einstellungen ausweichen.

Im zweiten Teil dieses Kapitels habe ich dargestellt, dass die verwendete Software auf offenen Standards basiert und somit einfach auf andere Plattformen und damit auf andere Problemfelder zu übertragen ist. Durch eine möglichst genaue Abschätzung der benötigten Computer-Kapazitäten können die Bedingungen für einen routinemäßigen Einsatz dieses Verfahrens prognostiziert werden.

Image Mosaicing

Wie bereits in der Einleitung ausgeführt, ist die Erweiterung der lateralen Ebene eine wichtige Voraussetzung für das Erkennen der Zellen. Auch hierbei spielt wiederum das Problem der diskreten Intervalle eine Rolle. Ebenso wie der Farbraum aus 256 diskreten Abstufungen besteht, ist ein Bild in ein Raster aus diskreten Pixeln eingeteilt. Mit der heutigen Technik ist die Auflösungsbegrenzung durch die Kamera bei einem optisch gut korrigiertem Mikroskop zumeist deutlich schlechter als die durch die Optik verursachte Begrenzung.

Bei der mikroskopischen Digitalaufnahme von Präparaten ist es deshalb notwendig, einen Kompromiss zwischen Auflösung und Größe des Bildausschnittes zu finden. Ist der Ausschnitt groß genug, um die zu untersuchende Struktur aufzunehmen, ist die Auflösung der einzelnen Objekte oft ungenügend. Umgekehrt reicht der Abbildungsbereich der Digitalkamera bei einer hohen Auflösung des einzelnen Objektes meist nicht aus, um die gesamte Struktur abzubilden. Mehrere Einzelbilder aufzunehmen und manuell mit einer Bildbearbeitungssoftware wie z.B. GIMP oder Photoshop zusammenzufügen ist zeitaufwendig, fehleranfällig und für große Serien nicht praktikabel. So würde die Bearbeitung einer Serie von 1.000 Schnitten mit jeweils drei Bildern auch bei einiger Erfahrung mindestens 40 Arbeitstage benötigen.

Das Problem, mehrere Bilder zu kombinieren, ist aus der Panoramafotografie bekannt. Die unterschiedlichen Techniken des sogenannten Image Mosaicings, d.h. dem Zusammenfügen mehrerer Bilder (siehe Abbildung 3.1 auf der nächsten Seite), sind teilweise auch in der Histologie einsetzbar. Das hier vorgestellte Programm `imosic` ist nicht nur in der Lage, ein Image Mosaicing mit zwei unterschiedlichen Verfahren durchzuführen; es implementiert vor allem auch umfangreiche Kontrollmechanismen, um falsch kombinierte Bilder auszusortieren und ist so in der Lage, auch große Serien automatisch und fehlerfrei zu bearbeiten und so das Problem der lateralen Auflösungsbegrenzung zu umgehen.

Dieses Kapitel beginnt mit einer Einführung in die Grundlagen der beiden verwendeten Methoden, geht dann auf Verbesserungen und Modifikationen, die daran im Rahmen dieser Arbeit vorgenommen wurden, näher ein und präsentiert zum Schluss die Implementierung und die Ergebnisse einer Validierung des Verfahrens im Allgemeinen und der Kontrollmechanismen im Besonderen.

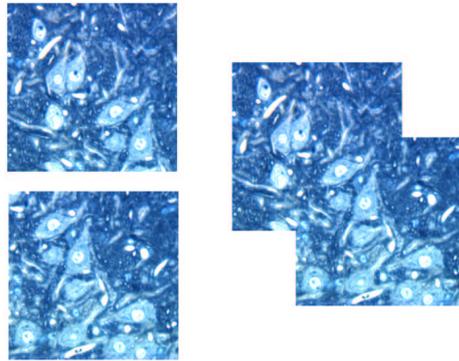


Abbildung 3.1: Schematische Darstellung eines Mosaicings: Die beiden Einzelbilder links werden rechts zu einem Bild kombiniert.

3.1 Die Image Mosaicing-Verfahren

Im Gegensatz zum klassischen Anwendungsbereich des Image Mosaicings [35] muss für mikroskopische Aufnahmen nur die Translation korrigiert werden. Eine Korrektur von Drehung oder Vergrößerung sind bei der Verwendung eines XY-Tisches nicht notwendig, könnten aber z.B. unter Verwendung des von Reddy [81] beschriebenen Verfahrens leicht ergänzt werden.

Als Absicherung gegen mögliche Fehler habe ich zwei unabhängige Verfahren entwickelt: Zunächst werden die Transformationsparameter (die Verschiebung in x- und y-Richtung) mit dem Verfahren der Phasenkorrelation [62], basierend auf einer Fourier-Transformation der Bilder, berechnet. Aus dem Shift im Frequenzbereich können die Transformationsparameter bestimmt werden.

Parallel dazu werden die Transformationsparameter durch eine Kreuzkorrelation der Bilder ermittelt. Dabei wird die Differenz der Farbwerte im Überlappungsbereich minimiert. Die korrekte Translation entspricht der Stelle der geringsten Abweichung.

Um den Rechenaufwand gering zu halten, wird das Bild zuvor mittels einer bikubischen Interpolation [98] um einen definierten Faktor (Werte zwischen 3 und 5 haben sich in der Praxis als sinnvoll erwiesen) verkleinert. Nachdem mit den verkleinerten Bildern eine korrekte Ausrichtung gefunden wurde, wird die resultierende Transformation mit dem Skalierungsfaktor multipliziert. Im Umfeld von 10 Pixeln um diese Koordinaten wird in den unskalierten Bildern das Image Mosaicing erneut durchgeführt.

Im folgenden möchte ich die beiden einzelnen Verfahren näher beschreiben.

3.1.1 Image Mosaicing mittels Phasenkorrelation

Das erste der beiden vorgestellten Verfahren verwendet die Fourier Transformation der beiden Bilder, um das Image Mosaicing durchzuführen. Diese Methode kann sehr effizient umgesetzt werden, hat eine geringe Laufzeit und sollte deshalb normalerweise die Methode der Wahl sein. Jedoch bringt die Beschränkung auf nur eine Methode

einige Probleme mit sich, auf die in Kapitel 3.3 auf Seite 47 noch näher eingegangen wird.

Die Daten, die als Ausgangsmaterial für eine Fourier-Transformation dienen, bestehen aus einem realen und einem komplexen Anteil. Bei einem Bild als Ausgangsmaterial entspricht der reale Anteil dem Farbwert des Pixels; der komplexe Anteil ist Null. Man kann eine zweidimensionale Fourier-Transformation als die zweifache Anwendung einer eindimensionalen Fourier-Transformation betrachten, indem zuerst die Zeilen Fourier-transformiert werden, diese dann durch die Werte der Fourier-Transformation ersetzt werden und dann diese Daten wiederum – dieses Mal allerdings spaltenweise – Fourier-transformiert werden. Dies kann direkt aus der Definition der DFT (Diskreten Fourier Transformation) in einem diskreten System hergeleitet werden [14, 77].

Es sei an dieser Stelle angemerkt, dass auf Grund von Implementationsdetails der FFT (Fast Fourier Transformation) [65, 77] die Seitenlänge der Bilder immer auf die nächsthöhere durch 2^x darstellbare Integer-Zahl (ganze Zahl) erhöht werden muss. Ein Bild mit der Fläche 257×257 Pixel benötigt für die Berechnung also genau so lange wie ein Bild mit einer Fläche von 512×512 Pixeln, da ersteres in einem Array mit der gleichen Größe wie das erste Bild platziert werden muss.

Das von Kuglin [62] entwickelte Verfahren der Phasenkorrelation beruht auf der Fourier-Transformation zweier Bilder. Dieses Verfahren stellt – vorausgesetzt, es handelt sich nur um eine Translation und nicht um eine Rotation oder Skalierung – ein sehr präzises und schnelles Verfahren für das Image Mosaicing dar.

Wie von Gottesfeld-Brown [35] dargelegt, beruht die Phasenkorrelation auf dem sogenannten *Shift-Theorem*, das die Translationseigenschaften der Fouriertransformation ausnutzt. Es besteht zwischen zwei Bildern I_1 und I_2 , die sich nur durch eine Verschiebung von d_x und d_y voneinander unterscheiden folgende Beziehung:

$$I_2(x, y) = I_1(x - d_x, y - d_y) \quad (3.1)$$

Diese Abhängigkeit im x-/y-System bedingt nach dem Shift-Theorem eine Abhängigkeit in der Fourier-Domäne, die wie folgt formuliert werden kann:

$$F_2(\omega_x, \omega_y) = e^{-j(\omega_x d_x + \omega_y d_y)} F_1(\omega_x, \omega_y) \quad (3.2)$$

Bei dieser und den folgenden Formeln ist zu beachten, dass das in den Naturwissenschaften üblichere j , wobei $j = \sqrt{-1}$ gilt, dem in der mathematischen Literatur üblichen i für den selben Ausdruck vorgezogen wurde. ω steht für die Raumfrequenz der entsprechenden Koordinate.

Die Funktionen F_1 und F_2 sind dabei die Fourier-Transformationen der beiden Bilder. Die Bilder haben demnach in der Fourier-Domäne die gleiche Amplitude, unterscheiden sich aber in der Phase um einen Wert, der direkt proportional zu der Verschiebung der beiden Bilder ist. Verwendet man für die Fourier-Transformation die exponentielle Schreibweise, dann ist $F_i(\vec{\omega}) = |F_i| e^{j\phi_i(\vec{\omega})}$ für $i = 1, 2$ und die Phasendifferenz der Phasenwerte ϕ_1 und ϕ_2 der beiden Bilder ist gleich $e^{j(\phi_1 - \phi_2)}$. Das Shift-Theorems besagt, dass die Phasendifferenz dem Crosspower-Spektrum der Funktionen entspricht:

$$\frac{F_1(\omega_x, \omega_y)F_2^*(\omega_x, \omega_y)}{|F_1(\omega_x, \omega_y)F_2^*(\omega_x, \omega_y)|} = e^{-j(w_x d_x + w_y d_y)} \quad (3.3)$$

Hierbei kennzeichnet * das komplexe Konjugat. Bildet man wiederum die inverse Fourier-Transformation, erhält man eine Deltafunktion, die auch als Phasenkorrelations-Oberfläche bezeichnet wird. Eine solche Oberfläche ist beispielhaft für zwei Bilder in Abbildung 3.2 gezeigt. Das Maximum dieser Fläche entspricht der Verschiebung. Die Größe des Maximums dient als Qualitätsmarker für die Exaktheit des Mosaicing.

Ein Problem dieser Methode ist, dass es mittels der Phasenkorrelation nicht möglich ist, zwischen negativen Verschiebungen und Verschiebungen größer als $N/2$ zu unterscheiden, wobei N die Größe des Bildes (in Pixeln) in der entsprechenden Dimension ist.

Dieses Problem kann gut am Beispiel des eindimensionalen Analogons gezeigt werden [71]. Dabei ist das normalisierte Crosspower-Spektrum durch $S(k)$ dargestellt. Der Index für den x-/y-Raum ist dabei n , der Index für den Frequenzraum k , δ steht für die diskrete Fourier-Transformation und $IDFT$ für die inverse diskrete Fourier-Transformation:

$$\begin{aligned} f'(n) &= f(n + n_0) \\ \Rightarrow S_{f,f'}(k) &= e^{j2\pi \frac{k}{N} n_0} \underbrace{IDFT}_{\leftarrow} \delta(n - N + n_0) \end{aligned} \quad (3.4)$$

$$\begin{aligned} f'(n) &= f(n - N + n_0) \\ \Rightarrow S_{f,f'}(k) &= e^{j2\pi \frac{k}{N} (-N + n_0)} = e^{j2\pi \frac{k}{N} n_0} \underbrace{IDFT}_{\leftarrow} \delta(n - N + n_0) \end{aligned} \quad (3.5)$$

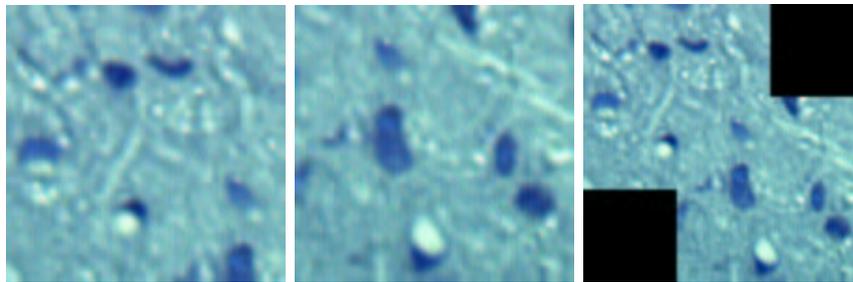
Um dieses Problem zu lösen, existieren zwei Möglichkeiten. Entweder man geht den in [71] vorgeschlagenen Weg und verdoppelt einfach die Größe der FFT, was den Fall, dass der Shift größer als $N/2$ ist ausschließt. Dies führt allerdings zu einem höheren Rechenaufwand und vor allem zu einem immensen Speicherbedarf. Eine Alternative ist der weiter unten im Detail beschriebenen Weg, indem man die vier möglichen Richtungen an Hand eines kleinen Bildausschnitts testet.

Die Technik der Phasenkorrelation ist auch dann geeignet, wenn die Belichtungsbedingungen, wie es in der Mikroskopie manchmal der Fall ist, geringfügig variieren. Diese Variation der Beleuchtung beeinflusst normalerweise nur die unteren Frequenzen und beeinträchtigen das Ergebnis somit nicht. Ebenso ist die Technik gegenüber Änderungen im Farbspektrum relativ insensitive und ist somit gegenüber Veränderungen in der Farbempfindlichkeit des CCD-Sensors sehr tolerant.

Korrektur der Translationsrichtung

Wie weiter oben erläutert, liefert das Verfahren der Phasenkorrelation nur ein Wertepaar, das jedoch gegebenenfalls in seiner Richtung korrigiert werden muss.

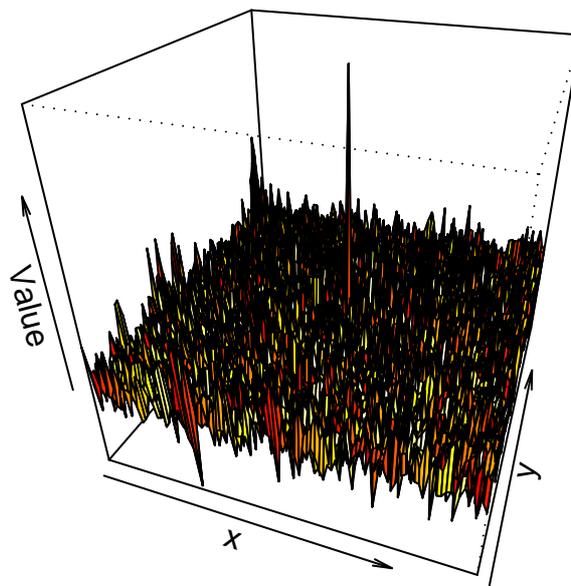
Es handelt sich bei der Verschiebung nicht um ein fehlendes Vorzeichen, sondern um die fehlende Richtung, welche der Korrektur bedarf. In Abbildung 3.3 auf Seite 42 ist dieser Sachverhalt dreidimensional dargestellt.



(a)

(b)

(c)



(d)

Abbildung 3.2: Wie im Text beschrieben wird auf die beiden Bildern in 3.2(a) und 3.2(b) der Phasenkorrelations-Algorithmus angewandt. Das fertige Image Mosaicing in 3.2(c) beruht dabei auf der in 3.2(d) gezeigten Phasenkorrelations-Oberfläche der beiden Bilder. Deutlich ist das Maximum in der Mitte bei $x=50$, $y=50$ zu sehen (die Kantenlängen des Würfels betragen in x - und y -Richtung 100 Pixel).

Die vom Algorithmus ausgegebenen Koordinaten (50/50) sind in der Abbildung durch einen roten Pfeil gekennzeichnet. Für das Mosaicing gibt es vier Möglichkeiten: es könnte jede der vier Ecken des zweiten Bildes an dieser Stelle ansetzen, was durch die roten Rahmen angedeutet wird. Ober- und unterhalb des Bildes in der Mitte befinden sich Kopien des zweiten Bildes, die dessen Ausrichtung deutlich machen sollen und bei genauer Betrachtung erkennen lassen, dass die Verschiebung nach rechts unten die richtige ist.

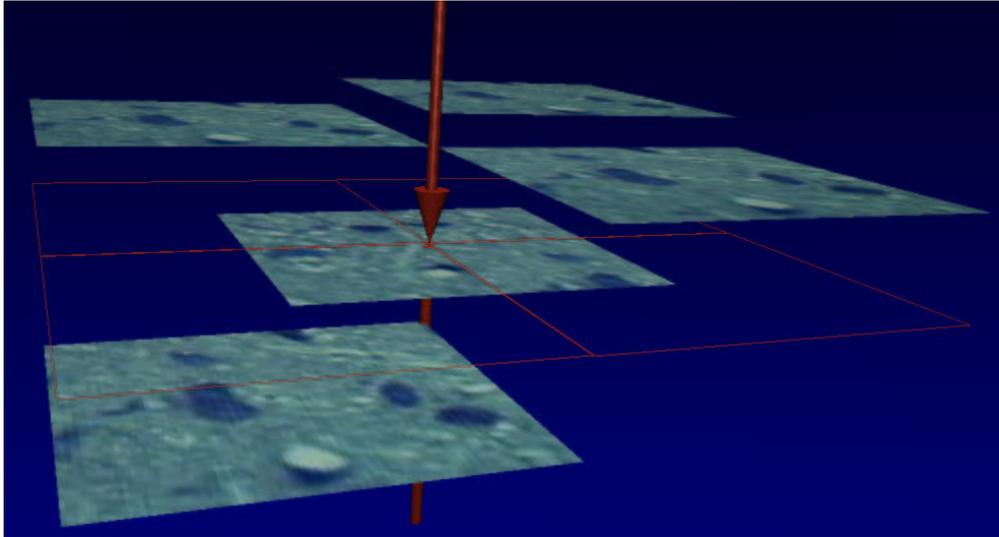


Abbildung 3.3: Die Abbildung zeigt mögliche Positionen eines Mosaicings bei gleichen Koordinaten. Die vom Phasenkorrelations-Algorithmus ausgegebenen Koordinaten sind mit dem roten Pfeil markiert. Die daraus resultierenden vier möglichen Positionen sind durch roten Rahmen gekennzeichnet, die sich darüber bzw. darunter befindlichen Bilder. Die Kombination des mittleren Bildes mit dem Bild ganz rechts würde zu dem korrekten Mosaicing führen.

Um die richtige Translation zu identifizieren werde ich das von Richmond [83] vorgeschlagene Verfahren verwenden. Das Verfahren ist sehr einfach und hat eine ähnliche Basis wie das weiter unten beschriebene Verfahren der Kreuzkorrelation, jedoch ist der Rechenaufwand deutlich geringer.

Um die richtige Richtung festzustellen, verwende ich kleine Bildausschnitte, wie in Abbildung 3.4 auf der gegenüberliegenden Seite zu sehen. Die dort farbig markierten Quadrate haben normalerweise eine Seitenlänge von 20-50 Pixeln und dienen der Bewertung der unterschiedlichen Richtungen.

Zunächst wird jeweils zwischen den gleichfarbigen Quadraten $k = 1..4$ der mit der Seitenlänge a und den Intensitäten $I_{k1}(x, y)$ und $I_{k2}(x, y)$ eine Fehlersumme gebildet:

$$F_k = \frac{\sum_{x=0}^a \sum_{y=0}^a |I_{k1}(x, y) - I_{k2}(x, y)|}{a^2} \quad (3.6)$$

Derjenige Wert F_k , der den geringsten Wert und damit den kleinsten Fehler aufweist, wird der richtigen Richtung entsprechen. Die Werte, um die das zweite Bild dann gegenüber dem ersten verschoben wird, berechnen sich wie folgt:

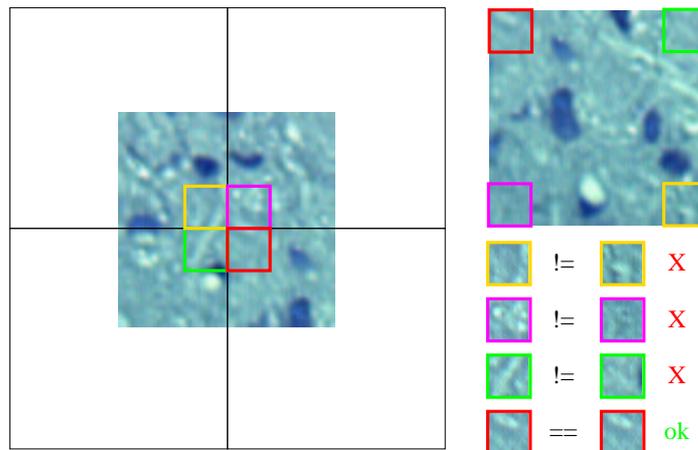


Abbildung 3.4: Schematische Darstellung der getesteten Flächen, um die richtige Positionierung nach der Phasenkorrelation zu finden. Es werden jeweils die farbige gleich markierten Bildausschnitte im Bild links und rechts oben verglichen. In diesem Beispiel ist die Abweichung der beiden roten Rahmen bei einem pixelweisen Vergleich am geringsten. Das Mosaicing erfolgt also wie in Abb. 3.2(c) gezeigt.

$$shift_x = \begin{cases} shift'_x & \text{wenn Verschiebung nach rechts} \\ shift'_x - width & \text{wenn Verschiebung nach links} \end{cases} \quad (3.7)$$

$$shift_y = \begin{cases} shift'_y & \text{wenn Verschiebung nach unten} \\ shift'_y - height & \text{wenn Verschiebung nach oben} \end{cases} \quad (3.8)$$

Dabei ist *width* die Breite und *height* die Höhe des Bildes. Verschiebt man das zweite Bild um die entsprechende Anzahl von Pixeln gegenüber dem ersten, sollte ein korrektes Image Mosaic entstehen.

3.1.2 Image Mosaicing mittels Kreuzkorrelation

Wenn man auf einer Transformation in den Phasenraum verzichtet und mit einer normalen Kreuzkorrelation arbeitet, hat dies einige Vorteile wie z.B. die größere Robustheit bei einer leichten Rotation und der doppelten Kontrolle, aber auch einen wichtigen Nachteil: Das Verfahren ist sehr rechenintensiv.

Die Kreuzkorrelation in ihrer ursprünglichen Form ist deshalb in meinen Augen für die Praxis nicht anwendbar. Ich werde im folgenden zunächst auf das klassische Verfahren eingehen und dann beschreiben, wie der Algorithmus von mir verbessert und erweitert wurde, so dass er bei Laufzeit und Fehlerquote dem Verfahren der Phasenkorrelation kaum nachsteht.

Das Prinzip der Kreuzkorrelation ist sehr einfach. Wie in Abbildung 3.5 auf der nächsten Seite dargestellt, wird das eine Bild (in blau dargestellt) zeilenweise über das andere (in gelb dargestellte) Bild „geschoben“, wobei im Überschneidungsbereich die Differenz der Farbwerte der beiden Bilder gebildet wird. Die mittlere Abweichung

(Fehlersumme) wird als Bewertung verwendet. Die Verschiebung, die zu der geringsten Abweichung führt, ist das Ergebnis des Algorithmus.

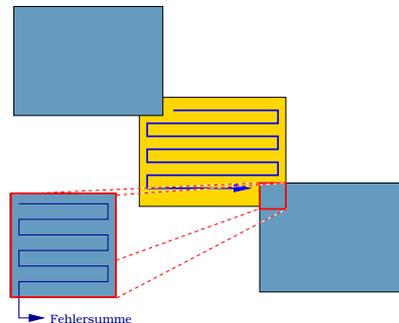


Abbildung 3.5: Schematische Darstellung einer Kreuzkorrelation

Im Listing 4 auf der gegenüberliegenden Seite ist die „klassische“ Kreuzkorrelation gezeigt, bei der über das gesamte Bild iteriert wird. Zunächst wird eine sehr hohe Zahl als Minimum definiert, so dass auf jeden Fall der erste Wert bereits unter diesem Initialisierungswert liegt. Die Variablen `min_x` und `min_y` werden auf einen prinzipiell beliebigen Wert gesetzt.

In der äußersten doppelten Schleife wird das eine Bild pixelweise über das andere geschoben; mit der inneren Schleife wird die Fehlersumme gebildet und mit der bisher gespeicherten verglichen. Ist sie niedriger als die zuletzt gespeicherte, wird der Wert ersetzt, ansonsten bleibt der alte Wert für das Verschieben der Bilder erhalten.

Dieser Lösungsansatz hat einen entscheidenden Vorteil, nämlich dass das Verfahren pixelgenau ist und ein vorhandenes Minimum auf jeden Fall findet. Ein entscheidender Nachteil des Verfahrens ist die extrem schlechte Laufzeit von $O(n^4)$.

Die Laufzeit eines Programms wird in der Informatik üblicherweise in der so genannten O-Notation $O(x)$ angegeben. Da die Zeit, die die Ausführung eines Programms benötigt, von Computer zu Computer unterschiedlich ist, definiert die O-Notation die Schranke für das Wachstum der Laufzeit. Bei einer Laufzeit $O(n)$ verdoppelt sich die Laufzeit mit einer Verdopplung der Eingabeparameter, bei einer Laufzeit von $O(n^2)$ würde sie maximal quadratisch anwachsen. In diesem Fall steigt die Laufzeit demnach in Abhängigkeit von der Größe der Bilder mit der vierten Potenz an.

Ein von Montgomery [72] beschriebener Ansatz, den Algorithmus für die parallele Ausführung auf einer SPARC-Workstation mit 128 Prozessoren zu optimieren, kann die Laufzeit immerhin auf $O(n^2)$ verringern. Er hilft ohne die dazugehörigen 128 Prozessoren allerdings nur bedingt weiter und ist auch nur dann von der Laufzeit $O(n^2)$, wenn die Höhe des Bildes in Pixeln geringer oder gleich der Anzahl der verfügbaren Prozessoren ist.

Ich habe den einfachen Algorithmus an einigen Stellen geringfügig und in einem Punkt entscheidend verbessert, so dass in der Praxis mit meinen Optimierungen eine Laufzeit von ca. 400 Sekunden einer Laufzeit von 24,5 h mit dem Originalalgorithmus gegenübersteht. Diese Verbesserungen möchte ich im folgenden kurz erläutern:

Bilder ausreicht. Somit wäre es erstrebenswert, nicht nur in der inneren Schleife, sondern auch in der äußeren Schleife die Anzahl der Pixel und damit die Anzahl der Durchläufe zu reduzieren. Hierbei stößt man auf das Problem, dass es nicht, wie bei der inneren Schleife, möglich ist, z.B. einfach nur jeden zehnten Pixel zu verwenden, da es sonst sein könnte, dass z.B. im ersten Bild die Pixel 1, 10, 20, . . . verwendet werden, im anderen Bild aber – mangels eines Bezugspunktes, der ja erst gefunden werden soll – die Pixel 5, 15, 25, . . . benutzt werden und man somit niemals eine Übereinstimmung finden kann.

Man muss demnach herausfinden wie eine große Anzahl der Pixel weggelassen werden kann, deren Information allerdings trotzdem in den verbleibenden Pixeln gespeichert wird. Für diesen Zweck kann man entweder eine Gaußpyramide oder wie in meinem Fall ein mit einem bikubischen Interpolator skaliertes Bild verwenden.

Der bikubische Interpolator [98] dient dem Skalieren, d.h. dem Vergrößern und Verkleinern von Bildern. Er reduziert die normalerweise bei einer Skalierung auftretenden Artefakte, die auch als Resampling-Artefakte bezeichnet werden und dafür sorgen, dass ein skaliertes Bild sehr „pixelig“ aussieht, in etwa so, als wäre es aus großen Vierecken zusammengesetzt. Dies liegt daran, dass der normale Algorithmus zum Skalieren von Bildern dem Pixel im skalierten Bild einfach nur den Wert desjenigen Pixels, der diesem am nächsten liegt, zuweist. Der bikubische Interpolator hingegen berücksichtigt auch die umliegenden Pixel.

Er verwendet die 16 nächsten Nachbar-Bildpunkte und benutzt an Stelle der normalen linearen Funktion (auch als bilineare Interpolation bezeichnet) eine bikubische Funktion, um den Wert des Pixels im skalierten Bild zu berechnen. So kann der bikubische Interpolator feine Details darstellen und – was für die vorliegende Anwendung besonders wichtig ist – die Nachbarpixel in die Skalierung mit einbeziehen. Der erhöhte Zeitaufwand gegenüber eine bilinearen Interpolation fällt in Relation zum Zeitaufwand des restlichen Verfahrens kaum ins Gewicht.

Die Implementierung des Interpolators von Hand ist nicht notwendig, da in der von mir zur Bildbearbeitung verwendeten Java Advanced Imaging (JAI) API verschiedene Interpolatoren und mit der Klasse `javax.media.jai.InterpolationBicubic` unter anderem auch ein bikubischer Interpolator bereitgestellt wird.

Mit diesem Interpolator ist es dann möglich, eine verkleinerte Form der beiden Bilder zu erstellen, und so mit dem schon oben beschriebenen Algorithmus eine ungefähre Position zu bestimmen. Diese Position ist dann noch relativ ungenau, kann aber in einem zweiten Schritt leicht korrigiert werden.

Indem man die, auf Grundlage der verkleinerten Bilder erhaltenen Werte, mit dem Faktor, mit dem die Bilder skaliert wurden, multipliziert, erhält man die ungefähre Position. In einem weiteren Schritt kann man dann mit den unskalierten Bildern – wiederum mit dem gleichen Algorithmus und entsprechenden Vorgaben für `xminwidth` und die anderen Variablen – die genaue Position feststellen.

Indem man den Bereich, der von der äußeren Schleife durchlaufen wird auf wenige Pixel unterhalb und oberhalb der gefundenen Variablen reduziert, kann man mit einer hohen Präzision und in relativ kurzer Zeit die richtige Ausrichtung der Bilder finden.

3.2 Implementierung des Image Mosaicings

Gerade bei einem so rechenintensiven Verfahren wie dem Image Mosaicing bietet es sich an, die Implementierung nicht vollständig in Java durchzuführen. Besonders die Implementierung der FFT in Java benötigt sehr viel Speicher und ist im Vergleich mit einer nativen Implementierung in C deutlich weniger performant.

Aus diesem Grund wurde die Phasenkorrelation zweifach implementiert: Eine Version benutzt eine Bibliothek, die die FFT in C implementiert. Diese sehr schnelle Implementierung der Fourier-Transformation stammt vom MIT (Massachusetts Institute of Technology) und wird als FFTW-Bibliothek (Fastes Fourier Transformation in the West) bezeichnet. Sie steht unter [31] zur freien Verfügung; die Details und den mathematischen Hintergrund der Implementation finden sich in [32].

Die Bibliothek lässt sich von Java aus über das Java Native Interface (JNI) ansprechen. Mit der JNI-API [53] stellt Java eine Schnittstelle zu nativen Programmen zur Verfügung, mit der man aus Java heraus auf die, in einer plattformabhängigen Sprache wie C oder Fortran geschriebenen Bibliotheken, zugreifen kann.

In der Praxis schränkt diese Art der Implementierung die Verwendbarkeit des Programms allerdings stark ein; so ist es z.B. nicht unter Windows zu verwenden, da der native Teil des Programms dort nicht ohne weiteres kompiliert werden kann. Des Weiteren ist die Installation durch die benötigten Bibliotheken relativ komplex.

Deshalb wurde das Verfahren zusätzlich nochmals vollständig in Java implementiert. Dies gewährleistet die Plattformunabhängigkeit des Programms und vereinfacht die Benutzung sowie die Installation deutlich.

Welche Version letztlich zum Einsatz kommt, hängt maßgeblich davon ab, ob es mehr auf die reine Performance oder auf die einfache Handhabung und eine schnelle Installation ankommt.

Sollen mehr als zwei Bilder zu einem Mosaicing zusammengesetzt werden, ist der naheliegende Gedanke, zunächst die Bilder eins und zwei zusammenzusetzen, dann diesem Bild das dritte hinzuzufügen, u.s.w.. In der Praxis scheitert dies daran, dass die Bilder schnell so groß werden, dass sie nur noch schwer handhabbar sind. Es bietet sich deshalb an, zuerst das Mosaicing zwischen Bild eins und zwei auszuführen und dann Bild zwei und drei zusammenzuführen. Dabei wird die beim ersten Mosaicing errechnete Translation zu den Koordinaten des dritten Bildes addiert. Dieses kann dann mit dieser Translation an das Mosaicing der ersten beiden Bilder angefügt werden.

Der Nachteil, dass sich durch diese Modifikation die jeweils aufeinander folgenden Bilder überschneiden müssen, kommt in der Praxis kaum zum Tragen, da man sich mit dem Mikroskop normalerweise immer von einem zum nächsten Bild bewegt.

3.3 Kontrollmechanismen

Wie die meisten Verfahrensschritte bietet auch das Image Mosaicing mehrere Möglichkeiten zur Kontrolle des Ergebnisses. Dieser Schritt ist für die automatische Verarbeitung von größter Wichtigkeit, denn bei einer großen Serie sind Fehler – und seien sie durch den Benutzer verursacht – niemals auszuschließen. Wenn nun ein Verfahren

keine Möglichkeit zur automatischen Selbstkontrolle böte, wäre der Aufwand für eine manuelle Kontrolle der Ergebnisse immens, da jedes Einzelbild betrachtet und auf Fehler hin untersucht werden müsste.

Allgemein besteht die Strategie hier, wie auch später beim Ausrichten der Schnitte darin, unterschiedliche Verfahren für die Bewertung heranzuziehen und deren Ergebnisse zu vergleichen. Beide Verfahren arbeiten mit einer sehr geringen Fehlerquote und beruhen auf unterschiedlichen Prinzipien, so dass Abweichungen einfach detektiert werden können.

Durch den hier vorgestellten automatischen Kontrollmechanismus wird zudem die Qualität der beiden Mosaicings nochmals separat beurteilt, so dass es in praktisch allen Fällen möglich sein sollte, fehlerhafte Bildpaare auszusortieren oder zur späteren manuellen Kontrolle zu markieren.

Die Qualitätskontrolle umfasst mehrere Mechanismen: Zunächst ist die Höhe des Phasenkorrelations-Maximums ein guter Marker für die Qualität der Phasenkorrelation. Der Phasenkorrelations-Algorithmus wird separat für die drei Farbkanäle rot, grün und blau durchgeführt. Unterschreitet ein Maximum die Kontrollgröße oder wurden für die drei Kanäle unterschiedliche Transformationskoordinaten berechnet, wird dieser Datensatz je nach Einstellung entweder verworfen oder markiert, so dass er zu einem späteren Zeitpunkt manuell korrigiert werden kann.

In Abbildung 3.6 sind die Phasenkorrelationsoberflächen von zwei Bildpaaren zu sehen. In (a) wurde das Bild nicht verändert. Dort ist an den Koordinaten, welche der richtigen Ausrichtung entsprechen, ein deutliches Maximum zu erkennen. In (b) hingegen wurde jeder Pixel mit einem Gausschen Rauschen von 25 versehen, d.h. es wird ein normalverteilter Zufallswert mit einer Standardabweichung von 25 zu dem Farbwert jedes Pixels hinzu addiert. Zwar werden auch hier die Bilder noch korrekt zugeordnet, allerdings zeigt der sehr viel kleinere Phasenpeak deutlich, dass die Qualität des Mosaicings abgenommen hat.

Bei der Kreuzkorrelation wird die mittlere Differenz der Farbwerte im Überschneidungsbereich gebildet. Wenn dieser Wert zu groß wird, ist auch dies ein Zeichen für einen möglichen Fehler. Die absolute Größe der einzelnen Kontrollwerte ist abhängig vom Signal/Rausch-Verhältnis der Aufnahmen und muss empirisch auf Grundlage der Bilder justiert werden.

Als letzte Kontrolle wird die Differenz der von der Phasenkorrelation und Kreuzkorrelation berechneten Transformationsparameter bewertet. Nur wenn beide Verfahren die gleiche Transformation berechnet haben, wird das Ergebnis akzeptiert.

Des weiteren ist es möglich, eine minimale oder maximale Verschiebung festzulegen. Dies ist vor allem dann sinnvoll, wenn z.B. auf Grund der Art und Weise wie die Schnitte aufgenommen wurden, nur eine Verschiebung in einer Richtung vorliegen kann.

3.4 Überblenden der Bilder

Nachdem die korrekten Translationskoordinaten ermittelt und kontrolliert wurden, muss für die weitere Auswertung aus zwei Bildern ein einzelnes Bild erstellt werden.

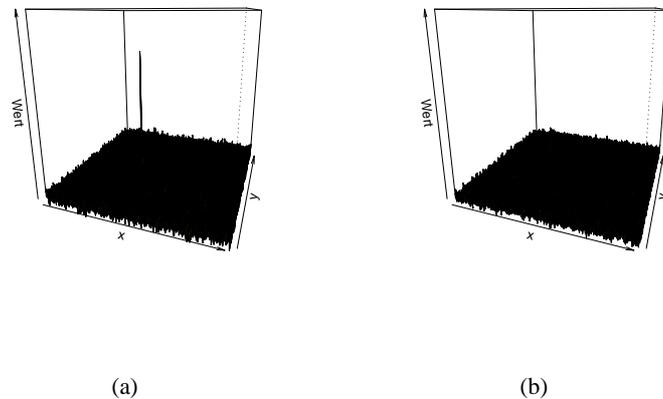


Abbildung 3.6: Vergleich zweier Phasenkorrelationsoberflächen: Bei dem Originalbild (a) ist ein hoher, vom Umfeld deutlich abgegrenzter Phasenpeak zu sehen. In (b) hingegen ist das selbst Bild mit einem Gaußschen Rauschen von 25 versehen. Dort ist das Maximum praktisch nicht auszumachen.

Verwendet man dabei ein Format wie z.B. das PNG (Portable Network Graphic) - Format, das einen transparenten Kanal beinhaltet, so kann man in diesem die Information ablegen, welche Bereiche Bilddaten und welche Bereiche nur Hintergrund enthalten.

Bildpunkte, die auf keinem der beiden Bilder enthalten sind, werden als „durchsichtig“ markiert und nur den Punkten, die in mindestens einem der Bilder enthalten sind, wird im transparenten Kanal das Attribut „undurchsichtig“ zugewiesen.

Für den Bereich, in dem sich die beiden Bilder überschneiden, gibt es vier Möglichkeiten, den Wert der Pixel zu bestimmen:

- Man verwendet den entsprechenden Wert aus dem ersten Bild
- Man verwendet den entsprechenden Wert aus dem zweiten Bild
- Man bildet den Mittelwert zwischen den beiden Werten aus dem ersten und dem zweiten Bild
- Man bildet einen gewichteten Mittelwert zwischen den Werten aus dem ersten und dem zweiten Bild.

Bei den beiden ersten Möglichkeiten kann man manchmal eine leichte Grenze zwischen den beiden Bildern erkennen. Im Allgemeinen ist diese Methode vor allem dann sinnvoll, wenn man das Mosaicing visuell testen möchte, da in dem überblendeten Bild ein Fehler teilweise nur schwer zu erkennen ist.

Im Allgemeinen sollte die dritte Form ausreichen. Hierbei weist man jedem Farbband b im neuen Bild $I(x, y, b)$ in Abhängigkeit von den Werten $I_1(x_1, y_1, b)$ im ersten und $I_2(x_2, y_2, b)$ im zweiten Bild folgenden Wert zu:

$$I(x, y, b) = \frac{I_1(x_1, y_1, b) + I_2(x_2, y_2, b)}{2} \quad (3.9)$$

Die Koordinaten x_1 , x_2 , y_1 und y_2 ergeben sich durch die zuvor berechnete Verschiebung.

Nur in wenigen Fällen kann es notwendig sein, die letzte Methode zu verwenden. Dies kann z.B. dann der Fall sein, wenn man entweder eine besonders kritische Auswertung macht, die beiden Einzelbilder im Kontrast sehr voneinander abweichen oder im Randbereich Artefakte auftreten.

Bei diesem Verfahren wird ein gewichteter Mittelwert gebildet, indem die Farbwerte mit dem Abstand des Punktes zum Zentrum des anderen Bildes multipliziert und durch die Summe der Entfernungen zu den Zentren dividiert wird. I ist hier der Farbwert im Zielbild, I_1 und I_2 die Farbwerte der Ausgangsbilder und d_1 und d_2 die Distanz zum jeweiligen Zentrum des Bildes. Die Koordinaten x_1 , x_2 , y_1 und y_2 ergeben sich wie oben erläutert durch die zuvor berechnete Verschiebung.

$$I(x, y, b) = \frac{I_1(x_1, y_1, b)d_2(x, y) + I_2(x_2, y_2, b)d_1(x, y)}{d_1(x, y) + d_2(x, y)} \quad (3.10)$$

In Abbildung 3.7 sind die vier Verfahren miteinander verglichen. Da die Unterschiede bei einem Beispiel aus der Praxis schon am Monitor nur schwer zu erkennen sind und auf einem Ausdruck praktisch kein Unterschied mehr festzustellen ist, habe ich hier ein einfarbig rotes und ein einfarbig blaues Bild ineinander überblendet.

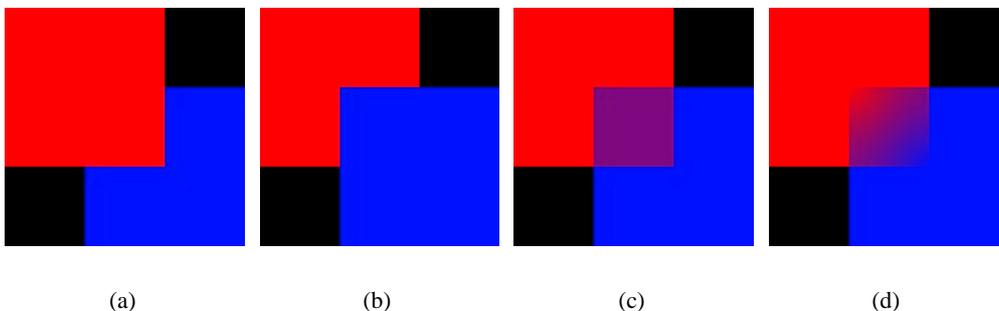


Abbildung 3.7: In der Abbildung sind unterschiedliche Methoden zum Überblenden zweier Bilder dargestellt. Zur besseren Verdeutlichung der Verfahren ist ein Bild rot, das andere blau. Ganz links ist im Überschneidungsbereich der Farbwert des ersten Bildes eingesetzt, im zweiten wird der aus dem zweiten Bild verwendet. Im dritten Bild wird bei jedem Punkt der Mittelwert und ganz rechts schließlich der gewichtete Mittelwert der Punkte gebildet.

Durch die Abbildung werden die Vorteile der einzelnen Verfahren deutlich: Wenn sich z.B. die gesamte Struktur auf einem der beiden Bilder befindet und der Rest nicht ausgewertet wird, sondern nur dem Betrachter zur Orientierung dient, kann man die erste Methode verwenden und somit eine Beeinflussung der Pixel durch die des anderen Bildes verhindern.

Normalerweise wird man jedoch eine der beiden Mittelwertbildungen bevorzugen; in der für die Berechnung benötigten Zeit unterscheiden sie sich praktisch nicht und da auch das Ergebnis weitestgehend identisch ist, bleibt es letztlich dem Geschmack überlassen, ob man die normale Mittelwertbildung vorzieht, so dass Fehler aus beiden Bildern gleichwertig eingehen oder ob man die Methode des gewichteten Mittelwertes verwendet und so einen weichen Übergang erhält. Nach dem Überblenden ist das Bild endgültig fertig und kann weiter bearbeitet werden.

3.5 Interface und Grafische Benutzeroberfläche

Um auch für ungeübte Anwender eine einfache Bedienung des Programms zu gewährleisten, wurden alle Parameter unter einer grafischen Oberfläche zusammengefasst. Diese erstellt eine Konfigurationsdatei, die dann abgearbeitet wird. So ist es möglich, die Kontrollparameter optimal auf unterschiedliche Gegebenheiten anzupassen. Zur Entwicklung komplexer Kontrollstrukturen und zur manuellen Korrektur steht eine Kommandozeilen-Version des Programms zur Verfügung.

In Abbildung 3.8 auf der nächsten Seite ist die grafische Oberfläche des Programms zu sehen. Sie ist in vier Bereiche aufgeteilt. Der erste Schirm dient dem Laden und Speichern der Einstellungen. Die so erzeugten Dateien sind einfache ASCII-Dateien, die auch manuell oder durch andere Programme erstellt werden können.

Der zweite Bildschirm ermöglicht die Auswahl der Verfahren. Es lässt sich einstellen, ob das Verfahren mit der Phasenkorrelation, der Kreuzkorrelation oder beiden Verfahren laufen soll. Ferner kann der Skalierungsfaktor für letzteres festgelegt werden. Die weiteren Felder dienen dazu, das Mosaicing wie oben beschrieben auf einen vorgegebenen Bereich zu beschränken.

Auf dem dritten Bild sind die Ein- und Ausgabeparameter zu sehen. Diese Parameter geben den Namen der Eingangs- und Ausgangsbilder an und legen das Ausgabeformat fest. Es kann außerdem angegeben werden, ob die Ausgabe gespeichert werden soll oder nicht.

Der letzte Bildschirm ist der wichtigste, da hier die Kontrollparameter eingestellt werden können. Es lassen sich alle oben beschriebenen Grenzen festlegen und ferner kann definiert werden, wie verfahren werden soll, wenn ein Fehler auftritt. Es besteht die Möglichkeit, einfach fortzufahren und den Fehler nur festzuhalten; das Verfahren kann an dieser Stelle auch abgebrochen werden oder man verwendet zunächst nur die Phasenkorrelation und erst, wenn diese ein Problem meldet (z.B. weil der Phasenpeak zu klein ist), wird das etwas langsamere Kreuzkorrelations-Verfahren zusätzlich angewendet.

3.6 Validierung des Verfahrens

Im Allgemeinen arbeiten die beiden Mosaicing-Verfahren fehlerfrei und es ist nach den bisherigen Erfahrungen auch nicht anzunehmen, dass der Algorithmus durch real auftretende Bildfehler fehlerhaft arbeiten würde. Sämtliche in dieser Arbeit verwendeten Bildpaare wurden korrekt ausgerichtet.

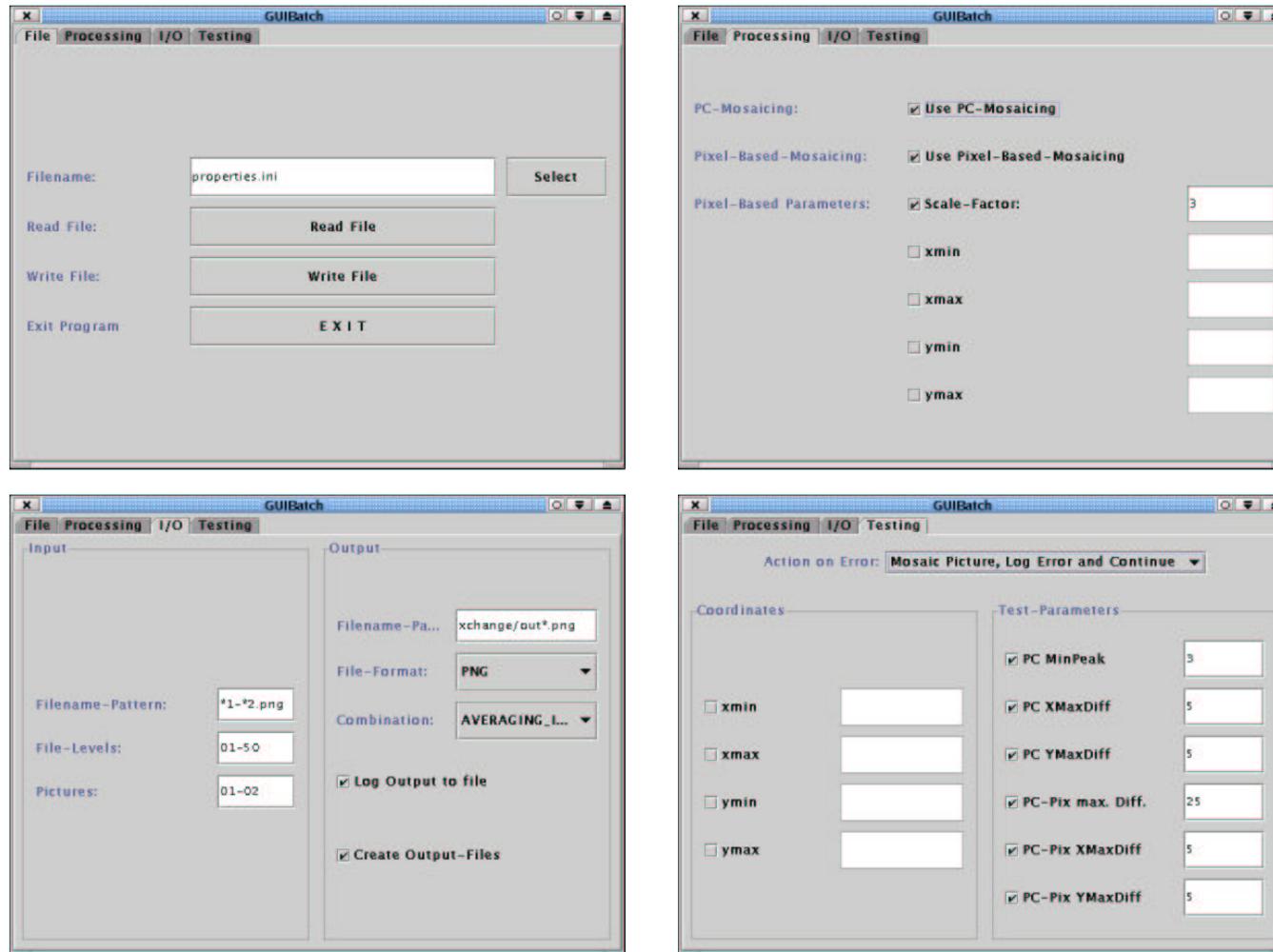


Abbildung 3.8: GUI des imosic-Programms. In der Grafik sind die einzelnen Interfaceteile der grafischen Benutzeroberfläche zu sehen.

