

- Zusammen:

```

unsigned int tex[N];
glGenTextures( N, tex );
glBindTexture( GL_TEXTURE_2D, tex[0] );
pixels = loadImage(...);
glTexImage2D( GL_TEXTURE2D,
              0,          // mipmap level
              3,          // components [1,2,3,4]
              width, height, border,
              format,     // of the pixel data (GL_RGB..)
              type,       // GL_FLOAT...
              pixels );   // the data
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP );
...
glBindTexture( GL_TEXTURE_2D, tex[1] );
pixels = loadImage(...);
glTexImage2D( GL_TEXTURE2D, ...);

```

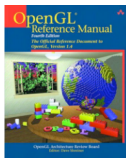
```

// 1-tes Objekt
glBindTexture( GL_TEXTURE_2D, tex[0] );
glBegin( GL_... )
    glVertex2f(...);
    glNormal3f(...);
    glVertex3f(...);
    ...
glEnd();
// 2-tes Objekt
glBindTexture( GL_TEXTURE_2D, tex[1] );
glBegin( GL_... )
    glVertex2f(...);
    glNormal3f(...);
    glVertex3f(...);
    ...
glEnd();

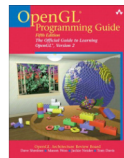
```

Zum Nachlesen

- Texturierung an sich ist eine sehr mächtige (und etwas komplexe) Technik
- Texturierung in OpenGL ist – zwangsläufig – etwas komplexer als die meisten anderen Teile des APIs
- Besser vor einer Implementierung nochmals nachlesen



Auch als HTML auf der Homepage der CG-1-Vorlesung

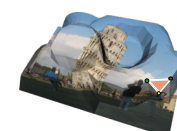
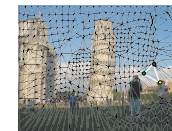
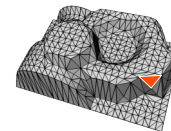


Man Pages

Oder im Netz unter <http://www.opengl.org/sdk/docs/man/>

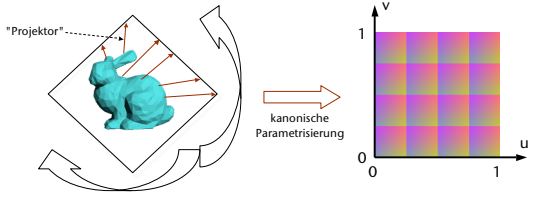
Parametrisierung

- Wie kommt man zu den Texturkoordinaten an jedem Vertex?
- Triviale Texturierung eines Terrains:
 - 3D-Koordinaten nach unten projizieren



- Achtung: dies ist nicht notwendig eine "gute" Texturierung!

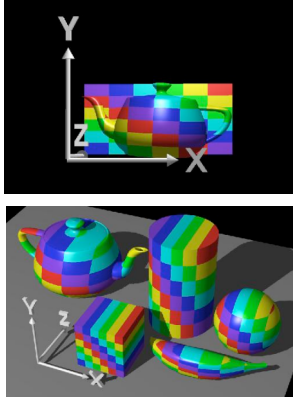
- Idee: 2-stufiger Prozess [Bier & Sloan, 1986]
 - Lege (konzeptionell) einen "kanonisch" parametrisierbaren Hüllkörper um das ganze Objekt
- 1. Projiziere Vertices (nicht notw. Vertex-Koord.!) auf diesen Hüllkörper
- 2. Verwende die Texturkoordinaten des projizierten Punktes auf dem Hüllkörper



G. Zachmann Computer-Graphik 2 - SS 07 Texturen 67

Einige Hüllkörper und deren Parametrisierung

- Ebene:
 - Projiziere Punkt (x,y,z) auf Ebene $\rightarrow (x,y)$
 - $(u,v) = (s_x x + t_x, s_y y + t_y)$
- Verallgemeinerung:
 - Definiere 2 beliebige Ebenen E_1 und E_2
 - $u := \text{dist}(P, E_1)$
 - $v := \text{dist}(P, E_2)$
 - Dieses Feature bietet OpenGL



G. Zachmann Computer-Graphik 2 - SS 07 Texturen 68

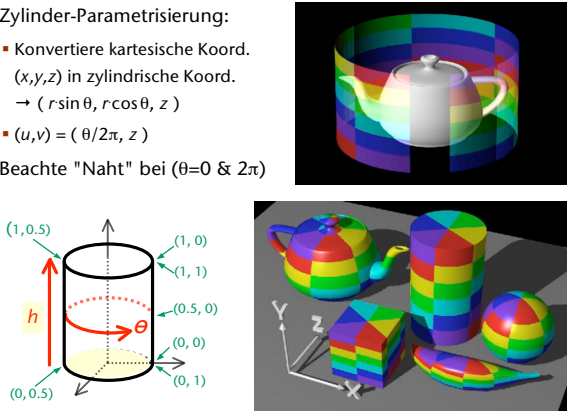
- Beispiel für Höhenlinien mittels dieser Technik:
 - 1D-Textur
 - Projiziere Punkt-Koord. auf eine Gerade \rightarrow u-Koord.
 - Abstand Punkt zu einer Ebene (senkrecht zu Gerade)



