Generierung virtueller Geländeformen durch Rauschfunktionen und Filter

Thomas Jourdan

`http://innerworld.sourceforge.net`

9. August 2008

Inhaltsverzeichnis

Zusammenfassung

Synthetisch erzeugtenen Geländeformen begegnet man in Computerspielen und Filmen. Der ersten Teil des Textes enthält eine Einführung in die mathematischen Grundlagen dieser Artefakte. Im zweiten Teil wird gezeigt wie diese mathematischen Grundlagen mit dem Blender-Plugin Innerworld praktisch angewendet werden können.

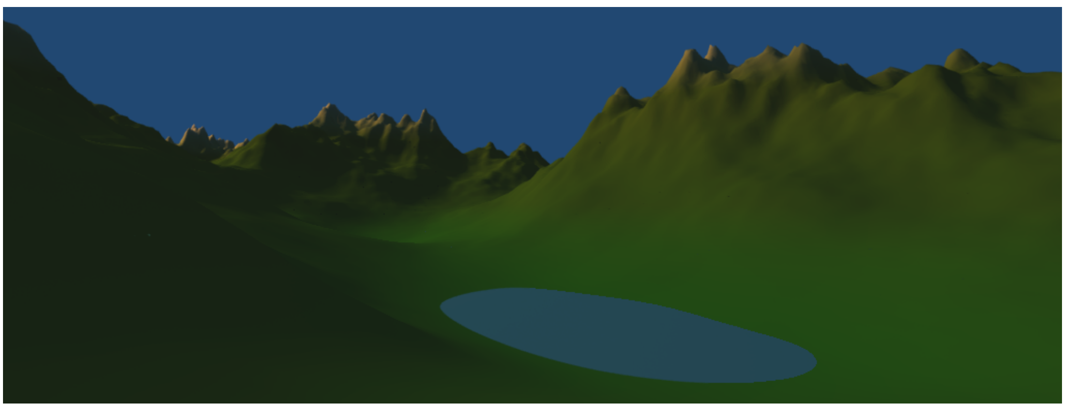

Abbildung 1:

1 Übersicht

Jeder hat es schon mal gesehen: Meer, Steilküste, dramatische Wolkenformation und Sonnenuntergang, besser noch Mondaufgang. Nicht in fernen Ländern, sondern auf seinem Computerbildschirm. Alles in Farbe, vielleicht etwas zu dramatisch. Auf jeden Fall aber künstlich.
Um solche virtuellen Welten zu erzeugen werden zwei Dinge benötigt. Als Erstes ein Algorithmus mit dem 3D Modelle fotorealistisch visualisiert werden können. Bei diesen Algorithmen spricht man je nach dem zugrunde liegenden optischen Model von Raytracing oder Radiosity. Man kann sich einen solchen Algorithmus wie einen Compiler vorstellen, der als Input eine Szenenbeschreibung entgegennimmt um daraus das Bitmap - Bild für den Betrachter zu berechnen. Aus den Geometriedaten, den Farbtexturen, den Lichtquellen und der Simulation atmosphärischer Effekte compiliert die Rendering Pipeline ein mehr oder weniger realistisches Bild.
Solch ein Visualisierungstool benötigt Modelldaten, welche die Szene beschreiben. In der Regel werden die Objekte in der Szene wie in einem CAD-System konstruiert oder als Geodaten importiert. Um Gegenstände wie Häuser, Maschinen oder Möbel zu modellieren, ist die Verwendung eines CAD ähnlichen Systems optimal. Anders sieht es in Outdoor Szenarien aus. Hier wird eine spezielle Form der Modelldaten, die Höhenprofile der Geländeform benötigt. Im folgenden Text soll nun die mathematische Erzeugung von Höhenprofilen durch Filtern von Rauschquellen dargestellt werden.
Höhenprofile sind zweidimensionale Felder. Jedes Element des Feldes beschreibt die Höhe des Terrains an diesem Ort. Man kann diese Daten auch als Graustufenbild darstellen. In solchen Bildern steht die Helligkeit eines Punktes für die Höhen über der Basisebene. Der einfache Zusammenhang zwischen Graustufenbild und Höhenprofil kann benutzt werden um diese Profile selbst zu „zeichnen“. Höhenprofile können erstellt werden in dem man mit einem Zeichenprogramm Graustufenbilder manuell mit einem Pinselwerkzeug, das keine scharfen Kanten hinterlässt, zeichnet. Dies hat den Vorteil, dass der grobe Aufbau des Terrains genau nach den eigenen Vorstellungen gestaltet werden kann. Jedoch wäre es sehr mühevoll die vielen kleinen Höhenunterschiede, welche für ein natürlich wirkendes Terrain gebraucht werden, alle händisch in das Graustufenbild zu zeichnen. Im Folgenden spielt dieser manuelle Ansatz nur eine untergeordnete Rolle. Vielmehr werden die Höhenmodelle, einschließlich ihrer Details, automatisch generiert. Die Graustufendarstellung dient lediglich zur Veranschaulichung.
Heutige 3D Modellierungswerkzeuge verbinden unter einer Benutzeroberfläche sowohl die Modellierung der 3D-Geometrie, die Farbgebung der Materialien, als auch die Berechnung des fotorealistischen Bildes. Die Beispiele in diesem Text beziehen sich auf Blender `http://blender.org/` ein Open Source 3D Modellierungswerkzeug. Blender wurde um ein Plugin `http://innerworld.sourceforge.net/` zur Modellierung von Höhenprofilen durch Rauschgeneratoren und Filterketten erweitert. Blender selbst ist ein Profiwerkzeug. Das Plugin hat jedoch nur einen Proof of Concept Status erreicht. Blender muss mit großen Datenmengen zurechtkommen und verfügt deshalb für das berechnen der Bilder, über gut optimierte Algorithmen in C. Im Plugin für die Höhenprofile sind die Algorithmen lediglich, in einer nicht optimierten Form, als Python Skripte implementiert.

Digitale Signalverarbeitung und Terraingenerierung

In der digitalen Signalverarbeitung (DSV) werden die zu verarbeitenden Signale durch Zahlenfolgen repräsentiert.
Digitale Terraingenerierung:

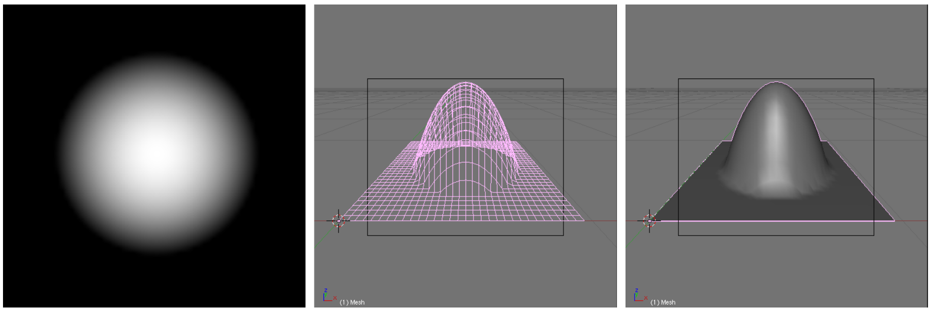

Abbildung 2: a) Mit einem Gradientenwerkzeug wurde ein radialer Gradient erstellt und als Graustufenbild gespeichert. b) Dieses Graustufenbild in einem 3D-Modellierungstool als Wireframe-Modell. Die hellsten Bereiche des Ausgangsbildes erzeugen die Spitze des Ellipsoids. c) Das Wireframe–Model mit 'Haut'. In dieser Darstellung wurde bewusst auf die Darstellung der Materialeigenschaften des Ellipsoids, zum Beispiel seiner Oberflächenfarbe, verzichtet.

- Das Signal wird als Höhe des Terrains über einem Punkt der X/Y Ebene interpretiert.
- Das Signal durchläuft eine Folge von Knoten / Filteroperationen.
- Vollständige Systeme werden aus wenigen elementaren Filteroperationen aufgebaut.
- Die Filterstrukturen können keine Rückkopplungen enthalten.
- Bevorzugt werden stochastische Signalquelle eingesetzt.
- Die fraktale Dimension ist der wichtigste Kennwert einer stochastische Signalquelle.
- Die physikalische Größe Zeit spielt keine Rolle bei der Beschreibung des Systemverhaltens.

