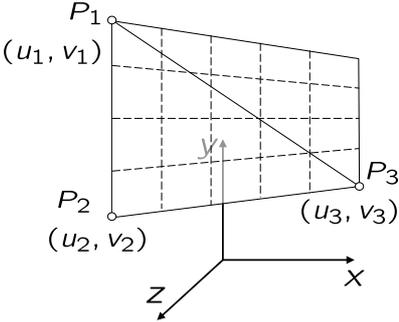


- Erinnerung: was passiert bei der perspektivischen **Projektion**

$$P_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} x_i \\ y_i \\ z_i \\ w_i \end{pmatrix} \equiv \begin{pmatrix} x_i/w_i \\ y_i/w_i \\ z_i/w_i \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} x_i/w_i \\ y_i/w_i \end{pmatrix} = \hat{P}_i$$

wobei $w_i = \frac{z_i}{z_0}$,
 $z_0 = \text{Proj.ebene}$

- Erinnerung: baryzentrische Koord. auf dem Rand des Dreiecks = lineare Interpolation zwischen den beiden Ecken

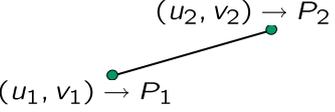


G. Zachmann Computer-Graphik 2 – SS 10 Texturen 24

- Betrachte im Folgenden nur die Interpolation auf einer Linie
- Gegeben: t , zur linearen Interpolation zwischen \hat{P}_1 und \hat{P}_2 , d.h.

$$\hat{P}(t) = t\hat{P}_1 + (1-t)\hat{P}_2$$
- Gesucht: Funktionen f_1, f_2 (möglichst ähnlich zu linearer Interpolation), so daß

$$\begin{pmatrix} u \\ v \end{pmatrix} (t) = f_1(t) \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} + f_2(t) \begin{pmatrix} u_2 \\ v_2 \end{pmatrix}$$
 die "richtigen" Texturkoordinaten für \hat{P} sind



G. Zachmann Computer-Graphik 2 – SS 10 Texturen 25

- Problem:

$$P(t) = tP_1 + (1 - t)P_2 \quad t \in [0, 1]$$

$$\hat{P}(s) = s\hat{P}_1 + (1 - s)\hat{P}_2 \quad s \in [0, 1], \hat{P}_i = \text{Proj}(P_i)$$

ergeben zwar dieselbe Gerade auf dem Bildschirm, wenn $P(t)$ projiziert wird, aber i.A. ist

$$\text{Proj}(P(t)) \neq \hat{P}(t) !$$
- Frage: wie sieht $\text{Proj}(P(t))$ aus?

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 26

- Gegeben:

$$P(t) = t \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} + (1 - t) \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$$
- O.B.d.A. betrachte nur die x-Koordinate:

$$x(t) = tx_2 + (1 - t)x_1 \mapsto \frac{tx_2 + (1 - t)x_1}{tw_2 + (1 - t)w_1}$$

wobei $w_i = \frac{z_i}{z_0}$
- Behauptung:

$$\frac{tx_2 + (1 - t)x_1}{tw_2 + (1 - t)w_1} = \frac{x_1}{w_1} + \frac{tw_2}{w_1 + t(w_2 - w_1)} \left(\frac{x_2}{w_2} - \frac{x_1}{w_1} \right)$$

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 27

▪ Beweis:

$$\frac{x_1}{w_1} + \frac{tw_2}{w_1 + t(w_2 - w_1)} \left(\frac{x_2}{w_2} - \frac{x_1}{w_1} \right) =$$

$$\frac{x_1(w_1 + t(w_2 - w_1)) + tw_2w_1 \left(\frac{x_2}{w_2} - \frac{x_1}{w_1} \right)}{w_1(w_1 + t(w_2 - w_1))} =$$

$$\frac{x_1w_1 + tw_2x_1 - tw_1x_1 + tw_1x_2 - tw_2x_1}{w_1^2 + tw_2w_1 - tw_1^2} =$$

$$\frac{x_1w_1 - tw_1x_1 + tw_1x_2}{w_1^2 + tw_2w_1 - tw_1^2} = \frac{x_1 - tx_1 + tx_2}{w_1 + tw_2 - tw_1} =$$

$$\frac{x_1 + t(x_2 - x_1)}{w_1 + t(w_2 - w_1)} = \frac{tx_2 + (1-t)x_1}{tw_2 + (1-t)w_1}$$

