



LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Dipl.Inf. Otmar Hilliges

Programmierpraktikum 3D Computer Grafik

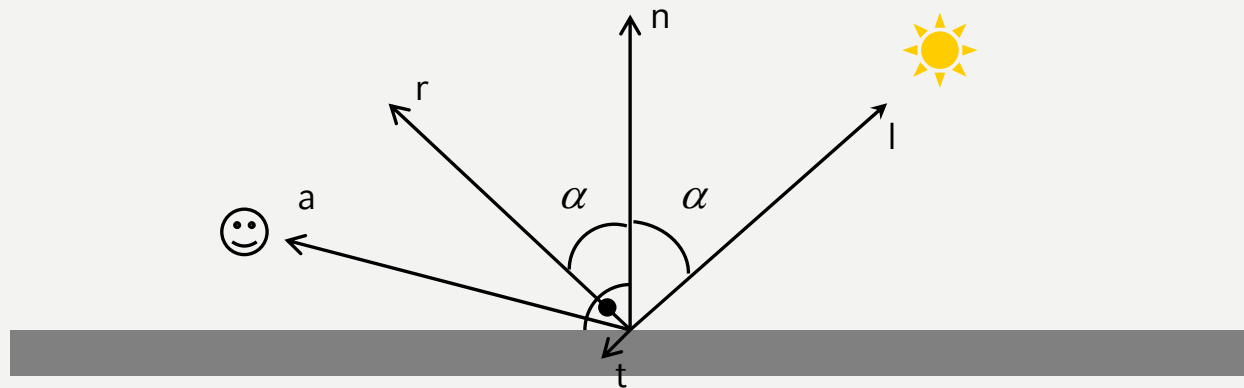
Szenegraphen, Texturen und Displaylisten.



- Beleuchtungsmodelle in OpenGL
- Bump-Maps zur Erzeugung von Reliefartigen Oberflächen
- Height-Maps zur Erzeugung von Landschaften

■ Mathematisch komplexe Operation in die verschiedene Größen Einfließen

- Unterschiedliche Lichtquellen
- Licht und Materialeigenschaften
- Art der Approximation der tatsächlichen physikalischen Vorgänge
- Vereinfachtes Modell:





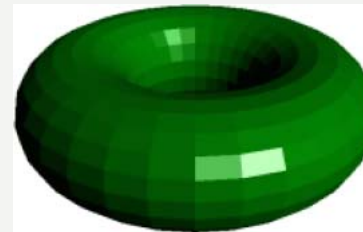
Shading:

- Einfache Techniken: interpolative Verfahren
- Über die Beleuchtungsgleichung lassen sich die Farbwerte der Eckpunkte berechnen
- Die Farbwerte aller Punkte ergeben sich dann aus der Interpolation der Farbwerte an den Eckpunkten
- Drei Verfahren:
 - Flat-Shading
 - Gouraud-Shading
 - Phong-Shading



Flat-Shading:

- Keine Interpolation
- Jedes Polygon hat genau einen Farbwert (z.B. Mittelwert der Eckpunkte)
- Einzelne Polygone eines Objekts sichtbar
- Leichte Implementierung aber sehr geringe Qualität



Gouraud-Shading:

- Berechne Helligkeitswerte an den Eckpunkten eines Polygons und interpoliere diese über die Fläche des Polygons
- Bessere Qualität, aber aufwendiger
- Keine Kanten mehr sichtbar
- Glanzpunkte werden nicht dargestellt



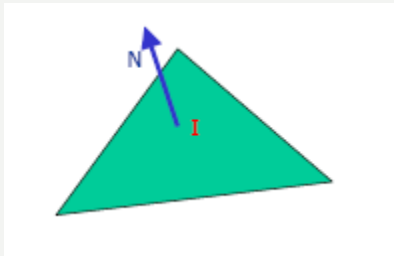


Phong-Shading:

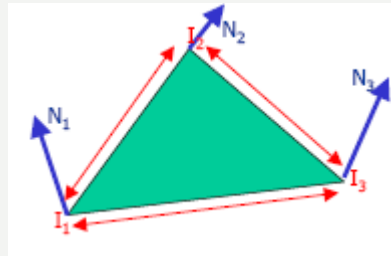
- Interpoliere die Normalenvektoren der Eckpunkte über die Fläche berechne an jedem Punkt den Farbwert aus den interpolierten Normalen
- Aufwendiger als Gouraud-Shading
- Glanzpunkte werden korrekt dargestellt



