

3D - Modellierung

Arne Theß

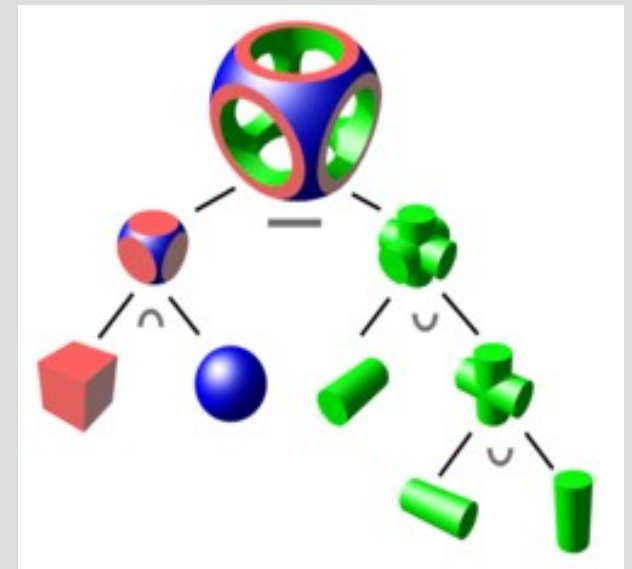
Proseminar Computergraphik
TU Dresden

Gliederung

- Darstellungsschemata
 - direkte
 - Constructive Solid Geometry (CSG)
 - Generative Modellierung
 - Voxelgitter
 - indirekte
 - Drahtgittermodell
 - Boundary Representation
- Texturierung
 - Texture Mapping
 - Bump Mapping
 - Normal Mapping
 - Displacement Mapping

Constructive Solid Geometry

- Verknüpfung von Basisobjekten mittels Boolescher Operatoren
- Basisobjekte („Primitive“): Körper mit mathematisch einfach zu beschreibender Oberfläche (Kugeln, Würfel, Zylinder, ...)
- CSG-Baum: hierarchische Anordnung der Operationen in binärem Baum
- Vorteile:
 - Relativ leicht zu erzeugen
 - Praktisch für CAD-Anwendungen, da Objekte vor allem im technischen Entwurf immer wieder auf ähnlichen Formen basieren
 - Mathematisch genaue Berechnung und Repräsentation der Körper

