

# Coded Beauty

Die generative Methode als ästhetische  
Praxis der Computergrafik



# Coded Beauty

Die generative Methode als ästhetische Praxis der Computergrafik

## Diplomarbeit

eingereicht an der

FH Joanneum

Fachhochschulstudiengang Informations-Design

vorgelegt im September 2008 von

Thomas Kräftner

Franz Broschgasse 17

2380 Perchtoldsdorf

Personenkennzeichen: 0410086020

Betreuer

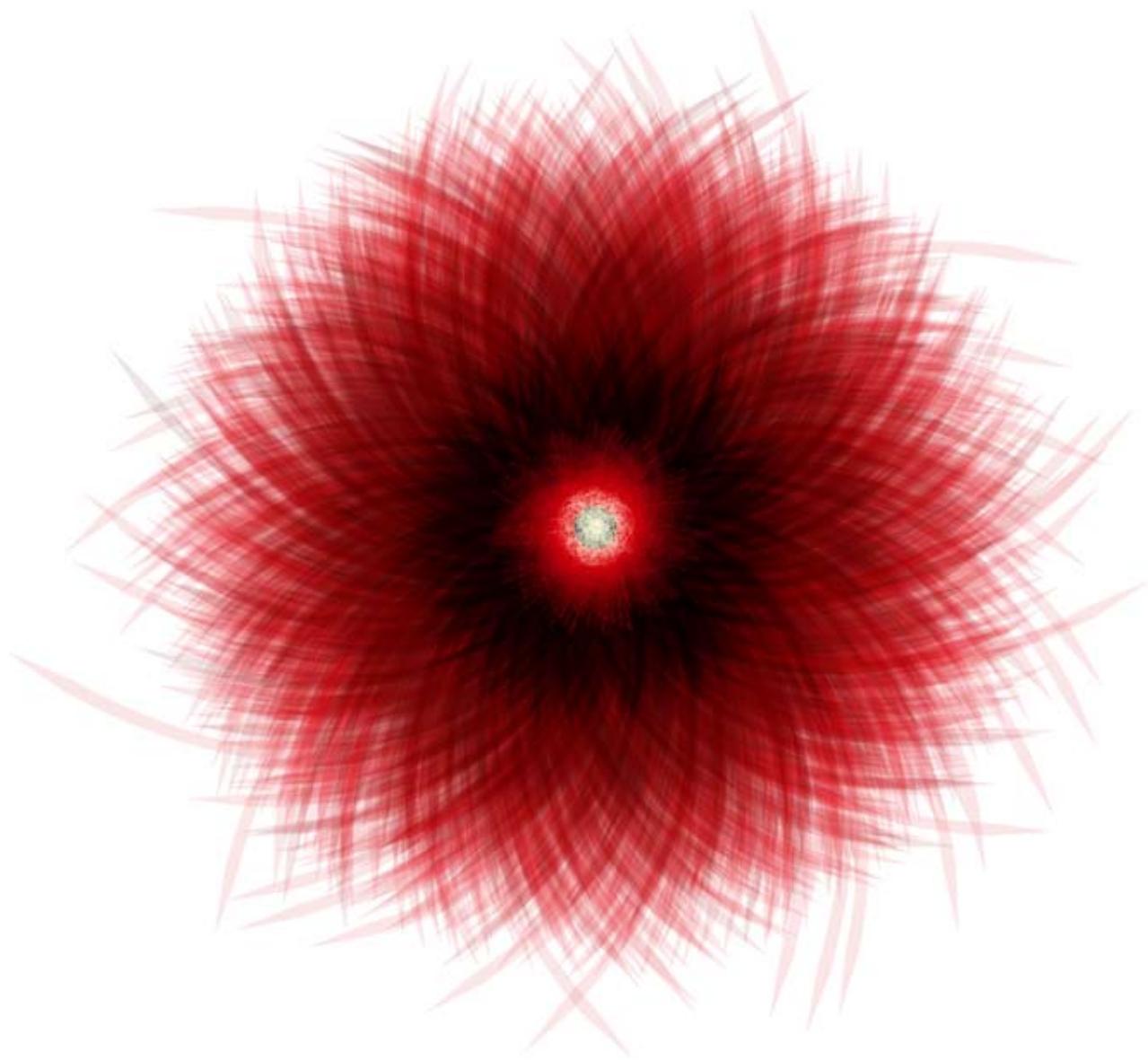
Dipl.-Ing. Jochen Martin



# Eidesstattliche Erklärung

Ich erkläre hiermit eidesstattlich, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Perchtoldsdorf, am 25.9.2008



# Danksagung

Vielen Dank meiner Familie für ihre Unterstützung während den langen Monaten des Entstehens meiner Diplomarbeit, im Besonderen an meine Schwester und meine Mutter für das Korrekturlesen. Vielen Dank meiner Freundin Daisy für ihre Geduld wenn mich die Diplomarbeit gestresst hat, ihre regelmäßige Motivation und ihre kritischen Kommentare. Vielen meiner Freunde gebührt Dank für ihr Feedback, besonders Maria für das Korrekturlesen und die damit verbundenen hilfreichen Verbesserungsvorschläge. Danke ebenfalls an meinen Betreuer Dipl.-Ing. Jochen Martin für seine Unterstützung und Kritik, die mir aber immer genug Freiraum gelassen hat „mein Ding“ durchzuziehen.

Danke auch an Serena dafür, dass sie mir einen anderen Blick auf Kunst, die kleinen Dinge und „garbage“ eröffnet hat. Danke an Ulrike Schitter, besser bekannt als lia<sup>1</sup>, dafür dass sie mir in ihren Vorlesungen auch gegen den mehrheitlichen Widerstand meines Jahrgangs eine für mich damals neue Welt eröffnet hat. Ohne diesen Einblick wäre ich auf die Idee zu dieser Diplomarbeit wahrscheinlich niemals gekommen.

Nicht vergessen werden dürfen auch jene großzügigen Menschen die ihre Arbeit als Open Source kostenlos zur Verfügung gestellt haben und damit den Entstehungsprozess des Werkstücks dieser Arbeit deutlich vereinfacht haben: Vielen Dank Keith Peters<sup>2</sup>, Grant Skinner<sup>3</sup> und dem gesamten FlashDevelop Team<sup>4</sup>.

---

1 <http://www.strangethingshappen.org>

2 <http://www.bit-101.com>

3 <http://www.gskinner.com/blog>

4 <http://www.flashdevelop.org>

◀ Abb. 1: Thanks, mit dem Werkstück generiert (Thanks.xml)

# Kurzfassung

Vom Begriff des Algorithmus ausgehend wird in dieser Arbeit eine allgemeine Definition der generativen Methode vorgestellt. Darauf aufbauend wird anschließend eine nähere Betrachtung im Kontext von *New Media* abgeleitet. Es werden dabei die Zusammenhänge und Diskrepanzen zwischen Technik und Kreativität, sowie die Bezüge zu Kunst und Design behandelt. Dabei wird klar, dass die generative Methode im Computerzeitalter von wachsender Bedeutung ist. Denn auch wenn der Erstellungsprozess in fast allen Medienbereichen bereits in digitaler Form stattfindet, hinken die Methoden den technischen Möglichkeiten noch weit hinterher und sind noch viel zu sehr im Vor-Computerzeitalter verhaftet.

Aufbauend auf den eingeführten Begriffen und den gewonnenen Erkenntnissen wird das eigentliche Kernthema behandelt – die generative Computergrafik.

Da die generative Computergrafik kein Gestaltungsstil sondern eine Methode ist, impliziert sie auch keine konkrete ästhetische Form. Dennoch zeigen sich bei den Anwendern einige gemeinsame Herangehensweisen und Inspirationsquellen. Diese finden sich in der Natur, der Mathematik und den Naturgesetzen, sowie in Kunst und Malerei.

Im Anschluss an diese gestalterischen Überlegungen werden die technischen Grundlagen und Anforderungen zur Erstellung generativer Grafiken behandelt. Verschiedene Technologien (Processing, Flash, vvvv) werden evaluiert und darauf aufbauend schließlich die Entscheidung getroffen, Flash für das Werkstück zu dieser Arbeit einzusetzen.

Danach werden, bereits mit besonderem Augenmerk auf jene, die Flash einsetzen, exemplarisch einige Vertreter der generativen Arbeitsweise vorgestellt.

Da sich der Nachweis der Nutzbarkeit und der Praxisrelevanz als notwendig erweist, werden mögliche Anwendungsgebiete für generative Grafik aufgezeigt. Die Datenvisualisierung wird dabei als ein exemplarisches Beispiel im Detail vorgestellt.

Der Schluss dieser Arbeit widmet sich schließlich der praktischen Anwendung der zuvor gewonnenen Erkenntnisse in Form eines Werkstücks. Zu diesem Zweck werden konkrete technische Anforderungen definiert und die unter Einsatz von Flash notwendigen Elemente und der Aufbau des Werkstücks festgelegt. Die darauf folgenden Beispiele zur Anwendung, sowie die dieser Arbeit in digitaler Form beiliegende technische Dokumentation des Werkstücks, machen die Erkenntnisse dieser Arbeit schließlich für den Leser persönlich nachvollziehbar.

# Abstract

Starting from the term Algorithm this paper presents a general definition of the generative method. Based on this definition a closer examination is derived, within the context of New Media. Thereby connections and discrepancies between technology and creativity, as well as references to art and design, are being examined. It will become clear that the generative method has increasing relevance in the computer age. Even though the creation process is already digitally informed in much of the media sector, it remains bound to the pre-computer era, and has yet to develop its technological potential.

This thesis then examines generative computer graphics, based on established terms and research findings.

The field of generative computer graphics describes a method rather than referring to a certain style or implying a concrete aesthetic form. Despite the broad scope of it, designers and artists share approaches and sources of inspiration; nature, mathematics and laws of nature, as well as art and painting.

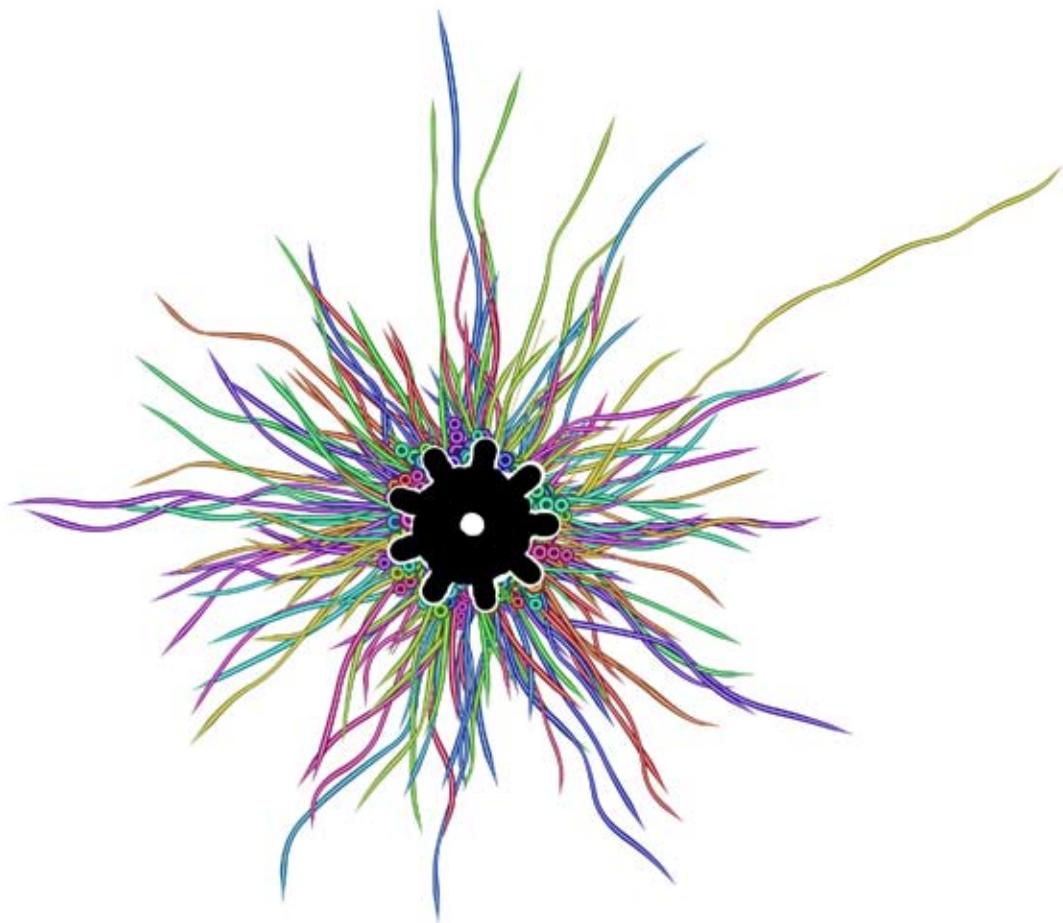
Subsequent to the creative and conceptual thoughts, this thesis also examines the technical principles and requirements of creating generative graphics. Different technologies (Processing, Flash, vvvv) are evaluated; based on the research findings, Flash has proven to be the most appropriate platform for the workpiece.

Although generative graphics can be created using a variety of methods, this thesis will focus on Flash-users, and present their works as examples.

A prime example of the practical application of generative graphics is the field of Data Visualization. This example showcases the relevance and technological possibilities for the use of generative graphics.

This paper concludes with a focus on the practical application of the gained findings. For this purpose concrete technical requirements are defined and the necessary elements for the use of Flash are assigned

and described. The following examples of application, as well as the enclosed digital documentation of the workpiece, make the insights of this thesis comprehensible and reproducible for the reader.



# Vorwort

Es ist nicht einfach zu sagen, woher die Idee zu dieser Arbeit kam. Seit ich mich mit ihnen beschäftige, hat es mich gestört, dass Computer viel zu oft auf ihre Eigenschaft als bloße Rechenmaschine reduziert werden. Als neues Medium sollte der Computer genau das gleiche Spektrum an möglicher kultureller Bedeutung abdecken wie andere Medien. Dazu gehören natürlich auch Ästhetik, Kunst und Design, welche neben meiner Begeisterung für Technik meine zweite große Motivation für diese Arbeit darstellen. Gerade die phantastischen Möglichkeiten, die sich ergeben, wenn man diese zwei oftmals getrennten Bereiche vereint, haben schließlich zu dieser Arbeit geführt. Oder in Mark Tribe's Worten:

„We monkey around with new technologies in an effort to see what they can do, to make them do things the engineers never intended, to understand what they might mean, to reflect on their effects, to push them beyond their limits, to break them.“ (Manovich, Media, XI)

Immer schon waren Computer für mich nicht nur einfach Werkzeuge, sondern bis zu einem gewissen Grad magische Wundermaschinen und Spielzeug. Gerade diese Freiheit des Experimentierens und Spielens mit der Technologie ist es, die mich an der generativen Methodik so fasziniert. Im Gegensatz zu klassischem Design weiß man dabei nicht immer exakt, was am Ende herauskommen wird, Fehler sind oft das Beste, was einem passieren kann, und man muss nicht immer von Anfang an ein bis ins letzte Detail geplantes Konzept haben.

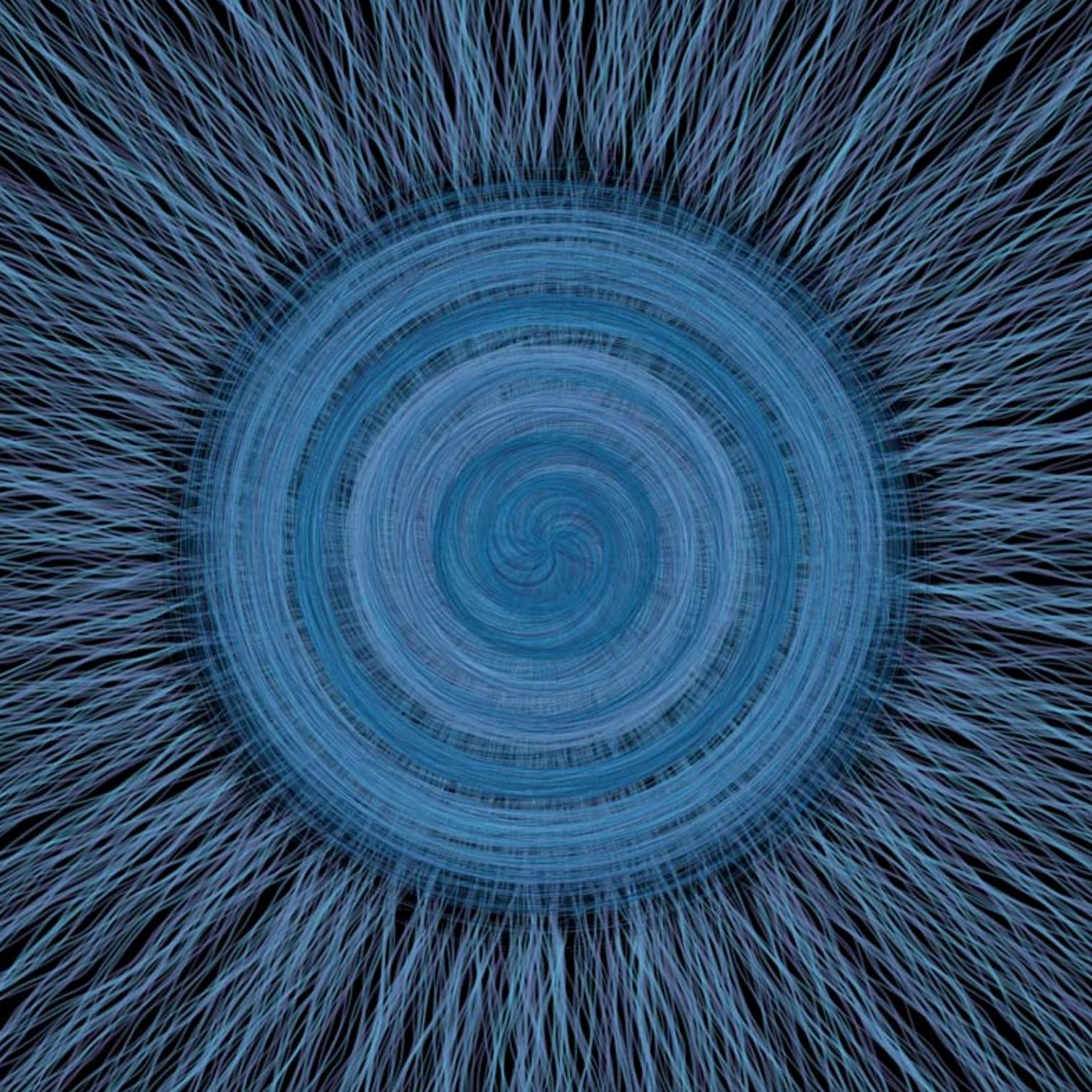
Die Faszination liegt also in der Gratwanderung zwischen Programmierung, Gestaltung, Design, Kunst und den unglaublichen Synergien, die sich durch die Kombination dieser Bereiche ergeben können.

