

**Philip Lehmann-Böhm**

**Echtzeitdarstellung von Volumendaten mit  
einem Level of Detail System**

**Masterarbeit**

# BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei [www.GRIN.com](http://www.GRIN.com) hochladen  
und kostenlos publizieren



## **Bibliografische Information der Deutschen Nationalbibliothek:**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

## **Impressum:**

Copyright © 2012 GRIN Verlag  
ISBN: 9783668702363

## **Dieses Buch bei GRIN:**

<https://www.grin.com/document/425416>

**Philip Lehmann-Böhm**

# **Echtzeitdarstellung von Volumendaten mit einem Level of Detail System**

## **GRIN - Your knowledge has value**

Der GRIN Verlag publiziert seit 1998 wissenschaftliche Arbeiten von Studenten, Hochschullehrern und anderen Akademikern als eBook und gedrucktes Buch. Die Verlagswebsite [www.grin.com](http://www.grin.com) ist die ideale Plattform zur Veröffentlichung von Hausarbeiten, Abschlussarbeiten, wissenschaftlichen Aufsätzen, Dissertationen und Fachbüchern.

### **Besuchen Sie uns im Internet:**

<http://www.grin.com/>

<http://www.facebook.com/grincom>

[http://www.twitter.com/grin\\_com](http://www.twitter.com/grin_com)

# **Masterthesis**

**Zur Erlangung des akademischen Grades eines  
Master of Science  
über das Thema**

## **Echtzeitdarstellung von Volumendaten mit einem Level of Detail System**

Eingereicht am Fachbereich 4 Wirtschaftswissenschaften II  
der Hochschule für Technik und Wirtschaft Berlin

Von:  
Philip Lehmann-Böhm

Berlin, den 06.11.2012

# Zusammenfassung und Abstract

## Zusammenfassung

Außenszenarien in der Computergrafik beinhalten oft die Darstellung von weitflächigem Terrain. Volumenbasiertes Terrain ist eine flexible Methode, die Überhänge, Kliffe oder Höhlen ermöglicht. Die Echtzeitdarstellung benötigt dabei höchste Effizienz.

Die vorliegende Arbeit beschreibt ein neues Verfahren zur Darstellung volumenbasierter Daten mit einem Level of Detail Mechanismus und Terrain als Hauptanwendung. Dabei wird dargestellt, woher die Daten stammen, wie aus ihnen ein Polygonnetz gebildet wird, wie die Dreiecke eine Textur und Relief erhalten und wie mittels des Level of Detail Algorithmus der Detailgrad der Darstellung je nach Entfernung zum Betrachter variiert.

Die Quelle der Daten sind zwei Verfahren. Constructive Solid Geometry baut einen Baum aus Grundformen wie Kugeln und Ebenen sowie Operationen wie Vereinigung und Schnitt auf. 3D Texturen, die aus einem externen Editor stammen, können verarbeitet werden.

Durch eine Kombination aus Dual Marching Cubes und Marching Squares Skirts werden Polygonnetze aus den Volumendaten generiert und mittels triplanarer Texturierung mit einem passenden Material versehen.

Zur Darstellung des Volumens werden viele separate Polygonnetze mehrerer Detailstufen erzeugt und in einer Baumstruktur organisiert. Je nach Entfernung zum Betrachter werden aus diesen „Chunks“ die passenden ausgewählt. Dieser Level of Detail Algorithmus sorgt dafür, dass auch große Volumendaten mit flüssiger Bildrate darstellbar sind.

## Abstract

Outdoor scenes in the field of computer graphics often contain the rendering of large terrains. Volume based terrain is a flexible solution allowing overhangs, cliffs or caves. It is important for real-time rendering, that this solution is as efficient as possible.

---

This work describes a novel method for rendering volume based data with a level of detail mechanism with terrain as the main application. Areas covered will include: where the data comes from, how a mesh is generated, how the triangles get their texture and relief, and finally how the amount of detail varies over the distance with the level of detail algorithm.

