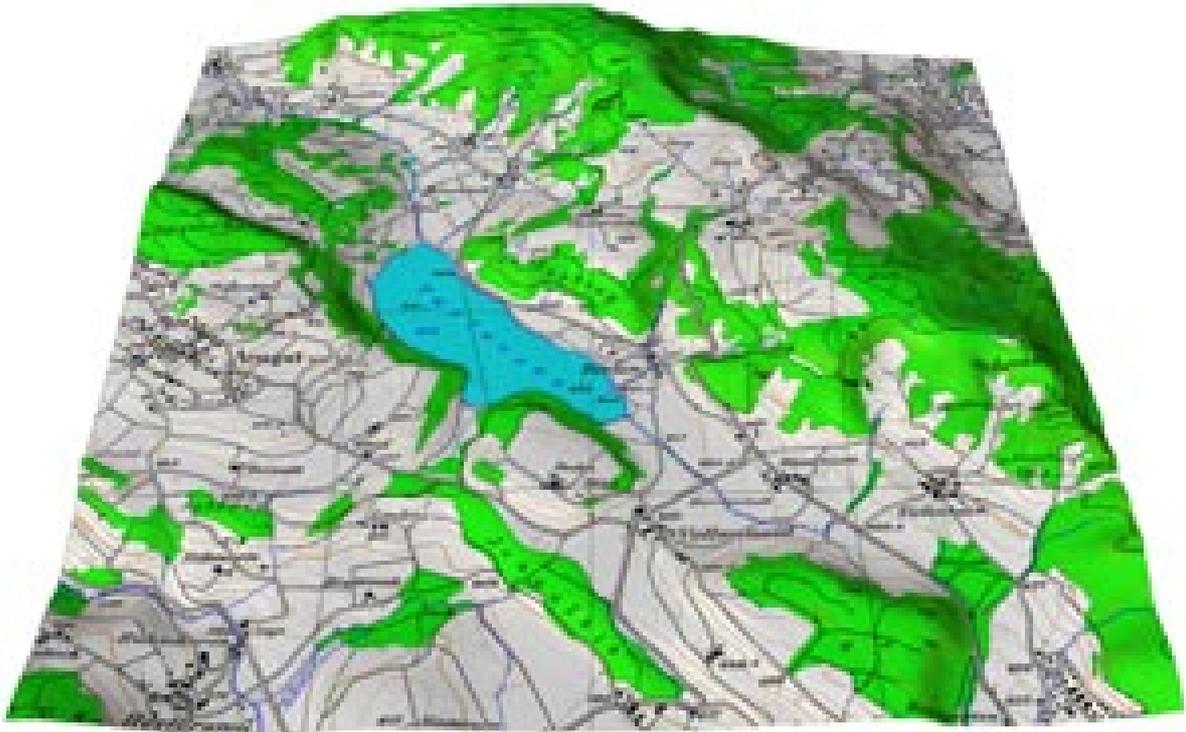


Real-Time, Continuous Level of Detail Rendering of Height Fields



Fachseminar Graphische Datenverarbeitung SS '97

Vortrag gehalten am 1. Juli 1997 von Martin Näf

Basierend auf der gleichnamigen Publikation von

Peter Lindstrom, David Koller, William Ribarsky, Larry F. Hodges, Nick Faust
Georgia Institute of Technology

Gregory Turner
Army Research Laboratory

Inhaltsverzeichnis

Einleitung	3
Grundlagen	4
Triangulated Irregular Network (TIN)	4
Uniformly Gridded Network	4
Andere	4
Anforderungen	5
Polygonreduktion	6
Datenstruktur	6
Vereinfachung auf Stützpunktstufe	6
Vereinfachung auf Blockstufe	7
Stützpunktabhängigkeiten	8
Resultate	9
Implementation der Autoren des Originalpublikation	9
Eigene Implementation	10
Fazit	11
Literaturverweise	12

Einleitung

Moderne Grafikhardware kann Modelle mit einigen Tausend Polygonen in Echtzeit darstellen. Es gibt aber einige Anwendungen mit Modellen, welche die Möglichkeiten aktueller Hardware bei weitem überschreiten. Ein Beispiel sind Höhenmodelle in geographischen Anwendungen.

