

## Prozedurale Modelle

Virtuelle Welten können auf viele Arten erlebt werden. Eine frühe Variante, z.B. das Spiel *Rogue*,<sup>1</sup> zeigte dem Spieler lediglich einfache Zeichen-basierte Grafik (siehe Abb. 1). Mit dem Fortschritt der Computergrafik und deren Anwendung in Spielen und Filmen verbindet man heutzutage mit virtuellen Welten aber immer eine realistische Darstellung. Diese benötigt noch eine Menge händischer Arbeit und ist damit zeit- und kostenintensiv. Soll die Welt dann auch noch grenzenlos wirken, ist dies auf händische Art und Weise nicht mehr zu bewältigen.

Ein Ausweg sind sogenannte prozedurale Modelle. Interessanterweise sind solche generativen Verfahren schon lange im Einsatz; so sind die Spielwelten in dem eben erwähnten *Rogue* nach bestimmten Regeln und unter Verwendung von Zufallszahlen erstellt und so für den Spieler jedes Mal neu. Mit steigenden Möglichkeiten der Darstellung wird die Einsatz dieser Methoden aber komplexer und es müssen unterschiedliche Verfahren für den jeweiligen Einsatzzweck entwickelt werden.

Im folgenden werden wir die verschiedenen Arten beschreiben. Schließlich werden wir uns auch mit der Frage beschäftigen, wie sich die Verfahren kombinieren und effizient für virtuelle Welten nutzen lassen.



Abbildung 1: Szene des von *Rogue* inspirierten Spiels *NetHack*

1 Wikipedia: Rogue.

## 1 Definition

*Prozedurale Modelle* sind Modelle, die aus einer gegebenen Prozedur und einem kleinen (Initial-)Datensatz erzeugt werden.<sup>2</sup> Daraus ergeben sich direkt mehrere Merkmale:

- *Automatische Erzeugung mit keiner oder geringer manueller Arbeit.* Dies resultiert vor allem aus der Anforderung an einen kleinen Datensatz. Idealerweise sollten die Daten und deren Auswirkung auf das entstehende Modell für den Bearbeiter nachvollziehbar sein, so dass das Modell gezielt angepasst werden kann.
- *Wenig direkte Eingriffsmöglichkeiten für Änderungen.* Dies ist die Kehrseite der automatischen Erzeugung. Notwendigerweise werden Modelle erzeugt, die nur indirekt kontrolliert werden können. Je nach Einsatzzweck gibt es verschiedene Möglichkeiten, dieses Problem in den Griff zu bekommen. Diese werden später detaillierter beschrieben.
- *Detailreichtum, Komplexität.* Durch die gegebene Prozedur können fast beliebige Mengen an neuen Daten erzeugt werden und damit eine ebenso unbegrenzte Komplexität erzeugen.
- *Geringer Speicherplatz bzw. Ressourcenverbrauch zum Speichern des Modells.* Der kleine Datensatz erlaubt das Erzeugen des vollständigen Modells nur genau dann, wenn es wirklich gebraucht wird. Aus Geschwindigkeits-Gründen kann es dennoch notwendig sein, die prozedurale Erzeugung ganz oder in Teilen vorzuziehen und z.B. ein erzeugtes Modell im Speicher abzulegen, bzw. zu cachen.

Generell eignen sich prozedurale Modelle damit für viele Einsatzzwecke. Ideal sind sie für die Entwicklung von Prototypen. Hier kann der Eindruck durch die erhöhte Detailmenge schon in frühen Phasen besser eingeschätzt werden. Aufgrund geringer Spezifikationen sind sie aber auch einfacher anzupassen: die Stadt Köln im Mittelalter kann in die Stadt Köln heute verwandelt werden, in dem man die Regel der Generierung aus den Topographiedaten entsprechend anpasst.

Auch technisch können prozedurale Modelle neue Möglichkeiten eröffnen. So kann ein bestehender Datensatz, der z.B. manuell erstellt oder automatisch erfasst worden ist, entsprechend angereichert werden, um auch unter näherer Betrachtung noch gut auszusehen. Die Darstellung und Beleuchtung kann je nach verwendeten Verfahren sogar direkt benutzt werden, um eine sowohl

---

2 Alvy Ray Smith bezeichnete dies als *Database Amplification* in Smith: Plants, fractals, and formal languages

auf Qualität als auch auf Geschwindigkeit optimierte Darstellung zu erreichen. Beispiele sind automatisierte, gut kontrollierbare Level-of-Detail Mechanismen.