Datenfluss im Softwaresystem

Der Datenfluss in einem Grafiksystem wird als Rendering Pipeline bezeichnet. In Anlehnung an die digitale Signalverarbeitung geht man von der Vorstellung aus, dass Informationen aus einer Quelle über mehrere Filter in eine Senke fließen. Das künstliche Bild, welches mit Hilfe von 3D Algorithmen berechnet wird, stellt die Senke dar. Als Besonderheit sind in diesem Ansatz die Signalquellen Zufallsprozesse.
Bei den Berechnungsverfahren werden implizite und explizite Prozeduren unterschieden. Unter einer impliziten Prozedur versteht man eine Funktion $f(x, y, z)$, die für jeden Aufruf ihren Funktionswert jeweils unabhängig von anderen Aufrufen berechnen kann. Aus mathematischer Sicht ist das eine Selbstverständlichkeit. In der Softwareentwicklung ist der Funktionsbegriff jedoch schwammiger gefasst. Implizite Prozeduren werden deshalb bevorzugt, weil sie 'just in time' die benötigten Daten berechnen. Sie greifen nicht auf bereits berechnete Datensätze zurück. Dieses Vorgehen spart Ressourcen, da in großen Szenarien nicht alle Objekte in derselben Detaillierung sichtbar sind. Explizite Prozeduren, die auf im Voraus berechneten Datensätzen arbeiten, können nicht vollständig vermieden werden.

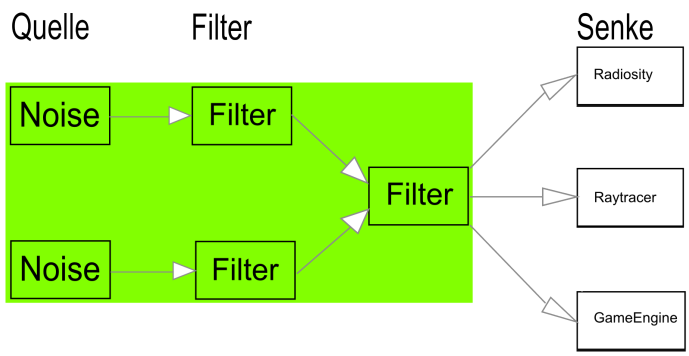

Abbildung 3: Zufallsprozesse erzeugen Daten, die nach einer Filterung, als Modelldaten für Raytracer oder Game Engines verwendet werden können.

2 Mathematische Grundlagen

Die mathematische Erzeugung von Höhenprofilen ist stark von der fraktalen Geometrie beeinflusst. Es geht bei dieser Art der Datenerzeugung nicht darum, die physikalischen Gesetzmäßigkeiten der Auffaltung und Erosion von Gebirgen oder das Entstehen einer Küstenlinie in physikalische Modelle zu fassen. Um den Rechenaufwand abzukürzen werden Methoden angewendet, die Daten generieren, welche dieselben statistischen Eigenschaften aufweisen wie Daten die in einer natürlichen Umgebung gemessen werden könnten. Der wichtigste Kennwert um die Glaubwürdigkeit des virtuellen Artefakts zu untermauern ist die fraktale Dimension. Dieser Kennwert soll von der gewohnten Rauigkeit des natürlichen Objektes möglichst wenig abweichen. Bei Fraktalen, im mathematischen Sinn, ist die Anzahl der sichtbaren Details vom Vergrößerungsfaktor unabhängig. Dadurch werden Fraktale, im Gegensatz zu euklidschen Formen, beim Vergrößern nicht glatt. Diesen Effekt nennt man Skaleninvarianz. Bei Terraingeneratoren spielt die statistische Selbstähnlichkeit der Fraktale eine untergeordnete Rolle. Terraingeneratoren sind streng mathematisch gesehen keine Fraktale, denn die Selbstähnlichkeit tritt nicht über mehrere Größenordnung auf.

Hausdorff Dimension Gewöhnlich entspricht die Dimension der Anzahl der Parameter, die benötigt wird um die Lage eines Punktes, mithilfe eines Koordinatensystems zu beschreiben. Zur Unterscheidung von neueren Dimensionsbegriffen, spricht man von der euklidschen oder topologischen Dimension. Die Lage eines Punktes im euklidschen Raum wird durch drei unabhängige Parameter, zum Beispiel die kartesischen Koordinaten x, y und z, beschreiben. Für sehr raue Geometrien hat sich dieser, auf ganzzahligen Dimensionen beruhende, Ansatz als nicht ausreichend erweisen. Fraktale Gebilde werden deshalb mit der nicht ganzzahligen Hausdorff Dimensionen beschrieben. Wobei die Hausdorff Dimension um Bruchteile höher ist als die topologische Dimension. Topologisch ist das Terrain eine zweidimensionale Ebene. Um einen natürlichen Eindruck zu vermitteln, ist die Ebene irregulär verformt. Diese Eigenschaft schlägt sich im gebrochenen Anteil der Dimension wider.

Definition der Hausdorff Dimension für Fraktale:

$$D = -\lim_{R \to 0} \frac{\log(N)}{\log(R)}$$

N ist die Mindestanzahl an Kugeln vom Radius R, die nötig sind um das zu messende Objekt zu überdecken.

Hurst Exponent In den konkreten Algorithmen wird jedoch anstelle der Hausdorff Dimension der Hurst Exponent verwendet.

Definition des Hurst Exponenten:

$$H = \frac{\log(\frac{R}{S})}{\log(T)}$$

Dabei ist T die Zeitdauer der Datenaufzeichnung und R/S der zugehörige Wert aus dem skalierten Bereich.

Interpretation des Hurst Exponent Der Hurst Exponent einer Zahlenfolge gibt an, ob die Folge rein zufällig ist oder ob ihr ein Trend zugrunde liegt.

- Ein Hurst Exponent von $h = 0.5$ deutet auf einen unabhängigen Zufallsprozess hin. Random Walk werden diese völlig zufälligen gaußverteilten Zahlenfolgen genannt.
- Ein Hurst Exponent im Bereich $0.5 < H < 1.0$, deutet auf einen Zufallsprozess mit Gedächtnis hin. Zahlenreihen, die in diesem Bereich liegen, bezeichnet man als fraktale brownsche Bewegung. Die gebräuchliche Abkürzung lautet: `fBM`.

Für die Terrain Generatoren lautet der Zusammenhang zwischen der fraktalen Dimension D, der euklidschen Dimension E und dem Hurst Exponenten H:

$$D = (E + 1) - H$$

$$D = 3 - H$$

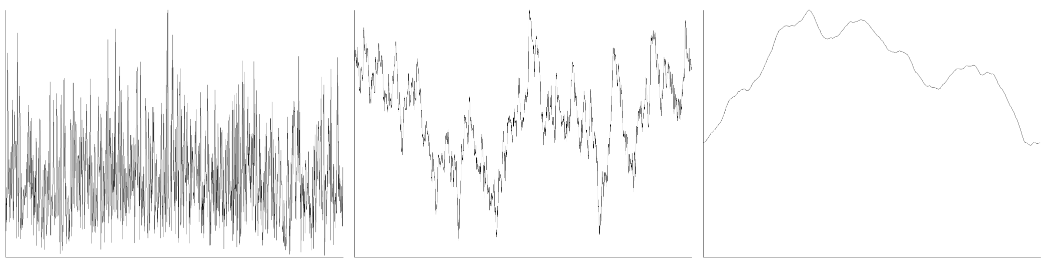

Abbildung 4: Rauschfunktionen liefern simple synthetische Horizontlinien. Von links nach rechts: $1/f^0$, $1/f^1$ und $1/f^2$. In den weiter unter verwendeten Rauschgeneratoren erscheint die Horizontlinie in Abhängig vom Hurst Exponent mehr oder weniger irregulär. Quelle: Wikipedia

Rauschen In der Physik werden unerwartete Schwankungen eines Messwertes über der Zeit als Rauschen bezeichnet. Diese Vorstellung lässt sich vom Zeitbereich auf den Raum übertragen. Bei einem künstlich erzeugten zufälligen Terrain spricht man deshalb von Rauschen. Im Gegensatz zur Messtechnik ist man bei der Terraingenerierung nicht daran interessiert Nutzsignal und Rauschen zu trennen. Vielmehr gibt es nur künstliches, näherungsweise erzeugtes Rauschen. Nicht der Messwert soll rekonstruiert werden, sondern der Zufallsprozess selbst wird zum Nutzsignal.

$1/f^0$ **Rauschen** $1/f$ Rauschen kann durch einen völlig unabhängigen gaussverteilten Zufallszahlengenerator angenähert werden. Seine spektrale Leistungsdichte ist eine Gerade parallel zur Frequenzachse. Damit wird ausgedrückt, dass alle Frequenzen mit gleichem Betrag vertreten sind. Korrekt müsste man diesen Typ als $1/f^0$ Rauschen bezeichnen. Um weißes Rauschen auf einem digitalen System zu erzeugen müsste dieses unendlich lange rechen, da beliebig hohe Frequenzen vertreten sind. Dies ist weder möglich noch erwünscht, denn reale Terrains sind nicht beliebig stark zerklüftet.

