

Lokale Beleuchtungsrechnung und Texturen

Einführung in die Computergrafik

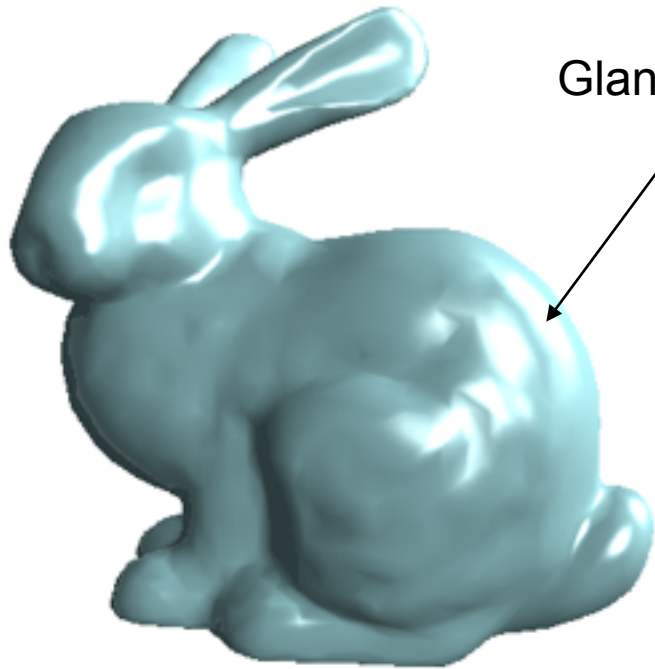
Wiederholung

- Einführung
- Normalen
- Volumen
- Glättung
- Nachbarschafts-Datenstrukturen
- Triangulierung von Polygonen
- Subdivision
- Vereinfachung
- Zusammenfassung

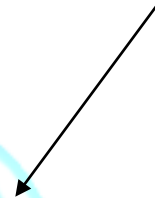
Ausblick



Worum gehts?



Glanzpunkte



Texturen



Agenda

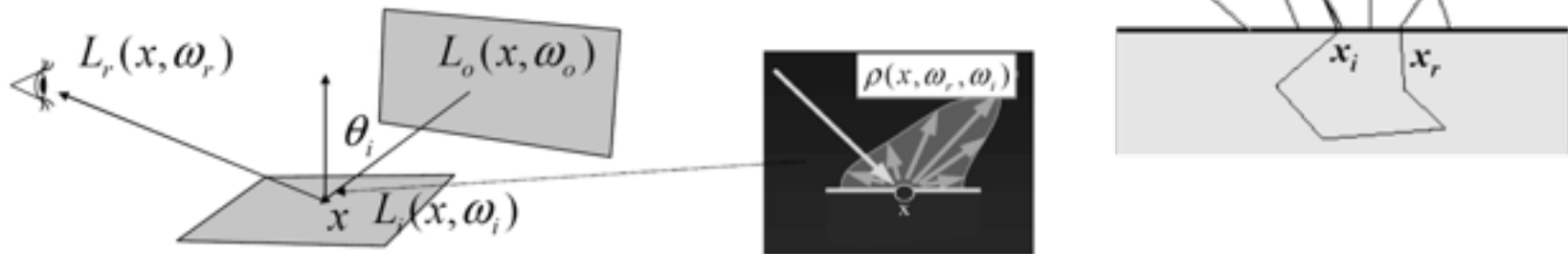
- Einführung
- Lokale Beleuchtungsmodelle
- Shading
- Texturen



Phong Beleuchtungsmodell

Einführung

- Strahlungsaustausch zwischen Oberflächen
- sehr komplexes Verhalten: Streuung und Reflexion
- Rendering-Gleichung beschreibt Verhalten



$$L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\Omega_i} \rho(x, \omega_r, \omega_i) L_i(x, \omega_i) \cos(\theta_i) d\omega_i$$

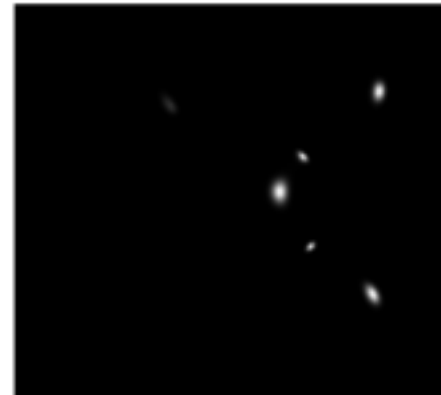
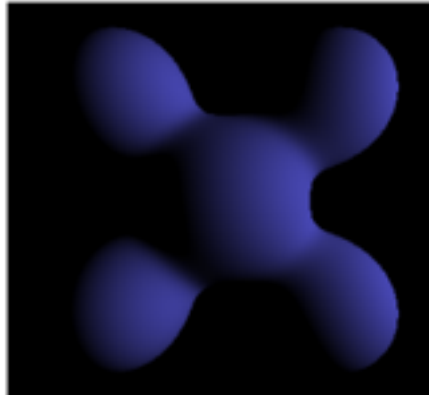
- nicht bei interaktiven Bildwiederholraten berechenbar!