## 1.1 Klassifizierung

Prozedurale Verfahren lassen sich in nach der Art der verwendeten Prozedur einteilen. Als *direktes Verfahren* kann man die Verwendung einer speziellen Prozedur für einen Anwendungsfall bezeichnen. Dies wird z.B. häufig bei der Erzeugung von Pflanzen und Bäumen benutzt: In der Arbeit von Weber und Penn<sup>3</sup> wird ein Baum in einem rekursiven Verfahren erzeugt, wobei für jede Stufe eine eigene Funktion Geometrie erzeugt. Die Grundstruktur ist somit fest vorgegeben und kann nur über eine Vielzahl spezifischer Parameter gesteuert werden. Dennoch lassen sehr verschiedene Art von Bäumen damit erzeugen. Die Anzahl der Parameter ist zwar groß, dafür ist ihre Bedeutung klar erkennbar. Konkrete Änderungen an einem Baum lassen sich so gut durchführen.

*Grammatik-basierte Verfahren* arbeiten mit einer abstrakten Definition der zu erstellenden Geometrie. Es müssen Ersetzungsregeln definiert werden, die ein Symbol durch eine Menge anderer Symbole ersetzt. Werden eine Menge an Regeln angewandt, entstehen eine Folge an Symbolen, die dann als Beschreibung einer Geometrie interpretiert werden kann. Dies eignet sich sehr gut für Modelle, die klar strukturiert sind, wie z.B. Pflanzen<sup>4</sup> oder Gebäude<sup>5</sup>.

Bei der *additiven Synthese* werden Basisfunktionen moduliert und summiert um eine neue Funktion zu erzeugen. Dies wird vor allem für die Erzeugung von Oberflächen und Materialien, die stark strukturiert sind, wie z.B. Marmor und Holz, benutzt. Im Abschnitt 2 werden wir näher darauf eingehen.

Zur Erzeugung von Animationen von vielen einzelnen Objekten werden *regelbasierte Verfahren* eingesetzt. Ein einzelnes Objekt erhält einen Satz Regeln, der beschreibt, wie es sich aufgrund seiner *Sinnesindrücke* verhalten soll. Durch eine Simulation kann dann für eine Vielzahl von Objekten ein stimmiges Gesamtergebnis berechnet werden.

## 2 Umsetzung und Anwendbarkeit

Als Beispiel, wie die oben skizzierten Verfahren umgesetzt und angewendet werden können, gehen wir im Folgenden näher auf die additive Synthese ein.

---

3 Weber/Penn: Creation and rendering of realistic trees.

4 Prusinkiewicz/Lindenmayer: The algorithmic beauty of plants.

5 Müller et al.: Procedural Modeling of Buildings

## 2.1 Basisfunktionen und Synthese

Zur Synthese wird eine kleine Menge an Basisfunktionen benötigt. Diese unterscheidet sich nach Art der Anwendung. Soll z.B. ein Helligkeitswert für jeden Punkt im Raum ermittelt werden, z.B. um einen massiven Körper darzustellen, wird eine Funktion der folgenden Art benötigt:

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}$$

Diese Funktion wird zur Synthese nun im Wertebereich skaliert, d.h. gestreckt oder gestaucht, und gewichtet addiert:

$$s(x) = \sum_{k=0}^n \alpha_k f(\beta_k x),$$

mit  $\alpha_k$  als Gewichtungsfaktor und  $\beta_k$  als Frequenz der  $k$  Funktionen.

Basisfunktionen müssen mehrere Eigenschaften erfüllen um sinnvoll einsetzbar zu sein:

- Eine Basisfunktion muss wiederholbar sein, d.h. gleiche Eingabewerte müssen die gleichen Ausgabewerte erzeugen. Dies ist notwendig, um beim Berechnen mehrerer Bilder, wie dies u.a. bei der Echtzeit-Computergrafik der Fall ist, immer das gleiche Modell darstellen zu können.
- Die Funktion muss *Bandbreiten-limitiert* sein, idealerweise normiert, d.h. mit einer Maximalfrequenz von 1. Die Notwendigkeit dieser Eigenschaft für die Darstellung wird in Abschnitt 2.2 näher erklärt, hier sei nur darauf hingewiesen, dass durch die Begrenzung der Maximalfrequenz eine Skalierung im Definitionsbereich erst möglich ist und dies für das Formen von Mustern eine wichtige Voraussetzung ist.
- Analog zur Maximalfrequenz muss auch der Wertebereich beschränkt sein, idealerweise wieder in einem normierten Bereich, etwa von  $[0, 1]$  oder  $[-1, 1]$ .