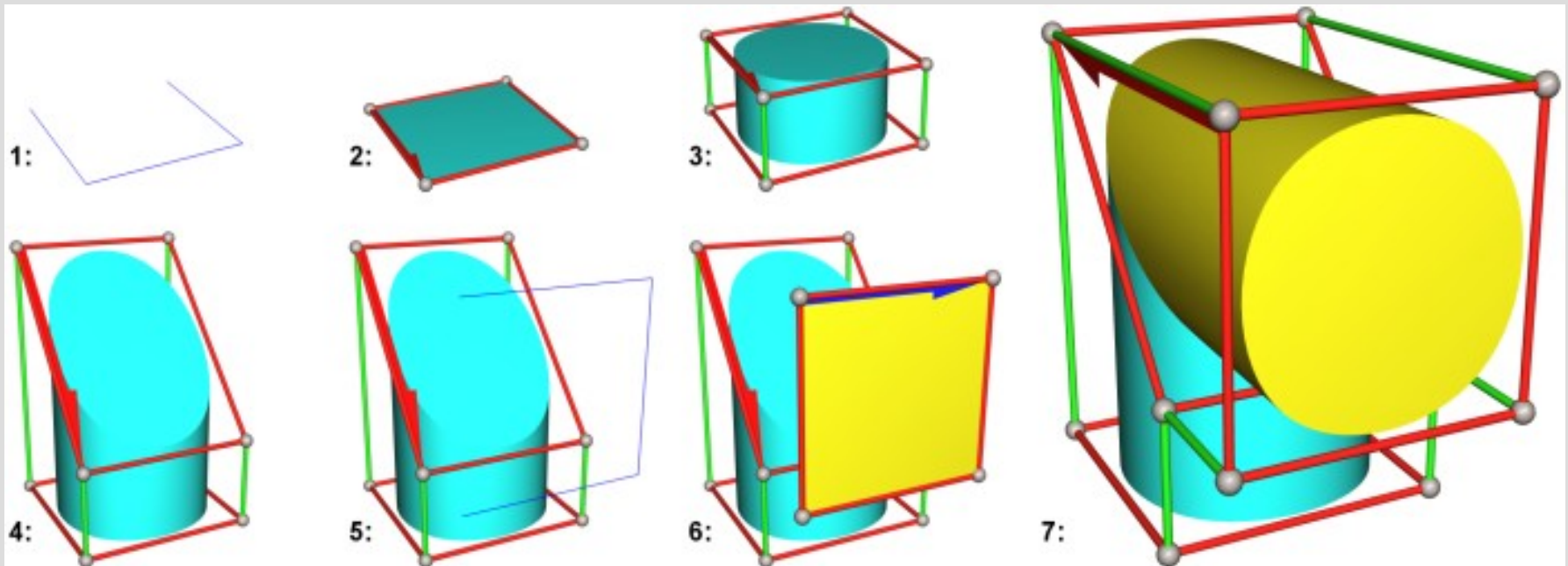


Generative Modellierung

- Modell wird nicht über Ansammlung geometrischer Primitive beschrieben, sondern durch Funktionen
 - bereits erstellte Konstruktionen können leicht verändert wieder verwendet werden
- Verwendete Programmiersprache: „Generative Modelling Language“

Generative Modellierung

- 1: (0,0,-2) (1,1,0) 2 quad
- 2: /cyan setcurrentmaterial 5 poly2doubleface
- 3: (0,1,1) extrude
- 4: (0,0,1) (1,0,1) normalize 0 project_ringplane
- 5: (2,0,0) (0,1,-1) 2 quad
- 6: /yellow setcurrentmaterial 5 poly2doubleface
- 7: 0 bridgerings