Brownsche Bewegung Das am stärksten korrelierte Rauschen, das in der Terraingenerierung eingesetzt werden kann, ist die Brownsche Bewegung. Sie entsteht durch Integration von weißem Rauschen. Ihr Name stammt aus der Brownschen Molekularbewegung. Ein Tröpfchen wird in einer Emulsion von den benachbarten Molekülen durch deren thermisches Rauschen angestoßen. Dabei betrachtet man die thermischen Bewegungen der Moleküle als weißes Rauschen. Der Weg des Tröpfchens kann als das Integral der Rauschfunktion verstanden werden. Korrekt wird dieser Zufallsprozess als $1/f^2$ Rauschen bezeichnet. Dieses Rauschen besitzt deutlich mehr niedere Frequenzen als hohe Frequenzen. Für die Verteilung der Höhen in einem Terrain passt dieser Rauschtyp gut. Für die Texturierung einer Oberfläche ist $1/f^2$ Rauschen jedoch zu stark abgedämpft.

$1/f$ **Rauschen** Ein großer Teil, des in der Natur beobachteten Rauschens hat ein $1/f$ Leistungsspektrum. Vom Rauschen eines Wasserfall, über Feldeffekttransistoren bis zum Verlauf von Börsenkursen ist $1/f$ Rauschen nachweisbar. Allerdings gibt es keine ursächliche Begründung dafür, warum dieser Rauschtyp so weit verbreitet ist. Wenn er jedoch in der Natur oft anzutreffen ist, kann er auch in virtuellen Umgebungen verwendet werden, um einen natürlichen Eindruck vorzugaukeln. $1/f$ Rauschen kann ziemlich direkt aus gleich verteilten Zufallszahlengeneratoren angenähert werden. Durch Integration kann daraus, mit etwas mehr an Rechenaufwand, $1/f^2$ Rauschen gewonnen werden. Ungeschickterweise gibt es keinen einfachen Algorithmus um $1/f$ Rauschen zu generieren. Mit der später beschriebenen Turbulenzfunktion wird man den Anforderungen jedoch näherungsweise gerecht.

Diskretes Gitter Terraingeneratoren müssen keine beliebig feinen Details erzeugen. Dies hat zwei Gründe. Zum einen muss man beim korrekten Erzeugen von Rauschen das Nyquist-Kriterium beachten. Hohe Frequenzen führen wegen des diskreten Gitters zum Alias-Effekt. Wenn niedere Frequenzen mit einer Periode, die größer ist als die zu berechnende Fläche, vorhanden sind, schlagen sich diese nur in der Neigung der Fläche wider. Deshalb werden bandbegrenzte Rauschgeneratoren eingesetzt.

Zum Zweiten besteht das Ausgabemedium aus einer abzählbaren Menge diskreter Bildschirm–Pixel. Geht man nun von der Situation aus, dass das Höhenfeld den Bildschirm ausfüllt und aus der Vogelperspektive betrachtet wird, so reichen für das Höhenfeld 512×512 Pixel. Dies entspricht ungefähr der halben Auflösung eines 1024×768 Bildschirms. Bei Rauschgeneratoren, die pro Gitterpunkt mehrere Oktaven überlagern, kann die Anzahl der benötigten Oktaven wie folgt abgeschätzt werden.

$$Oktaven = \log_2(Bildschirmaufloesung) - 2$$

$$Oktaven = \log_2(1024) - 2 = 8$$

Der Term -2 hat hängt mit der Nyquist-Grenze zusammen. Die Bildschirmauflösung liefert eine diskrete Abtastung des Höhenfeldes. Es kann nur ein Höhenfeld vollständig dargestellt werden das die halbe Frequenz, das heißt die halbe Anzahl an Stützpunkten, wie die Bildschirmauflösung hat. Des Weiteren entspricht die höchste Frequenz, welche auf dem diskreten Gitter des Höhenfeldes darstellt werden kann der halben Gittergröße. Ergibt $1/2 * 1/2 = 1/4$ und $\log_2(1/4) = -2$. Deshalb ist es in der Regel ausreichend 6 bis 8 Oberschwingungen zu überlagern.
Der zweite, eher praktische Grund ist die Reduzierung der Datenmenge und der Berechnungsdauer. Dies wird im Kapitel 6 näher beschrieben.

fBM auf einem diskreten Gitter Die fraktale Brownsche Bewegung stellt den klassischen Ansatz zum Erzeugen von Höhenprofilen dar. Die Grundfläche wird in ein Gitter mit äquidistanten Punkten eingeteilt. Für jeden Punkt wird durch einen Zufallsprozess ein Höhenwert bestimmt. Der Zufallsprozess erstellt die Werte so, dass benachbarte Punkte mit hoher Wahrscheinlichkeit ähnliche Werte erhalten. Für weiter entfernte Punkte ist jedoch keine stochastische Abhängigkeit vorhanden. In seiner allgemeinen Form wird pro Punkt des Gitters eine gewichtete Summe mehrere Rauschfunktionen ermittelt.

$$fBM(x) = \sum_{i=0}^{n-1} N(xLi)L^{-iH}$$

n ist die Anzahl der Oktaven. Also die Anzahl wie oft die Rauschfunktion mit einer jeweils höheren Frequenz, überlagert wird.
L sind die Lücken oder Sprünge zwischen den verwendeten Frequenzen. Der Fachbegriff lautet: lacunarity.
$N(x)$ ist der Wert der Rauschfunktion am Gitterpunkt x.
H hat die Funktion des Hurst Exponenten und bestimmt die fraktale Dimension der Oberfläche.

Die fBM Funktion ist homogen und isotrop. Dies bedeutet, dass das erzeugte Terrain überall die gleiche Rauigkeit besitzt. Dies führt zu einem unnatürlich wirkenden Terrain. Der Betrachter erwartet, dass ein Tal deutlich gleichförmiger als ein Berggipfel gestaltet ist.

Pseudozufallsgeneratoren und $1/f$ Rauschen Die Abarbeitung eines Algorithmus auf einem Computer erfolgt deterministisch. In einem deterministischen System kann es keinen echten Zufall geben. Aus der Physik sind nicht deterministische Vorgänge bekannt, zum Beispiel der Zeitpunkt des Zerfalls eines Atomkerns. Diese stochastischen Vorgänge können durch einen Algorithmus nur näherungsweise nachgebildet werden. Zur Unterscheidung verwendet man für die deterministische Generierung von scheinbar zufälligen Werten den Begriff Pseudozufallszahlengenerator.
Ein Pseudozufallszahlengenerator liefert:

1. Eine Zahlenfolge, die sich erst nach sehr vielen Ziehungen wiederholt.

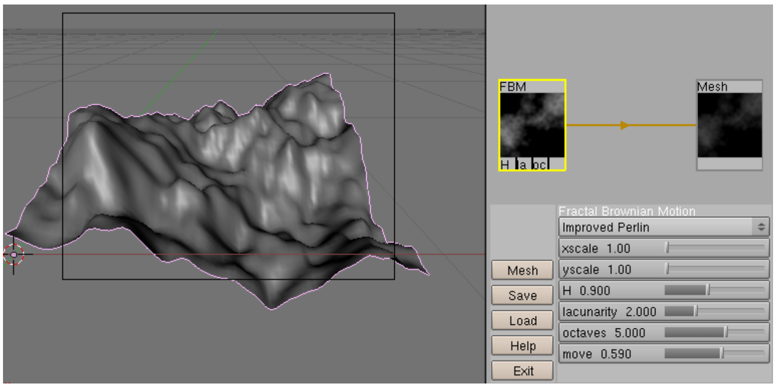

Abbildung 5: Dieses, mit fraktaler brownscher Bewegung erzeugte, Terrain hat durch seinen Hurst Exponenten von 0.9 im Bereich des Berges eine natürlich wirkende Geländeform. Aber im Tal rechts vorne erwartet der Betrachter einen deutlich sanfteren Geländeverlauf.

2. Ganzzahlige Zahlen, die in einem vorgegebenen Wertebereich mit gleicher Wahrscheinlichkeit auftreten.

3. Die Möglichkeit dieselbe pseudozufällige Zahlenfolge beliebig oft identisch zu generieren.

Die Zufallszahlen werden durch einen linearen Kongruenzalgorithmus erzeugt. Die Implementierung des Algorithmus für den `nrand48` Generator aus der GCC-Bibliothek ist wie folgt definiert:

1. Die Funktion liefert nicht negative, ganzzahlige Werte X_n mit 48-Bit-Auflösung.

2. Die Werte sind im Bereich 0 bis 2^{31} gleich verteilt.

3. Die Werte X_n werden mit der linearen Kongruenz Formel iterativ berechnet.

$$X_n = (aX_{n-1} + c) \, mod \, m, \text{ mit } n >= 0$$

$$m = 2^{48}, \text{ da 48-bit Integer} - \text{Arithmetik zu Grunde liegt.}$$

$$a = 0x5DEECE66D$$

$$c = 0xB$$

Benötigt wird jedoch eine Zahlenfolge, die ein $1/f$ Spektrum aufweist. Diese Aufgabe erweist sich als schwieriger, als sie auf den ersten Blick erscheint. Um dieses Ziel zu erreichen, sind zwei Schritte nötig. Zuerst wird, mithilfe des gleich verteilten Pseudozufallsgenerators, eine normal verteilte, pseudozufällige Zahlenfolge erzeugt. Diese normal verteilte, pseudozufällige Zahlenfolge stellt ein weißes Rauschen mit konstanter Amplitude im Leistungsdichtespektrum dar. Im zweiten Schritt wird daraus $1/f$ Rauschen erzeugt. Dieser Schritt ist in Kapitel 3 beschrieben.