Auch wenn, wie in Abbildung 3.9 der Kontrast des Bildes möglichst weit gespreizt und das Resultat mit Falschfarben dargestellt wird, ist es nicht möglich, den Übergangsbereich zu identifizieren. Von den 50 Probepildpaaren wurde nur bei einem einzigen ein falsch-positiver Fehler gemeldet, da das Bild im blauen Kanal leicht gestört war.

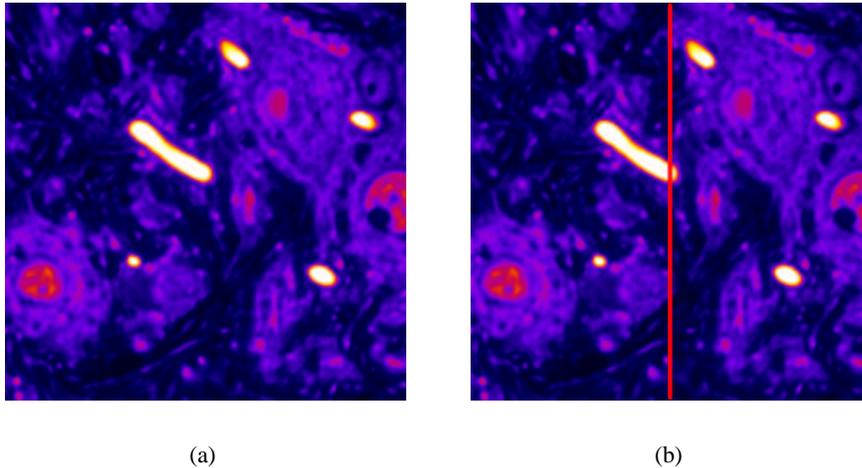


Abbildung 3.9: Ausschnittsvergrößerung eines Image Mosaics. In der Abbildung rechts ist die Grenze des zweiten Bildes eingezeichnet. Man kann selbst bei dieser Darstellung die Grenze im linken Bild nicht erkennen.

Das Verfahren arbeitet selbst bei einem sehr hohen Rauschen immer noch zuverlässig. Erst bei einem Gaußschen Rauschen von deutlich mehr als 50 ist bei einigen Bildern keine Zuordnung mehr möglich. Da bei einem solchen Bild allerdings auch die Zellen kaum noch zu erkennen sind und solche Probleme in der Praxis nicht vorkommen, wird hier auf eine Betrachtung der Robustheit verzichtet.

Eine große Fehlerquelle stellt jedoch die Aufnahme am Mikroskop dar, weil es hier leicht vorkommen kann, dass sich zwei Bilder entweder nur sehr gering oder gar nicht überlappen. In diesen Fällen wird das Mosaicing fehlerhafte Ergebnisse liefern.

Um diesen Fall zu simulieren, wurde das zweite Bild eines jeden Paares gegen ein zufällig gewähltes anderes ausgetauscht. Hierdurch sind die Bilder zwar hinsichtlich ihrer Struktur und Farbgestaltung immer noch ähnlich, allerdings kann kein sinnvolles Mosaik mehr erstellt werden. Dies entspricht etwa der Situation, wenn ein falscher Bildausschnitt gewählt wurde.

Das Verfahren wies bei allen Bildern auf einen Fehler hin. Von den zwölf Kontrollen meldeten bei dieser Serie pro Bild zwischen 5 und 12 (Mittelwert: 9,3 +/- 1,8) einen Fehler. In Abbildung 3.10 ist gut zu erkennen, dass bei allen untersuchten Kontrollen beim normalen Mosaicing nur der eine oben erwähnte Fehler im blauen Kanal auftritt, während die fehlerhaften Bilder alle einen sehr hohen Fehler aufweisen.

Man kann also festhalten, dass alle Bilder korrekt ausgerichtet wurden, nur sehr selten (ein Mal) ein falsch-positiver Fehler gemeldet wurde und die hier künstlich erzeugten Fehler alle erkannt wurden.

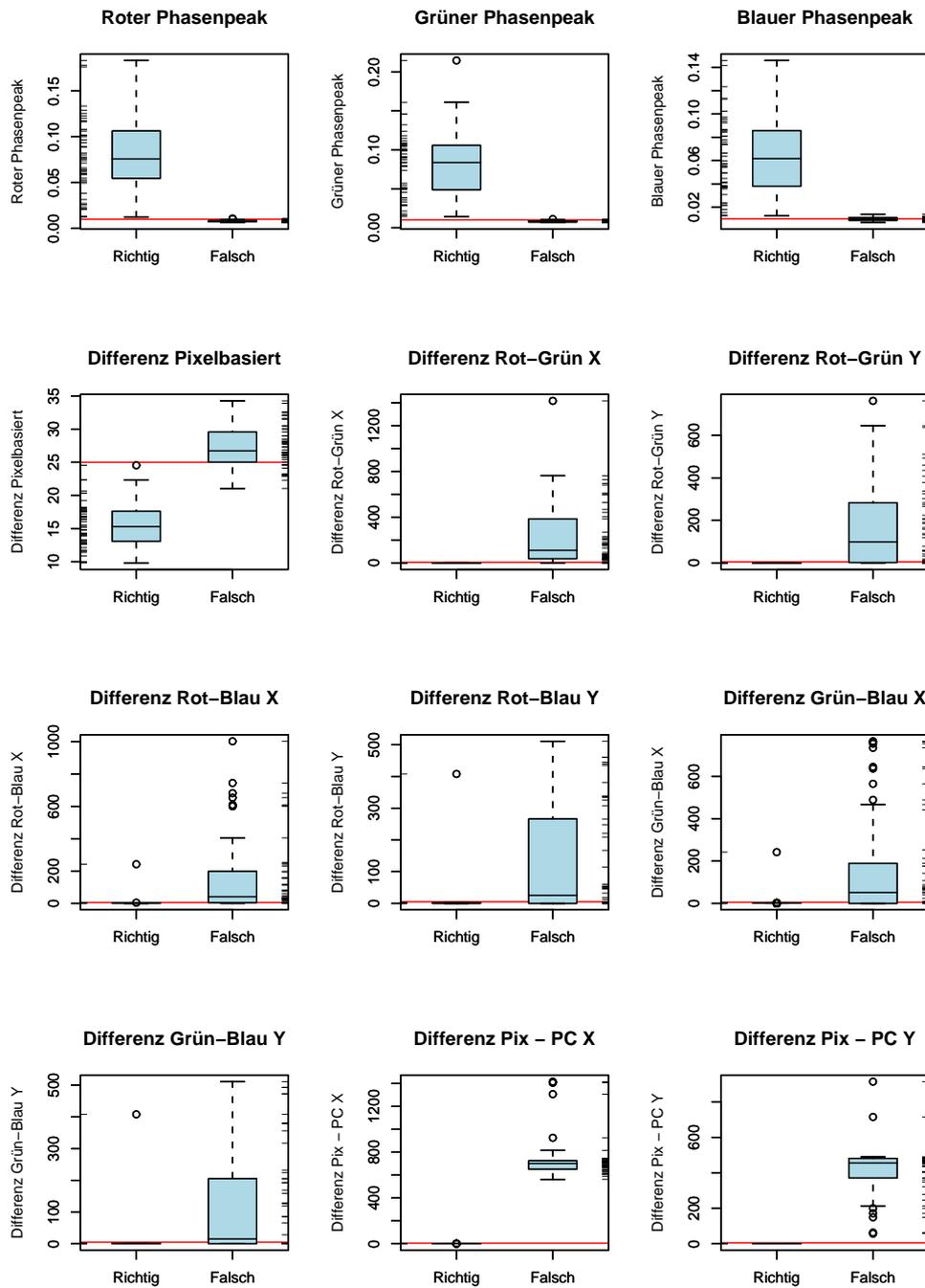


Abbildung 3.10: Ergebnis der Kontrollmechanismen bei den richtig und falsch kombinierten Bildpaaren. Die rote Linie zeigt die gesetzte Fehlergrenze an.

3.7 Fazit

Das hier vorgestellte Verfahren des Image Mosaicings stellt eine Kombination von zwei konventionellen Algorithmen dar, wobei die Kreuzkorrelation zunächst massiv modifiziert werden musste damit sie in der Praxis angewendet werden kann.

Gerade für automatische Rekonstruktionen einer großen Anzahl von Bildern bietet sich das Verfahren an, da es durch die integrierte Kontrolle der Ergebnisse und die automatische Korrektur durch ein zweites Verfahren praktisch fehlerfrei arbeitet.

So funktioniert normalerweise die Phasenkorrelationsmethode sehr sicher; bei leichten Rotationen in einem der Bilder treten jedoch schnell Fehler auf. Umgekehrt scheitert bei wenig strukturierten Bildern die Kreuzkorrelation häufig.

Auch wenn das hier vorgestellte Verfahren bisher in allen Tests fehlerfrei gearbeitet hat, sind Fehler – und sei es durch menschliches Versagen – nie auszuschliessen. Deshalb sind für ein effektives Arbeiten – insbesondere bei großen Serien – die Kontrollparameter wichtig, da nur so entschieden werden kann, ob das Bild korrekt ausgerichtet wurde oder nicht – ohne dass eine umfangreiche manuelle Kontrolle notwendig wäre.

Durch diesen Verfahrensschritt ist die Begrenzung der lateralen Ebene praktisch aufgehoben. Die maximal mögliche Ausdehnung einer Struktur im lateralen Bereich ist nur noch durch den zur Verfügung stehenden Speicherplatz limitiert.

Mustererkennung

Nachdem durch das Image Mosaicing eine Erweiterung der lateralen Ebene erfolgt ist, wäre der nächste logische Schritt, die axiale Ebene ebenfalls zu erweitern. Für diesen Schritt ist es wiederum notwendig, Bilder aneinander auszurichten – in diesem Fall die Aufnahmen von aufeinander folgenden Schnitten.

Diese Problemstellung unterscheidet sich von der im vorherigen Kapitel grundlegend. Innerhalb einer Ebene unterscheiden sich die Bilder lediglich im Rahmen der Messgenauigkeit von Kamera und Mikroskop voneinander. Dieser, meist deutlich unterhalb von 5 Farbstufen liegende Fehler, ist so gering, dass in der Praxis eine exakte Zuordnung einzelner Pixel zueinander möglich ist.

Wie in Kapitel 5 gezeigt wird, sind die Verfahren des Image Mosaicings auf das Problem der Zuordnung in axialer Richtung nicht anwendbar. Dies ist darauf zurückzuführen, dass die Strukturen sich von einer zur nächsten Ebene deutlich verändern und somit auf der Ebene einzelner Pixel keine Zuordnung möglich ist.

Als Alternative bietet sich deshalb die Orientierung an sogenannten Landmarken an. Dies sind markante Punkte innerhalb des Bildes, die auch bei einer größeren Abweichung noch korrekt zugeordnet werden können.

In den Schnitten bieten sich zwei Klassen von Strukturen als Landmarken an: Zum einen die Kapillaren, zum anderen die Zellkerne. Zu Anfang dieses Kapitels wird beschrieben, warum die Wahl auf die Zellkerne fiel. Diese sollen zum Ausrichten der Schnittebenen entlang der z-Achse verwendet werden.

Dazu ist es notwendig, die Zellkerne vom Hintergrund zu trennen. Diese Extraktion gliedert sich in mehrere Schritte, die im weiteren Verlauf des Kapitels beschrieben werden.

Da bei den verwendeten Verfahren die Farbe als Merkmal herangezogen werden soll, muss zunächst gewährleistet sein, dass die Zellkerne in allen Bildern eine identische Färbung aufweisen. Da die meisten in diesem Bereich normalerweise verwendeten Verfahren hierfür nicht geeignet sind, wurde ein speziell auf die Bedürfnisse der Histologie angepasstes Verfahren entwickelt. Dieses wird im Vergleich zu den traditionellen Methoden präsentiert.

Nachdem der Farbraum mit diesem Verfahren korrigiert wurde, werden zunächst die Kapillaren entfernt, so dass diese nicht mehr mit den Zellkernen verwechselt werden können.

Danach werden die Zellkernbereiche vom restlichen Hintergrund mittels des von mir entwickelten *Color-Sensitive Edgedetection*-Algorithmus getrennt. Aus den so vorbereiteten Bildern werden – gegebenenfalls nach einer weiteren Nachbearbeitung der Kontraste – die Zellkerne extrahiert.

Den Abschluss dieses Kapitels bildet eine Evaluation der Ergebnisse. Dabei werden die Ergebnisse von zwei menschlichen Experten dem Computer-System gegenübergestellt und in ihrer Präzision verglichen. Auf diese Weise kann gezeigt werden, dass der hier entwickelte Algorithmus einem menschlichen Experten weitestgehend ebenbürtig ist.

4.1 Auswahl der Landmarken

Betrachtet man die Problemstellung des Ausrichtens von Gehirnschnitten unvoreingenommen, so scheint die Verwendung der Kapillaren als Landmarken die bessere Wahl zu sein. Dies liegt zum einen daran, dass sie sehr einfach zu extrahieren sind und zum anderen an einem Phänomen, das im Englischen als *learning tower*-Problem bezeichnet wird. Wenn man eine Struktur, die natürlicherweise entlang der Diagonalen eines Würfels verläuft, an sich selbst ausrichtet, so wird ein gerader Turm entstehen, der nicht den wahren Gegebenheiten entspricht. Deshalb wäre es theoretisch besser, die Schnitte an den Kapillaren auszurichten; denn würde man die Zellkerne zum Ausrichten und anschließend zur Rekonstruktion verwenden, könnte genau dieses Problem auftreten.

Aus diesem Grund wurden im Rahmen dieser Arbeit umfangreiche Versuche unternommen, die Kapillaren zum Ausrichten der Bilder heranzuziehen. Doch selbst ein sehr aufwendiger und komplexer Ansatz, bei dem zunächst die Kapillaren extrahiert, danach deren Momente [43,49,67] berechnet und diese mittels eines Neuronalen-Netz-Klassifikators [43,67] verglichen wurden, ergab nur eine unzureichende Erfolgsquote.

So wurden 39 Bildpaare getestet; davon waren 26 Paare korrekt ausgerichtet, neun waren leicht verschoben und vier der Bilder waren vollkommen falsch ausgerichtet.

Bei den leicht verschobenen Bildern hätte man durch eine Nachkorrektur sehr wahrscheinlich noch einen großen Prozentsatz korrekt ausrichten können – dennoch ist für den praktischen Einsatz eine Fehlerquote zwischen zehn und dreißig Prozent nicht akzeptabel.

Versucht man den für diese hohe Fehlerrate verantwortlichen Programmteil zu identifizieren, wird man feststellen, dass ein solcher nicht existiert, da der Fehler nicht im Programm, sondern in der Aufgabenstellung steckt. Das Programm arbeitet unter Berücksichtigung der Problematik sogar äußerst effektiv.

Wenn man in zwei aufeinander folgenden Bildern die Kapillaren wie in Abbildung 4.1 auf der nächsten Seite unterschiedlich einfärbt und den Überschneidungsbereich durch eine dritte Farbe (hier violett) kennzeichnet, kann man sehen, dass sich die Kapillaren von einer zur nächsten Ebene deutlich verändern.

Nur in sehr wenigen Fällen sind die Kapillaren nahezu identisch (violetter Bereich). In den meisten Fällen kommt es entweder zu gar keiner Überschneidung oder nur zu einer teilweisen Überlappung.

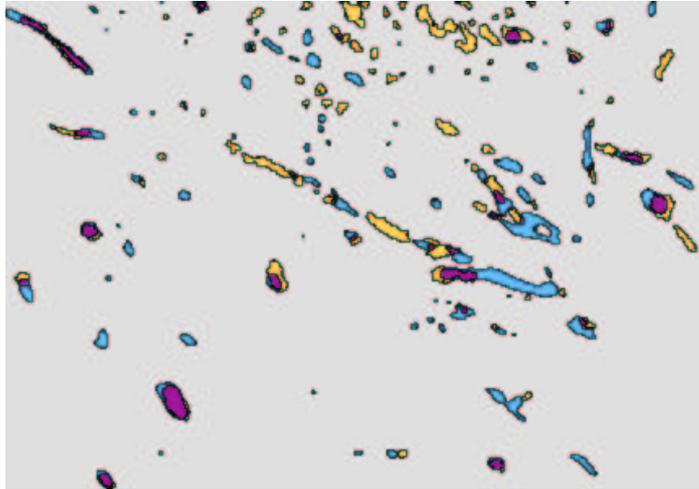


Abbildung 4.1: Problem beim Ausrichten der Bilder mit den Kapillaren als Referenz: Die Kapillaren des ersten Bildes (blau) und die des zweiten Bildes (Gelb) überschneiden sich nur in sehr wenigen Bereichen (violett).

Die möglichen Veränderungen einer Kapillare sind in Abbildung 4.2 auf der folgenden Seite schematisch dargestellt. Im Idealfall sollte eine Kapillare wie in (a) in zwei aufeinander folgenden Ebenen in der gleichen Größe und an der gleichen Position zu sehen sein. Dies ist leider nur sehr selten der Fall (siehe oben). In vielen Fällen verändert die Kapillare zumindest ihre Form und Größe (b) oder spaltet sich in zwei Äste auf (c). Leider ist es auch sehr häufig der Fall, dass eine Kapillare völlig verschwindet (d).

Ist eine Kapillare in der nächsten Ebene nicht sichtbar, so kann dies unterschiedliche Gründe haben. Sie kann in einem der Bilder eine so geringe Größe aufweisen, dass sie bei der Mustererkennung ignoriert wurde, sie kann aber auch so schräg verlaufen, dass sie nicht weiter beachtet wurde oder aber sie wurde beim Schneiden in ihrer Form so verändert, dass sie nicht mehr als Kapillare klassifiziert wird.

Das Problem, das sich bei der Verwendung der Kapillaren als Landmarken stellt, beruht zu einem großen Teil auf den Eigenschaften derselben, so dass auch durch eine weitere Verbesserung des Programms kaum eine geringere Fehlerrate zu erreichen wäre.

Außer den Kapillaren können jedoch nur die Zellkerne als Landmarken dienen, da das umliegende Gewebe keine weitere Struktur enthält, die genügend Charakteristika aufweist, um eindeutig identifiziert zu werden.

Wie sich bei den Versuchen herausstellte, hat das *learning tower*-Problem in der Praxis nur eine geringe Bedeutung. Dies hängt mit mehreren Faktoren zusammen: Da die Zellkerne im Vergleich zum Gesamtnukleus relativ klein sind und sich nur über weniger Ebenen erstrecken, können sie kaum eine Richtung vorgeben. Da sie zudem

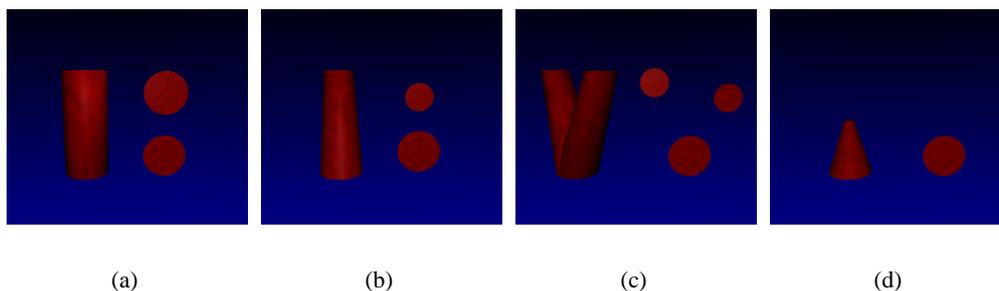


Abbildung 4.2: Probleme bei der Zuordnung von Segmenten: Auf der linken Seite einer jeden Abbildung befindet sich eine dreidimensionale Darstellung der Kapillaren, auf der rechten die Darstellung eines Schnittes in der oberen und der unteren Ebene. Wenn die Kapillare wie in 4.2(a) verläuft, ist eine Zuordnung relativ problemlos möglich. Wenn hingegen die Kapillare entweder wie in 4.2(b) ihre Größe ändert, sie sich wie in 4.2(c) aufteilt oder wie in 4.2(d) ganz verschwindet, so wird die Zuordnung schwierig bzw. wie im letzten Fall sogar unmöglich.

näherungsweise kugelförmig sind, ist es zudem relativ unwahrscheinlich, dass sie in einer bestimmten Richtung schräg verlaufen. Und letztlich werden für das Ausrichten jeweils mehrere Zellen verwendet. Es ist in der Praxis nahezu ausgeschlossen, dass alle eine Schräglage in die gleiche Richtung aufweisen.

Ein anderes Problem wiegt hingegen wesentlich schwerer. So ist die Identifikation der Zellkerne im dreidimensionalen Raum relativ einfach, da man bestimmte Formcharakteristika zu Grunde legen kann: Der Zellkern sollte näherungsweise rund sein, eine bestimmte Größe nicht unter- und eine andere nicht überschreiten. Auf diese Weise ist ein Zellkern recht einfach zu identifizieren. Um die dreidimensionale Form jedoch bestimmen zu können, müssen die Schnitte bereits ausgerichtet sein. Benötigt man nun die Zellkerne, um die Schnitte auszurichten, entsteht ein Paradoxon, das sich nur schwer auflösen lässt.

Die einzige Möglichkeit besteht in einem mehrstufigen Prozess, bei dem man zunächst die Zellkerne identifiziert, an diesen dann die Schnitte ausrichtet, um anschließend nochmals eine Nachkorrektur vorzunehmen und fehlerhafte Schnitte auszusortieren. Der erste Schritt, die Identifikation der Zellkerne, wird im weiteren Verlauf des Kapitels erläutert.

4.2 Farbkonstanz

Die meisten Verfahren, die in der Literatur zur Mustererkennung eingesetzt werden, sind vor allem darauf ausgerichtet, Objekte an Hand ihrer Struktur, Form oder geometrischen Eigenschaften zu erkennen. Die Farbe eines Objekts wird nur sehr selten für die Objekterkennung herangezogen. Dies hängt neben der bis vor ca. 15 Jahre noch sehr raren Verbreitung von farbsensitiven Sensoren vor allem damit zusammen, dass die Farbe eines Objektes von der Leuchtkraft $E(\lambda)$, dem Anteil des Lichts $S(\lambda)$, der von der Oberfläche reflektiert wird und der Antwortfunktion des Sensors $R_i(\lambda)$ abhängig ist. Dabei entspricht i dem Typ des Sensors (hier rot, grün und blau) und die

Antwort ρ_i des Sensors i , ist definiert als:

$$\rho_i = \int_{\lambda} R_i(\lambda)S(\lambda)E(\lambda)d\lambda \quad (4.1)$$

Diese Abhängigkeit soll den Gegebenheiten des menschlichen visuellen Systems entsprechen und stellt die Basis des CIE¹-Standards [96]. Es ist möglich, die Funktionen $S(\lambda)$ und $E(\lambda)$ als lineare Modelle darzustellen. Auf dieser Grundlage wurden unterschiedliche Erklärungsansätze, wie das menschliche Sehsystem seine Farbkonstanz erreicht, entwickelt.

Während das menschliche Sehsystem unterschiedliche Umweltbedingungen automatisch kompensiert, ist das Problem, unter verschiedenen Bedingungen die gleiche Farbe zu erkennen in einer Software nur schwer lösbar.

Im folgenden sollen verschiedene Algorithmen für die Berechnung der Farbkonstanz (im Englischen *color constancy*) vorgestellt und auf einen beispielhaften Bildausschnitt aus zwei Bildern angewendet werden. Im zweiten Teil dieses Abschnitts soll gezeigt werden, warum diese Modelle auf die histologischen Präparate nicht direkt anwendbar sind und es deshalb notwendig war, einen neuen Algorithmus zu entwickeln.

In diesem Kapitel werden Aufnahmen verwendet, die noch nicht mit der in Kapitel 2.2.2 gezeigten Farbkorrektur bearbeitet wurden. Natürlich wurde bei der Serie, welche auch für die Validierung des Verfahrens verwendet wurde, die weiter vorne beschriebene Aufnahmetechnik verwendet; jedoch ist der Effekt einer Farbanpassung bei einem durch Transformationsmatrix und Lookup-Table veränderten Bild optisch nur sehr schlecht zu beurteilen, da z.B. der grüne Kanal in einem Bereich liegt, in dem das menschliche Auge Kontraste nur sehr schlecht wahrnimmt. Die Verwendung der „normal gefärbten“ Bilder dient also nur der Veranschaulichung der Verfahren und hat ansonsten keinerlei Bedeutung.

4.2.1 Klassische Algorithmen zur Farbkonstanz

Es gibt mehrere Möglichkeiten, eine Farbkorrektur vorzunehmen. Eine einfache Möglichkeit ist es, den Farbraum zu wechseln. In dem normalerweise verwendeten RGB-Farbraum werden die verschiedenen Farben durch eine Kombination von Rot-, Grün- und Blau-Werten zwischen 0 und 255 dargestellt. Ein anderes Beispiel für einen Farbraum ist der vor allem im Bereich der Printmedien verbreitete CMYK-Farbraum, bei dem die Farben durch die Kombination von Cyan, Magenta, Yellow und Schwarz dargestellt werden. Bei dem folgenden Verfahren wird der sogenannte HSI-Farbraum verwendet, bei dem zwei der drei Komponenten von der Beleuchtung unabhängig sind.

Im HSI-Farbraum werden die Farben durch eine Kombination aus Hue (Farbton), Saturation (Farbsättigung) und Intensity (Intensität) dargestellt. In Abbildung 4.3 ist ein Beispielbild in der oberen Reihe in die Kanäle Rot, Grün und Blau, in der unteren Reihe in Hue, Saturation und Intensity zerlegt.

Der RGB-Farbraum lässt sich wie folgt in den HSI-Farbraum überführen [100]:

¹Commission International d'Éclairage, bzw. International Commission of Illumination

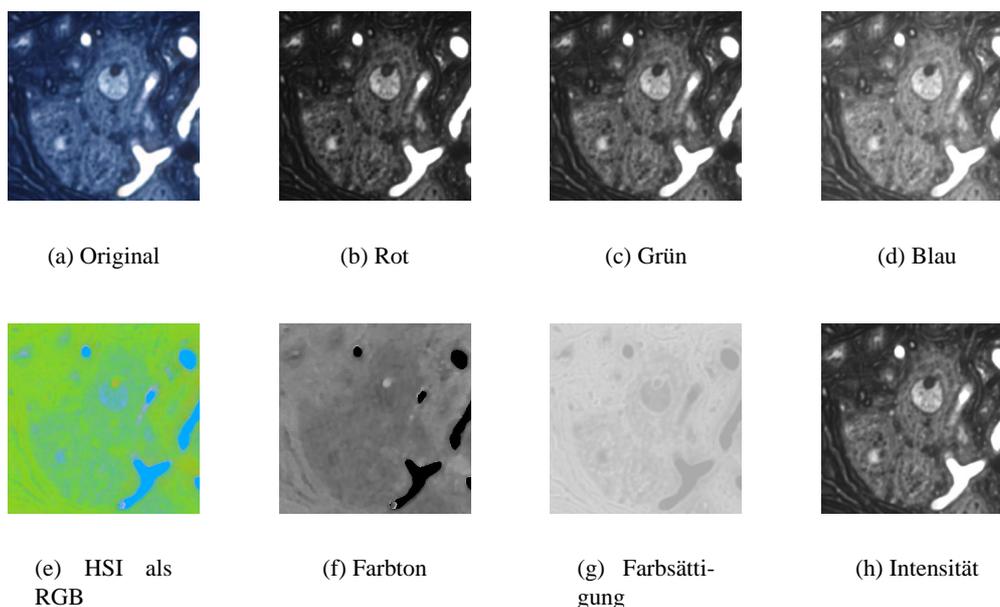


Abbildung 4.3: Ein Beispielbild in unterschiedliche Farbräume zerlegt. Zur Illustration wurde in Abbildung (e) ein Bild aus den drei Kanälen H,S und I zusammengesetzt, wobei H als roter, S als grüner und I als blauer Kanal eingesetzt wurden.

$$h = \arctan \frac{3^{\frac{1}{3}}(g-b)}{(r-g)+(r-b)} \quad (4.2)$$

$$s = 1 - \frac{\min(r,g,b)}{r+b+g} \quad (4.3)$$

$$i = r + g + b \quad (4.4)$$

Die Funktion $\min(a, b, c)$ steht dabei für den kleinsten der drei Werte a , b und c .

Um die Auswirkungen eines Korrekturalgorithmus bestimmen zu können, wurde das im folgenden beschriebene Verfahren entwickelt:

Es wurden in zehn unterschiedlichen Bildern jeweils zwischen acht und zehn Zellen ausgewählt. Bei diesen Zellen wurden mit dem Programm ImageJ [50] die Übergangsbereiche zwischen Hintergrund, Soma und Zellkern entlang einer Linie markiert. Aus diesen Daten berechnet ein Programm eine Regressionsgerade. Deren Verlauf bildet das Profil, das durch die einzelnen Punkte in Abschnitte unterteilt wird.

In Abbildung 4.4 auf der nächsten Seite ist das Ergebnis einer solchen Analyse für die unveränderten Bilder zu sehen. Dabei sind zwei Schnitte (1 und 2, 3 und 4, ...) jeweils direkt aufeinander folgend, während zwischen den Paaren ein Abstand von je $140 \mu\text{m}$ liegt.

In der folgenden Abbildung 4.5 sind die gleichen Plots gezeigt – allerdings nach der Transformation in den HSI-Farbraum. Man kann deutlich sehen, dass alle die Werte

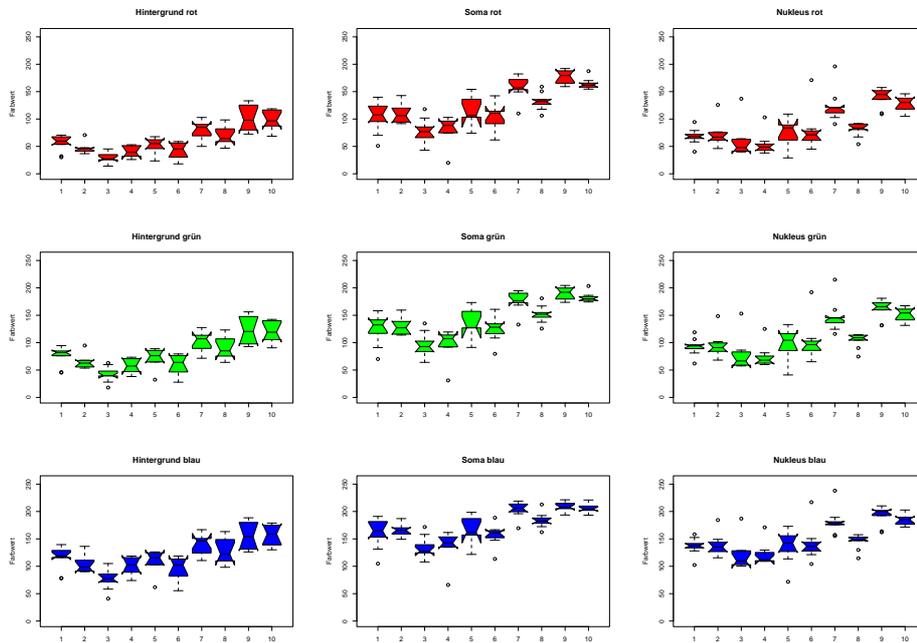


Abbildung 4.4: Ergebnis des Profil-Plots ohne jegliche Korrektur

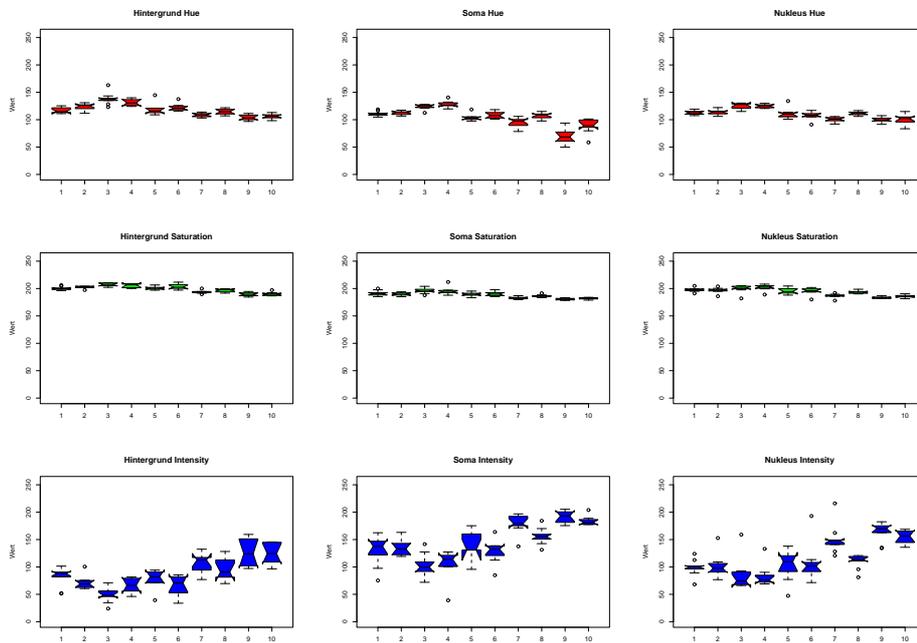


Abbildung 4.5: Ergebnis des Profil-Plots nach der Korrektur der Werte in den HSI-Farbraum.

für Hue und Saturation zwischen den einzelnen Segmenten kaum noch variieren. Dies wäre ein fast perfekte Voraussetzung für die weitere Verarbeitung, wenn nicht auch der Kontrast zwischen Nukleus und Soma fast vollständig verschwunden wäre. Dies lässt sich auch an Hand von Abbildung 4.3 (f) und (g) gut nachvollziehen. Der Farbton unterscheidet sich bei Soma und Nukleus praktisch gar nicht und auch die Farbsättigung ist nur minimal unterschiedlich.

Eine andere sehr einfache und pragmatische Form der Anpassung ist das Normalisieren der drei Kanäle [68,96]:

$$R = \frac{r}{r+g+b} \quad (4.5)$$

$$G = \frac{g}{r+g+b} \quad (4.6)$$

$$B = \frac{b}{r+g+b} \quad (4.7)$$

Dieses Verfahren hat normalerweise den gravierenden Nachteil, dass der Informationsgehalt von drei Werten auf zwei reduziert wird, da sich der dritte Wert vollständig aus den beiden anderen herleiten lässt:

$$B = \frac{b}{r+g+b} = \frac{1-r-g}{r+g+b} \quad (4.8)$$

Bei manchen Bildern kann es durch das Normalisieren zu einem deutlichen Informationsverlust kommen. Dieser Nachteil kommt bei den histologischen Präparaten kaum zum Tragen, da hier die drei Kanäle sehr eng miteinander korreliert sind (siehe Abbildung 2.4 in Kapitel 2.2.2). Auch wenn bereits ein Teil der redundanten Informationen durch die Anpassung der Aufnahmetechnik eliminiert wurde, ist die Farbinformation doch immer noch zweifach vorhanden. Zum einen ist der Farbwert über den roten und grünen Kanal verteilt gespeichert und zum anderen erhält der blaue Kanal immer noch die mit dieser Information ebenfalls korrelierte Originalinformation. Durch diesen starken Zusammenhang der Kanäle geht nur wenig Information verloren.

Der entsprechende Plot für die Korrektur durch Normalisieren des RGB-Farbraums ist in Abbildung 4.10 auf Seite 69 zu sehen. Auch hierbei ergibt sich eine deutliche Verbesserung der Farbabweichung; auch wenn der Kontrast zwischen den einzelnen Bereichen (Hintergrund, Soma und Zellkern) hier bereits stärker ist als bei der Transformation in den HSI-Farbraum, ist doch insbesondere der Unterschied zwischen Soma und Nukleus zu sehr reduziert.

Zwar kann man in Abbildung 4.6 ganz deutlich sehen, dass die Unterschiede zwischen Soma und Nukleus (gerade im roten Kanal z.B.) deutlich größer sind, als bei der Normalisierung mittels des HSI-Farbraumes, allerdings ist leider auch die Schwankung zwischen den einzelnen Schnitten deutlich größer als bei der HSI-Normalisierung, so dass der größere Kontrast teilweise durch die Schwankungsbreite der Werte kompensiert wird.

Es sollen im folgenden noch drei weitere Algorithmen, die traditionellerweise für die Farbkorrektur verwendet werden vorgestellt und auf ihre Tauglichkeit für die Verwendung bei histologischen Präparaten untersucht werden.

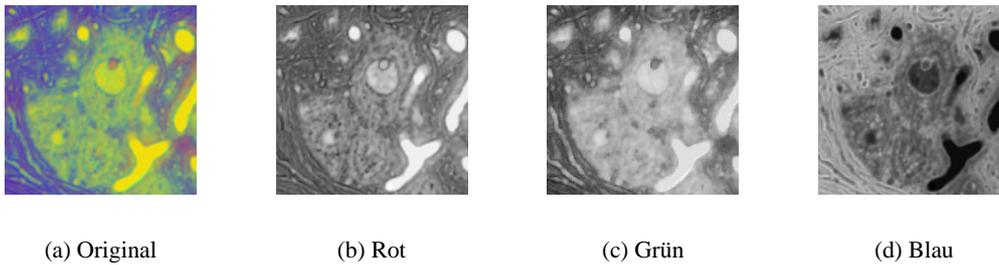


Abbildung 4.6: Das Beispielbild aus Abbildung 4.3 wurde normalisiert und hier sowohl als zusammengesetztes Bild als auch als separate RGB-Kanäle dargestellt.

Während die bisherigen beiden Verfahren auf jedes Bild anwendbar sind, gehen die folgenden Algorithmen von einer Voraussetzung aus, die jeweils namensgebend für das Verfahren war.

Das erste Verfahren, nach seiner Voraussetzung *White World*-Verfahren [70] genannt, geht davon aus, dass in jeder Szene ein rein weißes Element vorhanden ist. Deshalb kann eine Farbkorrektur einfach dadurch erreicht werden, dass die Farben des Bildes so angepasst werden, dass der hellste Punkte weiß ergibt.

Dieser Ansatz ist bei den histologischen Präparaten von vornherein zum Scheitern verurteilt, da immer ein absolutes Weiß (mit den Werten (255,255,255) im RGB-Farbraum) vorhanden ist. Dieses Weiß entspricht den Kapillaren, ist damit von der Färbung nicht beeinflusst und kann ergo nicht zum Normieren des Farbraumes verwendet werden.

Der entgegengesetzt Ansatz – den ich *Black World*-Ansatz genannt habe – ist hingegen durchaus einsetzbar. Dabei werden die histologischen Schnitte so normiert, dass in jedem Bild zumindest ein Punkt schwarz ist. Dazu wird die Distanz d eines jeden Pixels $p_{i,j}$ zu dem RGB-Tupel (0,0,0) berechnet. Dabei ist b die Anzahl der Bänder (im RGB-Farbraum drei):

$$d(p_{i,j}) = \sqrt{\sum_{i=0}^b (p_{i,j,b})^2} \quad (4.9)$$

Derjenige Pixel, der den minimalen Abstand zum Farbwert Schwarz aufweist, wird dem Vektor $min_k, k = 0..b$ zugewiesen. Die Werte des Ausgabebildes $o_{i,j,k}$ ist wie folgt definiert:

$$o_{i,j,k} = \frac{255 \cdot (p_{i,j,k} - min_k)}{255 - min_k} \quad (4.10)$$

Wie in Abbildung 4.8 auf Seite 68 und den entsprechenden Bildern in der oberen Reihe von Abbildung 4.7 zu sehen ist, bringt auch dieser Ansatz nicht den gewünschten Erfolg. Die Unterschiede zwischen den einzelnen Schnitten sind auch bei diesem Verfahren zu groß.

Der sogenannte *Gray World*-Ansatz [17] geht von einer anderen Annahme aus. Hier ist die Voraussetzung der Farbkorrektur, dass der Mittelwert eines Bildes immer ein Grauwert ist. Zunächst wird dazu die Summe aller Pixel $p_{i,j}$ eines bestimmten Farbkanales k berechnet:

$$sum_k = \sum_{x=0}^{breite} \sum_{y=0}^{hoehe} p_{i,j,k} \quad (4.11)$$

Diese Summe wird dann verwendet, um das Spektrum zu normalisieren und das Ausgabebild mit den Pixeln $o_{i,j,k}$ zu berechnen:

$$o_{i,j,k} = \frac{p_{i,j} \frac{hoehe \cdot breite}{baender}}{sum_k} \quad (4.12)$$

Dabei entspricht *hoehe* der Anzahl der Pixel in y- und *breite* der Anzahl der Pixel in x-Richtung. *baender* ist die Anzahl der Farbkanaäle; bei einem RGB-Bild drei.

Zwar ist das Ergebnis der Gray World-Normalisierung gerade im blauen Kanal deutlich besser als der zuvor vorgestellte Black World-Ansatz – jedoch ist auch hier die Streuung in vielen Fällen (z.B. zwischen den Schnitten 8 und 9) immer noch deutlich größer als der Unterschied zwischen Soma und Nukleus.

Als letzten Algorithmus möchte ich in diesem Kapitel eine sehr elegante, von Graham Finlayson [29] entwickelte Methode vorstellen, die zwei der bereits vorgestellten Verfahren kombiniert und so eine optimale Farbkorrektur erreicht. Die Methode beruht schlicht darauf, iterativ zwei der zuvor vorgestellten Ansätze, nämlich die Normalisierung der RGB-Kanäle und den Gray World-Ansatz, abwechselnd auszuführen. In dem oben genannten Artikel zeigen die Autoren, dass dieser Algorithmus für jedes Bild gegen eine eindeutige Lösung konvergiert. Bis zum Auftreten des Konvergenzkriteriums, dass beide Algorithmen die gleiche Lösung ergeben, vergehen normalerweise maximal vier bis fünf Iterationszyklen.

Wie in Abbildung 4.11 auf Seite 69 deutlich zu sehen ist, bringt dieser Algorithmus eine nahezu perfekte Farbkorrektur. Die Farbkorrektur ist die mit Abstand beste aller Algorithmen, allerdings ist der Farbraum so weit normiert, dass auch der Kontrast zwischen den Strukturen sehr gering ist.

Diese Methode wäre sehr erfolgreich anzuwenden, wenn man den Farbraum nicht durch Ganzzahlen zwischen 0 und 255, sondern durch die Fließkommazahlen beliebiger Größe darstellte. In diesem Fall würde das Verfahren eine gute Korrektur erbringen, ohne jedoch Kontrastinformationen zu verlieren. Ich habe mich in dieser Arbeit aus den bereits früher angeführten Gründen wie Speicherplatz, praktische Handhabung und Rechenleistung gegen diese Möglichkeit entschieden. Sollten Rechenleistung und Speicherplatz keine Rolle spielen, so ist dieses Verfahren allen anderen vorzuziehen.

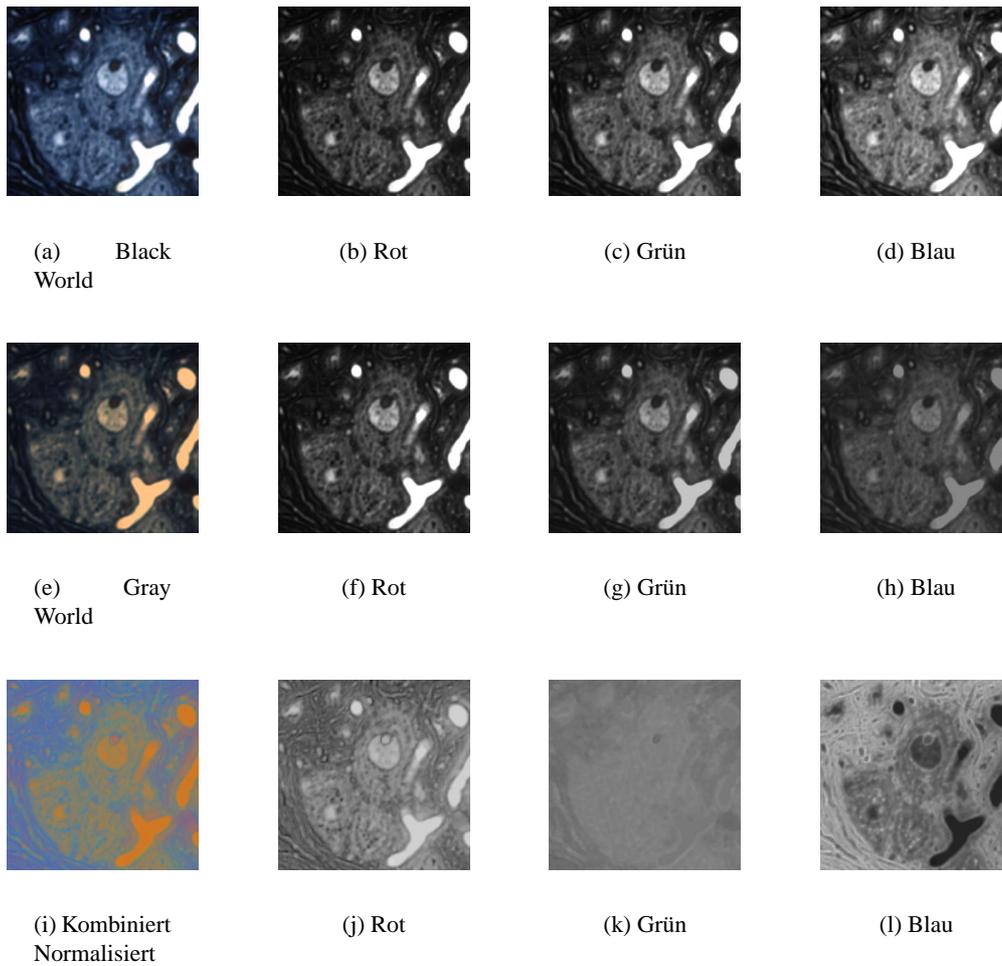


Abbildung 4.7: Das Beispielbild aus Abbildung 4.3 wurde normalisiert und hier sowohl als zusammengesetztes Bild als auch als separate RGB-Kanäle dargestellt.