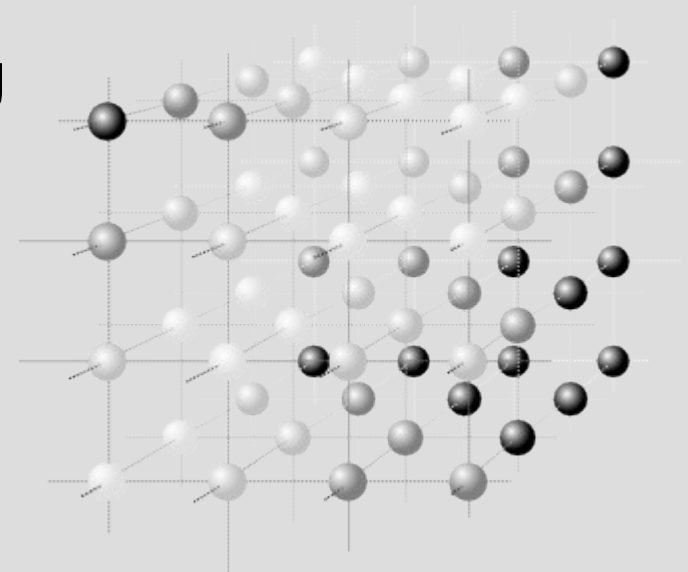


Generative Modellierung



Voxelgitter

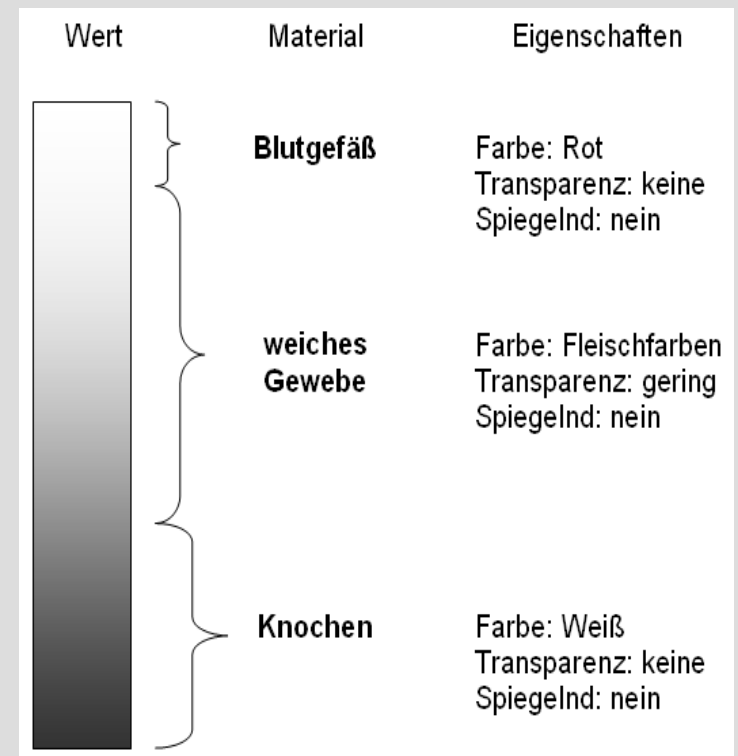
- räumlich gitterförmig angeordnete Werte, die die „Dichte“ eines Objektes beschreiben
 - Möglichkeit, Teile eines Objekts „wegzuschneiden“, um ins Innere zu sehen
 - CSG-Operationen problemlos möglich
 - Modellierung von teilweise durchsichtigen Objekten wie Wolken und Flüssigkeiten
- Beim Rendern Verzicht auf Streuung aufgrund zu großer Komplexität
- Rendern erfordert 4 Schritte:



Voxelgitter

1. Klassifikation:

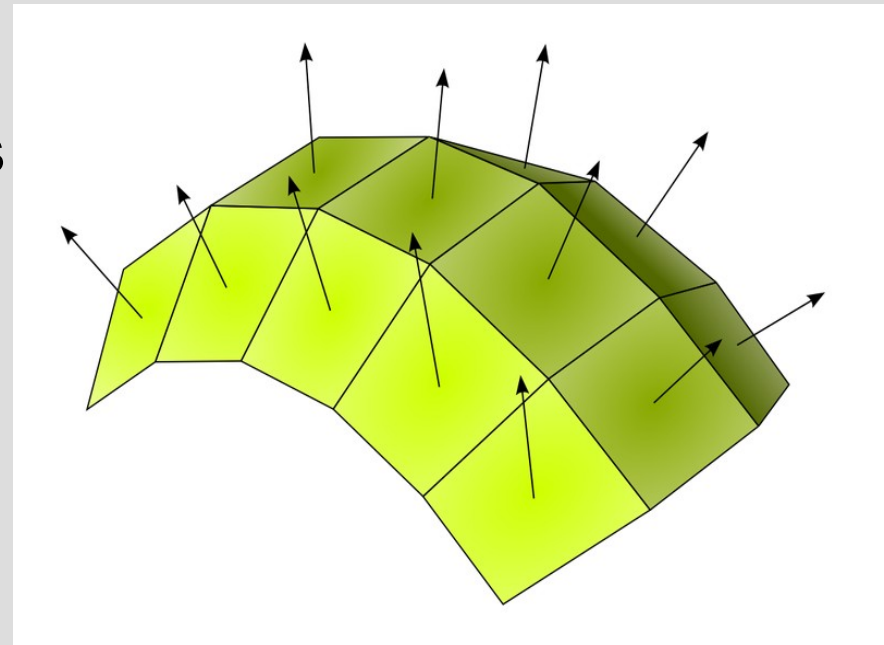
- Zuweisung von Materialeigenschaften
- bei Erzeugung von Voxeln nur ein Wert gelesen (z.B. Röntgendichte bei CT)
- liefert kaum Informationen über Material (Farbe, Reflexion, ...)
- Zuweisung von mehreren Werten nach Benutzervorgaben