Perlin Noise Im Frequenzspektrum des weißen Rauschen treten alle Frequenzen mit derselben Energie auf. Auch Frequenzen, die deutlich höher sind als die Nyquistfrequenz sind vertreten.

Dies ist in der digitalen Signalverarbeitung nicht handhabbar. Zum einen kann weißes Rauschen nur mit einem unendlichen Rechenaufwand auf einem digitalen System berechnet werden. Zum anderen können durch die diskreten Bildschirmpixel oder dem Punkteraster einer Druckmaschine diese hohen Frequenzen nicht mehr dargestellt werden. Oberhalb der Nyquist - Grenze treten Antialiasing Effekte auf. Von Ken Perlin wurde deshalb ein Zufallszahlengenerator entwickelt, der mit Absicht weit von der mathematischen Idealvorstellung des weißen Rauschens entfernt ist, aber gut zum Anwendungsfeld passt.
Die Perlin Noise Funktion weist folgende Eigenschaften auf:

- Perlin Noise ist eine Pseudozufallszahlenfolge, die beliebig oft identisch wiederholt werden kann.
- Perlin Noise hat einen bekannten eingeschränkten Wertebereich. Zum Beispiel -1 bis 1.
- Das Spektrum ist bandbegrenzt.
- Die pseudozufällige Zahlenfolge weist keine offensichtliche Periodizität auf. Der darunterliegende Pseudozufallsgenerator muss eine entsprechend hohe Periodenlänge aufweisen.
- Die statistischen Eigenschaften der erzeugten Zahlenfolge ist stationär. Dies bedeutet, dass diese Eigenschaften unabhängig vom Ort sind.

Aus Sicht der Generierung von Terraindaten ist die letztgenannte Eigenschaft nicht hilfreich. Perlin Noise erzeugt auf der gesamten Fläche denselben optischen Eindruck. Nach heutigem Stand der Technik dient Perlin Noise, wegen seiner guten Eignung für digitale Signalverarbeitung, in den meisten Terrain- oder Texturgeneratoren als innerer Kern. Allerdings sind zusätzliche Maßnahmen erforderlich um realistisch wirkende Ergebnisse zu erreichen.

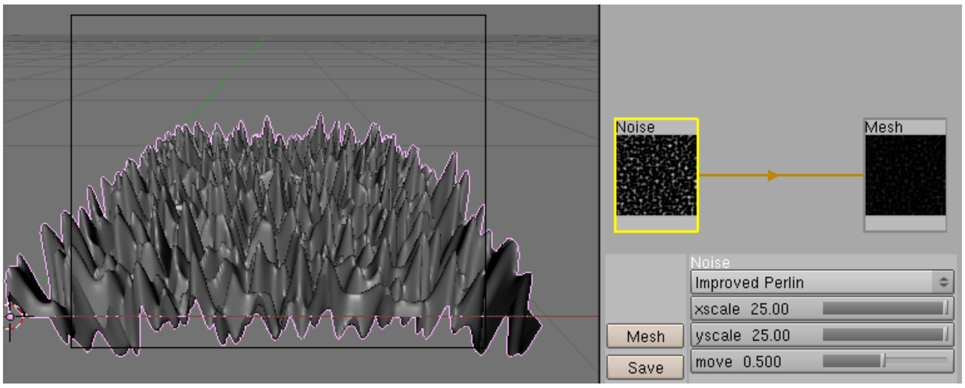

Abbildung 6: Perlin Noise hat auf der gesamten Fläche dieselben statistischen Eigenschaften. Um überzeugende Terraindaten zu liefern, muss Perlin Noise so nachbearbeitet werden, dass ein $1/f$ Leistungsspektrum entsteht. Das Ergebnis ist in Abbildung 10 zu sehen.

Alias Effekt Die mathematischen Funktionen zur Erzeugung der Terraindaten arbeiten diskret, aber dadurch, dass sie in Gleitkommaarithmetik mit 64 Bit Auflösung berechnet werden, liefern sie ein annähernd kontinuierliches Signal. Dieses Signal wird nun auf ein quadratisches Gitter mit relativ geringer Auflösung übertragen. Dieser Vorgang entspricht dem Abtasten eines analogen

Signals in der digitalen Signalverarbeitung. Es kommt zu Alias Effekten, wenn das Abtasttheorem verletzt wird. Liefert die Generatorfunktion Frequenzanteile, die höher sind als die halbe Abtastfrequenz des Gitters, kommt es zu Artefakten im Höhenfeld, die im Bild deutlich auffallen. Selbst wenn die Nyquistgrenze nicht überschritten wird, kann es möglich sein, dass das zugrunde liegende quadratische Gitter visuelle Störungen verursacht. Als Beispiel dient eine Cosinusfunktion, welche auf dem Einheitsquadrat circa 16 Perioden produziert. In Abbildung 7 wird diese stetige Funktion auf ein 20×20 Gitter übertragen. Die Abtastung liegt deutlich unter der geforderten Abtastfrequenz $f_{\text{abtast}} > 2\,f_{\max}$ als $f_{\text{abtast}} > 2 * 20$. Auf einem 64×64 Gitter wird das Abtastkriterium eingehalten, aber das zugrunde liegende Gitter ist ersichtlich. Ein 512×512 Gitter erzeugt ein zufriedenstellendes Bild. Allerdings steigt die Anzahl der Gitterpunkte und damit der Rechenaufwand quadratisch mit der Gitterauflösung. Damit kann jede Grafikhardware an ihre Leistungsgrenze gebracht werden. Es sind deshalb intelligentere Verfahren zur Reduzierung der irrelevanten Details wünschenswert.

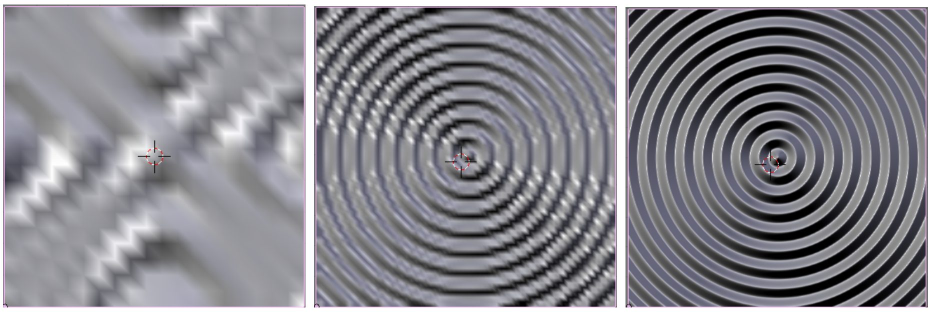

Abbildung 7: a) Die linke Abbildung zeigt eine unterabgetastete Cosinusschwingung. Die zu geringe Auflösung des Gitters erzeugt ein durch den Alias Effekt gestörtes Bild. b) In der mittleren Abbildung wird die Nyquist Grenze eingehalten, aber das 64×64 Gitter des Höhenfeldes erzeugt sichtbare Störungen. c) Das rechte Bild entspricht den Erwartungen. Durch sein 512×512 Gitter ist der Rechenaufwand aber um den Faktor $\frac{512^2}{64^2} = 64$ gestiegen.

3 Stochastische Synthese

Die unterschiedlichen stochastischen Terraingeneratoren lassen sich auf nur drei unterschiedliche Verfahren zurückführen: Midpoint Displacement, inverse Fourier Transformation und Perlin Noise. Die ersten zwei genannten Verfahren werden in diesem Kapitel kurz beschrieben. Heute wird überwiegend der in Kapitel 2 beschriebene Perlin Noise eingesetzt.

Stochastisch abhängige Zufallsgeneratoren Ein einzelner Wert in einem Höhenfeld soll stark von den Werten in seiner Nachbarschaft abhängig sein, aber unabhängig von den weiter entfernten Orten. Dadurch entstehen Terrains, die lokal glatt sind, aber global nicht voraussehbar sind. Betrachtet man dies aus der Sichtweise der digitalen Signalverarbeitung, kann man formulieren, dass ein zufälliges Signal, das Rauschen, erzeugt werden muss, bei dem die hohen Frequenzen kontrolliert gedämpft werden können. Im Folgenden werden die wichtigsten Rauschgeneratoren für diesen Zweck vorgestellt und gezeigt, wie sie mit den hohen Frequenzen umgehen.