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 28

▪ Gegeben:

$$\hat{P}(s) = s \begin{pmatrix} \hat{x}_2 \\ \hat{y}_2 \end{pmatrix} + (1-s) \begin{pmatrix} \hat{x}_1 \\ \hat{y}_1 \end{pmatrix}$$

▪ Frage: welches t passt zu diesem s , d.h., für welches t ist

$$\text{Proj}(P(t)) = \hat{P}(s)$$

$$s\hat{x}_2 + (1-s)\hat{x}_1 = \frac{x_1}{w_1} + s \left(\frac{x_2}{w_2} - \frac{x_1}{w_1} \right)$$

$$\Rightarrow s = \frac{tw_2}{w_1 + t(w_2 - w_1)}$$

$$\Rightarrow t = \frac{sw_1}{w_2 + s(w_1 - w_2)}$$

▪ Mit diesem t kann man die Texturkoord. u, v linear interpolieren!

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 30



- Analog funktioniert es bei 3 baryzentrischen Koordinaten,
- Etwas anders hingeschrieben:

$$t(P) = \frac{\sum_{i=0}^2 \lambda_i(P) \frac{t_i}{w_i}}{\sum_{i=0}^2 \lambda_i(P) \frac{1}{w_i}}$$

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 31



Moment Mal!

- War alles im ersten Semester eine große Lüge?
- Was ist der Unterschied zwischen Interpolation von Farben und der Interpolation von Textur-Koordinaten?!
- Kein Unterschied ...
- Dann hätten wir Farben auch perspektivisch korrekt interpolieren müssen!
- Richtig. Sieht man aber nicht ...

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 32

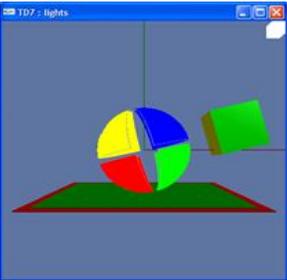
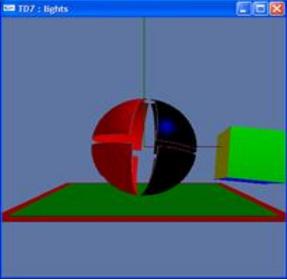
Beeinflussung der Beleuchtung durch Texturen

- Wie kann ein Texturwert die Beleuchtungsrechnung beeinflussen, was kann man mit einer Textur machen?
- Erinnerung: Blinn-Phong Modell

$$L_{\text{Phong}} = r_a L_a + \sum_j (r_d(\mathbf{n} \cdot \mathbf{l}_j) + r_s(\mathbf{n} \cdot \mathbf{h}_j)^m) L_j$$

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 33

- Ersetzen der Objektfarbe (*replace*)
 - Einfachste Art der Texturierung
 - Jegliche Beleuchtung wird entfernt
$$L_{\text{out}} = C_{\text{tex}}(u, v)$$
- A posteriori Skalierung der Farbe (*modulate*)
 - Häufigste Art der Texturierung
 - Komponentenweise Skalierung des Farbwertes
$$L_{\text{out}} = L_{\text{Phong}} \cdot C_{\text{tex}}(u, v)$$

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 34

3. A priori Skalierung der Materialfarbe

- Farbe des Objektes wird im wesentlichen durch r_a und r_d bestimmt

$$r_a = k_a \cdot C_{\text{tex}}(u, v) \quad r_d = k_d \cdot C_{\text{tex}}(u, v)$$

- Wichtig: im Unterschied zu 2 bleibt der spekulare Anteil von der Textur **unbeeinflusst**
- In OpenGL mittels `GL_SEPARATE_SPECULAR` erreichbar

4. Modulation der spekularen Reflexion (*gloss mapping*)

- Analog zu 3 für r_s

$$r_s = k_s \cdot C_{\text{tex}}(u, v)$$

- Erlaubt Modellierung unregelmäßiger "shininess" (z.B. verschmutzte Flächen)
- Geht nur mit Vertex- und Fragment-Shaders



G. Zachmann Computer-Graphik 2 – SS 10 Texturen 35

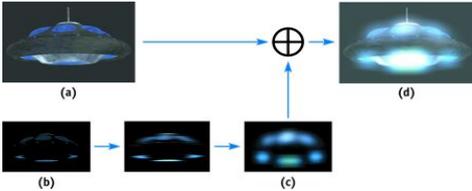
4.1 "Glow"-Effekt:

$$L_{\text{out}} = C_{\text{tex}}(u, v) + L_{\text{Phong}}$$

- For neon signs, TV, laser beams etc.