Voxelgitter

2. Shading

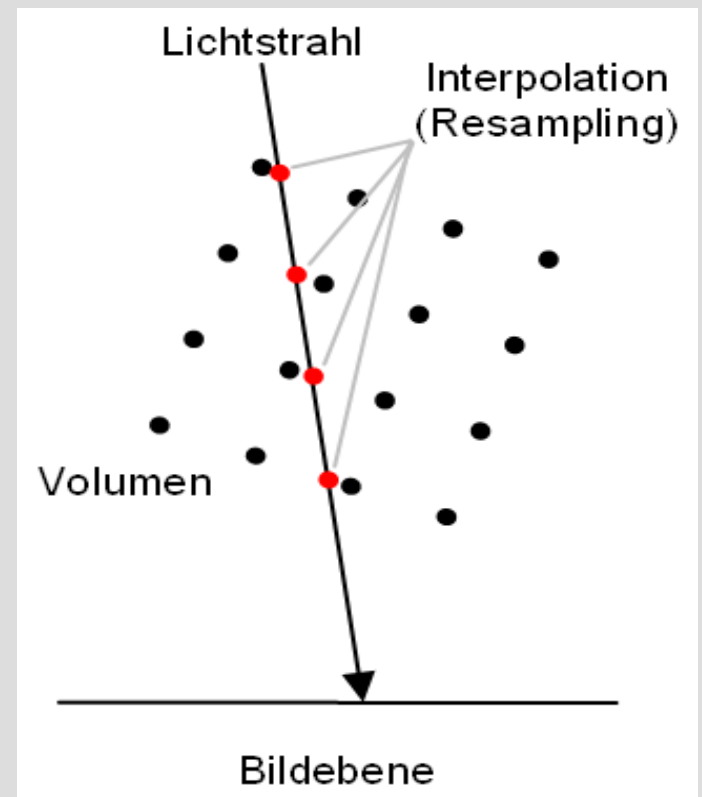
- üblicherweise Phong-Shading
 - Normale benötigt
 - keine Oberflächen → keine Normalen vorhanden → Gradient des Voxels genutzt
- Gradient zeigt in Richtung der stärksten Materialänderung (physikalischer Gedanke, dass Reflexion nur am Übergang zwischen verschiedenen Materialschichten erfolgt)



Voxelgitter

3. Interpolation

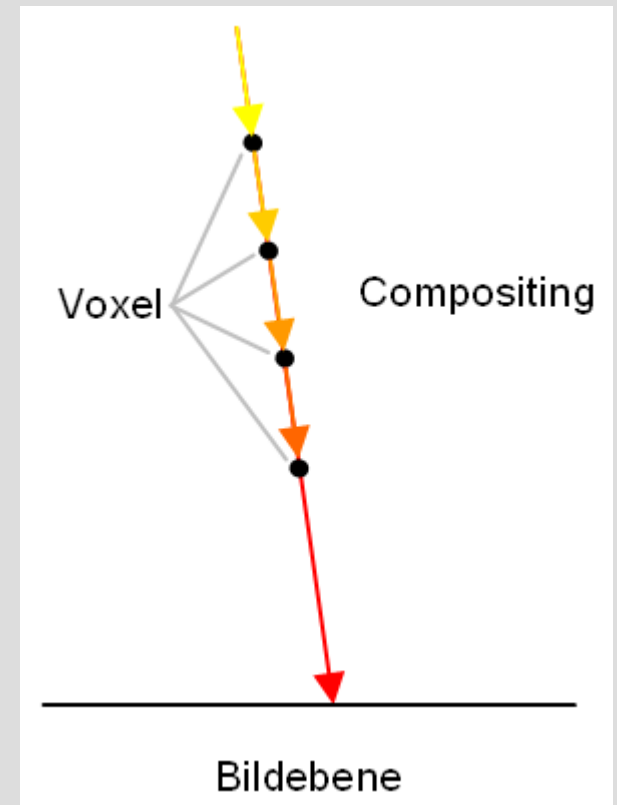
- Materialeigenschaften an Punkten zwischen den Voxeln aus den umgebenden Voxeln angenähert
- Voxel sind nulldimensional → unwahrscheinlich, dass ein das Volumen durchlaufender Sichtstrahl ein Voxel direkt trifft
- Festlegung, wie man aus Voxeln Informationen über Zwischenraum gewinnt
→ lineare Interpolation