◀ Abb. 2: Broken Technology, mit dem Werkstück generiert (BrokenTechnology.xml)

# Inhaltsverzeichnis

Eidesstattliche Erklärung.....	V
Danksagung.....	VII
Kurzfassung.....	VIII
Abstract.....	X
Vorwort.....	XIII
1. Einleitung.....	1
2. Begriffsdefinitionen und Arbeitsparadigmen.....	5
2.1. Der Algorithmus.....	5
2.2. Die generative Methode.....	6
2.3. New Media.....	9
2.4. Die generative Methode im Kontext von New Media.....	12
2.5. Technik und Kreativität.....	17
2.6. Kunst oder Design.....	19
3. Generative Computergrafik.....	21
3.1. Gestalterische Arbeitsweise und ästhetische Kriterien.....	22
3.1.1. Natur.....	23
3.1.2. Mathematik und Naturgesetze.....	24
3.1.3. Kunst, Malerei.....	25
3.2. Technische Grundlagen und relevante Technologien.....	28
3.2.1. Existierende Tools und ihre Einschränkungen.....	28
3.2.2. Technische Anforderungen.....	29
3.2.3. Relevante Technologien.....	30
3.2.4. Auswahl der Technologie für das Werkstück.....	37

4. Die Szene – exemplarische Vertreter der generativen Arbeitsweise .....	39
4.1. Yugo Nakamura .....	40
4.2. Joshua Davis .....	42
4.3. Mario Klingemann .....	44
4.4. Marius Watz .....	46
4.5. Jonathan Harris .....	47
4.6. Zusammenfassung.....	49
5. Mögliche Anwendungsgebiete .....	51
5.1. Datenvisualisierung.....	52
6. Das Werkstück.....	55
6.1. Anforderungen .....	55
6.2. Aufbau .....	56
6.2.1. Geometrische Grundformen - Zeichnungsfunktionen .....	56
6.2.2. Farbmanagement.....	57
6.2.3. Exportfunktionen .....	57
6.2.4. GUI.....	57
6.2.5. Organisations-Framework .....	58
6.3. Verwendung des Werkstücks .....	59
7. Conclusio .....	63
8. Anhänge .....	67
8.1. Literaturverzeichnis.....	67
8.2. Abbildungsverzeichnis .....	72



# 1. Einleitung

Die Computerisierung unserer Welt ist heute ein nicht mehr wegzudenkender Faktor unseres täglichen Lebens und damit auch unserer Kultur. Selbstverständlich hat sie daher auch schon Einzug in die gestalterische Praxis gehalten. Doch in welcher Form? Sind die derzeit gebräuchlichen gestalterischen Methoden noch zeitgemäß oder gibt es Herangehensweisen, die besser geeignet sind das volle Potential von New Media zu nützen?

Die Verbreitung des Computers hat in einem außergewöhnlich schnellen Prozess ein vollkommen neues Medium geschaffen. Die ästhetische Praxis und die gestalterischen Methoden sind jedoch noch sehr stark im Vor-Computerzeitalter verhaftet. Dies zeigt sich beispielsweise darin, dass gängige Bildbearbeitungsprogramme immer noch Metaphern wie die des Pinsels oder des Radierers verwendet.

Ziel ist es daher, mit der generativen Methode einen Ansatz zu präsentieren, der den spezifischen Eigenschaften und den daraus resultierenden Möglichkeiten von *New Media* gerecht wird und deren Potential in einem neuen Ausmaß nutzbar macht.

Zu diesem Zweck befasst sich diese Arbeit mit den theoretischen Grundlagen zur Erstellung generativer Grafiken, sowie mit Beispielen und Anwendungsszenarien. Auch wenn die generative Methode als Grundlage im Allgemeinen behandelt wird, ist das besondere Augenmerk der Arbeit auf den Teilbereich der generativen Computergrafik gerichtet.

Aufgrund der komplexen Querbezüge zu Bereichen wie Informatik, Kunst, Design, Geschichte und Medientheorie wird es oftmals nicht möglich sein, alle Zusammenhänge und Abhängigkeiten bis ins letzte Detail zu behandeln. Was daher nicht geboten werden kann, sind unter anderem konkrete Programmieranleitungen, ausführliche Erläuterungen der (kunst)historischen Entwicklung sowie um-

◀ Abb. 3: Malstrom, mit dem Werkstück generiert (Malstrom.xml)

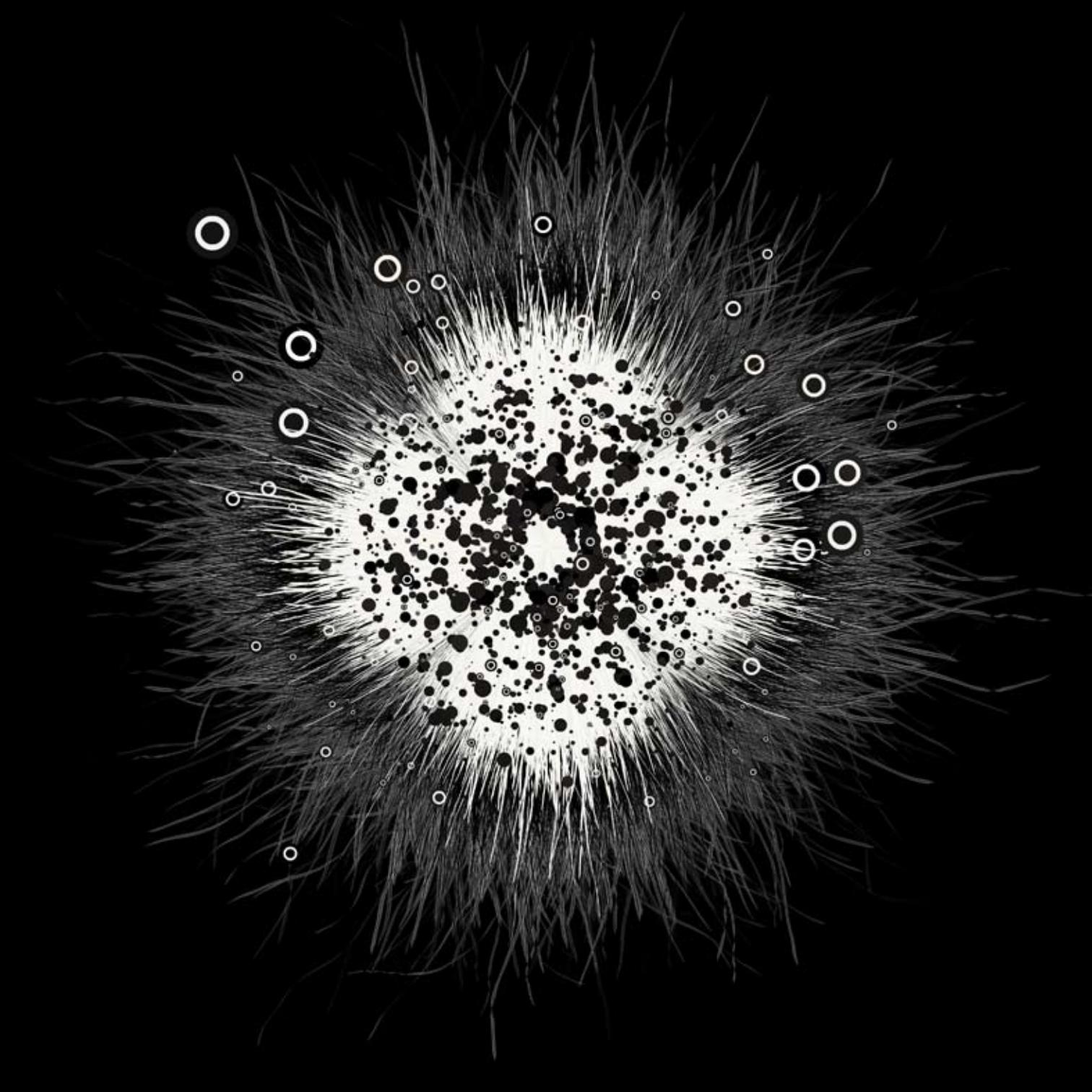
fassende Anwendungsbeispiele für die generative Methode. Im Zentrum dieser Arbeit steht vielmehr, aufbauend auf die Begriffe Algorithmus und *New Media*, die Herleitung einer Definition der generativen Methode und damit der generativen Grafik. Besonders die dabei auftauchende Problematik der Barrieren zwischen den technischen Ursprüngen und dem kreativen Potential von Computern wird dabei einer genauen Betrachtung bedürfen. Ziel soll es schließlich sein, die Erkenntnisse dieser Arbeit in Form eines beiliegenden Werkstücks konkret zu demonstrieren.

Der Aufbau dieser Arbeit orientiert sich vereinfacht gesagt an der Beantwortung der vier grundlegenden Fragen „Was? Wie? Wer? und Wofür?“.

In Kapitel 2 wird erklärt, was man unter generativ und der generativen Methode im Allgemeinen versteht, und es werden alle für diese Arbeit notwendigen und zu Grunde liegenden Begriffe und Paradigmen definiert. Kapitel 3 widmet sich anschließend dem „Wie?“. Es wird im Detail erklärt, auf welchen Prinzipien die generative Computergrafik basiert. Dabei widmet sich Kapitel 3.1 den gestalterischen Herangehensweisen, Inspirationen und ästhetischen Gesichtspunkten in Form von Bezügen zu Natur, Mathematik und Naturwissenschaften sowie zu Kunst. Verfügbare Technologien sowie ihre Eignung für das Werkstück werden in Kapitel 3.2 näher beleuchtet. Das „Wer?“ wird in Kapitel 4 anhand konkreter Beispiele von Vertretern des generativen Schaffens näher unter die Lupe genommen.

Kapitel 5, das letzte Kapitel des theoretischen Teils, befasst sich mit dem „Wofür?“, den möglichen Anwendungsgebieten des generativen Konzepts, und es liefert somit die Basisüberlegungen für das Werkstück und stellt das Bindeglied zu diesem dar. In Kapitel 6 werden dann die relevanten Überlegungen für dessen Erstellung erläutert. Hierfür werden zuerst die Anforderungen definiert, notwendige Grundelemente festgelegt und konzeptioniert und anschließend die Struktur und das Konzept für das finale Werkstück erarbeitet. Dessen ausführliche (technische) Dokumentation befindet sich in durchsuchbarer, digitaler Version auf der beiliegenden DVD sowie auf der Webseite zu dieser Arbeit unter <http://codedbeauty.kraftner.com>. Abschließend werden in Kapitel 7 die gewonnenen Erkenntnisse,

sowohl jene theoretischer als auch jene praktischer Natur, in einer abschließenden Conclusio zusammen-gefasst.



## 2. Begriffsdefinitionen und Arbeitsparadigmen

Zur Behandlung der generativen Computergrafik ist es notwendig einige zu Grunde liegende Begriffe zu definieren, missverständliche Begriffe zu klären und die generellen Rahmenbedingungen festzulegen.

### 2.1. Der Algorithmus

Anfang des 20. Jahrhunderts fand sowohl in der Wissenschaft, als auch einige Jahre später in der Kunst die Algorithmische Revolution statt, welche sich bis heute maßgeblich auf unsere Welt auswirkt. Revolutionen sind im Allgemeinen nicht zu übersehen, doch diese Revolution fand leise, jedoch mit umso bedeutenderen Auswirkungen auf unsere gesamte Welt statt. Es gibt kaum einen Bereich unserer Welt, in dem Algorithmen noch keinen Einzug gehalten haben. (Weibel, Revolution, online)

Was ist also ein Algorithmus? Wie für die meisten Begriffe mit einer großen Tragweite gibt es auch für den Begriff des Algorithmus unzählige Definitionen.

Hier seien nun exemplarisch zwei Definitionen erwähnt:

Algorithmus, eine Folge von Anweisungen (Rechenschritten), die einen Prozess definieren, der mit gewissen Dateneingaben beginnt und nach endlicher Zeit ein durch die Eingabedaten eindeutig bestimmtes Resultat liefert. (Maus, Algorithmus, online)

Diese Definition ist bereits eher spezifisch in Hinsicht auf die Anwendung in Mathematik und Informatik formuliert, doch lässt sich auch eine allgemeinere Definition finden, welche sich für die folgende Definition der generativen Methode als vorteilhafter erweist.

◀ Abb. 4: Monochrome Supernova, mit dem Werkstück generiert (MonochromeSupernova.xml)

Unter einem Algorithmus versteht man eine Entscheidungsprozedur, eine Handlungsanweisung, die aus einer endlichen Menge von Regeln besteht, eine endliche Folge von eindeutig bestimmten Elementaranweisungen, die den Lösungsweg eines spezifischen Problems exakt und vollständig beschreiben. (Weibel, Revolution, online)

Auch wenn von Algorithmen, wie man schon bei der ersten Definition bemerkt, primär im Kontext von Mathematik und Informatik gesprochen wird, findet dieses Prinzip auch in vielen anderen Bereichen, wenn auch oft unter anderem Namen, seine Anwendung. Der Begriff ist somit intuitiv gegeben. Man denke nur an alltägliche Handlungsanweisungen und Prozessbeschreibungen wie Kochrezepte, Baupläne, Gebrauchsanweisungen, Wegbeschreibungen usw.

Algorithmen bilden nicht nur die Grundlage für die generative Computergrafik, sondern das Fundament für die gesamte auf Computerprogrammierung basierende Technik.

## 2.2. Die generative Methode

Unter der generativen Methode versteht man im Allgemeinen eine Vorgehensweise, die auf Regeln und reglementierten Prozessen basiert. Im Gegensatz zu einem klassischen Erstellungsprozess wird beim generativen Arbeiten nicht direkt am Ergebnis gearbeitet. So malt man beispielsweise nicht direkt wie z.B. beim Malen mit einem Pinsel anhand einer Vorstellung intuitiv auf eine Leinwand, sondern der Erstellungsprozess beginnt viel mehr mit der Definition von Regeln und Abläufen, wobei die eigentliche Erstellung anschließend strikt gemäß dieser Regeln abläuft. Diese Regeln und Abläufe sind nichts Anderes als das was im vorherigen Kapitel als Algorithmus definiert worden ist.

Aufbauend auf diesen Begriff lässt sich nun eine genauere Beschreibung der generativen Methode erstellen. Auch wenn sie hier im Kontext von Kunst formuliert ist, kann man diese Definition auch auf jedes andere Anwendungsgebiet umlegen.

Generative art refers to any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art. (Galanter, Generative, 4)

Was schon an diesem Punkt sehr klar wird ist, dass es um eine Herangehensweise, eine Methode, und nicht um eine konkrete Ausdrucksform geht. Generative Methoden können und werden daher auch in vielen verschiedenen Disziplinen wie Musik<sup>1</sup>, Design, Kunst, Literatur oder Architektur angewendet.

Ein essentieller Bestandteil der generativen Methode ist, wie man auch in der Definition von Philip Galanter sehen kann, die Automation. Ein System muss ein automatisch laufendes, mehr oder weniger autonomes System sein um als generativ bezeichnet werden zu können. Dabei stellt sich natürlich sofort die scheinbar naheliegende Frage, wo bei einem automatisch ablaufenden autonomen System noch die Leistung des Erstellers bleibt. Dieser Vorwurf entpuppt sich bei näherer Betrachtung jedoch als Vorurteil.

Dieses Missverständnis ist wohl eines der ältesten bei allen automatisierten Vorgängen, wie etwa bei jeder Arbeit mit Computern, und natürlich tritt es besonders dann auf, wenn man generativ arbeitet. Nur weil ein Computer etwas automatisch generieren kann, ist er noch lange nicht in der Lage, selbstständig zu arbeiten und den Menschen tatsächlich zu ersetzen. Im Endeffekt läuft diese Problematik auf das „Infinite monkey theorem“ hinaus. Dieses besagt, dass ein Affe an einer Schreibmaschine, wenn man ihn nur lange genug tippen lässt irgendwann die gesammelten Werke von Shakespeare schreiben wird. (Vgl. Infinite Monkey Theorem, Wikipedia) Das Problem dabei ist nur, wer zwischen den Unmengen von Belanglosigkeiten und „Informationsmüll“, die er vorher getippt hat, die Meisterwerke finden würde? John Simon bringt die Problematik auf den Punkt:

---

1 Für eine Betrachtung der generativen Methode im Rahmen von Musik empfiehlt sich der Artikel Ihmels, Tjark: Die Methodik der generativen Kunst, [http://www.medienkunstnetz.de/themen/generative\\_tools/generative\\_art](http://www.medienkunstnetz.de/themen/generative_tools/generative_art), zuletzt aufgerufen am 10. 8.2008

Ich habe begriffen, dass man mehr Bilder erzeugen kann – mehr als man jemals sehen beziehungsweise als der Computer jemals darstellen können wird. Deswegen wird man auch weiterhin kreative Menschen als Bild-Erzeuger brauchen, die in der Lage sind, das entscheidende Signal aus dem ganzen Rauschen herauszufiltern. (Simon, Kreativität, 46)

Die Frage, die man sich also beim Einsatz der generativen Methode immer wieder stellen muss, ist die Frage nach dem Grad der Automatisierung. Um dies zu erläutern zuerst ein Zitat:

Die konsequente Anwendung eines vordefinierten Handlungsprinzips zum bewussten Ausschluss oder als Ersatz individueller ästhetischer Entscheidungen setzt die Generierung [...] in Gang. (Ihmels, Methodik, online)

Das Missverständnis, das hier leicht passiert ist, dass man denken könnte, der generelle Ausschluss aller ästhetischer Entscheidungen sei das Ziel. Diese werden aber eigentlich nur verlagert, und zwar vom Moment der eigentlichen Erstellung hin zur vorherigen Auswahl der Regeln, sprich Algorithmen. Der entscheidende Punkt liegt also bei der individuellen Entscheidung. Ästhetische Entscheidungen werden nicht aus dem Prozess entfernt, weil dies wegen der zuvor genannten Problematik des Filterns des Rauschens nicht möglich bzw. nicht effektiv ist. (Vgl. Wands, Digital, 171) Sie werden nur an eine andere frühere Stelle verschoben, von der Ausführung zur Konzeption hin, womit man dann natürlich auch einen weniger mittelbaren Einfluss auf die Ästhetik hat. Es kommt zu einer Verschiebung von expliziten ästhetischen Entscheidungen zu impliziten.<sup>2</sup>

---

2 Konkret veranschaulicht hat Mario Klingemann diese Tatsache mit seinem Programm „Sketchmaker“, welches anhand verschiedener (optischer) Kriterien Bildern eine Wertung von „keine Kunst“ bis „Kunst“ zuteilt. Auf diesem Lernprozess basierend ist das Programm in der Lage mehr oder weniger selbständig zu lernen und Bilder zu generieren, welche es wiederum nach diesen Kriterien bewertet. Wir driften hier bereits in den Bereich der Künstlichen Intelligenz (Artificial Intelligence oder AI) ab. Wer sich genauer mit dieser Thematik befassen möchte, dem ist der Vortrag „The Blind Sketchmaker“ von Mario Klingemann zu empfehlen. Klingemann, Mario: The Blind Sketchmaker. Exploring Generative and Evolutionary Art with Flash, [http://lectures.quasimondo.com/the\\_blind\\_sketchmaker](http://lectures.quasimondo.com/the_blind_sketchmaker)

Am besten lässt sich die Art des Einflusses wahrscheinlich über den Unterschied zwischen einem Puppenspieler und einem Schauspieler verstehen. Beide haben Einfluss auf die Darstellung eines Charakters, der Puppenspieler, analog zum Einfluss des Erstellers eines generativen Systems hat durchaus Einfluss auf das Ergebnis, aber weniger direkt.

Was die Automation an sich, die Ausführung des Algorithmus, betrifft, kann der Ablauf eines generativen Systems auf unzählige verschiedene Weisen geschehen, sei es z.B. durch die manuelle Produktion eines vorgegebenen systematisch aufgebauten Musters, den Einsatz einer selbst laufenden Maschine oder die programmierte Generierung mittels eines Computers. Auch wenn die generative Methode an sich also weder maschinell noch digital oder computerisiert ist, hat sie ihre heutige Blüte dennoch dem Computer und seinen Möglichkeiten der Programmierung und Automation und vor allem seiner unübertroffenen Geschwindigkeit (sprich Rechenleistung) zu verdanken. (Vgl. Watz, *Generative*, online und Weibel, *Revolution*, online)

## 2.3. New Media

Da wir uns mit der Computergrafik im Bereich der *New Media* bewegen, sollen deren Prinzipien kurz besprochen werden, um den in dieser Arbeit im Vordergrund stehenden computergestützten Bereich der generativen Methode anschließend besser verstehen zu können.

Es gibt viele Definitionen von *New Media*. Lev Manovich hat mit seinen fünf „Principles of New Media“ eine der anerkanntesten und populärsten Definitionen geliefert. Er gliedert die charakteristischen Merkmale von *New Media* in fünf Prinzipien auf, wobei die letzten drei Prinzipien auf den ersten zwei aufbauen.

## Numerische Darstellung (Numerical Representation)

Alle neuen Medienobjekte bestehen aus digitalem Code, daher sind sie numerische Darstellungen.

Daraus ergeben sich zwei Konsequenzen:

1. Ein Medienobjekt kann formal bzw. mathematisch beschrieben werden.
2. Ein Medienobjekt kann daher mit Hilfe eines Algorithmus manipuliert werden. „Media becomes programmable.“ (Vgl. Manovich, Media, 27)

## Modularität (Modularity)

Medienelemente können zu komplexeren Objekten zusammengesetzt werden ohne ihre separate Identität zu verlieren. Diese Objekte wiederum können erneut zu einer größeren Einheit zusammengesetzt werden, wiederum ohne ihre Eigenständigkeit zu verlieren, usf.