Nachfolgend werden zuerst ein paar grundlegende Techniken zur geometrischen Repräsentation von Höhenfeldern behandelt, anschliessend ein Algorithmus zur Polygonreduktion für Höhenfelder mit gleichmässigem Stützpunktabstand. Der Algorithmus lässt sich durch folgende Punkte charakterisieren:

- Reduktion der Anzahl Polygone um mehrere Grössenordnungen
- Kontinuierliche Änderungen zwischen einzelnen Detailstufen
- Dynamisches Erstellen der Detailstufen in Echtzeit
- Unterstützung benutzerdefinierter Fehlertoleranz (Pixel)

Der Bericht schliesst mit einer Übersicht der erzielten Resultate der Implementation der Autoren der Originalpublikation sowie einer eigenen Implementation.

Grundlagen

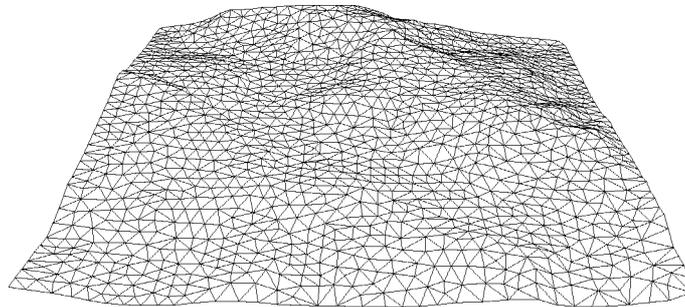
Es existiert eine grosse Anzahl Algorithmen zur Approximation von Geländeoberflächen mittels Polygonnetzen. Diese versuchen, bei gegebener Anzahl Stützpunkte, maximaler Fehlertoleranz oder spezieller Berücksichtigung von Geländemerkmale, eine optimale Darstellung der Oberfläche.

Die verwendeten Methoden können grob in folgende Kategorien eingeteilt werden:

Triangulated Irregular Network (TIN)

Nicht-uniform triangulierte Oberflächen erlauben eine unterschiedliche Dichte von Stützpunkten. Damit lässt sich ein Netz konstruieren, welches die gegebene Oberfläche beliebig genau approximiert.

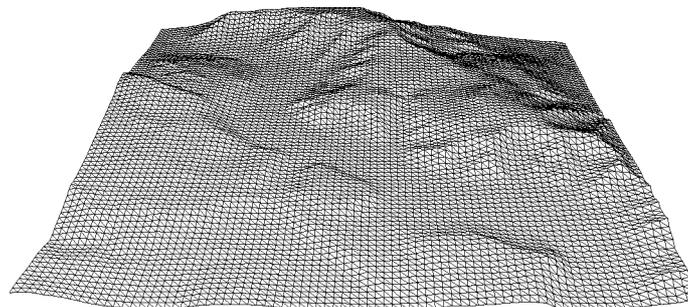
TINs sind zur dynamischen Generierung von Detailstufen im Allgemeinen ungeeignet, der Rechenaufwand hierzu wäre zu gross.



TIN Höhenmodell Albis

Uniformly Gridded Network

Die Stützpunkte sind auf einem gleichmässigen Raster angeordnet. Uniformly gridded networks bestehen typischerweise aus wesentlich mehr Polygonen als TINs bei gleicher Approximationsgenauigkeit, können aber meist kompakter dargestellt werden (nur Höheninformation pro Stützpunkt). Ein weiterer Vorteil ist die Möglichkeit, durch Subsampling leicht verschiedene Detailstufen gewinnen zu können.



Höhenmodell Albis 50m Raster

Andere

Weitere Methoden sind häufig Hybride der beiden oben genannten Verfahren oder nutzen spezielle, informationsreiche Geländeeigenschaften wie Grate oder Täler aus. Weiter ist noch die Methode der Wavelets Transformation zu erwähnen, welche eine lokale Kontrolle der Detailstufe erlaubt.