Voxelgitter

4. Compositing

- von in einer Reihe liegenden Voxeln gelieferte Lichtbeiträge werden miteinander verrechnet
- Licht durchquert ein Voxel und ändert dabei Intensität und Farbe, durchquert nächstes Voxel und ändert wiederum Intensität und Farbe, → fällt auf Bildebene und färbt dort ein Pixel ein

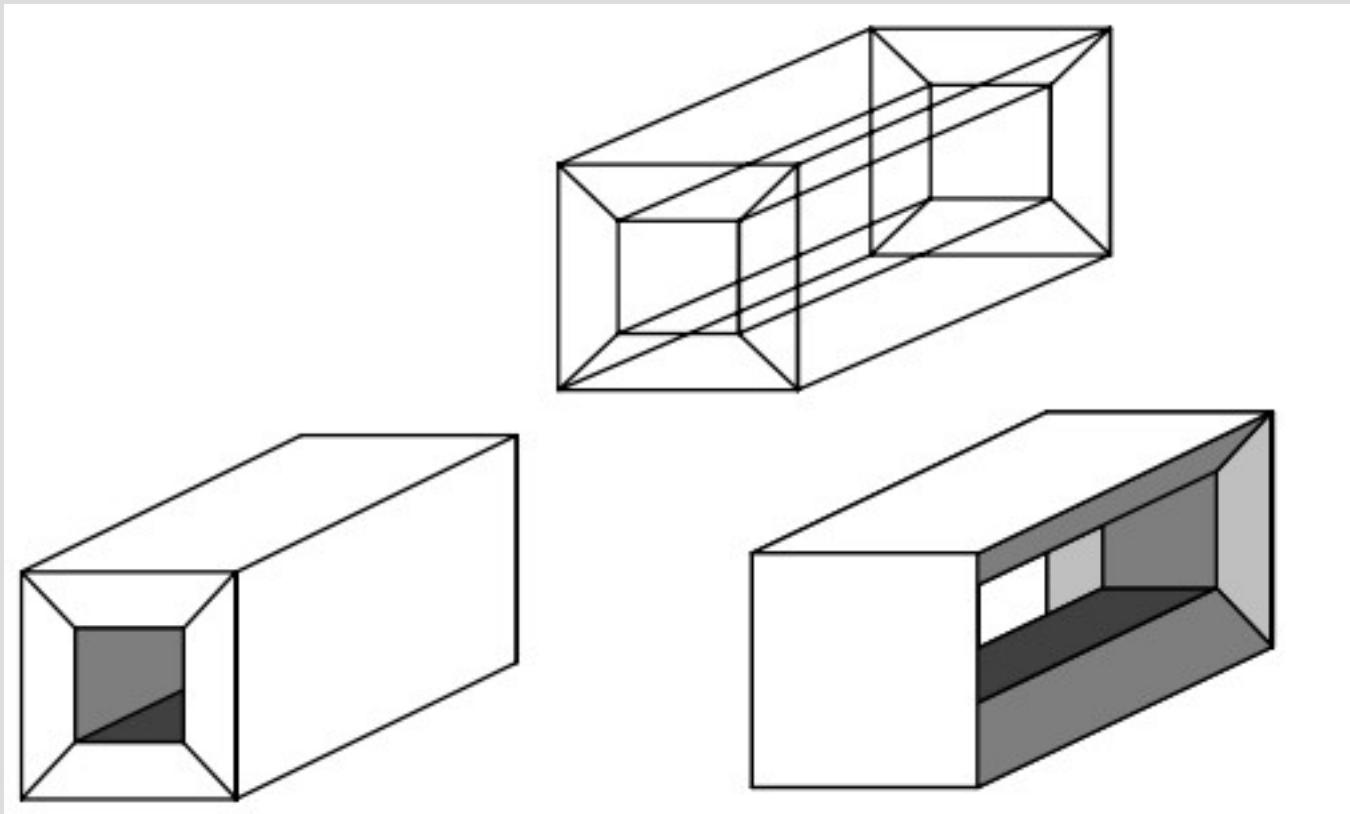


Voxelgitter



Drahtgittermodell

- Beschreibt Körper lediglich über Kanten
 - Bietet Geschwindigkeitsvorteile
 - Problem: potenziell mehrdeutig
- wird meist für vorläufige Ansicht genutzt, vor dem Rendern



Boundary Representation

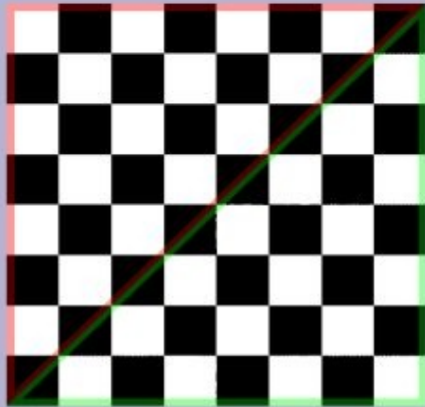
- Beschreibung eines Objekts anhand der Oberfläche
 - Flächen, Kanten und Vertices werden definiert