Beispielsweise sind Webseiten aus Text, Bildern, Videos usw. zusammengesetzt, welche ein Ganzes ergeben ohne dabei ihre Eigenschaften als eigenständige Objekte zu verlieren. (Vgl. Manovich, Media, 30f)

## Automation (Automation)

Die zwei zuvor genannten Prinzipien ermöglichen die Automation der Medienerstellung, der Medienmanipulation und des Zugriffs auf Medien. Dadurch lässt sich die menschliche unmittelbare Absichtlichkeit aus dem kreativen Prozess entfernen, zumindest zu einem gewissen Anteil.

Generell gibt es drei Formen der Automation: Die „low-level automation“, welche auf Templates oder einfachen Algorithmen basiert, die „high level automation“, welche auf komplexen Algorithmen bzw.

Artificial Intelligence basiert, und schließlich den automatisierten Zugriff auf die durch die soeben genannten zwei Arten erstellten Medieninhalte. Gerade letztere ist angesichts der Unmengen an (automatisch) generierten Daten zur Zeit die bedeutendste und am meisten im Wandel befindliche Form der Automation.<sup>3</sup> (Vgl. Manovich, Media, 32-36)

## Variabilität (Variability)

Objekte der neuen Medien sind nichts Fixiertes, Endgültiges, vielmehr etwas, das in unzähligen, ja unendlich verschiedenen Versionen existieren kann und auch existiert. Auch dieses Prinzip basiert auf den ersten beiden Prinzipien, der numerischen Kodierung und der modularen Struktur von neuen Medien.

Beispiele für dieses Prinzip sind z.B.:

- Datenbanken mit ihrer Ordnung ohne definierte Abfolge.
- Die darauf basierende Unterschiedlichkeit verschiedener Interfaces zum Zugriff auf ein und dieselben Daten.
- Individualisierung durch personenbezogene Informationen.
- Verzweigte Menüstrukturen, die im Rahmen Menü-basierter Interaktion verschiedene Zugriffswege auf Informationen ermöglichen.
- Hypermedia mit seinen netzartigen Strukturen.
- (Software-) Updates mit ihrer laufenden Veränderung und Erweiterung bestehender Systeme.

---

3 Man denke nur an die bereits etablierten Suchmaschinen, relativ neue Ansätze wie Bookmarking-Plattformen, das zunehmende Tagging (Versehen von Inhalten mit Schlagwörtern), Feeds und Feed-Aggregatoren und die dynamische Informationsvisualisierung.

- Skalierbarkeit d.h. Anpassung der Darstellung ein und der selben Information an die Art der Ausgabe, „Zoomstufe“ usw. (Vgl. Manovich, Media, 36-45)

## Transkodierung (Transcoding)

Das letzte der fünf Prinzipien besagt im Grunde, dass die Computerisierung unserer Kultur Einfluss auf alle kulturellen Kategorien und Konzepte hat und sie verändert. Neue Medien sind keine Weiterentwicklung alter Medien sondern aufgrund der vollkommen neuen Eigenschaft der Programmierbarkeit etwas fundamental Neues. (Vgl. Manovich, Media, 45-48)

## 2.4. Die generative Methode im Kontext von New Media

Nicht ohne nochmals zu erwähnen dass es sich bei der Anwendung der generativen Methode im Rahmen von *New Media* nur um einen Teilbereich, wenn auch den derzeit bedeutendsten der generativen Arbeitsweise handelt, widmet sich diese Arbeit mit der generativen Computergrafik einem Teilbereich der generativen Methode, welche auf dem Fundament der Computerisierung steht und somit in unmittelbarer Nähe zu den neuen Medien anzusiedeln ist. Daraus ergibt sich auch, dass die aus Kapitel 2.2 bekannten Prinzipien in diesem Fall von Bedeutung sind.

## Numerische Darstellung, Modularität, Variabilität

Das erste Prinzip der numerischen Darstellung ist beim Einsatz von Computern offensichtlich durch die Digitalität erfüllt.

Das zweite Prinzip der Modularität ergibt sich aus dem technischen Aufbau als Programm. Einzelne Regeln und Prozesse in Form von Algorithmen lassen sich in neue Arbeiten integrieren oder auch in abgewandelter Form bzw. als Remix, womit wir bereits beim vierten Prinzip der Variabilität wären, als

neues Werk interpretieren. Diese zwei Punkte werden besonders später bei der konkreten technischen Konzeption und Umsetzung des Werkstücks von großer Bedeutung sein.

Leider sind dies in der Realität zwar die bedeutendsten Vorteile, müssen aber auch adäquat genutzt werden um ihre Macht entfalten zu können. Viel zu oft wird Software aufgrund schlechter Konzeption und auch aus kommerziellen Interessen trotz ihres modularen und variablen Aufbaus nicht ideal als solche nutzbar gemacht. Um diese zwei Gegebenheiten vollständig ausnützen zu können, muss Software wirklich „soft“, also weich und daher formbar, sein und sich auf ihre einzelnen Bestandteile reduzieren lassen. (Vgl. Maeda, Maeda@Media, 225)

## Automation

Das zuvor übersprungene Prinzip der Automation ist, wie bereits in Kapitel 2.2. behandelt, ein essentielles Prinzip für die generative Methode. Der Computer ermöglicht hoch komplexe Arbeiten, welche ohne die Rechenleistung eines Computers nur durch einen enormen Einsatz von Material, Zeit und Geduld möglich wären. Komplexität ist dabei das Stichwort. Sie ist nicht nur das offensichtlichste Unterscheidungsmerkmal der neuen Medien, sie ist gleichzeitig in einer messbaren Form ein Qualitätsmerkmal generativer Grafik.

## Komplexität

Um die Komplexität zu definieren, bedient sich die Komplexitätstheorie dem Begriff der *Effektiven Komplexität*. Sie stellt den Versuch dar, das was gemeinhin intuitiv als komplex bezeichnet wird, in eine eindeutige Definition zu fassen. Komplex ist dabei in dem Sinne zu verstehen, dass etwas komplex ist, wenn es für den Menschen interessant oder bedeutungsvoll ist. Die Komplexitätstheorie versucht diese intuitiv formulierte, eher schwammige Definition eindeutig festzulegen.

Die Sichtweise der Komplexitätstheorie unterscheidet in Anlehnung an die Informationstheorie zwischen *Algorithmischer Komplexität* und *Effektiver Komplexität*. Die *Algorithmische Komplexität* bezieht sich einzig und allein auf den Informationsgehalt, d.h. je mehr nicht weiter vereinfachbare Information, desto mehr Komplexität.

## Algorithmische Komplexität



Abb. 5: Algorithmische Komplexität (Adaption Philip Galanter)

Dies mag auf den ersten Blick richtig erscheinen, doch anhand eines konkreten Beispiels wird die Unzulänglichkeit der *Algorithmischen Komplexität* zur Beschreibung des menschlichen Empfindens ersichtlich: Bei einem Bild ist der maximale algorithmische Informationsgehalt ein vollkommen zufälliges Rauschen, wie wir es von jedem Fernseher mit gestörtem Empfang kennen. Keiner würde allerdings behaupten, dass dieses Bild die Komplexität eines Gemäldes oder auch nur eines simplen Urlaubsschnappschusses übersteigt. (Vgl. Galanter, *Generative*, 9f)

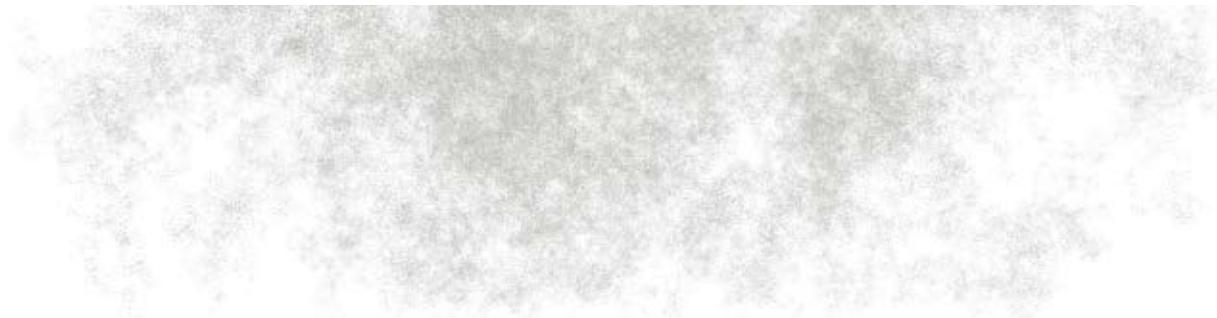


Abb. 6: Zufall, mit dem Werkstück generiert (Random.xml)

John Maeda äußert sich in diesem Zusammenhang folgendermaßen über *Algorithmische Komplexität*.

Generating complexity for complexity's sake is similar to shouting complete nonsense at the top of your voice. Both are embarrassments that are best avoided [...] (Maeda, Maeda@Media, 53)

Hier kommt die *Effektive Komplexität* der Komplexitätstheorie ins Spiel. Sie ist anhand der Regelmäßigkeiten innerhalb eines Systems messbar. Das Maximum an *Effektiver Komplexität* befindet sich genau in der Mitte zwischen perfekter Ordnung (zu viel Regelmäßigkeit d.h. etwas ist langweilig) und absolutem Zufall und Willkürlichkeit (zu wenig bzw. keine Regelmäßigkeit). Die *Effektive Komplexität* deckt sich somit viel besser mit der menschlichen Wahrnehmung als *Algorithmische Komplexität*. (Vgl. Galanter, Generative, 10f)

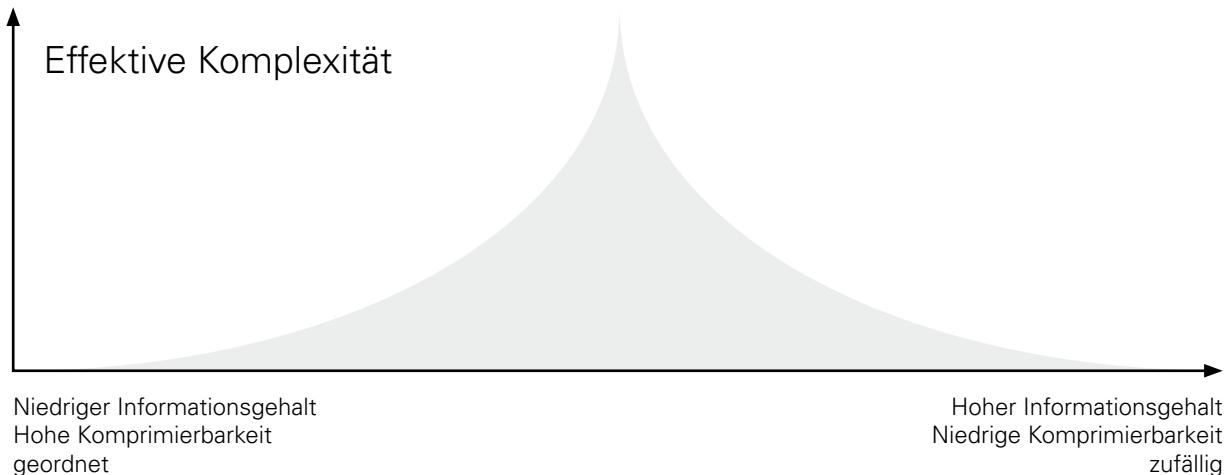


Abb. 7: Effektive Komplexität (Adaption Philip Galanter)

## Transcoding

Kommen wir nun zum letzten Punkt – dem Transcoding. Der bedeutende Einfluss des Computers auf unsere heutige Welt ist nicht zu bestreiten. Golan Levin beschreibt diese fundamentale Bedeutung von Software folgendermaßen:

Meiner Meinung nach zeichnet Software die Vorstellung davon nach, wie die Welt sein sollte. Eine fragile, aber kompromisslose Idee, die nicht nur ein Modell des eigenen Standpunkt enthält, sondern aktiv daran mitwirkt, diesen Standpunkt auf die Umwelt auszudehnen. Wenn das Schreiben ein Medium des Denkens ist, dann ist die Software eine Vermittlerin des Willens. Wenn sie meinen Willen ausführt, bilden die Software und ich eine einzigartige Einheit aus Denken und Zielsetzung. (Vgl. Levin, Werkzeug, 140)

Aus diesem Grund wird es Zeit, auch in den Disziplinen von Kunst und Design mit beiden Armen nach den neuen Möglichkeiten die der Computer eröffnet, zu greifen. Man sollte sich nicht in Form von Nachahmung der Realität an der Vergangenheit festklammern, sondern sich auf die Möglichkeiten konzentrieren, die die neue Dimension des Computers eröffnet. (Vgl. Maeda, Maeda@Media, 128, 153-156)

An dieser Stelle sei auf die Unterschiede zwischen generativer Kunst und Software Kunst hingewiesen. Jene, die mit der generativen Methode arbeiten, wenn auch innerhalb von *New Media*, konzentrieren sich auf die Ästhetik und ästhetische Systeme, während Softwarekunst sich mehr mit dem Computer selbst, und zwar in Form der damit verbunden Kultur, und dem Code der Software selbst beschäftigt. (Vgl. Himmelsbach, Softwarekunst, online und Watz, Generative, online)

## 2.5. Technik und Kreativität

Es sollte mittlerweile klar sein, dass die der generativen Arbeit zu Grunde liegenden Algorithmen in Form von Programmcodes das Herzstück einer generativen Arbeit darstellen. Die Einflussnahme auf diese Algorithmen ist theoretisch ein völlig freier, kreativer Akt. Dennoch handelt es sich bei Computern um Geräte, die nach gewissen technischen und vordefinierten Prinzipien funktionieren, was gewisse Einschränkungen bewirkt und vor allem Wissen um diese Prinzipien und Einschränkungen voraussetzt. Im Laufe dieser Arbeit und besonders im Rahmen von Kapitel 4 wird daher immer wieder die vorherrschende Diskrepanz zwischen der Denk- und Vorgehensweise von Technikern und Kreativen auftauchen. Natürlich handelt es sich dabei um Stereotypen. Dennoch zieht sich diese Kluft auch praktisch durch die ganze Welt der Medienproduktion, wie sich in deren Alltag allorts beobachten lässt.

Mit seiner Tradition als Rechenmaschine ist der Computer von Anfang an ein technisches Gerät für Ingenieure und Techniker gewesen. Daher ist es auch immer schwierig, das kreative Potential dieser „Rechenmaschinen“ den so genannten „Kreativen“ zur Verfügung zu stellen. Auch wenn man meinen könnte, dass sich dies seit den Anfängen der Computer bereits radikal geändert haben sollte, sind wir dabei immer noch nicht so weit, wie man denken würde.

Zwischen

„... artists are ‚illogical, intuitive, and impulsive.‘ They needed programmers who where ‚constrained, logical and precise‘ as translators and interfaces to the computers of the 60’s.“ (Frank, Intelligence, 35f)

und

„Many people think programming is only for people who are good at math and other technical disciplines. One reason programming remains within the domain of this type of personality is that the technically minded people usually create programming languages.“  
(Reas, Processing, 2)

aus dem Jahr 2007 scheint sich nicht viel verändert zu haben.

Oftmals wird der Versuch unternommen, die zwangsläufige Kooperation von Technikern und Kreativen, sowie die Missverständnisse, die sich daraus ergeben, schön zu reden, indem man davon spricht, dass durch ihre Kooperation Ergebnisse entstehen können, mit denen keiner der beiden gerechnet hat. Dennoch scheint es für die Zuverlässigkeit und damit auch den Erfolg nicht gerade vorteilhaft zu sein, wenn man nur auf glückliche, unvorhersehbare Zufälle setzt. (Vgl. Maeda, Maeda@Media, IV)

Gerade die Überbrückung der Kluft zwischen Technik und Kreativität bzw. deren Vereinigung ist eine der schwierigsten Aufgaben für alle digitalen, interaktiven Design- und Kunstdisziplinen. Für generative Arbeiten braucht es schließlich unweigerlich diese noch relativ seltene Hybridform eines programmierenden Designers/Künstlers bzw. kreativen Programmierers. Schon bei dem Versuch dieser Hybridform einen Namen zu geben merkt man die problematische Kluft zwischen diesen beiden Welten. Diejenigen, die den Spagat meistern, haben daher oftmals eigene Wortkreationen wie beispielsweise „Computational Artisan“ (Klingemann, Quasimondo, online) oder „Humanist-technologists“ (Maeda, Maeda@Media, 439) für sich entwickelt. Das zeigt auch, dass sie sich oft weder als Künstler noch als Designer oder Techniker, sondern einfach als kreative Menschen sehen, die gerne mit Computern und Technik arbeiten und experimentieren.

Da sich auf jeden Fall noch kein einheitlicher Begriff durchgesetzt hat, wird in dieser Arbeit immer wieder die Rede von Künstlern, Programmierern, Autoren und dergleichen sein. Wo nicht ausdrücklich auf eine anderweitige Verwendung hingewiesen wird, ist damit aber, um Missverständnissen vorzubeugen, immer der gleiche „Hybrid-Typ“ aus kreativem Techniker und technischem Kreativen gemeint.

## 2.6. Kunst oder Design

Am Rande der Frage „Kreativer oder Techniker“ zeichnet sich bereits die nächste Frage ab – ist generative Gestaltung Kunst oder Design? Wann ist sie was und warum? Diese Frage entspringt der alten Frage nach der Grenzlinie zwischen Kunst und Design.

Paul Rand formulierte es im Gespräch mit John Maeda einmal folgendermaßen:

Sowohl für Design als auch für Kunst gibt es viele Definitionen. Design kann Kunst sein, aber auch Ästhetik oder etwas anderes. Es ist so simpel, dass es wieder kompliziert wird. Im Endeffekt unterscheidet sich ein Künstler nicht von einem Designer, denn beide arbeiten mit Form und Inhalt. (Vgl. Maeda, Maeda@Media, 306)

Diese Definitionen macht vor allem bei der gesonderten Betrachtung von Kunst und Design Sinn. Im Kontext des Einsatzgebiets macht sich dennoch ein Unterschied bemerkbar, welchen Carlos Obers im Rahmen einer Plakatkampagne für die Neue Sammlung München mit folgendem Satz ausgezeichnet auf den Punkt brachte:

Design ist Kunst die sich nützlich macht. (Vgl. Sammlung, Erscheinungsbild, online)

Nun ist klar, dass der Unterschied, ganz abgesehen vom fließenden Übergang der beiden Disziplinen, egal nach welcher Definition, weder in der Ästhetik noch in den Methoden zu suchen ist, sondern sich wenn dann viel mehr erst durch das Ziel und den Einsatzzweck erkennbar macht. Da die generative Methode offensichtlich eine Methode ist und somit per se frei von jeglicher Ästhetik, Kunstrichtung, Ideologie und ähnlichem, ist eine Unterscheidung zwischen Design und Kunst daher meistens, und besonders im Rahmen dieser Arbeit, vollkommen irrelevant. Einzig wenn es um die (kommerzielle) Verwertbarkeit und praktische Anwendbarkeit geht, machen die Begriffe Design bzw. Designer mehr Sinn. Sonst wird es allerdings einfach um einen Autor gehen, womit die Frage Künstler/Designer in

dieser Arbeit nicht weiter behandelt werden wird und Begriffe wie Künstler, Designer oder Ersteller daher als gleichwertige Synonyme verwendet werden. Einzige Ausnahme ist Kapitel 5 wenn es um mögliche Anwendungsgebiete geht. Dort ist die Unterscheidung zwischen Arbeiten welche nur ästhetischen Kriterien entsprechen müssen und als Kunst fungieren und deren angewandtem Gegenstück, wie beispielsweise bei der Datenvisualisierung, angebracht.

### 3. Generative Computergrafik

In Kapitel 2 sind die Eigenschaften der generativen Methode im Allgemeinen und im Rahmen von *New Media* behandelt worden. In diesem Kapitel wird nun der Schaffensprozess einer generativen Computergrafik von der Methodik, über die Ästhetik bis zur geeigneten Technologie behandelt.