- Solche Effekte (Halos) gehen aber (vermutlich) nur mit Multi-Pass-Rendering

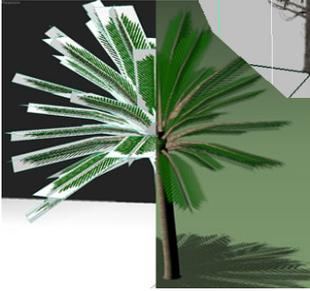


G. Zachmann Computer-Graphik 2 – SS 10 Texturen 36

5. Modulation der Transparenz

- Speichern der „Durchsichtigkeit“ in einer Textur:

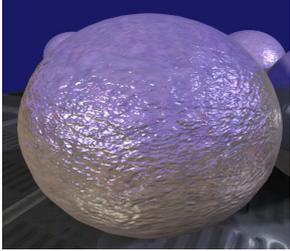
$$\alpha = \alpha_{\text{obj}} \cdot \alpha_{\text{tex}}(u, v)$$
- Pixel mit $\alpha=0$ sind voll durchsichtig und Pixel mit $\alpha=1$ sind voll undurchsichtig
- Ermöglicht komplexe Umrisse


G. Zachmann Computer-Graphik 2 – SS 10 Texturen 37

6. Perturbation der Normale (*Bump- / Normal-Mapping*)

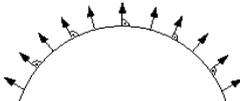
- Speichern von Höhenwerten oder Normalen einer Offsetfläche in einer Textur

$$\mathbf{n} = n_{\text{tex}}(u, v)$$


G. Zachmann Computer-Graphik 2 – SS 10 Texturen 38

Exkurs: Bump Mapping (ursprüngliche Idee)

- Ziel: eine "einfache" Geometrie mit hohem Detailgrad darstellen
- Idee:



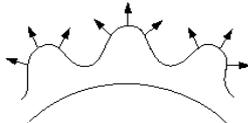
Original-Oberfläche $P(u, v)$
mit Normalen $N(u, v)$



Bump Map $F(u, v)$



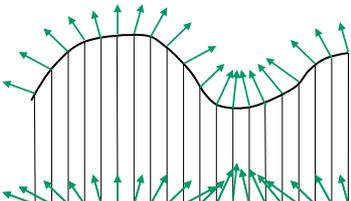
Offset-Oberfläche \hat{P}



Perturbierte Normalen $\hat{N}(u, v)$
- **Bump-Map** = Offset entlang der orig. Normale = skalare Textur
- Resultierende Oberfläche : $\hat{P}(u, v) = P(u, v) + F(u, v) \frac{N(u, v)}{\|N(u, v)\|}$

G. Zachmann Computer-Graphik 2 – SS 10
Texturen 39

- Beobachtung: ins Beleuchtungsmodell geht **nicht direkt** $P(u, v)$, sondern **nur** $N(u, v)$ ein.
- Hauptidee des **Bump-Mapping**: für kleine Unebenheiten genügt Visualisierung von $P(u, v)$ mit Normalen $\hat{N}(u, v)$



- Wie berechnet man $\hat{N}(u, v)$:

$$\hat{N}(u, v) = \hat{P}_u(u, v) \times \hat{P}_v(u, v)$$

G. Zachmann Computer-Graphik 2 – SS 10
Texturen 40

▪ Richtungsableitungen mit Summen- und Kettenregeln:

$$\hat{P}_u(u, v) = P_u(u, v) + F_u(u, v) \frac{N(u, v)}{\|N(u, v)\|} + F(u, v) \frac{d}{du} \frac{N(u, v)}{\|N(u, v)\|}$$

$$\hat{P}_v(u, v) = P_v(u, v) + F_v(u, v) \frac{N(u, v)}{\|N(u, v)\|} + F(u, v) \frac{d}{dv} \frac{N(u, v)}{\|N(u, v)\|}$$