- meist über Polygonnetze
- Vorteile:
 - effizientes Rendern allgemeiner Oberflächen möglich
 - lokale Änderungen am Modell möglich

Texture Mapping

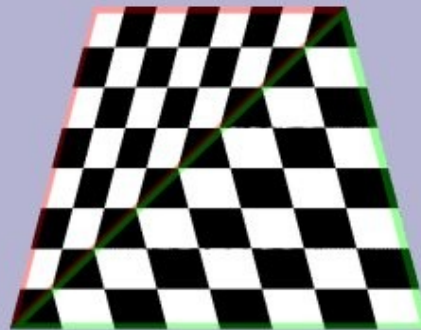
- Verfahren, um zweidimensionale Bilder auf Objekte zu projizieren
- jedem Vertex des Objekts werden zusätzlich zu den xyz-Koordinaten (Position im Raum) uvw-Koordinaten (Entsprechungen auf der Textur) zugeordnet
- Textur zweidimensional $\rightarrow w=0$
- Zuordnung erfolgt meist durch Interpolation

Texture Mapping

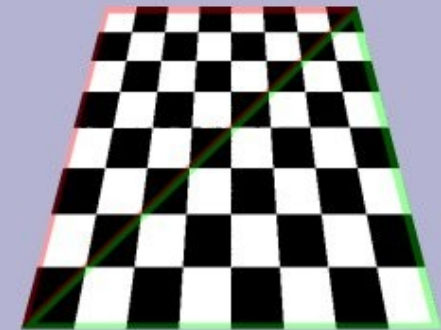
- bei ungünstiger Perspektive: optische Verzerrung der Textur
→ statt u und v werden u/z und v/z interpoliert (z gibt die Entfernung des Punktes von der Kamera an)