Im Kontext der generativen Grafik interpretiert und um für eben jene spezifische zusätzliche Merkmale erweitert, sind folgende Punkte, beginnend mit den übernommenen allgemeinen Eigenschaften der generativen Methode im Rahmen von *New Media*, für generative Computergrafik ausschlaggebend:

- Automation anhand von Regeln
- Regeln in Form von Algorithmen
- veränderbare, erweiterbare (Variabilität) und kombinierbare (Modularität) Algorithmen in Form von Programmcodes
- selbst laufendes, autonomes System in Form von Programmcodes

Folgende Eigenschaft kommt für die generative Computergrafik im Allgemeinen hinzu:

- eine wie auch immer geartete graphische, visuelle, in irgend einer Art und Weise ästhetische Ausgabe

Zusammenfassend lässt sich also folgende Definition festhalten:

Eine generative Computergrafik ist jedes in gewisser Weise ästhetisches graphisch-visuelles Werk, das durch die automatische Ausführung eines Programms entsteht, welches die Implementation einer beliebigen Anzahl von in Form von Algorithmen definierten Regeln und Anweisungen oder deren Kombinationen und Variationen darstellt.

### 3.1. Gestalterische Arbeitsweise und ästhetische Kriterien

Mit diesem Kapitel verlassen wir nun erstmals vollkommen das allgemeine, methodische Terrain der generativen Methode bzw. der generativen Grafik. Hier soll es nun darum gehen, verschiedene gestalterische Arbeitsweisen und ästhetische Kriterien für diese konkrete Arbeit und das daraus resultierende Werkstück zu erarbeiten und festzulegen. Es geht also nicht mehr um die Methode der Erstellung sondern um das gestalterische Ziel in diesem konkreten Fall.

*New Media* und damit auch die generative Computergrafik sind natürlich ein neues Medium. Einer der ältesten und zugleich fatalsten Irrtümer, denen man in diesem Bereich allerdings erliegen kann, ist zu glauben man erreiche den Bereich von *New Media* schon allein durch die Digitalisierung analoger Medien, also die Ansicht digitale Medien seien nicht mehr als konvertierte, in Einsen und Nullen, in mathematische, informatische Darstellungen umgewandelte analoge Medien. (Vgl. Manovich, *Media*, 49)

Software und damit auch generative Grafik ist ein einzigartiges Medium mit einzigartigen Qualitäten und bedarf daher auch einzigartiger Konzepte. Sie sollte in keinem Fall in eine falsche Relation zu älteren Medien wie Film, Photographie oder Malerei gestellt werden. Wie die Geschichte gezeigt hat, haben neue Medien immer auch neue Arten der Kommunikation und des Ausdrucks ermöglicht und forciert. Um dieses Potential voll auszunutzen ist es wichtig, sich auf die zuvor genannten spezifischen Eigenschaften des Mediums zu konzentrieren. (Vgl. Reas, *Processing*, 1)

Mit dieser vorangestellten Warnung ist das Folgende weniger missverständlich. Denn selbstverständlich existiert kein Medium in vollkommener Autonomie, quasi wie in einer von einem Vakuum umgebenen Blase. So kann sich auch ein Autor einer generativen Grafik natürlich nicht dem kunst- und kulturhistorischen Kontext, den Eigenschaften und Eigenarten der Natur oder den fundamentalen naturwissenschaftlichen Grundsätzen der sichtbaren und unsichtbaren Welt entziehen. Aus diesem

Grund soll in den folgenden Kapiteln ein möglichst breit gefächertes Überblick über die verschiedenen Ansätze für generative Grafik verschafft werden.

Da dadurch offensichtlich der Versuch unternommen wird, ein universales Inspirations-Kompendium zu kreieren, muss von Anfang an klar sein, dass diese Aufstellung in keiner Weise Anspruch auf Vollständigkeit erhebt. Sie ist viel mehr als Sammlung aller im Rahmen der Recherche für diese Arbeit aufgetauchten Herangehensweisen zu verstehen und soll einerseits als Ergänzung zu den persönlichen Arbeitsweisen der später in Kapitel 4 vorgestellten Personen dienen, andererseits aber auch den Grundstein für die Überlegungen zum in Kapitel 6 näher erläuterten Werkstück legen.

### 3.1.1. Natur

Die Anziehungskraft der Natur war immer schon einer der stärksten Einflüsse auf die Menschen und damit auch auf ihre Kunst und Kultur, nichts ist schließlich so natürlich wie die Natur selbst. Maler, Bildhauer, Architekten und viele andere Kunstschaffende haben immer schon Inspirationen und Denkanstöße aus der Natur bezogen. Dennoch ist mit der generativen Grafik eine vollkommen neue Sichtweise auf die Natur entstanden. War die bisher vorherrschende Wahrnehmung der Natur meist eine ästhetische, ermöglicht die prozedurale Beschaffenheit der generativen Grafik die Behandlung der Abläufe aus der Natur in Form von Algorithmen.

Bei der Beschäftigung mit der Natur darf man niemals vergessen, dass es keinen Sinn macht die Natur einfach so, wie sie ist, zu simulieren. Ganz abgesehen von den Schwierigkeiten dieses Vorhabens<sup>4</sup> ist eine sinnvolle Beschäftigung mit der Natur jene, die die Eigenständigkeit der neuen Medien respektiert und ernst nimmt. Sie sollte sich nicht an der „Digitalisierung der Natur“ versuchen, sondern die zu Grunde liegenden Strukturen und Konzepte von Komplexität, Mehrdeutigkeit und Vielschichtigkeit

---

4 Zur Problematik der Simulation der Natur und den Grenzen ihrer Sinnhaftigkeit empfiehlt sich die Kurzgeschichte Borges, Jorge Luis: *On Exactitude in Science*, Harmondsworth: Viking 1998

analysieren und im Kontext der neuen Medien in einer eigenen Form interpretieren. (Vgl. Co, Pixeln, 172) Diese aufmerksame Wahrnehmung der Natur kann ungeahnte Formen der Inspiration zum Vorschein bringen.

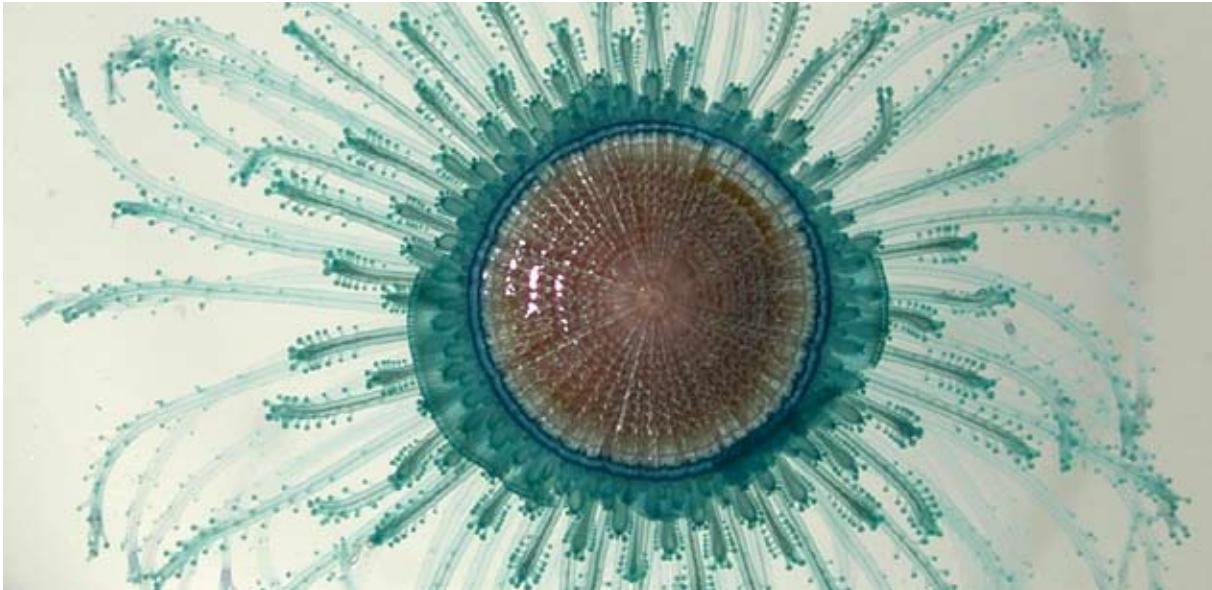


Abb. 8: Porpita porpita

### 3.1.2. Mathematik und Naturgesetze

Wenn man sich eingehender mit der Natur beschäftigt, stößt man dank der Erkenntnisse der modernen Wissenschaft bald auf die untere Ebene der zu Grunde liegenden Strukturen – die sogenannten Naturgesetze. Diese wiederum sind nichts Anderes als mathematische Systeme. Besonders der Computer eignet sich durch seine Eigenschaft als zutiefst mathematisches Medium natürlich ausgezeichnet, um diese Sphäre zu untersuchen.

Bei der Arbeit mit dem Computer besteht allerdings immer die Gefahr, dass sich die Mathematik still und leise einnistet und zum alles dominierendem Element wird. Wenn es dann schließlich soweit kommt, dass mathematische Systeme zur Kunst ernannt werden, stellt sich die Frage, wo Mathematik endet und wo Kunst beginnt. Ein gutes Beispiel für eine solche Grenzüberschreitung stellen die immer wieder auftauchenden eingefärbten Mandelbrot-Fraktale dar. (Vgl. Maeda, Creative, 101)

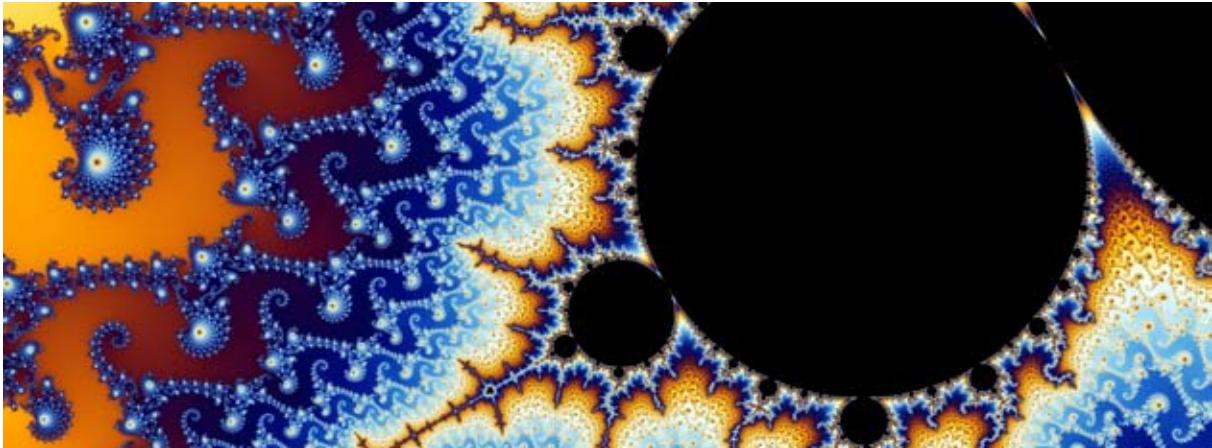


Abb. 9: Fraktal Mandelbrot-Menge

### 3.1.3. Kunst, Malerei

Doch nicht nur die Mathematik im Kontext von Physik, Chemie und anderen Naturwissenschaften befasst sich mit den der Welt zu Grunde liegenden Prinzipien.

Auch in der Kunst, welche natürlich auch immer in einem direkten Wechselspiel mit Wissenschaft und Technik steht, sind viele Strömungen entstanden, die sich aus der rein ästhetischen Repräsentation der Welt heraus entwickelt haben und einen dem Computerzeitalter näher stehenden konzeptuellen, prozeduralen Charakter angenommen haben.

Besonders in den 1960er Jahren tat sich analog zur Serientauglichkeit und der ersten Entwicklung von „Personal Computern“ in der analogen Kunst viel in dieser Richtung. Egal ob man den Einfluss der Kunst auf die Technik oder andersherum der Technik auf die Kunst sieht, oder ob man das alles einfach als gesamtgesellschaftliche Entwicklung betrachtet – es ist mit Sicherheit kein Zufall, dass gerade in den 60er Jahren Kunstrichtungen wie Op-Art, Kinetische Kunst, Prozesskunst, Aktionskunst, Konzeptkunst usw. entstanden sind. (Vgl. Weibel, Revolution, online)

Als Beispiele für deren Bedeutung im Zusammenhang mit generativer Grafik seien hier nun einige Kunstrichtungen exemplarisch genannt.

Die Op-Art (von engl. Optical Art) befasst sich mit den Gesetzmäßigkeiten der visuellen Wahrnehmung und versucht so die dieser zu Grunde liegenden Algorithmen zu erforschen. Dabei entstehen auf geometrischen Formen basierende abstrakte Muster, welche durch Interaktion zwischen dem bewegten Betrachter und dem statischen Muster Flimmer- und Vibrationseffekte hervorrufen. Zu den bedeutendsten Vertretern gehören unter anderem Victor Vasarely und Bridget Riley.



Abb. 10: Fassade des Bonner Juridicums von Victor Vasarely

Die Kinetische Kunst geht einen Schritt weiter. Sie führte die Bewegung, die Dynamik genauso wie die Op-Art als künstlerisches Element ein. Kunstwerke sind aber, im Gegensatz zur Op-Art, nicht länger statische Objekte, sondern werden von verschiedenen Kräften wie Wind oder Motoren angetrieben. Dadurch führen sie die vierte Dimension, die Zeit, in ihre Kunstwerke ein. Natürlich stellt diese Dynamik im Gegensatz zur Dynamik im Sinne von Software nur einen sehr eingeschränkten Ansatz dar, der für jede neue Form von Dynamik die neuerliche Konstruktion einer eigenen Maschine benötigt. Dennoch ist sie eine wichtige Entwicklung, die die Verbreitung der generativen Methode im Kunstkontext mit Sicherheit gefördert hat. (Vgl. Szope, Kinetische, online)



Abb. 11: Eos von Jean Tinguely

Die Konzeptkunst schließlich wandte sich von der Bedeutung des real existierenden Kunstwerkes ab und konzentrierte sich dafür auf das Konzept und die zu Grunde liegende Idee als das eigentliche Kunstwerk. Als bedeutender Vertreter schuf Sol LeWitt dabei Kunstwerke in Form von Handlungsanweisungen. Diese Handlungsanweisungen sind im Endeffekt nichts anderes als Algorithmen. (Galanter, Generative, 13 und 18)

## 3.2. Technische Grundlagen und relevante Technologien

### 3.2.1. Existierende Tools und ihre Einschränkungen

Die Prinzipien der Modularität und Variabilität sind für *New Media* von essentieller Bedeutung. Sie bergen aber auch eine Gefahr, denn sie verlocken mit ihren vorgefertigten Elementen wie Plugins und Komponenten dazu in einem Remix des Vorhandenen zu verharren und damit den Fortschritt in Form von fundamentalen Neuerungen zu blockieren. Die Mehrheit der heutigen Computernutzer sind Anwender. Auch unter den so genannten „Kreativen“ ist der wirklich kreative Umgang mit Tools und die Anpassung an persönliche Bedürfnisse in einem stärkeren Maß als durch Presets und Skins nur wenig verbreitet. Doch vorgefertigte Komponenten und Tools schränken den Spielraum ein. Es wird Zeit, dass wir nicht nur die wenigen häufiger verwendeten Tools wie Photoshop und Illustrator verwenden, die uns die großen Softwarefirmen zur Verfügung stellen, sondern unsere eigenen schaffen.

Dabei muss man natürlich zugeben, dass der Aufwand zur Erstellung eines eigenen Programms oftmals immens ist. Von Grund auf selbst gemachte Tools werden außerdem nach ihrer Fertigstellung häufig vom Ersteller nicht einmal mehr verwendet, da durch die lange Beschäftigung mit einem Programmiermuster der Reiz daran bereits verbraucht ist. (Vgl. Maeda, Maeda@Media, 401-404)

Das folgende Kapitel soll daher die Anforderungen für Software festlegen, welche geeignet ist, zur Erstellung generativer Grafik herangezogen zu werden.

## 3.2.2. Technische Anforderungen

### Arbeits-, Speicher- und Exportformate

Bevor man noch zu den konkreten Werkzeugen und Möglichkeiten einer Technologie kommt, ist es wichtig, sich die fundamentale Unterstützung der grundlegenden Formate anzusehen. Unter grundlegenden Formaten sind hier die Möglichkeiten sowohl vektorbasiert als auch rasterbasiert arbeiten zu können, zu verstehen. Weiter ist auch die native Unterstützung von 3D zu beachten, wobei hier wichtig ist, dass sie nicht als Grundelement implementiert ist, sodass sie im 2D-Einsatz die Sache nicht unnötig verkompliziert. Ebenfalls wichtig ist, dass das Zusammenspiel von Vektorgrafik und Pixelgrafik sowie die Rasterung von Vektordaten möglich ist.

Auch der Arbeitsablauf ist von Bedeutung. Kann man seine generativen Systeme speichern? Kann man nur das System an sich oder auch einen konkreten Ablauf speichern? Ist das Speicherformat proprietär oder z.B. als XML-Format von anderen Technologien verarbeitbar?

Es ist allerdings nicht nur wichtig, auf welche Weise das Programm intern arbeitet, sondern auch, welche Exportmöglichkeiten es anbietet. Dabei ist sowohl die Anzahl und die Art der unterstützten Formate, als auch die erreichte Qualität von Bedeutung.

### Verfügbare Werkzeuge und Grundfunktionen

Ein weiteres entscheidendes Merkmal sind die zur Verfügung stehenden Grundfunktionen einer Technologie. Hier muss nicht unbedingt „Mehr ist besser“ gelten. Sofern alle für die Erstellung generativer Grafiken notwendigen Grundfunktionen verfügbar sind, kann zu Gunsten der Übersichtlichkeit ein eingeschränkter Funktionsumfang durchaus vorteilhaft sein.

## Anpassbarkeit und Erweiterbarkeit

Es ist aber nicht nur wichtig welche Funktionalitäten zu Verfügung stehen, sondern auch, inwieweit sie anpassbar und durch neue, eigene Werkzeuge erweiterbar sind.

Das kann sowohl in Form von Plugins oder Skripting-Möglichkeiten, Kommunikation mit externen Modulen oder anderen Anwendungen als auch in Form von direkten Zugriff auf den Quellcode der Technologie geschehen.

## Lizenzbedingung und Verlässlichkeit des Systems

In diesem Zusammenhang stellt sich auch die Frage nach der Lizenz der Technologie. Freeware und besonders Open Source Technologie ist prinzipiell zu bevorzugen. Weiter ist die Größe der Community oder der Firma, die hinter einem System steht, für die langfristige Verlässlichkeit hinsichtlich Support und Weiterentwicklung zu beachten.

### 3.2.3. Relevante Technologien

Nun werden einige verfügbare Technologien näher betrachtet, um zu einer Entscheidung hinsichtlich der im Werkstück gewählten Technologie zu kommen.

#### 3.2.3.1. Processing und Java

Processing ist eine auf Java basierende Open Source Technologie. Die offizielle Beschreibung der entsprechenden Webseite sieht folgendermaßen aus:

Processing is an open source programming language and environment for people who want to program images, animation, and interactions. It is used by students, artists, designers,

researchers, and hobbyists for learning, prototyping, and production. It is created to teach fundamentals of computer programming within a visual context and to serve as a software sketchbook and professional production tool. Processing is an alternative to proprietary software tools in the same domain. (Processing, online)

Processing scheint derzeit unter allen verfügbaren Tools für generative Grafik neben Flash das Populärste zu sein. Dafür gibt es einige Gründe:

Ein erster ist mit Sicherheit die Tatsache, dass mit Ben Fry und Casey Reas<sup>5</sup> als Initiatoren des Projekts zwei Koryphäen der Szene als Aushängeschild fungieren. Weiter ist die Programmiersprache durch die ausdrückliche Auslegung ihrer Syntax auf visuell denkende Menschen ideal für das Einsatzgebiet in und rund um die generative Grafik geeignet.

