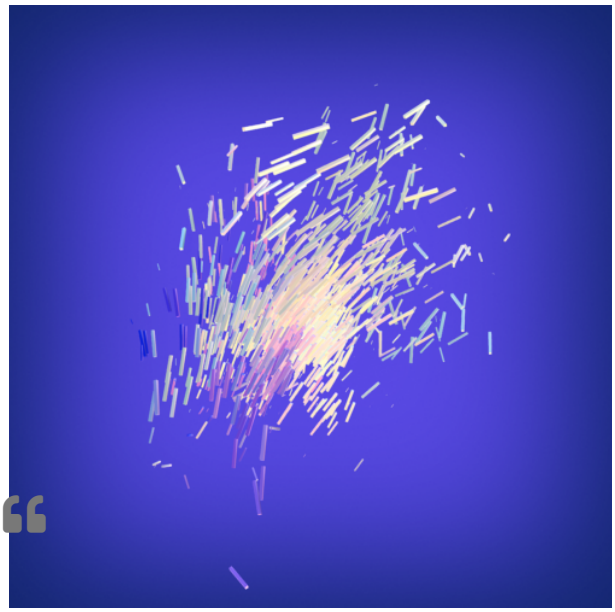
vi·son – Mixing Senses: Generative Art – Wie sich Kunst programmieren lässt

VON BENJAMIN DOUBALI UND GUIDO SCHMIDT · VERÖFFENTLICHT 27/07/2020 · AKTUALISIERT 27/07/2020

Wie entstehen audiovisuelle Daten-Skulpturen?

Kunst aus Daten, Codes und Zufällen: In der Generative Art erschaffen sich die Kunstwerke selbst – der Künstler liefert lediglich den Regelsatz dafür. Das Digitalkunst-Projekt „vi·son" programmiert digitale Klangskulpturen, denen ein Algorithmus zugrunde liegt. Hier beschreiben sie, wie das geht.



Guido Schmidt (vi·son): Sound Data Sculpture Momentum Particles 2020.07.19-14.19-001 (2020)

Künstlerische Ausdrucksweisen unterliegen einem stetigen Wandel. Nicht allein die Innovation künstlerischer Techniken treibt diesen voran. In einem von Informationstechnik geprägten Alltag und einer zunehmend digitalen Gesellschaft ist es nur folgerichtig, dass auch immer mehr digitale Kunst entsteht. Disziplinen wie Design, Fotografie und Medienkunst verwenden schon lange Software, um Entwürfe zu entwickeln, zu verfeinern und zu verfremden. Demgegenüber steht ein künstlerischer Schaffenszweig, der sich nicht nur der digitalen Technik bedient, sondern ganz und gar mit ihr verwoben ist: die Computerkunst und die sogenannte *Generative Art.*

Kurator und Digitalkunstexperte Jason Bailey schreibt hierzu:

Over the last 50 years, our world has turned digital at breakneck speed. No art form has captured this transitional time period – our time period – better than generative art. Generative art takes full advantage of everything that computing has to offer, producing elegant and compelling artworks that extend the same principles and goals artists have pursued from the inception of modern art.[1]

Neue Kunst für eine neue Gesellschaft?

Solch gemeinsame Prinzipien oder wiederkehrende Themen sind für Bailey etwa geometrische und musterhafte Ästhetiken. Maßgeblich ist die Abkehr von der figürlichen Darstellung – wie das Bauhaus oder der Kubismus sie programmatisch gesetzt hat. Der Schwerpunkt liegt darüber hinaus auf der methodischen Einführung des Zufalls und der Verfremdung ins künstlerische Schaffen, die der Dadaismus und Surrealismus forciert haben.