Diese Eigenschaften werden von vielen Funktionen erfüllt, wie z.B. der Sinus-Funktion. In der Tat lassen sich sämtliche andere Funktionen durch Summation von gewichteten Sinus-Basisfunktion verschiedener Frequenzen darstellen.

Interessanter sind aber Basisfunktionen, die mit Hilfe von (Pseudo-)Zufallszahlen funktionieren, häufig auch Noise<sup>6</sup> genannt werden. Die Noise-Funktion

---

6 engl. für Rauschen

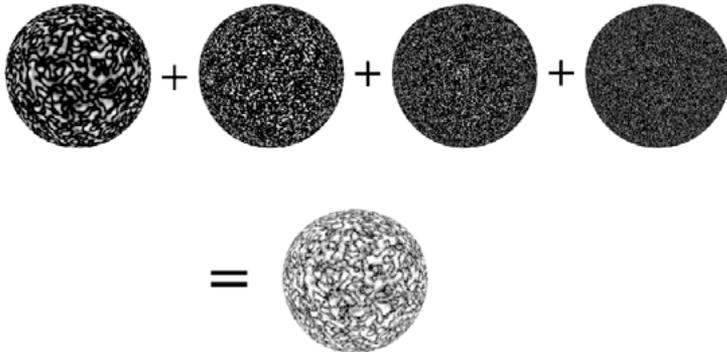


Abbildung 2: Mehrere Noise-Funktionen mit unterschiedlicher Maximalfrequenz ergeben addiert eine granitartige Struktur

von Perlin<sup>7</sup> ist dabei einer der bekanntesten Vertreter und ist darüber hinaus relativ schnell zu berechnen<sup>8</sup>. In Abb. 2 sieht man ein einfaches Beispiel dazu.

## 2.2 Adaptive Komplexität

Zur Generierung von Bildern in der Computergrafik werden heutzutage eigentlich immer Rasterisierungsverfahren eingesetzt, die die Farbwerte einzelner weniger Lichtstrahlen ermitteln. Bei diesen Verfahren ist es notwendig, dass Modelle nicht zu viele Details enthalten, weil ansonsten sogenannte *Aliasing*-Artefakte auftreten. Dies äußert sich z.B. bei Animationen in einem flackernden Rauschen. Je besser ein Modell diese Details adaptiv zur Verfügung stellen kann, desto weniger muss auf aufwendige und rechenintensive Gegenmassnahmen, sogenanntes *Anti-Aliasing*, zurückgegriffen werden.

Die additive Synthese bietet hierfür eine sehr einfache und effektive Lösung. Die einzelnen Funktionen haben als Voraussetzung eine Maximalfrequenz, d.h. eine maximale Menge an Details. Bei der Berechnung des Farbwertes kann nun eine je nach Blickpunkt auf das Modell maximal darstellbare Frequenz berechnet werden und alle Funktionen bei der Synthese mit einer höheren Frequenz nicht verwenden. Normalerweise wird dies nicht mit einem harten Schwellwert sondern einem Übergangsbereich realisiert, in dem das Gewicht der Funktion schrittweise reduziert wird.

<sup>7</sup> Perlin: An Image Synthesizer.

<sup>8</sup> Eine Verbesserung des Verfahrens wurde von Perlin selber in Perlin: Improving Noise beschrieben. Andere Verfahren finden sich z.B. in Ebert et al.: Texturing and Modeling

Diese einfache Massnahme erhöht nicht nur die visuelle Qualität, sie ist auch ein einfaches *Level-of-Detail*-Verfahren, welches immer weniger Details berechnet und damit weniger Rechenzeit verbraucht, je weiter sich der Betrachter von dem Modell entfernt.

### 2.3 Reduzierung von Laufzeitkosten

Die adaptive Synthese aus dem vorherigen Abschnitt sorgt dafür, dass der Rechenaufwand an die darzustellende Komplexität gekoppelt ist. Die Laufzeit kann aber auch beliebig gross werden, wenn entsprechend viele Details dargestellt werden sollen, z.B. weil der Betrachter sehr nahe am Modell steht.

Hinzu kommt, dass die Berechnung der Basisfunktionen an sich auch höhere Laufzeitkosten verursachen, als z.B. ein einfacher Zugriff auf eine Textur<sup>9</sup> und die Daten immer wieder neu berechnet werden: So verändern sich Modelle oft nur allmählich über die Zeit. Eine Wand aus Marmor hat z.B. immer das gleiche Muster und auch der Blickwinkel oder Abstand wird sich nicht viel verändern, wenn der Betrachter längere Zeit vor einer Skulptur steht und die Wand nur im Hintergrund sichtbar ist.