The source of the data consists of two methods. Constructive solid geometry builds up a tree made of basic shapes like spheres or planes as well as operators like union and intersection. 3D textures coming from an external editor can be processed.

With the use of Dual Marching Cubes and Marching Squares Skirts, meshes from the volume data are generated and with triplanar texturing, they get a fitting material.

For the rendering of the volume, many separate meshes with different levels of detail are created and organized in a tree structure. Depending on the distance to the viewer, the matching „chunks“ amongst them are chosen. This level of detail algorithm takes care of displaying huge volumes within a fluent frame-rate.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>vii</b>
<b>Formelverzeichnis</b>	<b>ix</b>
<b>Abkürzungsverzeichnis</b>	<b>xi</b>
<b>Glossar</b>	<b>xii</b>
<b>Quellcodeverzeichnis</b>	<b>xiii</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Problemstellung . . . . .	2
1.2. Vorgehensweise . . . . .	3
<b>2. Volumenmodell</b>	<b>5</b>
<b>3. Zusammenhängende Entwicklung</b>	<b>7</b>
3.1. Isosurface Generierung . . . . .	7
3.2. Material . . . . .	10
3.3. Höhenfeld basiertes Terrain . . . . .	11
3.4. Volumenbasiertes Terrain . . . . .	12
<b>4. Dichtefunktion</b>	<b>14</b>
4.1. 3D Texturen . . . . .	14
4.2. Constructive Solid Geometry . . . . .	17
4.2.1. Grundformen . . . . .	18
4.2.2. Operatoren . . . . .	19
4.2.3. Ein Constructive Solid Geometry Baum . . . . .	21
<b>5. Verfahren der Volumendarstellung</b>	<b>22</b>
5.1. Dual Marching Cubes . . . . .	22
5.1.1. Generierung des Octrees . . . . .	23
5.1.2. Isolierung von Features . . . . .	26
5.1.3. Bilden des Dualgitters . . . . .	27
5.1.4. Konturierung des Dualgitters . . . . .	37

5.2. Marching Squares Skirts . . . . .	40
5.3. Material . . . . .	42
5.3.1. Triplanare Texturierung . . . . .	43
5.3.2. Normal Mapping . . . . .	46
5.3.3. Beleuchtung . . . . .	48
5.3.4. Nebel . . . . .	51
5.4. Level of Detail . . . . .	53
5.4.1. Aufbau des Chunkbaums . . . . .	54
5.4.2. Chunkauswahl beim Zeichnen . . . . .	57
<b>6. OGRE</b>	<b>60</b>
6.1. Google Summer of Code 2012 . . . . .	61
6.2. Wichtige Klassen für das Projekt . . . . .	62
<b>7. Softwarearchitektur</b>	<b>64</b>
7.1. Aufbau des Chunkbaums . . . . .	65
7.2. Auswahl der Chunks beim Zeichnen . . . . .	68
7.3. Weitere Funktionen . . . . .	68
<b>8. Implementierung</b>	<b>69</b>
8.1. Dichtefunktion . . . . .	69
8.2. Chunkbaum . . . . .	69
8.3. Material . . . . .	71
8.4. Chunkauswahl beim Zeichnen . . . . .	72
<b>9. Abschlussbetrachtung</b>	<b>73</b>
9.1. Leistungsfähigkeit . . . . .	73
9.1.1. Ladezeit . . . . .	75
9.1.2. Laufzeit . . . . .	76
9.2. Ergebnis des Google Summer of Code 2012 . . . . .	82
9.3. Zukünftige Entwicklung . . . . .	83
9.3.1. Entwicklung eines grafischen Editors . . . . .	83
9.3.2. Paging . . . . .	84
9.3.3. Constructive Solid Geometry . . . . .	85
9.3.3.1. CSGXML . . . . .	85
9.3.3.2. Rauschfunktion . . . . .	86
9.3.3.3. Verbesserte CSG Operatoren . . . . .	86
9.3.3.4. Matrix CSG . . . . .	86
9.3.3.5. Weitere Grundformen . . . . .	87
9.3.4. Schnitt mit Strahlen und den generierten Dreiecken . . . . .	87
9.3.5. Material . . . . .	88