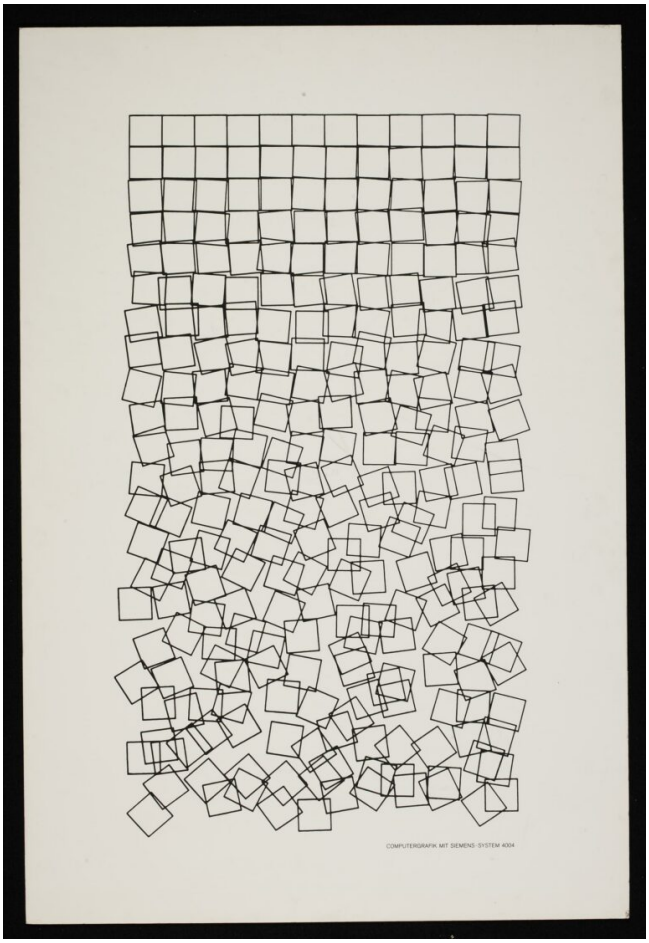
Was aber ist die von solch breiten Einflüssen inspirierte *Generative Art*? Bailey bietet eine schlanke, aber prägnante Definition: *Generative Art* "is art programmed using a computer that intentionally introduces randomness as part of its creation process."[2] Der Prozess zufallsbasierter Kunstprogrammierung in Baileys Sinne bedeutet dabei weder die völlige Autonomie des Computers noch eine vollständige Kontrolle über ihn: "The truth is that generative artists skillfully control both the magnitude and the locations of randomness introduced into the artwork."[3]

Vom zufallsbasierten Plottings zur KI

Im Aufsatz "Why Love Generative Art?" skizziert Bailey die geschichtliche Entwicklung der Generativen Kunst: Nach Pionierleistungen in den 60er Jahren mit ersten Computerkunst-Experimenten, die vor allem das Spiel mit dem Zufall in den Mittelpunkt stellten (wichtige Namen sind hier beispielsweise Georg Nees, Frieder Nake sowie Manfred Mohr), waren es in den darauffolgenden Jahrzehnten insbesondere weibliche Computerkünstlerinnen wie Vera Molnár, Lilian Schwartz und Muriel Cooper, die mit ihren Arbeiten zum Stil, zur Ästhetik und letztendlich auch zur explosionsartigen Verbreitung der *Generative Art* seit den Nullerjahren beitrugen. Ein Meilenstein war zudem die Veröffentlichung der Software-Plattform *Processing*, die aus einem Projekt rund um John Maeda, Ben Fry und Casey Reas hervorging.

In den letzten Jahren wurden die rein algorithmischen Verfahren durch den Einsatz von künstlicher Intelligenz (KI) erweitert. Sicherlich werden wir in Zukunft eine Vielzahl faszinierender Kunstwerke sehen, die unter Beteiligung einer KI entstanden sind. KI-basierte Digitalkunst lässt sich insofern als ein Subgenre der *Generative Art* begreifen.