Die einfachste Reduzierung der Rechenleistung ist auf Kosten des Speicherbedarfs der Anwendung möglich und besteht darin, das Modell komplett vorzuberechnen. Dies wird *baking*<sup>10</sup> genannt. Bei einem prozeduralen Oberflächenmodell wird ein passendes Bild berechnet und als Textur auf die Geometrie aufgebracht.

Durch die festgelegte Auflösung des Bildes geht einer der Vorteile der prozeduralen Modelle verloren: Der Betrachter sieht ab einem Punkt keine weiteren Details mehr. Gemildert werden kann dies durch *Detail*-Texturen. Hierbei werden ähnlich der Basisfunktionen mehrere Texturen summiert, wobei die Detail-Textur erst von nahen sichtbar ist und kachelbar ist, d.h. nahtlos aneinander gelegt werden kann.

Sollen *echte* Details sichtbar sein, muss eine entsprechend grosse Textur verwendet werden. Sogenannte *Clipmaps*<sup>11</sup> sind ein Verfahren, das es erlaubt grosse Texturen zu verwenden, die nur in Teilen im Speicher der Grafikkarte vorliegen müssen, um dargestellt werden zu können. Bei geringen Veränderungen des Blickpunktes ergeben sich hierbei auch nur geringe Änderungen an der Untermenge der Bilddaten, so dass bei kontinuierlichen Bewegungen der zusätzliche Rechenaufwand minimal ist.

9 Insgesamt steigt die Rechenleistung schneller als die zur Verfügung stehende Speicherbandbreite. Von daher reduzieren sich die realen Kosten der Berechnung immer mehr im Vergleich zum *einfachen* Texturzugriff.

10 engl. für backen

11 Tanner/Migdal/Jones: The clipmap: a virtual mipmap.

Anstelle die Daten aus dem Hauptspeicher bzw. von Festplatte nachzuladen, können die Daten aber auch bei Bedarf erst erzeugt werden. Ein solches hybrides Verfahren vereint die Vorteile der prozeduralen Verfahren, in dem (fast) beliebig viele Details sichtbar sind, zugleich aber Daten nicht für jedes Bild erneut berechnet werden müssen. Auch wird lediglich ein Ausschnitt der Daten benötigt, d.h. der zusätzliche Speicherbedarf ist moderat.

Die bisher beschriebenen Verfahren lassen sich nicht nur auf Farbwerte anwenden. Speziell Clipmaps können z.B. auch gut benutzt werden, um aus einem Bild, welches Höhenwerte anstelle von Farbwerten speichert, eine Höhenkarte zur Laufzeit zu erstellen.<sup>12</sup>

## 2.4 Komposition

Die oben angesprochenen Detail-Texturen sind ein Beispiel für die allgemeinere Technik der Komposition. Ähnlich der Synthese von Basisfunktionen können verschiedene Verfahren kombiniert werden, um so ein optimiertes Gesamtergebnis zu erhalten. Dies ist vor allem deswegen interessant, weil auf diese Art und Weise auch ein Künstler in den Prozess einwirken kann und so ein für ein bestimmten Anwendungszweck optimiertes Ergebnis entstehen kann.

Auch wenn der Art der Kombination eigentlich keine Grenzen gesetzt sind, werden wir uns im folgenden auf eine Untermenge beschränken, in denen sich die zu kombinierenden Funktionen als eine Abbildung eines Punktes im Raum auf einen Farbwert ansehen lassen. Den Farbwert kann als Zahlentupel definiert werden, welcher – wie in der Computergrafik üblich – die Mischanteile von Rot, Grün und Blau angibt. Zusätzlich wird ein sogenannter *Alpha-Wert*<sup>13</sup> definiert, der wie die Farbwerte in einem normierten Wertebereich von  $[0, 1]$  liegt und den Anteil der Verdeckung angibt.

Die Eingabefunktionen können sowohl prozedural erzeugt sein, als auch z.B. in Form von Bild- bzw. Texturdaten von einem Künstler eingebracht werden. In Abb. 3 ist dafür ein Beispiel zu sehen.

Es gibt verschiedene Operatoren, die diese Eingabefunktionen zur einer neuen verknüpfen:

- Stellen die Eingaben mehrere Ebenen eines Materials dar, können sie mittels des sogenannten *Over-Operators* visuell realistisch übereinander gelegt werden.