Anforderungen

Ein typisches Einsatzgebiet von Echtzeit LOD Algorithmen sind Simulatoren verschiedener Art. Dies stellt gewisse Anforderungen an den verwendeten Algorithmus:

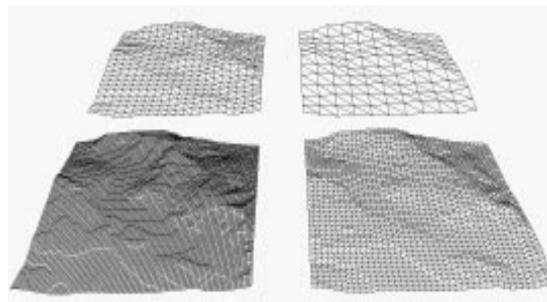
1. Es sollte möglich sein, die Höhe eines beliebigen Punktes ohne Aufwand berechnen zu können.
2. Dynamische Änderungen der Geometrie sollten keinen signifikanten Einfluss auf die Leistung des Systems haben.
3. Der Einfluss von hochfrequenten Strukturen sollte lokal beschränkt sein.
4. Kleine Änderungen des Kamerastandpunktes sollten nur kleine Änderungen der Geometriekomplexität zur Folge haben.
5. Der maximale Darstellungsfehler soll beschränkt sein und durch einen Parameter direkt und intuitiv bestimmt werden können.

Punkt 1 ist wichtig für Kollisionserkennung bei Spielen oder Simulatoren oder für der Oberfläche folgende Objekte. Dieses Ziel wird von TIN Algorithmen im allgemeinen nicht erreicht.

Punkt 2 fällt in erster Linie für militärische Simulationen und Spiele ins Gewicht, wo häufig Änderungen der Oberfläche stattfinden (Explosionen).

Die grosse Schwäche von Uniformly Gridded Networks liegt darin, dass die Auflösung des gesamten Netzes erhöht werden muss, um auch kleine Strukturen optimal darstellen zu können (Punkt 3). Dies führt zu «Popping», welches durch zusätzliche Massnahmen wie Blending oder Morphing reduziert werden müssen.

Typisch für die üblichen Detailstufenalgorithmen ist die diskrete Abstufung der Auflösung. Dies führt zu grossen Änderungen der Polygonzahl bei Übergängen zur nächsten Stufe und damit zu deutlichen Änderungen der Bildrate. Punkt 4 sollte erfüllt sein, damit die Bildrate vorhersehbar und konstant ist.



Diskrete Detailstufen am Beispiel Höhenmodell Albis:
50m, 100m, 200m und 400m Raster

Der Eigenschaft 5 sollte besondere Beachtung geschenkt werden. Viele Detailstufenalgorithmen erlauben keinen direkten Rückschluss vom verwendeten Fehlerparameter auf den effektiven Fehler nach der Projektion in den 2D-Bildraum. Dies erschwert die Wahl eines geeigneten Parameters und birgt die Gefahr für «Popping» oder einen unnötig hohen Detaillierungsgrad in sich.

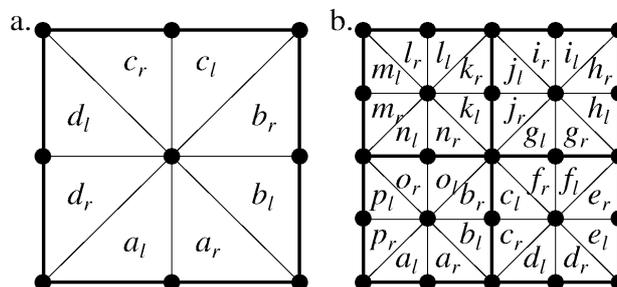
Polygonreduktion

Der hier vorgestellte Algorithmus besteht aus zwei Schritten: Einer groben Vereinfachung zur Bestimmung der diskreten Detailstufe und einer anschließenden Retriangulierung der entstandenen Blocks, in welcher Stützpunkte einzeln entfernt werden. Der Algorithmus stellt sicher, dass keine Vereinfachungen auf Blockstufe vorgenommen werden, welche nicht auch bei vollständiger Anwendung der Retriangulierung auf dem gesamten Gitter erfolgen würden.

Beide Schritte werden für jedes Frame einer Animation vollständig und abhängig vom aktuellen Kamerastandpunkt gerechnet.