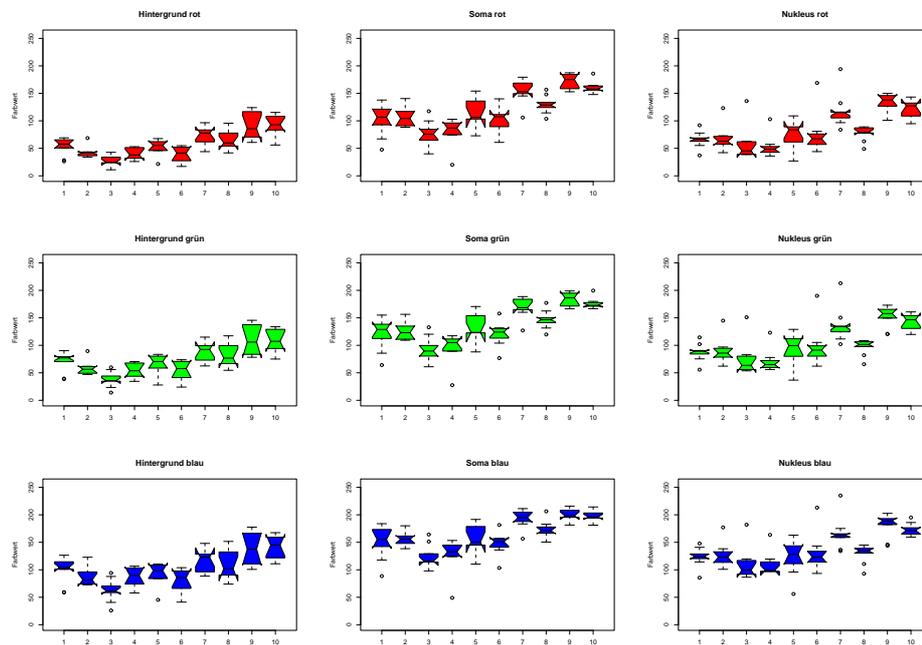


Abbildung 4.8: Ergebnis des Profil-Plots nach der Black World-Normalisierung

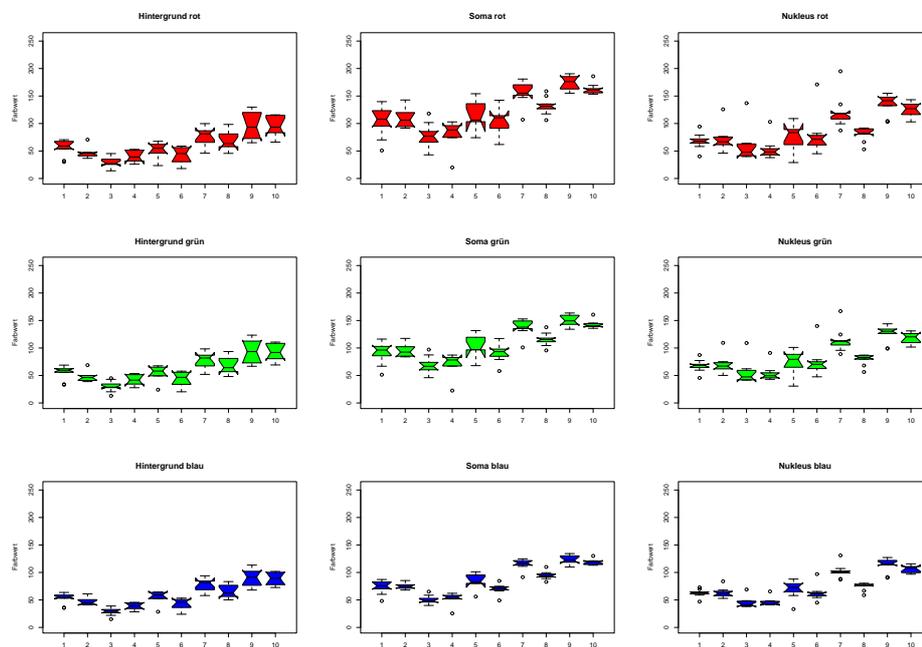


Abbildung 4.9: Ergebnis des Profil-Plots nach der Gray World-Normalisierung

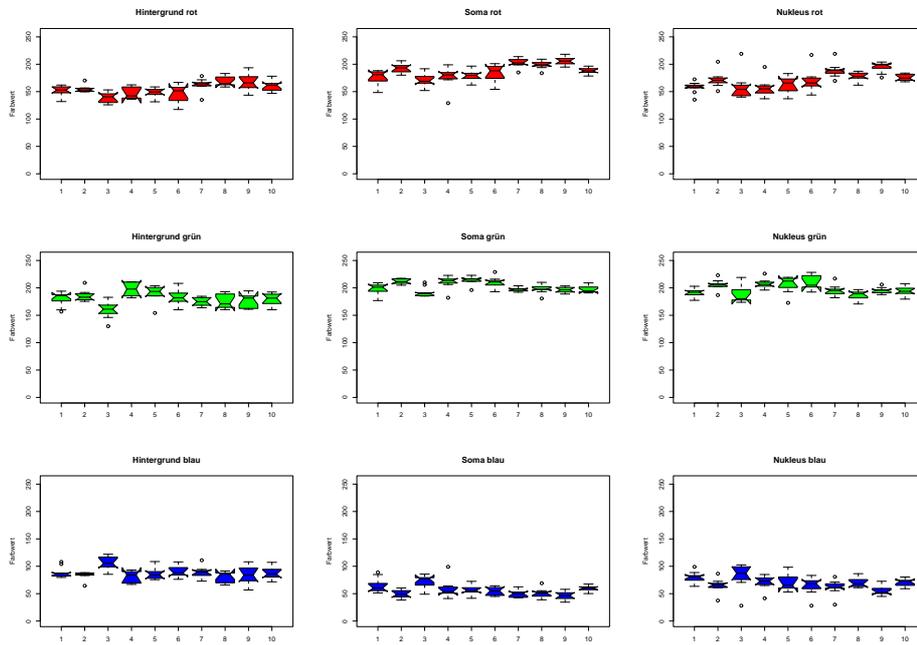


Abbildung 4.10: Ergebnis des Profil-Plots mittels RGB-Normalisieren

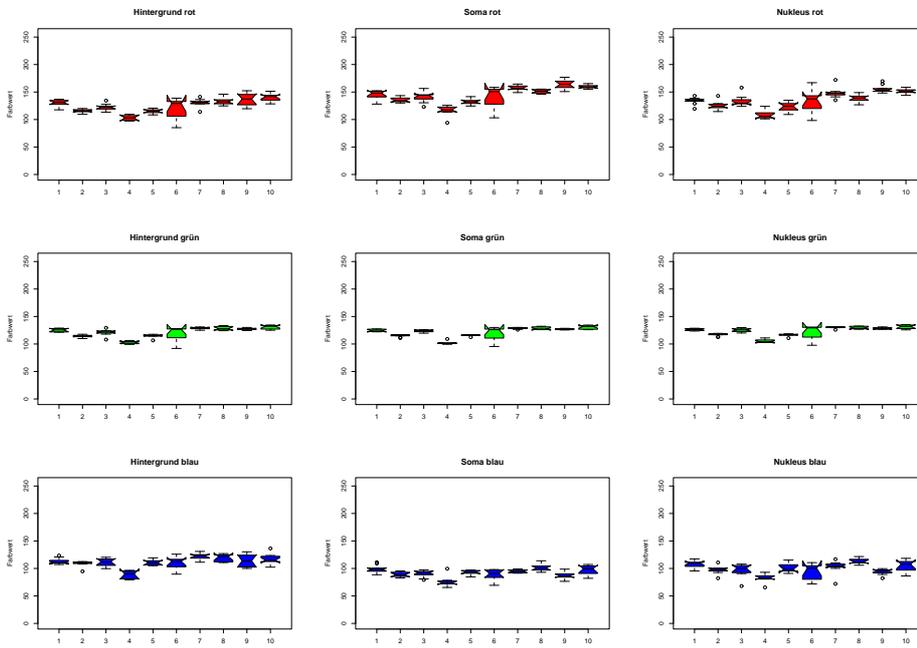


Abbildung 4.11: Ergebnis des Profil-Plots mittels kombinierter RGB-Normalisieren

4.2.2 Farbkorrektur bei histologischen Präparaten

Nachdem die von mir untersuchten „klassischen“ Korrektur-Algorithmen versagt haben oder nicht anwendbar sind, muss man sich zunächst die Frage stellen, worin sich die histologischen Präparate von gewöhnlichen Aufnahmen unterscheiden.

Die Situation bei der Farbkorrektur histologischer Präparate unterscheidet sich von der Ausgangssituation für die die oben beschriebenen Verfahren entwickelt wurden, grundlegend. Gleichung 4.1 auf Seite 61 lässt sich zunächst auf die Gegebenheiten der Durchlichtmikroskopie anpassen, indem der Anteil des reflektierten Lichts $S(\lambda)$, durch den Anteil des durchgelassenen, nicht absorbierten Lichtes $T(\lambda)$ ersetzt wird:

$$\rho_i = \int_{\lambda} R_i(\lambda)T(\lambda)E(\lambda)d\lambda \quad (4.13)$$

Diese Gleichung lässt sich in der Praxis weitestgehend vereinfachen, da die Beleuchtung durch das Mikroskop so präzise justiert werden kann, dass der Fehler unterhalb oder im Rahmen der Messgenauigkeit von einem Grauwertunterschied von eins liegt und somit $R_i(\lambda) = E(\lambda) = 1$ gesetzt werden kann.

Problematischer ist hingegen der Anteil des nicht absorbierten Lichtes, da diese Funktion sich wiederum in mehrere Faktoren aufspaltet. Die Menge des nicht absorbierten Lichtes ist abhängig von der Beschaffenheit der Zelle $C(\lambda)$ und den Rahmenbedingungen des Färbvorgangs $F(\lambda, p_1 \dots p_n)$:

$$T(\lambda) = C(\lambda)F(\lambda, p_1 \dots p_n) \quad (4.14)$$

Die physikalischen Rahmenbedingungen beim Färbvorgang bestimmen entscheidend den Grad der Absorption. Die Faktoren $p_1 \dots p_n$, die in $F(\lambda, p_1 \dots p_n)$ eingehen, sind:

- Färbedauer
- Farbstoffkonzentration
- pH-Wert
- Temperatur
- Auswaschung des Fixativs
- ...

Die Anzahl der Parameter ist groß und bei einer manuellen Färbung ist es nicht zu vermeiden, dass es im Verlauf einer Serie zu großen Abweichungen kommt.

Neben den hauptsächlichen Faktoren, die zu einer Färbung der Zelle führen, nämlich der Dichte ungesättigter Fettsäuren im Falle von Osmiumtetroxid und der Ribonukleoprotein-Dichte im Fall der Nissl-Färbung hängt die Intensität der Färbung von weiteren Faktoren ab, so dass es häufig zu großen Differenzen kommt.

Allerdings bieten die histologischen Schnitte gegenüber anderen Bildern nicht nur Nachteile, sondern auch einen großen Vorteil. Bei den meisten anderen Bildern variieren die Histogramme zwischen den Bildern sehr. So ist z.B. das Bild eines Sonnenuntergangs von Rottönen dominiert, während ein Bild vom Meer hauptsächlich blau ist.

Die histologischen Präparate hingegen weisen fast immer ein ähnliches Histogramm auf, da der hauptsächlichste Anteil des Histogramms durch den Hintergrund zu Stande kommt und dieser immer gleich gefärbt sein sollte.

Wenn das Histogramm zwischen zwei Bildern deutlich abweicht, so ist dies auf die oben genannte Funktion $F(\lambda)$ zurückzuführen, da die Anzahl und Dichte der Zellen meist nur gering gegenüber dem Hintergrund und im Rahmen einer akzeptablen Schwankungsbreite von einem Schnitt zum nächsten konstant ist.

Die Anpassung des Histogramms sollte robust und schnell berechenbar sein. Versuche, den Mittelwert der Histogramm anzugleichen, resultiert in einem den Black-/Gray World-Ansätzen ähnlichen Ergebnis.

Nach unterschiedlichen Ansätzen, den Mittelwert oder den Modalwert zu verwenden, habe ich ein Verfahren entwickelt, das den Median des Histogramms benutzt, um die Farbverfälschung durch den Färbeprozess zu korrigieren.

Das „Real Median Shift“-Verfahren

Voraussetzung für das von mir *Real Median Shift (RMS)* genannte Verfahren ist ein Bild, das als Vorlage dient. Prinzipiell kann dies jedes Bild sein. Damit jedoch bei der Korrektur keine Werte aus dem Definitionsbereich (0 bis 255) herauslaufen, wäre es sinnvoll, eine repräsentative Auswahl der zu untersuchenden Serie zu verwenden. Bei der relativ kleinen Serie von 50 Bildern wurde einfach die gesamte Serie als Referenz verwendet. Bei umfangreicheren Rekonstruktionen ist die Anzahl der Bilder der Schwankungsbreite des Farbraumes anzupassen. Es sollten jedoch zumindest 10% der Bilder als Referenz verwendet werden.

Dabei ist darauf zu achten, dass in keinem der Bilder größere Artefakte wie z.B. weite Kapillaren auftreten oder nur ein Teil des Bildes von dem Objekt bedeckt ist, da dies das Histogramm verzerren würde.

Ausgehend von diesem Histogramm $H_{i,k}, i = 0..255$ werden die Mediane M_k für die einzelnen Farbkanäle berechnet:

$$M_k = \begin{cases} H_{\frac{\sum_{i=0}^{255} H_{i,k+1}}{2}, k} & \text{falls } \sum_{i=0}^{255} H_{i,k} \text{ ungerade} \\ H_{\frac{\sum_{i=0}^{255} H_{i,k}}{2}, k} & \text{falls } \sum_{i=0}^{255} H_{i,k} \text{ gerade} \end{cases} \quad (4.15)$$

Entsprechend dieser Gleichung werden später auch für die zu korrigierenden Bilder die Mediane m_k für die einzelnen Kanäle berechnet. So ergibt sich der Korrekturwert g_k :

$$g_k = M_k - m_k \quad (4.16)$$

Die Werte des Ausgabebildes $o_{i,j,k}$ berechnen sich demnach durch eine einfache Addition:

$$o_{i,j,k} = p_{i,j,k} + g_k \quad (4.17)$$

Durch diese Berechnung wird das Histogramm sozusagen gegenüber der Farbachse verschoben. Diese Verschiebung reicht normalerweise aus, um das Bild zu korrigieren (siehe Abbildung 4.13 auf der gegenüberliegenden Seite). Dadurch, dass – wie in Abbildung 4.12 gezeigt – das Bild dadurch praktisch keine relevanten Daten verliert, ist dieser Ansatz außerdem relativ platzsparend, da die Korrektur direkt am Originalbild vorgenommen werden kann und keine Sicherungskopie notwendig ist.

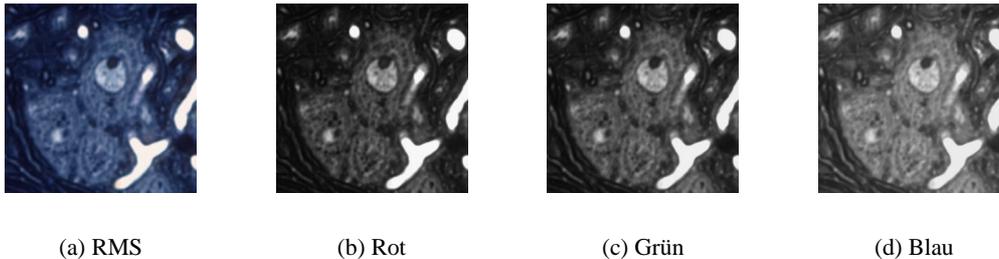


Abbildung 4.12: Das Beispielbild aus Abbildung 4.3 wurde mittels RMS normalisiert und hier sowohl als zusammengesetztes Bild als auch die separaten RGB-Kanäle dargestellt.

Das „Adaptive Median Shift“-Verfahren

Das im letzten Abschnitt vorgestellte Verfahren funktioniert in der Praxis gut und weist eine hohe Robustheit auf. Es hat jedoch in zweifacher Hinsicht ein Problem:

Zum einen ist es nicht in der Lage, ein Histogramm zu korrigieren, bei dem nicht nur die Position verschoben, sondern auch dessen Form beeinflusst ist. Dies kann entweder durch eine Abnormalität im Färbeprozess oder durch im Bild liegende Objekte wie z.B. eine sehr große Kapillare entstehen. Zum anderen widerspricht es der den bisher vorgestellten Verfahren zu Grunde liegenden Philosophie, dass der Algorithmus keinen Kontrollmechanismus aufweist.

Aus diesen Überlegungen heraus habe ich das RMS-Verfahren zum *Adaptive Median Shift (AMS)*-Verfahren erweitert.

Wie auch bei dem RMS-Algorithmus wird zunächst von einer Referenzmenge ein Histogramm gebildet und gespeichert. Wie auch beim RMS wird das Verfahren mit den vorgegebenen Bildern automatisch und ohne Eingriff des Benutzers trainiert. Soll nun ein Bild entsprechend dieser Vorgabe korrigiert werden, so werden in einem ersten Schritt die untypischen Bildteile entfernt.

In jedem Bild gibt es Bereiche, die mehr und solche, die weniger gut mit der Referenz übereinstimmen. Im Extremfall könnte z.B. der Rand des Objektes aufgenommen worden sein, was zu einer weißen Bildhälfte führen würde. Aber auch große Kapillaren, Risse im Präparat und ähnliche Artefakte können das Histogramm des Gesamtbildes beeinflussen. Durch diese Artefakte wird die Grundannahme des RMS-Algorithmus – dass die Histogramme einander entsprechen – verletzt.

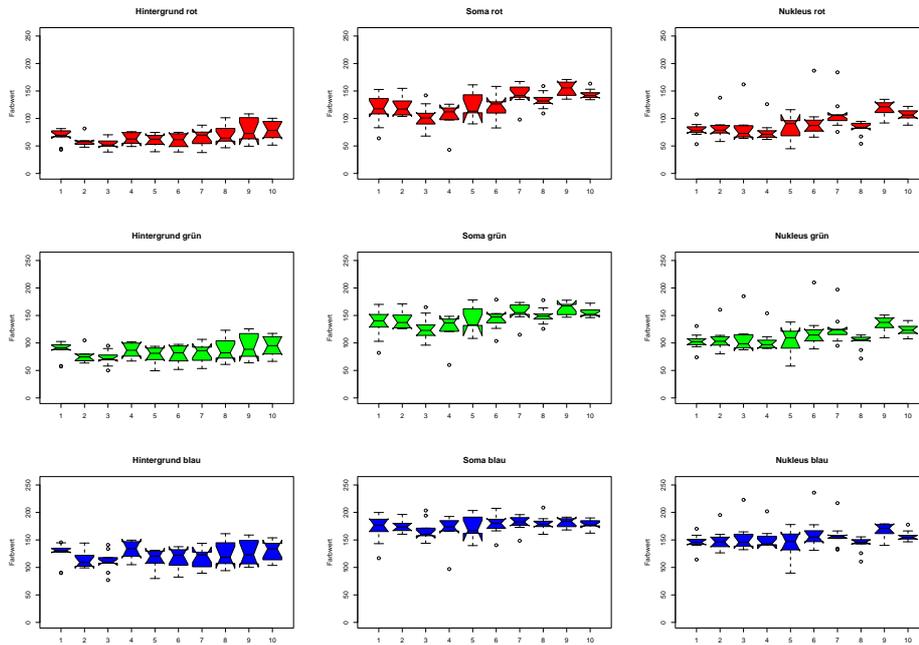


Abbildung 4.13: Ergebnis des Profil-Plots nach der RMS-Normalisierung

Deshalb wird das Bild in einem ersten Schritt in ein Raster unterteilt und für jedes einzelne Quadrat ein Histogramm gebildet. Das Histogramm wird mit der Referenz mittels der Methode der kleinsten Fehlerquadrate verglichen und das Bestimmtheitsmass R^2 dieser Korrelation berechnet.

Im zweiten Schritt wird dann eine Region of Interest (ROI) definiert, die diejenigen Quadrate erfasst, deren Bestimmtheitsmass in den oberen drei Vierteln der Verteilung liegt. In Abbildung 4.14 auf der nächsten Seite ist zu sehen, wie die dunklen Außenbereiche des Bildes aus der ROI entfernt wurden und somit bei der folgenden Histogrammbildung nicht berücksichtigt werden.

Von der ROI wird als nächstes ein Histogramm berechnet, das dann in c Teile $p_j, j = 0 \dots c$ mit gleichem Volumen (gleicher Anzahl) aufgeteilt wird.

Dazu wird die Summe der im Histogramm enthaltenen Pixel durch c geteilt. Von den beiden Farbwerten f_i und f_{i-1} , wobei gilt:

$$f_i \geq p_j \quad \wedge \quad f_{i-1} < p_j, \quad j = 0 \dots c \quad (4.18)$$

wird dann ein gewichteter Mittelwert \bar{f}_j gebildet:

$$\bar{f}_j = f_{i-1} + \frac{p_j - h(f_{i-1})}{h(f_i) - h(f_{i-1})} \quad (4.19)$$

Mit dem Referenzhistogramm wird ebenso verfahren. Dadurch ergeben sich dann Werte $f_j, j = 0 \dots c$ für die beiden Histogramme. Die entsprechenden Wertepaare werden dann dazu benutzt, eine glatt interpolierte Spline (siehe Seite 27 für eine

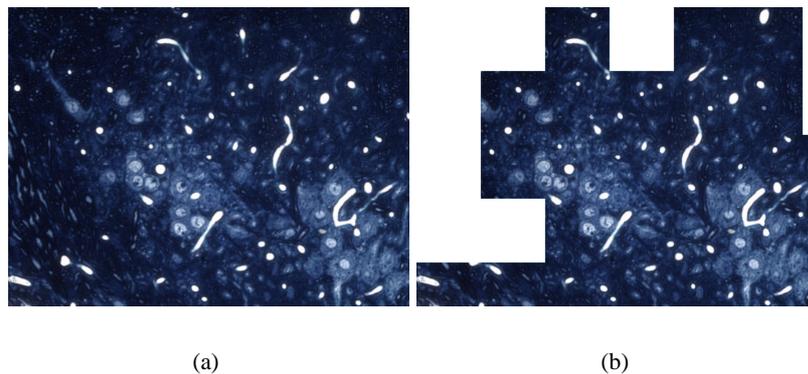


Abbildung 4.14: Das in (a) gezeigte Originalbild wurde in (b) durch die erste Stufe des AMS-Algorithmus bearbeitet. Es wurde eine ROI angelegt, die die atypischen Bildbereiche maskiert.

ausführliche Diskussion dieser Technik) hindurch zu legen. Zuvor werden die beiden Wertepaare $(255/255)$ und (l_1/l_2) dem Datensatz hinzugefügt. l_x entspricht dabei dem unteren Rand des jeweiligen Histogramms. Dies entspricht dem Vorabwissen, dass ein rein weißer Punkt immer weiß ist und sich die unteren Ränder der Histogramme ebenfalls entsprechen sollten.

Die sich aus dieser Berechnung ergebenden Werte können direkt zur Korrektur des Bildes verwendet werden.

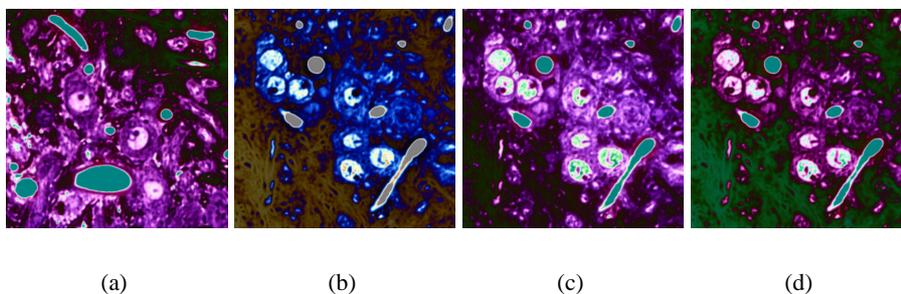


Abbildung 4.15: Da die Korrektur im normalen Bild praktisch nicht wahrnehmbar ist, wurden die hier gezeigten Bilder in einem Falschfarben-Raum transformiert. Die in (a) gezeigte Referenz unterscheidet sich zunächst deutlich von dem in (b) gezeigten Bildausschnitt. Sowohl die Korrektur mittels RMS (c) als auch das AMS-Verfahren korrigieren diese Abweichung zufriedenstellend.

Für die hier untersuchten Schnitte ist das Ergebnis mit den zuvor beim RMS-Verfahren gezeigten nahezu identisch (siehe Abbildung 4.15) und wie unten gezeigt, sogar geringfügig schlechter, was darauf zurückzuführen ist, dass durch die unterschiedlichen Korrekturfaktoren „Lücken“ im Histogramm auftreten können, die zu einer Reduktion der Kontrastinformation führen. Dies ist etwa der Fall, wenn der Farbwert 100 zu 105,55 und der Wert 101 zu 106,45 korrigiert wird. Beide Werte würden

zu 106 transformiert und der Wert 105 bliebe eventuell unbesetzt. Das Verfahren bietet jedoch zwei wesentliche Vorteile:

Zum einen ist es mit dem AMS-Verfahren auch möglich, stark modifizierte Histogramme mit einer Verzerrung zu korrigieren. So wäre es auch bei einer Veränderung der Beleuchtung oder anderen starken Einflüssen noch gut möglich, die Farbverschiebung zu korrigieren. Zum anderen wurde die Robustheit gegenüber dem RMS-Verfahren in zweifacher Hinsicht deutlich verbessert. So werden zum einen Artefakte in den Bildern nicht mehr berücksichtigt und zum anderen wurde durch das Verfahren gleichzeitig ein Qualitätsmarker eingeführt, der dazu verwendet werden kann, die Erfolgswahrscheinlichkeit der Farbkorrektur zu prognostizieren.

Das Verfahren zur Qualitätskontrolle, d.h. die Korrelation der einzelnen Bildausschnitte kann durchaus mit anderen Histogramm-basierten Algorithmen wie dem Gray World, Black World oder eben dem RMS verwendet werden, um das Zutreffen der von diesem Algorithmen gemachten Voraussetzungen zu überprüfen. In bewährter Art und Weise können die fehleranfälligen Schnitte somit aussortiert oder markiert werden.

Durch den ersten Schritt, das Maskieren der atypischen Bereiche, ist es nunmehr möglich, zu bestimmen, wie groß der Anteil des Bildes ist, der mit dem Referenzhistogramm übereinstimmt. Wenn ein relativ hoher Anteil der Fläche, etwa mehr als zwei Drittel ein Bestimmtheitsmass von unter 0,7 aufweist, sollte das Bild für eine spätere Kontrolle markiert werden, da es mit hoher Wahrscheinlichkeit eine sehr starke Farbabweichung aufweist.

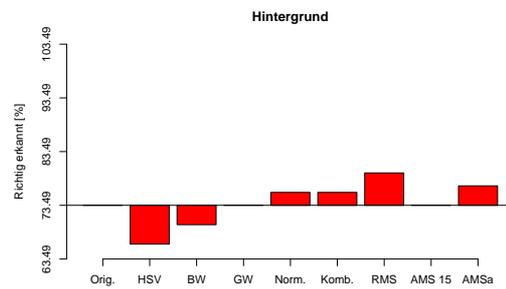
4.2.3 Ergebnisse der Farbkorrektur

Nachdem die Ergebnisse der Verfahren bisher optisch beurteilt wurden, soll im folgenden Abschnitt zusätzlich eine qualitative Beurteilung vorgenommen werden. Im Wesentlichen bestätigt diese die im vorherigen Abschnitt gemachten Beobachtungen.

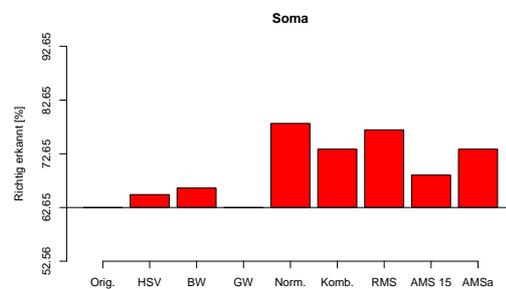
Um die Qualität der Farbkorrektur zu bewerten, wurde wie folgt vorgegangen: Über die Gesamtheit der Mitglieder der jeweiligen Klassen Hintergrund, Soma und Vordergrund wurde ein Mittelwert der RGB-Tupel berechnet. Dieser Mittelwert wurde verwendet, um alle Mitglieder mit einem Minimum-Distance-Verfahren [43] einer der drei Klassen zuzuordnen. Dabei wurde eine bestimmte Erkennungsrate erreicht, die bei den farbkorrigierten Bildern mit dem Original verglichen und in Abbildung 4.16 auf der folgenden Seite dargestellt wird.

Ich möchte an dieser Stelle ausdrücklich darauf hinweisen, dass diese Erkennungsrate ein rein arbiträres, relatives Maß für die Qualität der Farbkorrektur ist und in keinsten Weise mit der Erkennungsrate der gesamten Mustererkennung gleichzusetzen ist. Da der Gesamtprozess wie in den folgenden Abschnitten erklärt, kein einfaches Thresholding ist, sondern ein vielschichtiger Prozess, der mehrere Korrekturschritte beinhaltet, dem aber andererseits die Vorabinformation des Mittelwertes der Klassen nicht zur Verfügung steht, sind die Erkennungsraten des Gesamtalgorithmus vollkommen unabhängig von den hier berechneten Erkennungsraten.

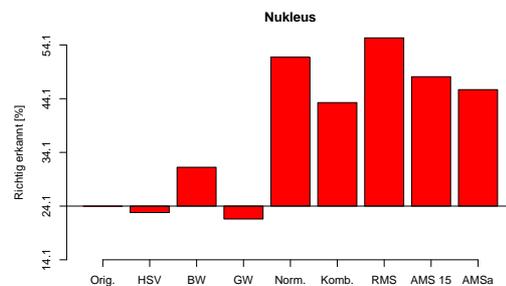
Bei der Betrachtung des Ergebnisses in Abbildung 4.16 fällt auf, dass einige der Verfahren die Erkennungsrate beim Hintergrund sogar unter den Originalwert drücken. Sowohl der Black World- als auch der HSI-Ansatz verschlechtern die Erkennungsra-



(a) Hintergrund



(b) Soma



(c) Nukleus

Abbildung 4.16: Vergleich der verschiedenen Verfahren zur Farbkorrektur: Wie im Text beschrieben, wurde ein Minimum-Distance-Clustering durchgeführt, dann der Wert für das unbearbeitet Bild als Standard genommen und die anderen Verfahren daran gemessen. Ganz links ist jeweils das Originalbild dargestellt, daneben die folgenden Verfahren: HSI-Korrektur (HSI), Black World (BW), Gray World (GW), Normalisiert (Norm.), das Kombinierte Verfahren (Komb.), sowie der RMS und der AMS, wobei letzterer einmal ohne (AMS15) und einmal mit (AMSa) vorheriger Adaptation der ROI angewendet wurde.

te des Hintergrunds. Die einfache Normierung des Farbraums entspricht in etwa dem kombinierten Ansatz. Das RMS-Verfahren schafft eine deutliche Verbesserung und ist etwas besser als das AMS-Verfahren. Bei letzterem kann man sehen, dass die Adaptation der ROI (AMSa) einen deutlichen Vorteil gegenüber dem Verfahren ohne eine solche Anpassung (AMS15) bringt.

Die nur geringe Verbesserung aller Verfahren gegenüber dem Original beruht darauf, dass der Hintergrund eine deutliche Distanz zu den beiden anderen Klassen aufweist und somit schon im Originalbild wesentlich besser zu isolieren ist. Da schon im Original fast drei Viertel aller Bereiche korrekt erkannt wurden, ist es – besonders in Anbetracht der immer vorhandenen Ausreißer – schwer, diese Rate noch weiter zu verbessern.

Bei den Erkennungsraten für die Somabereiche ist keiner der Algorithmen schlechter als das unbearbeitete Bild. Die Verfahren HSI, Black World und der nicht korrigierte AMS-Algorithmus stehen jedoch deutlich hinter den anderen Verfahren zurück. Die Verfahren des einfachen Normierens, das kombinierte Verfahren, sowie RMS und adaptierter AMS sind in etwa gleichwertig. Wenn beim AMS die ROI nicht angepasst wurde, schneidet der Algorithmus schlechter ab als mit einer Korrektur. Da in den ausgesuchten Bildern jedoch praktisch keine Artefakte aufgetreten sind, ist ein solcher Unterschied eher zufälliger Natur. Dies zeigt sich auch deutlich beim nächsten Bild, bei dem sich das Verhältnis umkehrt.

Die Auswertung der Ergebnisse für den Nukleus bestätigt die zuvor gemachten Beobachtungen: Die einfachen Verfahren Black World, Gray World und HSI-Korrektur sind in allen Bereichen deutlich schlechter als die restlichen Algorithmen. Das RMS-Verfahren kann insbesondere bei den beiden wichtigen Bereichen Nukleus und Hintergrund überzeugen (die Korrektur des Somas ist weniger wichtig, da dieses erst nachdem der Nukleus korrekt erkannt wurde, isoliert wird). Neben dem adaptierten AMS erzeugen auch die Verfahren der einfachen RGB-Normalisierung und das kombinierte Verfahren nach Finlayson gute Ergebnisse.

Wie bei den meisten Verfahrensschritten, gibt es auch für die Farbkorrektur kein „bestes“ Verfahren. Auch wenn bei der vorliegenden Schnittserie das RMS-Verfahren die besten Resultate erzielt hat, kann dies schon bei einer leicht veränderten Färbemethode deutlich anders aussehen. Wenn etwa größere Artefakte auftreten, wird das RMS-Verfahren dem adaptierten AMS und sehr wahrscheinlich auch einfachen Normalisierungsansätzen unterlegen sein. Umgekehrt wird eine Normalisierung des RGB-Farbraumes bei Bildern mit einem geringeren Kontrast als den hier gezeigten, deutlich schlechtere Ergebnisse hervorbringen und wenn weder Speicherplatz noch Rechenkapazität eine Beschränkung darstellen, ist auch das Verfahren von Finlayson eine denkbare Alternative.

Welches Verfahren in der Praxis Verwendung findet, muss der Anwender empirisch bestimmen. Für die allermeisten histologischen Präparate ist der RMS ein guter Ausgangspunkt. Sollte das Verfahren nicht funktionieren, so gilt es, die Alternativen zu testen. Dieser Test kann ähnlich wie bei den oben gezeigten Auswertungen durchgeführt werden, indem einige Zellen markiert und die Auswirkungen der verschiedenen Algorithmen beurteilt werden.

4.3 Maskierung der Kapillaren

Ein relativ einfacher, jedoch sehr effektiver Schritt besteht nun darin, die Kapillaren zu maskieren. Dies ist deshalb sinnvoll, da die Kapillaren leichter von den Zellkernen zu unterscheiden sind als die Zellkerne von den Kapillaren.

Diese Erklärung klingt zunächst paradox, ergibt jedoch bei näherer Überlegung durchaus einen Sinn: Beim Erkennen der Zellen und der Zellkerne kann kein fester Filter angewendet werden, vielmehr wird immer ein dynamisches Verfahren verwendet, bei dem die Grenzen der Zellkerne in Relation zu der Umgebung erkannt werden. Hier kann eine sehr helle Kapillare leicht mit einem Zellkern verwechselt werden.

Im Gegensatz dazu ist es bei der Maskierung der Kapillaren möglich, die störenden Bereiche zu einem großen Teil durch einen festen Filter zu maskieren.

Listing 5 Maskieren der Kapillaren

```
for all pixels p(i,j)
  delete = true
  for all bands k
    if p(i,j,k) < BORDER
      delete = false
  if delete
    mark(i,j) = true

for all segments s(i)
  if size(s(i)) < MINIMUM
    kill(s(i))

modify = true
while modify
  modify = false
  for all pixels p(i,j)
    if p(i,j) > SUBBORDER and mark(neighbor(i,j))
      mark(i,j) = true
      modify = true
```

Das Vorgehen dazu ist – im Vergleich zu den anderen Algorithmen – relativ einfach. Wie in Listing 5 zu sehen ist, besteht der erste Schritt darin, über alle Pixel eines Bildes zu iterieren und bei jedem Pixel $p(i, j)$ zu überprüfen, ob eines der Farbbänder k oberhalb eines definierten Wertes `BORDER` liegt. Der Grenzwert dieses Thresholdings sollte normalerweise relativ hoch, z.B. bei 253 liegen. Wenn einer der Kanäle unterhalb von `BORDER` liegt, wird dieser Punkt nicht markiert.

Nach diesem Durchgang werden alle so gebildeten Segmente $seg(i)$ darauf überprüft, ob die Anzahl der Pixel in diesem Segment, $size(s(i))$ oberhalb eines bestimmten Wertes, `MINIMUM` liegt. Wenn das Segment kleiner ist, wird es entfernt.

Dadurch wird verhindert, dass einzelne Punkte oder kleine weiße Flächen innerhalb eines Zellkerns diesen als Kapillare markieren.

In einem letzten Schritt werden alle Pixel darauf hin überprüft, ob sie oberhalb eines wesentlich niedriger angesetzten Wertes `SUBBORDER` liegen und einer der benachbarten Punkte bereits markiert ist. Auf diese Weise werden die Punkte, die am Rand der Kapillaren liegen und ebenfalls zu Irritationen führen können, eliminiert.

Das Ergebnis ist in Abbildung 4.17 auf der nächsten Seite zusehen. Die Kapillaren sind fast alle schwarz markiert. Nur einige kleine Kapillaren wurde nicht markiert – diese lassen sich jedoch hinterher relativ einfach von den größeren Zellkernen unterscheiden.

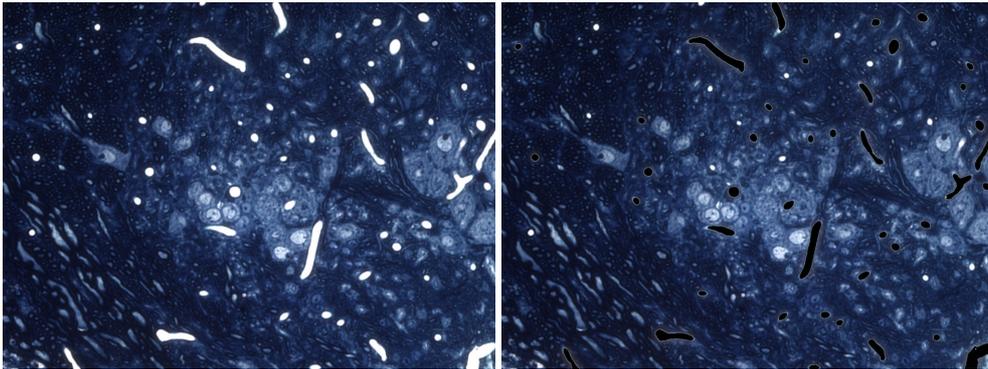


Abbildung 4.17: Die Kapillaren des linken Bilds wurden mittels des im Text beschriebenen Algorithmus maskiert. Im rechten Bildern sind alle Kapillaren, die zuvor weiß waren nun schwarz markiert und stören bei den folgenden Schritten nicht weiter.

Das Maskieren der Kapillaren ist ein wichtiger Schritt, um die Zellkerne hinterher einfacher zu extrahieren. Das Maskieren ist der erste von mehreren Filtern, mit denen sukzessive der nicht zu den Zellkernen gehörende Bereich entfernt wird. Die hier verwendete Strategie baut darauf auf, den Hintergrundbereich zu entfernen, was bei einem so komplizierten Fall wie dem Erkennen der Zellkerne sinnvoller ist als der Versuch, diese direkt zu erkennen. So ist es möglich, mehrere Verfahren einzusetzen, die gezielt den Hintergrund entfernen und sich dabei gegenseitig ergänzen. Dadurch ist es möglich, dass jedes der Verfahren einen relativ hohen Fehler aufweisen darf, diese sich allerdings gegenseitig kompensieren und so zu einer sehr guten Vorverarbeitung führen. Im nächsten Schritt wird nun der Hintergrund entfernt.

4.4 Vorfilterung der Zellen durch „Color Sensitive Edgedetection“

Für den nächsten Schritt, die Entwicklung eines Filters zur Diskriminierung von Hintergrund- und Zellkernbereich wurden verschiedene Verfahren getestet; die meisten erwiesen sich jedoch als zu fehleranfällig für den praktischen Einsatz.

Das Verfahren der Histogramm-Backprojection [43] erwies sich zunächst als das effektivste Verfahren. Der Histogramm-Backprojection-Algorithmus wird dazu verwendet, ein bekanntes Objekt in einem Bild zu detektieren. Es wird zunächst ein Histogramm aus dem Verhältnis der Farbhistogramme des gesuchten Objekts und des Bildes errechnet.

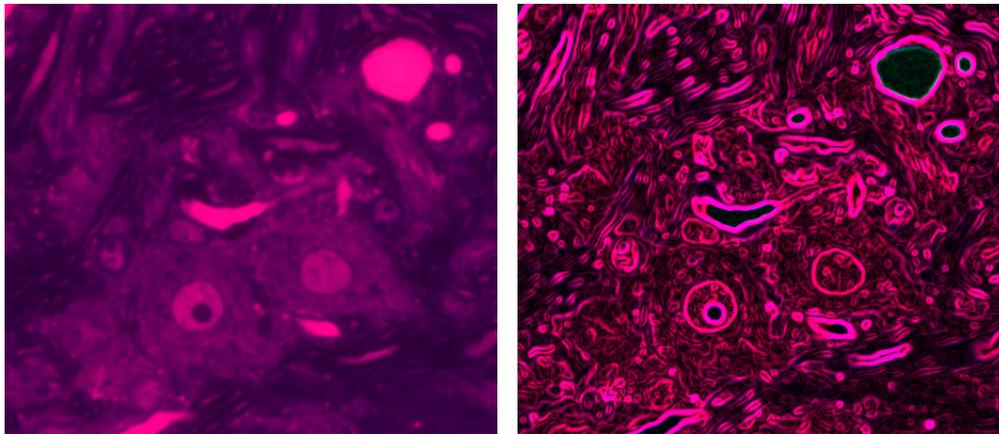
Anschließend findet eine Rückprojektion des so erzeugten Histogramms auf das Bild statt, in dem nach dem Objekt gesucht werden soll. Durch diesen Ansatz werden alle Farben im Bild, die nicht im gesuchten Objekt vorkommen, unterdrückt. Im Ergebnisbild sind nur noch die Objektfarben vorhanden.

Leider war aber auch nach der Farbkorrektur die Variationsbreite der Zellen noch so groß, dass das Verfahren in diesem Fall nicht anwendbar war. Sowohl Fehler erster Art (eine Zelle wird nicht als solche erkannt) als auch Fehler zweiter Art (ein Bereich wird als Zelle erkannt, obwohl dort keine Zelle liegt) traten gehäuft auf.

In diesem Fall sind die Fehler zweiter Art weitestgehend zu vernachlässigen, da – wie weiter oben beschrieben – noch weitere Filter folgen. Diese Fehler werden erst dann kritisch, wenn sie ein solches Ausmaß annehmen, dass regelmäßig weite Teile des Bildes als Zellen markiert werden.

Die Fehler erster Art hingegen sind sehr problematisch, da in einem solchen Fall die Zelle für alle weiteren Auswertungen verloren ist. Aus diesem Grund sind in dem im folgenden vorgestellten Algorithmus die Bedingungen auch eher darauf ausgerichtet, einen Fehler erster Art zu vermeiden.

Das Verfahren ist von seinem Prinzip her der klassischen Kantendetektion sehr ähnlich. Das Problem mit diesen traditionellen Verfahren ist, wie in Abbildung 4.18 am Beispiel des Sobel-Algorithmus zur Kantendetektion [43] zu sehen ist, dass zwar die Grenze zwischen Zellkern und Soma erkannt wird, das Bild insgesamt jedoch zu diffus ist, als dass diese Kante detektiert werden könnte.



(a) Original

(b) Sobel-Kantendetektion

Abbildung 4.18: Beispiel für eine Kantendetektion mit dem Sobel-Filter: Das Bild in (a) ist rechts mit dem Sobel-Filter behandelt. Für eine bessere Darstellung wurde der Farbraum gespreizt und das Bild invertiert.

Ähnliche Probleme bringt die Histogramm-Backprojection mit sich, da auch bei diesem Verfahren sowohl durch den Hintergrund, als auch durch die Kapillaren, die nicht herausgefiltert wurden, Artefakte in großer Anzahl auftraten.

Die von mir als *Color-Sensitive Edgedetection (CSE)* bezeichnete Methode kombiniert die Vorteile der beiden zuvor genannten Verfahren. Das Ziel der CSE ist es, mit einer hohen Toleranz diejenigen Bereiche des Bildes zu detektieren, in denen zwei Flächen mit einer spezifischen Farbe aufeinander stoßen. In diesem speziellen Fall besteht das Ziel also darin, die Grenzbereiche zwischen Zellkern und Soma zu detektieren.

Dabei bezieht CSE einen größeren Bereich in die Entscheidung ein, als dies z.B. bei der Sobel-Kantendetektion [43] der Fall ist. Die beiden Sobel-Kerne sind wie folgt aufgebaut:

$$H_x = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad \text{und} \quad H_y = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad (4.20)$$

Werden diese beiden Filterkerne auf das Bild I angewendet, ergeben sich die Bilder I_x und I_y :

$$I_x = I \cdot H_x \quad (4.21)$$

$$I_y = I \cdot H_y \quad (4.22)$$

Dazu wird jeder Pixel mit seinen acht Nachbarn betrachtet. Diese neun Grauwerte werden dann Element für Element mit dem Filterkern multipliziert. Die Summe der Produkte ergibt dann den Grauwert für den betrachteten Pixel im Zielbild. Dieser Vorgang wird mit allen Pixeln des Originalbildes durchgeführt. Der Zielgrauwert ist daher der Grauwert des betrachteten Pixels minus dem Grauwert des oberen Nachbarn. Die restlichen Nachbarn werden nicht bewertet. Das eigentliche Ausgabebild I_o ergibt sich wie folgt:

$$I_o = \sqrt{I_x^2 + I_y^2} \quad (4.23)$$

Die Sobel-Kantendetektion beruht darauf, dass einer der Kernel horizontale, der andere vertikale Kanten detektiert.

Um die Vorteile der Histogramm-Backprojection und der Sobel-Kantendetektion zu kombinieren, ist es notwendig, mit einer deutlich größeren Matrix zu arbeiten und an Stelle von Farbwerten mit einem Histogramm-Vergleich zu arbeiten. Auf diesen Prinzipien baut das von mir entwickelte Verfahren der *Color-Sensitive Edgedetection* (CSE) auf.

4.4.1 Color-Sensitive Edgedetection

Im wesentlichen erweitert die CSE die klassische Kantendetektion, indem nicht einzelne Pixel, sondern ganze Kompartimente betrachtet werden und die Analyse etwas komplexer ist, als dies bei der normalen Kantendetektion der Fall ist.

Um das Ziel, eine Detektion der Grenzflächen Soma-Nukleus zu ermöglichen, müssen zunächst die Charakteristika der beiden Flächen definiert werden. Dazu werden aus den beiden Bereichen Soma und Nukleus in einem Trainingsbild Histogramme gebildet und deren Mittelwert M_{Soma} und $M_{Nukleus}$ in Form eines Vektors (r,g,b) berechnet.

$$M = \begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} \frac{\sum_{i=0}^{255} i * H_r(i)}{N_r} \\ \frac{\sum_{i=0}^{255} i * H_g(i)}{N_g} \\ \frac{\sum_{i=0}^{255} i * H_b(i)}{N_b} \end{pmatrix} \quad (4.24)$$

Dabei ist $H_x(i)$ der i -te Wert im Histogramm des Farbkanals x und N_x die Gesamtanzahl der Pixel in einem Farbkanal; wobei normalerweise $N_r = N_g = N_b$ gelten sollte.

In Abhängigkeit von der Streuung um den Mittelwert wird ein Zurückweisungsradius R definiert. Dieser ist so zu wählen, dass möglichst alle Trainingsmuster in der Menge enthalten sind, es nicht zu einer Überschneidung der Mengen kommt und der Radius möglichst klein ist. Die richtige Größe ist in Abhängigkeit des gewünschten Fehlers erster und zweiter Art empirisch festzulegen. Je wichtiger es ist, dass es nicht dazu kommt, dass Bereiche falsch positiv erkannt werden, desto kleiner sollte R gewählt werden. Werte in der Größenordnung von fünfzig haben sich in der Praxis als geeignet erwiesen.

In einem nächsten Schritt folgt dann die eigentliche Kantendetektion. Dazu wird für jeden Punkt des Bildes eine Matrix von neun Quadraten definiert. Jedes Quadrat hat eine Seitenlänge von q Pixeln. Der Wert von q ist abhängig von der Bildauflösung und sollte im Allgemeinen klein gegenüber dem Zelldurchmesser gewählt sein. Eine Größe von 3 bis 20 Pixeln ist bei einer normalen Auflösung des Bilds sinnvoll.

Die Farbverteilung dieser Bereiche wird analysiert und der Mittelwert entsprechend Gleichung 4.24 gebildet. Die Mittelwerte M_{pos} werden über einen Minimum-Distance-Klassifikator mit den Vektoren von Soma und Nukleus verglichen. Dazu werden zunächst die Distanzen d_{Soma} und $d_{Nukleus}$ berechnet:

$$d_{Soma} = \sqrt{\sum_{k=0}^{bands} (M_{pos} - M_{Soma})^2} \quad (4.25)$$

$$d_{Nukleus} = \sqrt{\sum_{k=0}^{bands} (M_{pos} - M_{Nukleus})^2} \quad (4.26)$$

Die Variable k iteriert dabei über die Farbkanäle. Entsprechend seiner Distanz zu den beiden Klassen wird jedes Quadrat so einer der drei Klassen

- Nukleus
- Soma
- nicht definiert

zugewiesen. Dies geschieht über den sogenannten *Minimum-Distance-Klassifikator*:

$$class(d_{Soma}, d_{Nukleus}) = \begin{cases} nukleus & d_{Nukleus} < R \wedge d_{Nukleus} < d_{Soma} \\ soma & d_{Soma} < R \wedge d_{Soma} < d_{Nukleus} \\ undefined & sonst \end{cases} \quad (4.27)$$

Die Zuweisung des mittleren Quadrates ist unwesentlich und wird deshalb nicht berechnet. Nur wenn die Quadrate in einer der in Abb. 4.19 auf der nächsten Seite gezeigten Konfigurationen (dunkelblau = Soma, hellblau = Nukleus) vorliegen, wird dem Punkt ein Wert zugewiesen. Die dort gezeigten Konfigurationen umfassen alle

möglichen Anordnungen, bei denen sich Soma- und Nukleus-Kompartimente gegenüberliegen.

Liegen sich mehrfach Soma und Nukleus gegenüber, wird diejenige Kombination verwendet, bei der der Abstand der Vektoren zu den mittleren Vektoren der beiden Klassen möglichst gering ist. Der Pixel bekommt dann die Summe der Abweichungen der beiden Vektoren zugewiesen. Je geringer der Wert, desto qualitativ hochwertiger ist dieser Grenzbereich.

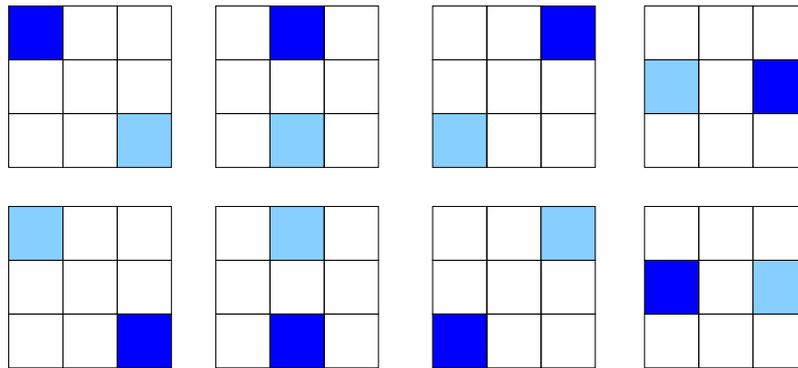


Abbildung 4.19: Mögliche Konfigurationen für eine positive Erkennung von Soma und Nukleus mittels CSE: Nur wenn sich die beiden Bereiche gegenüberliegen, wird die mittlere Zelle markiert.

Da aus diesen Werten im Anschluss eine binäre Maske erstellt werden muss, ist eine Nachbearbeitung des Ergebnisses erforderlich. Hierzu werden einige Standardmethoden der Bildverarbeitung [43] angewendet.

Wie in Abbildung 4.20 dargestellt, wird zunächst ein Gaussfilter angewendet, welcher das Bild weichzeichnet, auf diese Weise glättet und somit die Konturen schließt. In dem so entstandenen Bild werden die Lücken ausgefüllt und das Bild anschließend binarisiert.