12 Siehe Losasso/Hoppe: Geometry clipmaps: Terrain rendering using nested regular grids und Asirvatham/Hoppe: Terrain rendering using GPU-based geometry clipmaps. Auch hierbei wird eine (prozedural) erzeugte Detail-Textur benutzt werden, um den Realismus zu erhöhen.

13 Eingeführt in Porter/Duff: Compositing digital images. In dieser Arbeit ist auch der oben beschriebenen *Over-Operatoren* der *Compositing Algebra* definiert.



Abbildung 3: Das linke Bild zeigt eine texturierte Teekanne. In dem rechten Bild wurde eine Schachbrettmuster auf die Teekanne angewendet, wobei die weißen Flächen durch die Textur des linken Bildes ersetzt wurde.

- Ein kontinuierlicher Wechsel zwischen mehreren Materialien kann durch eine weitere (Selektions-)Funktion beschrieben werden. So kann z.B. eine abstrakte, manuell erstellte Karte angeben, wo Grass, Sand und Wasser sind und an entsprechenden die jeweilige Eingabefunktionen ausgewählt werden.
- Als Hilfsoperatoren ist es nützlich Funktionen skalieren und mit logischen Operatoren verknüpfen zu können, z.B. um die Selektionsfunktion aus anderen Eingaben herzuleiten.

### 3 Ganzheitliche Prozesse

Der letzte Abschnitt zeigte bereits, dass die Kombination von mehreren Verfahren eine Vereinfachung der Handhabbarkeit, sowie einer weitere Steigerung der visuellen Komplexität und Qualität mit sich bringt.

Von daher erscheint es sinnvoll, die Frage nach einem ganzheitlichen Prozess zu stellen, der prozedurale Verfahren sowie manuell erstellte Modelle aller Art erfasst und direkt kombinierbar macht. Er sollte folgende Eigenschaften besitzen:

- Modelle sollen kombinierbar sein, wobei der Prozess die Anordnung der Modelle zueinander unterstützen sollte. Hierbei sind Informationen über Einschränkungen und Hilfestellungen notwendig, die ein Modell als Metadaten enthalten muss.
- Da eine globale Sicht aller Daten und Verknüpfungen vorliegt, kann eine Realisierung des Gesamtmodells an einen speziellen Kontext angepasst werden. So kann auf Geräten mit geringer Leistung, wie z.B. auf einem Mobiltelefon, das Modell automatisch in geringere Komplexität erstellt werden. Dies schließt sowohl die Reduzierung einzelner Komponenten, aber auch das Weglassen oder Ersetzen von Komponenten ein.

- Die Hilfestellung bei der Erstellung kann weiter fortgeführt werden, indem der Prozess nicht nur bei der Kombination, sondern auch schon bei der Auswahl der einzelnen Modelle hilft, z.B. in dem in entsprechenden Bibliotheken gesucht werden kann.
- Die vorhandenen Strukturen können genutzt werden, um eine semantischen Informationen abzuleiten, die dann von der Anwendung verwendet werden können.

Solch ein Prozess hat für alle beteiligten Personen Vorteile:

- Die Autoren von Modellen profitieren von einem breiteren Publikum, da die Modelle besser wiederverwendbar sind.
- Anwendungsentwickler können sich mehr auf den Inhalt und weniger auf die notwendige Technologie konzentrieren. Die erlaubt vor allem die Einbindung von Personen mit geringen Technik-Know-how in den Entwicklungsprozess.
- Die Benutzer profitieren von besseren und vielfältigeren virtuellen Welten.

Ein Beispiel kann dieses Vorgehen verdeutlichen: Ein Spielautor möchte eine Szene mit einer Hütte im Wald in einer typischen hügeligen mitteleuropäischen Landschaft erstellen. Dazu modelliert er spielrelevante Aspekte in der Hütte mit traditionellen Mitteln. Zudem gibt er in seinen Basisdaten die grundlegenden Daten in maschinenlesbarer Form an: Hügelige Landschaft, geographische Lage, die Hütte, ein Weg.

Der Steuerungsprozess der Geometrieerzeugung verfügt über ein Portfolio von Generatoren und über einen Regelsatz, der allgemeines, szenenunabhängiges Wissen über räumliche Eigenschaften der Welt enthält. Aufgrund der Basisdaten wählt der Prozess einen Hügelgenerator, der die Grundfläche der Szene erstellt und mit Metadaten über die Grundbeschaffenheit versieht, die den nachfolgenden Schritten helfen.