Die *Generative Art* zieht ihre Ästhetik und Wirkung aus einem eigenwilligen Verhältnis von Kontrolle und Zufall. Das Ergebnis ist nie ganz vorhersehbar. Allerdings lässt sich – und das ist sicher eine Besonderheit – der Schöpfungsprozess bis ins kleinste Detail dokumentieren. Diese Eigenheit nutzen Künstler*innen, um nachzuvollziehen, wie die visuellen Digitalkunstwerke entstehen. Auch das Digitalkunstprojekts *vi·son* arbeitet im Rahmen der Reihe *Sound Data Sculpture Sketches* mit solchen Digitalskulpturen (die Beiträge der Reihe sind auf Instagram veröffentlicht), die im folgenden als Anschauungsbeispiel dienen.



Bevor es Computermonitore gab: Georg Nees: Schotter (1968-1970), Lithographie auf Papier © Victoria and Albert Museum, London.

Bauplan einer audiovisuellen Skulptur

Im Digitalkunstprojekt *vi·son* sollen Darstellungsweisen gefunden werden, um Musik visuell erlebbar zu machen. Das Ziel: Klang soll in einer fast synästhetischen Weise erlebbar werden. Diese Erkundung wird insbesondere durch die *New Media Art* beziehungsweise *Generative Art* beeinflusst. Vorderes Prinzip bei der Erstellung einer digitalen Skulptur ist das kreative Zusammenspiel von Unabgeschlossenheit, Iterationen und Parametersuche – in Jason Baileys Worten das gesteuerte Zufallsverhalten. Natürlich kennt der Programmierer sein selbstkreiertes System; der visuelle Output wird aber im sozio-technischen Arrangement zwischen *Augen – Händen – Tastatur – Monitor* ständig neu bewertet. Auf Basis dieser Neubewertung wird das System unablässig angepasst. Es eröffnen sich stets neue Möglichkeiten, die erprobt, überarbeitet, verworfen oder weiterentwickelt werden. Dieser "Feedbackloop" ist Teil der kreativen Dynamik.

Programmieren statt Pinsel und Leinwand

Bei der Anwendung klassischer Programmiersprachen geschieht dies in regelmäßigen Abständen, nachdem das Programm kompiliert und ausgeführt wurde. Etwas agiler ist der Prozess bei Echtzeitsystemen, wie zum Beispiel beim *Touchdesigner*, der auch für dieses Projekt eingesetzt wurde. Er stellt eine visuelle Programmierumgebung bereit, die trotzdem noch Freiheit für das Schreiben von Skripten, Grafikkartenprogrammen oder textueller Programmierung lässt. Die eingesetze Algorithmik umfasst unter anderem **(1)** die Fourieranalyse zum Umwandeln von Klängen in ihre einzelnen Partialtöne, **(2)** "Instance-Rendering", bei der eine einzelne Geometrie

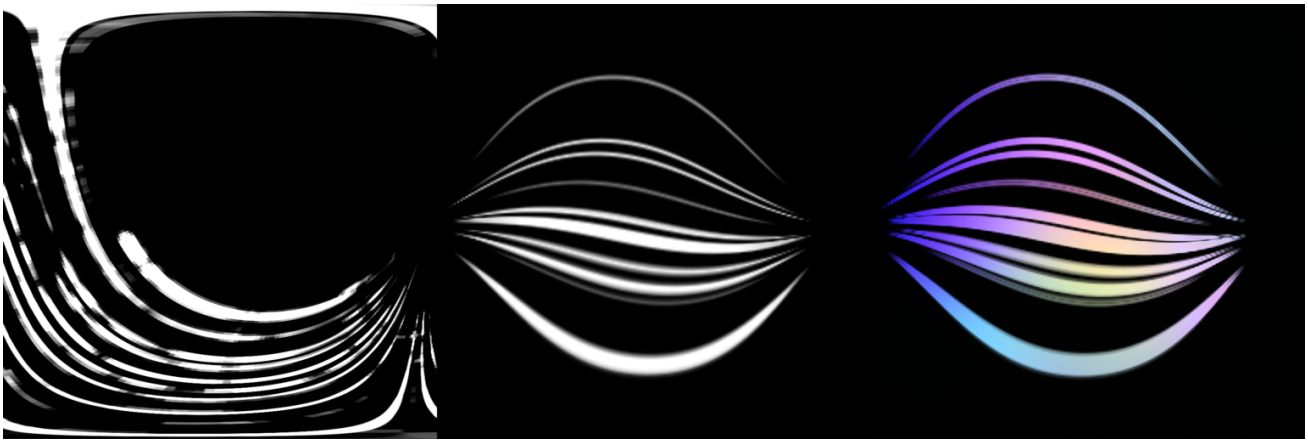
Audiosignal (stereo) der Eingabemusik. Pro Zeitschritt wird die Mischung der in der Musik enthaltenen Töne abgebildet (Sample). Eine Fourrier-Transformation erlaubt das Differenzieren der ursprünglichen Tonfrequenzen in dieser Mischung.