▪ Falls $F(u, v)$ klein \rightarrow Weglassen des letzten Teilterms:

$$\hat{P}_u(u, v) \approx P_u(u, v) + F_u(u, v) \frac{N(u, v)}{\|N(u, v)\|}$$

$$\hat{P}_v(u, v) \approx P_v(u, v) + F_v(u, v) \frac{N(u, v)}{\|N(u, v)\|}$$

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 41

▪ Für $\hat{N}(u, v)$ folgt damit:

$$\begin{aligned} \hat{N} &= \hat{P}_u \times \hat{P}_v \\ &= P_u \times P_v + F_u \left(\frac{N}{\|N\|} \times P_v \right) + F_v \left(P_u \times \frac{N}{\|N\|} \right) + F_u F_v \left(\frac{N}{\|N\|} \times \frac{N}{\|N\|} \right) \\ &= P_u \times P_v + F_u \left(\frac{N}{\|N\|} \times P_v \right) + F_v \left(P_u \times \frac{N}{\|N\|} \right) \\ &= N + \frac{1}{\|N\|} (F_u(N \times P_v) - F_v(N \times P_u)) \end{aligned}$$

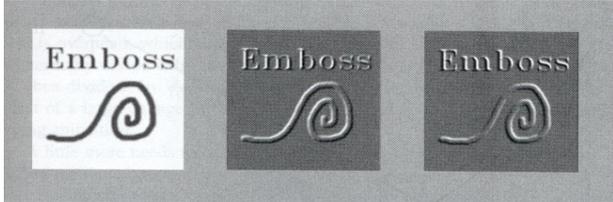
G. Zachmann Computer-Graphik 2 – SS 10 Texturen 42

Bemerkungen

- Die Ableitungen F_u und F_v können mit **finiten Differenzen** approximiert werden.
- Finite Differenzen auf uniformem Gitter der Gittergröße h (im 1D)
 - Vorwärtsdifferenzen $f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$
 - Rückwärtsdifferenzen $f'(x_i) = \frac{f(x_i) - f(x_{i-1}))}{h}$
 - Zentrale Differenzen $f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1}))}{2h}$

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 43

- Speicherung:
 - Höhenfeld als Grauwertbild in Rot-Kanal (z.B. mit Malprogramm erstellt)
 - Richtungsableitungen (mit finiten Differenzen berechnet) in G/B speichern

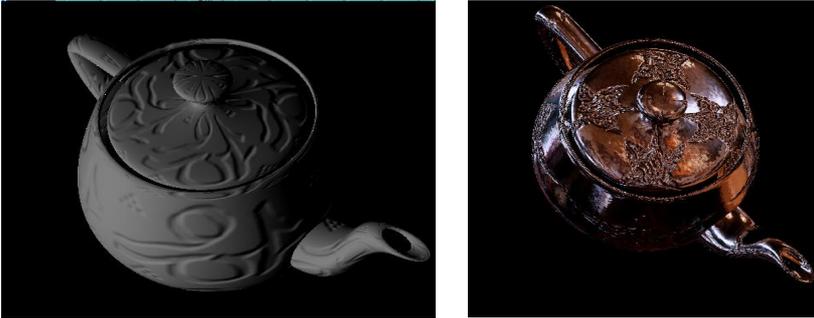
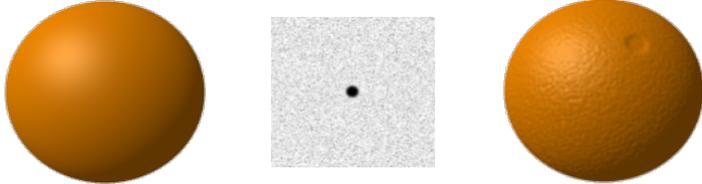


Original Höhenfeld u-Richtungsableitung v-Richtungsableitung

- Voraussetzung: **Beleuchtung erst bei der Rasterisierung**, oder sehr feintesselierte Geometrie und dann Berechnung der Normalen an jedem Vertex "von Hand"

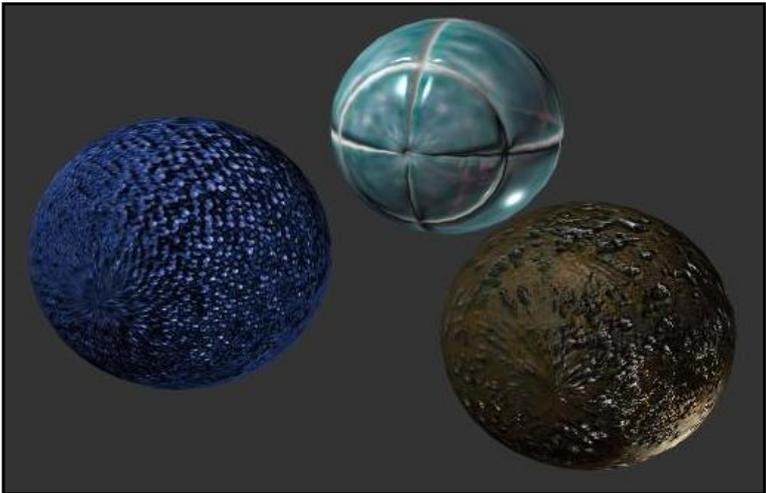
G. Zachmann Computer-Graphik 2 – SS 10 Texturen 44