Flat



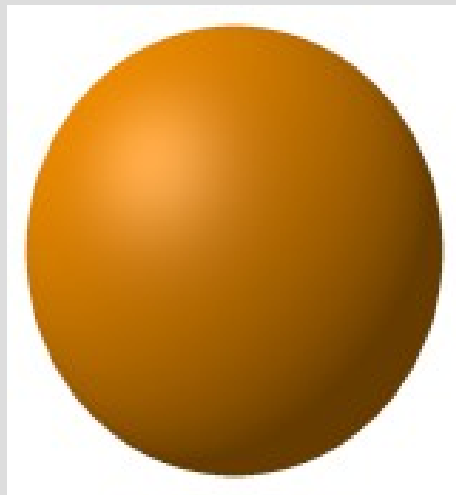
Affine



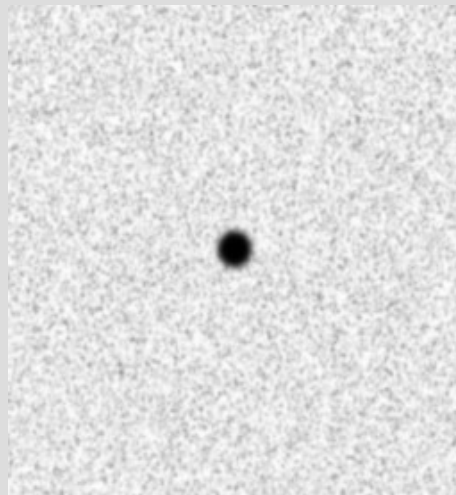
Correct

Bump Mapping

- Erhöhung des Detailreichtums von Objekten, ohne die Geometrie zu verändern
- zu jedem Pixel wird eine Änderung des Normalenvektors in einer Height Map „nachgeschlagen“
- Bei frontaler Ansicht starke Verbesserung der Bildqualität und der Realismusgrades, die Oberfläche bleibt aber flach → problematisch z. B. in der Reliefansicht
- am häufigsten genutzte Technik: Normal Mapping