als “Instanz” mehrfach mit verschiedenen Eigenschaften (Transformation, Farbe) dargestellt werden kann und **(3)** Projektionen, die allerdings nichts mit klassischen Lichtprojektion zu tun haben.

Die Reise der “Momentum Partikel”

Für die Erstellung der Skulptur werden Daten verarbeitet – genauer das Musikstück “Momentum”. Zu Beginn liest das Programm den Song ein, um die darin codierte Tonspur auszuwerten. Achtung, jetzt wird es kompliziert: **(1)** Mittels Frequenzanalyse über eine Fouriertransformation, also eine mathematische Methode, entstehen zwei Spektralbilder; eines des aktuellen Tonspektrums (S_aktuell) und zusätzlich eines der Tonverteilung über ein Zeitintervall (S_interval) von 512 Samples. Diese beiden Bilder werden per Plate-Carrée-Projektion in ein Kugelkoordinatensystem projiziert, wodurch vereinfacht gesagt verschnörkelte Spektralbilder entstehen. Zusätzlich zu diesen Spektralanalysen berechnet das Programm eine Audioanalyse, um markante Punkte für tiefe, mittlere und hohe Tonbereiche herauszufiltern, die später als “Trigger” für weitere Aktionen dienen. Damit gibt es ein Grundset von Elementen, die sich für die Visualisierung nutzen lassen.

Das der Skulptur zugrundeliegende Partikelsystem besteht aus zwei Datenfeldern, genannt Texturen. Eines für die dreidimensionale Position und eines für einen Geschwindigkeitsvektor. Diese Texturen haben eine quadratische Größe von 128 mal 128 Pixeln mit jeweils vier Kanälen (Rot/Grün/Blau/Opazität), dadurch lassen sich 16.384 visuelle Partikelinformationen gleichzeitig berechnen. Welche Farbe hat der Partikel, wie satt leuchtet er oder ist er gar durchsichtig? Ein sogenanntes *Compute Shader*-Programm berechnet zusätzlich aus den Texturen für jeden Zeitpunkt Partikelposition und Geschwindigkeit aufgrund der Werte davor. Dies sorgt für den Eindruck von Bewegung, es entsteht eine Partikel-Simulation. Jeder Partikel hat dabei eine Lebenszeit, in der er sich sichtbar bewegt. Diese beginnt mit einer zufälligen Startlebenszeit im Intervall von null bis eins, etwa 0,6. Die Lebenszeit eines Partikels wird dann immer weiter um einen minimalen Faktor verringert – der Partikel verwelkt – bis sie von 0,6 bei null angekommen ist. Dann wird die Lebenszeit zurück auf Startposition gesetzt.



links: Zeitliches Tonspektrum, mitte: Tonspekturm des aktuellen Musiksamples, rechts: Farbgebung der Partikel verrechnet mit dem Tonspektrum des aktuellen Musiksamples