Datenstruktur

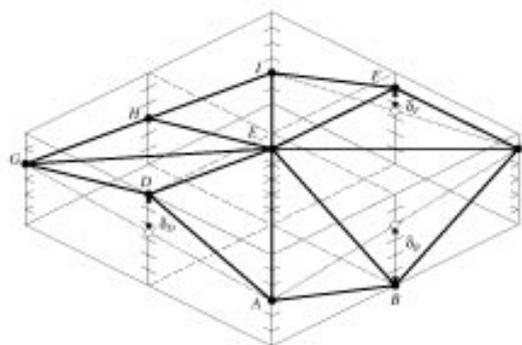
Das Höhenfeld wird durch ein uniformes Gitter von Punkten mit Abstand x_{res} und y_{res} beschrieben. Die kleinste Netzrepräsentation besteht aus einem 3x3 Raster der Form a.:



Grössere Blöcke werden aus einer 2x2 Anordnung der jeweils kleineren Stufe gebildet (b). Kleinere Detailstufen werden durch Weglassen jeder zweiten Reihe/ Kolonne erreicht. Eine Quadtree-Datenstruktur bietet daher sich automatisch an.

Vereinfachung auf Stützpunktstufe

Ausgangspunkt ist konzeptionell die kleinste Auflösung des Gitters. Während der feinen Phase der Vereinfachung werden jeweils 2 benachbarte Dreiecke verschmolzen, sofern gewisse Bedingungen erfüllt sind. Die resultierenden Dreiecke werden rekursiv weiter verschmolzen.



Benachbarte Dreiecke können dann verschmolzen werden, wenn der Winkel zwischen den Dreiecken klein genug ist. Die Winkeländerung wird als vertikaler Abstand des gemeinsamen Eckpunktes B von der Geraden zwischen A und C gemessen. Diese Distanz δ stellt den grössten Fehler dar, der bei der Verschmelzung entsteht.

Die Entscheidung, ob verschmolzen wird, hängt ab von der projizierten Länge (in Pixel) von δ . Ist diese Länge kleiner als ein Schwellwert τ , wird der Stützpunkt weggelassen.

Nach ein paar Vereinfachungen resultiert daraus die folgende Ungleichung:

$$\frac{d^2 \lambda^2 \delta^2 ((e_x - v_x)^2 + (e_y - v_y)^2)}{((e_x - v_x)^2 + (e_y - v_y)^2 + (e_z - v_z)^2)^2} \leq \tau^2$$

e ist der Viewpoint, d der Abstand der Projektionsebene zum Viewpoint, λ die Anzahl Pixel pro Weltkoordinateneinheit.

Nach weiteren Umformungen kommt man auf die folgende Form, welche nur noch wenige Multiplikationen und Additionen enthält:

$$\delta^2 \left((e_x - v_x)^2 + (e_y - v_y)^2 \right) \leq \kappa^2 \left((e_x - v_x)^2 + (e_y - v_y)^2 + (e_z - v_z)^2 \right)^2$$

$\kappa = \tau \div (d \times \lambda)$ ist eine Konstante.

Auf eine genauere Herleitung der Ungleichung wird hier verzichtet.

Wählt man $\tau \leq 1$ Pixel, ist die Verschmelzung, abgesehen von kleinen Effekten durch Rundungsfehler und Shading, nicht sichtbar.

Vereinfachung auf Blockstufe

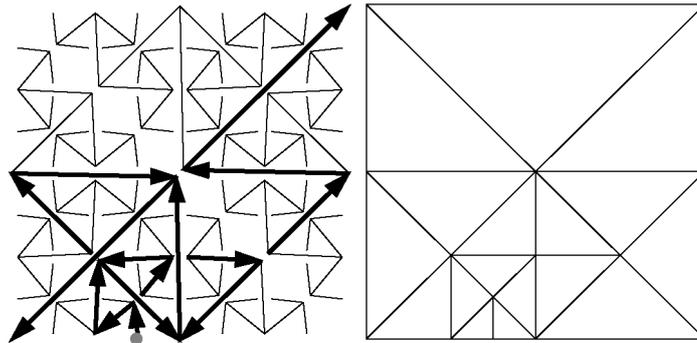
Typische Höhenmodelle können leicht aus mehreren Millionen Stützpunkten bestehen. Es ist weder sinnvoll, noch möglich, in jedem Frame für jeden Stützpunkt die Delta-Projektion durchzuführen.