Heuristische Methoden Die ersten Verfahren bauten auf dem Prinzip auf, dass eine Fläche in Teilflächen zerlegt wird. Diese Teilflächen werden wiederum rekursiv in kleinere Teilflächen aufgeteilt. Jede Teilfläche wird um einen zufälligen Betrag in der Höhe verschoben. Je tiefer die Rekursion, also je kleiner die Teilfläche ist, um so stärker wird der Zufallsprozess für die Translation der Höhe abdämpft. Damit erfüllt man die Forderung nach einem lokal abhängigen, aber global unabhängigen Höhenfeld.

Midpoint Displacement Der bekannteste Vertreter dieser Generatoren trägt den Namen Midpoint Displacement. In seiner einfachsten Form wird das Ausgangsquadrat rekursiv in kleinere Quadrate unterteilt.

Erzeugt wird ein Zufallsfeld.

$$X(t_1, t_2, ..., t_n)$$

Die Höhenunterschiede $X(t_1, t_2, ..., t_n) - X(s_1, s_2, ..., s_n)$ sind gaussverteilt mit dem Mittelwert 0. Die Varianz der Höhenunterschiede $X(t_1, t_2, ..., t_n) - X(s_1, s_2, ..., s_n)$ ist von der Distanz abhängig.

$$\sqrt{\sum_{i=1}^{n}(t_i - s_i)^2}$$

Alle Punkte des Zufallsfeldes haben die selben statistischen Eigenschaften. Im Frequenzbereich gilt für das Spektrum:

$$S(f_1, ..., f_n) \propto \frac{1}{(\sum_{i=1}^{n} f_i^2)^{2H+n}}$$

Dieses Verfahren hat jedoch zwei Mängel. Zum einen sehen damit erzeugte Terrains sehr zackig aus. Des Weiteren weist die Geländeform unabhängig von der Höhe dieselbe Rauheit auf. Das Verfahren eignet sich gut zur Simulation von Hochgebirge und Wolken. Eine detaillierte Darstellung findet man in PIESAU1988.

Synthese mithilfe der inversen Fouriertransformation.

iFFT Eine andere Methode zur Generierung von Terrainformen erzeugt im ersten Schritt ein zufällig verteiltes Spektrum. Die Beträge der Frequenzen in diesem Spektrum bestimmen das Verhältnis zwischen großen und kleinen Strukturen. Anschließend wird dieses Spektrum mithilfe der diskreten inversen Fourier Transformation aus dem Frequenzbereich in den räumlichen Bereich transformiert. Der Einsatz der Fourier Spektralsynthese dient hauptsächlich zum Erzeugen von Wasseroberflächen. Im Unterschied zu den oben genannten Rauschgeneratoren werden durch die Fourier Synthese sanftere Formen erzeugt. Siehe Abbildung 8.
Terrains, welche durch FFT erzeugt werden, sind ein Ausschnitt $[0..2\pi]$ aus einer periodischen Funktion. Dadurch kann ein und dasselbe Terrain nahtlos gekachelt werden. Die Ränder passen sowohl in der Höhe, als auch in der Steigung zusammen. Wir haben es hier mit einer Funktion zu tun, die im Detail zufällig ist, aber trotzdem periodisch. Siehe Abbildung 9.
Bei Wasseroberflächen ist es zu aufwendig ein Höhenfeld zu berechnen. In den Szenen modelliert man Wasseroberflächen durch Ebenen. Allerdings stört man auf diesen Ebenen die Flächenormale durch eine Rauschfunktion. Damit bleibt die Wasseroberfläche für die Sichtbarkeitsberechnung im Raytracers ein simples geometrisches Objekt. Die Oberflächeneigenschaften sind jedoch komplex.

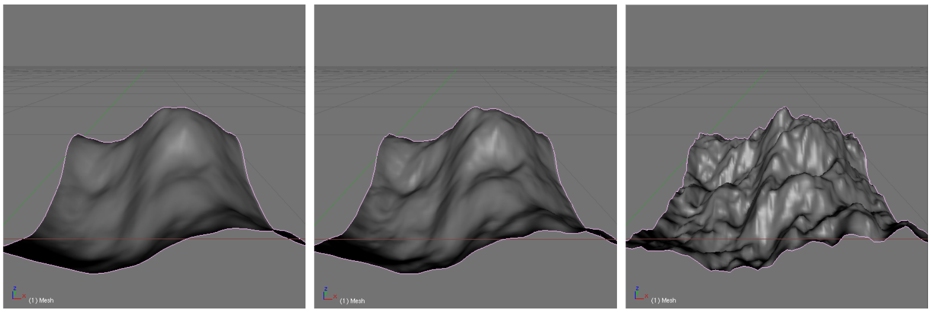

Abbildung 8: Ein durch spektrale Synthese erzeugtes Terrain. Von links nach rechts steigt die fraktale Dimension der Zufallsfunktion.

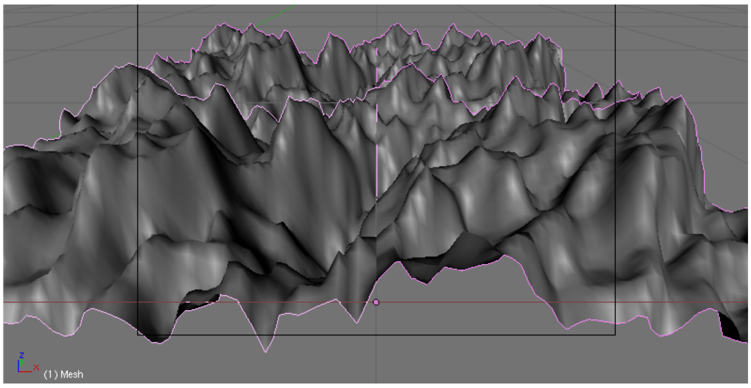

Abbildung 9: Hier ist vier Mal dasselbe Höhenfeld aneinandergesetzt. Da die Kantenlänge der Grundfläche genau einer Periode entspricht, passen die Ränder nahtlos zusammen. Die gesamte Ebene könnte in X- als auch in y-Richtung unendlich oft mit der gleichen Kachel bedeckt werden, ohne dass Sprünge in der z-Achse an den Rändern sichtbar werden.

Damit ein Betrachter einen Teil der virtuellen Umgebung als Wasseroberfläche wahrnimmt, ist die Simulation der Reflexionseigenschaften weit wichtiger, als das Höhenmodel.

Verwante des Perlin Noise Die in Abbildung 6 gezeigte Perlin Noise Funktion erzeugt bei weitem noch kein realistisch wirkendes Terrain. Allerdings wird sie in etlichen Rauschfunktionen als innerer Kern verwendet. Als Beispiel wird hier eine in der Literatur als Turbulenz bezeichnete Funktion vorgestellt. Auch sie wurde ursprünglich von Ken Perlin entwickelt.
Die Turbulenzfunktion besitzt folgende Eigenschaften:

- Sie erzeugt ein Leistungsspektrum, in dem die Amplitude im Verhältnis $1/f$ zur Frequenz abfällt.
- Es werden 2 bis 6 Oberschwingungen überlagert.
- Die Überlagerung der Oberschwingen erfolgt über den Absolutwert. Dadurch ist die Funktion nicht überall differenzierbar.
- Durch diese Unstetigkeit ist die Turbulenzfunktion nicht bandbegrenzt.

Wie die Noise Funktion hat die Turbulenzfunktion über die gesamte Fläche dieselben statistischen Eigenschaften. Im Kapitel 5 wird gezeigt, wie dieser Mangel an Realität umgangen werden kann. Einsatzgebiete für Rauschfunktionen, welche auf Perlin Noise aufbauen: Gebirge, Wolken.

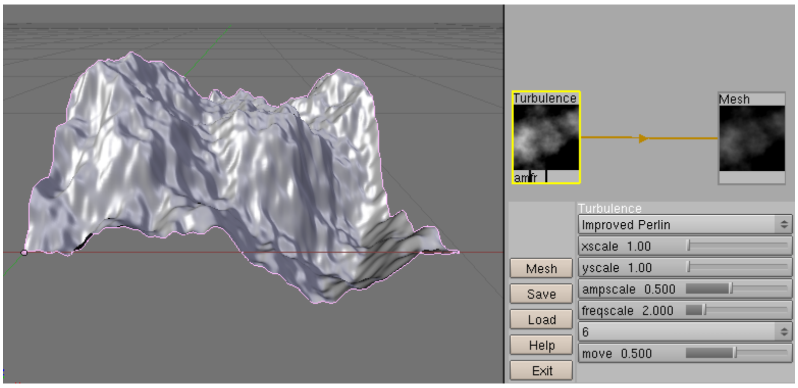

Abbildung 10: Diese Turbulenzfunktion überlagert 6 Oktaven einer Perlin Noise Funktionen. Auch hier besitzt das Tal dieselbe Rauigkeit wie die Bergspitzen.