Weitere Beispiele



Multi-Textures (Bump und Environment)

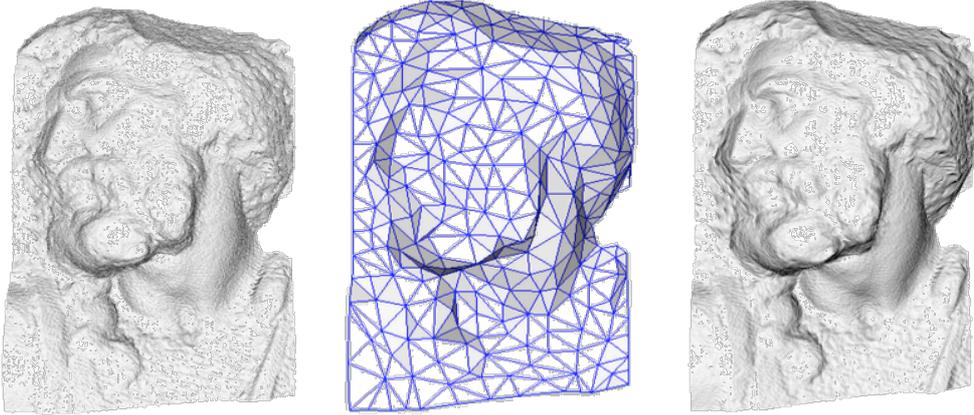
G. Zachmann Computer-Graphik 2 – SS 10 Texturen 45



G. Zachmann Computer-Graphik 2 – SS 10 Texturen 46

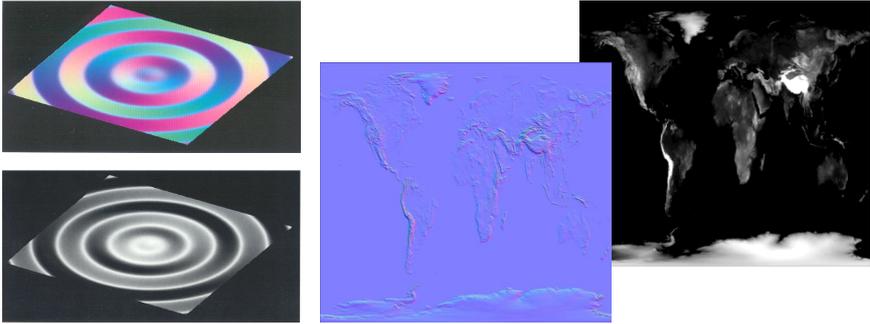
Normal Maps

- Normalen in hoher Auflösung in Textur speichern
- Für niedrig aufgelöste Geometrie

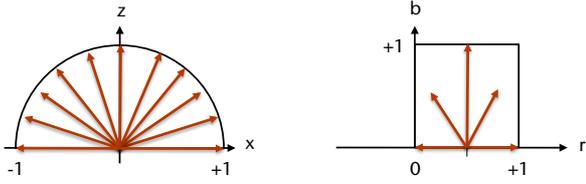


G. Zachmann Computer-Graphik 2 – SS 10 Texturen 47

Beispiele:



Kodierung der Normalen:



G. Zachmann Computer-Graphik 2 – SS 10 Texturen 48

- Unterschied zwischen Normal Maps und Bump Maps:
 - Bump Maps sind unabhängig von der Geometrie, man kann sie auf jede beliebige (genügend "flache") Geometrie aufbringen.
 - Normal Maps kann man nur für genau die Geometrie verwenden, für die sie erzeugt wurden.

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 49

Exkurs: Trompe l'œil - Malerei



Pere Borrell del Caso,
Der Kritik entfliehend, 1874

G. Zachmann Computer-Graphik 2 – SS 10 Texturen 50



G. Zachmann Computer-Graph

Universitätskirche in Wien mit trompe-l'oeil Deckenfresken, die den Eindruck einer Kuppel geben. Gemalt von Andrea Pozzo im 17. Jahrhundert

Texturen 51