- Viele weitere ungewöhnliche Anwendungen von Texture-Mapping auf <http://www.graficaobscura.com/texmap/index.html>

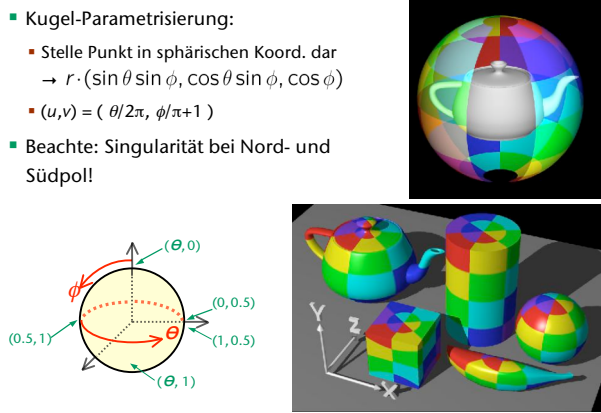
G. Zachmann Computer-Graphik 2 - SS 07 Texturen 69

- Zylinder-Parametrisierung:
 - Konvertiere kartesische Koord. (x,y,z) in zylindrische Koord. $\rightarrow (r \sin \theta, r \cos \theta, z)$
 - $(u,v) = (\theta/2\pi, z)$
 - Beachte "Naht" bei $(\theta=0 \text{ \& } 2\pi)$



G. Zachmann Computer-Graphik 2 - SS 07 Texturen 70

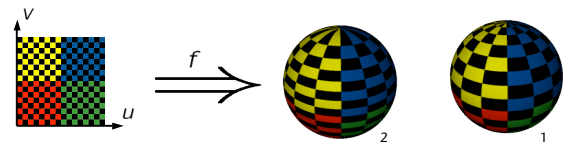
- Kugel-Parametrisierung:
 - Stelle Punkt in sphärischen Koord. dar
 $\rightarrow r \cdot (\sin \theta \sin \phi, \cos \theta \sin \phi, \cos \theta)$
 - $(u,v) = (\theta/2\pi, \phi/\pi+1)$
- Beachte: Singularität bei Nord- und Südpol!



G. Zachmann Computer-Graphik 2 - SS 07 Texturen 72

- Hier gibt es 2 Varianten:

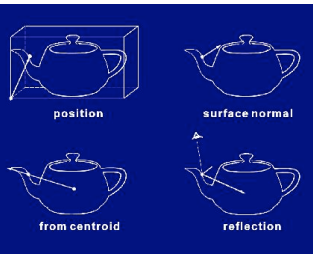
1. $f(u, v) = \begin{pmatrix} \sin(\pi v) \cos(2\pi u) \\ \sin(\pi v) \sin(2\pi u) \\ \cos(\pi v) \end{pmatrix}$
2. $f(u, v) = \begin{pmatrix} \sqrt{2v-1} \cos(2\pi u) \\ \sqrt{2v-1} \sin(2\pi u) \\ 2v-1 \end{pmatrix}$



G. Zachmann Computer-Graphik 2 - SS 07 Texturen 73

Was soll man projizieren?

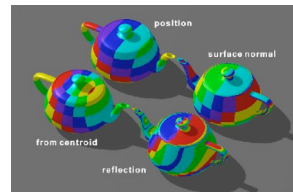
- Bisher: einfach die Koordinaten (x,y,z) des Vertex projiziert
- Verallgemeinerung: statt dessen kann man genauso gut (oder schlecht) andere Attribute des Vertex projizieren, z.B.
 - Normale
 - Vektor vom Zentrum des Objektes durch den Vertex
 - Reflektierter Viewing-Vektor
 - ...



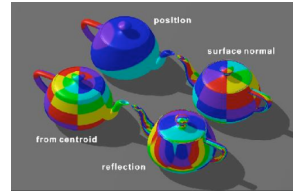
G. Zachmann Computer-Graphik 2 - SS 07 Texturen 74

- Beispiele:

planar



cylindrical



G. Zachmann Computer-Graphik 2 - SS 07 Texturen 75

- Parametrisierung parametrischer Flächen ist oft "kanonisch":