Dieses Bild kann bereits als Maske auf das Original angewendet werden. Am Ende werden noch kleinere Artefakte durch einen Median-Filter eliminiert und das Bild dabei gleichzeitig geglättet.

Ein Problem des Verfahrens ist die relativ lange Laufzeit. Dadurch, dass über jeden Pixel sowie über die Umgebung eines jeden Pixels iteriert wird, ist der Algorithmus sehr langsam. Dieses Manko kann verbessert werden, indem man das Bild entsprechend einem Schachbrett in Felder unterteilt. Der Algorithmus iteriert dann nicht mehr über das gesamte Bild, sondern betrachtet jedes einzelne Feld des Schachbretts als Einheit, die in ihrer Gesamtheit analysiert wird.

Vergrößert man die Seitenlänge eines Feldes, so erhöht sich dadurch die Geschwindigkeit – was allerdings zu Kosten der Genauigkeit des Ergebnisses geht. Die Größe der Schachfelder sollte nach Möglichkeit nicht oberhalb der Größe des Quadrates, über das iteriert wird, liegen. Wie in Abbildung 4.21 dargestellt, ist die Größe des Schachbretts jedoch von der Größe der für die Klassifizierung verwendeten Quadrate unabhängig.

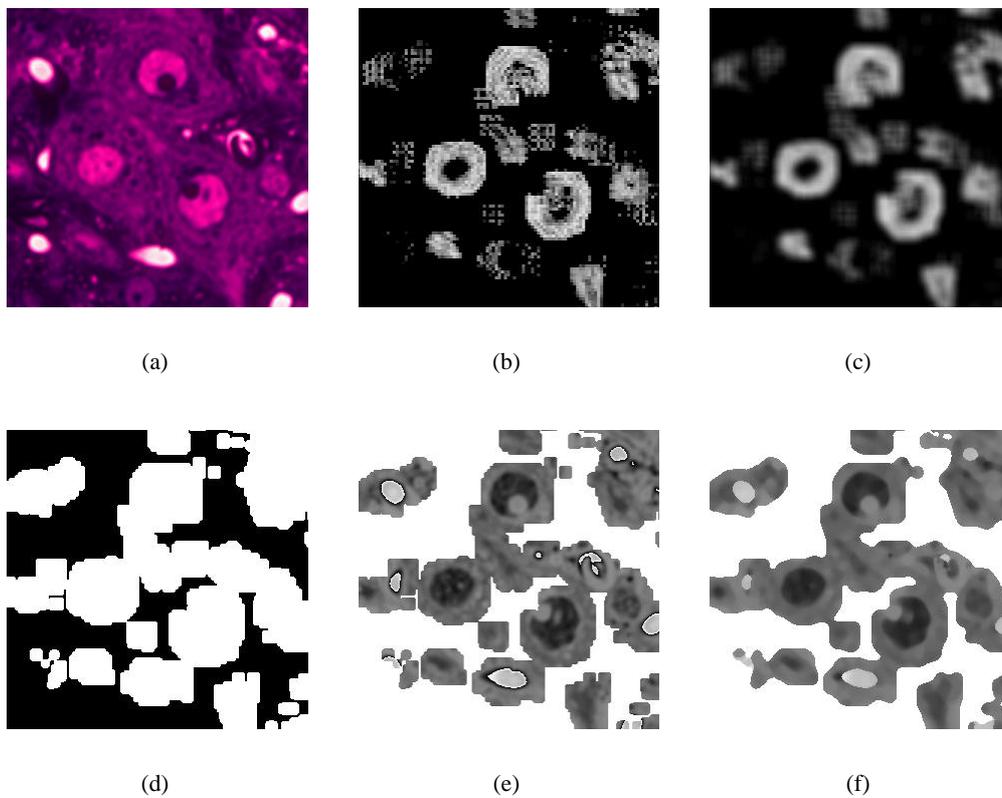


Abbildung 4.20: Die einzelnen Schritte des CSE-Algorithmus: Das Originalbild (a) wird zunächst wie im Text beschrieben mit dem CSE-Algorithmus bearbeitet (b). Um die Konturen zu schließen wird zunächst ein Gaußfilter angewendet (c). Das daraus resultierende Ergebnis wird binarisiert und die Löcher geschlossen (d). Das so erzeugte Bild wird auf das Original als Maske angewendet (e). Artefakte werden mittels eines Median-Filters (f) entfernt.

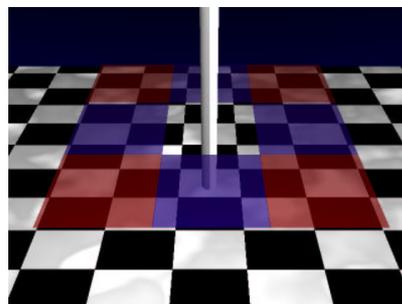


Abbildung 4.21: Schematische Darstellung des Schachbrettmusters mit darüber liegendem Bewertungsquadrat. Es wird deutlich, dass das Bewertungsquadrat in seiner Größe von der Schachbrettgröße unabhängig ist.

Zusammenfassend kann man festhalten, dass der CSE-Algorithmus den Hintergrund sehr sicher entfernt und die Grenzbereiche zwischen Soma und Nukleus sicher erkennt.

Zwar ist auch hier wieder eine große Fläche markiert, die teilweise nicht zu den Zellen gehörende Bereiche enthält doch ist es durch diese Vorverarbeitung nun möglich, die Zellkerne mit vergleichsweise einfachen Verfahren zu isolieren.

4.5 Lokale Kontrasterhöhung

Das hier beschriebene Verfahren der lokalen Kontrasterhöhung war bei der letztlich verwendeten Serie nicht notwendig, da dort der Kontrast durch die korrekte Justierung des Mikroskops (vergleiche Abschnitt 2.2.2 auf Seite 13) ausreichend gut war, um die Zellkerne ohne weitere Vorverarbeitung zu extrahieren. Will man das Verfahren jedoch mit einem anderen Mikroskop ohne die Möglichkeit zur Anpassung des Farbraumes durchführen, wird man auf eine Verbesserung des Kontrastes nicht verzichten können.

Bei der folgenden Binarisierung der Bilder ist es notwendig, dass die Zellen alle gleich hell bzw. gleich dunkel erscheinen. Dies ist leider nicht immer der Fall. Dieses Problem ist ähnlich dem der Farkorrektur, mit dem Unterschied, dass die unterschiedlichen Kontraste in einem Bild liegen. Würde man nun das gesamte Bild dunkler gestalten, so würden die bereits dunklen Stellen aus dem Wertebereich von 0 bis 255 herauslaufen und die Kontraste in diesen Bereich würden sich stark verringern.

Aus diesem Grund habe ich einen sogenannten adaptiven Algorithmus verwendet. Dieser als UNAHE (Uniform Non-Adaptive Histogram Equalization) bezeichnete Algorithmus [65] verfährt nach dem „Teile und Herrsche“-Prinzip, indem er jeweils in einem Bereich den Kontrast erhöht, diesen Bereich dann aufteilt und in den so entstandenen Bereichen den Algorithmus wiederholt.

Der dabei verwendete UNAHE-Algorithmus ist relativ einfach. Zunächst wird das Histogramm h_i ($i = 0 \dots 255$) gebildet. Darauf aufbauend wird die CMF (Color Matching Function) definiert:

$$cmf_i = \frac{\sum_{j=0}^i h_j}{\sum_{j=0}^{255} h_j} \quad (4.28)$$

Der neue Wert eines Punktes ergibt sich nun durch die UNAHE-Funktion wie folgt:

$$unahe(i) = 255cmf_i \quad (4.29)$$

In Abbildung 4.22 auf der nächsten Seite ist an einem Beispielbild die Wirkungsweise des UNAHE-Algorithmus demonstriert. Damit der Effekt besser deutlich wird, ist das Bild noch nicht mit dem CSE-Algorithmus bearbeitet. Es ist zunächst relativ dunkel, bei der ersten Aufteilung in Abbildung 4.22(b) sind die Zellen sehr hell und der Kontrast zwischen Zelle und deren Umgebung ist relativ gering.

Mit den beiden weiteren Aufteilungen wird der Kontrast innerhalb eines Quadrates immer besser und die Zellen treten in Abbildung 4.22(d) wesentlich deutlicher hervor, als dies im Ausgangsbild der Fall war.

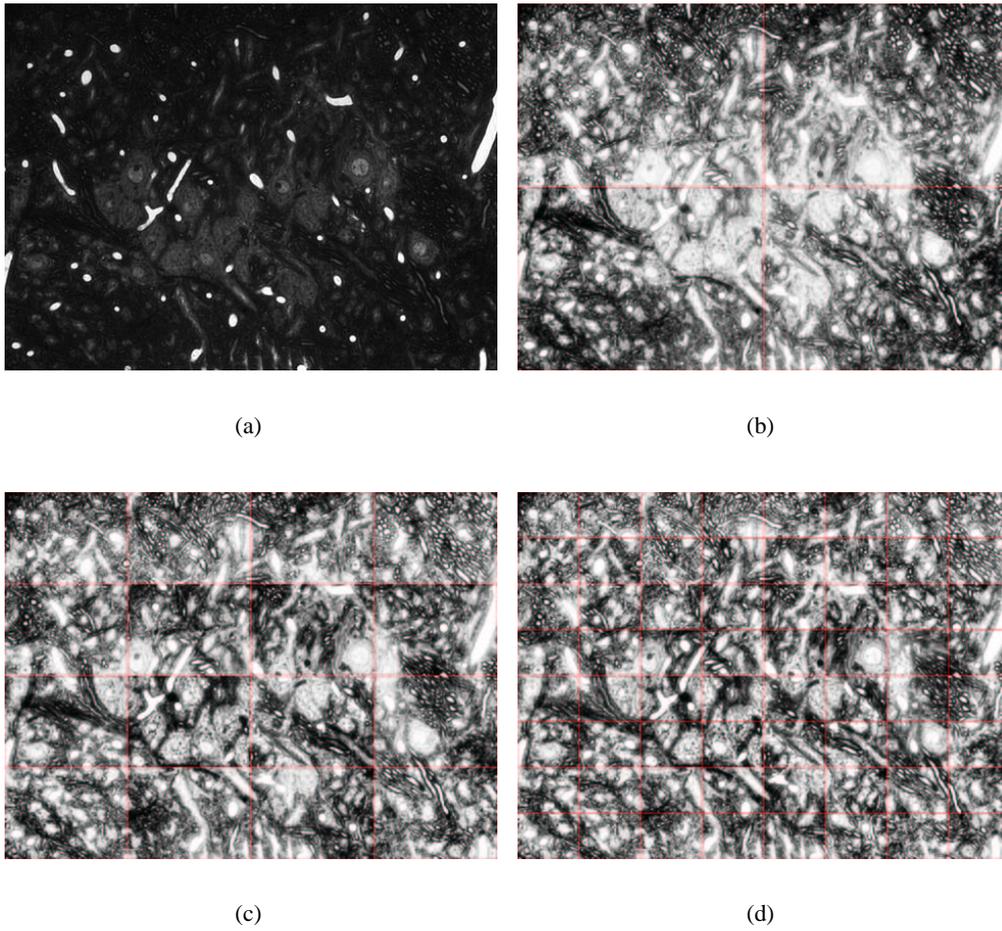


Abbildung 4.22: UNAHE-Kontrasterhöhung an einem Beispielbild

Einziges Manko dieses Verfahrens ist die Tatsache, dass die Kontraste eines Objekts in den Grenzbereichen (in der Abbildung rot gekennzeichnet) auf beiden Seiten unterschiedlich ausfallen können. Dies fällt bei der weiteren Verarbeitung kaum ins Gewicht, da die Kontraste in einem solch fein abgestuften Bereich eher unwichtig sind.

4.6 Lokalisieren der Zellkerne

In dem aus dem CSE hervorgegangen Bild sind die Zellkerne (eventuell nach einer lokalen Kontrasterhöhung) zu lokalisieren. Nach dem CSE sind die Zellbereiche absichtlich nur grob segmentiert, da so die Wahrscheinlichkeit, dass der oben angesprochene Fehler erster Art – das Nichterkennen von Zellen – auftritt, geringer wird.

In der vorliegenden Serie waren die Kontraste wie bereits erwähnt sehr gut, so dass ein einfaches Thresholding mit anschließender morphologischer Filterung ausreichend war. Bei einem Thresholding wird allen Farbwerten unterhalb einer Schwelle der Wert Null, allen darüber liegenden Werten der Wert 255 zugewiesen. So entsteht ein binarisiertes Bild.

Das schon im Abschnitt über das CSE-Verfahren gezeigte Bild mit drei Zellkernen ist in Abbildung 4.23 nochmals gezeigt. In der mittleren Abbildung wurden die mit dem CSE-Algorithmus extrahierten Zellkerne durch ein Thresholding mit einem Schwellenwert von 64 isoliert.

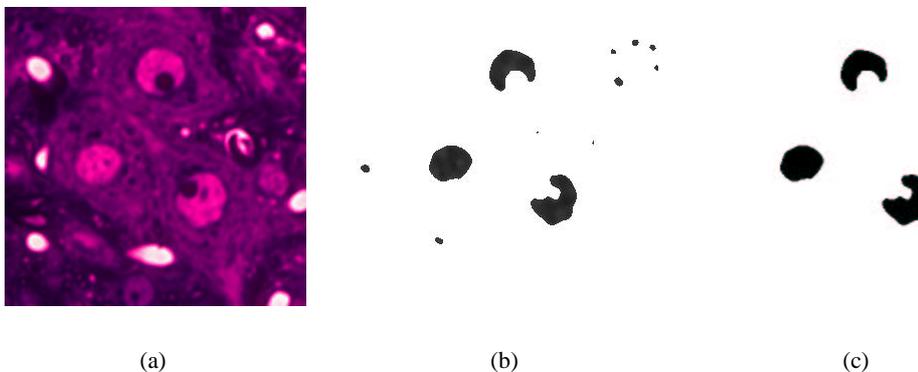


Abbildung 4.23: Das in (a) gezeigte Bild wurde nach der Verarbeitung mit dem CSE-Algorithmus einem Thresholding (b) unterzogen und die resultierenden Segmente wie im Text beschrieben an Hand morphologischer Kriterien gefiltert (c).

Das in Abbildung 4.23(c) zu sehende Bild ist durch einen zusätzlichen morphologischen Filter entstanden. Dabei können die Formen nach zwei Kriterien beurteilt und gefiltert:

- Größe
- Rundheit

Bei der Größe wurde schlicht eine minimale Größe in Pixeln empirisch festgelegt. Alle Segmente, die kleiner als 400 Pixel waren, wurden in der vorliegenden Serie entfernt. Wie man sehen kann, werden dadurch kleine Bereiche wie in Abbildung 4.23 entfernt.

Die Rundheit eines Segmentes – ebenfalls charakteristisch für einen Zellkern – ist ebenfalls relativ einfach zu bestimmen. Hierzu wird eine Achse zwischen den beiden Randpunkten, die am weitesten voneinander entfernt sind, konstruiert. Ausgehend von dieser Achse werden – vergleichbar der Windrose eines Kompasses – drei (oder mehr) Linien definiert. Die Länge dieser Linien wird ebenfalls gemessen und in Relation zu der maximalen Strecke gesetzt. Bei einem idealen Kreis wäre der Quotient dieser beiden Werte 1. Ist die Form hingegen langgestreckt, nähert sich der Wert Null.

Die Rundheit wurde nach mehreren Versuchen als Kriterium wieder verworfen, da auch die Zellkerne – bedingt durch den Nukleulus, welcher dunkler ist und deshalb aus der Form herausfällt – häufig eine nicht exakt runde Form aufweisen und fälschlicherweise entfernt wurden.

4.6.1 Validierung der Zellkern-Mustererkennung

Bei vielen Problemen ist die Validierung des dazugehörigen Computerprogramms relativ einfach. So kann etwa ein geübter Betrachter das Image Mosaicing manuell problemlos mit einer Genauigkeit von wenigen Pixeln ausführen. Nimmt man dann dieses Ergebnis als „Goldstandard“ (die Bezeichnung für eine absolut korrekte Lösung), ist es einfach zu bestimmen, in wievielen Fällen der Algorithmus versagt hat.

Bei dem Problem, einen Zellkern von seiner Umgebung zu trennen, ist eine Validierung hingegen deutlich schwerer, da kein eindeutiger Goldstandard existiert. Zwar ist es auf der Basis der dreidimensionalen Daten vergleichsweise einfach, die Zellkerne zu lokalisieren und deren Anzahl zu bestimmen; in der Ebene ist es hingegen eine Frage der Interpretation, wann ein angeschnittener Zellkern noch als solcher identifiziert wird.

Es ist demnach schwierig, festzulegen, ob ein nicht erkannter Zellkern auf einen Fehler des Programms zurückzuführen ist oder ob es sich um einen Fall handelt, bei dem sich auch zwei menschliche Experten nicht einig wären. Um dennoch einen Eindruck von der Fehlerrate des Programms zu erhalten, wurden in mehreren Bildern die Zellkerne von zwei menschlichen Experten und der Software markiert.

Diese drei Auswertungen wurden dann wechselseitig als Goldstandard definiert und die anderen zwei Verfahren damit verglichen. Die in Tabelle 4.1 und Abbildung 4.24 auf der nächsten Seite gezeigten Ergebnisse zeigen, dass die Software der Leistung der menschlichen Experten weitestgehend entspricht.

Nimmt man einen der menschlichen Experten als Goldstandard, so hat die Software in beiden Fällen sogar eine marginal geringere Fehlerquote als der zweite Experte. Das einzige Manko, welches ins Auge sticht, ist, dass die Software deutlich mehr Zellkerne erkennt, als die beiden menschlichen Experten. Dieser Fehler beruht darauf, dass die Software – wie weiter oben erklärt – so eingestellt ist, dass die Quote der falschnpositiven Zellkerne deutlich höher ist als die der nicht erkannten Zellkerne. Dieser Fehler ist jedoch weniger gravierend, als es zunächst scheint, da ein großer Anteil dieser

Standard	Experte 1	Experte 2	Software
Experte 1	-	17,81%	35,30%
Experte 2	6,65%	-	32,35%
Software	6,35%	13,74%	-

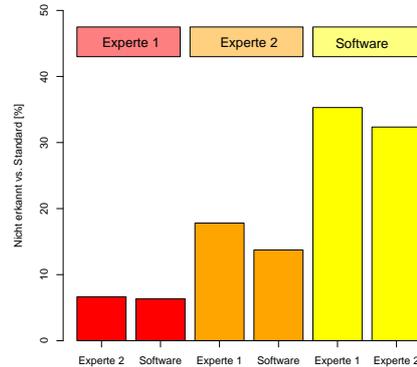


Tabelle 4.1: Vergleich des Ergebnisses der Software mit dem zweier menschlicher Experten: in den Spalten ist der jeweilige Goldstandard aufgetragen, gegen den das Ergebnis der jeweiligen Zeile validiert wird.

Abbildung 4.24: Vergleich des Ergebnisses der Software mit dem zweier menschlicher Experten

falsch erkannten Zellkerne später beim Prozess der dreidimensionalen Rekonstruktion eliminiert werden.

4.7 Fazit

In diesem Kapitel wurde zunächst ein auf die speziellen Bedürfnisse der Histologie abgestimmtes vollautomatisches Verfahren präsentiert, das es erlaubt, die bei der Färbung auftretenden farblichen Unterschiede weitestgehend zu eliminieren. Im Vergleich mit anderen Verfahren erzielen die von mir entwickelten Methoden eine gute Korrekturrate. Mittels des AMS-Verfahrens ist auch dieser Schritt automatisch zu kontrollieren.

Die Isolation der Zellkerne verläuft in mehreren Schritten. Zunächst werden die Kapillaren entfernt. Im zweiten Schritt können dann die Zellkerne durch den von mir entwickelten CSE-Algorithmus extrahiert werden.

Das automatische Verfahren erkennt die Zellkerne ähnlich präzise wie der Mensch. Auf dieser Verarbeitungsebene werden relativ viele Artefakte als Zellkern erkannt, was jedoch kein Problem darstellt, da dieser Fehler in einem späteren Verfahrensschritt sehr einfach korrigiert werden kann.

Ausrichten

Wie schon im Kapitel Mustererkennung angedeutet, besteht das große Problem des Ausrichtens darin, dass aus zwei Bildern drei Variablen abgeleitet werden müssen, nämlich die Translation in x - und y -Richtung sowie die Rotation des Bildes. Diese drei Freiheitsgrade zu bestimmen, ist eine komplexe Aufgabe.

In der Literatur finden sich drei prinzipiell unterschiedliche Vorgehensweisen zur Lösung dieses Problems:

Zunächst einmal gibt es einige Verfahren, bei denen das Problem der richtigen Ausrichtung durch die verwendete Technik gewährleistet ist. Dazu gehören neben der Computertomographie (CT) und den ihr verwandten bildgebenden Verfahren vor allem die konfokale Laserscanning Mikroskopie [26], die bei der Rekonstruktion einzelner Zellen häufig verwendet wird [46] und auch die Nuclear Magnetic Resonance (NMR) - Spektroskopie, die ebenfalls vereinzelt [2] für die Aufnahme von Zellen verwendet wurde.

Alle diese Verfahren sind sogenannte Scanning-Verfahren, die das Objekt in der z -Achse ebenso wie in der x -/ y -Ebene abtasten und somit bereits einen Voxelaum mit einem festen Bezugssystem entlang der axialen Ebene besitzen. Diese Verfahren kommen jedoch wegen der eingangs diskutierten Probleme in diesem Fall nicht in Frage.

Für das Ausrichten von Serienschnitten gibt es bisher einerseits rein manuelle Methoden und andererseits solche, die mittels des Computers und der manuellen Auswahl von Referenzpunkten die Schnitte ausrichten. Die im folgenden beschriebenen Methoden wurden zur Rekonstruktion von einzelnen Zellen verwendet; sie wären von ihrem Aufbau her jedoch prinzipiell auch zur Rekonstruktion eines Nukleus geeignet.

Beim Ausrichten mittels rein manueller oder nur sehr gering automatisierter Verfahren sind die Arbeiten von Pasko Rakic [78] und Cyrus Levinthal [63] erwähnenswert: Rakics Arbeit handelt von der Rekonstruktion von Neuronen im Gehirn des Rhesusaffen auf der Grundlage von EM-Aufnahmen. Dazu wurden die Zellen an Hand von Kapillaren und ähnlichen Hilfsstrukturen ausgerichtet und auf transparente Azetat-Folien durchgezeichnet. Mit einem grafischen Tablett wurden die Konturen nachgezogen und so in den Computer übertragen. Gegenüber der damals üblichen Rekonstruktion mittels Wachsplatten stellte das Verfahren einen großen Fortschritt dar.

Noch weiter verfeinert wurde die Technik des Ausrichtens von Levinthal, indem die Präparate in 1000 Ångstroem dicke Schnitte geschnitten, unter dem Elektronenmikroskop betrachtet und fotografiert wurden. Die Neuerung dieses Ansatzes bestand darin, dass man die Konturen der Zellen nicht mehr auf eine Folie übertrug, sondern mittels einer geschickten Anordnung von halbdurchlässigen Spiegeln und zweier Stroboskoplampen abwechselnd die beiden aufeinander folgenden Schnitte einblendete, so dass der Betrachter die Originale aneinander ausrichten und fotografieren konnte. So entstand ein „Film“ durch das Präparat. Bei einer weiteren Auswertung dieser Filme konnte der Betrachter am Computer mit der Maus sehr einfach den Verlauf der Neurone verfolgen, Verzweigungen kennzeichnen und somit eine Auswertung vornehmen, wie sie heute in Programmen wie z.B. NeuroLucida umgesetzt ist. So entstanden simplifizierte Graphen der Neurone, die Aussagen über deren Verzweigungsgrad, Länge der Äste und ähnliche Parameter erlaubten.

Der nächste große Fortschritt [92] bestand darin, den Computer nicht nur für das Nachfahren der Zellkonturen, sondern auch für das Ausrichten der bereits digitalisierten Schnitte zu verwenden. Diese Ansätze, wie z.B. [16], beruhen zumeist auf dem Setzen von Referenzmarken im Präparat, (etwa einer Rußmarkierung) an Hand derer die Software dann die Translation und die Rotation des Objektes berechnet.

Bei der Anwendung dieser Techniken [23] gibt es vor allem zwei Probleme: Zum einen ist der Bildausschnitt bei der Rekonstruktion auf Zellebene meist so klein, dass die Referenzmarken weit außerhalb des Bildes liegen, außerdem ist es bei vielen Methoden – wie auch bei der von uns verwendeten Einbettung in Kunstharz – nur sehr schwer möglich, künstliche Referenzpunkte zu setzen.

Zwar gibt es Ansätze, die Referenzpunkte im Gewebe mittels eines Hochgeschwindigkeitsbohrers [36] oder eines Lasers [93] zu setzen, allerdings sind auch diese Marker mit einer Größe von ca. 200 µm für die zelluläre Ebene immer noch zu groß und man läuft Gefahr, interessante Strukturen zu zerstören. Von den theoretischen Problemen einmal abgesehen, ist der Kosten- und Arbeitsaufwand für eine Umsetzung dieser Methoden immens, was in der Praxis ebenfalls ein großes Hindernis darstellt.

Andere Ansätze [6,28], bei denen der Anwender im Bild befindliche Referenzpunkte auswählt, vermeiden zwar das Problem der zu setzenden Referenzen; allerdings ist auch hier ein massiver Zeitaufwand seitens des Benutzers notwendig. Bei umfangreicheren Rekonstruktionen wächst der Zeitbedarf so sehr an, dass der Aufwand nicht mehr zu rechtfertigen ist.

Versucht man einen Algorithmus für das automatische Ausrichten zu finden, ist es ein naheliegender Gedanke, „einfach“ sämtliche Möglichkeiten auszuprobieren. Bei den ersten Versuchen zum Image Mosaicing haben ich genau dies getan und sämtliche Möglichkeiten der Translation ausprobiert. Diese Berechnung hat für ein Bildpaar ca. 24 Stunden in Anspruch genommen. Wollte man für jeden Punkt zusätzlich die Rotation auch nur gradgenau austesten, kämen für jeden Punkt nochmals 360 weitere Möglichkeiten hinzu. Die Laufzeit für das Ausrichten eines einzigen Schnittpaares würde dann sehr wahrscheinlich mehr als ein Jahr betragen. Somit kommt ein solcher „Brute-Force“-Ansatz nicht in Frage.

Bevor ich den von mir entwickelten Algorithmus zum Ausrichten näher beschreibe, will ich noch auf ein anderes Verfahren eingehen, das zunächst sehr vielversprechend aussah und von mir umfangreich getestet wurde.

Dieses Verfahren, das Alignments mittels Phasenkorrelation, ist direkt von dem weiter vorne beschriebenen Image Mosaicing abgeleitet. Da das in dem Verfahren verwendete Prinzip eine wichtige Grundlage für meine eigene Konzept-Entwicklung darstellt, werde ich es kurz beschreiben.

Danach folgt eine Beschreibung des von mir konzipierten Algorithmus, des Archimedes-Alignment-Algorithmus (A^3). In mehreren Abschnitten wird detailliert die Methode dieses Algorithmus beschrieben. Es wird gezeigt, dass bei dem A^3 – wie bei fast allen im Rahmen dieser Arbeit entwickelten Prozeduren – eine Fehlerkontrolle integriert ist und diese ein fehlerfreies Alignment ermöglicht.

Am Ende des Kapitels sind wiederum die Ergebnisse dieses Schrittes gegenüber zwei menschlichen Experten validiert.

5.1 Alignments auf Basis der Phasenkorrelation

Ein Verfahren, das in der Panoramafotografie eingesetzt wird, ist eine von Reddy entwickelte Erweiterung des Image-Mosaicings [81]. Zwar ist das Verfahren aus den weiter unten beschriebenen Gründen nicht auf meine Daten anwendbar, da es jedoch ein wesentliches Prinzip für meine eigenen Algorithmen enthält, möchte ich es hier kurz vorstellen.

Das Image Mosaicing mittels Phasenkorrelation [62] beruht auf der Fourier-Transformation zweier Bilder. Der normale Algorithmus stellt – vorausgesetzt, es handelt sich nur um eine Translation und nicht um eine Rotation oder Skalierung – ein sehr präzises und schnelles Verfahren für das Image Mosaicing dar (siehe auch Kapitel 3.1.1).

Die Erweiterung von Reddy erlaubt es nun, durch eine einfache Modifikation das Verfahren auch auf rotierte und skalierte Bilder anzuwenden.

Dazu wird der Phasenkorrelations-Algorithmus wie in [35,62] beschrieben durchgeführt; jedoch wird auf die Fourier-transformierten Bilder nochmals eine Transformation angewendet: Bevor der Phasenpeak festgestellt wird, findet eine Transformation in das polar-logarithmische Koordinatensystem statt.

Wie in Abbildung 5.1 auf der folgenden Seite gezeigt, entspricht eine Rotation im kartesischen Koordinatensystem einer Translation im Polarkoordinatensystem. In den beiden rechten Abbildungen entspricht ein Pixel in x-Richtung einem Grad und somit ist das unten links zu sehende Bild gegenüber dem darüber liegenden um 90° gedreht, was dazu führt, dass das transformierte Bild gegenüber dem Bild rechts oben um 90 Pixel verschoben ist.

Wenn man die zweite Achse zusätzlich noch logarithmiert, wird das Verfahren auch gegenüber einer Skalierung invariant (siehe [81]). Da eine Skalierung bei den Schnittserien nicht auftrat, werde ich darauf nicht weiter eingehen.

Es ist wichtig, dass zuerst eine Transformation in den Fourier-Raum stattfindet. Denn im Gegensatz zum Originalbild ist es im Fourier-Raum (in Abbildung 5.2 auf der nächsten Seite zu sehen) kein Problem, den Koordinatenursprung des Polarkoordinatensystems zu setzen. Im normalen Raum hingegen würde die Transformation, je

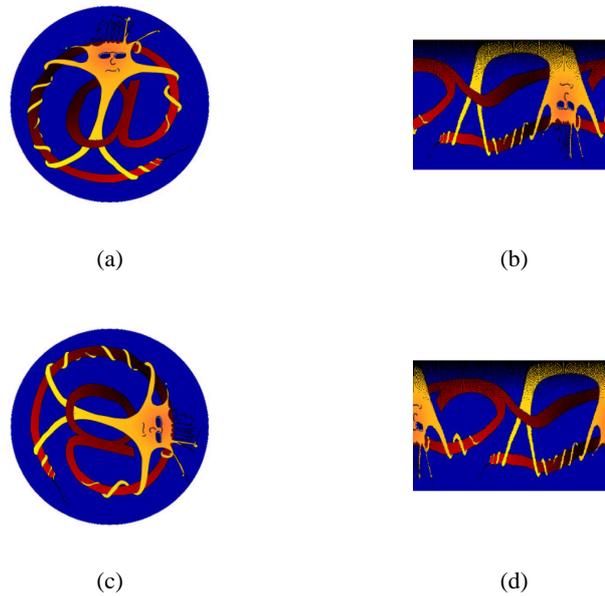


Abbildung 5.1: Beispiel für den Effekt einer Transformation in das Polar-Koordinatensystem: Nachdem die Bilder (a) und (c) jeweils in ein polares Koordinatensystem überführt wurden ((a) nach (b) und (c) nach (d)), kann man deutlich erkennen, dass bei der Wahl eines sich entsprechenden Mittelpunktes die Translation der beiden Bilder rechts der Rotation der beiden linken Bilder entspricht.

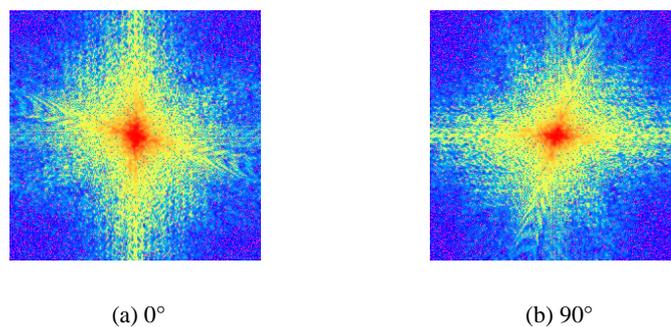


Abbildung 5.2: Transformation des Testbildes in den Fourier-Raum, links aufrecht, rechts um 90° gedreht.

nachdem wo der Ursprung des Koordinatensystems gesetzt wird, unterschiedlich ausfallen.

Leider konnte dieses Verfahren für meine Arbeit nicht angewendet werden, da die histologischen Präparate in Bezug auf Farbe, Form und Position der Strukturen sehr voneinander abweichen, was die enge Toleranzgrenze des Algorithmus überschreitet. Ich habe mehrere Tests mit unterschiedlich vorverarbeiteten Bildern durchgeführt; jedoch wurden diese nur in sehr wenigen Fällen korrekt ausgerichtet.

Dies liegt daran, dass beim Image Mosaicing, von leichten Abbildungsfehlern und einem sehr geringen Rauschen abgesehen, die aufeinander abzubildenden Einzelbilder gleich sind und durch exakte mathematische Transformationen in einander überführt werden. Bei dem hier vorliegenden Problem hingegen sind die Einzelbilder aufeinanderfolgende Schnitte. Diese sind zwar ähnlich, weichen aber doch in weiten Teilen voneinander ab. Diese Abweichung kann von dem Algorithmus nicht kompensiert werden.

5.2 Archimedes-Alignment-Algorithmus

Auch wenn der Algorithmus von Reddy sich nicht auf mein Problem anwenden lässt, so habe ich dennoch das darin enthaltene Prinzip der Transformation in Polarkoordinaten übernommen.

Den Algorithmus habe ich Archimedes-Alignment-Algorithmus (A^3) genannt – in Anlehnung an das berühmte Zitat „*Man gebe mir einen festen Punkt, und ich bewege die Erde!*“. Dieser Satz liefert im übertragenen Sinn eine sehr gute Erklärung für den Ansatz meines Algorithmus: Ziel des A^3 ist es, einen Punkt im ersten Bild zu finden, der einem Punkt im zweiten Bild zugeordnet werden kann. Hat man diese fixe Zuordnung getroffen, so kann man zwar noch nicht die Welt, wohl aber die beiden Bilder bewegen.

Wie weiter oben beschrieben, ist es nur möglich, aus der Translation zweier polartransformierter Bilder auf die Rotation zu schließen, wenn der Mittelpunkt der beiden polaren Koordinatensysteme identisch ist. Dadurch erscheint es zunächst so, als wäre das Verfahren nicht zu verwenden, da diese Punkte ja gerade gesucht werden sollen. Man kann diesen Zusammenhang aber auch ausnutzen, um die Identität zweier Punkte zu testen.

Die Zuordnung zweier Punkte kann dadurch auf ihre Korrektheit überprüft werden, indem das Umfeld in Polarkoordinaten transformiert wird, dann ein Vergleich bei mehreren „Translationen“ – sprich Rotationen um verschiedene Winkel stattfindet und überprüft wird, wie groß die geringste Abweichung ist. Wenn sich die beiden Punkte entsprechen, sollte die Abweichung dementsprechend gering sein.

Wollte man nun jedes Pixelpaar testen, so wäre die Laufzeit immer noch deutlich zu lang. Eine Alternative besteht darin, nur die Zentren der ja bereits isolierten Zellkerne zu verwenden und diese zu vergleichen.

Wenn man diese Tests mit den rohen Bilddaten ausführt, so kostete dies immer noch eine immense Zeit, da für n Segmente in einem Bild und m Segmente im anderen Bild insgesamt nm Transformationen in ein anderes Koordinatensystem durchgeführt

werden müssten und in ebenso vielen Fällen die richtige Translation gefunden werden müsste. Der dadurch bedingte Zeitaufwand wäre nicht zu vertreten.

Deshalb wird in einer ersten Stufe mit den aus den Bildern extrahierten Daten der Zellkerne gearbeitet. Hierbei macht sich der Algorithmus die Tatsache zu Nutze, dass die Zellkerne – wie viele biologischen Strukturen – näherungsweise rund sind, praktisch nie schräg verlaufen und in mehreren Ebenen sichtbar sind. Durch diese Eigenschaften eignen sie sich nicht nur optimal für das Ausrichten, sondern sie lassen sich außerdem als kreisförmige Segmente darstellen, die eine wesentlich schnellere Berechnung erlauben als dies bei Verwendung der Rohdaten der Fall wäre.

Wie auch bei fast allen anderen Verfahren wird auch für den A^3 eine hohe Fehlertoleranz des Algorithmus erzielt, indem die Ergebnisse überprüft und „gegenkorrigiert“ werden.

5.2.1 Methode

Prinzipiell ließe sich das Verfahren auch mit dem eigentlichen Bildmaterial durchführen; es ist jedoch – wie im letzten Abschnitt erwähnt – auf Grund der „computational costs“ effizienter, den Umfang der Daten zu reduzieren. Dazu wird in den Bildern I_ϕ , $\phi = 0 \dots 1$ jedes Segment, d.h. jeder erkannte Zellkern, als Kreis mit dem Radius $r_{\phi i}$ und dem Mittelpunkt $cx_{\phi i}/cy_{\phi i}$ dargestellt. Der Radius berechnet sich aus der Fläche $A_{\phi i}$ des entsprechenden Segments ($i = 0 \dots n_\phi$) nach

$$r_{\phi i} = \sqrt{\frac{A_{\phi i}}{\pi}} \quad (5.1)$$

Das Zentrum des Kreises entspricht dem Schwerpunkt der Fläche aus $N_{\phi i}$ Punkten mit den Koordinaten px_j/py_j , $j = 0 \dots N_{\phi i}$

$$cx_{\phi i} = \frac{\sum_{j=0}^{N_{\phi i}} px_j}{N_{\phi i}}, \quad cy_{\phi i} = \frac{\sum_{j=0}^{N_{\phi i}} py_j}{N_{\phi i}} \quad (5.2)$$

Das Ziel besteht nun darin, für jedes Segment i des ersten Bildes I_0 ein Segment j im zweiten Bild I_1 und einen Winkel α zu finden, so dass eine maximale Überlappung aller Segmente gegeben ist, wenn die Segmente in I_0 mit Segment i und jene in I_1 mit Segment j als Koordinatenursprung in ein Polarkoordinatensystem überführt werden und letzteres um den Winkel α verschoben wird.

Dazu werden zunächst die Zentren aller Segmente in I_0 in ein polares Koordinatensystem mit dem Zentrum von Segment i als Ursprung transformiert. Die Koordinaten des Zentrums des k ten Segments in I_0 lauten nun $\theta_{\phi i}(k)$, $R_{\phi i}(k)$.

Als nächstes werden dasjenige Segmentzentrum j aus I_1 und der Winkel α gesucht, für das sich eine maximale Überlappung ergibt, wenn das zweite Bild nach einer Transformation in das polare Koordinatensystem um α verschoben wird.

Dazu ist es notwendig, den Überlappingsgrad der Bilder bei gegebenen i , j und α zu definieren.

Der Überlappungsgrad definiert sich dadurch, dass für jedes Segment k des ersten Bildes dasjenige Segment l des zweiten Bildes gesucht wird, das die größte Überlappung aufweist.

Der Überlappungsgrad definiert sich als Summe der maximalen Überlappung der Segmente aus I_0 mit einem beliebigen Segment l aus I_1 :

$$\Delta\theta_{ij\alpha} = \frac{\sum_{k=0}^{n_0} \max \Delta_{ij\alpha}(k)}{n_0} \quad (5.3)$$

Der maximale Überlappungsgrad $\max \Delta_{ij\alpha}(k)$ des Segments k definiert sich als

$$\max \Delta_{ij\alpha}(k) = \max(\Delta_{ij\alpha}(k, l)), \quad l = 0 \dots n_1 \quad (5.4)$$

Die Überlappung $\Delta_{ij\alpha}(k, l)$ von zwei Segmenten k und l definiert sich wie folgt:

$$\delta_{ij\alpha}(k, l) = \sqrt{(\theta_{0i}(k) - ((\theta_{1j}(l) + \alpha)\%360))^2 + (R_{0i}(k) - R_{1j}(l))^2} \quad (5.5)$$

$$\Delta_{ij\alpha}(k, l) = \begin{cases} (r_0k + r_1l) & \delta(\alpha, k, l) = 0 \\ 0 & \delta(\alpha, k, l) > (r_0k + r_1l) \\ \frac{r_0k + r_1l}{\delta(\alpha, k, l)} & \text{sonst} \end{cases} \quad (5.6)$$

Das Zeichen % steht dabei für den Modulo-Operator. Es gilt nun denjenigen Winkel α und das Segment j zu finden, für die die Überlappung maximal wird.

Aus diesen Daten, dem Winkel α und der Segmentzuordnung $i \rightarrow j$ errechnet sich die Translation in x- und y-Richtung:

$$\Delta x = c_{x0i} - \cos(-\alpha)c_{x1j} + \sin(-\alpha)c_{y1j} \quad (5.7)$$

$$\Delta y = c_{y0i} - \sin(-\alpha)c_{x1j} - \cos(-\alpha)c_{y1j} \quad (5.8)$$

Mit den drei Parametern α , Δx und Δy wird nun eine Clusteranalyse nach dem Minimal-Distance-Verfahren durchgeführt. Dabei wird ein Zurückweisungsradius ρ definiert, der der gewünschten Genauigkeit entspricht. Das Ergebnis eines solchen Verfahrens ist in Tabelle 5.1 zu sehen und in Abbildung 5.3 grafisch dargestellt.

In fast allen Fällen werden die Parameter des größten Clusters dem richtigen Alignment der Schnitte entsprechen. In jedem Fall sollte das Ergebnis allerdings – wie auch bei allen anderen Verfahren – nochmals überprüft werden.

5.2.2 Test und Korrektur der Parameter

Da das Ziel dieser Methode darin besteht, ein vollständig automatisches Verfahren zu entwickeln, ist es wichtig, die Ergebnisse des soeben beschriebenen Verfahrens zu validieren.

Cluster	α	Δx	Δy
0	85.00	-38.66	1583.26
0	86.00	-27.31	1597.64
0	85.00	-39.64	1584.38
0	85.00	-36.80	1584.56
0	85.00	-37.84	1582.72
0	85.00	-37.91	1583.05
0	86.00	-26.52	1597.82
0	85.00	-39.43	1582.72
0	85.00	-39.13	1582.49
0	85.00	-39.97	1581.93
0	85.00	-36.62	1577.47
0	86.00	-26.72	1605.80
0	85.00	-38.04	1573.51
1	78.00	307.98	1416.94
1	78.00	307.75	1416.32
2	179.00	2598.50	1750.40
3	237.00	2123.58	481.86
4	118.00	544.60	1714.07
5	140.00	841.11	2003.57
6	190.00	1964.86	1304.72
7	102.00	201.87	1791.59

Tabelle 5.1: Die hier abgebildete Ausgabe entspricht einem typischen Cluster nach dem Minimum-Distance-Verfahren. Die Punkte der einzelnen Cluster sind in der Abbildung rechts in unterschiedlichen Farben dargestellt.

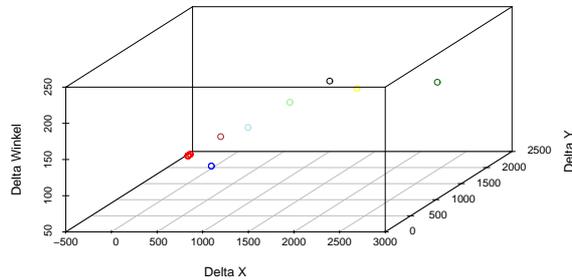


Abbildung 5.3: Dreidimensionale Darstellung des Ergebnis eines Minimum-Distance-Clusterings. Jede Farbe codiert ein weiteres Cluster. Das größte (und korrekte) Cluster ist rot markiert.

Dies kann vergleichsweise einfach geschehen, indem ein kreisförmiger Ausschnitt um die Zentren der einander zugeordneten Segmente i und j gewählt wird, dessen Radius in etwa dem einer Zelle entspricht. Diese Ausschnitte werden entweder mittels der von Reddy entwickelten Modifikation der Phasenkorrelation [81] oder einer Transformation in ein polares Koordinatensystem und anschließender Kreuzkorrelation verglichen. Bei der Phasenkorrelation ergibt sich der Rotationswinkel β , um den die beiden Bilder gegeneinander gedreht sind, nach dem Shift-Theorem [62] auf der Basis der Fourier-Transformation der Bilder. Bei der Kreuzkorrelation kann man den gleichen Parameter aus der optimalen Verschiebung entlang der Winkelachse bestimmen. (siehe Kapitel 3).

Das Verfahren ist in Abbildung 5.4 auf der gegenüberliegenden Seite dargestellt. Die dort in (a) abgebildete Zelle ist in das polare Koordinatensystem überführt. Die selbe Zelle in der nächsten Ebene ist in (d) zu sehen. Die Zelle, welche in (b) zu sehen ist, wurde hingegen willkürlich gewählt. In (c) ist die Zelle aus (d) um 86 Pixel (den optimalen Betrag) verschoben. Im Gegensatz zu einer willkürlich ausgewählten Zelle (b) und deren Differenzbild mit der Zelle in (a), welches in (e) zu sehen ist, weist das Differenzbild zwischen (a) und der richtig verschobenen Zelle (c) eine deutlich geringere Färbung auf (f). Dasjenige Segment j , das die minimale Differenz aufweist, definiert durch die Rotation bzw. Translation, bei der die minimale Differenz auftritt, den Winkel β .

Der mit diesem Verfahren bestimmte Winkel β sollte – im Rahmen der Messgenauigkeit – mit dem oben bestimmten Winkel α bei der Mehrheit (75% hat sich in der Praxis als sinnvoller Wert erwiesen) der Mitglieder des größten Clusters übereinstimmen.

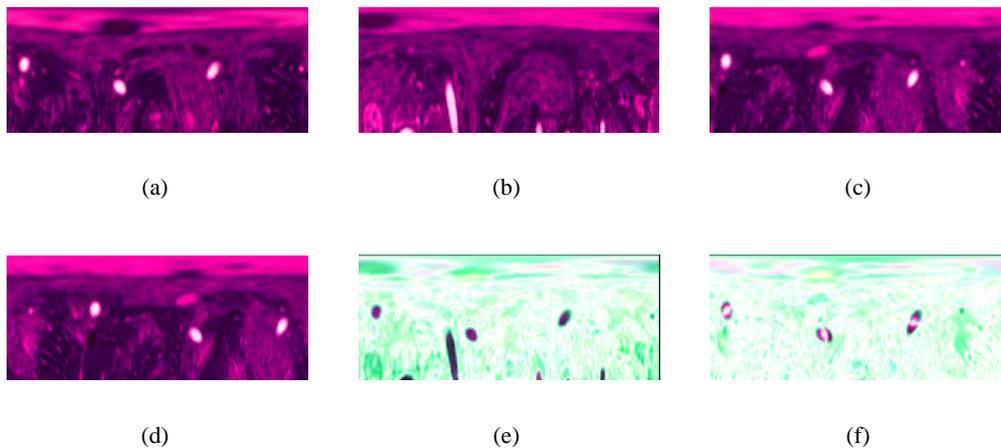


Abbildung 5.4: Validierung des Ausrichtens: In dieser Abbildung ist in (a) eine Zelle zu sehen, die in das polare Koordinaten System transformiert wurde. Die entsprechende Zelle in der nächsten Ebene ist in (d) zu sehen. In (c) ist die Zelle aus (d) um 86 Pixel (den optimalen Betrag) verschoben. Im Gegensatz zu einer willkürlich ausgewählten Zelle (b) und deren Differenzbild mit der Zelle in (a), welches in (e) zu sehen ist, weist das Differenzbild zwischen (a) und der richtig verschobenen Zelle (c) eine deutlich geringere Färbung auf (f).

Zum einen können so fehlerhafte Alignments markiert werden; es kann jedoch ferner auch eine automatische Korrektur stattfinden, indem in den Fällen, in denen keine Übereinstimmung erzielt werden kann, derjenige Cluster, der den höchsten Prozentsatz an Mitgliedern mit identischem α und β aufweist, verwendet wird.

So ist es in den meisten Fällen möglich, die Schnitte korrekt auszurichten. Sollte doch einmal ein Fehler auftreten, kann der entsprechende Schnitt automatisch für eine spätere manuelle Korrektur markiert werden.

5.3 Validierung der Ergebnisse

Das Verfahren wurde an den 50 in dieser Arbeit verwendeten Schnitten getestet.

Die Schnitte wurden vom Computer alle korrekt ausgerichtet. Mit Ausnahme von drei Alignments wurden die korrekten Transformationskoordinaten bereits auf Basis der Segmente allein, d.h. im ersten Teil des Verfahrens, gefunden. In drei Fällen entsprach der größte Cluster nicht dem korrekten Alignment; wobei in einem der drei Fälle dies darauf zurückzuführen ist, dass ein Schnitt fehlte und die Distanz somit $4\mu\text{m}$ an Stelle der normalerweise verwendeten $2\mu\text{m}$ betrug.

Alle Fehler – auch der auf dem fehlenden Schnitt beruhende – wurden jedoch bei der automatischen Validierung festgestellt und auf Basis der Bildausschnitte mittels Kreuzkorrelation in allen Fällen korrigiert. Nach diesem Verfahrensschritt war kein Paar mehr fehlerhaft ausgerichtet.

Um die Genauigkeit des Alignments zu beurteilen, wurde die Abbildung der Segmente aufeinander zunächst optisch kontrolliert und anschließend mit dem Ergebnis

eines manuellen Alignments zweier menschlicher Experten verglichen.