+



=

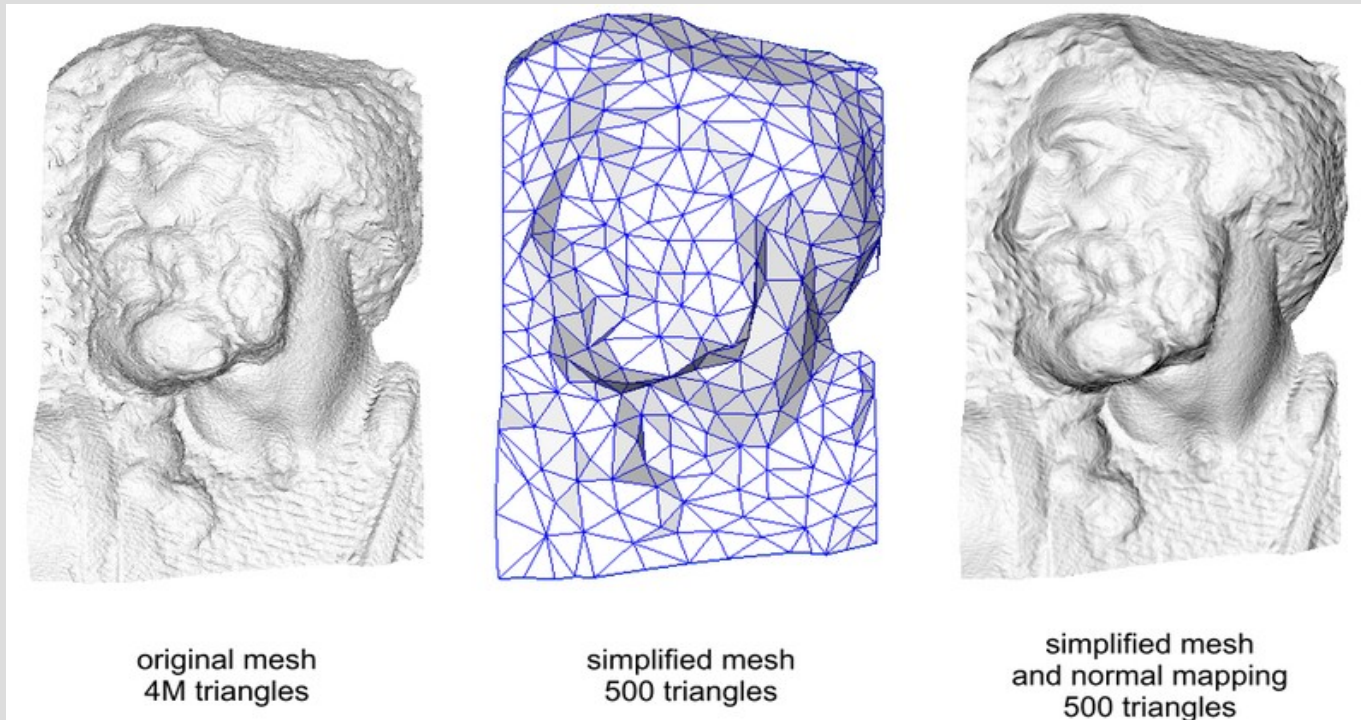


Normal Mapping

- Verfahren:
 - 2 Versionen eines Modells erstellt: eine hochauflösende, eine undetaillierte
 - Programm berechnet den Unterschied zwischen beiden Modellen, erstellt aus gewonnenen Daten eine Textur
 - Textur auf undetailliertes Modell gespannt → Eindruck entsteht, dass das detaillierte Modell verwendet wird
 - Erhöhung des Detailreichtums ohne starke Auswirkungen auf die Performance

Normal Mapping

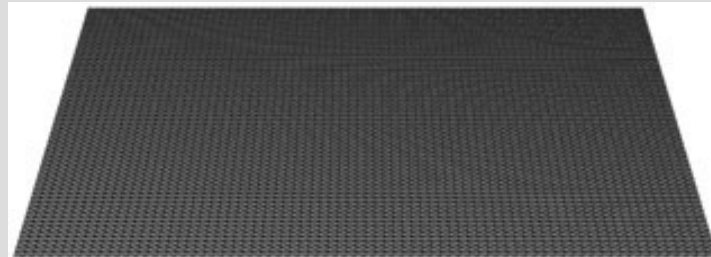
- Niedriges Detail lediglich an Silhouette sichtbar



Displacement Mapping

- Erhöhung des Detailreichtums von Objekten, wobei die Geometrie verändert wird
- Verfahren:
 - Außer eigentlicher Textur des Objektes wird Height map erstellt (meist aus der Farbtextur)
 - Height map besteht aus Graustufen, die zur Angabe von Höheninformationen genutzt werden
 - Vertices des Objekts werden entsprechend der Farbwerte entlang ihrer Normalen verschoben
 - höherer Realismusgrad durch Veränderung der Geometrie (vor allem hinsichtlich Relief und Schatten)

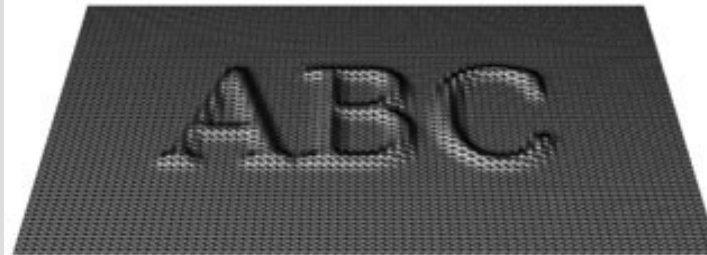
Displacement Mapping



ORIGINAL MESH



DISPLACEMENT MAP



MESH WITH DISPLACEMENT

Ende

Vielen Dank für Ihre
Aufmerksamkeit!

Quellen

- <http://de.wikipedia.org/>
- <http://en.wikipedia.org/>
- http://www-dept.cs.ucl.ac.uk/staff/A.Steed/book_tmp/CGVE/chapter_18.htm
- <http://www.digibib.tu-bs.de/?docid=00000008>
- Vorlesungsscript Computergraphik I WS 07/08
- Skript Technische Visualisierung Uni Hanover