Dabei unterstützt Processing sowohl Vektorgrafik, Rastergrafik als auch den 3D-Raum in Form des softwaregestützten P3D-Renderers und des hardwaregestützten OpenGL Renderers. Als ausdrücklich auf die Erstellung visueller Programme ausgerichtete Technologie bietet Processing sowohl alle grundlegenden Zeichnungsfunktionen im 2D-Raum als auch erweiterte 3D-Funktionen wie Kameras, Licht und Texturen. Processing bietet allerdings keine Möglichkeiten abseits vom Code über ein Graphical User Interface (GUI) Elemente zu erstellen, zu ordnen oder zu bearbeiten.

Das Open Source Prinzip und somit die freie Verfügbarkeit erleichtert den Einstieg und das unverbindliche Experimentieren. Durch die Tatsache, dass Processing auf Java basiert, ist es außerdem in Windows, Mac OS X und Linux einsetzbar, bei Bedarf mittels Java erweiterbar und vor allem äußerst leistungsfähig, besonders im Vergleich zu Scriptsprachen.

---

5 Ben Fry und Casey Reas waren Schüler von John Maeda in der Aesthetics and Computation Group am MIT Media Lab und haben bereits einige Standardwerke zur kreativen Programmierung und Datenvisualisierung veröffentlicht. Darunter auch Reas, Casey: Processing : a programming handbook for visual designers and artists, Cambridge: MIT Press 2007 und Fry, Ben: Visualizing Data, Sebastopol: O'Reilly Media 2008

Der einzige wirklich große Nachteil liegt darin, dass es sich aufgrund der im Vergleich zu Flash mäßigen Verbreitung des Java-Browser-Plugins eher nicht für reine Online Projekte für die Breite Masse eignet. Processing ist also mehr im Kontext von Kunst, Ausstellungen und Prototyping angesiedelt und daher ist der Übergang zu einer kommerziellen, massentauglichen Verwendung momentan problematisch.

### 3.2.3.2. Flash, Flex, AIR

Flash ist ein ursprünglich als reines Vektor-Animationsprogramm entwickeltes System, welches sich aber in den letzten zehn Jahren durch die Einführung immer leistungsfähigerer Scriptingmöglichkeiten zu einer vollwertigen Rich Internet Application (RIA)<sup>6</sup>-Plattform entwickelt hat, ohne dabei die ursprünglichen Möglichkeiten als Animationstool auf der Strecke zu lassen.

Durch seine lange Geschichte als Vektoranimationsprogramm verfügt Flash über ausgezeichnete Funktionen zur Erstellung und Bearbeitung von Vektorgrafiken, sowohl über das GUI des Authoringprogrammes als auch per Code. Seit einigen Versionen hat auch die Rastergrafik Einzug gehalten und ermöglicht die Grafikbearbeitung auf Pixelebene per Code. In der zur Zeit der Erstellung dieser Arbeit noch in der öffentlichen Beta-Phase<sup>7</sup> befindlichen Version 10 des Flash-Players hat schließlich eine rudimentäre native Unterstützung für 3D Einzug gehalten.

Weiter wird eine neue Sprache für die Erstellung von Grafikfiltern namens Pixel Bender eingeführt. Pixel Bender soll eine in verschiedenen Adobe Produkten eingesetzte Sprache zur Grafikprogrammierung werden, welche (wenn auch in reduzierter Form) in Flash die gleichen Effekte wie in Adobe

---

6 RIAs sind Anwendungen, welche im Browser ablaufen, aber im Gegensatz zu einfachen Webseiten besonders leistungsfähig sind und komplexe Aufgaben vergleichbar mit lokal installierten Programmen erledigen können.

7 Letzte Stufe vor der Veröffentlichung einer Version. Zuerst stehen interne Rohversionen, genannt Alpha-Versionen. Dann folgen interne Beta-Versionen und schließlich der Entwicklergemeinde zugängliche öffentliche Beta-Versionen welche die letzte Stufe vor der Veröffentlichung für den Endnutzer darstellen.

After Effects und anderen Adobe Produkten ermöglichen soll. Pixel Bender soll optional auch erstmals in der Geschichte von Flash hardwareseitige Unterstützung durch die Grafikkarte ermöglichen. (Vgl. Pixel Bender Toolkit, online)

Vor allem durch die volle Integration in die Produktpalette von Adobe bietet die Flash-Authoring-umgebung eine große Menge an Import- und Exportmöglichkeiten für Vektor-, Pixel- und Videodaten.

Adobe bietet neben dem Flash Authoring Tool und dem zugehörigen Flash-Player auch noch zwei bedeutende aufbauende Technologien an.

Flex ist ein auf Flash aufbauendes Framework, welches sich besonders für die schnelle Entwicklung von RIAs durch die effektive Zusammenarbeit von Designern und Programmierern eignet. Wenn auch für die generative Grafik nicht unbedingt das ideale Werkzeug, ermöglicht es zumindest die unproblematische Entwicklung eines Interfaces, wie man z.B. bei Mario Klingemanns „The Blind Sketchmaker“<sup>8</sup> ausgezeichnet sehen kann.

Dank Flex gibt es mittlerweile auch einen Open Source Compiler, womit für den Einsatz von Flash nicht mehr zwangsläufig der Kauf der Authoringumgebung notwendig ist.

Die zweite auf Flash aufbauende Technologie von Adobe ist die Adobe Integrated Runtime, besser bekannt als AIR. AIR ermöglicht es, unter Einsatz von Web Technologien wie Flash, aber auch HTML, CSS und Java Script lokal laufende Applikationen zu erstellen. Das hat viele Vorteile. Einerseits muss man eine Applikation nicht extra als Webapplikation und als lokal laufende Applikation entwickeln und zweitens ermöglicht es viele erweiterte Funktionalitäten beim Einsatz von Flash. Um nur eine, für die Erstellung generativer Grafik besonders nützliche Funktion zu nennen, ist mit AIR durch Funktionen wie Drag and Drop eine reibungslose Integration in das Betriebssystem möglich.

Der weitaus größte Vorteil von Flash liegt aber im Rahmen von Onlineprojekten – die Verbreitung

8 Klingemann, Mario: The Blind Sketchmaker. Exploring Generative and Evolutionary Art with Flash, [http://lectures.quasimondo.com/the\\_blind\\_sketchmaker/index.php](http://lectures.quasimondo.com/the_blind_sketchmaker/index.php)

des Flashplayers. Das Flash Plugin scheint, wenn man den offiziellen Herstellerstatistiken glauben schenkt, das am weitesten verbreitete Browser-Plugin überhaupt zu sein und, wenn auch nicht immer in der aktuellsten Version, auf nahezu jedem Rechner, der im Internet unterwegs ist, installiert zu sein.

Eine Menge an auf diese Technologie aufbauenden Systemen machen Flash weiter attraktiv. Hier sei nur eine kleine exemplarische Auswahl in Form von zwei noch in der Entwicklung befindlichen Projekten erwähnt.

Das erste Projekt heißt Bloom und versucht, ohne eine direkte Portierung sein zu wollen, die Vorteile und große Teile der Syntax von Processing auf die Flash Welt zu übertragen. Gelingt dies, könnte man alle Vorteile von Processing plus alle Vorteile von Flash wie z.B. die erhöhte Plugin-Verbreitung nutzen. (Vgl. Hope, Bloom, online)

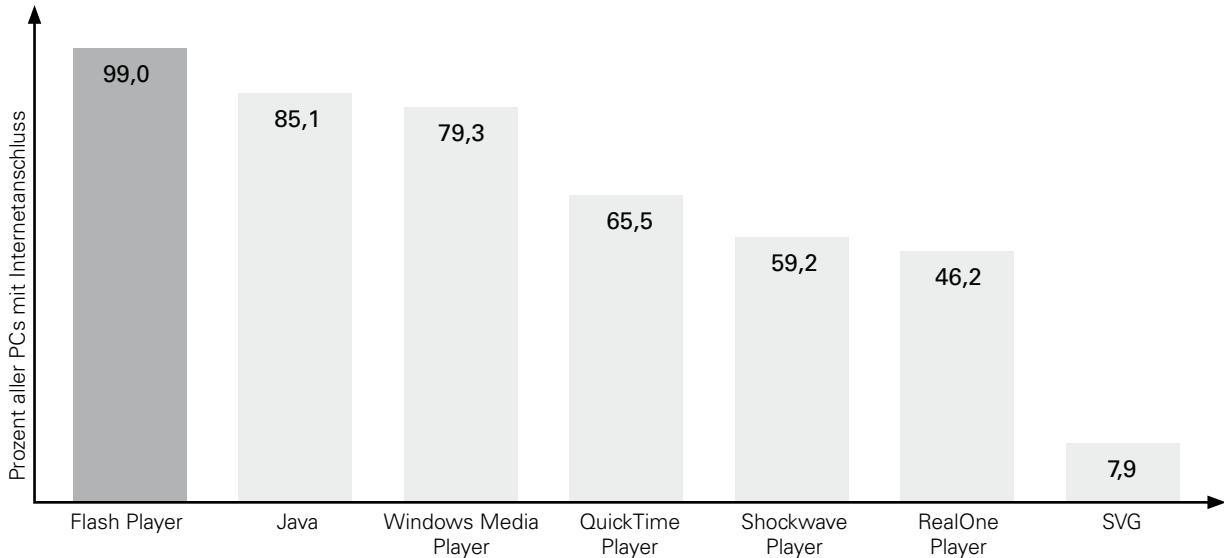


Abb. 12: Verbreitung des Flash Players laut Adobe

SourceBinder ist ein weiteres auf Flash basierendes Projekt, welches den Versuch wagt das Konzept der graphischen Programmierumgebung, wie man es z.B. von vvvv kennt (siehe unten), auf die Flash-Plattform zu übertragen. Mittels einfachen Blocks von Programmcode lassen sich Programme erstellen, ohne direkt im Quellcode zu arbeiten. Der Vorteil von SourceBinder gegenüber reinen graphischen Programmierumgebungen ist, dass man dennoch auch direkt am Quellcode arbeiten kann und auch ursprünglich nicht für SourceBinder entwickelte Klassen mittels Wrappern einbinden kann. (Vgl. Visual, SourceBinder, online)

### 3.2.3.3. vvvv

vvvv ist eine graphische Programmierumgebung, welche primär auf Installationen, 3D, Video und Audio Processing ausgelegt ist. Die Beschreibung der vvvv Webseite nennt folgende Fakten:

vvvv is a toolkit for real time video synthesis. It is designed to facilitate the handling of large media environments with physical interfaces, real-time motion graphics, audio and video that can interact with many users simultaneously.

vvvv uses a visual programming interface. Therefore it provides a graphical programming language for easy prototyping and development. (vvvv, multipurpose, online)

vvvv ist im Gegensatz zu Flash und besonders Processing nicht gänzlich frei verfügbar und erfordert für den kommerziellen Einsatz Lizenzzahlungen. Für den privaten und nicht kommerziellen Einsatz ist es jedoch kostenlos. Leider läuft vvvv derzeit nur auf Windows.

Interessant ist der Ansatz, dass es keine Unterscheidung zwischen Erstellungs- und Runmodus gibt. Alle Änderungen am Programm werden sofort im laufenden Programm umgesetzt.

vvvv bietet eine große Bandbreite an Funktionen, welche aber offensichtlich sehr stark auf 3D, Video, Animation und Kommunikation mit externen Geräten ausgelegt sind. Für den Einsatz in einem Live-

Szenario kann vvvv neben den üblichen Protokollen wie HTTP auch außergewöhnlich viele externe Schnittstellen wie MIDI oder DMX nativ unterstützen.

Weiter zeigt sich die Ausrichtung von vvvv auf den Einsatz in Ausstellungs- oder Installationsszenarien durch eine „Boygrouping“ genannte Funktionalität. Sie ermöglicht es, das Programm bzw. die Rechenlast auf mehrere Computer zu verteilen, und das Programm ist somit besonders für hochperformante Anwendungen wie Echtzeit-Videoanalyse oder Physiksimulationen geeignet. Dies zeigt sich auch darin, dass beispielsweise für Physiksimulationen bereits entsprechende Funktionalitäten vorhanden sind. Möchte man sich allerdings nur im 2D-Bereich bewegen, ist vvvv etwas überproportioniert. Durch die Auslegung auf hochperformante Live-Szenarios ist es nicht überraschend, dass vvvv keine Möglichkeit der Veröffentlichung für das Internet bietet. Außerdem ist vvvv so konzipiert, dass damit erstellte Programme nicht als eigenständige Programme lauffähig sind, sondern immer auf die vvvv Runtime angewiesen sind. Die Verwendung durch Enduser abseits des Entwicklers ist dadurch sehr problematisch.

Wenn auch der Ansatz der graphischen Programmierung interessant erscheint, ist vvvv generell für den Einsatz als generatives Instrument im Sinne dieser Arbeit aus den zuvor genannten Gründen nicht besonders gut geeignet.<sup>9</sup>

#### 3.2.3.4. Hybride

In der Praxis kann es durchaus sinnvoll sein, einen hybriden Einsatz von verschiedenen Technologien anzudenken. Exemplarisch sei hier die Kommunikation von Flash und Processing erwähnt, zu der Mario Klingemann ein Proof of Concept zur Verfügung stellt. (Vgl. Klingemann, Communication, online)

---

9 Für alle jene, die mit einer anderen Herangehensweise einen Einblick in die Verwendung von vvvv bekommen möchten, sei die Diplomarbeit von Thomas Hitthaler erwähnt: Hitthaler, Thomas: Generative Erzeugung von Design mit vvvv, Diplomarbeit, FH Salzburg 2005

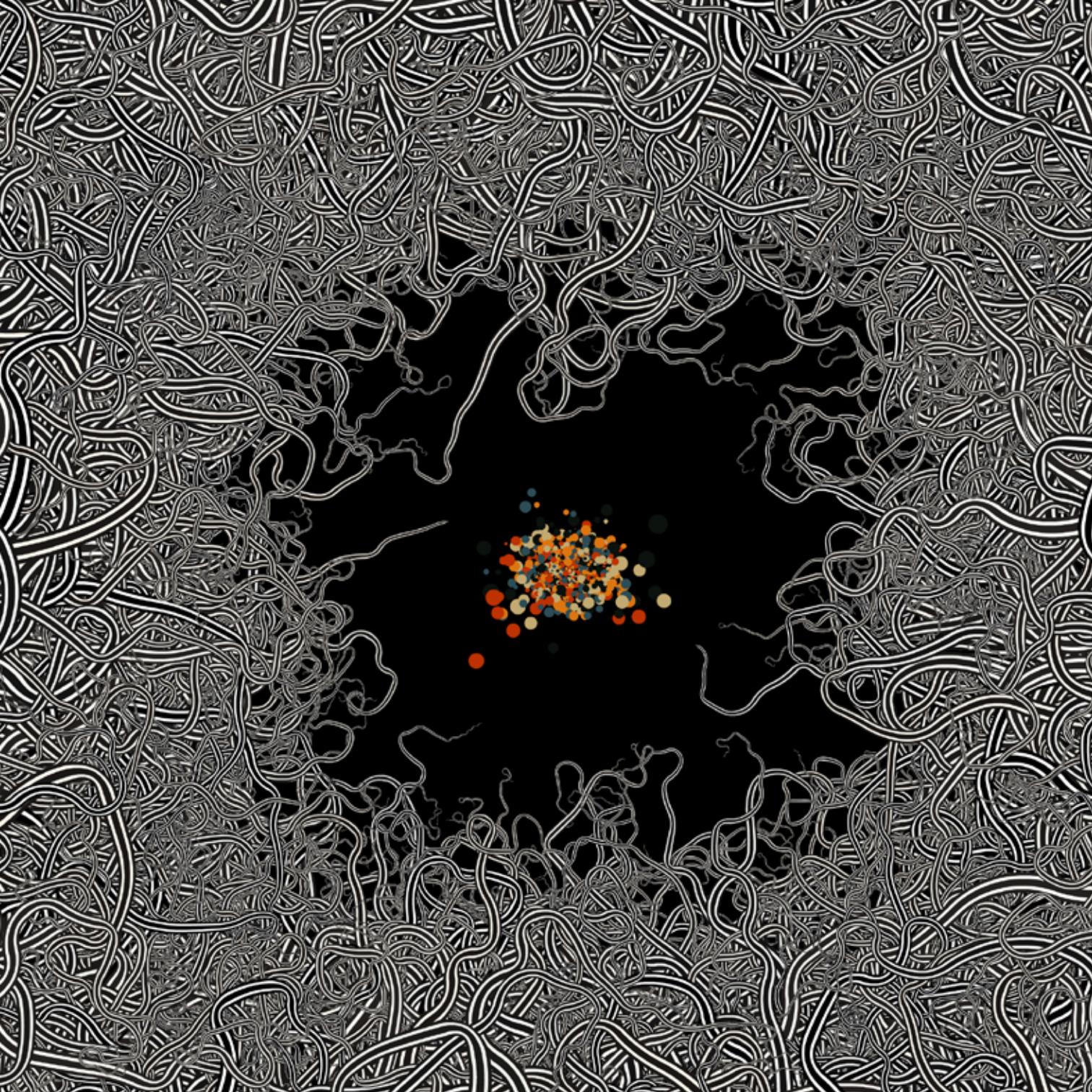
Es sind aber auch andere Kombinationen möglich, solange eine einfache und verlässliche Schnittstelle zwischen den Systemen herstellbar ist.

### 3.2.4. Auswahl der Technologie für das Werkstück

Jede der zuvor genannten Technologien hat mit Sicherheit ihre Vorteile und natürlich auch sinnvolle Einsatzszenarios. Die getroffene Entscheidung ist daher nicht als allgemeine Wertung zu verstehen, sondern als Eignung für diesen konkreten Fall.

www disqualifiziert sich im Rahmen dieser Arbeit durch den unnötig komplexen Funktionsumfang und den Mangel an Exportmöglichkeit für Endnutzer abseits der Entwicklergemeinde.

Processing hat schon weitaus bessere Karten. Es bietet die Möglichkeit Projekte als eigenständige Programme zu exportieren und als Java-Applets im Internet zu veröffentlichen. Dennoch bietet Flash eine unübertroffene Reichweite. Auch die angenehme Integration in den Layoutprozess unter Verwendung von anderen Adobe-Programmen die zur Erstellung dieser Arbeit zum Einsatz kommen ist nicht zu vernachlässigen. Nicht zuletzt meine jahrelange Erfahrung mit Flash hat daher schlussendlich zu der Entscheidung diese Technologie für das Werkstück einzusetzen geführt.



## 4. Die Szene – exemplarische Vertreter der generativen Arbeitsweise

Die generative Computergrafik ist fast genauso alt wie der Computer selbst. Als erster Vertreter wird gemeinhin A. Michael Noll gesehen welcher Mitte der 60er Jahre erste computergenerierte Grafiken anfertigte. (Vgl. Maeda, Maeda@Media, 21)

Sie blieb aber lange Zeit ein Nischenbereich, welcher sich einerseits Mangels Ressourcen, andererseits mangels Anerkennung in der Gesellschaft und besonders in der Kunstwelt nicht recht entfalten konnte. Ganz abgesehen von der Problematik, in der vom Preisschema des Originals bestimmten Kunstwelt nicht finanziell Fuß fassen zu können, stellt sich noch ein weiteres Problem. (Vgl. Wands, Digital, 12) Dynamik und Veränderung lässt sich schwer als Element eines gestalterischen Statements akzeptieren, da wir aus der Tradition von Kunst heraus an Materialien und damit auch Medien mit bestimmten fixierten Eigenschaften gewöhnt sind. (Vgl. Maeda, Maeda@Media, 64)

Das zweite Probleme war der Ressourcenmangel, welcher sich aber neben der offensichtlichen Bedeutung der damals schlichtweg nicht vorhandenen Verfügbarkeit von Computern auch noch anders auswirkte. Es mangelte an Publikum, an Distributionswegen und somit auch an einer interessierten Zielgruppe bzw. der Erreichbarkeit dieser. Erst seit sich das Internet zum Massenmedium entwickelt hat, steht dieses Publikum in fast unbegrenztem Ausmaß zur Verfügung. (Vgl. Maeda, Maeda@Media, 241) Die wahre Blütezeit der generativen Grafik ging also mit dem Aufstieg des Internets einher. Deswegen sollen in Folge auch jene Personen der letzten Jahre vorgestellt werden, welche in dieser Zeit und bis heute den Bereich des generativen Designs geprägt haben.