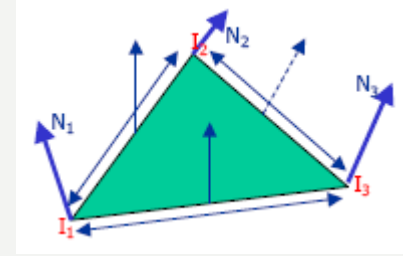
Shading (Zusammenfassung):



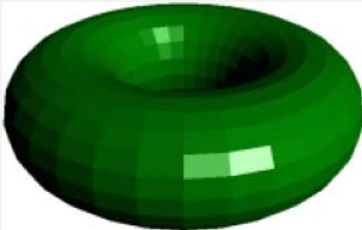
Flat-Shading

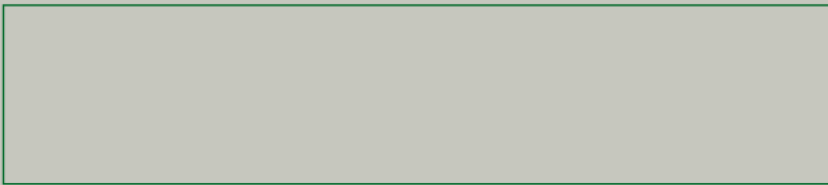


Gouraud-Shading



Phong-Shading





Bump Mapping



Grundlagen:

- Technik, die Objekten mehr Details gibt, ohne dabei mehr Dreiecke zu verwenden
- Bump-Map ist eine weitere Textur, deren Helligkeitswerte entscheidend für die Relief-Stärke ist
- Relief wäre zwar ohne Bump-Map möglich, die Lichtposition wird aber nicht berücksichtigt
- Schattenwürfe des Reliefs werden dynamisch berechnet

Mögliche Anwendungsbeispiele:

- Kratzer in einem Blech
- Rauhe Mauerstrukturen
- Struktur einer Wasserflasche

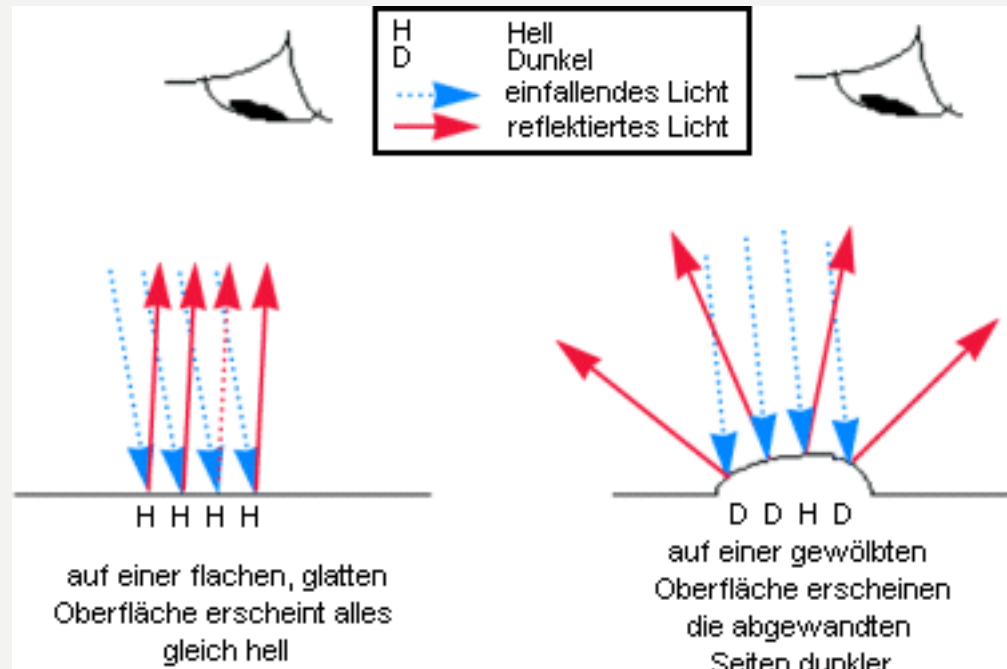


Textur der Erde



Relief-Textur der Erde

Verfahren:





Vorteile:

- Verwendet nicht mehr Polygone → Schnell
- Leicht zu implementieren → zweite Textur

Nachteile:

- Funktioniert nur bei einer direkten Draufsicht → Lösung hierfür:
Displacement-Mapping
- Verändert die Polygonstruktur nicht, sondern nur die Helligkeitswerte



OpenGL Extensions:

- Bump-Mapping ist ein Multi-Texture Verfahren
- OpenGL 1.1 unterstützt dies nicht
- Daher:
 - Verwendung der Extensions
 - Einbinden von `gl_ext.h` **oder**
 - Verwendung von GLEW (Open**GL** Extension **W**rangler), zu finden unter:
<http://glew.sourceforge.net/>



OpenGL Extensions (Tutorials):

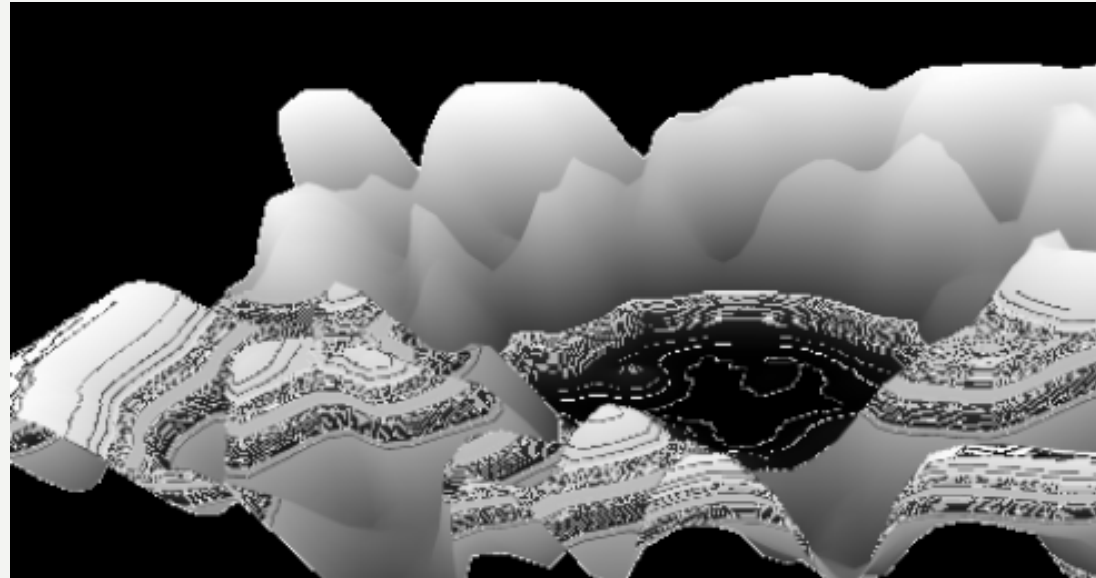
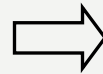
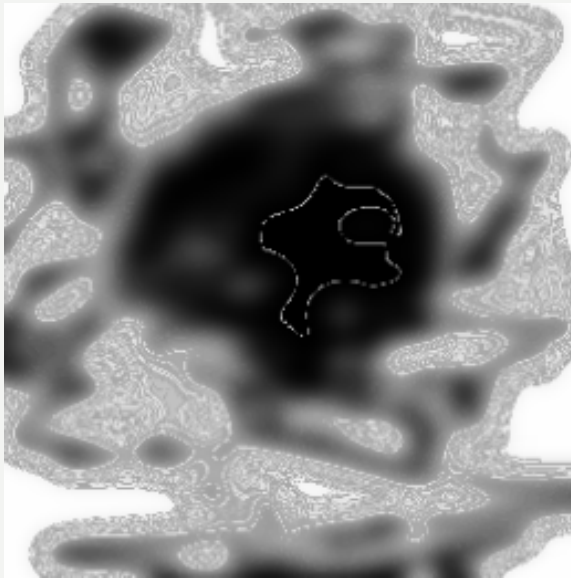
- NeHe Productions: OpenGL Lesson #22:
<http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=22>
- CodeSampler.com – OpenGL – Page 4:
http://www.codesampler.com/oglsrc/oglsrc_4.htm
- Simple Bump Mapping:
<http://www.paulsprojects.net/tutorials/simplebump/simplebump.html>