Bei der optischen Kontrolle wiesen alle Paare eine maximale Überlappung der Segmente auf. In dem in Abbildung 5.5 gezeigten Beispiel wurden zwei aufeinander folgende Ebenen mittels des A^3 aneinander ausgerichtet. Die sich überschneidenden Bereiche sind rot, die Bereiche, die nur von den Segmenten in Ebene eins oder zwei bedeckt sind, sind gelb bzw. blau markiert. Die Bildpaare sind so ausgerichtet, dass durch eine Veränderung der Transformationskoordinaten keine Verbesserung erreicht werden könnte.

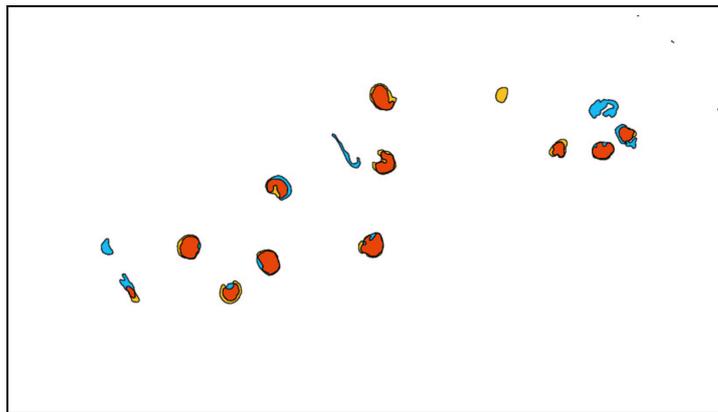


Abbildung 5.5: In dem hier gezeigten Beispiel wurden zwei aufeinander folgende Ebenen mittels der hier vorgestellten Software aneinander ausgerichtet. Die sich überschneidenden Bereiche sind rot, die Bereiche, die nur von Ebene 1 oder 2 bedeckt sind, sind gelb bzw. blau markiert.

In Ermangelung eines geeigneten „Goldstandards“ wurden auch bei diesem Verfahren die Ergebnisse zweier menschlicher Experten als Vergleich herangezogen.

Dabei wurden die Schnitte von beiden Experten mittels Referenzpunkten so gut wie möglich ausgerichtet und die daraus resultierenden Transformationskoordinaten mit den von der Software bestimmten verglichen.

Die Ergebnisse wurden verglichen, indem der absolute Unterschied des Transformationswinkels einerseits, sowie die euklidische Distanz der Translationskoordinaten andererseits berechnet wurden. Diese beiden Parameter wurden als Fehlermaß verwendet und einmal zwischen den beiden Experten und jeweils zwischen einem Experten und dem Computer verglichen.

Trägt man die Fehlerhäufigkeit als Histogramm auf, so ergibt sich weder für den Winkel (Abbildung 5.6(a)), noch für die Translationskoordinaten (Abbildung 5.6(b)) ein messbarer Unterschied zwischen der Übereinstimmung der Experten untereinander und jener gegenüber der Software. Dies bedeutet, dass die Genauigkeit der hier vorgestellten Softwarelösung mit der eines menschlichen Experten absolut vergleichbar ist.

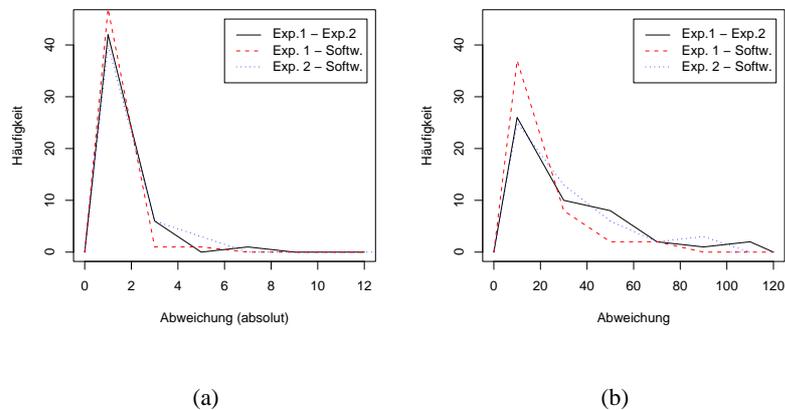


Abbildung 5.6: In (a) ist die absolute Häufigkeit vom Betrag der Winkelabweichung einmal zwischen zwei Experten und je einmal zwischen Experte und Software gezeigt. In (b) wurde die euklidische Distanz zwischen den bestimmten Translationsparametern als Fehlermaß verwendet. Deren Häufigkeit wurde wiederum für die Experten im Vergleich mit der Software und im Vergleich untereinander gezeigt.

5.4 Fazit

Das manuelle Ausrichten von histologischen Präparaten ist zeitaufwendig, ungenau und kostenintensiv. Die hier vorgestellte Methode erlaubt ein automatisches Ausrichten mit hohen Präzision in kurzer Zeit.

Wie durch den Test des Verfahrens an 50 Schnitten gezeigt werden konnte, arbeitet es unter Normalbedingungen praktisch fehlerfrei. Sollten doch einmal Fehler auftreten, werden diese zumeist korrigiert, in jedem Fall aber detektiert, so dass eine manuelle Korrektur später möglich ist.

Das hier vorgestellte Verfahren erlaubt ein Alignment mit einer ähnlichen Genauigkeit wie man sie durch ein manuelles Ausrichten der Schnitte erreichte. Da kein menschlicher Eingriff für das Ausrichten notwendig ist, können mit diesem Verfahren auch größere Gehirnbereiche einfach und schnell ausgerichtet werden.

Rekonstruktion

Bevor die Zellkerne rekonstruiert und dreidimensional dargestellt werden können, ist es zunächst notwendig, die extrahierten Segmente zu filtern und die fälschlicherweise erkannten Kapillaren und andere Artefakte auszusortieren. Wie bereits bei der Mustererkennung in Kapitel 4 erläutert, wurde beim Extrahieren der Zellkerne sehr darauf geachtet, keinen richtigen Zellkern zu verlieren; dafür wurde in Kauf genommen, dass einige Kapillaren und andere Strukturen fälschlicherweise als Zellkerne identifiziert wurden. Dieser Fehler lässt sich im dreidimensionalen Raum wesentlich leichter beheben, als dies im zweidimensionalen möglich gewesen wäre.

Im ersten Teil des Kapitels werden mögliche Verfahren zum Filtern der Segmente vorgestellt. Wie bereits erwähnt, bietet der dreidimensionale Raum mannigfaltige Möglichkeiten, die mutmaßlichen Zellkerne zu filtern. Prinzipiell ist die Anzahl der möglichen Filterkriterien nahezu unbegrenzt. In diesem Teil werden einige einfache Kriterien vorgestellt, die sich in der Praxis als sinnvoll erweisen haben.

Im zweiten Teil dieses Kapitels werde ich mit dem *CellInspector* eine grafische Oberfläche vorstellen, mit der es überhaupt erst möglich ist, die Ergebnisse der Filterung und der Rekonstruktion zu überprüfen. Das Programm eignet sich außerdem aber auch für die weiterführende Analyse und Auswertung der Präparate.

Vor der abschließenden Rekonstruktion erfolgt noch eine Betrachtung der Ergebnisse der mit dem *CellInspector* durchgeführten Validierung.

Der letzte Teil beschäftigt sich dann mit der Rekonstruktion der Zellkerne und deren Darstellung im dreidimensionalen Raum. Es gibt viele unterschiedliche Möglichkeiten, eine Rekonstruktion anzufertigen und darzustellen. Es ist stark von dem gewünschten Verwendungszweck abhängig, welche Form man im Einzelfall wählt.

Da der Sinn einer Rekonstruktion nie isoliert von ihrem Verwendungszweck gesehen werden kann, werden in dem letzten Teil dieses Kapitels nicht nur verschiedene Verfahren der Rekonstruktion, sondern auch verschiedene Möglichkeiten der späteren Auswertung bereits angesprochen. Eine ausführliche Diskussion dieser Auswertungsmethoden erfolgt allerdings erst in Kapitel 7.

6.1 Filtern der Segmente im dreidimensionalen Raum

Der erste Schritt für das Bewerten und Filtern von Segmenten im Raum besteht darin, diese zu Formen zusammen zu fassen. Dies kann nach verschiedenen Kriterien geschehen; im Allgemeinen sollte es jedoch ausreichen, lediglich alle Segmente, die in aufeinander folgenden Ebenen liegen und sich überlappen zu einer Form zu kombinieren.

Sollten die Schnitte relativ ungenau ausgerichtet sein oder eine geringe Auflösung aufweisen, wäre es notwendig, dieses Kriterium zu lockern und z.B. alle Segmente zusammenzufassen, deren Konturen sich überlappen würden, wenn das Segment entsprechend der Abweichung des Alignment-Algorithmus verschoben wäre.

Umgekehrt wäre es natürlich ebensogut möglich, einen Wert zu definieren, um den sich zwei Segmente mindestens überlappen müssen, damit diese als eine Form definiert werden. Es ist jedoch im Allgemeinen sinnvoller, an dieser Stelle nicht pauschal zu filtern, sondern eher an einem späteren Zeitpunkt eine Auswahl zu treffen, da es dann auch möglich ist, andere Faktoren gegen die Anzahl der Segmente und deren Überlappungsgrad aufzurechnen.

Die durch die Kombination von Segmenten entstandenen Formen werden im Anschluss weiter gefiltert. Dabei stellt sich zunächst die Frage, ob man die gesamte Form filtert oder sich auf die einzelnen Ebenen bezieht. Auch wenn es zunächst attraktiver erscheint, die gesamte dreidimensionale Form zu betrachten, so ist es meiner Meinung nach besser, die zu der Form gehörigen Segmente einzeln oder ggf. in Kombination zu betrachten.

Bei einer Betrachtung der gesamten 3D-Form machen sich die unterschiedlichen Auflösungen in axialer und lateraler Richtung sehr stark bemerkbar. Wollte man etwa in axialer Richtung die „Rundheit“ einer Form bestimmen, so wird dies in den seltensten Fällen gelingen. Dies hat mehrere Gründe: Zunächst einmal werden die äußersten Bereiche des Nukleus in axialer Richtung zumeist auf Grund ihrer geringen Größe nicht mehr erkannt. Dadurch ist der Abschluss der Zellkern eine Ebene weiter im Innenbereich, was zu einem kantigen Umriss führt. Zum anderen führt der Auflösungsunterschied in lateraler Richtung immer zu einem treppenartigen Profil. Diese beiden Faktoren führen leicht zu Fehlinterpretationen und dem Herausfiltern von eigentlich richtig erkannten Zellkerne. Deshalb ist es im Allgemeinen sinnvoller, die einzelnen Segmente einer Form zu betrachten, als diese als Ganzes zu analysieren.

Im Wesentlichen lassen sich drei Charakteristika eines Zellkerns verwenden, um ihn von Kapillaren und anderen Artefakten zu unterscheiden:

- Er hat eine bestimmte Größe.
- Er ist näherungsweise kugelförmig.
- Er weist eine charakteristische Färbung auf.

Das erste Kriterium lässt sich auf unterschiedliche Art und Weise definieren. So kann man das Volumen des Zellkerns bestimmen, indem man die Voxel zählt und ein bestimmtes Mindestvolumen festlegt. Man kann aber auch eine minimale Ausdehnung in lateraler Richtung festlegen oder definieren, dass zumindest ein Segment eine gewisse Mindestgröße aufweist.

Um die Kugelform zu bestimmen, ist es aus den weiter oben erwähnten Gründen sinnvoll, dies auf der Basis der einzelnen Segmente zu tun. Eine Möglichkeit, die im Ergebnisteil (Abschnitt 6.3) noch näher beschrieben wird, beruht darauf, dass man eine Art „Windrose“ in jedes Segment legt, wobei „Norden“ bzw. Null Grad der Achse durch die beiden am weitesten voneinander entfernten Punkte entspricht. Mittels der acht (oder mehr) Strecken vom Zentrum zum Rand bei unterschiedlichen Winkeln lässt sich die Rundheit definieren.

Denkbare Kriterien wären neben dem Mittelwert und der Standardabweichung, die ebenfalls in Abschnitt 6.3 verwendet werden, natürlich auch der Median und entsprechend die Interquartilsdistanzen als Maß für die Streuung oder aber man betrachtet einfach die minimale und die maximale Entfernung eines Punktes vom Mittelpunkt, wie dies ebenfalls im übernächsten Kapitel gezeigt wird.

Eine andere Möglichkeit ist es, alle zur Kontur gehörigen Punkte zu bestimmen und diese in Relation zu dem Umfang eines Kreises mit identischem Volumen zu setzen. Je weiter der entsprechende Wert von 1 abweicht, desto weniger entspricht die Form einem Kreis.

Das letzte Kriterium, die spezifische Färbung ist durch den Prozess der Mustererkennung bereits sehr gut ausgewertet, so dass es an dieser Stelle wenig Sinn macht, nochmals nach dem Farbwert zu filtern.

In dem weiter unten beschriebenen Beispiel werden starre Filter verwendet, um den Prozess des Filterns einfacher und übersichtlicher zu gestalten; bei schwierigen Fällen sollte man jedoch in jedem Fall versuchen, diese durch ein auf Fuzzy-Logik basierendes System zu ersetzen.

6.2 CellInspector

Nach dem Filtern der Segmente und der im folgenden Kapitel beschriebenen Rekonstruktion erhält der Benutzer eine Anzahl von Formen. Bei einigen dieser Formen ist es möglich, auf Grund der Größe oder anderen Kriterien zu beurteilen, ob es sich dabei um ein Artefakt oder einen tatsächlichen Zellkern handelt. Bei den meisten Formen ist dies allein an Hand der dreidimensionalen Form jedoch nicht möglich. Für den menschlichen Betrachter ist die Entscheidung, ob es sich bei der Form um ein Artefakt oder einen Zellkern handelt, ohne das Originalmaterial nicht zu treffen.

Doch auch mit dem Originalmaterial gestaltet sich eine Verifizierung sehr schwierig, da auf Grund der natürlichen Beschränkung eines Monitors auf zwei Dimensionen es nur möglich ist, die einzelnen Schnitte nacheinander zu betrachten und in der Bildbearbeitung den jeweiligen Ausschnitt zu markieren. Diese Vorgehensweise ist mühsam und selbst für die geringe Anzahl der hier zu untersuchenden Schnitte nicht durchführbar.

Aus diesem Grund habe ich das Programm *CellInspector* entwickelt, das durch eine grafische Oberfläche eine einfache und komfortable Untersuchung der Zellen im dreidimensionalen Raum gestattet.

Die grafische Oberfläche ist in Abbildung 6.1 auf der nächsten Seite gezeigt. Die Oberfläche ist in vier Bereiche aufgeteilt. Am oberen Bildrand befinden sich die Navi-

gationsleiste; rechts ist der aktuelle Schnitt in der Übersicht zu betrachten, am linken Rand befindet sich der aktuelle Ausschnitt in diesem und den vorhergehenden, sowie den folgenden vier Bildern. Am unteren Rand ist schließlich die aktuelle Form mit den dazugehörigen Segmenten abgebildet.

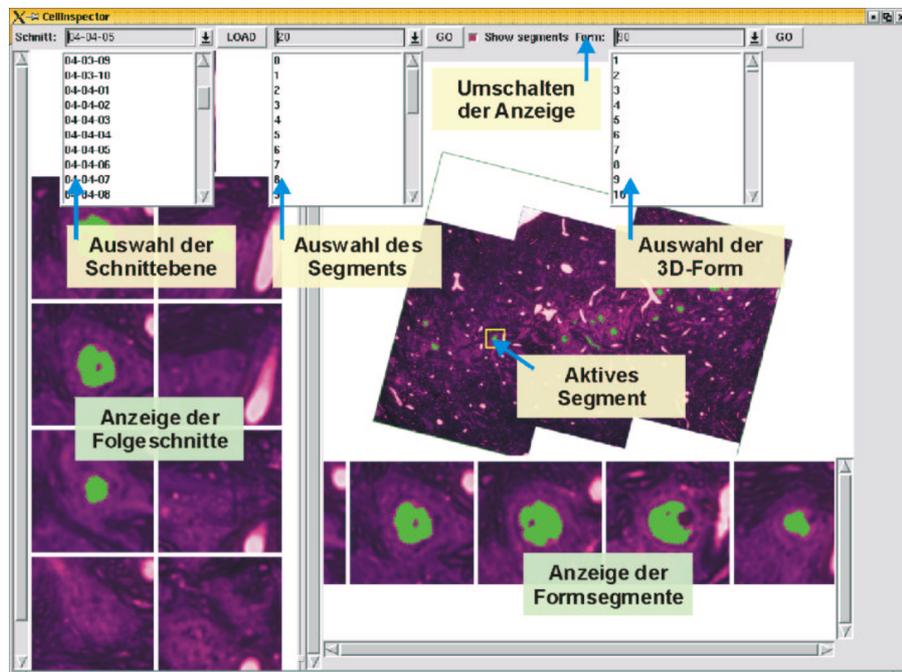


Abbildung 6.1: Die grafische Oberfläche des *CellInspector* mit Erläuterungen zu deren Bedienung

Die Bedienung des Programms ist denkbar einfach gestaltet. Zunächst ist es mittels der beiden Auswahlfenster für den Schnitt und das Segment möglich, ein bestimmtes Segment innerhalb einer Schnittebene an Hand seiner Nummer zu finden. Alternativ dazu kann das Segment nach Auswahl der richtigen Schnittebene auch durch einen Mausklick in der Übersicht ausgewählt werden. Wie nach jedem Schritt werden nach dieser Auswahl alle Ansichten aktualisiert und – wenn das Segment zu einer 3D-Form gehört – wird auch diese sowohl als Nummer als auch im unteren Bereich als Grafik angezeigt.

Alternativ zur Auswahl eines bestimmten Segmentes kann auch eine 3D-Form im rechten Auswahlfenster angewählt werden. Bei einer 3D-Form wird deren mittleres Segment und der dazugehörige Schnitt angezeigt, sowie die Form selbst im unteren Fenster.

Mittels des Checkbuttons in der Navigationsleiste ist es möglich, die grüne Markierung der Segmente abzuschalten oder zu aktivieren. Diese beiden Darstellungsweisen sind in Abbildung 6.2 auf der gegenüberliegenden Seite zu sehen.

Mittels dieses Werkzeugs ist es nun relativ einfach möglich, die Ergebnisse der Filterung zu überprüfen. Zum einen können die vom Computer definierten Formen der Reihe nach betrachtet werden, um zu entscheiden, ob es sich dabei tatsächlich um

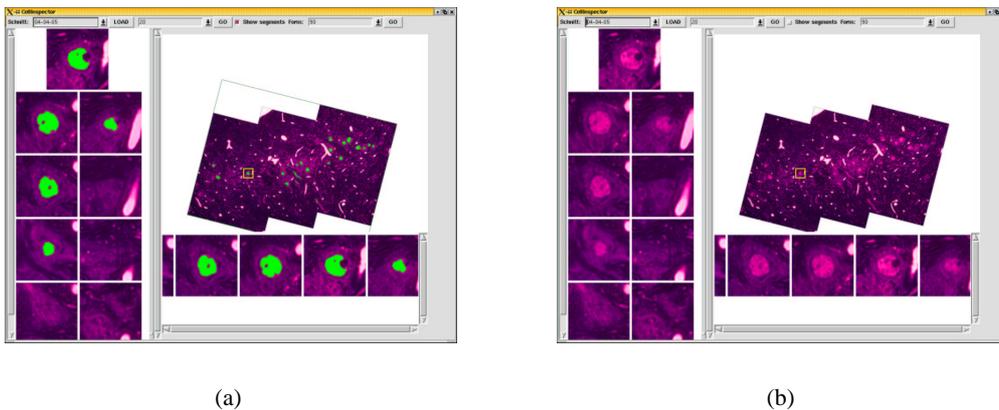


Abbildung 6.2: Das Programm *CellInspector* mit an- und ausgeschalteter Segmentenmarkierung

einen Zellkern handelt; zum anderen ist es aber auch möglich, die Schnitte der Reihe nach zu inspizieren, um herauszufinden, ob ein Kern nicht markiert wurde.

Bevor ich im nächsten Abschnitt genauer auf die Validierung der Methode eingehe und ihr Ergebnis vorstelle, möchte ich zunächst noch kurz einige weitere Verwendungszwecke dieses Programms skizzieren.

Zunächst einmal ist der *CellInspector* nicht nur ein Werkzeug zum Validieren der Ergebnisse, sondern vielmehr ein generelles Bindeglied zwischen der digitalen Aufnahme der Schnitte und der dreidimensionalen Rekonstruktion. Wie im Abschnitt 6.4 noch näher erläutert wird, ist es mit einer speziellen Rekonstruktion einfach möglich, die fortlaufende Nummer einer bestimmten 3D-Form herauszufinden. Diese kann dann im *CellInspector* weiter betrachtet und untersucht werden.

Neben dieser Verbindungs-Funktion zwischen 2D und 3D kann der *CellInspector* außerdem zu einem umfangreichen Toolset zur Analyse der Strukturen ausgebaut werden.

So wäre es z.B. möglich, das Programm so zu erweitern, dass einzelne Zellkerne manuell entfernt werden können, ihre Kontur verbessert werden kann oder der Benutzer zu einzelnen Zellen eine Notiz ablegen kann. Auf diese Weise wäre es mit vertretbarem Aufwand möglich, nach Merkmalen zu screenen, die von einem Computerprogramm nicht analysiert werden können.

Als Fazit lässt sich festhalten, dass der *CellInspector* ein leistungsfähiges Werkzeug für die Validierung der Ergebnisse ist und ein großes Potenzial für die Integration weiterer Analysemöglichkeiten besteht.

6.3 Ergebnis

Nachdem die Schnitte korrekt aneinander ausgerichtet sind, besteht das Ziel darin, die Zellkerne dreidimensional darzustellen und zuvor die Artefakte herauszufiltern.

Zunächst werden aus den isolierten Segmenten Formen gebildet, indem einfach sämtliche Segmente, die sich mit einem anderen Segment der vorausgehenden oder der nachfolgenden Ebene überlappen, zu einer Form zusammengefasst werden.

Diese Formen wurden dann mit dem *CellInspector* einzeln manuell kontrolliert und klassifiziert. Dabei wurde jede Form entweder als Zellkern oder als Artefakt klassifiziert, wobei alle Formen, die nur unvollständig abgebildet waren, ausgelassen wurden. Diese unvollständige Abbildung rührt daher, dass der hier verwendete Ausschnitt der MSO keinen ganzen Nukleus abbildet und somit einige Zellkerne in den Randregionen angeschnitten sind. Bei einer kompletten Rekonstruktion eines Nukleus sollten solche Zellen nicht vorkommen.

Unter Auslassung dieser Zellkerne wurden in der Probe insgesamt 270 Formen gezählt, wobei von mir 178 als Zellkern und 92 als Artefakt klassifiziert wurden.

Diese hohe Fehlerrate von 92 zu viel erkannten Strukturen soll nun durch unterschiedliche morphologische Filter reduziert werden. Die Konstruktion eines solchen Filters kann prinzipiell beliebig komplex sein; im folgenden wird jedoch ein vergleichsweise einfacher Filter vorgestellt. Die einfache Struktur garantiert eine höhere Übertragbarkeit der Ergebnisse auf anderen Strukturen, als dies bei einem hochspezialisierten, extra für diese Serie zu entwickeltem Filter der Fall gewesen wäre.

In einem ersten Schritt wird von einer Zelle gefordert, dass sie in der axialen Richtung eine gewisse minimale Ausdehnung aufweisen muss. Deshalb werden zunächst alle Formen, die weniger als drei Segmente in Richtung der z-Achse aufweisen entfernt.

Dieser erste Filterschritt bringt bereits einen großen Effekt. Wie in Abbildung 6.3(b) zu sehen ist, werden nach diesem Filtervorgang 175 Formen als Zellkern klassifiziert, wobei 13 Artefakte als Zellkern und 16 Zellkerne als Artefakte klassifiziert wurden.

Um insbesondere die kleinen Interneurone, die oft nur über zwei Schichten reichen, in die Klasse der Zellkerne zu integrieren, wird im zweiten Schritt jedoch ein Teil des ersten Filters wieder aufgehoben, indem eine Form dann als Zellkern klassifiziert wird, wenn sie besonders rund ist.

Um die Rundheit einer Struktur zu definieren gehe ich wie folgt vor:

Es wird zunächst dasjenige Paar zweier Punkte p_1 und p_2 innerhalb eines Segmentes gesucht, so dass gilt:

$$\sqrt{(p_{1x} - p_{2x})^2 + (p_{1y} - p_{2y})^2} = \max(\sqrt{(p_{ix} - p_{jx})^2 + (p_{iy} - p_{jy})^2}), \quad (6.1)$$

$$i = 0 \dots n, j = 0 \dots m$$

wobei n bzw. m die Anzahl der Punkte des entsprechenden Segmentes ist und die Indices x und y für die x- bzw. y-Koordinate des jeweiligen Pixels stehen. Bei der Betrachtung der Rundheit wird die laterale Achse nicht weiter betrachtet, da – wie oben ausgeführt – deren um ca. den Faktor 10 schlechtere Auflösung keine exakte Aussage erlaubt.

Diese Gerade wird nun als Hauptachse des Segmentes definiert. Der Mittelpunkt dieser Achse wird im folgenden als Mittelpunkt des Zellkerns bezeichnet.

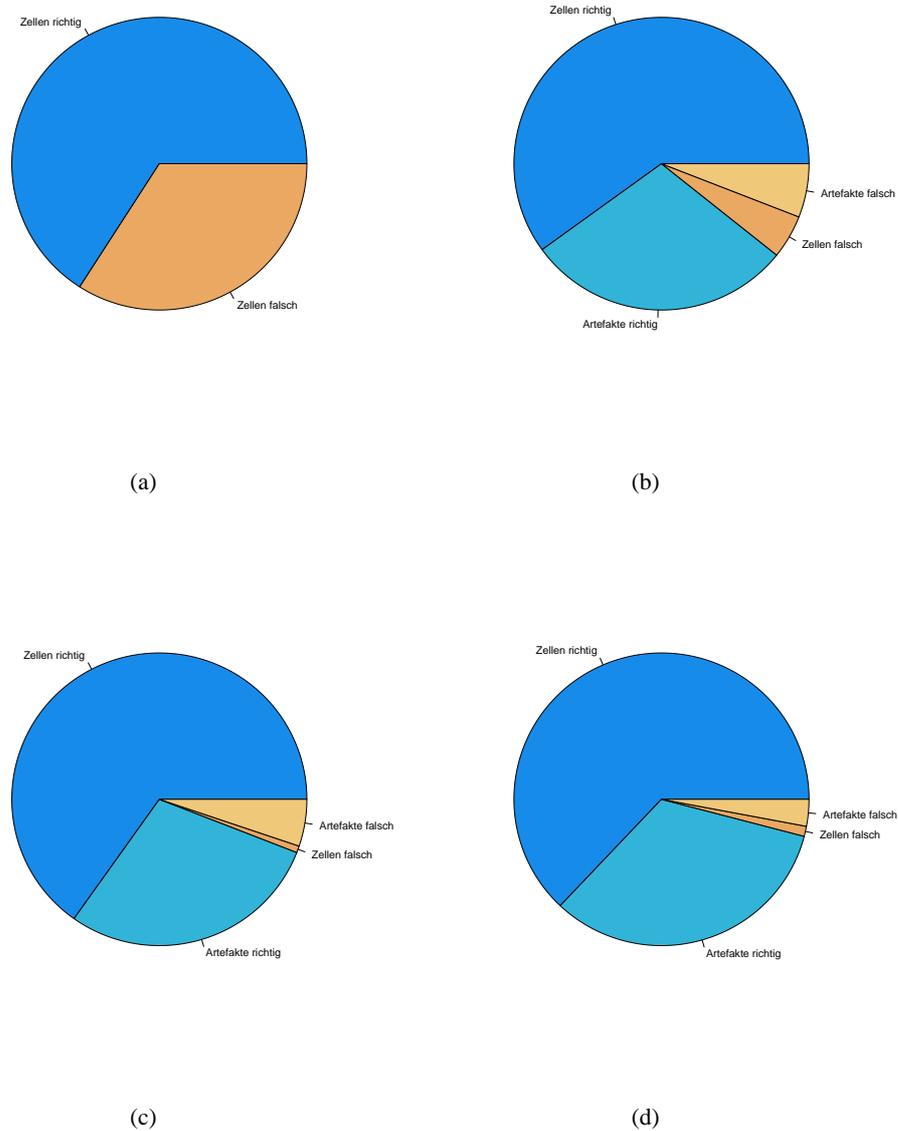


Abbildung 6.3: In der Abbildung sind die Ergebnisse der einzelnen Filterschritte abgebildet. In (a) ist die Ausgangssituation zu sehen, in (b) sind die Segmente nach ihrer Größe gefiltert, in (c) wurde die Rundheit und in (d) die minimale und maximale Ausdehnung als Filterkriterium hinzugefügt (Näheres im Text).

Nun wird für die Winkel 0, 45, ... 315, 360 jeweils eine Gerade vom Mittelpunkt des Zellkerns mit dem entsprechenden Winkel relativ zur Hauptachse gezogen und der Schnittpunkt dieser Geraden mit der Segmentgrenze festgehalten. Die Distanzen zwischen dem Mittelpunkt und dem entsprechenden Schnittpunkt werden im folgenden als d_α bezeichnet.

Ein erstes einfaches Charakteristikum für die Rundheit eines Segmentes lässt sich aus dem Mittelwert und der Standardabweichung der Werte berechnen:

$$f_1 = \frac{std(d_\alpha)}{mean(d_\alpha)} \quad (6.2)$$

Wobei $std()$ die Standardabweichung und $mean()$ das arithmetische Mittel bezeichnet. Liegt dieser Faktor unter einem bestimmten Wert, wobei sich 0,25 als geeignet erwiesen hat, so wird dieser Form das Attribut Zellkern zugewiesen, auch wenn sie nur zwei Segmente hat.

Nach diesem zweiten Filter wurden 190 Formen als Zellkern erkannt, wobei 14 Artefakte fälschlicherweise als Zellkern identifiziert wurden, allerdings nur zwei Zellkerne nicht erkannt wurden.

Diese zwei einfachen Filterregeln ergeben bereits ein gutes Ergebnis, das von ausreichender Qualität sein sollte, wenn man mehr Wert darauf legt, alle Zellkerne zu identifizieren und falsch erkannte Kapillaren das geringere Problem darstellen.

Ansonsten erfolgt nun noch ein dritter Filter, der das Verhältnis der Fehler verschiebt und einige Fehlklassifikationen beseitigt, indem wiederum einige Formen aussortiert werden. Bei diesem Schritt geht es darum, extrem lange und dünne Segmente auszusortieren.

Dazu wird folgender Filter definiert:

$$f_2 = \frac{max(d_\alpha)}{min(d_\alpha)} \quad (6.3)$$

Liegt dieser Faktor oberhalb von 2,5, so wird diese Form aussortiert, da dies sehr dafür spricht, dass es sich hierbei um eine Kapillare handelt. Diese werden häufig fälschlicherweise erkannt und können durch ihre langgestreckte Form aussortiert werden.

Wenn man auch diesen Filter anwendet, so sind lediglich drei Artefakte als Zelle klassifiziert und acht Zellen wurden den Artefakten zugeordnet (siehe Abbildung 6.3 auf der vorherigen Seite(c)).

Es wurden demnach von 270 Formen 11 falsch klassifiziert; dies entspricht einer Fehlerrate von 4,07% bezogen auf die gesamte Anzahl der Formen. Bezieht man die 11 fehlerhaften Formen auf die 178 Zellen, so ergibt sich eine Gesamtfehlerrate von 6,18%.

Es wäre prinzipiell sehr einfach, diese Fehlerrate weiter zu verringern und durch komplexere Filterregeln; etwa auch solchen, die nicht auf der klassischen booleschen Logik, sondern auf einem Konzept von Fuzzy-Entscheidungen beruhen, eine fehlerfreie Klassifikation zu erreichen (mittels eines genetischen Algorithmus wäre die Su-

che nach einem solchen Klassifikator sogar mit vergleichsweise wenig Aufwand verbunden.).

Jedoch wäre ein solches Ergebnis nicht auf die allgemeine Anwendungspraxis übertragbar, da es prinzipiell nicht zulässig ist, einen Klassifikator mit einer Menge zu entwickeln, die zugleich Trainings- und Testset ist. Die hier definierten einfachen Regeln hingegen sollten sich – gegebenenfalls mit einer geringfügigen Anpassung – auf jedes beliebige Präparat übertragen lassen.

6.4 Verschiedene Möglichkeiten der Rekonstruktion

Bei der Rekonstruktion der Zellkerne hat man die Wahl zwischen vielen unterschiedlichen Methoden. Prinzipiell lassen sich die Form, die Farbe sowie eine eventuelle Anmerkung in Textform zu einem Zellkern nutzen, um unterschiedliche Informationen zu vermitteln.

Die realistischste Form der Darstellung ist eine Rekonstruktion der Oberflächen. Ich habe dazu eine Variation des *Marching Cubes*-Algorithmus [64] verwendet, das ich im folgenden kurz beschreiben möchte. Der einzige Unterschied zum Originalalgorithmus besteht darin, dass auf einen Teilschritt zur Anpassung der Objektkontur verzichtet wird:

Der Marching Cubes Algorithmus ist ein Verfahren, das für das Erstellen von dreidimensionalen Oberflächenmodellen aus Serienbildern konzipiert wurde. Das Hauptanwendungsgebiet sind im medizinischen Bereich die bildgebenden Verfahren wie z.B. die Computertomografie.

Die Grundlage des Algorithmus bildet ein beliebig zu konfigurierendes Raster aus Würfeln. Im vorliegenden Fall wurde das Raster so gewählt, dass für jede Form die gleiche Anzahl von Würfeln verwendet wurde. Dazu wurde aus allen zu der Form gehörigen Voxeln p_{xyz} eine einhüllende Box vom Punkt $p_{\min(x)\min(y)\min(z)}$ nach $p_{\max(x)\max(y)\max(z)}$ aufgespannt und entsprechend der gewünschten Auflösung unterteilt.

Das Problem der Oberflächenrekonstruktion wird von dem Algorithmus durch die Strategie des „Teile und Herrsche“ gelöst, indem für jeden einzelnen Würfel bestimmt wird, ob ein Teil der Oberfläche durch diesen Würfel verläuft.

Dazu wird für jeden der acht Eckpunkte des Würfels festgestellt, ob dieser innerhalb oder außerhalb der Kontur liegt. Hierzu wird einfach getestet, ob der diesem Punkt am nächsten liegende Voxel ein Teil der untersuchten Form ist oder nicht.

Wie in Abbildung 6.4 auf der folgenden Seite schematisch dargestellt, werden dann innerhalb des Würfels eine oder mehrere Flächen konstruiert, so dass sämtliche Würfecken, die innerhalb der Form liegen (in der Abbildung rot) von jenen, die außerhalb der Form liegen, getrennt sind.

Ein großer Vorteil dieses Algorithmus liegt darin, dass sämtliche Möglichkeiten, wie die acht Ecken belegt sein können, in einem Byte gespeichert werden können, welches dann dafür verwendet werden kann, die geeigneten Formen aus einem Lookup-Array abzufragen. Dadurch benötigt die Rekonstruktion nur sehr wenig Rechenkapazität und ist schnell durchführbar.

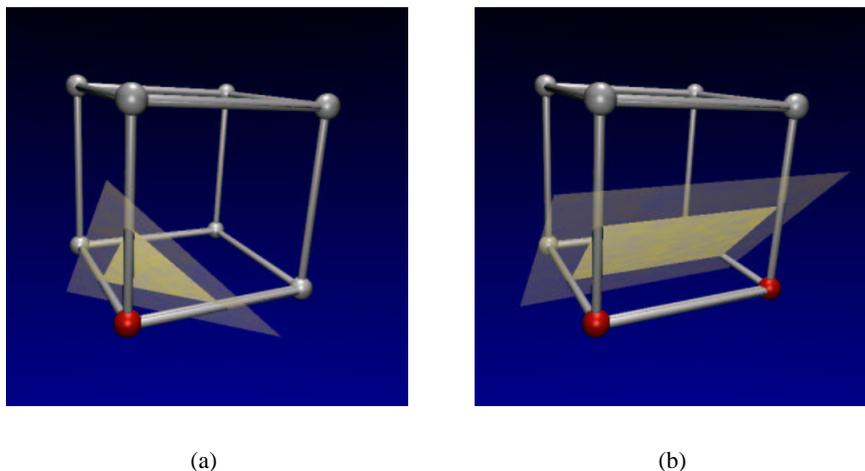


Abbildung 6.4: Schematische Darstellung zweier Konfigurationen für einen Würfel beim Marching Cubes Algorithmus

Die so erzeugte Rekonstruktion gibt die Kontur und die Größe der Zellkerne sehr exakt wieder und vermittelt zudem einen sehr plastischen Eindruck der Verteilung. In Abbildung 6.5 auf der nächsten Seite ist der hier verwendete Teil der MSO aus unterschiedlichen Richtungen gezeigt. Um die Orientierung zu erleichtern, ist in der Mitte eine einfache 3D-Form abgebildet, auf die eine der äußeren Abbildung entsprechende Transformation angewendet wird.

Der vielleicht größte Vorteil dieser Art der Darstellung kann in gedruckter Form jedoch nicht vermittelt werden. Die Rekonstruktion wird als VRML-Dokument ausgegeben, das mit frei erhältlichen, leistungsfähigen Betrachtern in Echtzeit manipuliert und betrachtet werden kann. So ist es möglich, die Anordnung zu drehen, zu vergrößern, sich durch die Formen hindurch zu bewegen oder aber auch gezielt Veränderungen vorzunehmen.

Jedoch ist eine solch detaillierte Rekonstruktion nicht immer ideal, da sich bestimmte Informationen besser darstellen lassen, wenn die Darstellungsweise deutlich einfacher ist.

So bietet es sich z.B. an, die Zellkerne nicht in ihrer realen Form abzubilden, sondern sie durch formangepasste oder sogar gleichgroße Kugeln zu ersetzen. Dadurch entsteht ein ruhigeres Bild, in dem zusätzliche Informationen besser eingebracht werden können.

Im folgenden werde ich einige Möglichkeiten der Veranschaulichung vorstellen und dabei auf deren potenzielle Anwendung eingehen.

In Abbildung 6.6 auf Seite 115 ist oben zunächst eine einfache Rekonstruktion zu sehen, bei der die einzelnen Zellkerne durch Kugeln ersetzt wurden. Insbesondere bei großen Arealen kann so besser die Übersicht gewahrt bleiben.

Vor allem aber bietet es sich an, die so erstellten Zellkerne mit weiteren Informationen zu versehen. So wurde z.B. in (b) eine hierarchische Clusteranalyse [34] mit den

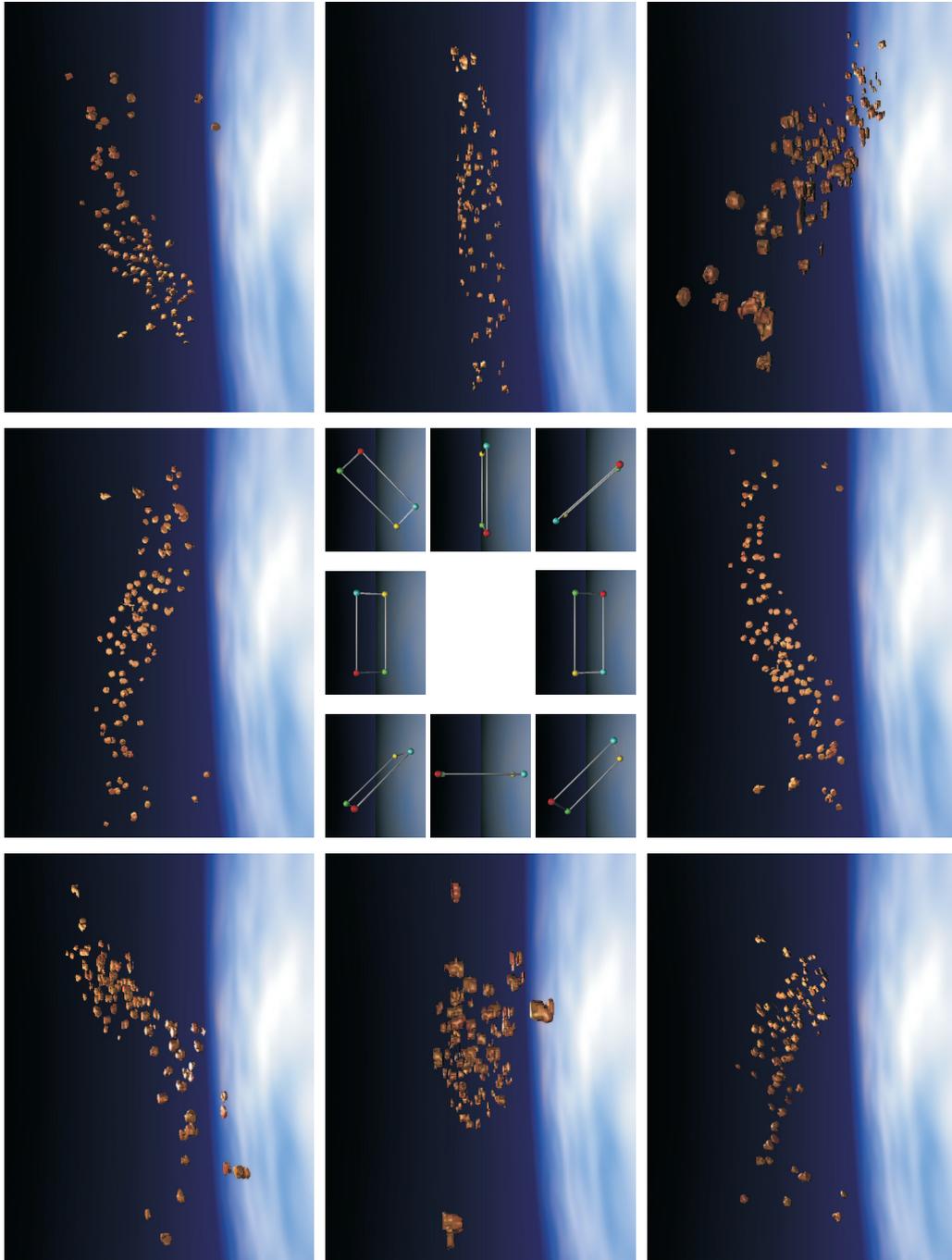


Abbildung 6.5: Rendering der Zellkerne aus verschiedenen Perspektiven

drei Raumkoordinaten durchgeführt. Die einzelnen Cluster werden dabei mit einer zufällig gewählten Farbe markiert.

Diese Methode könnte bei der komplett rekonstruierten MSO z.B. dazu dienen, Gruppen von Zellen zu identifizieren oder generell die Grenze eines Nukleus mathematisch exakt zu bestimmen. Prinzipiell kann fast jedes beliebige Merkmal über die Farbe kodiert werden. So kann man die Farbe z.B. dazu verwenden, die Größe einer Zelle, die bei der Zelle vorhandenen Rezeptoren oder beliebige andere Daten anzuzeigen – vorausgesetzt diese wurden zuvor erfasst.

In Abbildung 6.6(c) wurde zum einen die Farbe verwendet, um kenntlich zu machen, wie weit die Entfernung zur nächste Zelle ist und zum anderen wurde jeweils eine Verbindung zur räumlich nächsten Zelle gezogen. Rot kodiert eine weite Entfernung, grün eine sehr kurze Entfernung und die gelb markierten Zellen befinden sich zwischen diesen Maßen. Bei dieser Auswertung sollte es leicht möglich sein, eine räumliche Gliederung, wie etwa einen schichtartigen Aufbau oder andere Gruppierungen sichtbar zu machen – sofern eine solche vorhanden ist.

Die unterschiedlichen Darstellungsweisen sollen vor allem deutlich machen, dass die Möglichkeiten, weitere Informationen zu integrieren nahezu unbegrenzt sind. Insbesondere wenn zu einem späteren Zeitpunkt weitere Informationen wie Somagröße, Projektionswege oder die Rezeptorenverteilung zur Verfügung stehen, wird es immer wichtiger, diese Informationen in einer übersichtlichen Art und Weise darzustellen.

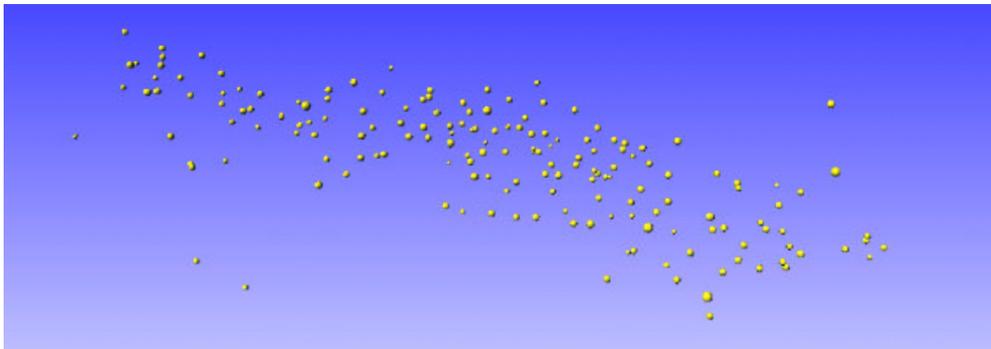
In Abbildung 6.7 auf Seite 116 sind weitere Möglichkeiten der Markierung von einzelnen Zellen dargestellt. So ist es nicht nur möglich, Zellen mit einer bestimmten Farbe (a) zu markieren, sondern man kann statt dessen auch eine Textur, d.h. ein Muster aus zwei unterschiedlichen Farben verwenden (b), ferner kann man die Form des Symbols ändern (c) oder die Oberfläche mit unterschiedlichen Markierungen versehen (d) und (e). Besonders nützlich ist die Möglichkeit, zu einer Zelle einen Text hinzuzufügen (f).

Auf die zuletzt genannte Möglichkeit möchte ich kurz eingehen. Dadurch, dass jeder Zellkerne eine eindeutige Identifikations-Nummer trägt, ist es wie in Abbildung 6.8 gezeigt, einfach möglich, eine bestimmte Zelle zu identifizieren und sämtliches Datenmaterial zu dieser abzurufen.

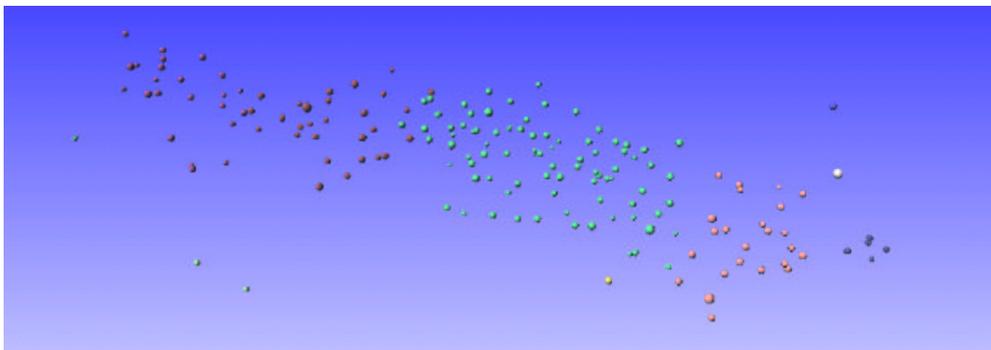
Im Beispiel wurde die Zahl dazu verwendet, um einen Bezug zu der ursprünglichen Aufnahme herzustellen und die Zelle im *CellInspector* zu betrachten. Ebenso wäre es möglich, weitere erfasste Daten zu betrachten. Würde man an Stelle der bisher verwendeten VRML-Ausgabe einen eigenen Viewer verwenden, könnten die verschiedenen Daten sogar direkt während des Betrachtens eingeblendet werden, wenn die Maus das entsprechende Symbol anklickt.

Auch wenn die hier gezeigten Möglichkeiten nur exemplarisch sind, wird wohl deutlich, dass die dreidimensionale Darstellung viele Möglichkeiten beinhaltet, weitere Informationen zu integrieren und miteinander zu verknüpfen.

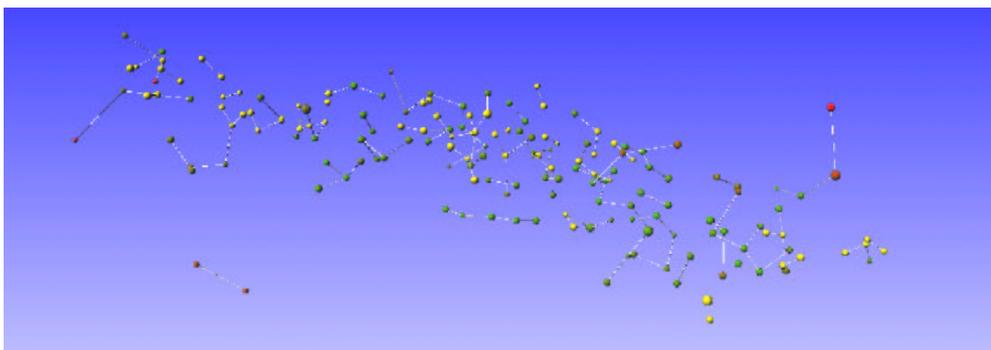
Der vielleicht wichtigste Aspekt kann leider nicht gezeigt werden, denn die eigentliche Stärke der Darstellung entsteht erst dadurch, dass der Benutzer interaktiv zwischen unterschiedlichen Darstellungsformen wechseln kann, diese kombinieren und einen beliebigen Blickwinkel wählen kann. Durch diese Interaktion bieten die 3D-Rekonstruktionen eine wichtige Basis für Modellvalidierungen, morphologische Untersuchungen und die weiteren in Kapitel 7 diskutierten Ansätze.