An dieser Stelle, ganz besonders an dieser Stelle, muss noch einmal darauf hingewiesen werden,

◀ Abb. 13: Central Darkness, mit dem Werkstück generiert (CentralDarkness.xml)

dass es sich bei der generativen Methode um eine Methode handelt, die keine konkrete Ästhetik oder Ausdrucksform impliziert. Das ist in diesem Kapitel besonders deswegen von so großer Bedeutung, weil wir es hier mit einer Menge absolut unterschiedlicher Personen und Arbeitsgebiete zu tun haben. Manche würden sich wahrscheinlich selbst nicht einmal in diese Liste einordnen. Dennoch sind sie alle auf ihre ganz individuelle Art von großer Bedeutung für die Entwicklungen und Trends der letzten Jahre und haben in gewisser Weise doch wieder ihre Gemeinsamkeiten oder gehören einer ähnlichen Denkschule an. (Vgl. Watz, Generative, online)

## 4.1. Yugo Nakamura

Es war schwer zu entscheiden, ob Yugo Nakamura am Anfang der Liste oder am Ende nahe dem Kapitel über Anwendungsgebiete in Visualisierung und Interface Design stehen soll. Einerseits ist er wohl der Pionier, was kreativ-experimentellen Umgang mit Flash betrifft, andererseits verkörpert er eine überraschend frische Mischung aus gewagten, innovativen Experimenten und einem fast schon konservativem Interesse an menschlichen, natürlichen Interface Design, wenn auch in einer nicht an Standards festhaltenden, sondern zukunftsorientierten Art und Weise, die dem „Usability Guru“ Jacob Nielsen<sup>10</sup> wahrscheinlich die Zornesröte ins Gesicht treiben würde. Die Entscheidung, Yugop (unter welchem Synonym Yugo Nakamura aufgrund seiner Webseite <http://yugop.com> berühmt wurde) dann doch an den Anfang zu stellen, liegt vor allem daran, dass er das perfekte Bindeglied zwischen den beiden Extremen autonome Kunst und kommerzielles Mainstream-Design darstellt. Und natürlich ist seine fundamentale Bedeutung als Urgestein im Flashbereich nicht zu leugnen.

Seine ursprüngliche Ausbildung war (zumindest auf den ersten Blick) weit entfernt von den Dingen, die er jetzt erschafft und womit er seine heute internationale Berühmtheit erreicht hat. Er studierte

---

<sup>10</sup> Jacob Nielsen gehört zu den international anerkanntesten Usability-Experten und gehört zu den vehementesten Gegnern von Flash. <http://www.useit.com/alertbox/20001029.html>

Bauingenieurwesen und Landschaftsarchitektur an der Universität von Tokyo. Dort packte ihn die Begeisterung für das Verhältnis zu seiner Umgebung. Getrieben von dem Verlangen, eine abstrakte Version dieses Verhältnisses zu kreieren, und unter John Maedas Einfluss begann er sich mit dem Computer und seinen interaktiven Möglichkeiten zu beschäftigen. Er gehörte zu den ersten, die die Bedeutung der aufkommenden Browserplugins wie Shockwave/Director, Flash usw. erkannten und diese Möglichkeiten im Internet zu nutzen wussten. Er experimentierte auf eine abstrakte, aber meistens Interface-orientierte Art und Weise und half dadurch mit, einen Trend loszutreten, der heute fast schon zum guten Ton für Interactive Designer und Developer gehört. (Vgl. Nakamura, online)

Wie die meisten Pioniere dieser „Szene“ trat er dem Ganzen mit einer naiven, vollkommen von Kunsttheorie oder anderen akademisch-theoretischen Disziplinen freien Einstellung gegenüber.

Whether or not something is viewed as design or art is just a result, so I'm not really concerned with it. I think the current trend or tendency of the designers and artists is that as long as the end-user is able to touch and feel and that the work arouses some kind of emotion, its category doesn't matter. In that sense, yes I think art and design are melding.

Later on, as the web rapidly shifted towards commercialism, I began to seriously think about how to combine what I like (which is interactive expression) with design. Thus there are two sides – one is the initial enjoyment that I get from interactive expression, and the other is how to then construct it into reasonable design: currently I work trying to find a way between the two. (Patrick, Craftsman, online)

Selbst würde er sich wohl gar nicht in die Reihe der Vertreter der generativen Methode einordnen, sieht er sich doch selbst eher als Interface Designer. Dennoch sind viele seiner Experimente äußerst generativ und haben vom technischen Standpunkt aus das Fundament für eine Heerschar von Nachfolgern gelegt.

## 4.2. Joshua Davis

Joshua Davis ist ein Illustrator/Künstler/(Web)Designer, der in New York lebt und arbeitet. Sein Einstieg war 1995 HTML, kurz darauf entdeckte er Flash und beschäftigte sich ab diesem Zeitpunkt ausgiebig mit den neuen gestalterischen Möglichkeiten, die diese Plattform bietet.<sup>11</sup> Selbst sieht er sich immer noch als der gleiche Künstler, der er früher als Maler war, nur mit neuen (digitalen) Werkzeugen. Er gehört damit zu jenen Künstlern, die keinen fundamentalen, konzeptionellen Bruch zwischen klassischer Kunst und New Media sehen. Der Computer bzw. das Programm ist für ihn einfach ein neues Werkzeug, eine neue Art von Pinsel. Er ist somit konzeptionell und stilistisch sehr nahe an klassischer Malerei und Design und unterscheidet sich primär durch die schiere Komplexität seiner Arbeiten von den manuell erstellten Arbeiten. (Vgl. Davis, Abstraktions-Maschine, 142)

Seine Arbeiten sind immer eindeutig für den Print-Bereich gedacht, weswegen es eben auch nicht verwundert, dass er den Computer „nur“ als Werkzeug, also mächtigeren Pinsel, sieht. Gerade das ist allerdings manchmal ein Kritikpunkt an seiner Arbeit, weil er sich mit seinen Werken für manche zu sehr an den klassischen Kunst- und Designmarkt anbietet, besonders wenn es darum geht, limitierte, signierte Prints seiner Arbeiten herauszugeben. Dafür gibt ihm das die Freiheit, nur sehr wenige kommerzielle Projekte annehmen zu müssen und frei experimentieren zu können.

Was die Inspiration für seine Arbeit betrifft, kommt er ganz klar von einer ästhetischen, von Kunst und Design gesteuerten Seite. Seine Arbeiten sind alle zweidimensional und er verwendet häufig eine Kombination aus vordefinierten Formen und Symbolen, generativer Komposition und manueller Nachbearbeitung und Komposition. Die generativen Konzepte basieren dabei häufig auf Modellen aus der Natur, wie fließendem Wasser, Wind oder Bewegungen von Insekten. (Vgl. Malmberg, Interesting, online)

---

<sup>11</sup> Für eine ausführliche Autobiographie, welche sich allerdings primär mit seinem Leben unabhängig von seiner Arbeit beschäftigt, empfiehlt sich <http://www.k10k.net/issues/issuewarp.aspx?IssueID=100>.

In diesem kurzen Zitat wird seine Sichtweise recht klar:

Von allen modernen Künstlern hat mich Jackson Pollock konzeptuell am meisten beeinflusst. [...] Pollock hat sich stets als Künstler bezeichnet, obwohl sein Pinsel die meiste Zeit die Leinwand gar nicht berührte. (Davis, Abstraktions-Maschine, 142)

Das interessante dabei ist, dass er sich nicht für Pollocks Bilder interessiert, sondern für den Prozess, wie dieser Bilder erstellt. (Vgl. Wizard, 25)

Als weitere Quellen der Inspiration nennt er Künstler wie Jean-Michel Basquait, Cy Twombly und Ito Jakuchu. Er nennt in seinen Vorträgen aber auch immer wieder andere, weitaus ungewöhnlichere Inspirationen, wie Muster auf japanischen Kimonos oder Hotelteppichen. Nur das Internet sieht er, für manche vielleicht überraschend, als schlechte Inspirationsquelle. Er meint, es sei zu nahe an seiner eigenen Arbeit, es würde damit in gewisser Weise zu einer Art „Inzucht“ kommen, weswegen er Inspirationen aus der „Offline-Welt“ den Vorzug gibt. (vgl. Davis, Presentation)

Durch seine große Popularität ist glücklicherweise eine ziemlich präzise Beschreibung seiner Arbeitsweise verfügbar. Seinen Arbeitsprozess beschreibt er als fünfstufigen Ablauf wie folgt:

#### Collecting

There's a period where I'm constantly absorbing things, cataloging them in the back of my brain. I may go on a trip and take pictures of something, and then four months later I actually get around to doing something with them.

#### Drawing

When the time is right, I say, "Okay, I've always wanted to do this thing I saw in Barcelona." But I still have to draw the artwork elements I'll be using, which can take anywhere from two to four days.

### Programming

Then it can take anywhere from two to four more days to write the program that generates a composition from the artwork.

### Refining

After that, I might spend two weeks refining the program, changing some of the variables, eliminating artwork that doesn't work, adding things that do work, rerunning the program, and watching it.

### Generating

The process is long and intense, so what are the benefits? Well, if I made the work in Illustrator alone, I'd only have one composition. If I had to do ten different posters based on the same theme, it would take much longer to do it manually. But with my process, I can output ten different designs in ten minutes. (Vgl. Malmberg, Interesting, online)

## 4.3. Mario Klingemann

Man könnte Mario Klingemann wohl kaum besser vorstellen als er es selbst in einem Interview gemacht hat:

Guten Tag, mein Name ist Mario Klingemann und ich mache schöne Sachen aus Code. Auf meiner Visitenkarte steht „Computational Artisan“, was man auf deutsch wohl am ehesten mit „Rechenkunsthändler“ übersetzen könnte (aber das lassen wir mal lieber). Das Problem ist, dass mir „Künstler“ zu affektiert und „Programmierer“ zu normal ist, aber irgendwo dazwischen bin ich zu finden. Ich bin einer von den Jungs, die mit Lego, Fischertechnik und einem C64 aufgewachsen sind und irgendwann den Absprung verpasst haben. (Reitberger, Klingemann, online)

Im deutschen Sprachraum kann man Mario Klingemann eindeutig zu den bedeutendsten Vertretern der generativen Methode zählen. Zur Zeit arbeitet er für Aviary, eine Firma die eine Suite von RIAs für Computerkünstler entwickelt, an einem Tool namens Peacock<sup>12</sup>, welches laut Eigendefinition ein „Computer algorithm-based pattern generator“ (Vgl. Aviary, online) ist.



Abb. 14: Mit Peacock erstelltes Bild „Dynamic Chaos“

Ohne akademischen Hintergrund gehört Mario Klingemann mit Sicherheit zu jenen Personen welche eine Motivation für ihre Arbeit haben, die frei von akademischen und kunsttheoretischen Überlegungen ist. Sein Ziel ist es, Schönheit und Dinge, die den Betrachter überraschen, zu schaffen. (Vgl. Rhizome, Online )

Es ist schön, dass sich der Begriff „Generative Art“ als Genre immer mehr etabliert, denn endlich kann man seiner Mutter erzählen, was man eigentlich macht. Zur Kunst habe ich ein eher gespaltenes Verhältnis – was vor ein paar Jahren noch reine Experimente waren oder Programmierbeispiele in einem „Graphik auf dem Amiga“-Buch, wird heute an Ausstellungswände projiziert. Da bleibt die Frage, ob die Autoren damals ihr Potenzial

12 Beispielbilder unter <http://flickr.com/photos/quasimondo/sets/72157603292202052>

unterschätzt haben bzw. nicht die richtigen Freunde hatten oder ob die Messlatte heute einfach niedriger hängt. Andererseits finden wir im Bereich der generativen Kunst einzelne Werke, mit denen ein „klassischer“ Künstler früher ein ganzes Lebenswerk begründen konnte. (Reitberger, Klingemann, online)

Nicht zuletzt der Tatsache, dass er keine akademische Ausbildung für das, was er tut, genossen hat, verdankt er wohl seine offene Herangehensweise. Seine Inspiration bezieht er offensichtlich aus einer breiten Palette zwischen der naturwissenschaftlich-technischen Seite mit Mathematik, Physik und Biologie und der kreativ-künstlerischen Seite mit Musik und klassischer Kunst.

#### 4.4. Marius Watz

Er ist der Theoretiker unter den generativen Künstlern. Was seine Herangehensweise betrifft, ist er das europäische Pendant zu Maeda (wenn auch nicht ganz so bedeutend) und der Gegenpol zu Joshua Davis. Er ist informiert über Kunsttheorie und Kunstgeschichte, liest theoretische Abhandlungen wie Philip Galanters Arbeiten und ist dennoch keiner von jenen verstaubten Universitätsprofessoren, die nur in der Theorie beheimatet sind.

Die Ursprünge seiner Arbeit sind auf die Techno Rave Szene der frühen 90er und ihre starke Auseinandersetzung mit dem Medium Computer zurückzuführen. Nachdem er 1993 sein begonnenes Informatikstudium abgebrochen hatte, begann er Grafiken für Technoraves zu entwerfen. Dabei entdeckte er die neuen Möglichkeiten des visuellen Ausdrucks des Computers und die Chance sein Interesse für das Programmieren dafür einzusetzen. Wie für viele andere begründete sich sein Interesse stark aus der äußerst problematischen Kombination von Interesse am visuellen Ausdruck und dem Mangel an handwerklichen Fähigkeiten und Talent beim Zeichnen und Malen. (Vgl. Watz, Generative, online und Biografie, online)

Was seinen visuellen Stil betrifft, ist seine Arbeit abstrakt und bezieht sich daher nicht explizit auf irgendetwas außerhalb des Werkes selbst. Diese Selbstbezüglichkeit und die volle Konzentration auf das Medium Computer in sich ordnet ihn jenen Künstlern zu, die den Schritt gewagt haben sich mit beiden Beinen in das neue Medium zu begeben.

Dennoch hat er natürlich auch Inspirationsquellen innerhalb der realen Welt. Seine ersten Anfänge sich mit der generativen Methodik zu beschäftigen, schildert er selbst folgendermaßen:

My first experience of wanting to articulate a complex algorithm came when I stood as a child under a street light in heavy snowfall. Looking up at the constantly shifting spirals formed by the snow falling, I had the sense that it must be possible to describe the forces causing those chaotic yet recognizable forms. That sensation of being just on the verge of understanding is always there when I try to create new work. (Thalmair, Mimesis, online)

Nach seinem Start in der Musikszene, wo er sich später auch als VJ betätigte, gründete er 2005 die Online-Plattform Generator.x, welche sich auf das Thema generativer Strategien und Softwareprozesse in digitaler Kunst, Architektur und Design konzentriert. Es folgten drei Konferenzen in den Jahren 2005, 2006 und 2008. (Vgl. Watz, Generator.x, online)

Heute hat er den Sprung vom Hobby zum Vollzeit-Künstler geschafft. (Vgl. Ravera, Aesthetic, online)

## 4.5. Jonathan Harris

Mit Jonathan Harris begeben wir uns jetzt am weitesten an den Rand des Bereichs der Vertreter der generativen Grafik. Selbst würde sich Harris mit ziemlicher Sicherheit nicht in diese Reihe einordnen, sieht er sich doch primär als Geschichtenerzähler. Dennoch bildet er, wie wir bald sehen werden, den perfekten Übergang zum nächsten Kapitel, den praktischen Anwendungen der generativen Grafik.

Jonathan Harris studierte Informatik an der Princeton University, New Jersey, USA und arbeitet nach einem kurzen Aufenthalt im Benetton Think Tank Fabrica selbstständig für verschiedene bedeutende Auftragsgeber wie New Yorks Museum of Modern Art oder Yahoo. In seinen Arbeiten kombiniert er Elemente aus Informatik, Anthropologie, Kunst und dem Erzählen von Geschichten. (Vgl. Harris, biography, online)

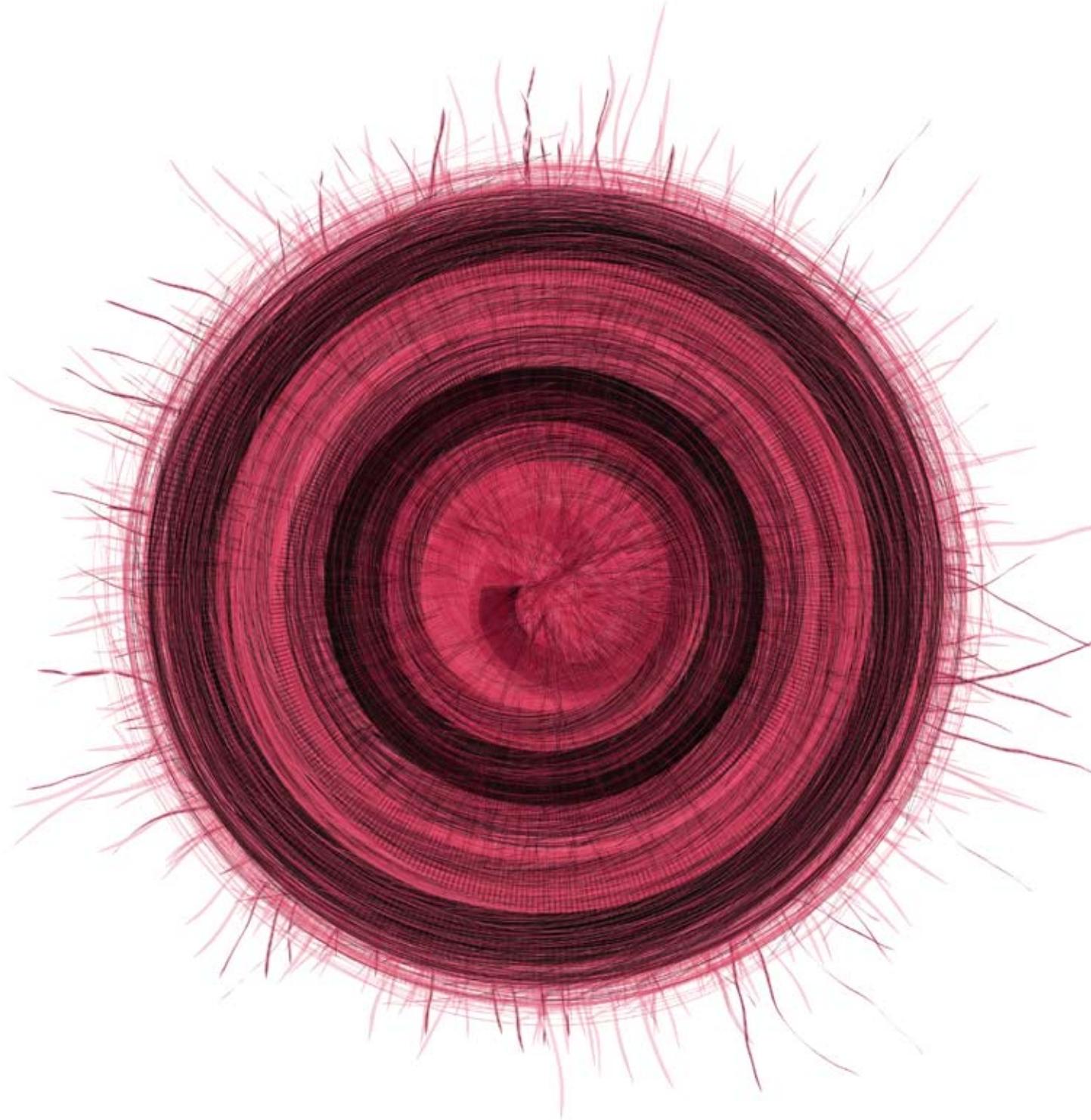
Er vertritt die Ansicht, dass die Metapher der Seite tot ist und die Menschheit andere, neue Organisationsstrukturen benötigt. Auch wenn er der Meinung ist, die Lösung sei noch nicht gefunden, glaubt er, dass es sich, einmal gefunden, um ein globales Werkzeug im Kontext des Internets handeln wird. Selbst forscht er in Richtung vernetzter Strukturen abseits der etablierten linearen Methoden. (Vgl. m ss ng p eces, Harris, online)

Sein Ziel ist es dabei in einer Welt, die täglich mehr Informationen und Daten sammelt, diese auf eine menschliche Art und Weise zu visualisieren. (Vgl. Danzico, stories, online) Wie zuvor erwähnt befinden wir uns dabei eindeutig am Rand generativer Grafik, manche würden auch sagen, wir haben sie bereits verlassen. Dennoch ist seine Arbeit immer eine visuelle Interpretation von Daten, Daten die nach gewissen Regeln und Strukturen automatisch visualisiert werden. Aus diesem Grund kann man ihn durchaus auch im Bereich der generativen Grafik einordnen.