4 Homogene und heterogene Terrains

Die bisher aufgezählten Rauschgeneratoren können nur genau einen Terraintyp, zum Beispiel Gebirge oder Hügel, pro Parametrisierung der Kennwerte des Rauschgenerators generieren. Sie erzeugen auf der gesamten Fläche dieselbe fraktale Dimension. Die statistischen Kennwerte sind unabhängig von Ort oder Zeit. In der Statistik wird diese als stationärer Zufallsprozess bezeichnet. Im Kontext der Terraingenerierung spricht man von einem homogenen Terrain. Infolge des stationären Zufallsprozesses erscheinen Täler gleich rau wie Bergspitzen. Um diesen Mangel an Realität zu verbergen, zeigten die ersten computergenerierten Terrains Felsinseln, die aus einer Wasseroberfläche aufragen. Eine Wasseroberfläche füllt dabei die niederen Bereiche des Höhenprofils aus, um die unrealistisch hohe Zerklüftung dieses künstlich erzeugten Terrains in den Tälern zu verbergen.

Multifraktale Rauschgeneratoren Mutifraktale unterscheiden sich von klassischen Fraktalen dadurch, dass sie an jedem Ort eine andere fraktale Dimension besitzen. Mit einem multifraktalen Rauschgenerator ist man in der Lage innerhalb einer einzigen Funktion ein Terrain zu erzeugen, das unterschiedliche Rauigkeiten aufweist. Damit das generierte Höhenfeld als Terrain erkennbar ist, müssen Randbedingungen beachtet werden. Zwei benachbarte Orte sollten sich nicht nur in der Höhe, sondern auch in der fraktalen Dimension wenig unterscheiden. Für weiter entfernte Orte darf diese statische Abhängigkeit nicht gelten. Zusätzlich sollte die fraktale Dimension mit der Höhe oder der Steigung korrelieren. Der Betrachter erwartet zum Beispiel, dass Berggipfel oder Abhänge die stärkste Zerklüftung aufweisen.
In den zuvor vorgestellten Verfahren werden die Oktaven additiv überlagert. Die Rauschfunktion, mit einer homogenen fraktalen Dimension, kann man sich wie eine Fourierreihenentwicklung vorstellen. In multifraktalen Rauschfunktionen fließen die hohen Frequenzen jedoch multiplikativ ins Ergebnis ein. Dies führt dazu, dass diese Generatoren numerisch instabil sein können. Zumindest muss ihr Ergebnis, abhängig von der Parametrisierung, skaliert werden.

Die folgenden zwei Programmbeispiele zeigen den Unterschied zwischen einer homogenen und einer multifraktalen Generatorfunktion. Noise ist dabei die Perlin Noise Funktion. Beide Beispiele wurden 1994 von Kenton Musgrave in EBMU1994 veröffentlicht.

Homogenes Fraktal durch additive Überlagerung. Siehe Zeile 12.

```
1   /* Initialisierung des Spektrums */
2   for(i=0; i<=octaves; i++) {
3       exp[i]  = pow(frequence, -H );
4       frequence *= lacunarity;
5       ...
6   }
7
8   value = 0.0;
9   /* innere Schleife für spektrale Konstruktion */
10  for(i=0; i<octaves; i++) {
11     /* additiv für homogenes Fraktal */
12      value += exp[i]  * noise( point )
13      point *= lacunarity;
14  }
15  return value;
```

Multifraktal durch multiplikative Überlagerung. Siehe Zeile 5.

```
1   value = 0.0;
2   /* innere Schleife für spektrale Konstruktion */
3   for(i=0; i<octaves; i++) {
4      /* multiplikativ für Multifraktal */
5       value *= noise( point );
6       point *= lacunarity;
7   }
8   return value;
```

Einsatzgebiete für Rauschfunktionen, welche auf Multifraktalen aufbauen: Landschaften mit unterschiedlichen Typen von Bergen, Wolken.

5 Filterketten

Wie oben erwähnt erwartet der Betrachter, dass eine natürliche Landschaft unterschiedliche Geländeformen enthält. Im mathematischen Sinn haben sich die verschiedenen Gebiete eines Terrains durch unterschiedliche fraktale Dimensionen zu unterscheiden. Wobei nicht immer die Erhöhungen die höhere Dimension aufweisen muss. Im Fall eines Kannions, welcher sich in eine Hochfläche einschneidet, wird die höchste fraktale Dimension im Steilhang erwartet.

Nicht lineare Skalierung Eine simple, aber wirkungsvolle Methode um die Gleichförmigkeit des Rauschgenerators im generierten Terrain zu verbergen, ist die nichtlineare Skalierung der Höhendaten. Liegen die Werte des Rauschgenerators im Bereich 0 bis 1, dämpft die Funktion

$f(z) = z^2$ die kleinen Werte stärker als die Werte nahe 1. Abbildung 11 zeigt ein solches Terrain.

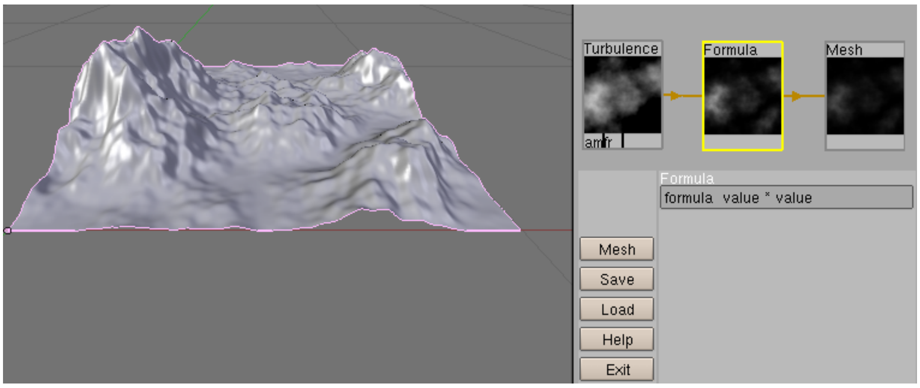

Abbildung 11: Das in Abbildung 10 dargestellte Höhenfeld wird durch die Funktion $f(z) = z^2$ nichtlinear skaliert.

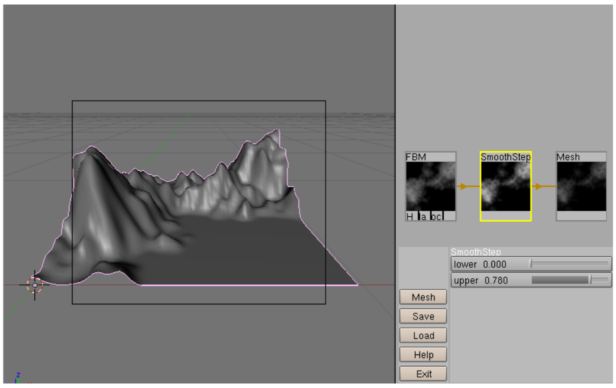

Abbildung 12: Das in Abbildung 5 dargestellte Höhenfeld wird in seinen unteren Bereichen durch den nichtlineare Smoothstep Operator gedämpft.

Statistische Kennzahlen in Abhängigkeit von der Höhe Ein Nachteil der bisher vorgestellten Generatoren sind die homogenen statischen Eigenschaften über der gesamten Fläche. Eine Ausnahme bilden lediglich die Multifraktale. Schaut man sich die homogenen Zufallszahlengeneratoren genauer an, so stellt man fest, dass sie sehr wohl über Parameter verfügen, um die Rauigkeit zu verändern. Diese Parameter sind zum Beispiel der Hurst-Exponent oder die Anzahl der überlagerten Oberschwingungen. Allerdings wurden bisher diese Werte für ein Terrain immer konstant gehalten. Interessant wäre es diese Werte über die Fläche zu variieren. Damit kann man eine lokale Kontrolle des Zufallsprozesses realisieren. In einem simplen Ansatz könnte man den Hurst-Exponenten, abhängig von der Position, verändern. Abbildung 13 ist auf diese Weise entstanden. Ein radialer Gradient steuert, abhängig von der Position, die fraktale Dimension im Rauschgenerator. Zusätzlich skaliert er direkt die erzeugten Höhendaten.
Ein weiterer Ansatz beruht auf folgender Überlegung: Die statistischen Eigenschaften sollten nicht vom Ort, sondern von der Höhe an diesem Ort abhängig sein. Der Generator sollte sich selbst beeinflussen können. Diese rekursive Denkweise läst sich mit dem Terrain - Plugin nicht direkt