Die Anzahl Projektionen lässt sich deutlich reduzieren, indem ganze Blöcke zusammengefasst werden und nur der grösste Delta-Wert eines Blockes projiziert wird. Fällt diese Projektion unter den Grenzwert τ , so können sämtliche Stützpunkte des Blockes in einem Schritt entfernt werden. Übersteigt sie dagegen den Grenzwert, muss eine höhere Detailstufe gewählt werden. Die resultierende diskrete Detailstufe wird anschliessend stützpunktbasiert neu trianguliert.

Da ein Block eine Ausdehnung $\gg 0$ besitzt, lässt sich kein fixer Schwellwert festlegen. Statt dessen wird ein Intervall $I = [\delta_{\min}, \delta_{\max}]$ definiert. Delta-Werte, welche in dieses Intervall fallen, müssen mittels regulärer Delta-Projektion einzeln evaluiert werden. δ_{\min} und δ_{\max} werden berechnet, indem der nächste bzw. der am weitesten vom Kamerastandpunkt entfernte Punkt des Blocks zur Bestimmung der Grenzen herangezogen wird.

Stützpunktabhängigkeiten

Ein Stützpunkt, welcher gemäss Delta-Projektion nicht entfernt werden darf, generiert gewisse Abhängigkeiten. Würden seine unmittelbaren Nachbarn entfernt, ergäbe dies Fehler in der Darstellung. Gelöst wird das Problem mittels Locking der benachbarten Knoten.



Abhängigkeitsgraph eines Punktes und die resultierende Triangulation

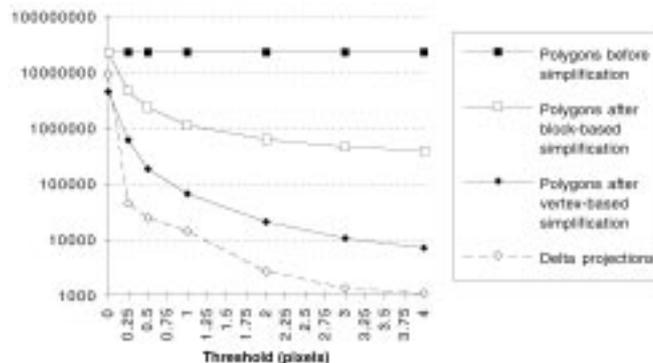
Durch die Einführung von diskreten Detailstufen durch die blockbasierte Vereinfachung ergeben sich zusätzliche Probleme bei der Darstellung von benachbarten Blöcken. Dreiecke können nur verschmolzen werden, wenn sie sich auf derselben Detailstufe befinden. Dies ist an Blockgrenzen häufig nicht gegeben. Eine Verschmelzung würde in einem solchen Fall zu sichtbaren Lücken in der Oberfläche führen.

Resultate

Implementation der Autoren der Originalpublikation

Die Autoren der Originalpublikation implementierten ihren Algorithmus auf einer 2-Prozessor SGI Onyx Reality Engine². Sie verwendeten das Höhenmodell der Hunter-Liggett Military Base in Kalifornien mit einer Schrittweite von 2×2 m und einer Höhenauflösung von 1 m.

Untenstehende Grafik zeigt den Umfang der Polygonreduktion nach den einzelnen Stufen sowie die Anzahl Delta-Projektionen, welche dazu durchgeführt wurden.



Man erkennt, dass bereits mit einem Maximalfehlerwert von 1 Pixel, bei welchem praktisch keine Popping-Effekte in Animationen sichtbar sind, die Anzahl Polygone um mehrere Größenordnungen reduziert werden konnte.

Zur Quantifizierung des Fehlers wurde das ursprüngliche Modell und die polygonreduzierte Variante sowohl mit Photo-Textur wie auch als Schachbrett gerendert und Pixelweise verglichen. Bei einem τ von 1 Pixel wurden nur gerade 5% der Pixel falsch gerendert. Bei $\tau = 2$ Pixel stieg dieser Wert auf ca. 9%, bei $\tau = 4$ auf ca. 13%.

Dieser Fehler ist bei kleinem τ praktisch nicht sichtbar.