Was ihn in seiner Herangehensweise mit Sicherheit von den zuvor genannten Personen unterscheidet, ist die Tatsache, dass seine Arbeiten zwar in ihrer Visualisierungsform oftmals abstrakt sind, allerdings niemals als ganze autarke selbstbezügliche Werke fungieren. Harris ist es auch nach eigener Aussage am wichtigsten den menschlichen Faktor in den Vordergrund zu stellen. Die Herausforderung besteht dabei darin, seinem auf den ersten Blick sehr kalt wirkenden Ausgangsmaterial eben diesen menschlichen Faktor zu entlocken. (Vgl. Danzico, stories, online)

## 4.6. Zusammenfassung

Zusammenfassend lässt sich also nochmal die aus dem Fakt, dass es sich bei generativer Grafik nicht um ein Genre, sondern eine Methode handelt, resultierende Tatsache festhalten, dass die Vertreter dieser Methode ein breites künstlerisches Spektrum darstellen. Von Interaktionsdesignern wie Yugop über Künstler wie Joshua Davis, technisch experimentierfreudige Autodidakten wie Mario Klingemann und theorieinteressierte Technik-Künstler wie Marius Watz kommen wir schließlich zum Menschheits-Visualisierer Jonathan Harris. Dieser bildet auch gleichzeitig den perfekten Übergang zum nächsten Kapitel, in dem die Frage nach den möglichen (kommerziellen) Anwendungsgebieten der generativen Methode behandelt wird.



## 5. Mögliche Anwendungsgebiete

Theoretisch gibt es unendlich viele Möglichkeiten, wie man generative Methodik zur Erstellung von Grafiken einsetzen kann. Dennoch sollen in diesem Kapitel, aus der Notwendigkeit produktive Einsatzszenarien aufzuzeigen, einige konkrete Anwendungsbereiche kurz vorgestellt werden. Denn nicht ohne Grund war die Frage nach der (kommerziellen) Nutzbarkeit abseits der Kunst in den Entstehungsmonaten dieser Arbeit in wechselnder Ausprägung mit Abstand die am häufigsten auftauchende Frage. Eine besonders populäre Anwendungsmöglichkeit wird schließlich im Kapitel 5.1 im Detail vorgestellt werden.

Wie meistens, wenn es um innovative Ideen geht, ist auch bei der generativen Grafik die Werbebranche bereits auf sie aufmerksam geworden. Individualisierte Werbungen sind besonders bei Online-Kampagnen zwar nichts Neues, durch die technischen Weiterentwicklungen der letzten Jahre und die damit verbundenen Möglichkeiten erleben sie aber gerade in letzter Zeit wieder ein gesteigertes Interesse. Joshua Davis hat beispielsweise mit „Amp Yourself“<sup>13</sup> gezeigt, wie sich generative Grafik ausgezeichnet kommerziell verwerten lässt.

Auch bei VJ's<sup>14</sup> sind generative Methoden selbstverständlich sehr populär. Obwohl wir hier nicht mehr im Bereich von generativer Grafik sind, sollte man diese Anwendung kurz erwähnen. Wenn man die Parameter einer generativen Grafik während des Entstehungsprozesses verändert, und das Ganze

---

<sup>13</sup> <http://www.ampenergy.com/ampyourself>

<sup>14</sup> VJ ist die Abkürzung für Visual Jockey. VJs verwenden Computer, um in Echtzeit ansprechende visuelle Performances zu gestalten. Dazu greifen sie auf Videoclips, Animationen aber auch auf eigene Programme, die individuell auf Musik reagieren, zurück. Auf diese Art und Weise treten sie primär in Clubs und bei Musikveranstaltungen als Ergänzung zur musikalischen Unterhaltung durch DJs auf.

◀ Abb. 15: PinkSwirl, mit dem Werkstück generiert (PinkSwirl.xml)

in einem zeitlichen Kontext betrachtet, entsteht ohnehin nichts anderes als eine Animation, ein Video.

Generative Techniken werden auch immer öfter bei professionellen Musikvideos eingesetzt. Radiohead stellt beispielsweise Material für Experimentierfreudige zur Verfügung. Menschen wie Robert Hodgin mit seinem Video zu dem Radiohead Song „Weird Fishes/Arpeggi“ zeigen dann, was man mit generativen Methoden erschaffen kann. (Vgl. Hodgin, Radiohead, online)

In Anbindung an das letzte Kapitel soll hier die im Rahmen der Vorstellung von Jonathan Harris bereits erwähnte Datenvisualisierung als ein mögliches Anwendungsszenario genauer vorgestellt werden.

## 5.1. Datenvisualisierung

Im Informationszeitalter, in dem wir uns heute zweifellos befinden, ist die Verarbeitung von Daten, also Information zu einer der größten Menschheitsaufgaben geworden. Waren es in früheren Zeiten primär die großen linearen Zusammenhänge der Mythen und Religionen und ein kleines Maß an allein-stehenden Informationen die die Menschheit prägten, so haben wir es heute mit einer stetig wachsenden Informationsflut zu tun, welche es zu organisieren gilt. Zugriff und Ordnung von Informa-tionen ist daher eine der größten Herausforderungen unserer Zeit. Es ist immer mehr notwendig eine „Infor-mationsästhetik“ zu entwickeln und dafür Werkzeuge zu schaffen, um die Datenflut anhand dieser Ästhetik verstehen und erfassen zu können. (Vgl. Manovich, Media, 213-217)

Datenbanken bzw. das Modell der Datenbank als kulturelles Phänomen sind zum zentralen Informa-tionsträger des Informationszeitalters geworden. Wenn man sie den Objekten der physikalischen Welt gleichsetzt, entspräche dem Prozess oder der Aufgabe der Algorithmus. Mit Hilfe von Algorithmen ist es möglich, verschiedene Interfaces zu den ohne eine fixe Struktur gegebenen Informationen einer Da-tenbank zu schaffen. Jedes dieser Interfaces erlaubt einen anderen Blick auf die zu Grunde liegenden Daten und macht sie dadurch verständlicher. (Vgl. Manovich, Media, 218-228)

Die Verbindung zur generativen Grafik zeigt sich hier insofern, als diese die Mittel liefert, die zuvor erwähnten Interfaces zu schaffen. Für die Informationsvisualisierung benötigt man Software, die in der Lage ist, Daten zu verarbeiten, sie nach gewissen Regeln zu strukturieren, umzuformen und schließlich visuell zu repräsentieren. Vergleichen wir diese Anforderung nun mit der Definition von generativer Grafik:

Eine generative Computergrafik ist jedes in gewisser Weise ästhetisches graphisch-visuelles Werk, welches durch die automatische Ausführung eines Programms, welches die Implementation einer beliebigen Anzahl von in Form von Algorithmen definierter Regeln und Anweisungen, oder Kombinationen und Variationen eben jener ist, entsteht.

Wendet man diese Definition in einer Form an, bei der das visuelle Ergebnis eine Repräsentation von Informationen darstellt, hat man die genaue Definition von Informationsvisualisierung.

Natürlich ist das Gebiet der Informationsvisualisierung wiederum ein äußerst umfangreiches, welches durchaus die Behandlung im Rahmen einer eigenen Arbeit verdienen würde. Aus diesem Grund wird diese Arbeit in dieser Hinsicht nicht weiter ins Detail gehen.<sup>15</sup>

Es soll hier abschließend allerdings noch ein konkretes Beispiel der Informationsvisualisierung als Anwendungsgebiet generativer Grafik präsentiert werden. Nachdem Jonathan Harris im vorigen Kapitel bereits als bedeutender Vertreter der Datenvisualisierung vorgestellt wurde, bietet es sich an, gleich daran anzuknüpfen und eine seiner Arbeiten vorzustellen.

„We Feel Fine“ ist ein zusammen mit Sepandar Kamvar 2006 veröffentlichtes Projekt zur Visualisierung und Erforschung menschlicher Emotionen im globalen Rahmen. Zu diesem Zweck werden im

---

15 Für eine ausführlichere Behandlung dieser Themen empfiehlt sich information aesthetics. FORM FOLLOWS DATA – DATA VISUALIZATION & VISUAL COMMUNICATION, <http://infosthetics.com>, zuletzt aufgerufen am 27.8.2008 sowie Fry, Ben: Visualizing Data, Sebastopol: O-Reilly Media 2008 und Tufte, Edward R: The Visual Display of Quantitative Information, Cheshire: Graphics Press 2007

großen Rahmen Blogs nach den Phrasen „I feel“ und „I am feeling“ durchsucht und die daraus resultierenden Emotionen auf verschiedene Weisen anschaulich gemacht.

Ganz nach dem Konzept von Manovich stellt „We Feel Fine“ dabei verschiedene Interfaces zu den zu Grunde liegenden Daten zur Verfügung. Man kann nach verschiedenen Kriterien wie Alter, Geschlecht oder Ort filtern und sich die daraus resultierenden Datensätze auf verschiedene Weisen anzeigen lassen. Prinzipiell wird jede Äußerung als farbiger Partikel dargestellt, wobei Farbe, Größe, Form und Transparenz die genaue Art der ausgedrückten Emotion repräsentieren. (Vgl. Harris, Mission, online)

Die sechs verschiedenen Darstellungsformen decken ein reiches Spektrum ab, sowohl im Sinne der visuellen Repräsentation als auch der inhaltlichen Bedeutung. Die erste Darstellungsform von chaotisch herumschwirrenden Partikel stellt einen eher spielerischen, wertungsfreien Ansatz dar, welcher allerdings verglichen mit den anderen Darstellungsformen den in gewisser Weise intensivsten visuellen Eindruck vermittelt. Weiter gibt es sehr strukturierte Ansätze einer chronologischen Liste, einer Montage von Bildern mit zugehörigen Emotionen und Mobs in Form von nach den Filterkriterien sortierten Gruppen. Die sechste und letzte Darstellung greift im Gegensatz zu allen anderen nicht auf eine aktuelle Stichprobe zu, sondern repräsentiert in Form von Hügeln die Häufigkeit der verschiedenen Informationen seit dem Start des Projekts. (Vgl. Harris, Movements, online)

Selbst wenn man die anthropologische Bedeutung dieses Projektes beiseite lässt und sich voll und ganz auf die Ästhetik konzentriert, bilden die tanzenden, leuchtenden Partikel in ihren verschiedenartigen Choreographien ein sehenswertes visuelles Erlebnis. Mit ihrer zu Grunde liegenden generativen Methode ist die Datenvisualisierung somit ein interessantes realitäts- und anwendungsbezogenes Beispiel für generative Grafik.

## 6. Das Werkstück

Das sechste und letzte Kapitel dieser Arbeit widmet sich nun dem dieser Arbeit beiliegenden Werkstück. Zu Beginn werden die Anforderungen an das Werkstück definiert. Darauf aufbauend wird ein technisches Konzept entwickelt und eine konkrete technische Ausführung erarbeitet. Eine ausführliche technische Dokumentation findet sich in digitaler, durchsuchbarer Version auf der beiliegenden DVD.

### 6.1. Anforderungen

Die Definition von generativer Grafik im Allgemeinen wurde bereits in Kapitel 3 behandelt. Die nun folgenden Überlegungen und Einschränkungen sind also einzig und allein auf das Werkstück zu dieser Arbeit zu beziehen. Da es nicht möglich ist, das gesamte Spektrum an generativen Vorgehensweisen zu demonstrieren, werden hier in Folge einige Einschränkungen vorgenommen, die zu einem konkreten Werkstück führen sollen.

- Es sollen keine wie auch immer gearteten zuvor generierten Medien wie z.B. Fotos, Videos, 3D Modelle usw. eingesetzt werden. Alle Darstellungen müssen direkt und individuell generiert werden und auf geometrischen Grundformen basieren.
- Da das Ausgabemedium, egal ob Bildschirm, Projektion oder Leinwand, zweidimensional ist, soll auch die generative Grafik zweidimensional sein. Manchmal können sich aber natürlich aus zweidimensionalen Elementen räumliche Eindrücke entwickeln, die aber niemals ausdrückliches Ziel sein sollen.
- Es soll zur Laufzeit des der generativen Grafik zu Grunde liegenden Programms eine Einflussnahme möglich sein. Entweder soll der Benutzer durch direkte Interaktion, wie z.B. mit der Maus, die Möglichkeit haben, den Ablauf zu beeinflussen oder das Programm soll selbst durch die Ein-

holung von Informationen oder externen Parametern unterschiedlich reagieren können.

- Das entwickelte System soll soweit modular aufgebaut sein, dass es nicht nur für ein konkretes Ergebnis geeignet ist. Für wiederkehrende Aufgaben wie Farbmanagement oder geometrische Grundformen sollen allgemein taugliche Klassen zur Verfügung stehen.
- Da es Ziel ist, das Werkstück unter einer Open Source Lizenz zur Verfügung zu stellen, müssen auch alle etwaigen externen Klassen unter einer derartigen Lizenz zur Verfügung stehen.
- Sofern ein GUI notwendig wird, sollte ein möglichst minimalistischer Ansatz gewählt werden.
- Um das Werkstück einem möglichst breiten Publikum zur Verfügung stellen zu können, soll es möglich sein Projekte mittels Kombination bereits existierender Elemente ohne die Notwendigkeit selbst zu programmieren zu müssen zu erstellen. Dies soll allerdings in keiner Weise die Möglichkeiten zur Erweiterung mittels eigenem Programmcode einschränken.

## 6.2. Aufbau

### 6.2.1. Geometrische Grundformen - Zeichnungsfunktionen

Flash bietet besonders seit Flash Player 10 eine ausgezeichnete Zeichnungs-API. Nativ stehen Zeichnungsfunktionen für Linien, Bézierkurven<sup>16</sup>, Rechtecke, Kreise, Ellipsen und Dreiecke zur Verfügung. Die einzige zusätzlich benötigte Funktionalität wird daher die automatische Erstellung einer Bézierkurve anhand von Endpunkten ohne explizite Definition von Kontrollpunkten sein.

---

<sup>16</sup> Unter einer Bézierkurve versteht man eine parametrisch modellierte Kurve. Sie ermöglicht es anhand von zwei Endpunkten und einem (quadratische Bézierkurve) oder zwei (kubische Bézierkurve) Kontrollpunkten eine „weiche Kurve“ zwischen den zwei Punkten zu zeichnen. So ist es möglich eine beliebige Anzahl von Punkten durch eine „weiche, fließende Kurve“ zu verbinden. (Vgl. Bézierkurve, Wikipedia)

## 6.2.2. Farbmanagement

Flash bietet mit der ColorMatrix eine gute Möglichkeit Farben zu manipulieren, allerdings immer nur anhand eines bereits existierenden DisplayObjects. Zusätzlich benötigt wird daher eine spezielle Farb-Klasse, die die Umrechnung von beliebigen Farbwerten zwischen RGB, HEX, HSV und HSL ermöglicht. Jede Farbe soll außerdem anhand der einzelnen Parameter wie z.B. Helligkeit, Kontrast oder Farbwert veränderbar sein.

Instanzen der Farb-Klasse sollen sich wiederum zu Paletten zusammenfassen lassen. Diese Paletten sollen sowohl statisch definierte Paletten sowie auch Paletten welche sich auf Regeln basierend selbst füllen ermöglichen.

## 6.2.3. Exportfunktionen

Adobe bietet im Flash Player 10 und in AIR die Möglichkeit Bilder im JPEG und PNG Format zu exportieren. Mittels eines Druckertreibers der PostScript unterstützt lassen sich aus dem Flash Player 10 Vektordaten im eps-Format exportieren.

## 6.2.4. GUI

Es soll basierend auf den MinimalComps von Keith Peters (Vgl. Peters, minimalcomps, online) ein externes Kontroll-GUI entwickelt werden, welche verschiedene Parameter extern steuern lässt. Dieses GUI soll optional sein, d.h. die Basisfunktionen, wie das pausieren und wiederaufnehmen des Generierungsprozesses, das Laden von Projekten sowie der Export von Bildern muss auch ohne das GUI möglich sein.

## 6.2.5. Organisations-Framework

Das Werkstück wird aus vier Ebenen aufgebaut sein. Die oberste bildet die Verwaltungsfunktionalität, welche die Unterelemente verwaltet, organisiert und aufruft.

Die drei anderen Elemente sind modular in Serie geschaltet.

### Stufe 1 – Lineserver

Jeder Lineserver liefert einen stetigen Strom aus 2D-Koordinaten. Diesen kann er entweder anhand vorgegebener Kriterien selbst generieren, aus externen Live-Quellen wie z.B. der Mausposition beziehen oder aus einer Aufzeichnung wiedergeben.

Jedem Lineserver können wiederum PointTransformer und Painter zugeordnet werden.

### Stufe 2 - PointTransformer

PointTransformer haben die Aufgabe, die vom LineServer gelieferten Koordinaten zu verändern. Diese Veränderung kann vollkommen frei definiert werden, solange wieder x- und y-Koordinaten ausgegeben werden.

### Stufe 3 - Painter

Der Painter schließlich stellt die gelieferten Koordinaten auf eine beliebige Art und Weise dar. Das kann von der simpelsten Variante, die Punkte einfach als Linie zu verbinden, bis zu komplexen um die Linie kreisenden Partikeln führen – der Fantasie sind keine Grenzen gesetzt.

## 6.3. Verwendung des Werkstücks

Da der Quellcode des Werkstücks öffentlich verfügbar sein wird, ist die offensichtlichste Möglichkeit ein eigenes Projekt zu erstellen die programmatische Erweiterung. Durch den modularen Aufbau ist es ohne großen Aufwand möglich eigene Lineserver, PointTransformer oder Painter zu erstellen.

Für jene Leser die nicht über die nötigen Programmierkenntnisse verfügen, oder sich nicht erst in die technische Struktur des Werkstücks einarbeiten möchten, besteht noch eine zweite Möglichkeit. Sie können durch Kombination und Konfiguration der bestehenden Elemente eigene Projekte zu erstellen. Dies lässt sich durch die im Vergleich zur Programmierung weitaus einfachere Erstellung einer Konfigurations-XML-Datei bewerkstelligen. Auch wenn auf der DVD eine weitaus umfassendere technische Dokumentation vorliegt, soll an dieser Stelle anhand eines einfachen Beispiels die Erstellung eines eigenen Projektes demonstriert werden.

Das Grundgerüst einer Konfigurations-XML-Datei sieht folgendermaßen aus:

```
<justlineon>
  <title></title>
  <description></description>
  <author></author>
  <bgcolor></bgcolor>
  <project>
    <lineserver>
      <pointtransformer></pointtransformer>
      <painter>
        <palette></palette>
      </painter>
    </lineserver>
  </project>
</justlineon>
```

Wie man sehen kann beginnt eine solche Datei mit einigen Beschreibungselementen. Diese sind für den Ablauf des Programms ohne Bedeutung. Danach folgt das eigentliche Projekt. Es besteht aus

einer beliebigen Anzahl von Lineservern. Diese Lineserver können wiederum eine beliebige Anzahl von PointTransformern und Paintern enthalten. Die Painter wiederum können eine beliebige Anzahl von Farbpaletten enthalten.

Jedes dieser Elemente verfügt über gewisse Parameter. Bis auf den type Parameter sind alle Parameter optional und überschreiben die vorgegeben Standardwerte auf die zurückgegriffen wird, falls sie nicht definiert werden.