(a)



(b)



(c)

Abbildung 6.6: Unterschiedliche Darstellungsformen: In (a) ist eine Rekonstruktion zu sehen, bei der die Somata als Kugeln dargestellt sind. In (b) wurden Cluster gebildet und diese farblich kodiert. In (c) schließlich wurde die Farbe verwendet, um die Entfernung zur nächsten Zelle zu markieren; gleichzeitig wurde zu dieser eine Linie gezogen.

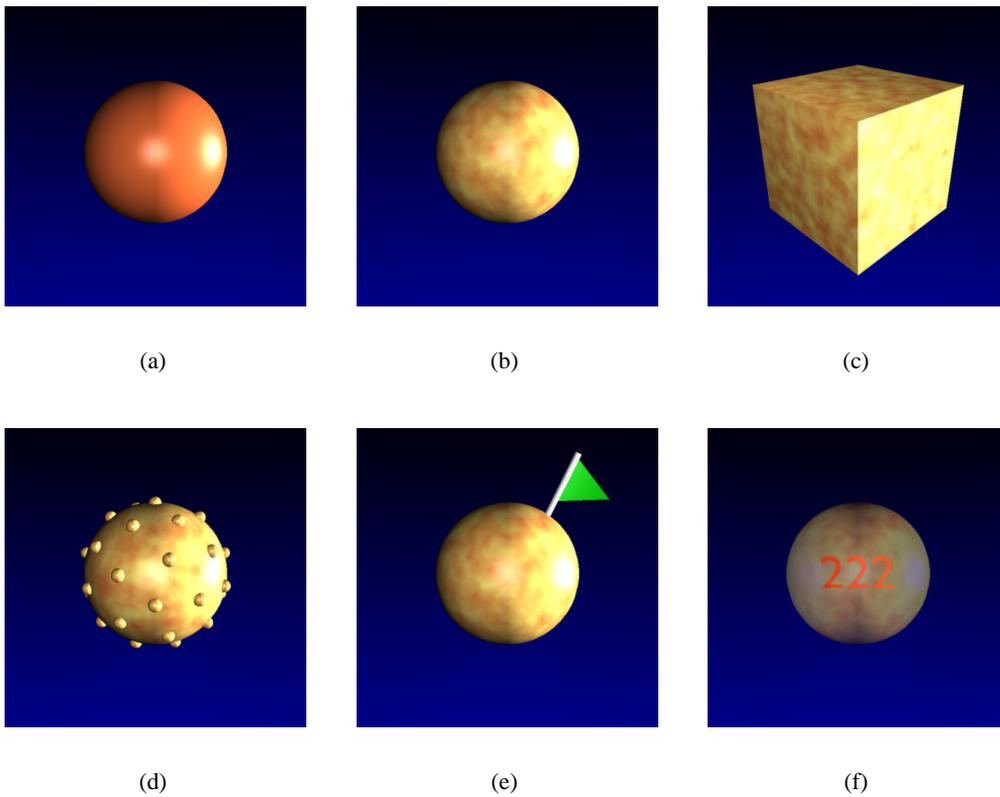


Abbildung 6.7: Verschiedene Möglichkeiten zusätzliche Informationen in eine 3d-Grafik zu integrieren: Verschiedene Daten können über die Farbe (a), die Textur (b), die Form (c), durch Markierungen (d+e) oder durch einen zusätzlichen Text (f) eingebracht werden.

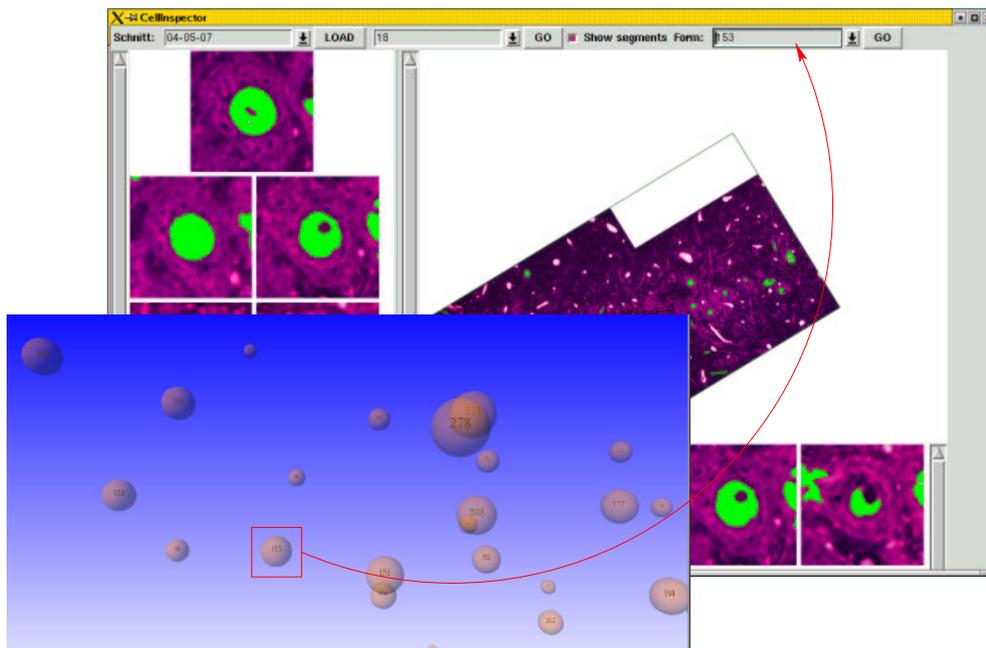


Abbildung 6.8: An einem Beispiel wird hier gezeigt, wie eine bestimmte Zelle in einer 3D-Ansicht identifiziert wird. Die gleiche Zelle kann dann im *CellInspector* betrachtet werden.

6.5 Fazit

In diesem Kapitel wurde zunächst gezeigt, wie die vergleichsweise hohe Fehlerrate, die durch die Mustererkennung entstanden ist, mit wenigen Filterschritten kompensiert werden kann, so dass eine Fehlerrate in der Größenordnung von ca. 5% erreicht wird. Das Verfahren wurde zwar nicht gegen einen menschlichen Experten validiert; es ist jedoch davon auszugehen, dass bei einer manuellen Rekonstruktion eine ähnliche Fehlerrate aufträte.

Mit dem *CellInspector* kann das Ergebnis gut validiert werden. Außerdem bietet diese grafische Oberfläche auch die Möglichkeit, jederzeit die Originalaufnahme einer in der 3D-Ansicht identifizierten Zelle aufzurufen und die histologischen Präparate zu betrachten.

Die Durchführung der eigentlichen Rekonstruktion basiert auf dem Marching Cubes Algorithmus. Die so generierten 3D-Formen entsprechen zwar weitestgehend der realen Form, wirken allerdings bei der Betrachtung oft störend. Aus diesem Grund werden am Ende des Kapitels unterschiedliche Formen der Darstellung präsentiert.

Die hier vorgestellten Verfahren der Rekonstruktion bilden eine gute Ausgangsbasis für sämtliche im folgenden Kapitel beschriebenen Auswertungsmöglichkeiten.

Diskussion

Da bereits in jedem der einzelnen Kapitel die jeweilige Methode und deren Ergebnisse diskutiert wurden, möchte ich diese im ersten Teil der Diskussion nur kurz wiederholen und den Schwerpunkt auf die Stärken und Schwächen der einzelnen Schritte legen.

Der zweite Teil geht näher darauf ein, an welchen Stellen die Software noch erweitert werden kann und welche Funktionen zusätzlich integriert werden können. Dieser Teil beschränkt sich dabei auf Erweiterungen, die lediglich auf einer Modifikation oder einem Ausbau der Software beruhen.

Der dritte Teil diskutiert darüber hinaus gehende Lösungen, wie durch weitere histologische Techniken und eine umfangreiche Erweiterung der mikroskopischen Apparatur das Verfahren beschleunigt und um die Erfassung anderer Daten erweitert werden kann.

Im letzten Teil schließlich werden die Anwendungsmöglichkeiten der Technik ausführlich erläutert. Dabei werde ich zunächst versuchen, das Potenzial dieser Methode im Bereich der Modellvalidierung und -bildung aufzuzeigen. Neben diesem Bereich, für den das Verfahren ursprünglich konzipiert wurde, werde ich jedoch auch weitere Anwendungsmöglichkeiten in anderen Bereichen der Biologie und der Medizin diskutieren.

7.1 Verfahrensschritte

Im Folgenden möchte ich das gesamte Verfahren kurz zusammenfassen und die Vor- und Nachteile gegenüber der manuellen Durchführung, sowie gegenüber anderen Verfahren – so diese existieren – diskutieren.

In Abbildung 7.1 auf Seite 121 ist der gesamte Verarbeitungsprozess nochmals als Übersicht dargestellt. Der Schwerpunkt liegt dabei auf den in den Verfahren implementierten Kontrollmechanismen sowie den Adaptationsmöglichkeiten.

Jeder Verfahrensschritt ist in einem gelben Kasten abgebildet, wobei für jeden Schritt die einzelnen Methoden separat aufgeführt und mit einem roten Rand markiert sind. Die Kontrollmechanismen der einzelnen Verfahren sind mit einem Lupensymbol gekennzeichnet.

Bei den meisten Verfahren hat der Anwender die Möglichkeit, deren Funktionsweise über verschiedene Parameter zu beeinflussen. Die wichtigsten Parameter sind jeweils in einem blauen Kasten zusammengefasst. Der blaue Kasten befindet sich immer neben einem roten Rechteck, das einen Zwischenschritt darstellt.

Der erste Schritt des Verfahrens ist die histologische Aufarbeitung. Wichtig ist es hierbei, dass die Schnitte dünn genug sind, so dass die Zellkerne auf mehreren Ebenen sichtbar sind und dass das für die Einbettung verwendete Material keinen Verzug zulässt. Bei der Färbung ist zu beachten, dass die Zellkerne deutlich sichtbar sind.

Es ist in diesem Rahmen prinzipiell möglich, alternative Färbe- und Einbettungsverfahren anzuwenden, sowie die Schnittdicke zu variieren.

In Kapitel 2.1 wurde bereits ausführlich besprochen, aus welchen Gründen Paraffin-Schnitte nicht geeignet sind. Durch die notwendige Dicke der Schnitte und den deutlichen Verzug, ist dort eine Zuordnung aufeinander folgender Zellkerne praktisch nicht möglich. Eine Alternative könnten hingegen Gefrierschnitte darstellen; wobei es sehr schwierig ist, diese mit einer solch geringen Dicke wie sie das Verfahren erfordert, anzufertigen.

Die Schnittdicke könnte noch dünner gewählt werden; dadurch würde jedoch kein weiterer Vorteil erzielt; dies wäre lediglich dann eine Option, wenn man zwei aufeinanderfolgende Schnitte jeweils unterschiedlich färben möchte. Eine Erhöhung der Schnittdicke ist prinzipiell möglich, jedoch wird dadurch die Fehlerrate besonders beim Ausrichten der Schnitte erhöht und sollte ein einzelner Schnitt verloren gehen, bedeutet das bei einer Schnittdicke von 4 μm oder mehr zumeist, dass einige Zellen nicht erfasst werden können. Da bei jeder Präparation einige Schnitte Falten werfen oder verloren gehen, kann durch die Wahl der Schnittdicke mit 2 μm bereits im ersten Schritt eine erhöhte Fehlertoleranz eingeführt werden.

Die Färbung kann leicht variiert werden; prinzipiell können alle Färbungen angewendet werden, die eine klare Trennung zwischen Zellkern, Soma und Hintergrund erreichen. Die Nissl-Färbung eignet sich jedoch in Kombination mit der Osmium-Fixierung sehr gut und erlaubt auf Grund der weiten Verbreitung eine einfache Nachvollziehbarkeit.

Die Schnittdicke ist somit ein relativ fixer Parameter, der mit 2 μm sehr gut dimensioniert ist. Die Färbemethode ist für die vorliegenden Aufgabenstellung optimiert; es ist allerdings gut möglich, dass durch eine alternative Färbung die Mustererkennung nochmals erleichtert werden könnte. Zur Einbettung in Kunstharz gibt es keine sinnvolle Alternative – jedoch besteht auch kein Bedarf, an dieser Stelle auf eine andere Methode zurückzugreifen.

Wie bereits des Öfteren betont, ist die folgende mikroskopische Aufnahme ein sehr wichtiger Schritt, bei dem noch viele Möglichkeiten zur Verbesserung vorhanden sind.

Eine Alternative zur Lichtmikroskopie wäre das konfokale Laserscanning-Mikroskop (CLSM), das jedoch nur fluoreszierende Proben verarbeiten kann. Dadurch wird die Bearbeitung der Proben teuer und es ist z.B. nur schwer möglich, die Soma-grenzen aller Zellen festzustellen. Des Weiteren würde auch beim CLSM die MSO nicht in einem Schnitt abzubilden sein, so dass in den Grenzbereichen die gleichen Probleme auftraten. Auch wenn die hier vorgestellten Verfahren prinzipiell auf andere Techniken, wie z.B. das CLSM oder die Elektronenmikroskopie anwendbar sind,

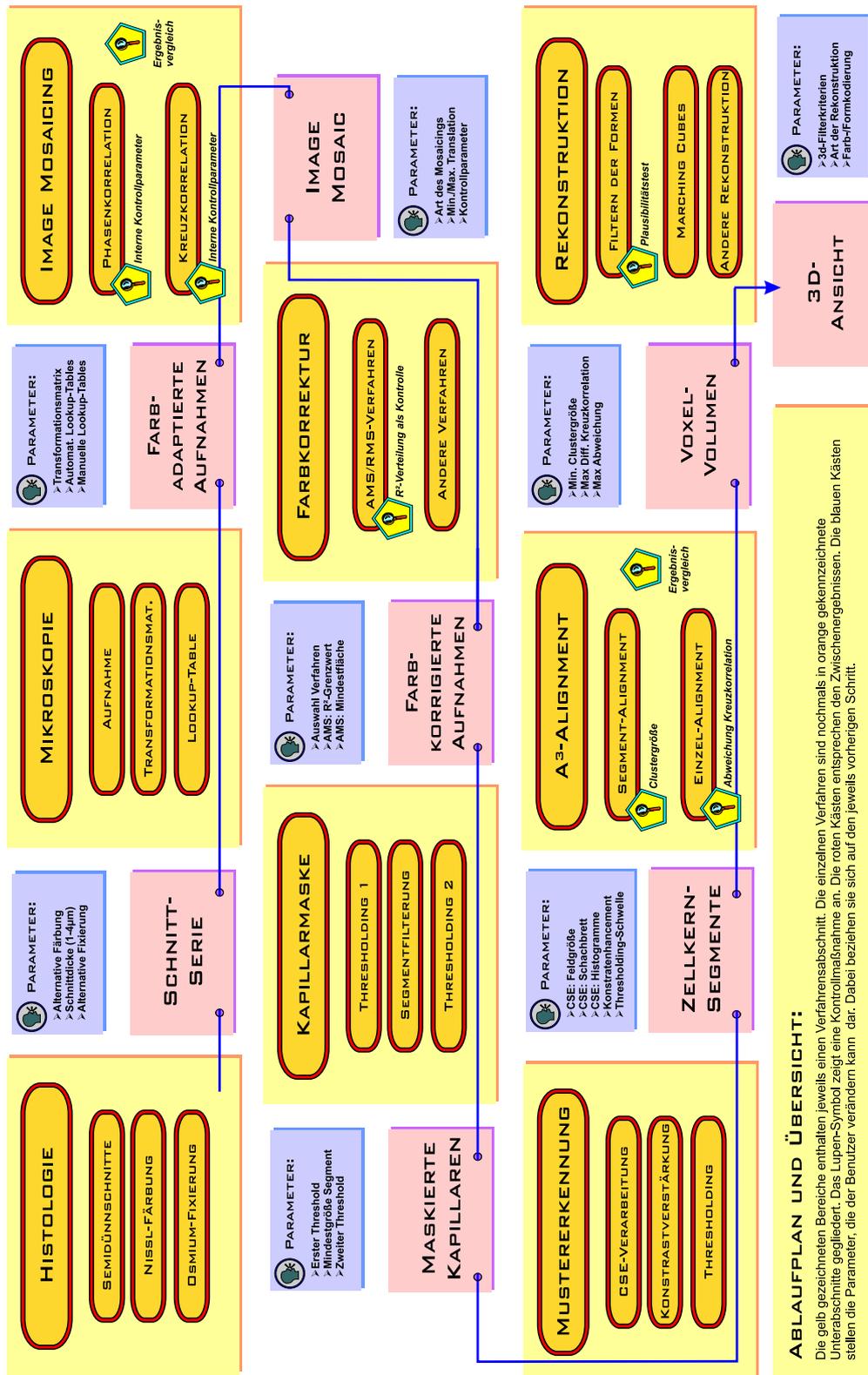


Abbildung 7.1: Übersicht des gesamten Verfahrens

so ist das Lichtmikroskop dennoch die universellste und damit auch die für unsere Fragestellung am besten geeignete Methode.

Die Verwendung einer Lookup-Table in Kombination mit der Transformationsmatrix bietet eine gute Möglichkeit, die Mustererkennung zu vereinfachen. Das Verfahren ist anderen Formen der Farbadaptation am Mikroskop, wie etwa einer einfachen Regelung der Beleuchtung, deutlich überlegen. Es wäre dennoch möglich, das Verfahren wesentlich zu verbessern, indem man eine eigene Software für die Aufnahme der Bilder schreibt, die zunächst eine Probeaufnahme des Bildes macht, dann dessen Histogramm bewertet und daraus dynamisch eine Lookup-Table berechnet, welche bereits die nötigen Korrekturfaktoren für die Berechnung der Far Konstanz enthält. Dieser Schritt wäre mit einigem Aufwand und einer Kooperation der Herstellerfirma verbunden, würde jedoch den Prozess der Mustererkennung deutlich verbessern.

Das Mikroskop ist auf Basis der momentan vorhandenen Möglichkeiten ausgereizt, allerdings bietet sowohl eine Modifikation der Software als auch ein Ausbau der Hardware, wie er weiter unten beschrieben wird, sicherlich eine attraktive Möglichkeit, das Verfahren weiter zu verbessern.

Der nächste Schritt ist das Image Mosaicing. Dieses Verfahren besitzt bereits ein grafisches Interface, es existiert eine native, d.h. eine beschleunigte Version und durch mehrfache Kontrollen wird jeder Fehler erkannt. Das Image Mosaicing ist meines Erachtens nicht weiter zu verbessern und es existiert auch keine sinnvolle Alternative. Es ist zwar prinzipiell möglich, die Bilder mit der Hand auszurichten, da dies allerdings weder zu einem genaueren Ergebnis noch zu einem geringeren Zeitaufwand führt, ist dies irrelevant.

Die einzige eventuell sinnvolle Erweiterung für das Image Mosaicing würde darin bestehen, die Kontrollparameter des Algorithmus automatisch auf der Grundlage einer vorhandenen Serie abzuleiten. Der Aufwand für ein solches Programm wäre allerdings nur dann gerechtfertigt, wenn das Verfahren wirklich in großem Umfang eingesetzt würde.

Der folgende Schritt hingegen, die Farbkorrektur, bietet sicherlich noch Raum für Optimierungen. Die Forschung im Bereich der „color constancy“ ist ein großer Zweig innerhalb der Bildanalyse und bringt laufend neue Verfahren hervor. Durch die Integration von weiteren, neuen Algorithmen könnte sicherlich die Anpassung der Farben noch weiter verbessert werden. Davon abgesehen, wäre es zum einen wünschenswert, die Korrektur der Farbe wie oben beschrieben direkt am Mikroskop vorzunehmen und zum anderen wäre es bei einer routinemäßigen Anwendung des Verfahrens mit unterschiedlichen Färbetechniken sicherlich sinnvoll, einen Algorithmus zu konzipieren, der automatisch eines der möglichen Verfahren zur Farbkorrektur wählt.

Die Farbkorrektur funktioniert zwar ausreichend gut für das zu lösende Problem, bietet aber sicherlich noch Raum für Optimierungen. Auch wenn die von mir entwickelte Lösung eine bessere Korrekturleistung als die traditionellen Verfahren aufweist, kann es durchaus sein, dass ein neueres Verfahren deutlich bessere Resultate liefert.

Die Mustererkennung ist für den vorliegenden Fall sehr gut angepasst und weist trotzdem genügend Spielraum auf, um auch andere, ähnlich aufgebaute Bilder zu bearbeiten. Durch eine Anpassung der Parameter des CSE-Algorithmus, eine zusätzliche Kontraststeigerung und eine Veränderung der Schwelle des Thresholdings kann sicherlich eine Anpassung an die meisten histologischen Präparate erreicht werden.

Wenn man die Erkennungsrate mit der einer manuellen Markierung vergleicht, so kann man sehen (siehe Kapitel 4.6.1), dass der Computer das Problem zumindest eben so gut wie der Mensch zu lösen vermag.

Will man hingegen vollkommen unterschiedlich aufgebaute Bilder verarbeiten (etwa elektronenmikroskopische Aufnahmen), so wird man natürlich die Methoden der Mustererkennung variieren und gegebenenfalls austauschen müssen.

In Zukunft sollte deshalb bei diesem Punkt das primäre Ziel auch weniger darin bestehen, immer exaktere Methoden zu entwickeln, sondern vielmehr darin, ein breites Methodenspektrum in die Software zu integrieren, so dass es einfacher wird, neue Problemfelder zu bearbeiten und alternative Lösungen zu finden.

Der Aristoteles-Alignment-Algorithmus (A^3) ist in der vorliegenden Form robust, produziert ein sehr präzises Alignment und ist nur schwer noch weiter zu verbessern.

Es konnte gezeigt werden (siehe Kapitel 5.3), dass das Alignment ähnlich genau arbeitet wie ein menschlicher Experte. Somit bringt das manuelle Ausrichten keinerlei Vorteile mit sich. Andere Algorithmen, die ohne manuelle Eingriffe in der Lage sind, die Schnitte auszurichten, sind mir nicht bekannt. Da auch der A^3 eine interne Fehlerkorrektur aufweist, arbeitet er zumeist fehlerfrei und erkennt auftretende Fehler automatisch.

Der letzte Schritt, die Rekonstruktion, beruht auf einem erprobten Algorithmus und kann nur schwer weiter optimiert werden. Die nach dem Filtern erreichte Fehlerrate von ca. 5% ist ebenfalls nicht zu beanstanden. Es ist jedoch möglich, die zur Verfügung stehenden Filterkriterien zu erweitern. So könnte man es dem Betrachter erlauben, an Hand einer manuell klassifizierten Menge von Formen entweder einen vollkommen automatisch arbeitenden Klassifikator (etwa ein Neuronales Netz oder einen Genetischen Algorithmus) zu verwenden oder aber mittels eines Fuzzy-Logik-Systems intuitive Regeln festzulegen.

Insgesamt arbeiten alle hier vorgestellten Verfahren äußerst robust, lassen sich vom Anwender sehr weitreichend konfigurieren und damit auch auf andere Problemfelder anpassen.

Wichtig ist außerdem hervorzuheben, dass an den drei Stellen, an denen ein menschlicher Fehler oder ein Problem mit der Software zu einem Misslingen der gesamten Rekonstruktion führen kann, der entsprechende Algorithmus immer eine oder mehrere Kontrollen aufweist, die es leicht machen, die Fehler hinterher manuell zu korrigieren:

- Beim Image Mosaicing kann leicht durch nicht überlappende Bilder ein Fehler auftreten. Dieser wird durch die Kontrollen erkannt.
- Wenn durch ein Problem beim Färbeprozess eine zu große Farbabweichung auftritt, kann dies zu Problemen mit der Mustererkennung führen. Dieses Problem wird beim AMS erkannt.
- Sollte die Anzahl der Zellkerne nicht ausreichen, um das Bild korrekt auszurichten, wird auch dieser Fehler von einer Kontrolle erkannt und in vielen Fällen sogar behoben.

Diese Kontrollen sind außerordentlich wichtig, da ohne diese Maßnahme selbst eine Fehlerrate von wenigen Prozent nicht akzeptabel wäre. Wenn nur ein Schnitt von 1000

falsch ausgerichtet ist, so muß der Anwender – wenn dieser nicht automatisch markiert wird – den gesamten Stapel sichten, um den Fehler zu finden.

Durch die Kontrollmechanismen an den beschriebenen Stellen sind auch die restlichen Verfahren relativ gut gesichert. Durch die Kontrolle der Farbkorrektur werden Ausreißer in der Mustererkennung vermieden und durch eine Kontrolle des Ausrichtens sind Fehler in der Rekonstruktion sehr unwahrscheinlich.

Als Fazit kann man festhalten, dass alle Verfahren entweder fehlerfrei oder mit einer sehr geringen Fehlerrate arbeiten, in ihrer Präzision einer manuellen Durchführung zumindest ebenbürtig sind und die Software zwar noch einige Bereiche aufweist, in denen man Verbesserungen vornehmen *kann*, jedoch keinen Bereich, in dem man eine Verbesserung vornehmen *muss*.

7.2 Erweiterungen der Software

Wie im vorherigen Abschnitt gezeigt, ist die Entwicklung des Verfahrens praktisch abgeschlossen und bietet nur noch geringfügige Möglichkeiten der Verbesserung hinsichtlich Qualität und Korrektheit.

Ich möchte in diesem Abschnitt einige Erweiterungen vorstellen, die in der Zukunft noch umgesetzt werden sollten. Diese sind zwar für das hier vorgestellte Verfahren nicht unbedingt notwendig, allerdings sind sie Voraussetzung für einige weitergehende Analysen.

Teilweise sind diese Verfahren auch bereits implementiert. Die Umsetzung weist allerdings noch nicht die notwendige Robustheit auf, als dass diese Techniken in der Praxis eingesetzt werden könnten. Aus diesem Grund sollen sie hier nur kurz diskutiert werden und sind nicht Bestandteil des eigentlichen Verfahrens.

Die Erweiterung der Software ist vor allem in drei Bereichen notwendig:

Zum einen werden die Zellkerne zwar bereits richtig erkannt, allerdings führt der Nukleolus häufig zu einer Beeinträchtigung der Form. Dies müsste für Untersuchungen, bei denen die genaue Größe des Zellkerns wichtig ist, korrigiert werden.

Zweitens werden die Somata selbst noch nicht korrekt erkannt. Will man zunächst nur Anzahl und Lage der Zellen bestimmen ist dafür der Zellkern ausreichend. Wenn man bei späteren Untersuchungen jedoch die Größe der Zelle bestimmen möchte oder einen genauen Wert für das Kern-Plasma-Verhältnis benötigt, so ist eine Erkennung des Somas unabdingbar.

Letztlich sollte eine grafische Oberfläche für das gesamte Programm entwickelt werden, die es auch dem Laien ermöglicht, das Verfahren ohne große Einarbeitungszeit zu verwenden.

Für alle drei Verfahren existieren bereits Lösungsansätze, die ich im folgenden kurz diskutieren möchte.

7.2.1 Bestimmen der Zellkern-Kontur

Nachdem die Zellkerne korrekt extrahiert wurden, können diese zunächst für das Ausrichten der Schnitte verwendet werden. Möchte man später jedoch die Zellkerne von

ihrer Fläche her quantifizieren, so stößt man auf ein durch die Nukleoli hervorgerufen Problem: Der Nukleolus hat eine deutlich andere Färbung als der Zellkern selbst und wird deshalb nicht zum Zellkern hinzugerechnet. Dies resultiert in einer sichelförmigen Gestalt des Zellkerns, wie sie in Abbildung 7.2(b) zu sehen ist.

Um die wirkliche Begrenzung der Zellkerne zu bestimmen, sind mehrere Methoden denkbar.

Zunächst besteht eine recht einfache Möglichkeit darin, die maximale und die minimale Ausdehnung eines jeden Zellkern-Segmentes zu bestimmen und entlang dieser Achse eine Ellipse an die Fläche des Zellkerns anzupassen. Die Implementierung dieses Verfahrens ist in [50] zu finden.

Das in Abbildung 7.2(c) gezeigte Ergebnis dieses Verfahrens ist nicht zufriedenstellend. Die Ellipsen entsprechen in ihrer Fläche – wie durch das Verfahren vorgegeben – der ursprünglichen Fläche. Das Verfahren ist somit lediglich eine optische Verbesserung; es ist jedoch nicht geeignet, die Zellkerne exakter zu charakterisieren.

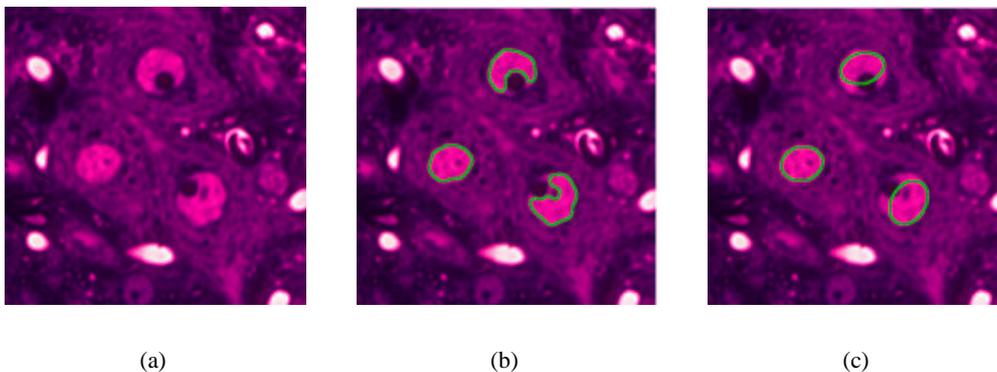


Abbildung 7.2: Aus dem in (a) gezeigte Originalbild wurden mit den zuvor beschriebenen Algorithmus die Zellkerne extrahiert (b). Diese Form wurde durch Ellipsen approximiert (c).

Bedingt durch die Form der Zellkerne ist ihr Schwerpunkt nicht in der Mitte, sondern stark zu der dem Nukleolus abgewandten Seite hin verschoben. Dieser Effekt ist auch gut an den Ellipsen zu sehen: sowohl die oben als auch die unten gezeigte Ellipse liegen mit ihren Schwerpunkt auf der dem Nukleolus abgewandten Seite.

Ein vollständig anderer Ansatz beruht auf dem Konzept der sogenannten „Aktiven Kontur“, einer Klasse von Algorithmen, die bildhaft auch als „Snake-Kontouring“-Algorithmen bezeichnet werden, da sich die Kontur vergleichbar einer Schlange um das Segment herum zuzieht [7, 8].

Da in diesem Ansatz jedoch auch eine Gradienteninformation enthalten ist, lässt sich der Zellkern zwar geringfügig besser approximieren; jedoch macht sich der Nukleolus immer noch als Einbuchtung bemerkbar (siehe Abbildung 7.3).

Ein recht guter Lösungsansatz ist das folgende vergleichsweise einfache Verfahren, das jedoch nur mittels des Vorwissens um die Form eines Zellkerns zu implementieren ist. Das Verfahren unterscheidet sich von den beiden zuvor genannten, da es weder die Gradienteninformation noch die Fläche als Information heranzieht, sondern lediglich auf der Kontur des Zellkerns basiert.

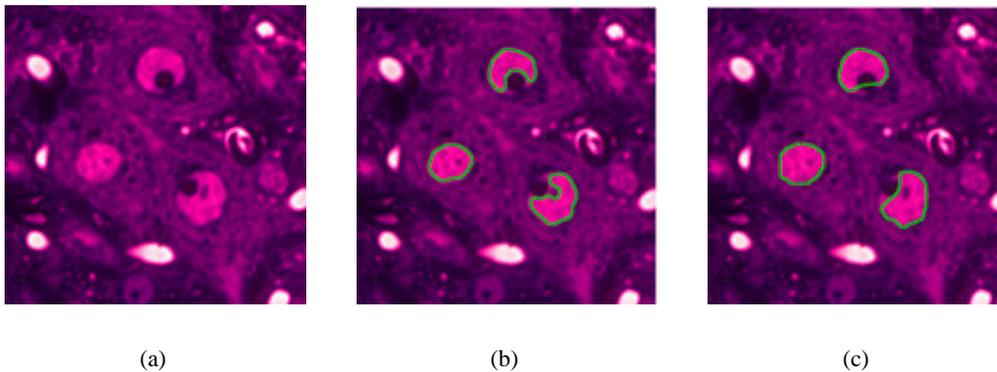


Abbildung 7.3: Aus dem in (a) gezeigte Originalbild wurden mit den zuvor beschriebenen Algorithmus die Zellkerne extrahiert (b). Diese Formen wurde in (c) mittels aktiver Konturen nachbearbeitet.

Ziel dieser Kreisapproximation ist es, diejenigen Bereiche der Kontur zu entfernen, die zu sehr von einer Kreisform abweichen. Dabei stellt sich das Problem, eine Referenzgröße zu definieren, um nicht den Fehler des zuerst vorgestellten Verfahrens zu wiederholen.

Als Referenz wurde hier nicht der Schwerpunkt der Fläche, sondern die Kontur selbst verwendet. Dazu werden die beiden Punkte entlang der Kontur extrahiert, die am weitesten voneinander entfernt sind. Die Strecke zwischen den beiden Punkten wird als Hauptachse, der Mittelpunkt dieser Strecke als Mittelpunkt des Zellkerns definiert. Dieser Punkt entspricht normalerweise sehr gut dem tatsächlichen Mittelpunkt des Zellkerns (vergleiche Kap. 6.3).

Ausgehend von der Annahme, dass ein Zellkern normalerweise näherungsweise eine Kugelform haben sollte, müssten nun sämtliche Punkte entlang der Kontur gleich weit vom Mittelpunkt entfernt sein.

Deshalb wird in einem nächsten Schritt für alle Punkte entlang der Kontur ihr Abstand zum Mittelpunkt bestimmt. Liegt dieser Wert um mehr als 50% unterhalb des maximalen Radius wird der entsprechende Punkt auf der Kontur durch den maximalen Radius ersetzt.

Durch dieses Verfahren ergibt sich in den meisten Fällen eine Kontur, die dem tatsächlichen Zellkern sehr ähnlich ist (Abbildung 7.4).

Leider ist das hier gezeigte Verfahren nicht immer erfolgreich. Gerade in jenen Fällen, in denen eine Nachbearbeitung dringend notwendig wäre – so z.B. bei Zellkernen, bei denen auch noch die direkt am Kern liegenden Bereiche fälschlicherweise dem Zellkern zugerechnet wurden – versagt das Verfahren. Die dadurch bedingte Fehlerrate ist für den praktischen Einsatz noch zu groß.

Es wird sicherlich notwendig sein, zum einen die davor und dahinter liegenden Ebenen mit einzubeziehen und andererseits die hier vorgestellten Verfahren der Kreisapproximation und der Aktiven Konturen zu kombinieren, um wirklich in allen Fällen eine zufriedenstellende Kontur zu bekommen.

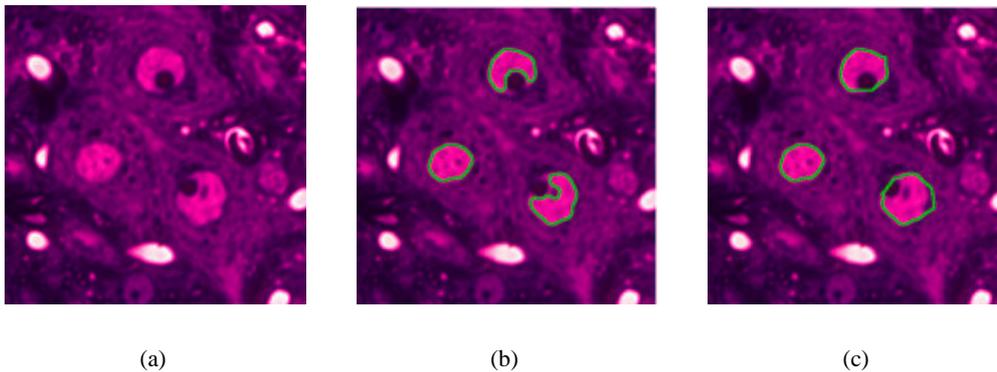


Abbildung 7.4: Aus dem in (a) gezeigte Originalbild wurden mit den zuvor beschriebenen Algorithmus die Zellkerne extrahiert (b). Diese Formen wurde in (c) mittels Kreisapproximation nachbearbeitet.

7.3 Detektion der Somatagrenzen

Die Detektion der Zellkerne und ihre Anordnung im dreidimensionalen Raum ist sehr wichtig und normalerweise bereits ausreichend, um weitreichende Aussagen über den Aufbau und die Struktur des Kerngebietes zu machen. Es kann jedoch in einigen Fällen sinnvoll sein, nicht nur die Zellkerne, sondern das gesamte Soma zu betrachten. Ich habe ein Verfahren entwickelt, das mittels einer einfachen Heuristik eine wahrscheinliche Form für die Zelle extrahiert.

Das hier vorgestellte Verfahren erlaubt nur eine grobe Annäherung an die tatsächliche Form der Zelle und bezieht die Informationen der folgenden und dahinter liegenden Schnitte noch nicht mit ein und ist somit auch noch nicht in der Lage, das Soma in den Bereichen, in denen der Zellkern nicht mehr angeschnitten ist, zu erkennen.

Das Verfahren ist schon jetzt vielen kommerziellen Lösungen für die Zellerkennung überlegen; dennoch ist zum einen die Erkennung im dreidimensionalen Raum noch nicht gegeben und zum anderen funktionieren die der Heuristik zu Grunde liegenden Regeln zwar relativ gut; diese müssen allerdings nochmals an einer größeren Anzahl von Proben validiert werden, um ihre Allgemeingültigkeit zu beweisen.

Das hier vorgestellte Verfahren ist schnell, vergleichsweise robust und durch Parameter an die gegebene Zellpopulation anzupassen. Es basiert im wesentlichen auf einer Heuristik, die auf den im folgenden beschriebenen Voraussetzungen beruht.

Die einzelnen Regeln der Heuristik sind in Abbildung 7.5 auf der folgenden Seite dargestellt. Zunächst wird die Größe des Somas festgelegt. Das Soma ist normalerweise zumindest doppelt so groß und maximal viermal so groß wie der Zellkern. Dies ist in (b) angedeutet.

Im nächsten Schritt wird die von mir gemachte Beobachtung ausgenutzt, dass sich das Soma praktisch nie um eine Kapillare herum legt. Somit bilden die Kapillaren (ebenso wie die anderen Zellkerne) ebenfalls eine Grenze (c).

Innerhalb dieser Grenze wird dann in einer durch einen einfachen Threshold bina-

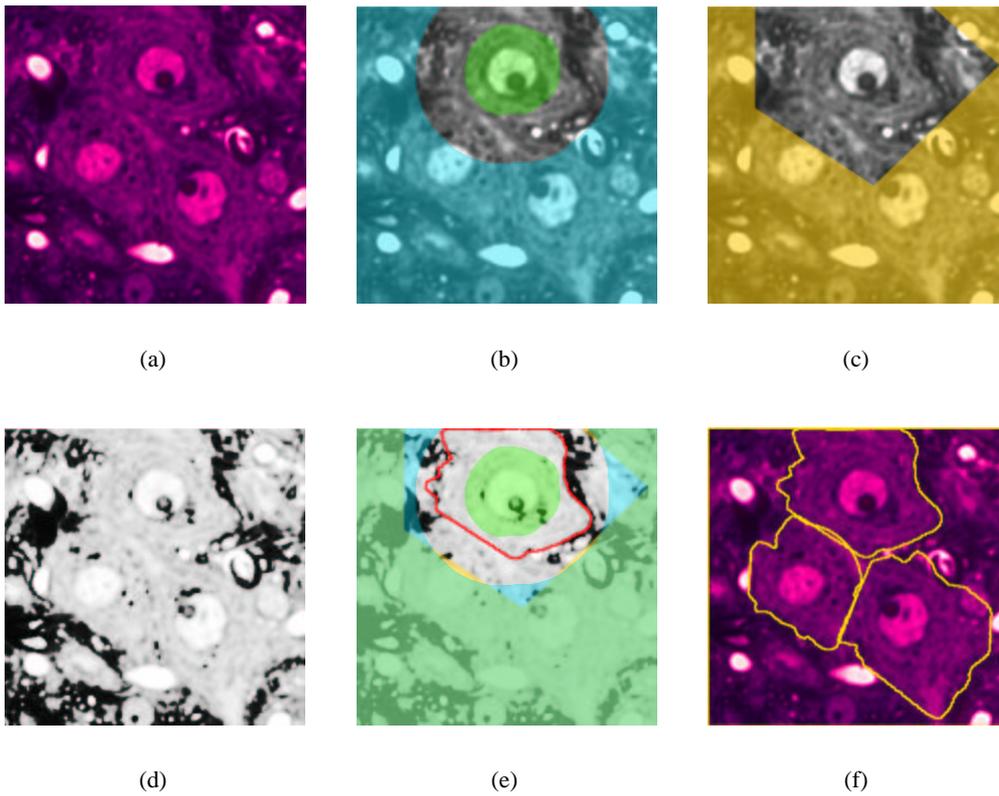


Abbildung 7.5: Einzelschritte zur Zellerkennung: Aus dem in (a) gezeigten Original-Bild soll die obere Zelle extrahiert werden. Dazu wird zunächst eine minimale und maximale Größe festgelegt (b), dann wird eine Begrenzung durch andere Zellkerne zu Kapillaren konstruiert (c), die dann zusammen mit der Kontrastinformation (d) eine gute Annäherung an die tatsächliche Kontur darstellt (e). In diesem Beispiel wurde die Konturen aller drei Zellen korrekt erkannt (f).

risierten Version des roten Kanals (d) nach einem möglichst starken Kontrast gesucht. Wenn ein solcher Grenzbereich nicht gefunden wird, ist die Begrenzung entweder durch die maximale Größe oder durch eine Kapillare gegeben.

Liegen zwei Zellen direkt aneinander, werden sie sich häufig überlappen. Dies wird einfach korrigiert, indem die Mitte zwischen zwei Zellen als Grenzlinie verwendet wird.

Das Ergebnis wird dann nochmal geglättet, indem von mehreren nebeneinander liegenden Punkten die Entfernung zum Kern berechnet und davon der Mittelwert gebildet wird. Die so entstandenen Formen (f) sind eine sehr gute Annäherung an die vom menschlichen Betrachter wahrgenommene Form.

Es bedarf allerdings noch umfangreicher Untersuchungen, um dieses Verfahren im dreidimensionalen Raum in der Praxis anzuwenden.

7.3.1 Entwicklung einer grafischen Bedienoberfläche

Neben dem Ausbau der Funktionen sollte ein Ziel der Weiterentwicklung der Software auch darin bestehen, die Anwendungsfreundlichkeit zu verbessern.

Insbesondere um es auch anderen Gruppen zu ermöglichen, ihre eigenen Analysen mit diesem System durchzuführen; aber auch um für unsere Gruppe die Kontrolle und die Steuerung des Programmablaufs zu erleichtern, wäre die Entwicklung eines grafischen Webfrontends sowie eines Verfahrens zur automatischen Parameteradaptation hilfreich.

Das Verfahren zur automatischen Parameteradaptation ist für einige Schritte bereits implementiert; für andere muss dies noch nachgeholt werden. Das Verfahren beruht darauf, dass der Nutzer ein geeignetes Bild und die in diesem zu lösende Aufgabe manuell vorgibt; etwa indem er die zu erkennenden Bereiche markiert.

Nun gibt es für die einzelnen Verfahren unterschiedliche Parameter, die in ihrer Kombination für den Erfolg des Verfahrens verantwortlich sind. Solange es sich nur um wenige Parameter handelt, ist es möglich, alle Kombinationen zu testen; bei komplexeren Berechnungen ist das hingegen nicht mehr möglich. Für diese Aufgabe kann man einen Evolutionsalgorithmus [80] verwenden, der die optimale Kombination von Parametern in relativ kurzer Zeit bestimmt.

Einige der vorgestellten Verfahren sind bereits mit einer grafischen Oberfläche versehen, für andere steht dieser Schritt noch aus. Es ist für die Zukunft wichtig, alle Programme unter eine einheitlichen Oberfläche zusammenzufassen und so dem Benutzer eine optimale Kontrolle über die Ausführung des Verfahren zu geben.

Dabei bietet es sich an, die Bedienung und die Kontrolle des gesamten Ablaufs unter einer Web-Oberfläche zusammenzufassen. Eine solche Oberfläche hätte den Vorteil, dass die Einstellungen von jedem Computer und absolut unabhängig von Betriebssystem und Hardware vorgenommen werden könnten.

Der Webserver wie auch die eigentliche Programmausführung würde dann auf einem Cluster laufen (siehe Abschnitt 7.4). Der Benutzer kann sich dann über den Internetbrowser am Cluster anmelden, seine Einstellungen tätigen und eine Berechnung starten. Wenn diese dann beendet ist, benachrichtigt ihn der Server per Email und er kann das Resultat ebenfalls im Webbrowser betrachten.

Wie auch für die zuvor besprochenen Erweiterungen existieren auch für dieses Problem bereits einige Ansätze, die bisher einfach deshalb nicht weiter verfolgt wurden, da die Integration der Programme in eine einheitliche Oberfläche nur dann Sinn macht, wenn auch Außenstehende das Programm benutzen möchten.

7.4 Apparativer Ausbau und Erweiterung der Histologie

Momentan hat das Verfahren vor allem einen entscheidenden Nachteil: Es ist zeitaufwendig, eine Rekonstruktion zu erstellen, wobei vor allem zwei Faktoren eine entscheidende Rolle spielen: Zum einen die manuelle Aufnahme mit dem Mikroskop und zum anderen die langwierige Berechnung.

Im diesem Abschnitt soll eine Lösung für die beiden Probleme sowie eine Möglichkeit der Erweiterung des Verfahrens vorgestellt werden. Die Lösung für das Problem der Rechenzeit ist sehr einfach: Man benötigt lediglich mehrere schnellere Computer. Durch einen sogenannten Cluster von Computern ist es möglich, die Rechenkapazität preiswert zu erhöhen und die benötigte Zeit somit deutlich zu verringern. Die möglichen Lösungen für einen Cluster werden im letzten der drei folgenden Abschnitte besprochen.

Schwieriger ist es, die für die Mikroskopie benötigte Zeit zu verringern; doch gerade diese ist ein großes Problem. Bei der Analyse der MSO ist davon auszugehen, dass es nach einer Einarbeitung in die Benutzung des Mikroskops möglich sein sollte, innerhalb von drei bis vier Wochen konzentrierter Arbeit eine komplette Serie aufzunehmen. Wenn man nun bedenkt, dass die MSO ein relativ kleiner Nukleus ist und man für die Analyse einer Struktur immer zumindest drei bis fünf Individuen benötigt, um auf abstrakte Organisationsmerkmale zu schließen, wird schnell klar, warum hier eine Verbesserung hilfreich wäre.

Ich werde im zweiten Teil dieses Abschnittes einen Ansatz zum Automatisieren der mikroskopischen Aufnahmen vorstellen. Technisch ist ein solcher Ansatz problemlos zu bewältigen und viele der dazu benötigten Verfahren existieren bereits. Ein automatisch ablaufendes Verfahren hätte vor allem den Vorteil, dass es „skalierbar“ ist. Das bedeutet, dass bei einer größeren Anzahl von Schnitten jederzeit weitere Mikroskope aufgebaut und für das Verfahren verwendet werden könnten. Dies ist ein großer Vorteil gegenüber der manuellen Ausführung. Außerdem sind automatische Systeme unter Einbeziehung der Personalkosten günstiger als die manuelle Ausführung, können 24 Stunden am Tag und 7 Tage die Woche arbeiten und von den Gründen der Effizienz einmal ganz abgesehen, ist das stunden- und tagelange Auswechseln und Aufnehmen von Schnitten eine Arbeit, die man niemandem zumuten möchte.

Im nächsten Unterabschnitt möchte ich jedoch zunächst noch einen anderen Punkt ansprechen, an dem ebenfalls einige Erweiterungen geplant sind. Bei der Histologie ist zwar die Nissl-Färbung ausreichend, um Zellkern und Soma zu erkennen; allerdings ist es natürlich von größtem Interesse, weitere Daten in das Modell zu integrieren. So wäre es sehr interessant, einerseits mit Tracern und andererseits mit Antikörperfärbungen die Ein- und Ausgänge sowie die Rezeptorpopulation einer Zelle genau zu beschreiben. Einen in diese Richtung weisenden Ansatz möchte ich im folgenden Abschnitt vorstellen.

7.4.1 Integration von weiterem Datenmaterial

Um weitere Daten über die Zellen innerhalb eines Nukleus zu erhalten ist es notwendig, eine umfangreiche Datenbank der Eigenschaften der Zellen in der MSO aufzubauen. Hierzu muss die Methode so erweitert werden, dass es nicht nur möglich wird, die Zellen zu rekonstruieren und ihre Form, Größe und Anordnung zu bestimmen, sondern außerdem zusätzliche Informationen über die Eingänge, die Projektionsgebiete sowie die Verteilung von Rezeptoren zu erhalten.

Zum einen kann ein zusätzlicher Farbstoff verwendet werden, um mit Hilfe von Tracern festzustellen, von welchen Bereichen ein Neuron seine Eingänge erhält und zum anderen, zu welchen Bereichen ein Neuron projiziert.

Hierbei wäre insbesondere die Kombination von zwei unterschiedlich gefärbten Tracern von Interesse. Mit einer solchen Kombination wäre es z.B. möglich, alle Neurone zu identifizieren, die von zwei Nuklei einen Input erhalten oder solche, die von einem bestimmten Nukleus einen Input erhalten und diesen an einen anderen Nukleus weiter leiten.

Zwar wird man bei der Färbung auf Grund der geringen Schnittdicke und dem damit verbundenen starken Photobleaching sehr wahrscheinlich auf die Anwendung von Fluoreszenzmarkierungen verzichten müssen, allerdings sollte es auch mit lichtstabilen Farbstoffen wie z.B. DAB möglich sein, das Verfahren umzusetzen.