Die massive Reduktion der Polygone ermöglichte sehr gute Renderzeiten. Bei einem Wireframe-Mesh der Auflösung 640×480 Pixel und einem $\tau = 2$ Pixel erreichten die Autoren eine konstante Frame-Rate von 20 Bildern pro Sekunde.

Es sollte noch erwähnt werden, dass die Implementation der Autoren fehlerbehaftet ist. Abhängigkeiten zwischen Blöcken verschiedener Auflösungen werden nicht korrekt behandelt, was zu Löchern in der Oberfläche führt. Ob und in welchem Ausmass die korrekte Behandlung dieser Fälle einen Einfluss auf die Leistung hat, blieb unkommentiert.

Eigene Implementation

Eine eigene Implementation des Polygonreduktionsalgorithmus basierend auf den gleichen Vereinfachungskriterien wurde auf einem Intel Pentium 90 MHz PC mit 3DLabs Glint Delta / Glint 500TX Hardware Renderer unter Windows NT 4.0 sowie auf einer DEC Alpha XL 300 (300 MHz Alpha 21164) getestet. Datensatz war das Höhenmodell der Schweiz im 250 m Raster, Höhenauflösung 1 m.

Gegenüber dem ursprünglichen Algorithmus wurden folgende Vereinfachungen vorgenommen: a) Nur Vertex-basierte Vereinfachung, b) Rekursion nur über 2 Stufen und c) nur horizontaler Abstand wird bei Delta-Projektion berücksichtigt. Damit wurde eine Behandlung der Abhängigkeiten auf ein Minimum reduziert, Löcher treten keine auf.

Die Vereinfachungen bewirken beim verwendeten groben Raster keine grosse Einbusse beim Reduktionsfaktor und sind somit unter den gegebenen Umständen vertretbar.

Die folgende Tabelle gibt die Grössenordnung der erzielten Resultate wieder. Sie entsprechen dem Durchschnitt eines Fluges quer durch den gesamten Datensatz. Sichtweite war 30 km, Gouraud-Shading, keine Texturen, Auflösung 768×576 Pixel (PAL Standard):

τ	Δ rendered	Δ removed	t_{render} Pentium	t_{render} Alpha	Framerate Pentium	Framerate Alpha
0	18000	0	330 ms	110 ms	3	9
1	10000	8000	250 ms	80 ms	4	12
2	6000	12000	200 ms	70 ms	5	14
4	4000	14000	160 ms	65 ms	6	15
8	2500	15500	120 ms	55 ms	8	18
16	2000	16000	110 ms	55 ms	9	18

t_{render} beinhaltet die komplette Polygonreduktion, das Rendering der Dreiecke sowie die Buffer Swap und Clear Operationen. Clipping und Polygonreduktion benötigten praktisch unabhängig vom gewählten τ auf dem Pentium PC ca. 20 ms, auf dem DEC Alpha ca. 8 ms.

Ab einem τ von 4 und grösser waren starkes «Popping» sichtbar.

Fazit

Der vorgestellte Algorithmus erfüllt sämtliche Anforderungen, welche zu Beginn gestellt wurden.

1. Die Bestimmung der Höhe eines bestimmten Punktes ist ohne Aufwand möglich, da zu jedem Zeitpunkt die vollständige Höheninformation zur Verfügung steht.
2. Dynamische Änderung der Geometrie stellt ebenfalls kein Problem dar, da bis auf δ -Werte und eventuell Flächennormale nichts vorberechnet wird.
3. Hochfrequente Strukturen bleiben durch grosse δ -Werte bedingt bestehen, ohne dass sich dies auf die Polygonzahl in anderen Gebieten auswirken würde.
4. Kleine Änderungen des Kamerastandpunktes bewirken eine kleine Änderung der Projektion und damit auch nur kleine Änderungen beim Entfernen der Stützpunkte. Eine kontinuierliche Änderung der Detailstufe ist damit gewährleistet.
5. Der Fehler bei der Vereinfachung lässt sich in Pixeleinheiten klar begrenzen. Damit ist sowohl die Bedingung der Beschränkung des Fehlers, wie auch ein intuitiver Fehlermassstab gegeben.

Der Algorithmus ermöglicht eine massive Polygonreduktion mit relativ wenig Aufwand.