Von der Datenverarbeitung zur Visualisierung

Diese Startposition ist ebenfalls dynamisch und wird durch die zuletzt gehörten Töne gesteuert. Die Geschwindigkeit eines Partikels wird auf Basis der aktuellen Position mit einer Curlnoise-Funktion verrechnet, dadurch sollen organische Verwirbelungen entstehen. Die Länge des Geschwindigkeitsvektors wird außerdem von S_aktuell bestimmt. Das heißt: ist im auslösenden Trigger-Spektrum gerade kein Ton zu hören, ist die Geschwindigkeit sehr gering; je höher der Ausschlag im Spektrum für eine entsprechende Tonfrequenz, desto schneller wird der Partikel. Die ganze Simulation wird also intensiver, wenn die Musik anschwillt. Betrachtende erleben Musik und Simulation als Einheit.

Um jeden einzelnen Partikel darzustellen, wird **(2)** Instance-Rendering verwendet – pro Partikelinformation wird immer derselbe Quader dargestellt, allerdings durch unterschiedliche Transformationen an andere Orte verschoben, gedreht und skaliert. Hier kommt

erneut das anfänglich erarbeitete Grundset an Elementen und Tiggern zum Einsatz. So wird zum Beispiel die Ausdehnung des Quaders durch den Parameter der Tonanalyse bestimmt.

Wie bei einer physischen Skulptur kommt der Feinschliff zum Schluss: **(3)** Das Farbspektrum der Partikel wird über eine vierdimensionale Perlinfunktion bestimmt und mit dem Spektrum S_aktuell verrechnet. Durch ein *Mapping* (ein Verfahren zum "Übereinanderlegen" verschiedener Datenelemente) entsteht eine Art Glühen der Partikel.

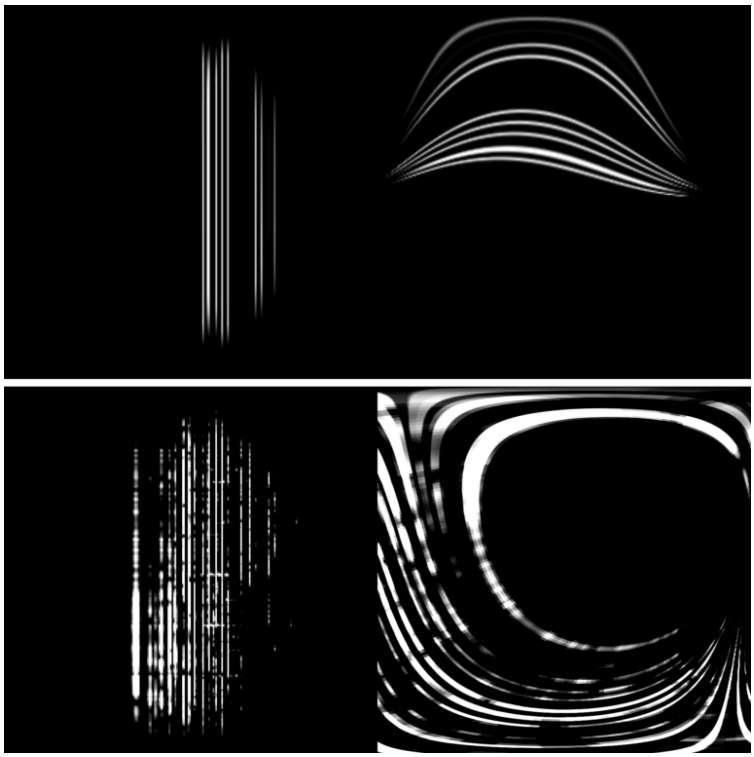


Plate-Carrée-Projektionen der Audiospektra (links: unverarbeitetes Spektrum, rechts: projiziertes Ergebnis), oben: Projektion des Tonspektrums eines einzelnen Samples, unten: Projektion des Tonspektrums eines zeitlichen Intervalls von Samples

Der Code und die Kunst

Die *Generative Art* scheint unserem gewohnten Bild der Kunst zu widersprechen. Sie ist unbestreitbar eine informatische Praktik, durch und durch technisch und präzise. Und das ist auch gut so: Code und Programmierung ist ihr zentrales Merkmal, mit der sie innovative und überraschende künstlerische Beiträge schaffen kann.