---

9.3.5.1.	Mehrere Textursets bei der triplanaren Texturierung . . . . .	88
9.3.5.2.	Ambient Occlusion . . . . .	89
9.3.5.3.	Parallax Occlusion Mapping . . . . .	89
9.3.6.	Verbesserung des Polygonnetzes . . . . .	90
9.3.6.1.	Implementierung von Marching Cubes 33 . . . . .	90
9.3.6.2.	Extraktion von scharfen Kanten . . . . .	90
9.3.6.3.	Verbesserung der LOD-Wechsel . . . . .	90
9.3.6.4.	Reduzierung der Anzahl kleiner Dreiecke . . . . .	91
9.4.	Fazit . . . . .	92
<b>Literaturverzeichnis</b>		<b>93</b>
<b>Einsatz der Volumen-Komponente</b>		<b>100</b>

# Abbildungsverzeichnis

2.1. Visualisierung eines 2D Kreisvolumens . . . . .	5
4.1. die Eckpunkte eines Kubus zur trilinearen Interpolation . . . . .	16
4.2. die CSG Grundformen dieses Projektes . . . . .	18
4.3. die CSG Operatoren dieses Projektes außer der Skalierung . . . . .	20
4.4. ein CSG-Baum . . . . .	21
5.1. Visualisierung eines Octrees mit Baumstruktur . . . . .	23
5.2. die Nummerierung der Kinder eines Octree-Knotens. . . . .	23
5.3. ein Octree-Knoten mit seinen Kindern und allen Eckpunkten . . . . .	24
5.4. der eindimensionale Fall der Fehlerberechnung zwischen Interpolation und Funktion, nach [ZBS05, S. 16] . . . . .	25
5.5. der Octree einer Kugel mit Zentrum im Ursprung des abgetasteten Volumens . . . . .	26
5.6. überschneidende Dreiecke mit geänderter Eckpunktfolgenfolge bei Dual Marching Cubes nach [MS10, S. 2] . . . . .	27
5.7. das grüne Dualgitter gebildet aus dem schwarzen Quadrees nach SCHAEFER[SW04, S. 3] . . . . .	28
5.8. die Aufrufe der Unterfunktionen von <i>nodeProc(n)</i> . . . . .	30
5.9. die Aufrufe der Unterfunktionen von <i>faceProcXY(n0, n1)</i> . . . . .	31
5.10. die Aufrufe der Unterfunktionen von <i>edgeProcX(n0, n1, n2, n3)</i> . . . . .	32
5.11. der Aufruf von <i>vertProc(n0, n1, n2, n3, n4, n5, n6, n7)</i> von sich selbst . . . . .	34
5.12. ein Dualgitter zu einem Octree ohne Randzellen . . . . .	34
5.13. ein Dualgitter zu einem Octree mit Randzellen . . . . .	37
5.14. das adaptive Dualgitter zu einem Octree . . . . .	38
5.15. links: zwei triangulierte Kuben mit Löchern, rechts: zwei triangulierte Kuben ohne Löcher nach [Hom10, S. 15] . . . . .	39
5.16. links: Lücken, die durch aneinanderliegende Polygonnetze gebildet durch Octrees unterschiedlicher Auflösung, entstehen, rechts: die Lücken sind gefüllt mit Marching Squares Skirts . . . . .	40
5.17. Marching Squares Skirts am Rand einer Viertelkugel . . . . .	41
5.18. die 16 möglichen Marching Squares Fälle . . . . .	41
5.19. Texturierung mittels planarer Projektion . . . . .	43