Des weiteren kann mit Hilfe von Antikörpern das Vorhandensein von bestimmten Rezeptoren oder Oberflächenproteinen nachgewiesen werden. Diese Methode kann natürlich auch mit der Anwendung eines Tracers verbunden werden, so dass es möglich ist, Zellen zu identifizieren, die von einem bestimmten Bereich einen Eingang erhalten und zudem z.B. einen inhibitorischen Rezeptortyp exprimieren.

Theoretisch wäre auch die Verwendung von transgenen Tieren denkbar, bei denen z.B. ein bestimmter Zelltyp mit GFP (green fluorescent protein) oder einem anderen Marker gekennzeichnet ist.

Außerdem besteht die Möglichkeit, alle Daten, die sich aus einer traditionellen histologischen Untersuchung des Areals ergeben, mittels des *CellInspectors* mit den Zellen zu verbinden. Alle Analysen, die heute von einem Menschen im Laborbuch festgehalten werden, können so in die Datenbank einfließen und das Modell erweitern.

Dadurch, dass immer mehrere Merkmale erfasst werden können, wird es – insbesondere bei einem Nukleus wie der MSO, der eine starke räumliche Gliederung aufweist – möglich sein, die Daten aus mehreren Individuen zu verbinden und in eine Datenbank zu integrieren.

Auf diese Weise wäre es möglich, eine Datenbank zu erstellen, die es in dieser Präzision und Genauigkeit auf dem Gebiet der Neurobiologie noch nie gegeben hat. Die weiteren Möglichkeiten der diskutierten Techniken werden in Kapitel 7.5 auf Seite 133 ausführlich diskutiert.

7.4.2 Erweiterung der Mikroskopie

Ein wesentlicher Punkt, an dem weitere Verbesserungen ansetzen könnten, ist die Aufnahme der Bilder mit dem Mikroskop. In diesem Bereich gibt es im Wesentlichen zwei

Möglichkeiten das Verfahren zu verbessern. Zum einen könnte eine bessere Integration der Algorithmen in die Software der Kamera die Auswertung effektivieren und zum anderen wäre es möglich, durch eine weitgehende Automatisierung der Hardware des Mikroskops einen deutlichen Geschwindigkeitszuwachs zu erreichen.

Die Möglichkeiten einer Integration der API (Application Programming Interface) der Mikroskop-Kamera in eine eigene Software wurden bereits oben ausführlich diskutiert. Dieser Abschnitt beschäftigt sich deshalb in erster Linie mit den Möglichkeiten, die ein Aus- und Umbau des Mikroskops selber bietet.

Das Mikroskopieren ist ein sehr zeitaufwendiger Vorgang. Für die Aufnahme eines kompletten Nukleus sind mit dem momentanen Setup bei einem erfahrenen Benutzer ca. vier Wochen für eine typische Serie einzuplanen. Dieser Zeitaufwand ist zwar für einige wenige Aufnahmen vertretbar; will man die Techniken jedoch auf eine größere Anzahl von Schnitten eines Nukleus mit größerer Ausdehnung anwenden, stößt die momentan verwendete Technik schnell an die Grenze dessen, was zeitlich noch vertretbar ist.

Eine Lösung wäre hier eine Automatisierung des Aufnahmeprozesses. Es ist sehr unwahrscheinlich, dass dies eine deutliche Beschleunigung mit sich brächte, allerdings hätte dieses Verfahren zum einen den Vorteil, dass es unüberwacht arbeiten könnte und zum anderen wäre ein solcher Prozess einfach zu „skalieren“, d.h. man könnte mit vergleichsweise wenig Aufwand mehrere Mikroskope parallel verwenden.

Im folgenden will ich den möglichen Aufbau eines solchen vollautomatischen Mikroskops skizzieren.

Das erste Problem bei einem vollautomatischen Mikroskop ist die Notwendigkeit, die einzelnen Schnitte automatisch in das Mikroskop einzulegen. Dies ist mit einem vergleichbar geringen Aufwand zu bewältigen. So könnte man z.B. jeden Objektträger mit einem schmalen Metallstreifen versehen. An diesem könnte dann ein Elektromagnet angreifen, um den Objektträger von einem Rondell, Fließband oder Kasten auf den xy-Tisch zu heben. Dort könnte der Objektträger ebenfalls magnetisch fixiert werden.

Das Mikroskop selbst würde komplett von einem Computer gesteuert, der die gesamte Mechanik über Elektromotoren bedient. Vergleichbare Projekte existieren bereits – etwa mit dem ferngesteuerten Mikroskop des MIT [60] oder dem motorisierten Mikroskop des Instituts für Medizintechnik und Biophysik in Karlsruhe [10]. Die dazu notwendige Steuerung kann entweder selbst entworfen oder teilweise (wie etwa ein motorisierter XY-Tisch) vom Hersteller des Mikroskops bezogen werden.

Die Software müsste den Schnitt zunächst mit einer etwas kleineren Vergrößerung betrachten, um den Bereich aufzufinden, in dem der Nukleus liegt. Dazu muss der Schnitt zuerst fokussiert werden, was durch eine Optimierung des hochfrequenten Anteils der Fourierfunktion des Bildes [12] leicht möglich ist.

Danach kann entweder durch die relative Lage des Nukleus oder mit einer Klassifikation, wie sie etwa auch in vielen Bereichen der medizinischen Bildverarbeitung verwendet wird, festgestellt werden, in welchem Bereich der Nukleus zu finden ist. Dieser wird dann mit einer höheren Vergrößerung aufgenommen und gespeichert.

Nachdem der Nukleus fertig aufgenommen wurde, kann der Schnitt auf die gleiche Art und Weise wie er auf den xy-Tisch transportiert wurde, wieder zurückgelegt werden.

Ich möchte mit dieser groben Beschreibung vor allem aufzeigen, dass eine Automatisierung dieses Schrittes in jedem Fall möglich ist. Stünde ein vollautomatisches Mikroskop zur Verfügung, so könnte man damit routinemässig Gehirnareale kartieren und katalogisieren. Ein solches Verfahren würde sich auch für die medizinische Diagnose anbieten.

7.4.3 Berechnung im Cluster

Ich möchte an dieser Stelle nicht auf die konkrete Umsetzung eines Clusters eingehen, da es für diese Aufgabe gute, standardisierte Ansätze gibt. Ich will nur kurz zwei Möglichkeiten diskutieren, um damit aufzuzeigen, dass die von dem Verfahren benötigte Rechenkapazität einfach zur Verfügung gestellt werden könnte.

Es muss eine Lösung gefunden werden, die eine einfache Ausführung der Programme in einem Cluster ermöglicht. Hierzu existieren zwei unterschiedliche Lösungsansätze: Zum einen ist es möglich, die Lastverteilung automatisch mittels der Software Mosix [73] zu bewältigen; eine Alternative bestünde in der Verwendung eines Batch-Systems, das die einzelnen Verfahrensschritte auf mehrere Computer verteilt.

Da diese Verfahren in einem kleinen Maßstab bereits funktionieren und auf Standardsoftware basieren, sollte die Umsetzung kein Problem darstellen – zumal die hier beschriebenen Probleme sehr einfach zu parallelisieren sind, da die Bearbeitung der einzelnen Schnittebenen unabhängig voneinander geschehen kann.

Ein solcher Cluster könnte etwa aus ca. 10 Standard-Computern, die die notwendige Rechenkapazität zur Verfügung stellen, sowie einem RAID (Redundant Array of Inexpensive Disks)-Array, welches an einen weiteren Computer angeschlossen ist und die benötigte Speicherkapazität für die Bildserien zur Verfügung stellt, bestehen.

Mit diesem Cluster wäre die Berechnung einer Serie innerhalb weniger Tage möglich.

7.5 Ausblick

In diesem letzten Abschnitt möchte ich die möglichen Anwendungen der automatischen Rekonstruktion diskutieren. Im Vordergrund steht dabei die Fragestellung, die zu dieser Arbeit geführt hat, nämlich die Vorarbeiten für eine Modellvalidierung und die Konstruktion eines neuen Modells für die Funktionsweise der MSO.

Allerdings ist das Verfahren auch in vielen anderen Bereichen anwendbar und könnte nicht nur in der Biologie, sondern auch in der Medizin interessante Ergebnisse hervorbringen. Im zweiten Teil dieses Abschnitts werde ich einige mögliche Anwendungen der hier vorgestellten Methode in unterschiedlichen Bereichen aufzeigen.

7.5.1 Möglichkeiten zur Modellvalidierung

In der Neurobiologie wird der Begriff Modell zur Zeit auf zwei unterschiedliche Arten verwendet. Auf der einen Seite bemühen sich viele Gruppen, Modelle für die Verarbeitung von Informationen durch größere Komplexe zu erstellen. Bei diesen Modellen

handelt es sich um *Modellvorstellungen*, wie eine bestimmte Aufgabe bewältigt werden könnte. Ein solches Modell besteht normalerweise – wenn es überhaupt digital umgesetzt wird – aus einfachen Punkt- oder Integrate-and-Fire-Neuronen und weist keinerlei Bezug zu den realen Gegebenheiten im Gehirn auf [25].

Demgegenüber stehen echte *Funktionsmodelle*, bei denen einzelne Funktionen am Computer mittels komplexer mathematischer Verfahren nachgebildet werden. Diese Funktionsmodelle sind zumeist derart dimensioniert, dass die Funktionalität eines Zellbestandteils (Kanal, Pore oder Membran), einer einzelnen Zelle oder aber eines Verbunds von wenigen Zellen nachgebildet wird [15].

Beide Modellarten können jedoch die Funktionsweise eines größeren Neuronenverbandes nicht korrekt darstellen. Dies liegt zumeist darin begründet, dass zur Zeit keine Methode existiert, mit der sämtliche Zellen eines Verbandes beschrieben und klassifiziert werden können.

Doch liegt – wie schon in der Einleitung beschrieben – gerade in diesem Bereich der Schlüssel, mit dem wir die Funktionen des Gehirns analysieren und verstehen können. Die enorme Leistungsfähigkeit des Gehirns liegt nicht in der einzelnen Zelle, sondern in deren Verknüpfung und der Interaktion vieler Zellen im Verbund begründet.

Ich möchte im folgenden die Problemstellung, auf Grund derer das vorliegende Verfahren entwickelt wurde, vorstellen und aufzeigen, wie mittels einer dreidimensionalen Rekonstruktion ein echtes Funktionmodell der MSO geschaffen werden könnte und gleichzeitig eine Validierung der bestehenden Modelle ermöglicht würde.

Die Lokalisation einer Schallquelle im Raum kann auf zwei unterschiedliche Arten erfolgen [38]: Während bei hochfrequenten Tönen der Kopf, wie in Abbildung 7.6 links gezeigt, einen Teil des Schalls reflektiert, dadurch ein „Schallschatten“ entsteht und eine Lokalisation auf der Basis der Intensitätsdifferenz zwischen den beiden Ohren (Interaural level difference ILD) stattfindet, ist diese Art der Lokalisation bei tieffrequenten Tönen nicht durchführbar. Wenn die Schwingungslänge groß gegenüber dem Ohrenabstand ist, lässt sich eine Schallquelle nur noch über die zeitliche Differenz (Interaural Time Difference ITD) zwischen den beiden Ohren lokalisieren (Abbildung 7.6 rechts).

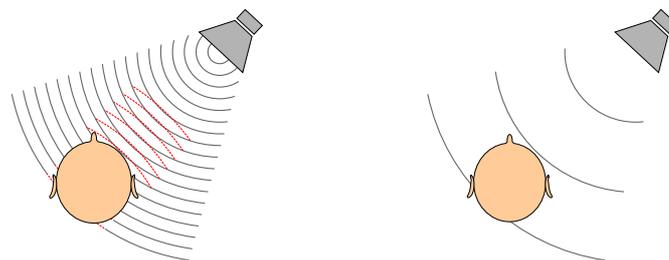


Abbildung 7.6: Schematische Darstellung der unterschiedlichen Prinzipien der Schalldetektion: Links ist die Detektion mittels ILD, rechts mittels ITD dargestellt. (nach [38])

Die erste Station im Hirnstamm, die einen Eingang von beiden Ohren erhält und somit die Möglichkeit einer Verarbeitung hat, ist der Komplex der Superioren Olive. In der Lateralen Superioren Olive (LSO) werden die ILDs ausgewertet. Die anatomischen

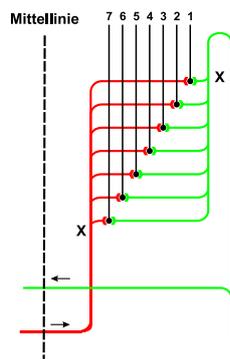


Abbildung 7.7: Schematische Darstellung des Jeffress-Modells für die rechte Seite (nach [54]).

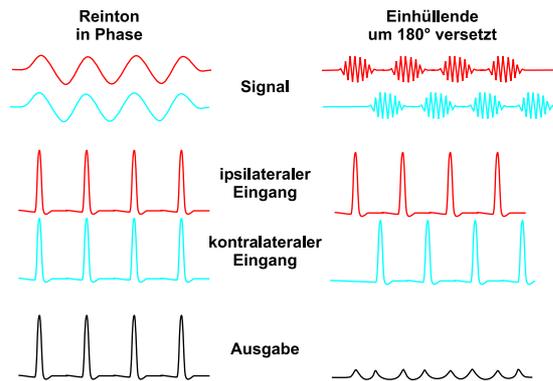


Abbildung 7.8: Darstellung des Phase-Lockings für tief- und hochfrequente Signale (mod. nach [38]).

und physiologischen Eigenschaften der LSO sind bei den Säugern weitestgehend einheitlich und ihre Rolle in der Verarbeitung ist unbestritten [97]. Bei der Verarbeitung der ITDs ist die Situation, wie im folgenden gezeigt, weniger klar.

Das bis heute gültige Modell für die Detektion von ITDs basiert auf einer von Jeffress 1948 formulierten Modellvorstellung der „coincidence detection with delay lines“ [52]. Wie in Abbildung 7.7 dargestellt, beruht das Modell auf binaural innervierten Neuronen (schwarz), die nur dann antworten, wenn von den beiden monoauralen Neuronen (rot und grün) gleichzeitig ein excitatorischer Reiz eintrifft. Bei einer gleich langen Wegstrecke für beide Signale würde es so nur dann zu einer Antwort kommen, wenn sich die Schallquelle frontal vor dem Kopf befindet. In Abbildung 7.7 sind die Weglängen vom ipsi- und contralateralen Ohr an den mit X bezeichneten Punkten gleich; bei einer frontal positionierten Schallquelle würde demnach das Neuron 4 antworten.

Kommt der Schall nun aus dem linken Gesichtsfeld, erreicht er das linke Ohr früher und es kommt am Neuron 4 zu keiner Koinzidenz mehr. Die beiden Signale können nur dann zusammentreffen, wenn das Signal von der linken Seite eine längere Wegstrecke zurücklegen muss als das von der rechten kommende. Dabei entspricht die zusätzliche Wegstrecke exakt der Verzögerung, die durch die unterschiedliche Laufzeit der Schallwelle entstanden ist. Dies könnte z.B. bei Neuron 1 der Fall sein. So wird ein zeitliches Signal in ein räumliches Signal umgewandelt.

Obwohl die MSO in wesentlichen Punkten den Voraussetzungen des Jeffress-Modells entspricht [54], gibt es dennoch viele Unklarheiten, die darauf hindeuten, dass das Modell nicht ausreicht, um die Funktionalität der MSO zu erklären. Im folgenden will ich kurz die Voraussetzungen, auf denen das Jeffress-Modell aufbaut, darstellen und aufzeigen, in wie weit die reale Situation davon abweicht. Das Jeffress-Modell geht von den folgenden Annahmen aus:

- Neurone von beiden Seiten konvergieren in einem binaural innervierten Nucleus.
- Die Neurone codieren mit ihrem Entladungsmuster die Information des akustischen Stimulus.

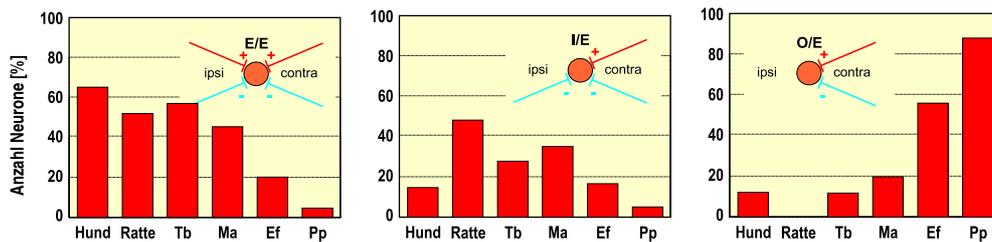


Abbildung 7.9: Verteilung der Projektionsmuster von MSO-Neuronen bei verschiedenen Säugern. Neben den Hund und Ratte sind vier verschiedene Fledermausarten dargestellt (Ef = *Eptesicus fuscus*, Ma = *Molossus ater*, Pp = *Pteronotus parnellii*, Tb = *Tadarida brasiliensis*). (mod. nach [38])

- Die Zellen im binauralen Nukleus antworten nur auf eine gleichzeitige excitatorische Erregung der beiden monoauralen Neurone.
- Die monoauralen Neurone verzweigen sich in einem leiterförmigen Muster mit unterschiedlichen Laufzeiten
- Diese Anordnung muss für jede Frequenz jeweils getrennt vorliegen, da nur identische Wellenlängen verglichen werden können.

Die erste Forderung, dass die MSO binaural innerviert werden soll, wird bei den meisten Säugern erfüllt. Bei der Schnurrbart-Fledermaus *Pteronotus p. parnellii* jedoch sind die ipsilateralen Eingänge der MSO so sehr reduziert, dass die MSO bei diesen Tieren ein fast vollständig monoauraler Nukleus ist [20]. Außerdem sind diese Tiere nicht in der Lage, Frequenzen unter 10 kHz zu hören, haben aber dennoch eine sehr große MSO. In diesem Spezialfall ist also bereits die Grundvoraussetzung des Jeffress-Modells nicht erfüllt; bei den meisten anderen Säugern ist die MSO jedoch binaural innerviert.

Diese von den beiden Seiten kommenden monoauralen Neurone konvergieren in der MSO. Das Jeffress-Modell sieht vor, dass diese Neurone durch das sogenannte Phase-Locking die Frequenz des akustischen Stimulus übertragen. Dabei entspricht die Entladungsrate der Neurone der Frequenz des akustischen Signals und gibt so die Information über dessen Frequenz weiter. Dieses Phänomen wurde auch für Zellen der MSO gezeigt [21]. Diese phasengebundene Antwort tritt bei Frequenzen unterhalb von ca. 3 kHz auf, was dem Jeffress-Modell entspricht (das phasengebundene Antwortverhalten ist im linken Teil der Abbildung 7.8 dargestellt).

Entgegen den Annahmen von Jeffress reagieren die Neurone jedoch nicht nur auf tieffrequente Töne, sondern auch auf Töne im hochfrequenten Bereich [55]. Dort kann aus physiologischen Gründen kein Phase-Locking zur Frequenz des Tones stattfinden, vielmehr entspricht dort die Entladungsrate des Neurons, wie in Abbildung 7.8 rechts gezeigt, der Einhüllenden des Stimulus. Dieses Verhalten ist im Jeffress-Modell nicht vorgesehen. Es konnte jedoch gezeigt werden, dass die Neurone der MSO auf eine solche sinusförmige Amplitudenmodifikation ähnlich reagieren wie auf einen „normalen“ tieffrequenten Reiz [41].

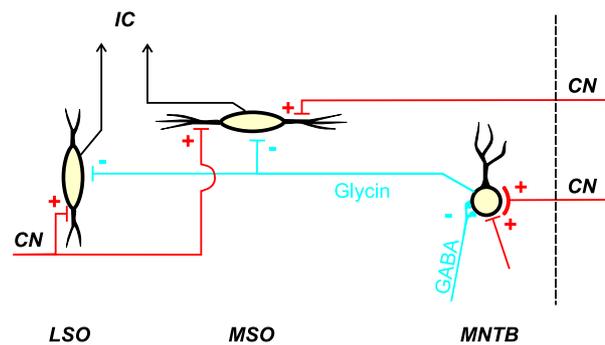


Abbildung 7.10: Das hier gezeigte Schema zeigt eine mögliche Verschaltung der Neurone in der MSO, basierend auf immunocytochemischen Untersuchungen (nach [1])

Der nächste Widerspruch zum Jeffress-Modell findet sich bei der Innervierung der MSO-Neurone. Für einen Koinzidenz-Detektor nach Jeffress würden excitatorische Eingänge genügen und lange Zeit ging man auch davon aus, dass alle Neurone der MSO so genannte E/E-Neurone sind, die zwei excitatorische Eingänge erhalten. Es wurde mittlerweile sowohl anatomisch als auch physiologisch gezeigt, dass die Neurone der MSO glycinerge, d.h. inhibitorische Eingänge von beiden Seiten erhalten [1, 15, 37, 39, 40, 42, 56, 69]. In vivo- und in vitro-Untersuchungen an Wüstenrennmäusen und Fledermäusen haben gezeigt, dass die Mehrzahl der Neurone inhibitorische Eingänge von beiden Seiten erhält [42]. Diese Neurone werden als EI/EI-Neurone bezeichnet. Ein geringerer Anteil der Neurone erhält nur von einer Seite einen inhibitorischen Eingang (I/EI-Neurone) und ein sehr geringer Teil der Neurone ist monoaural innerviert. Die Anzahl der Neurone mit unterschiedlichem Antwortverhalten ist für verschiedene Tiere in Abbildung 7.9 auf der vorherigen Seite dargestellt. Man kann sehen, dass die Anzahl inhibitorischer Eingänge bei kleineren Säugern größer ist, als etwa beim Hund; dennoch zeigen auch beim Hund 36% der Neurone keine binaurale Excitation und 12% sind strikt monoaural [33].

Es konnte außerdem gezeigt werden, dass der vom medialen Nucleus des Trapezkörpers (MNTB) kommende inhibitorische Input nicht einfach eine diffuse Inhibition ist, die die MSO von einer Antwort auf intensive monoaurale Reize abhält, sondern vielmehr die präziseste bekannte Inhibition darstellt [91]. Dies legt nahe, dass die inhibitorischen Eingänge im Rahmen der ITD-Verarbeitung eine wichtige Rolle spielen und ein komplexes Verschaltungsmuster, wie etwa in Abbildung 7.10 dargestellt, vorliegt. Die dort gezeigte Verschaltung basiert auf immunocytochemischen Untersuchungen von Adams und Mugnaini [1] und stellt ein grobes Schema der bereits sicher nachgewiesenen Verbindungen bei der Katze dar. Dieses Schema muss sicherlich noch deutlich erweitert und ergänzt werden.

Die bisher aufgezeigten Probleme beschäftigen sich vorwiegend mit den elektro-physiologischen Eigenschaften der MSO; allerdings sind auch die anatomischen und morphologischen Voraussetzungen des Jeffress-Modells in der Realität nur teilweise gegeben.

Eine grundlegende Voraussetzung des Jeffress-Modell ist eine Verzögerungsstrecke, die bestimmt, bis zu welcher Grenze es möglich ist, eine ITD noch zu detektieren.

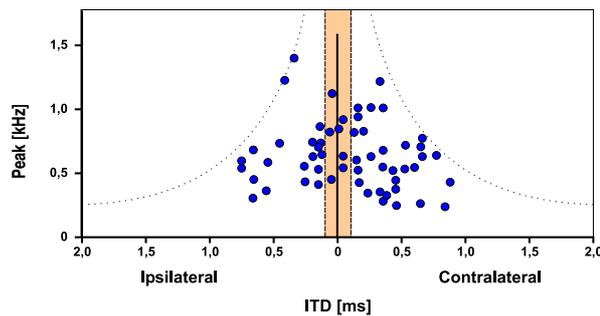


Abbildung 7.11: Darstellung von detektierbaren ITDs bei unterschiedlichen Frequenzen bei der Merriam-Kängururatte. Die maximale ITD ist als gepunktete Linie dargestellt; der theoretisch auf Basis des Kopfdurchmessers detektierbare Bereich ist farbig unterlegt. (mod. nach [21])

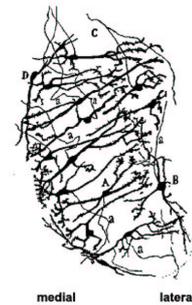


Abbildung 7.12: Golgi-Färbung der rechten MSO. Die bipolaren PCs sind gut zu erkennen. (aus [18])

Diese maximal detektierbare ITD lässt sich auf Basis des Kopfdurchmessers berechnen bzw. messen [44].

Wie in Abb. 7.11 für die Merriam-Kängururatte (*Dipodomys merriami*), ein der mongolischen Wüstenrennmaus sehr ähnliches Tier aus Amerika, gezeigt, findet man in der Realität ITDs, die weit über den theoretisch detektierbaren Bereich hinausgehen. Dies ist meiner Meinung nach der bisher stärkste Beweis dafür, dass der natürliche Mechanismus nicht nur auf dem Jeffress-Modell basieren kann.

Diese Diskrepanz zwischen theoretisch berechneten und praktisch gemessenen ITDs findet man auch beim Kaninchen [30] (sowohl für tiefe Töne als auch für die Einhüllende hochfrequenter Signale) und bei Fledermäusen. Die Sensitivität für ITDs der MSO-Neurone der Guano-Fledermaus *Tadarida brasiliensis* liegt für fast alle getesteten Neurone weit oberhalb von $100 \mu\text{s}$ [41]. Der Kopfdurchmesser dieser Tiere lässt jedoch nur ITDs von maximal $30 \mu\text{s}$ auftreten; d.h. die Neurone sind nicht in der Lage, die durch die Anatomie des Kopfes bedingten ITDs auszuwerten.

Natürlich werden die ITDs bei einem größeren Kopfdurchmesser größer und können besser ausgewertet werden; dennoch sind die Beobachtungen an der Merriam-Kängururatte und der Fledermaus insofern von Bedeutung, als die entwicklungs-geschichtlich frühen Säuger allesamt nur sehr geringe Kopfdurchmesser hatten [40], was den Schluss nahe legt, dass entweder die ursprüngliche MSO eine andere Aufgabe gehabt haben muss oder der Mechanismus der Schalllokalisation effizienter sein muss, als von Jeffress vorhergesagt.

Doch auch bei Tieren mit einem größeren Kopfdurchmesser sind die anatomischen Voraussetzungen nicht unbedingt gegeben. Zwar zeigen Zeichnungen von Ramón y Cajal (Abbildung 7.12), wie auch neuere anatomische Arbeiten [9] eine Anordnung der Neurone, die die Voraussetzungen des Jeffress-Modells in etwa erfüllt, allerdings kommen auch an dieser Anordnung immer mehr Zweifel auf.

Man findet in der MSO neben den von Jeffress geforderten bipolaren Zellen auch andere Zelltypen, die in einem Modell wie dem von Jeffress keine Aufgabe hätten. Mittlerweile haben unterschiedliche Autoren die Zellen auch elektronenmikroskopisch

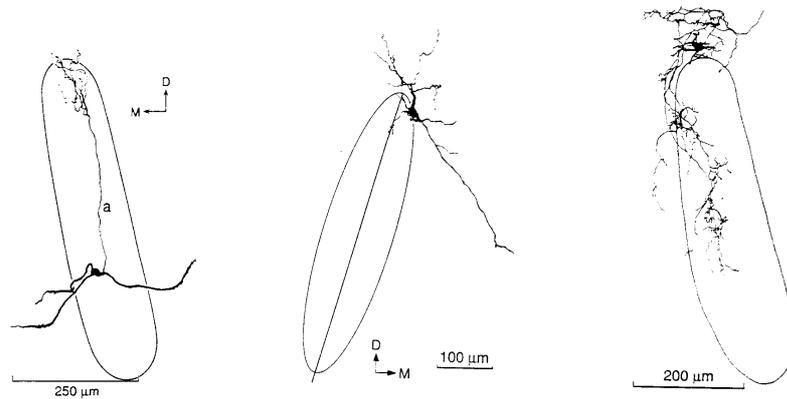


Abbildung 7.13: Typische Beispiele für eine Principal (links), eine Nonprincipal (mitte) und eine Marginal Cell (rechts) (mod. nach [90])

und physiologisch charakterisiert, wobei sich drei Haupttypen herauskristallisiert haben [89]:

- *Principal Cells (PCs)*: Diese Zellen entsprechen den bipolaren Zellen des Jeffress-Modells. Wie in Abb. 7.13 links zu sehen, haben diese Zellen zwei dicke Dendriten, die über 50 bis 150 µm gerade verlaufen und im wesentlichen entsprechen auch die electrophysiologischen Eigenschaften dieser Zellen denen eines Koinzidenzdetektors.
- *Nonprincipal Cells (NPCs)*: Die in der Mitte von Abb. 7.13 zu sehende Zelle ist eine multipolare Zelle, die innerhalb der MSO liegt. Deren Form und Eigenschaften unterscheiden sich deutlich von den bipolaren PCs.
- *Marginal Cells (MCs)*: In der Abbildung 7.13 rechts ist eine Zelle zu sehen, die ebenfalls multipolar aufgebaut ist. Die MCs sind mittelgroß, häufig von dreieckiger Form, haben keine spezifische Ausrichtung und liegen am Rand der MSO. Bisher konnte ihnen ebenso wie den NPCs noch keine schlüssige Funktion zugeordnet werden.

In einigen Punkten erfüllen die anatomischen Gegebenheiten die Voraussetzungen des Jeffress-Modells; das Vorhandensein der NPCs und der MCs hingegen lässt sich auf der Basis des Modells nicht erklären.

Es konnte außerdem gezeigt werden [59, 88], dass sich die NPCs über mehrere Frequenzschichten erstrecken und dort teilweise inhibitorisch wirken. Bisher fehlt eine schlüssige Erklärung für diese Anordnung.

Im Gegensatz zum Nucleus Laminaris (NL), dem bei Vögeln der MSO entsprechenden Nucleus, konnte bei der MSO die Existenz von Verzögerungsstrecken (*delay lines*) nicht einwandfrei nachgewiesen werden. Einige der Verschaltungen entsprechen der Modellvorstellung, während andere – vor allem auf der ipsilateralen Seite – diesem Konzept nicht entsprechen [9]. Auch dies spricht dafür, dass das Modell nur begrenzt mit der Realität übereinstimmt.

Ebenfalls im Gegensatz zum NL existieren für die MSO nur sehr wenige Hinweise auf eine systematische Repräsentation der ITDs entlang der rostro-caudalen Achse, wie es nach Jeffress notwendig wäre, um das zeitliche Muster in einen räumlichen Kode umzuwandeln. Wenn dieser Schritt allerdings nicht stattfindet, ist es schwer vorstellbar, wie in der nächsten Verarbeitungsstufe, dem zentralen Nukleus des Inferioren Colliculus – in der keine systematische Repräsentation der ITDs gefunden wurde – eine Auswertung erfolgen soll.

Mit der hier vorgestellten Methode wäre es möglich, diesen Problemkomplex einer Lösung näher zu bringen. Mittels der funktionalen Rekonstruktion der MSO wird es möglich sein, die anatomischen Voraussetzungen bezüglich Anzahl und Anordnung der Zellen zu validieren und so die Grundlage für alternative Modelle zu schaffen.

Wie in Abb. 7.14 schematisch dargestellt, ist die Voraussetzung für eine Funktionsweise nach Jeffress nicht nur eine Aufspaltung nach ITD-Sensitivität wie in der Mitte der Abbildung durch die monoauralen Neurone angedeutet (entsprechend den Neuronen in Abb. 7.7 auf Seite 135), sondern auch eine Gliederung in sogenannte Frequenzschichten (in der Abbildung übereinander liegend). Das bedeutet, es müssen genügend Neurone vorhanden sein, um einerseits den von dem Tier wahrgenommenen Frequenzraum abzudecken und andererseits eine genügend große ITD-Auflösung zu gewährleisten.

Abschätzungen auf Grundlage der bisherigen Ergebnisse haben ergeben, dass bei der Wüstenrennmaus die MSO – wie bei den meisten Säugern – die Form einer Platte hat und entlang der angenommenen Frequenzverteilung (dorso-ventral) aus ca. 12 Schichten besteht. In jeder dieser Schichten sind in rostro-caudaler Richtung ca. 30 Gruppen von jeweils 3 Neuronen angeordnet.

Im Verhaltensexperiment konnte gezeigt werden, dass die Tiere in der Lage sind, tieffrequente Töne zu lateralisieren [44,57]. In Abb. 7.15 ist deutlich zu sehen, dass die Tiere auch im tieffrequenten Bereich gut in der Lage sind, Töne zu lateralisieren. Damit trifft nicht nur die für die Fledermaus und die Merriam-Kängururatte vorgebrachte Kritik zu, sondern es bietet sich auch die Möglichkeit, die Untersuchungen an einem Säuger vorzunehmen, bei dem im Gegensatz zur Fledermaus die Lokalisation tieffrequente Töne im natürlichen Umfeld eine Bedeutung hat und deshalb wahrscheinlich auch stärker ausgeprägt ist.

Prinzipiell sollten 90 Zellen pro Frequenzschicht ausreichen, um die im Verhaltensversuch festgestellte Winkelauflösung mit einer Verrechnung nach Jeffress zu erreichen. Dabei treten allerdings drei grundlegende Probleme auf:

- Zunächst einmal müssten von den 90 Zellen mindestens 30 Zellen bipolare PC-Zellen sein, da nur diese als Koinzidenzdetektoren fungieren können.
- Diese Zellen müssen excitatorische Eingänge von beiden Seiten erhalten.
- Nach dem Jeffress-Modell muss für jede Frequenz eine eigene Schicht (Isfrequenzschicht) zur Verfügung stehen. Die Tiere müssen in der Lage sein, im tieffrequenten Bereich ca. 3-4 Oktaven zu verarbeiten. Somit würden einer Oktave 3-4 Schichten bzw. 270 bis 380 Neurone zur Verfügung stehen. Dies wäre für eine Verarbeitung selbst dann nicht ausreichend, wenn alle Zellen bipolare PCs wären.

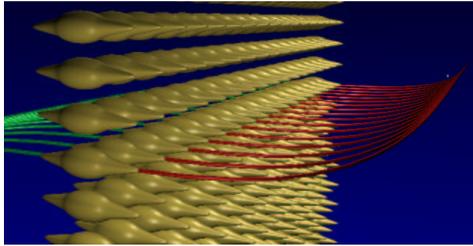


Abbildung 7.14: Schematische Darstellung der Frequenzschichten übereinander mit monoauralen Neuronen nach Jeffress für eine Schicht.

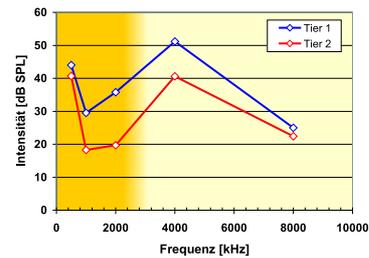


Abbildung 7.15: Fähigkeit der Wüstenrennmaus zur Lateralisation von Tönen bei verschiedenen Frequenzen (nach [57])

- Würden diese geschätzten Zahlen stimmen, würden – da pro Oktave ca. 1000-1200 Typ I Fasern Informationen an die MSO liefern – von jeder Seite her drei bis vier Fasern auf eine MSO-Zelle konvergieren, wofür es bei bisherigen Modellen keine sinnvolle Erklärung gibt.

Zunächst einmal können mit einer funktionalen Rekonstruktion die hier nur grob geschätzten Zahlen exakt bestimmen können. Auf dieser Basis kann dann das Jeffress-Modell validiert werden, indem man die reale Anzahl an Schichten und Neuronen mit der theoretisch benötigten vergleicht.

Wenn man neben den rein auf Nissl-Schnitten basierenden Daten außerdem weitere Daten auf Grundlage der weiter oben geschilderten Erweiterung mit Tracern und Antikörpern hinzunehmen kann, wird es sehr präzise möglich sein, die einzelnen Zellen zu charakterisieren.

Eine mögliche Anwendung von Tracern bei dieser Fragestellung bestünde z.B. darin, mittels zweier Tracer, die in die ipsi- und die contralaterale Cochlea eingebracht werden, jene Zellen zu markieren, die von beiden Seiten einen Eingang erhalten.

Des weitern könnte die gleiche Methode nochmals angewendet werden, wobei jedoch ein Tracer in die ipsilaterale Cochlea und ein anderer in den ipsilateralen Inferioren Colliculus (IC) injiziert wird, um so einen Ein- und einen Ausgang der MSO zu bestimmen.

Käme dann noch eine Färbung mit Antikörpern gegen Glycin und Glutamat hinzu, könnte außerdem gezeigt werden, welche der Zellen inhibitorische und excitatorische Eingänge erhalten.

Einem solchen Funktionsmodell käme die Rolle einer Transferfunktion zu, die auf der Grundlage der vorgegebenen biologischen Daten aus Morphologie, Biochemie, Histologie und Elektrophysiologie ein Funktionsmodell erstellt, dessen Leistungen und Defizite dem realen kognitiven Verhalten des Tieres entsprechen.

Durch die Kenntnis von Anzahl, Anordnung und Lage der Zellen wird es bereits möglich sein, die grundlegenden Annahmen von unterschiedlichen Modellen zu überprüfen und diese somit zu validieren. Gelingt es darüber hinaus, die oben genannten Daten zu integrieren, wird man auch in der Lage sein, existierende Alternativen zum Jeffress-Modell zu prüfen oder eigene Modelle aufzustellen.

Es sollte ferner möglich sein, die Ergebnisse aus unterschiedlichen Individuen miteinander zu kombinieren und zu vergleichen.

Kennt man etwa die in einer Zelle vorhandenen Rezeptoren, so ist es möglich, der Zelle einen elektrophysiologisch gewonnenen Datensatz zuzuordnen und zumindest bei einigen Zellen wird es auf Grund ihrer relativen Lage, der charakteristischen Form des Somas und der restlichen Daten auch möglich sein, dieser eine Einzelzellmorphologie zuzuweisen.

Diese Datenbank würde nicht nur eine gute Basis für die Bildung neuer und die Validierung alter Modelle bieten, sondern auch die Möglichkeit bieten, die Daten verschiedener Gruppen zu erfassen, zu kombinieren und in Einklang zu bringen. Ein solches Modell würde dynamisch mit den Anforderungen wachsen, Unstimmigkeiten offen legen und eine neue, integrative Art der Zusammenarbeit ermöglichen.

7.5.2 Andere Anwendungsgebiete

Neben dem hauptsächlichen Ziel der Modellanalyse wird die Arbeit jedoch – wie oben angesprochen – auch Lösungsmöglichkeiten für einige allgemeinere Fragestellungen bieten und eine gute Grundlage für weitere Untersuchungen darstellen. Einige dieser Analysemöglichkeiten möchte ich hier kurz aufzeigen.

Zuerst erläutere ich kurz die bereits weiter vorne angesprochene Möglichkeit, mit dem Ergebnis der Rekonstruktion eine Clusteranalyse durchzuführen und so die Abgrenzung eines Nukleus zu bestimmen.

Danach folgt ein Abschnitt über die Anwendung des Verfahrens auf unterschiedliche Fragestellungen bezüglich der Ontogenese und dem Vergleich verschiedener Arten.

Im vorletzten Abschnitt gehe ich auf die Möglichkeiten, eine funktionale Rekonstruktion bei der Erforschung degenerativer Krankheiten ein.

Abschließend folgt noch eine kurze Schlußbetrachtung einiger weiterer Anwendungsmöglichkeiten und der dabei besonders hervorzuhebenden Vorteile.

Abgrenzung eines Nukleus

Eine sehr spannende Fragestellung, die sich mit wenig Aufwand beantworten ließe, ist die Frage nach der Abgrenzung eines Nukleus. Auch wenn heute ein Nukleus mit physiologischen und immunochemischen Methoden charakterisiert wird, werden dessen Grenzen nur selten mit mathematischen Modellen festgelegt. Zumeist wird die Entscheidung über die Abgrenzung eines Nukleus von einem Histologen nach dessen Erfahrungswerten getroffen. Da dieser Entscheidung letztlich nur die Form- und Dichtecharakteristika der Zellen zu Grunde liegen, sollte es auch auf Basis der Rekonstruktion möglich sein, mit einer Clusteranalyse im dreidimensionalen Raum die Grenzen der MSO festzustellen. Es wäre sehr interessant, diese Grenze mit der von einem oder mehreren Experten bestimmten Grenzlinie zu vergleichen und vielleicht zu einem objektiven Kriterium für die Abgrenzung eines Nukleus zu gelangen.

Wie in Abbildung 6.6(b) auf Seite 115 gezeigt, wäre es mit einer solchen Analyse sehr einfach möglich, isoliert liegende Neurone farblich zu markieren und auf der Grundlage von Dichtekriterien eine Grenze zu ziehen.

Vergleich unterschiedlicher Tiere und Entwicklungsstadien

Durch die universelle Anwendbarkeit des Verfahrens und die relativ kurze Zeitdauer, die notwendig ist, um eine Rekonstruktion zu erstellen, eröffnet das Verfahren ein breites Spektrum von möglichen Anwendungen:

So wäre es möglich, die MSO von Jungtieren in unterschiedlichen Entwicklungsstadien zu rekonstruieren und mit der des adulten Tieres vergleichen. Auf diese Weise könnte die Entstehung der Anordnung der Neurone in der MSO untersucht werden.

Neben dem Vergleich verschiedener Entwicklungsstadien wäre auch der Vergleich verschiedener Tierarten – etwa der Wüstenrennmaus, der Katze und der Fledermaus – von Interesse, um festzustellen, ob die Grundprinzipien des Aufbaus identisch sind oder in wie fern eine Anpassung stattgefunden hat.

Dabei ist insbesondere die MSO ein interessantes Untersuchungsobjekt, da hier bekannt ist, dass der anatomische Aufbau bei Tieren, die tieffrequenten Schall lokalisieren können und ITDs verwenden, wie etwa der Katze oder dem Hund, die MSO vollständig anders aufgebaut ist als bei Ratten oder Fledermäusen, die nicht in der Lage sind, ITDs zu verarbeiten [38, 56].

So wird davon ausgegangen, dass insbesondere die Ausrichtung der bipolaren Zellen in Form einer sagittalen Platte (z.B. bei der Katze) eine Anpassung an die Notwendigkeit ITDs zu lokalisieren, darstellt [18, 39].

Hierbei ist die Wüstenrennmaus insofern von besonderem Interesse, da sie in der aktuellen Literatur bezüglich ihrer Fähigkeit zur Lokalisation von tieffrequentem Schall weitestgehend mit Hund und Katze auf eine Stufe gestellt wird. Dies beruht vor allem auf den Untersuchungen von Ryan und Heffner [45, 87]; es wurde jedoch meines Wissens nach nie gezeigt, dass die Fähigkeit zur Schallokalisation ähnlich präzise ist wie die der größeren Säuger, noch konnte eine Anordnung der Dendriten in der Form nachgewiesen werden, wie es bei Katze und Hase der Fall ist [18]. Die von Cajal auch in Nissl-Präparaten der Katze [18] gefundene Anordnung der Zellen und die für eine Bipolarzelle typische Spindelform konnte zumindest in dem bisher in dieser Arbeit untersuchten Areal nicht festgestellt werden.

Sollte sich bei anatomischen Untersuchungen herausstellen, dass die Struktur der MSO bei *Meriones ungiuculatus* weder der eines nicht ITD-Nutzer wie der Ratte, noch der eines ITD-Nutzers entspricht, sondern eher einen Zwischentypus darstellt und sollten dieses Ergebnis zudem durch eine schlechte Lokalisationsfähigkeit im Verhaltensversuch bestätigt werden, so würde dies einerseits die These einer direkten Beziehung zwischen Struktur und Funktion dieses Nukleus stärken, andererseits allerdings dazu führen, dass die Rolle, die die Wüstenrennmaus bei der Untersuchung der MSO bisher spielte, neu überdacht werden muss.

Im Zusammenhang mit einer Beobachtung der Ontogenese könnte auch festgestellt werden, ob die Organisationsprinzipien der MSO bei der Wüstenrennmaus in erster Linie auf Erfahrungswerte zurückzuführen sind oder ob diese angeboren sind. Dies wäre z.B. dadurch festzustellen, indem man das Tier in einer Umgebung ohne die entsprechenden Reize, z.B. in weißem Rauschen, aufwachsen lässt.

An solchen Tieren wäre es möglich, angeborene von erlernten Strukturmerkmalen zu unterscheiden und die eventuell auftretenden Degenerationserscheinungen zu quantifizieren.

Da die Untersuchung von Degenerationserscheinungen auch bei vielen Erkrankungen eine große Hilfe darstellen könnte, möchte ich auf dieses Thema im nächsten Abschnitt noch näher eingehen.

Untersuchung degenerative Prozesse

Bei vielen Krankheiten, deren Grundlagen bis heute unverstanden sind, spielen degenerative Prozesse des Nervensystems eine entscheidende Rolle. So sind etwa beim Parkinson- oder Chorea Huntington-Syndrom unterschiedliche degenerative Prozessen dafür verantwortlich, dass es beim Patienten zu einer massiven Störung des motorischen Systems kommt.

Die Untersuchung dieser Krankheiten ist bisher – ebenso wie die Forschung bei Patienten mit Hirntumoren oder Alzheimer – darauf beschränkt, mit vergleichsweise ungenauen Methoden die Ursachen dieser Krankheiten zu beschreiben [3, 24, 66, 99].

Mit der hier vorgestellten Methode wäre es möglich, diese Krankheiten besser zu verstehen und systematisch zu untersuchen. Die bisherigen Untersuchungen auf diesem Bereich haben vor allem das Problem, dass sie mit den in der Einleitung bereits beschriebenen Methoden arbeiten und deshalb naturgemäß die Degeneration entweder nicht oder nur auf zellulärer Ebene untersuchen können – es ist ihnen aber nicht möglich, die Degeneration im Kontext ihres Umfeldes zu untersuchen.

Im folgenden möchte ich kurz aufzeigen, welche Möglichkeiten der Untersuchung die von mir entwickelte Methodik für die unterschiedlichen Krankheitsbilder brächte.

Bei Morbus Parkinson handelt es sich um eine Degeneration dopaminerger Neurone in der Substantia nigra. Dies wiederum führt zu mangelnder Dopaminausschüttung im Striatum, was massive motorische Störungen zur Folge hat (Akinese, Rigor und Tremor) [27].

Die degenerativen Prozesse verlaufen langsam und bei einer frühzeitigen Diagnose können die Symptome des Parkinson-Syndroms mittels L-DOPA über eine lange Zeit unterdrückt werden. Die Ursache der Krankheit ist bisher unbekannt.

Mit der von mir entwickelten Methode würden sich vollkommen neue Möglichkeiten erschließen, den Degenerationsprozess zu untersuchen. Denkbare Fragestellungen wären dabei z.B., ob der Prozess immer in einem bestimmten, lokal begrenzten Gebiet beginnt, oder ob bestimmte Zelltypen weniger anfällig sind als andere. Insbesondere durch die Verwendung von transgenen Tieren, bei denen bestimmte Zellen markiert wurden oder aber durch die Identifikation bestimmter Zellmerkmale mit Antikörpern böte sich hier eine Möglichkeit, das Krankheitsbild besser zu untersuchen.

Während die Ursache der Degeneration beim Parkinson-Syndrom noch weitgehend ungeklärt ist, ist die Ursache der Degeneration beim Chorea Huntington-Syndrom bekannt: Es handelt sich um einen genetischen Defekt des Huntingtin-Genes auf Chromosom 4. Dessen Genprodukt bildet eine Amyloidstruktur aus, die Nervenzellen insbesondere in den Stammganglien und der Hirnrinde zerstört. Chorea Huntington führt immer zum Tod und ist nicht heilbar [48, 74].

Es gibt jedoch Ansätze, diese Krankheit mittels Stammzellen zu behandeln [76]. Auch bei einer solchen Therapie könnte man mit der hier gezeigten Methode neue Erkenntnisse über die Lokalisation, die Integration und letztlich die Funktionsweise der

eingebrachten Stammzellen erhalten, wenn man diese spezifisch markiert und somit im Zellverband identifizieren kann.

Generell bietet sich die Verwendung des Verfahrens immer dann an, wenn in einem strukturierten Gewebe entweder Zellen durch Degeneration oder Nekrose abgebaut werden oder neue Strukturen hinzu kommen .

Letzteres ist z.B. sowohl bei Alzheimer als auch bei einem Tumor der Fall. Während Alzheimer durch die Degeneration von Nervenzellen und die Bildung von senilen Plaques gekennzeichnet ist, kommt es bei einem Tumor zu einer unkontrollierten Zellvermehrung in einem spezifischen Bereich [19].

Durch eine funktionale Rekonstruktion eines betroffenen Bereichs wäre es möglich, die Bildung der Plaques zu verfolgen und deren Entstehung besser zu verstehen. Durch eine Analyse der Wuchsform eines Tumors wäre es ebenfalls möglich, weitere Informationen über dessen Ausbreitung zu erhalten.

Ich bin der Meinung, dass die hier vorgestellte Methode gut dazu verwendet werden könnte, degenerative Prozesse besser zu verstehen und zu deren Heilung beizutragen.

Weitere Anwendungen

Es würde zu weit führen, sämtliche Anwendungsmöglichkeiten der hier entwickelten Methode darzustellen. So wäre es z.B. denkbar, mit den gleichen Prinzipien, die hier auf lichtmikroskopische Aufnahmen angewendet wurden, Serienschnitte im Elektronenmikroskop aneinander auszurichten und gegebenenfalls darin enthaltene Strukturen zu rekonstruieren.