Aufgrund der Resultate der Autoren sowie der eigenen Implementation, welche zwar einige Vereinfachungen, gleichzeitig aber auch noch viel Raum für Verbesserungen enthält, ist der Schluss berechtigt, dass die Real-Time Bedingung nicht nur auf leistungsstarken Maschinen wie der SGI Onyx erfüllt wird, sondern auch auf kleineren Workstations der Preisklasse unter U\$ 15000.

Literaturverweise

Originalpublikation:

LINDSTROM, P., KOLLER, D., RIBARSKY, W., HODGES, L.F., FAUST, N., TURNER, G.
Real-Time, Continuous Level of Detail Rendering of Height Fields
Proceedings of SIGGRAPH 96., ACM SIGGRAPH, pp. 109–118.
<http://www.gvu.gatech.edu/gvu/reports/techreports96.html>

Weitere Referenzen:

DE BERG, M. and DOBRINDT, K. T. G.
On Levels of Detail in Terrains.

In 11th ACM Symposium on Computational Geometry, June 1995.

DE FLORIANI, L. and PUPPO, E.

Hierarchical Triangulation for Multiresolution Surface Description.
ACM Transactions on Graphics 14(4), October 1995, pp. 363–411.

DOUGLAS, D. H.

Experiments to Locate Ridges and Channels to Create a New Type of Digital Elevation Model.
Cartographica 23(4), 1986, pp. 29–61.

ECK, M., DEROSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERY, M., and STUETZLE, W.
Multiresolution Analysis of Arbitrary Meshes.

Proceedings of SIGGRAPH 95. In Computer Graphics Proceedings, Annual Conference Series, 1995,
ACM SIGGRAPH, pp. 173–182.

FALBY, J.S., ZYDA, M.J., PRATT, D.R., and MACKEY, R.L.

Hierarchical Data Structures for Real-Time Three-Dimensional Visual Simulation.
Computers & Graphics 17(1), 1993, pp. 65–69.

FERGUSON, R.L., ECONOMY, R., KELLY, W. A., and RAMOS, P. P.

Continuous Terrain Level of Detail for Visual Simulation.
In Proceedings, IMAGE V Conference, June 1990, pp. 144–151.

FOWLER, R.J. and LITTLE, J. J.

Automatic Extraction of Irregular Network Digital Terrain Models.
Proceedings of SIGGRAPH 79. In Computer Graphics 13(2) (August 1979), pp. 199–207.

GARLAND, M. and HECKBERT, P. S.

Fast Polygonal Approximation of Terrains and Height Fields.
Technical Report CMU-CS-95-181, CS Dept., Carnegie Mellon U., 1995.

GROSS, M.H., GATTI, R., and STAADT, O.

Fast Multiresolution Surface Meshing.
In Proceedings of Visualization '95, October 1995, pp. 135–142.

HECKBERT, P.S. and GARLAND, M.

Multiresolution Modeling for Fast Rendering.
In Proceedings of Graphics Interface '94, 1994, pp. 1–8.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., and STUETZLE, W.

Mesh Optimization.
Proceedings of SIGGRAPH 93. In Computer Graphics Proceedings, Annual Conference Series, 1993,
ACM SIGGRAPH, pp. 19–26.

SAMET, H.

The Quadtree and Related Hierarchical Data Structures.
ACM Computing Surveys 16(2), June 1984, pp. 187–260.

SCARLATOS, L. L.

A Refined Triangulation Hierarchy for Multiple Levels of Terrain Detail.
In Proceedings, IMAGE V Conference, June 1990, pp. 114–122.

SCHRODER, F. and ROSSBACH, P.

Managing the Complexity of Digital Terrain Models.
Computers & Graphics 18(6), 1994, pp. 775–783.

SCHROEDER, W.J., ZARGE, J.A., and LORENSON, W.E.

Decimation of Triangle Meshes.
Proceedings of SIGGRAPH 92. In Computer Graphics 26(2) (July 1992), pp. 65–70.

SOUTHARD, D. A.

Piecewise Planar Surface Models from Sampled Data.
Scientific Visualization of Physical Phenomena, June 1991, pp. 667–680.

TAYLOR, D.C. and BARRET, W. A.

An Algorithm for Continuous Resolution Polygonalizations of a Discrete Surface.
In Proceedings of Graphics Interface '94, 1994, pp. 33–42.