Das Grundwissen über allgemeine Beschaffenheit von Welten enthält beispielsweise Wissen darüber, dass auf Wegen und in der unmittelbaren Umgebung von Gebäuden normalerweise keine Bäume stehen. So kann er die entsprechenden Bereiche für Vegetation sperren.

Der Steuerprozess wählt im Folgenden einen Vegetationsgenerator, der Pflanzen und weitere Objekte wie z.B. Steine oder Flüsse in der Szene verteilt. Die Metadaten des Landschaftsgenerators helfen nun, geeignete Pflanzentypen aus einem Repertoire zu finden. Die Pflanzen selbst werden wiederum von spezialisierten Generatoren erzeugt.

Auf diese Weise können Umgebungen, weitläufige Landschaften und ähnliches automatisch aber kontrolliert erzeugt werden. Trotzdem bleiben die

Basisdaten übersichtlich und verständlich. Der Autor kann seine Angaben nach ersten Durchläufen verfeinern, um das gewünschte Ergebnis zu erhalten.

Dieser Prozess schafft bereits für eine einzige Szene erhebliche Arbeitserleichterungen. Aber darüber hinaus können etliche Szenenvariationen geschaffen werden, beispielsweise um spezifische Eigenheiten und Beschränkungen verschiedener Geräteklassen zu berücksichtigen. Der Regelsatz kann diese Eigenheiten besser berücksichtigen, wenn der Autor Prioritäten in seinen Grunddaten angibt. So kann z.B. verhindert werden, dass der Generator bei einer Szene für beschränkte Mobilgeräte wichtige Aspekte der Szene ausspart.

## 4 Fazit

Prozedurale Modelle werden heutzutage bereits vielseitig eingesetzt. Die steigende Grafikkartenleistung wird ihren Einsatz auch im Echtzeit-Bereich und somit für virtuelle Welten weiter voranbringen. Dennoch wird es immer notwendig sein, dass ein Künstler Anpassungen vornehmen kann, und in einer geeigneten Kombination aller Verfahren lässt sich dies, auch schrittweise, realisieren.

Der skizzierte ganzheitliche Prozess kann hierbei helfen, zuerst muss er aber konkretisiert und auf seine Zielsetzung hin überprüft werden. Der Prozess kann dann in der Rolle einer Basistechnologie für virtuelle Welten das Erstellen und Anpassen dieser deutlich vereinfachen und somit dazu beitragen, dass virtuelle Welten selber als Basistechnologie angesehen werden können, um Wissen und Erfahrungen zu vermitteln.

## Literatur

Asirvatham, Arul/Hoppe, Hugues: Terrain rendering using GPU-based geometry clipmaps, in: GPU Gems 2, Reading, CA 2005, S. 27–45.

Ebert, David S. et al.: Texturing and Modeling: A Procedural Approach, San Francisco, CA 2002.

Losasso, Frank/Hoppe, Hugues: Geometry clipmaps: Terrain rendering using nested regular grids, in: SIGGRAPH 2004, New York, NY 2004, S. 769–776.

Müller, Pascal et al.: Procedural Modeling of Buildings, in: SIGGRAPH 2006, New York, NY 2006, S. 614–623.

Perlin, Ken: An Image Synthesizer, in: SIGGRAPH 85, New York, NY 1985, S. 287–296.

Perlin, Ken: Improving Noise, in: SIGGRAPH 2002, New York, NY 2002, S. 681–682.

Porter, Thomas/Duff, Tom: Compositing digital images, in: SIGGRAPH '84, New York, NY 1984, S. 253–259.

Prusinkiewicz, Przemyslaw/Lindenmayer, Aristid: The algorithmic beauty of plants, New York, NY 1991/1996.

Smith, Alvy Ray: Plants, fractals, and formal languages, in: SIGGRAPH '84, New York, NY 1984, S. 1–10.

Tanner, Christopher C./Migdal, Christopher J./Jones, Michael T.: The clipmap: a virtual mipmap, in: SIGGRAPH 98, New York, NY 1998, S. 151–158.

Weber, Jason/Penn, Joseph: Creation and rendering of realistic trees, in: SIGGRAPH 95, New York, NY 1995, S. 119–128.

Wikipedia: Rogue, [URL:http://de.wikipedia.org/wiki/Rogue\\_](http://de.wikipedia.org/wiki/Rogue_(Computerspiel)) (Computerspiel) – Zugriff am 06.12.2008.