Height-Maps



- Einfaches Verfahren um „realistische“ Landschaften zu erzeugen.
- Höheninformation wird aus einer Textur (Height-Map) ausgelesen.
- Aus den Farb/Grauwerten lässt sich ein Höhenprofil triangulieren





Beispiel Code zur Triangulation:

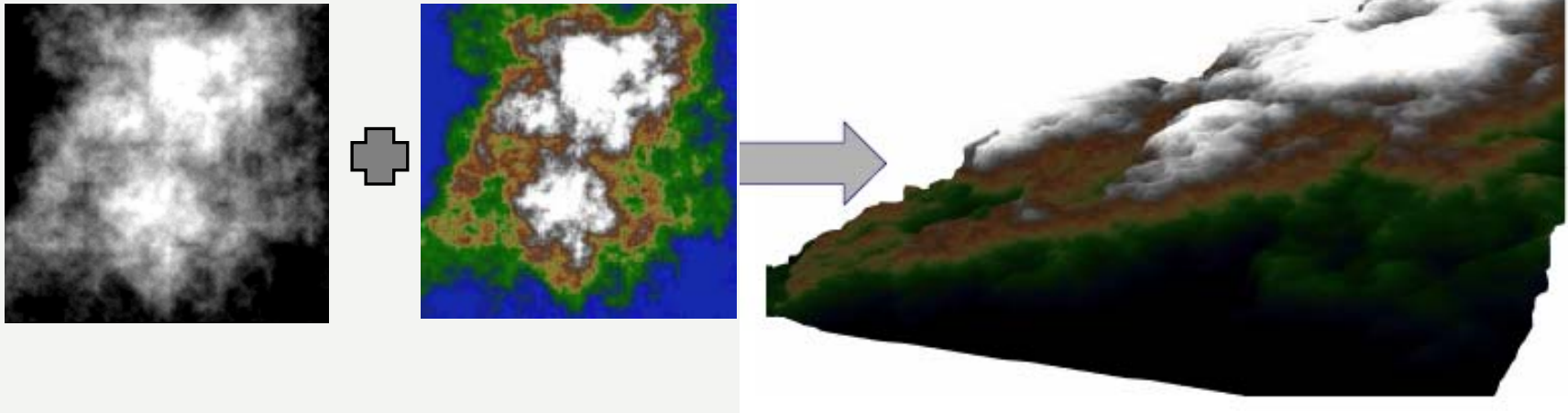
```
int x_max = myheightmap.getWidth();
int z_max = myheightmap.getHeight();           //Dimension der Landschaft
int stepsize = 10;                             //Schrittgröße
int scalefactor = 10;                           //Skalierung der
                                                //Normalisierten Farbwerte

for(int z=0;z<z_max-10; z+=10){
    glBegin(GL_TRIANGLE_STRIP);
        for(int x=0; x<x_max-10; x+=10){
            float color_y = myheightmap.getPixelValue(x,z);
            glColor3f(color_y/256,color_y/256,color_y/256);
            glVertex3f(x/scalefactor,color_y/scalefactor,-
                z/scalefactor);
            color_y = myheightmap.getPixelValue(x,z+stepsize);
            glColor3f(color_y/256,color_y/256,color_y/256);
            glVertex3f(x/scalefactor,color_y/scalefactor,-
                (z+stepsize)/scalefactor);
        }
    glEnd();
}
```

■ Wie wird die Landschaft farbig?

- Erstellen einer passenden Farbtextur zur Höheninformation
- (Es gibt freie Graphikprogramme die beide erzeugen)
- Beim berechnen des Höhenprofils Texturkoordinaten mit berechnen.

$$\text{glTexCoord 2f} \left(\frac{x}{x_max}, \frac{z}{z_max} \right)$$



- http://www.codeworx.org/opengl_tut34.php