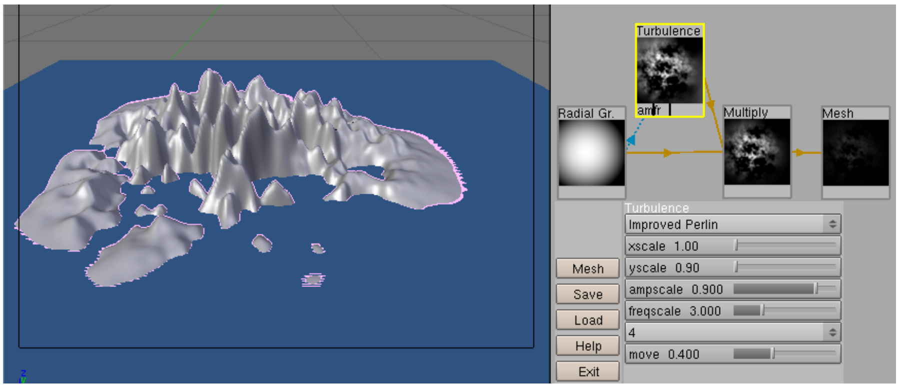

Abbildung 13: Diese Pirateninsel weist unterschiedliche Terrainformen auf. Außer den Klippen, welche die Bucht im Inneren begrenzen, sind in Küstennähe Dünen und Sandstrände auszumachen. Die genaue Lage des Schatzes ist nicht eingezeichnet.

modellieren, denn dort dürfen die gerichteten Graphen keine Schleifen aufweisen. In diesem Anwendungsfall kann man diese Einschränkung umgehen in dem man zwei Generatoren, welche identisch parametrisiert sind, einsetzt. Einer der Generatoren dient dazu, den Hurst-Exponenten des für das Terrain zuständigen Generators lokal zu beeinflussen. Wie in Abschnitt 2 erwähnt führen kleine Werte des Hurst-Exponenten zu einer zerklüfteten Geländeform, deshalb muss der Werte des ersten Generators invertiert und skaliert werden. Ohne diesen Zwischenschritt würde man eine Hochebene erzeugen in die sich eine Schlucht einschneidet. Abbildung 5 und Abbildung 14 beruhen auf denselben Parametereinstellungen, durch die Dämpfung der fraktalen Dimension wirkt das Tal in Abbildung 14 natürlicher.

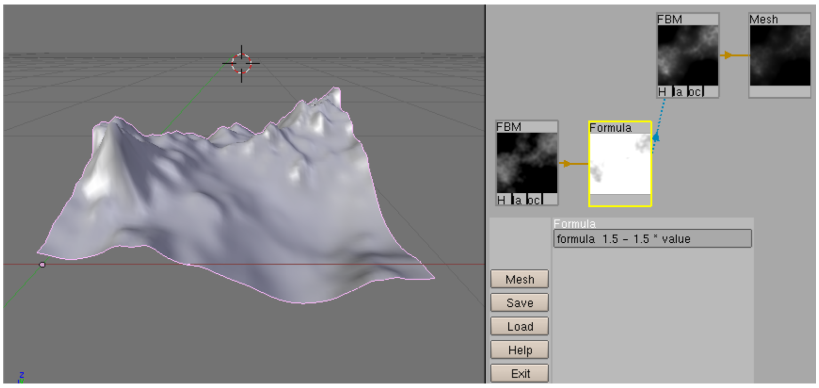

Abbildung 14: In dieser Abbildung ist die fraktale Dimension abhängig von der Höhe. Dadurch haben Berge und Täler unterschiedliche statistische Eigenschaften. Dies entspricht den Erwartungen eines Betrachters.

Mischen von Terraindaten Soll ein Terrain zwei völlig unterschiedliche Features aufweisen, ist es möglich diese Terrains mit unterschiedlichen Filterketten zu erstellen und dann mit der Maximum Funktion zu vereinigen. An den Übergangstellen ist das Resultat nicht differenzierbar. Diese Kanten wirken in der 3D-Darstellung unnatürlich, fallen aber im Graustufenbild zuerst nicht auf. Anzustreben ist, dass an der Übergangstelle zwischen den zwei Features außer der Höhe auch die erste Ableitung der Höhe gleich ist. In der Praxis werden Median-Filter verwendet. Abbildung 15 ist auf ähnliche Weise entstanden, ohne die Steigung an der Übergangskante zwischen Berg und Ebene zu beachten.

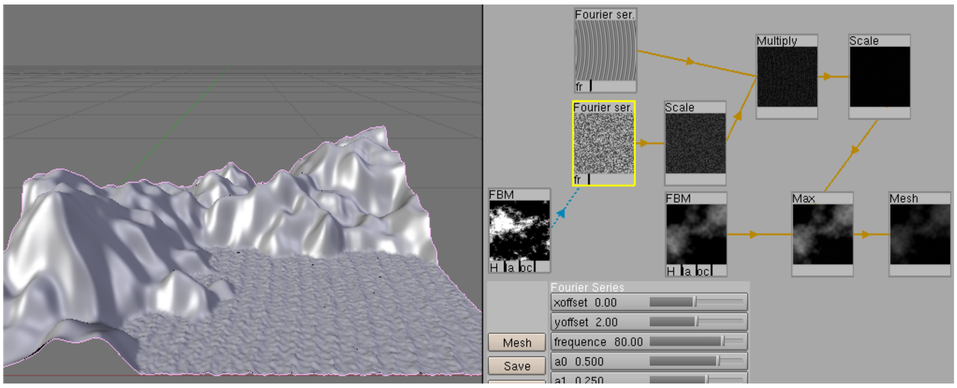

Abbildung 15: Fraktale brownsche Bewegung erzeugt die Berge. Die Ebene im Vordergrund basiert auf einer Fourierreihe.

In vielen Fällen sind mit den unterschiedlichen Terrainformen auch unterschiedliche optische Eigenschaften verbunden. Hier fällt es leichter, zwei getrennte Höhenfelder zu erzeugen und jedem seine spezifischen optischen Eigenschaften zuzuweisen. Die restliche Arbeit wird dann vom Raytracer oder Radiosity Tool erledigt. An den Übergangstellen treten Kanten auf, die wenig mit der physikalischen Wirklichkeit zu tun haben.

6 Grenzen und Erweiterungen

Level of Detail Ein Problem der Computer generierten Grafiken ist der Mangel an dargestellten Details. Aus der Sicht eines Betrachters wirken die erzeugten Grafiken zu monoton. Die für einen realistischen Eindruck benötigten Datenmengen sind in der geforderten Zeit nicht zu berechnen. Vor allem 3D Spiele, welche in Echtzeit gerendert werden, sehen zu simpel aus. Aus Sicht der grafischen Algorithmen kann man diese Frage auch so angehen: Wenn es mit der gegebenen Speicherausstattung und Grafikhardware möglich ist eine bestimmte Anzahl an Details darzustellen, welche Details können dann weggelassen werden ohne die Qualität deutlich zu mindern? Dies nennt man Level of Detail. Diese Aufgabenstellung stellt sich nicht nur für Höhenfelder. Auch Texturen und Beleuchtungsmodelle können schneller berechnet werden, wenn abhängig von den Randbedingungen, unwesentliche Details Wegegelassen werden.
Beim statischen Level of Detail wird das Terrain in Kacheln unterteilt. Für jede Kachel wird bestimmt, wie viele Stützpunkte für das Höhenfeld berechnet werden müssen. Als Kriterium könnten die Höhenunterschiede der Randpunkte herangezogen werden, aber auch die Entfernung vom Betrachtungsstandpunkt. Ein Vorteil dieses Ansatzes ist, dass die unerwünschten Daten von Anfang an gar nicht berechnet werden. Als Nachteil ist zu nennen, dass die Gefahr besteht, dass kleine, aber für den Bildaufbau relevante Details, verloren gehen.
Raytracer sind darauf optimiert Netze von Dreiecken zu rendern. Das Höhenfeld wird in einer Stufe der Rendering Pipeline in ein solches Dreiecksnetz gewandelt. Hier ist es möglich eine Polygonreduzierung zwischenzuschalten. Dreiecke mit geringen Höhendifferenzen werden zu größeren

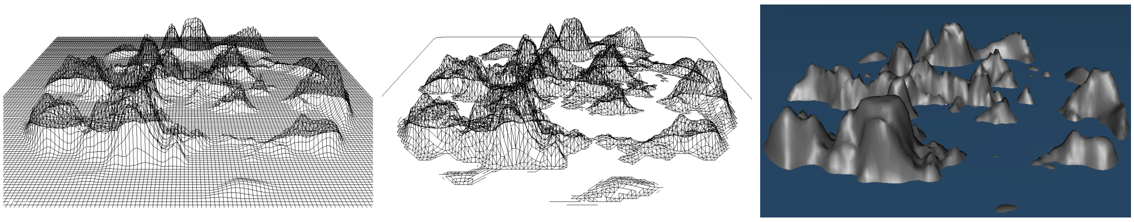

Abbildung 16: a) Ein aus Quadraten schematisch aufgebautes Netz. b) Dasselbe Netz nach der Polygonreduzierung. c) Das Höhenmodel in Grau. Die blaue Ebene wurde zusätzlich eingezogen.