Auch für die konfokale Laserscanning-Mikroskopie würde sich die Methode verwenden lassen. So wäre es etwa möglich, eine Fluoreszenzmarkierung mit einer Gegenfärbung der Zellkerne mit DAPI zu kombinieren und diese zu verwenden, um in den lateralen wie axialen Grenzbereichen den Abbildungsbereich des Mikroskopes zu erweitern.

Ohne auf diese und viele andere denkbare Anwendungen des Verfahrens einzugehen, möchte ich zum Schluss noch zwei Aspekte hervorheben, die ich persönlich für sehr wichtig halte: Objektivität und Integration.

Eine funktionale Rekonstruktion hat zunächst den Vorteil der Objektivität. In vielen Bereichen der Neurobiologie, in denen Strukturen gruppiert und analysiert werden sollen, geschieht dies meist nicht auf der Grundlage von objektiv bestimmbar Maßzahlen.

Insbesondere die Schichtung im Kortex, aber auch kortikale Strukturen wie die Kolumnen des Sehsystems oder die „Barrel“ des Barrel-Kortex geben immer wieder Anlaß zur Diskussion.

Mit einer Rekonstruktion der betreffenden Region könnte man die Frage, wie viele Schichten im Kortex vorhanden sind oder welche Größe eine Kolumne hat überführen in eine Diskussion, auf welche Weise man eine Schicht oder eine Kolumne definiert.

Da bei der Untersuchung eines anatomischen Sachverhaltes zumeist eine bestimmte Hypothese die Grundlage darstellt und der Betrachter somit immer seine subjektive Meinung in das Ergebnis hineinprojiziert, wäre es extrem hilfreich, die subjektive Be-

trachtungsweise durch das zumindest etwas objektivere Bild einer Rekonstruktion zu ersetzen. Diese bietet zwar immer noch Raum für Spekulationen, jedoch außerdem auch die Möglichkeit, diese quantitativ zu evaluieren.

Der zweite wichtige Aspekt ist die schon angesprochene Möglichkeit der Integration unterschiedlicher Daten. Der schnelle Fortschritt auf dem Gebiet der Genetik ist nicht zuletzt auch darin begründet, dass es in diesem Bereich gelungen ist, das Wissen um Sequenzen und deren Eigenschaften in verschiedenen Datenbanken zu sammeln und auszutauschen.

Obwohl das Gehirn ein zumindest ebenso komplexes Netzwerk bildet, wie es die in Wechselwirkung tretenden Proteine tun, ist die Neurobiologie noch weit von einer solchen Integration unterschiedlicher Daten entfernt.

Erst wenn unterschiedliche Daten aus der Morphologie, der Physiologie, der Ethologie und der Modellierung und damit auch Ergebnisse unterschiedlicher Methoden in einer Datenbank integriert werden können, werden wir in der Lage sein, die Funktionsweise unseres Gehirns besser zu verstehen.

Ich bin der festen Überzeugung, dass es mit der hier vorgestellten Methode möglich ist, eine Datenbank eines Gehirnareals zu erstellen, die mit unterschiedlichen Techniken gewonnene Ergebnisse integriert und zu starken Synergie-Effekten führt.

Nach Albert Einstein ist „*Das Schönste, was wir entdecken können, das Geheimnisvolle*“. Ich hoffe, dass diese Methode dazu beitragen wird, das geheimnisvolle Organ Gehirn besser zu verstehen.

Glossar

Alpha-Kanal Als Alpha-Kanal bezeichnet man einen vierten Kanal (neben Rot, Grün und Blau), der einen Transparentwert speichert. Je nach Wert des Alpha-Kanals ist der jeweilige Pixel durchsichtig (255) oder undurchsichtig (0).

API API steht für Application Programming Interface, frei übersetzt eine Programmierschnittstelle für Anwendungen. Ein API dient dazu, eine komplexe Funktionalität, wie etwa eine Hardwareschnittstelle oder komplizierte Bildbearbeitungsverfahren, anderen Programmen zur Verfügung zu stellen. Dazu stellt das API Funktionen bereit, die der Programmierer ohne die exakte Funktionsweise der Hardware bzw. der Verfahren zu kennen, in seinen Programmen verwenden kann.

Array Ein Array ist eine in der Informatik sehr gebräuchliche Struktur, die dazu dient, Daten im Arbeitsspeicher abzulegen und einen einfachen Zugriff zu gewährleisten. Im einfachsten Fall handelt es sich bei einem Array um eine Liste; dann spricht man von einem eindimensionalen Array. Bei einem zweidimensionalen Array werden die Daten vergleichbar einer Tabelle in zwei Dimensionen gespeichert. Ebenso kommen multidimensionale Arrays zur Anwendung.

Integer Als Integer bezeichnet man ein Zahlenformat, das die Menge der ganzzahligen Zahlen in einem Wertebereich von (abhängig vom Betriebssystem hier für ein 16-Bit-Betriebssystem) -65536 bis +65536 umfasst. Eine Integer-Variable kann als Wert eine dieser Zahlen, jedoch keine Gleitkommazahl annehmen.

JNI Das Java Native Interface (JNI) ist eine Schnittstelle, die es ermöglicht, Programme und Bibliotheken, die in anderen Programmiersprachen wie C oder Fortran geschrieben sind, in ein Java-Programm zu integrieren. Dadurch ist es möglich, leistungsfähige Bibliotheken zu verwenden; allerdings wird der Quelltext dadurch von einer bestimmten Plattform abhängig und die Portabilität geht verloren.

Lookup-Table Eine Lookup-Table ist eine Tabelle mit den diskreten Werten einer Funktion, die die Transformation der gemessenen Werte in den Farbraum von 0 bis 255 für die drei Kanäle Rot, Grün und Blau definiert. Eine Lookup-Table kann als Hardware-Lösung (Mikroskop-Kamera) als auch als reine Software-Lösung implementiert werden.

Spline Ursprünglich waren Splines Lehren aus Holz, die im Schiffbau benutzt wurden, um geschwungene Formen zu bauen. In der Informatik werden Kurvenzüge durch eine Punktmenge, bei denen die Kurve durch mehrere Polynome genähert wird, als Splines bezeichnet.

Voxel Als Voxel werden Pixel im dreidimensionalen Raum bezeichnet. Wenn man sich im zweidimensionalen Raum die Pixel als Quadrate vorstellt, so entspricht ein Voxel einem Würfel, der nicht nur in x- und y-, sondern auch in z-Richtung eine Ausdehnung hat.

Abkürzungsverzeichnis

ASCII	American Standard Code for Information Interchange
CLSM	Confocal Laser Scanning Microscopy
PET	Funktionale Magnetresonanztomographie
IC	Inferiorer Colliculus
JAI	Java Advanced Imaging
MSO	Mediale Superiore Olive
PET	Positronenemissionstomographie
PNG	Portable Network Graphic
PPM	Portable Pixmap
SOC	Superior Olivary Complex
TEM	Transmissionselektronenmikroskopie
TIFF	Tagged Image File Format

Index

- Alpha-Kanal, 32
- Archimedes-Alignment, 95
- Array, 39
- Auflösung
 - axiale, 2, 6
 - laterale, 2, 5
- Ausrichten, 91
 - manuelle Verfahren, 91
- CIE-Standard, 61
- CLSM, 120
- Cluster, 130
- color constancy, 61
- Compiler, 30
- Computertomographie, 91
- Crosspower-Spektrum, 39
- Damasio, 1
- Edelman, 4
- Einbettung, 10
- Elektronenmikroskop, 92
- Elektronenmikroskopie, 4
- Färbemethode, 11
- FFTW-Bibliothek, 47
- fMRI, 3
- Fourier Transformation, 38
- Gaußpyramide, 46
- Gottesfeld-Brown, 39
- Gray World, 66
- Histogramm-Backprojection, 79
- Histologie, **9**
- HSI-Farbraum, 61
- Image Mosaicing
 - Algorithmen
 - Kreuzkorrelation, 43
- Image Mosaicing, 5
 - Algorithmen
 - Phasenkorrelation, 38
- Inferioren Colliculus, 2
- Integer, 39
- Interpolator
 - bikubisch, 46
 - bilinear, 46
- JAI, 31, 46
- Java, 31
- Java Advanced Imaging, *siehe* JAI
- Java Native Interface, *siehe* JNI
- Jeffres-Modell, 4
- JNI, 47
- JVM, 30
- Koch, 4
- Konfokale Laser-Scanning-Mikroskopie, 3, 5
- Kreisapproximation, 126
- Kuglin, 39
- Laserscanning Mikroskopie, 91
- Laufzeit, 44
- Linux, **30**
- Lookup-Table, **17**
- Mediale Superiore Olive, 4
- Meriones unguiculatus, 6
- Meriones unguiculatus, 9
- Mikroskopie, **11**, 40
- Minimum-Distance-Klassifikator, 82
- Montgomery, 44
- MSO, 6
- NeuroLucida, 92

NMR, 91

Osmium-Fixierung, 11

PET, 3

Phasenkorrelation, 38, 39

Phasenkorrelations-Oberfläche, 40

PNG, 49

Präparation, 9

Real Median Shift, **71**

Rechenleistung, 31

Referenzpunkte, 91, 92

Rekonstruktion

- funktional, **6**
- optisch, **6**

RGB-Farbraum, 61

Richmond, 42

RMS, *siehe* Real Median Shift

Scanning-Verfahren, 91

Shift-Theorem, 39

Speicherkapazität, 32

Superiore Olive, 3

TEM, 3

Transformationsmatrix, 14

Überblenden, 48

Wüstenrennmaus, 9

White World, 65

Literaturverzeichnis

- [1] J. C. Adams und E. Mugnaini. *Immunocytochemical evidence for inhibitory and disinhibitory circuits in the superior olive*. *Hear Res*, 49(1-3): 281–98 (1990).
- [2] J. B. Aguayo, S. J. Blackband, J. Schoeniger, M. A. Mattingly und M. Hintermann. *Nuclear magnetic resonance imaging of a single cell*. *Nature*, 322: 190–191 (1986).
- [3] M. S. Albert. *Detection of very early Alzheimer disease through neuroimaging*. *Alzheimer Dis Assoc Disord*, 17 Suppl 2: S63–5 (2003).
- [4] AT&T Bell Laboratories, University of Tennessee und Oak Ridge National Laboratory. *Java Linpack List of Timings*. Internetadresse. http://www.netlib.org/benchmark/linpackjava/timings_list.html.
- [5] AT&T Bell Laboratories, University of Tennessee und Oak Ridge National Laboratory. *Netlib Repository at UTK and ORNL*. Internetadresse. <http://www.netlib.org/>.
- [6] R. Baldock und J. Graham, Hg. *Image Processing and Analysis*. Oxford university press, Oxford (2000).
- [7] P. Bamford und B. Lovell. *Robust Cell Nucleus Segmentation Using a Viterbi Search Based Active Contour*. In *Proceedings of DICTA '97, Conference on Digital Image Computing Techniques and Applications* (1997).
- [8] P. Bamford und B. Lovell. *Unsupervised Cell Nucleus Segmentation with Active Contours*. *Signal Processing Special Issue: Deformable Models*, 71(2) (1998).
- [9] G. E. Beckius, R. Batra und D. L. Oliver. *Axons from anteroventral cochlear nucleus that terminate in medial superior olive of cat: observations related to delay lines*. *J Neurosci*, 19(8): 3146–61 (1999).
- [10] M. Beller, R. Stotzka, H. Gemmeke, K. F. Weibezahn und G. Knedlitschek. *Bildverarbeitung für ein motorisiertes Lichtmikroskop zur automatischen Lymphozytenidentifikation*. In T. Wittenberg, P. Hastreiter, U. Hoppe, H. Handels, A. Horsch und H.-P. Meinzer, Hg., *Bildverarbeitung für die Medizin 2003*, S. 196–200. Gesellschaft für Informatik (2003).
- [11] J. Blinn. *Jim Blinn Corner Dirty Pixels*. Morgan Kaufmann Publishers, erste Aufl. (1998).

- [12] F. Boddeke, L. van Vilet, H. Netten und I. Young. *Autofocusing in Microscopy based on the OTF and Sampling*. Bioimaging, S. 193–203 (1994).
- [13] O. Bossert. *Computergestützte Rekonstruktion von Neuronen*. Diplomarbeit, JWG-Universität, Frankfurt am Main (2001).
- [14] P. Bourke. *2 Dimensional FFT*. Internetadresse (1998). <http://astronomy.swin.edu.au/pbourke/analysis/fft2d/>.
- [15] A. Brand, O. Behrend, T. Marquardt, D. McAlpine und B. Grothe. *Precise inhibition is essential for microsecond interaural time difference coding*. Nature, 417(6888): 543–7 (2002).
- [16] K. Brändle. *A New Method for Aligning Histological Serial Sections for Three-Dimensional Reconstruction*. Computers and Biomedical Research, 22: 52–62 (1989).
- [17] G. Buchsbaum. *A spatial processor model for object color perception*. Journal of the Franklin Institute, 310: 1–26 (1980).
- [18] R. y Cajal. *Histologie du Systeme Nerveux de l'Homme et des Vertebrates*. Malonie, Paris (1907).
- [19] G. Chetelat und J. C. Baron. *Early diagnosis of Alzheimer's disease: contribution of structural neuroimaging*. Neuroimage, 18(2): 525–41 (2003).
- [20] E. Covey, M. Vater und J. H. Casseday. *Binaural properties of single units in the superior olivary complex of the mustached bat*. J Neurophysiol, 66(3): 1080–94 (1991).
- [21] G. Crow, A. L. Rupert und G. Moushegian. *Phase locking in monaural and binaural medullary neurons: implications for binaural phenomena*. J Acoust Soc Am, 64(2): 493–501 (1978).
- [22] A. R. Damasio. *Descartes' Irrtum*. dtv, München, vierte Aufl. (1997).
- [23] D. Dehn. *Histologische Untersuchungen am Hirnstamm von Wirbeltieren*. Diplomarbeit, Johann-Wolfgang-Goethe Universtiät, Frankfurt am Main (2000).
- [24] A. Drzezga, N. Lautenschlager, H. Siebner, M. Riemenschneider, F. Willoch, S. Minoshima, M. Schwaiger und A. Kurz. *Cerebral metabolic changes accompanying conversion of mild cognitive impairment into Alzheimer's disease: a PET follow-up study*. Eur J Nucl Med Mol Imaging, 30(8): 1104–13 (2003).
- [25] J. Eggert und J. L. van Hemmen. *Modeling neuronal assemblies: theory and implementation*. Neural Comput, 13(9): 1923–74 (2001).
- [26] J. Engelhardt und W. Knebel. *Konfokale Laserscanning Mikroskopie*, Kap. 5, S. 219–232. In Robenek [84], erste Aufl. (1995).
- [27] S. Fahn. *Description of Parkinson's disease as a clinical syndrome*. Ann N Y Acad Sci, 991: 1–14 (2003).

-
- [28] J. Fiala und K. M. Harris. *Extending unbiased stereology of brain ultrastructure to three-dimensional volumes*. Journal of the American Medical Informatics Association, 8(1): 1–16 (2001).
- [29] G. D. Finlayson, B. Schiele und J. L. Crowley. *Comprehensive Colour Image Normalization*. Lecture Notes in Computer Science, 1406: 475–490 (1998).
- [30] D. C. Fitzpatrick, S. Kuwada und R. Batra. *Neural sensitivity to interaural time differences: beyond the Jeffress model*. J Neurosci, 20(4): 1605–15 (2000).
- [31] M. Frigo und S. G. Johnson. *FFTW-Homepage*. Internetadresse. <http://www.fftw.org>.
- [32] M. Frigo und S. G. Johnson. *FFTW: An Adaptive Software Architecture for the FFT*. In *ICASSP conference proceedings*, Bd. 3, S. 1381–1384 (1998).
- [33] J. M. Goldberg und P. B. Brown. *Functional organization of the dog superior olivary complex: an anatomical and electrophysiological study*. J Neurophysiol, 31(4): 639–56 (1968).
- [34] A. D. Gordon. *Classification (CRC Monographs on Statistics & Applied Probability (Hardcover))*. Chapman & Hall/CRC (1999).
- [35] L. Gottesfeld-Brown. *A Survey of Image Registration Techniques*. ACM Computing Surveys, 24(4): 325–376 (1992).
- [36] N. G. Gough et al. *A method for the accurate location and orientation of structures studied by the use of serial microscopical sections*. Journal of the Royal Microscopical Society, 88: 291–300 (1967).
- [37] B. Grothe. *Interaction of excitation and inhibition in processing of pure tone and amplitude-modulated stimuli in the medial superior olive of the mustached bat*. J Neurophysiol, 71(2): 706–21 (1994).
- [38] B. Grothe. *The evolution of temporal processing in the medial superior olive, an auditory brainstem structure*. Prog Neurobiol, 61(6): 581–610 (2000).
- [39] B. Grothe. *New roles for synaptic inhibition in sound localization*. Nat Rev Neurosci, 4(7): 540–50 (2003).
- [40] B. Grothe und G. Neuweiler. *The function of the medial superior olive in small mammals: temporal receptive fields in auditory analysis*. J Comp Physiol [A], 186(5): 413–23 (2000).
- [41] B. Grothe und T. J. Park. *Sensitivity to interaural time differences in the medial superior olive of a small mammal, the Mexican free-tailed bat*. J Neurosci, 18(16): 6608–22 (1998).
- [42] B. Grothe und D. H. Sanes. *Synaptic inhibition influences the temporal coding properties of medial superior olivary neurons: an in vitro study*. J Neurosci, 14(3 Pt 2): 1701–9 (1994).

- [43] P. Haberäcker. *Praxis der Digitalen Bildverarbeitung und Mustererkennung*. Hanser (1995).
- [44] H. Heffner und B. Masterton. *Hearing in glires: domestic rabbit, cotton rat, feral house mouse, and kangaroo rat*. J. Acoust. Soc. Am., 68: 1584–1599 (1980).
- [45] R. S. Heffner und H. E. Heffner. *Sound localization and use of binaural cues by the gerbil (*Meriones unguiculatus*)*. Behav Neurosci, 102(3): 422–8 (1988).
- [46] G. C. Heijke et al. *Method for morphometric analysis of axons in experimental peripheral nerve reconstruction*. Microsurgery, 20(5) (2000).
- [47] R. Hessling. *Maximale Schnittdicke bei der Laser-Scanning-Mikroskopie*. Persönliche Mitteilung (2003).
- [48] M. A. Hickey und M. F. Chesselet. *Apoptosis in Huntington's disease*. Prog Neuropsychopharmacol Biol Psychiatry, 27(2): 255–65 (2003).
- [49] M. K. Hu. *Visual Pattern Recognition by Moment Invariants*. IRE Transactions on Information Theory, IT-8: 179–187 (1962).
- [50] *ImageJ Homepage*. Internetadresse. <http://rsb.info.nih.gov/ij/>.
- [51] *Java Advanced Imaging Tutorial*. Internetadresse (2001). <http://developer.java.sun.com/developer/onlineTraining/javaai/>.
- [52] L. A. Jeffress. *A Place Theory of Sound Localization*. J. Comp. Psychol., 41: 35–49 (1948).
- [53] *Java Native Interface*. Internetadresse (2001). <http://java.sun.com/products/jdk/1.2/docs/guide/jni/>.
- [54] P. X. Joris, P. H. Smith und T. C. Yin. *Coincidence detection in the auditory system: 50 years after Jeffress*. Neuron, 21(6): 1235–8 (1998).
- [55] P. X. Joris und T. C. Yin. *Envelope coding in the lateral superior olive. I. Sensitivity to interaural time differences*. J Neurophysiol, 73(3): 1043–62 (1995).
- [56] C. Kapfer, A. H. Seidl, H. Schweizer und B. Grothe. *Experience-dependent refinement of inhibitory inputs to auditory coincidence-detector neurons*. Nat Neurosci, 5(3): 247–53 (2002).
- [57] M. Kausch. *Erarbeiten einer Methode zur Ermittlung von Richtungshörschwellen bei *Meriones unguiculatus* (Gerbillinae)*. Diplomarbeit, Johann Wolfgang Goethe-Universität, Frankfurt am Main (1984).
- [58] U. Kischka, C.-W. Wallesch und G. Wolf. *Methoden der Hirnforschung*. Spektrum Akademischer Verlag, Heidelberg, Berlin (1997).
- [59] A. Kiss und K. Majorossy. *Neuron morphology and synaptic architecture in the medial superior olivary nucleus. Light- and electron microscope studies in the cat*. Exp Brain Res, 52(3): 315–27 (1983).

- [60] S. Kittipiyakul. *Automated Remote Microscope for Inspection of Integrated Circuits*. Internetadresse (1996). <http://www-mtl.mit.edu/CAPAM/memos/96-9/96-9.html>.
- [61] C. Koch und J. L. Davis, Hg. *Large-Scale Neuronale Theories of the Brain*. MIT Press, Cambridge, London (1994).
- [62] C. Kuglin und D. Hines. *The phase correlation image alignment method*. In *Proceedings of the IEEE 1975 International Conference on Cybernetics and Society*, S. 163–165 (1975).
- [63] C. Levinthal und R. Ware. *Three dimensional reconstruction from serial sections*. *Nature*, 236: 207–210 (1972).
- [64] W. E. Lorensen und H. E. Cline. *Marching Cubes: A high resolution 3D surface construction algorithm*. *Computer Graphics*, 21(4): 163–169 (1987).
- [65] D. A. Lyon. *Image Processing in Java*. Prentice Hall, zweite Aufl. (1999).
- [66] M. F. Lythgoe, N. R. Sibson und N. G. Harris. *Neuroimaging of animal models of brain disease*. *Br Med Bull*, 65: 235–57 (2003).
- [67] T. Masters. *Signal and image processing with neural networks*. John Wiley & Sons (1994).
- [68] J. Matas, R. Marik und J. Kittler. *On representation and matching of multi-colored objects*. In *Proceedings of the fifth International Conference on Computer Vision*, S. 726–732. IEEE Computer Society (1995).
- [69] D. McAlpine und B. Grothe. *Sound localization and delay lines - do mammals fit the model?*. *Trends Neurosci*, 26(7): 347–50 (2003).
- [70] J. J. McCann, J. A. Hall und E. H. Land. *Color mondrian experiments: The study of average spectral distributions*. *J. Opt. Soc. Amer.*, A(67) (1977).
- [71] L. Meunier und M. Borgmann. *High-Resolution Panoramas Using Image Registration*. Internetadresse (2000). <http://www-ise.stanford.edu/class/ee368/project13/>.
- [72] K. Montgomery. *Automated Reconstruction of Neural Elements from Transmission Electron Microscope Images*. Diplomarbeit, University of California Santa Cruz (1996).
- [73] *MOSIX Homepage*. Internetadresse. <http://www.mosix.org>.
- [74] M. S. Okun. *Huntington's Disease: What We Learned From the Original Essay*. *Neurolog*, 9(4): 175–9 (2003).
- [75] J. Orwant, J. Hietaniemie und J. Macdonald. *Algorithmen mit Perl*. O'Reilly, Köln, erste Aufl. (2000). ISBN 3-89721-141-6.
- [76] M. Peschanski und S. B. Dunnett. *Cell therapy for Huntington's disease, the next step forward*. *Lancet Neurol*, 1(2): 81 (2002).

- [77] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. R. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, zweite Aufl. (1992).
- [78] P. Rakic, L. J. Stensas, E. P. Sayre und R. L. Sidman. *Computer-aided three dimensional reconstruction and quantitative analysis of cells from serial electron microscopic montages of foetal monkey brain*. *Nature*, 250: 31 (1974).
- [79] B. Randell, Hg. *The Origins of Digital Computers*. Springer, Berlin, Heidelberg, New York (1973).
- [80] I. Rechenberg. *Evolutionsstrategie '94*. frommann-holzboog, Stuttgart (1994).
- [81] B. Reddy und B. Chatterji. *An FFT-based technique for translation, rotation, and scale-invariant image registration*. *IEEE Trans. on Image Processing*, 3(8): 1266–1270 (1996).
- [82] S. Reuss, U. Disque-Kaiser, S. De Liz, M. Ruffer und R. Riemann. *Immunofluorescence study of neuropeptides in identified neurons of the rat auditory superior olivary complex*. *Cell Tissue Res*, 297(1): 13–21 (1999).
- [83] J. Richmond. *Image Mosaicing Final Report*. Techn. Ber., Department of Computer Science at the University of British Columbia (1999). <http://www.cs.ubc.ca/spider/jlrichmo/mosaic/>.
- [84] H. Robenek, Hg. *Mikroskopie in Forschung und Praxis*. GIT Verlag, Darmstadt, erste Aufl. (1995). ISBN 3-928865-18-8.
- [85] B. Romeis. *Mikroskopische Technik*. Urban und Schwarzenberg, München Wien Baltimore (1989). ISBN 3-930253-79-8.
- [86] R. Rubsamen, M. Gutowski, J. Langkau und G. J. Dorrscheidt. *Growth of central nervous system auditory and visual nuclei in the postnatal gerbil (Meriones unguiculatus)*. *J Comp Neurol*, 346(2): 289–305 (1994).
- [87] A. Ryan. *Hearing sensitivity of the mongolian gerbil, Meriones unguiculatus*. *J. Acoust. Soc. Am.*, 59(5): 1222–26 (1976).
- [88] M. E. Scheibel und A. B. Scheibel. *Neuropil organization in the superior olive of the cat*. *Exp Neurol*, 43(2): 339–48 (1974).
- [89] I. R. Schwartz. *Dendritic arrangements in the cat medial superior olive*. *Neuroscience*, 2(1): 81–101 (1977).
- [90] P. H. Smith. *Structural and functional differences distinguish principal from nonprincipal cells in the guinea pig MSO slice*. *J Neurophysiol*, 73(4): 1653–67 (1995).
- [91] P. H. Smith, P. X. Joris und T. C. Yin. *Anatomy and physiology of principal cells of the medial nucleus of the trapezoid body (MNTB) of the cat*. *J Neurophysiol*, 79(6): 3127–42 (1998).

- [92] I. Sobel, C. Levinthal und E. Macagno. *Special techniques for the automatic computer reconstruction of neuronal structures*. Annual Review of Biophysics and Bioengineering, 9: 347–362 (1980).
- [93] J. Spacek. *Three-dimensional reconstruction of astroglia and oligodendroglia cell*. Zeitschrift für Zellforschung und mikroskopische Anatomie, 112: 430–442 (1971).
- [94] S. P. Springer und G. Deutsch. *Linkes - Rechtes Gehirn*. Spektrum Akademischer Verlag, Heidelberg, Berlin, vierte Aufl. (1998). ISBN 3-8274-0366-9.
- [95] Sun Microsystems, Palo Alto. *Java Advanced Imaging API White Paper*, erste Aufl. (1999).
- [96] M. J. Swain und D. H. Ballard. *Color Indexing*. International Journal of Computer Vision, 7(1): 11–32 (1991).
- [97] D. J. Tollin. *The lateral superior olive: a functional role in sound source localization*. Neuroscientist, 9(2): 127–43 (2003).
- [98] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos (1990).
- [99] H. Wolf, V. Jelic, H. J. Gertz, A. Nordberg, P. Julin und L. O. Wahlund. *A critical discussion of the role of neuroimaging in mild cognitive impairment*. Acta Neurol Scand Suppl, 179: 52–76 (2003).
- [100] G. Wyszecki und W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons, New York, zweite Aufl. (1982).

Einige der Quellen sind Dokumente im Internet. Diese Quellen können sich in der Zwischenzeit verändert haben oder wurden eventuell sogar vom Betreiber entfernt. Da jedoch viele Artikel nur in elektronischer Form vorliegen und wichtige Teile der Arbeit auf diesen Artikel beruhen, konnte ich leider nicht darauf verzichten, sie in das Literaturverzeichnis aufzunehmen.

Anhang

Puffer und Lösungen zur Histologie

Ringerlösung für Säugetiere

Für eine Menge von 1000 ml:

- 8,0 g Natriumchlorid (NaCl)
- 0,2 g Kaliumchlorid (KCl)
- 1,0 g Natriumbicarbonat (NaHCO₃)
- 0,22 g Calciumchlorid (CaCl₂)
- 1000 ml Aqua. bidest.

Das Natriumbicarbonat wird zusammen mit dem Natrium-, dem Kalium- und dem Calciumchlorid in Aqua bidest. gelöst.

Glutaraldehyd-Fixativ

Für eine Menge von 500 ml:

- 338,0 ml Aqua bidest.
- 5,0 g Paraformaldehyd
- 20,0 ml Gereinigtes Glutaraldehyd 25%
- 150,0 ml Phosphatpuffer (0,4 molar bei pH 7,2-7,4)
- 2,0 ml Calciumchlorid (CaCl₂) 0,5%

Das Aqua bidest. wird zusammen mit dem Paraformaldehyd erhitzt, so dass letzteres depolymerisiert. Zu der milchigen Mischung werden 1-2 Tropfen NaOH gegeben, bis sich der Niederschlag auflöst und die Mischung klar wird.

Der Erlenmeyerkolben wird mit Alufolie verschlossen und so schnell wie möglich in Eiswasser abgekühlt.

Zu der abgekühlten Lösung werden Glutaraldehyd, Phosphatpuffer und Calciumchlorid gegeben. Die Fixierflüssigkeit wird unter dem Abzug durch einen Filter filtriert.

Phosphatpuffer (0,4 molar bei pH 7,2-7,4)

Für eine Menge von 500 ml:

- 500 ml Aqua bidest.
- 5,3 g Prim. Natriumphosphat ($\text{NaH}_2\text{PO}_4 \times \text{H}_2\text{O}$)
- 28,0 g Sek. Kaliumphosphat (K_2HPO_4)

Das Natrium- und Kaliumphosphat langsam in 200 ml Aqua bidest. auflösen und danach auf 500 ml auffüllen.

Der pH-Wert wird mittels Indikatorpapier kontrolliert.

Waschflüssigkeit

Für eine Menge von 100 ml:

- 70 ml Aqua bidest.
- 30 ml Phosphatpuffer (0,4 molar bei pH 7,2-7,4)
- 8 g Glukose
- 20 Tropfen Calciumchlorid (CaCl_2) 0,5%

Calciumchlorid mit dem Phosphatpuffer zu dem Aqua bidest. geben und die Glukose darin lösen.

Osmium-Fixationslösung zum Nachfixieren

Für eine Menge von 20 ml:

- 10 ml OsO_4 (4%)
- 3 ml Aqua bidest.
- 7 ml Phosphatpuffer (0,4 molar bei pH 7,2-7,4)
- 1,4 g Glukose
- 4 Tropfen Calciumchlorid (CaCl_2) 0,5%

Aqua bidest. mit dem Phosphatpuffer mischen, Osmiumtetroxid und das Calciumchlorid dazu geben und die Glukose darin lösen.

Protokoll zur Mikroskopie

Einstellungen am Mikroskop

Vor dem Einschalten des Computers sind die folgenden Schritte durchzuführen:

1. Ausdruck der Formblätter zum Protokollieren (siehe Seite 168).
2. Kamera einschalten.
3. Beleuchtung einschalten.
4. Kappe von der Leuchtfeldblende entfernen.
5. Alle Graufilter im Strahlengang entfernen.
6. Darauf achten, dass der Kondensator auf der Einstellung **H**(ellfeld) steht.
7. Fluoreszenzfilter auf einen leeren Halter einstellen.
8. 40fache Vergrößerung einstellen.
9. Die Intensität der Beleuchtung auf 3,25 einstellen.
10. Eine Stelle des Präparats auswählen, so dass Gesichtsfeld ausgefüllt ist. Diese Stelle sollte einige Zellen enthalten.
11. Das Präparat fokussieren.
12. Beide Schieber oben rechts ganz herausziehen, so dass das Bild nur noch auf den Kamera-Schacht projiziert wird. Das Okular sollte nun dunkel sein.

Bedienung des Computers

1. Computer einschalten.

2. Wenn die Kamera zuvor farblich kalibriert wurde, die eigene Kalibrierungsdatei restaurieren. Ansonsten die Datei *EigeneDateien\PolaroidDMC2\FF_610410TR.fcal* umbenennen oder löschen.
3. Das Programm Polaroid DMC 2 starten
4. Preview auswählen
5. Einen Bildausschnitt durch Klicken auf das Bild und Auswahl des Icons mit einer Lupe und einem + am unteren Rand.
6. Diesen Ausschnitt mittels der Fokussierhilfe unter optischer Kontrolle scharf stellen.
7. Das Mikroskop *unter Verwendung der Kamera* köhlern:
 - Kondensor bis zum Anschlag nach oben stellen.
 - Die Aperturblende wird ganz geöffnet.
 - Die Leuchtfeldblende wird fast ganz geschlossen, bis nur noch ein kleiner unscharfer Lichtfleck zu sehen ist.
 - Den Kondensor in der Höhe nach unten verstellen bis die Ränder der Leuchtfeldblende scharfstellen eingestellt sind.
 - Optimalerweise sollte ein farbiger Schimmer um die Leuchtfeldblende herum verschwinden
 - Die Leuchtfeldblende wird so weit aufgezogen, dass das gesamte Blickfeld optimal ausgeleuchtet ist.
8. Das Okular *vorsichtig* herausnehmen und die Aperturblende so einstellen, dass ca. $\frac{2}{3}$ der hinteren Linse des Objektivs ausgeleuchtet sind.
9. In der Software unter `Tools - Histogramme` ein Fenster, das die Histogramme darstellt, öffnen und damit permanent die Qualität der Bilder kontrollieren.
10. Das Feld `Color` für farbige Aufnahmen aktivieren
11. Unter `Tools - Set Preferences` die folgenden Einstellungen vornehmen:
 - Multiframe-Averaging mit jeweils 2 Frames
 - Output – Channel Mixture: Identity
 - Output – Lookup-Table: Linear
12. Im Hauptfenster folgende Einstellungen vornehmen:
 - Auflösung: 1600x1200 Pixel
 - 24bit Frabtiefe
 - Sensitivity: High
13. Unter Histogramm-Kontrolle die Exposue-Time auf ca. 64 ms einstellen.

-
14. Probekbild aufnehmen.
 15. Das Probekbild auf sein Histogramm hin untersuchen.
 16. Wenn die Farbverteilung verändert werden soll:
 - Farbbereich, der getreckt werden soll, festlegen
 - Geeignete Lookup-Table berechnen
 - Die Datei `C:\WINDOWS\TWAIN_32\PolaroidDMC2\LOG.lut` umbenennen (z.B. `LOG.lut.old`)
 - Die neu erstellte Lookup-Table nach `C:\WINDOWS\TWAIN_32\PolaroidDMC2\LOG.lut` kopieren
 - Die Software neu starten
 - Unter `Tools - Set Preferences - Output - Lookup-Table` als Einstellung `Logarithm` auswählen.
 - Erneutes Probekbild machen
 - Nach dem Ende der Aufnahmen die Datei `LOG.lut.old` wieder in `LOG.lut` umbenennen.

Während der Aufnahme der Serie ist auf folgendes zu achten:

1. Alle Einstellungen des Mikroskops protokollieren
2. Jede Aufnahme im Protokoll vermerken
3. Vor jeder Aufnahme den Fokus überprüfen
4. Bei großen Serien immer darauf achten, ob sich die Farbverteilung im Histogramm deutlich ändert.

Skalierungsfaktoren

In der folgenden Tabelle sind die Umrechnungsfaktoren für die einzelnen Vergrößerungsstufen des Mikroskops angegeben:

Vegr.	800x600	1600x1200
5x	3,2258	1,6234
10x	1,6393	0,8064
20x	0,8197	0,4054
40x	0,3968	0,1992
63x	0,2577	0,1289

Mittels dieser Faktoren können Entfernungen und Flächen von Pixel nach Mikrometer umgerechnet werden. Die Umrechnung erfolgt nach folgender einfacher Formel:

$$\text{Mikrometer} = \text{Faktor} \cdot \text{Pixel} \quad (\text{B.1})$$

Protokollblätter

Mikroskopie-Protokoll

Struktur:	
Datum:	
Angefertigt von:	
Organismus:	
Verzeichnis:	

Mikroskop-Einstellungen

Licht-Intensität:	
Graufilter:	
Vergößerung:	
Bemerkungen:	

Einstellung der Software

Auflösung:	
Farbtiefe:	
Sensitivity:	
Exposure Time:	
Bild-Averaging:	
Matrix:	
Lookup-Table:	
Bemerkungen:	

Schnittdokumentation

Struktur:	
Datum:	
Angefertigt von:	
Organismus:	
Verzeichnis:	
Basisdateiname:	

Anordnung der Schnitte

Position	Block	Objektträger	Schnitt
1			
2			
3			
4			
5			

Schnittprotokoll

Block	Objekt-träger	Schnitt	Nicht vorhanden	Struktur nicht vorh.	Struktur beschädigt	Dateinamen-Erweiterung			
04	03	01	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	04	03	01	01
04	03	02	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	04	03	02	01
04	03	03	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	04	03	03	01
04	03	04	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	04	03	04	01
04	03	05	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	04	03	05	01
04	03	06	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	04	03	06	01
04	03	07	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	04	03	07	01
04	03	08	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	04	03	08	01
04	03	09	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	04	03	09	01
04	03	10	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	04	03	10	01

Tabellenverzeichnis

2.1	Speicherbedarf unterschiedlicher Bildformate für ein Bild der Größe 256x256	32
4.1	Vergleich des Ergebnisses der Software mit dem zweier menschlicher Experten	89
5.1	Beispiel eines Clusters	98

Abbildungsverzeichnis

1.1	Schematische Darstellung der methodischen Grenzen	2
1.2	Auflösungs- und Abbildungsbereiche der unterschiedlichen bildgebenden Verfahren	3
1.3	Schematische Darstellung der Methode	5
2.1	Paraffin-Schnitte als Vergleich	10
2.2	Vergleich der Färbung mit und ohne Osmium-Nachfixierung	11
2.3	Vergleich von unterschiedlichen Helligkeits/Kontrast-Einstellungen	13
2.4	Korrelation der einzelnen Farbkanäle eines Bildes	15
2.5	Boxplot der Werte vor der Transformation und der Ausgleichsfaktoren	16
2.6	Bildausschnitte, Lookup-Tables und das jeweilige Profil einer Zelle	19
2.7	Darstellung der drei unterschiedlichen Formen von Lookup-Tables	21
2.8	Erstellen des Zellprofils mit ImageJ	24
2.9	Flussschema zum Erstellen einer optimalen Lookup-Table	25
2.10	Darstellung der Mittelwerte einer Profilanalyse und den daraus resultierenden Kurven	26
2.11	Drei unterschiedliche Darstellungsformen zweier Zellen	27
2.12	Effekt der verwendeten Lookup-Table	28
2.13	Profilbild einer Zelle mit der endgültigen Kombination aus Transformationsmatrix und Lookup-Table	29
2.14	Speicherbedarf unterschiedlicher Bildformate	33
2.15	Korrelation zwischen der Prozessorgeschwindigkeit und dem im Linpack-Benchmark erzielten Wert	34
2.16	Benötigte Zeit unterschiedlicher Rechner für die Anwendung des ImageJ-Benchmarks auf das vierfach ergrößerte Mandrill-Image	35
3.1	Schematische Darstellung eines Mosaicings	38
3.2	Phasenkorrelations-Oberfläche	41
3.3	Mögliche Positionen eines Mosaicings bei den gleichen Koordinaten	42
3.4	Test der richtigen Positionierung nach der Phasenkorrelation	43
3.5	Schematische Darstellung einer Kreuzkorrelation	44
3.6	Vergleich zweier Phasenkorrelationsoberflächen	49
3.7	Methoden zur Überblendung	50
3.8	GUI des imosaic-Programms	52
3.9	Ausschnittsvergrößerung eines Image Mosaics	53

3.10	Ergebnis der Kontrollmechanismen bei den richtig und falsch kombinierten Bildpaaren	54
4.1	Problem beim Ausrichten der Bilder mit den Kapillaren als Referenz	59
4.2	Probleme bei der Zuordnung von Segmenten	60
4.3	Ein Beispielbild in unterschiedliche Farbräume zerlegt	62
4.4	Ergebnis des Profil-Plots ohne jegliche Korrektur	63
4.5	Ergebnis des Profil-Plots mit HSI-Korrektur	63
4.6	Das Beispielbild nach normalisieren des RGB-Farbraumes	65
4.7	Das Beispielbild nach normalisieren des RGB-Farbraumes	67
4.8	Ergebnis des Profil-Plots nach der Black World-Normalisierung	68
4.9	Ergebnis des Profil-Plots nach der Gray World-Normalisierung	68
4.10	Ergebnis des Profil-Plots mittels RGB-Normalisieren	69
4.11	Ergebnis des Profil-Plots mittels kombinierter RGB-Normalisieren	69
4.12	Das Beispielbild nach normalisieren mit dem RMS-Algorithmus	72
4.13	Ergebnis des Profil-Plots nach der RMS-Normalisierung	73
4.14	Erste Stufe des AMS-Algorithmus	74
4.15	Vergleich des AMS- und RMS-Verfahrens an Hand eines Beispiels	74
4.16	Vergleich der verschiedenen Verfahren zur Farbkorrektur	76
4.17	Maskieren der Kapillaren	79
4.18	Beispiel für eine Kantendetektion mit dem Sobel-Filter	80
4.19	Mögliche Konfigurationen für eine positive Erkennung von Soma und Nukleus mittels CSE	83
4.20	Die einzelnen Schritte des CSE-Algorithmus	84
4.21	Schematische Darstellung des Schachbrettmusters mit darüber liegendem Bewertungsquadrat	84
4.22	UNAHE-Kontrasterhöhung an einem Beispielbild	86
4.23	Thresholding und Segmentierung der Zellkerne	87
4.24	Vergleich des Ergebnisses der Software mit dem zweier menschlicher Experten	89
5.1	Beispiel für den Effekt einer Transformations in das Polar-Koordinatensystem	94
5.2	Transformation des Testbildes in den Fourier-Raum, links aufrecht, rechts um 90° gedreht.	94
5.3	Dreidimensionale Darstellung des Ergebnis eines Minimum-Distance-Clusterings	98
5.4	Validierung des Ausrichtens	99
5.5	Ausgerichtete Ebenen	100
5.6	Vergleich der menschlichen Experten mit der Software	101
6.1	Die grafische Oberfläche des <i>CellInspectors</i> mit Erläuterungen zu deren Bedienung	106
6.2	Das Programm <i>CellInspector</i> mit an- und ausgeschalteter Segmentenmarkierung	107
6.3	Richtig und falsch klassifizierte Zellkerne	109

6.4	Schematische Darstellung zweier Konfigurationen für einen Würfel beim Marching Cubes Algorithmus	112
6.5	Rendering der Zellkerne aus verschiedenen Perspektiven	113
6.6	Unterschiedliche Darstellungsformen	115
6.7	Verschiedene Möglichkeiten zusätzliche Informationen zu integrieren	116
6.8	Interaktion zwischen 3D-Ansicht und <i>CellInspector</i>	117
7.1	Übersicht des gesamten Verfahrens	121
7.2	Approximation der Zellkerne durch Ellipsen	125
7.3	Approximation der Zellkerne durch Aktive Konturen	126
7.4	Approximation der Zellkerne durch Kreisapproximation	127
7.5	Einzelschritte zur Zellerkennung	128
7.6	Schematische Darstellung der unterschiedlichen Prinzipien der Schall-detektion	134
7.7	Schematische Darstellung des Jeffress-Modells	135
7.8	Darstellung des Phase-Lockings für tief- und hochfrequente Signale	135
7.9	Verteilung der Projektionsmuster von MSO-Neuronen bei verschiedenen Säugern	136
7.10	Mögliche Verschaltung der Neurone in der MSO	137
7.11	Darstellung von detektierbaren ITDs bei unterschiedlichen Frequenzen	138
7.12	Golgi-Färbung der rechten MSO	138
7.13	Typische Beispiele für die Zelltypen der MSO	139
7.14	Schematische Darstellung der Frequenzschichten	141
7.15	Fähigkeit der Wüstenrennmaus zur Lateralisation von Tönen	141

Zusammenfassung

Ziel dieser Arbeit ist es, eine Lücke im methodischen Spektrum neurobiologischer Methoden zu schließen. Es ist heute möglich, das Gehirn auf unterschiedlichen Ebenen zu beschreiben. Es stehen jedoch keine Methoden zur Verfügung, um die Anordnung der Zellen innerhalb eines Nukleus quantitativ zu beschreiben.

Die Anzahl und die Anordnung der Zellen ist jedoch eine essentielle Voraussetzung, um die Funktion eines Nukleus zu verstehen. Das hier vorgestellte Verfahren zur Rekonstruktion eines Nukleus basiert auf Nissl-gefärbten Semidünnschnitten der Medialen Superioren Olive (MSO) der Wüstenrennmaus *Meriones unguiculatus*. Diese werden mit einer Digitalkamera lichtmikroskopisch aufgenommen und bilden die Basis der Rekonstruktion.

In einem ersten Schritt werden innerhalb einer Schnittebene mehrere Einzelbilder patchworkartig zu einem Image Mosaic zusammengefügt. Durch diesen Schritt ist die Auflösung innerhalb einer Schnittebene praktisch unbegrenzt. Das Verfahren beinhaltet mehrere Kontrollmechanismen und funktioniert praktisch fehlerfrei.

Danach werden die Bilder farblich korrigiert und mit Methoden der Mustererkennung werden die Zellkerne extrahiert. Die Extraktion der Zellkerne steht in ihrer Qualität einer manuellen Extraktion in nichts nach.

Die Zellkerne dienen als Grundlage für den Archimedes-Alignment-Algorithmus, der die Schnittserie in einen dreidimensionalen Bezug setzt, indem aufeinander folgende Schnitte aneinander ausgerichtet werden. Auch dieses Verfahren beinhaltet eine Kontrolle und funktioniert fehlerfrei.

Aus diesen Daten kann dann eine Rekonstruktion erstellt werden. Diese wird weiter ausgewertet und ergibt schließlich ein dreidimensionales Abbild des untersuchten Bereichs.

Sämtliche Verfahrensschritte arbeiten entweder fehlerfrei oder mit einer nur sehr geringen Fehlerrate. Somit stellt dieses Verfahren eine robuste, effiziente und universell anwendbare Möglichkeit für die umfassende Analyse der Neuronenverteilung im ZNS dar.

Das Verfahren eröffnet die Möglichkeit, Nuklei dreidimensional zu untersuchen, bietet aber auch einen Ansatzpunkt um mittels histologischer Daten weiteres Datenmaterial (etwa elektrophysiologischer oder morphologische Daten) zu integrieren.

Danksagung

Mein Dank gilt zuerst PD Dr. Wolfgang Plassmann für die fruchtbaren Diskussion, die guten Ideen und die praktische Hilfe. Ebenso möchte ich Prof. Dr. Manfred Kössl für die interessanten Diskussionen und die Möglichkeit, das Mikroskop seines Arbeitskreises zu verwenden, danken.

Ich möchte mich außerdem ganz herzlich bei der FAZIT-Stiftung dafür bedanken, dass sie meine Promotion mit einem so großzügigen Stipendium unterstützt hat. Ohne diese finanzielle Unterstützung wäre diese Arbeit so nicht möglich gewesen.

Ganz besonders möchte ich mich außerdem bei Marion Stöhr für ihre Sorgfalt und Schnelligkeit beim Anfertigen der Schnitte bedanken.

Herr Herberts vom technischen Support der Firma Polaroid war mir eine große Hilfe dabei, herauszufinden, wie die undokumentierten Funktionen der DMC 2 zu nutzen sind.

Diese Arbeit wurde durch das Vorhandensein freie Software sehr erleichtert und oftmals überhaupt erst möglich. Ich danke Linus Torvalds für sein Betriebssystem Linux, der Firma SUN für die Java-Entwicklungsumgebung, Donald Knuth für das Satzsystem \LaTeX sowie der Free Software Foundation und allen Entwicklern freier Software für die breite Basis an Programme, ohne die diese Arbeit nicht möglich gewesen wäre.

Meinen ganz besonderer Dank gilt meiner Freundin Anne Brunner, die viel Verständnis gezeigt hat und mir vor allem dann, wenn keines der Programme so lief, wie ich es mir vorstellte und Murphys Gesetze mit ihrer vollen Härte zuschlugen, ein unschätzbare Rückhalt war.

Versicherung

Diese Arbeit wurde unter Betreuung durch Herrn PD Dr. Plaßmann im Arbeitskreis CNB des Zoologischen Institutes der Johann Wolfgang Goethe-Universität angefertigt. Hiermit versichere ich, daß ich meine Arbeit ohne unzulässige – insbesondere dem Betreuer unbekannte Hilfe – selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Frankfurt am Main, den 21. August 2003

Oliver Bossert

Lebenslauf

Name		Bossert
Vorname		Oliver
Anschrift		Burgstraße 21 65817 Eppstein
Geburtsdatum		17. April 1977
Geburtsort		Frankfurt am Main
Staatsangehörigkeit		deutsch
Schulbildung	1983-1987	Holzhausen-Grundschule
	1987-1996	Liebig-Gymnasium (Frankfurt a. M.)
	17.06.1996	Abitur
Studium	1996-2001	Studium der Biologie (JWG-Universität Frankfurt a. M.)
	1996-1998	Studium der Informatik (Fernuniversität Hagen)
	24.09.1998	Vordiplom in Biologie
	25.10.2001	Diplom Biologie Neurobiologie, Zell- und Entwicklungsbiologie und Genetik Titel: <i>Computergestützte Rekonstruktion von Neuronen</i>
Stipendien und Preise	1996	Liebigpreis für besonderes Engagement und sehr gute Leistungen in der Schule
	2001	Reisestipendium der Freunde und Förderer der JWG-Universität
	1998-2001	Stipendium der Studienstiftung des deutschen Volkes
	2000-2002	Online-Stipendium von e-fellows.net
	2002-2003	Promotions-Stipendium der FAZIT-Stiftung