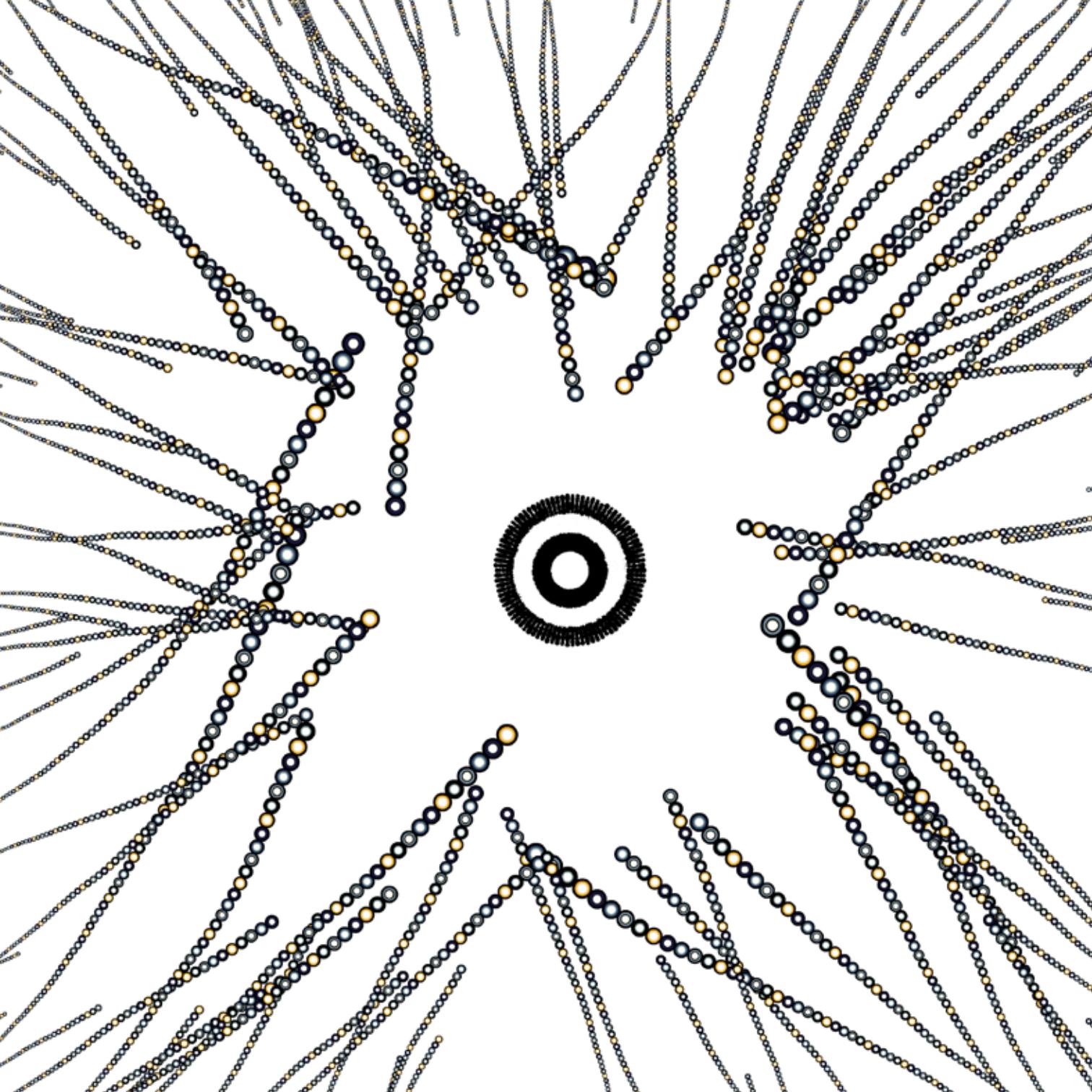
Ein läuffähiges Beispiel für ein Projekt wäre daher bereits folgendes:

```
<justlineon>
  <title>Beispielprojekt</title>
  <description>Beispiel für Seite 60 der Diplomarbeit.</description>
  <author>Thomas Kräftner</author>
  <bgcolor>0xFFFFFF</bgcolor>
  <project>
    <lineserver type="AlwaysTurn" angleDegreeMin="28" angleDegreeMax="28">
      <pointtransformer type="PointPlusRandom"></pointtransformer>
      <painter type="RibbonPainter">
        <palette type="dynamicRandom"></palette>
      </painter>
    </lineserver>
  </project>
</justlineon>
```

Es ist also durchaus möglich ohne umfassende Programmierkenntnisse eigene Werke zu erstellen. Für all jene, die sich gerne weiter mit dem Werkstück befassen möchten, finden sich auf der beiliegenden DVD eine umfassende Dokumentation und einige Beispielprojekte als Ausgangspunkt für eigene Experimente.

Abb. 16: Beispielprojekt, mit dem Werkstück generiert (Beispielprojekt.xml) ▶





## 7. Conclusio

Zu Anfang dieser Arbeit stand der Algorithmus und damit in Zusammenhang die generative Methode. Es zeigte sich dann, dass diese, auch wenn sie prinzipiell schon seit langem existiert und verwendet wird, erst durch die Computerisierung unserer Kultur ihr volles Potential entfaltet. Diese Erkenntnis führte zum eigentlichen Kerngebiet, der generativen Computergrafik. Deren Vertreter demonstrieren ein breites Spektrum an Herangehensweisen und kreativen Ergebnissen welche das enorme Potential der generativen Methode, auch in Hinblick auf mögliche Anwendungsgebiete, aufzuzeigen.

Dieses Potential, welches in der generativen Methode schlummert, aufzuzeigen, ist es schließlich, was diese Arbeit bezweckt. Denn auch wenn, oder gerade weil, die Computerisierung immer schneller voranschreitet, hinken die gestalterischen Herangehensweisen den technischen Möglichkeiten leider oft stark hinterher. Innovative Methoden, wie die der generativen Grafik, bieten aber dabei die Chance, Neues zu kreieren. Diese bietet eine Vielzahl an Vorteilen oder zumindest an erfrischend neuen Ansätzen gegenüber älteren Methoden der Medienproduktion.

Die generative Methode sieht Programmcode nicht als kaltes technisches Konstrukt sondern als kreatives Material. Das Besondere an diesem Material ist dabei, dass es nicht Träger für ein Resultat, ein statisches Ergebnis ist, sondern den Erstellungsprozess selbst repräsentiert. Es ermöglicht nicht nur eine konkrete Repräsentation einer Idee zu zeigen, es ist viel mehr die Manifestation der Idee selbst. Die daraus resultierenden Vorteile sind umfassend.

Man arbeitet nur noch an Ideen selbst und nicht direkt am Ergebnis, denn der eigentliche Erstellungsprozess wird an von selbst ablaufende (Computer-)prozesse delegiert. Das bedeutet in erster Linie eine Beschleunigung des Arbeitsablaufs, aber nicht nur das. Wenn die Last der tatsächlichen Durchführung von einem genommen wird, kann man sich um so mehr auf die Entwicklung, Ausformung und Perfek-

◀ Abb. 17: Conclusio, mit dem Werkstück generiert (Conclusio.xml)

tion von Ideen konzentrieren. Möchte man eine Arbeit verbessern, muss man nur die Idee adaptieren. Die Produktion des Ergebnisses geschieht von alleine. Variationen oder Rekombinationen bedeuten daher keinen erhöhten Produktionsaufwand sondern nur Anpassungen der Idee. Auch die Komplexität kann enorm gesteigert werden. Bei einem von Hand erstelltem Werk bedeutet jedes Element der selben oder einer ähnlichen Sorte einen eigenen Arbeitsschritt. Beim Einsatz generativer Methoden bedarf es nur noch einer minimalen Variation der Grundregel und die Übergabe der Idee an die automatische Ausführung.

Durch die fortschreitende Digitalisierung der Erstellungsprozesse verschiedenster Medienprodukte ist bereits ein Schritt in Richtung Arbeitserleichterung gemacht worden. In Form von individualisierter Werbung oder durch die VJ-Szene zeigt sich dies auch schon im kommerziellen Einsatz. Die generative Methode in einem breiteren Ausmaß einzusetzen ist daher nur der nächste logische Schritt. Änderungswünsche eines Kunden werfen einen so nicht wieder zurück an den Ausgangspunkt der Arbeit, denn die Ausführung passiert von alleine. Hoch komplexe Entwürfe sind auch ohne enormen Zeit- und Materialaufwand realisierbar. Und durch bewusst einsetzbare Anomalien lässt sich sogar ein kontrolliertes Überraschungsmoment erreichen.

Diese Eigenschaften der generativen Grafik ermöglichen es, den gestalterischen Prozess von Designern, Künstlern und allen anderen Medienschaffenden nachhaltig zu bereichern. Um diese Bereicherung zu veranschaulichen und für jedermann begreifbar zu machen, wurde schließlich auch das Werkstück geschaffen.

All das geschieht in der Hoffnung, dass diese Arbeit ihren Zweck erfüllt und dafür sorgt, dass die generative Methode weitere Verbreitung findet.





# 8. Anhänge

## 8.1. Literaturverzeichnis

„Bézierkurve“, in: Wikipedia, <http://de.wikipedia.org/wiki/B%C3%A9zierkurve>, zuletzt aufgerufen am 27.8.2008

„Infinite Monkey Theorem“, in: Wikipedia, [http://de.wikipedia.org/wiki/Infinite\\_Monkey\\_Theorem](http://de.wikipedia.org/wiki/Infinite_Monkey_Theorem), zuletzt aufgerufen am 10.8.2008

Adobe: What is Flash CS3 Professional?, [http://www.adobe.com/products/flash/ssi/iframe/product\\_overview.html](http://www.adobe.com/products/flash/ssi/iframe/product_overview.html), zuletzt aufgerufen am 7. 7. 2008

Aviary Tools, <http://a.viary.com/tools>, zuletzt aufgerufen am 10.8.2008

Co, Elise: jenseits von Pixeln, in: Maeda, John: Creative Code. Ästhetik und Programmierung am MIT Media Lab, Basel: Birkhäuser 2004, 172

Danzico, Liz: Telling stories using data: An interview with Jonathan Harris, <http://www.adobe.com/designcenter/thinktank/danzico2.html>, zuletzt aufgerufen am 10.8.2008

Davis, Joshua: Dynamische Abstraktions-Maschine, in: Maeda, John: Creative Code, Basel: Birkhäuser 2004, 142

Davis, Joshua: Presentation 2008, [http://workshop.joshuadavis.com/presentation\\_2008.zip](http://workshop.joshuadavis.com/presentation_2008.zip), zuletzt aufgerufen am 10.8.2008

Die Neue Sammlung München - FAQ - Graphisches Erscheinungsbild, [http://www.die-neue-sammlung.de/z/muenchen/faq/m-o\\_de.htm](http://www.die-neue-sammlung.de/z/muenchen/faq/m-o_de.htm), zuletzt aufgerufen am 16.9.2008

Dietrich, Frank: Visual Intelligence: The First Decade of Computer Art (1965-1975), in IEEE Computer

Graphics and Applications (Juli 1985), 33-45

Galanter, Philip: What is Generative Art? Complexity Theory as a Context for Art Theory, [http://www.philipgalanter.com/downloads/ga2003\\_paper.pdf](http://www.philipgalanter.com/downloads/ga2003_paper.pdf), zuletzt aufgerufen am 10.8.2008

Harris, Jonathan: Full biography, <http://number27.org/morebio.html>, zuletzt aufgerufen am 10.8.2008

Harris, Jonathan: Mission, <http://www.wefeelfine.org/mission.html>, zuletzt aufgerufen am 10.8.2008

Harris, Jonathan: Movements, <http://www.wefeelfine.org/movements.html>, zuletzt aufgerufen am 10.8.2008

Himmelsbach, Sabine: Softwarekunst, [http://www.zkm.de/algorithmische-revolution/index.php?module=pagemaster&PAGE\\_user\\_op=view\\_page&PAGE\\_id=31](http://www.zkm.de/algorithmische-revolution/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=31), zuletzt aufgerufen am 12.8.2008

Hodgin, Robert: Visualizing Radiohead, <http://www.flight404.com/blog/?p=121>, zuletzt aufgerufen am 11.9.2008

Hope, Jarrad: Bloom, <http://getbloom.com>, zuletzt aufgerufen am 12.8.2008

Ihmels, Tjar: Die Methodik der generativen Kunst, [http://www.medienkunstnetz.de/themen/generative\\_tools/generative\\_art/scroll](http://www.medienkunstnetz.de/themen/generative_tools/generative_art/scroll), zuletzt aufgerufen am 2. 7. 2008

Joshua Davis. The Wizard of Mineola, in: Creativity Februar 2007, 25

Klingemann, Mario: Flash to Processing Communication, [http://incubator.quasimondo.com/processing/flash\\_processing\\_communication\\_1.php](http://incubator.quasimondo.com/processing/flash_processing_communication_1.php), zuletzt aufgerufen am 10. 8. 2008

Klingemann, Mario: Mario Klingemann / Quasimondo, <http://www.mario-klingemann.com>, zuletzt aufgerufen am 2. 7. 2008

Levin, Golan: Ist der Computer ein Werkzeug?, in: Maeda, John: Creative Code. Ästhetik und Programmierung am MIT Media Lab, Basel: Birkhäuser 2004, 140

m ss ng p eces, Jonathan Harris, [http://www.coolhunting.com/archives/2007/03/jonathan\\_harris.php](http://www.coolhunting.com/archives/2007/03/jonathan_harris.php), zuletzt aufgerufen am 10.8.2008

Maeda, John: Creative Code. Ästhetik und Programmierung am MIT Media Lab, Basel: Birkhäuser 2004

Maeda, John: Maeda@Media, London: Thames&Hudson 2000

Malmberg, Elise: Joshua Davis: Infinitely Interesting, <http://www.apple.com/pro/profiles/joshuadavis>, zuletzt aufgerufen am 10.8.2008

Manovich, Lev: The language of New Media, Cambridge: MIT Press 2001

Maus, Eckart: Algorithmus, [http://de.encarta.msn.com/encyclopedia\\_761574357/Algorithmus.html](http://de.encarta.msn.com/encyclopedia_761574357/Algorithmus.html), zuletzt aufgerufen am 10.8.2008

Patrick, Yugo Nakamura: The Craftsman, <http://www.creativereview.co.uk/crblog/yugo-nakamura-the-craftsman>, zuletzt aufgerufen am 10.8.2008

Peters, Keith: minimalcomps, <http://code.google.com/p/minimalcomps>, zuletzt aufgerufen am 27.8.2008

Pixel Bender Toolkit, [http://labs.adobe.com/wiki/index.php/Pixel\\_Bender\\_Toolkit](http://labs.adobe.com/wiki/index.php/Pixel_Bender_Toolkit), zuletzt aufgerufen am 12.8.2008

Processing 1.0 (BETA), <http://processing.org/>, zuletzt aufgerufen am 7. 7. 2008

Ravera, Elena: Marius Watz: Code, Art & Aesthetic, [http://www.digicult.it/digimag\\_eng/article.asp?id=804](http://www.digicult.it/digimag_eng/article.asp?id=804), zuletzt aufgerufen am 10.8.2008

Reas, Casey: Processing : a programming handbook for visual designers and artists, Cambridge: MIT Press 2007

Reitberger, Frank: Mario Klingemann im Interview, <http://createordie.de/cod/artikel/Mario-Klingemann-im-Interview-001843.html>, zuletzt aufgerufen am 10.8.2008

Rhizome | Mario Klingemann, <http://rhizome.org/profile.php?1030968>, zuletzt aufgerufen am 10.8.2008

Simon, John Jr.: Urheberschaft, Kreativität und Code, in: Maeda, John: Creative Code. Ästhetik und Programmierung am MIT Media Lab, Basel: Birkhäuser 2004, 46

Szope, Dominika: Kinetische und Optische Kunst, [http://www.zkm.de/algorithmische-revolution/index.php?module=pagemaster&PAGE\\_user\\_op=view\\_page&PAGE\\_id=30](http://www.zkm.de/algorithmische-revolution/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=30), zuletzt aufgerufen am 11.8.2008

Thalmair, Franz: A form of technological mimesis, [http://www.furtherfield.org/displayreview.php?review\\_id=304](http://www.furtherfield.org/displayreview.php?review_id=304), zuletzt aufgerufen am 10. 8. 2008

Visual Minds, About SourceBinder, <http://www.sourcebinder.org/blog/info>, zuletzt aufgerufen am 12.8.2008

vvvv Group: vvvv: a multipurpose toolkit, <http://vvvv.org/tiki-index.php>, zuletzt aufgerufen am 7. 7. 2008

Wands, Bruce: Art of the digital Age, London: Thames&Hudson 2006

Watz, Marius: Biografie, <http://www.medienkunstnetz.de/kuenstler/watz/biografie>, zuletzt aufgerufen am 10.8.2008

Watz, Marius: Generative Art Now. An Interview with Marius Watz, <http://www.artificial.dk/articles/watz.htm>, zuletzt aufgerufen am 10.8.2008

Watz, Marius: Generator.x: A timeline, <http://www.generatorx.no/generatorx-about-20>, zuletzt aufgerufen am 10.8.2008

Weibel, Peter: Die Algorithmische Revolution. Zur Geschichte der interaktiven Kunst, <http://www.zkm.de/algorithmische-revolution/index.php>, zuletzt aufgerufen am 11.8.2008

Yugo Nakamura, <http://www.designmuseum.org/design/yugo-nakamura>, zuletzt aufgerufen am 10.8.2008

## 8.2. Abbildungsverzeichnis

Abb. 1, Thanks: Kräftner, Thomas

Abb. 2, Broken Technology: Kräftner, Thomas

Abb. 3, Malstrom: Kräftner, Thomas

Abb. 4, Monochrome Supernova: Kräftner, Thomas

Abb. 5, AC Measure: Galanter, Philip: What is Generative Art? Complexity Theory as a Context for Art Theory, [http://www.philipgalanter.com/downloads/ga2003\\_paper.pdf](http://www.philipgalanter.com/downloads/ga2003_paper.pdf), zuletzt aufgerufen am 10.8.2008

Abb. 6, Zufall: Kräftner, Thomas

Abb. 7, Complexity: Galanter, Philip: What is Generative Art? Complexity Theory as a Context for Art Theory, [http://www.philipgalanter.com/downloads/ga2003\\_paper.pdf](http://www.philipgalanter.com/downloads/ga2003_paper.pdf), zuletzt aufgerufen am 10.8.2008

Abb. 8, Porpita porpita: Ocean Explorer, [http://oceanexplorer.noaa.gov/explorations/o2sab/logs/aug15/media/porpida\\_post.html](http://oceanexplorer.noaa.gov/explorations/o2sab/logs/aug15/media/porpida_post.html), zuletzt aufgerufen am 15.8.2008

Abb. 9, Mandel zoom 08 satellite antenna: Beyer, Wolfgang, [http://de.wikipedia.org/w/index.php?title=Bild:Mandel\\_zoom\\_08\\_satellite\\_antenna.jpg&filetimestamp=20061204214657](http://de.wikipedia.org/w/index.php?title=Bild:Mandel_zoom_08_satellite_antenna.jpg&filetimestamp=20061204214657), zuletzt aufgerufen am 15.9.2008

Abb. 10, Bonn Juridicum Vasarely: Weingartz, Hans, [http://de.wikipedia.org/wiki/Bild:Bonn\\_Juridicum\\_Vasarely.jpg](http://de.wikipedia.org/wiki/Bild:Bonn_Juridicum_Vasarely.jpg), zuletzt aufgerufen am 15.9.2008

Abb. 11, Eos xk (3): Talmor, Yair, [http://de.wikipedia.org/wiki/Bild:Eos\\_xk\\_\(3\),\\_1965.JPG](http://de.wikipedia.org/wiki/Bild:Eos_xk_(3),_1965.JPG), zuletzt aufgerufen am 15.9.2008

Abb. 12, Flash Player Penetration: Adobe, [http://www.adobe.com/products/player\\_census/flashplayer](http://www.adobe.com/products/player_census/flashplayer), zuletzt aufgerufen am 15.9.2008

Abb. 13, Central Darkness: Kräftner, Thomas

Abb. 14, Dynamic Chaos: Klingemann, Mario, <http://flickr.com/photos/quasimondo/2323834960>, zuletzt aufgerufen am 15.9.2008

Abb. 15, Pink Swirl: Kräftner, Thomas

Abb. 16, Beispielprojekt: Kräftner, Thomas

Abb. 17, Conclusio: Kräftner, Thomas

Abb. 18, Cover: Kräftner, Thomas

Abb. 19, Spine: Kräftner, Thomas

Abb. 18: Cover, mit dem Werkstück generiert (Cover.xml) ▶

Abb. 19: Spine, mit dem Werkstück generiert (Spine.xml) ▶