Polygonen zusammengefasst. Diese Verfahren gehört in die Gruppe der kontinuierlichen Level of Detail Verfahren. Die Gefahr, dass kleine Details verloren gehen, besteht nicht. Sie wurden ja bereits erzeugt und werden vom Polygonreduzierer nicht entfernt. Der Nachteil liegt darin, dass nur der Rendering Prozess entlastet wird, nicht die Datengenerierung für das Terrain.

Entwässerungssysteme Der auffälligste Mangel dieser Modelle ist das fehlen von Entwässerungssystemen. Dieser Mangel wird bei einer Betrachtung aus der Vogelperspektive sofort sichtbar. In diesen Terrains gibt es keine erkennbar zusammenhängenden Täler. Keine baumartigen Strukturen bestehend aus Bächen, Flüssen und Strömen sind erkennbar. Es ist möglich Erosionseffekte zu simulieren, diese führen allerdings nur dazu, dass die lokalen Minima aufgefüllt werden, aber nicht dazu, dass ein zusammenhängendes System von Tälern entsteht. Seit der Entwicklung dieser Verfahren Ende der 80er Jahre wurde in diesem Bereich keine wirklichen Fortschritte gemacht. An dieser Stelle sei nochmals erwähnt, dass es sich hier nicht um die Simulation physikalischer Vorgänge handelt, sondern lediglich um generierte Daten mit statistischen Kennwerten, wie sie auch in natürlichen Umgebungen gefunden werden können.
Als Ausweg bleibt, das Höhenprofil in groben Zügen manuell als Graustufenbild zu zeichnen. Dieses Bild wird dann mit den aus dem System heraus generierten Daten gemischt, da man durch die manuelle Vorarbeit nur den Verlauf des Terrains im Groben, aber nicht die Details, vorgeben will. Das importierte Bild soll sowohl auf die Höhe des Terrains, als auch auf die fraktale Dimension einwirken. Das Ergebnis ist in Abbildung 17 zu sehen.

Überhänge Überhänge in einem Terrain sind mit Höhenfeldern nicht darstellbar. Damit ist es zum Beispiel unmöglich, mit dem vorgestellten Verfahren, einen Höhleneingang zu erzeugen. Diese Einschränkung hat man jedoch auch, wenn man mit Geodaten arbeitet, die von Satelliten erfasst werden. Auch dort ist ein Überhang nicht erfassbar.

Texturen Die Farbgebung, Transparenz und Reflexionseigenschaften eines geometrischen Objektes wird in computergenerierten Grafiken unter dem Begriff Textur zusammengefasst. Solche Texturen können mit den oben genannten Verfahren berechnet werden. Die berechnete 'Höhe' wird als Index in einen Farbgradienten aufgefasst. Im Unterschied zu den Höhenprofilen werden vom Betrachter bei den Texturen Diskontinuitäten akzeptiert. Diese Diskontinuität kann man nützen um Kanten in Texturen zu simulieren.

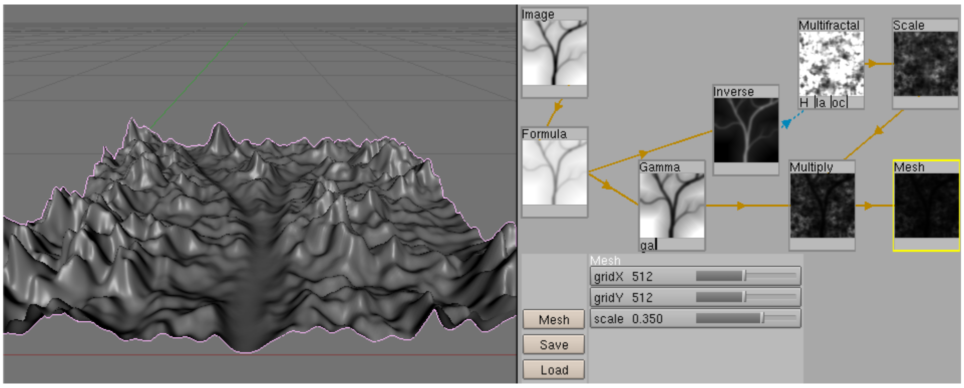

Abbildung 17: Mischen von manuell erzeugten und generierten multifraktalen Daten. Der Verlauf des Tales kann nicht aus dem System selbst heraus berechnet werden.

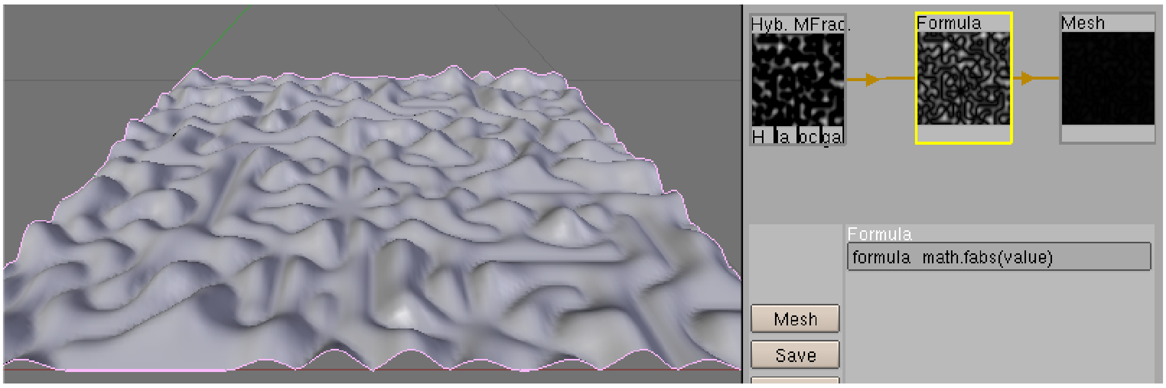

Abbildung 18: Durch die Absolutwertfunktion $F(z) = \mid z \mid$ treten Kanten im Höhenprofil auf.

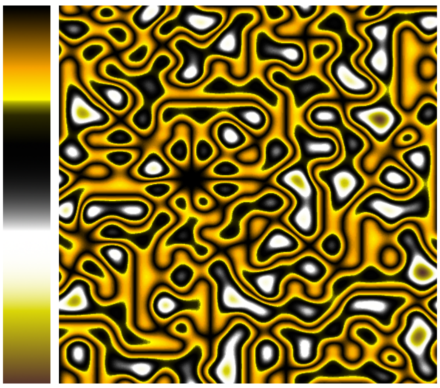

Abbildung 19: Mithilfe des Farbgradienten wird das Höhenfeld in ein RGB-Bild transformiert. Die Unstetigkeitsstellen in Abbildung 18 wirken in der Textur wie ein Ornament.

7 Quellen und Links

Quellen

F. Kenton Musgrave
Methods for Realistic Landscape Imaging
Yale University, 1993
http://www.kenmusgrave.com/dissertation.pdf

DABA2006
Carsten Dachsbacher
Interactive Terrain Rendering: Towards Realism with Procedural Models and Graphics Hardware
Friedrich-Alexander-Universität Erlangen-Nürnberg, 2006
http://www.opus.ub.uni-erlangen.de/opus/volltexte/2006/354/

PIESAU1988
Heinz-Otto Peitgen, Dietmar Saupe (Herausgeber)
The Science of Fractal Images (Gebundene Ausgabe)
Springer-Verlag (August 1988)
ISBN: 0387966080

GEMS2004
Jerry Tessendorf
Interaktive Wasseroberflächen
in
Andrew Kirmse (Herausgeber)
Spieleprogrammierung Gems 4
Hanser Fachbuchverlag (Oktober 2004)
ISBN: 3446229442

EBMU1994
David S. Ebert, F. Kenton Musgrave, Darwyn Peachey
Texturing and Modeling
Morgan Kaufmann; (1994)
ISBN: 1558608486

Bildnachweis

http://de.wikipedia.org/wiki/Bild:White.noise.png
http://de.wikipedia.org/wiki/Bild:Pink.noise.png
http://de.wikipedia.org/wiki/Bild:Red.noise.png

Links

Homepage von Dr. Forest Kenton Musgrave
http://www.kenmusgrave.com/
3D Modellierung mit Blender
http://www.blender.org/

Blender Plugin für Terraingenerierung
`http://innerworld.sourceforge.net/`