Es mag überraschend klingen, aber diese Eigenschaft eröffnet einen sehr offenen, partizipativen Zugang zu künstlerischem Schaffen: Interessierte finden im Internet haufenweise Anleitungen, die Bilder von Nees, Molnár und anderen einfach mit heutigen Mitteln nachzuprogrammieren. Der Code vieler Kunstwerke und Gestaltungsverfahren lässt sich bei Github einsehen und weiterverwenden. Im Zeitalter von Social Media ist aber auch der Zugang zur *Generative Art* als Betrachter*in sehr niedrigschwellig: unter Hashtags wie #generative findet sich auf Instagram eine riesige Bibliothek aktueller Beiträge. Szenegrößen wie Matt DesLauriers oder Zach Liebermann teilen regelmäßig Serien. Und der Digitalkünstler Felix Faire hat sich auf Musik- und Soundvisualisierungen spezialisiert, wie sie auch im Projekt *vi·son* entwickelt werden.

Jason Bailey beschreibt überdies, dass nach seinen Beobachtungen die Entwicklungsprozesse der *Generative Art* rund um Assoziation und Zufall sehr ähnlich zur analogen Kunst ablaufen. Dem kann man sicher zustimmen. Gleichzeitig scheint es diskussionswürdig, ob eine solche Gegenüberstellung überhaupt notwendig ist, sind es doch nicht die methodischen Gemeinsamkeiten im kreativen Schaffen, sondern die strukturellen Unterschiede des digitalen Raums, die die Faszination der *Generative Art* ausmachen und mit der sie das Kunsterleben bereichert.

***vi·son* wird eine audioreaktive Digitalkunst-Performance**. Im Projekt werden verschiedene Phänomene rund um die multisensorische Wahrnehmung thematisiert. Dabei liegt das Hauptaugenmerk auf der **Visualisierung von Musik**. Im November 2020 soll eine Online-Ausstellung gestaltet werden, aus der dann im Frühjahr 2021 eine Umsetzung in der Region Rhein-Neckar und Frankfurt entsteht. Zur Projekt-Homepage geht es hier. Bis zur digitalen Ausstellungs-Eröffnung im November erscheinen in regelmäßigen Abständen kunst- und sozial-theoretische Annäherungen an das Projekt auf **the ARTicle**.

vi·son ist eine Kooperation zwischen den Initiativen **KALANGU** und **Pendeloque**. KALANGU ist ein Künstler-Pseudonym unter dem überwiegend musikalische Projekte entstehen. Dahinter stecken wechselnde Studio- oder Livebesetzungen. Die Lichtkunstgruppe Pendeloque setzt sich mit der Kombination von Lichtkunst mit anderen Ausdrucksformen (wie Theater, Musik oder interaktiven Medieninstallationen) auseinander. Daraus entstanden bereits verschiedene Projekte, beispielsweise die Installation Anima\S in Mannheim. *vi·son* soll die Kompetenzen und Leidenschaften der beiden Initiativen zusammenbringen: daher ***vi*** für den visuellen Aspekt, der hauptsächlich von **Pendeloque** getragen wird. Und ***son***, das französische Wort für Klang und Ton – hier wird es exklusive Musik von **KALANGU** geben.

Mehr über die inhaltliche Ausrichtung des Digitalkunst-Projekts *vi·son* erfährst du hier:

1. Bailey, Jason (2018): Why Love Generative Art? URL: https://www.artnome.com/news/2018/8/8/why-love-generative-art (21.07.2020, 14:00) [↩]
2. Ebd. [↩]
3. Ebd. [↩]