Lokale Beleuchtungsmodelle

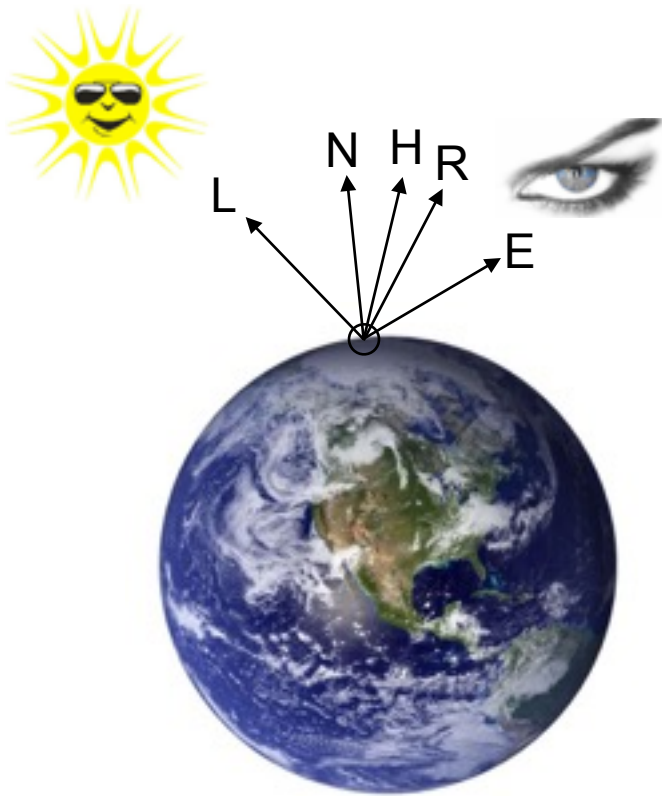
- beschreiben die in einem Punkt der Objektoberfläche wahrnehmbare Leuchtdichte
 - Auswerten des Zusammenwirkens von einfallendem Licht aus den Lichtquellen und dem Reflexionsverhalten der Objektflächen
 - sekundäre Effekte, wie der Strahlungsaustausch benachbarter Flächen, werden i.a. nicht berücksichtigt
- Approximation der physikalische Gesetzmäßigkeiten
 - berechnete Beleuchtung möglichst realistisch (Photorealismus)
 - bei minimalem Rechenaufwand

Phong Beleuchtungsmodell

- am weitesten verbreitetes lokales Beleuchtungsmodell
- besteht aus drei Komponenten
 - ambient
 - diffus
 - spekulär



Lokale Beleuchtungsmodelle



- Richtung zur Lichtquelle: L
- Richtung zum Beobachter: E
- Normale am Oberflächenpunkt x: N
- ideal reflektierte Lichtrichtung: R
- Bisektor („Halbrichtung“) zwischen E und L: H
- alle Vektoren sind normiert

Ambientes Licht

- nicht physikalisch
- dennoch in allen lokalen Beleuchtungsverfahren vorhanden
- vom Modell nicht erfasste, indirekte Beleuchtungen
- damit werden nicht direkt beleuchtete Szenenteile sichtbar
- auf allen Flächen der Szene mit gleicher Stärke

Ambientes Licht

$$L_{amb} = r_a \cdot L_a$$

Reflexionsfaktor

Leuchtdichte (bei uns: Farbe)



Ambiente
Beleuchtung
(Quelle: [1])

Diffuses Licht

- Licht wird gleichmäßig in alle Richtungen reflektiert
- Beispiel: weißes Blatt Papier
- Stärke abhängig vom Einfallswinkel
 - maximal: Licht fällt senkrecht auf die Oberfläche
 - minimal: Licht fällt unter 90-Grad-Winkel auf die Oberfläche



Diffuses Licht

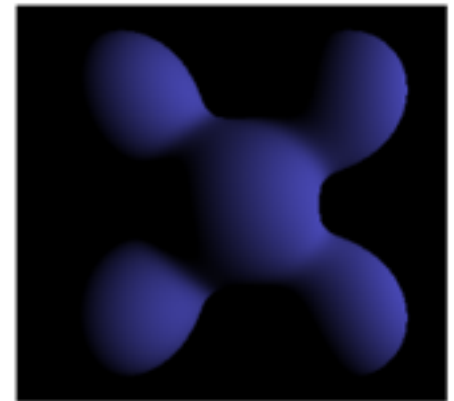
Winkel zwischen
Flächennormale und
Vektor zur
Lichtquelle: θ

$$L_{diff} = \begin{cases} r_d \cdot L_d \cdot \cos \theta = r_d \cdot L_d \cdot (\vec{N} \cdot \vec{L}), & \text{falls } |\theta| < \frac{\pi}{2} \text{ bzw. } (\vec{N} \cdot \vec{L}) > 0 \\ 0 & \text{sonst} \end{cases}$$

diffus
reflektierte
Leuchtdichte

diffuser
Reflektions-
grad

Maß für die von einer
Lichtquelle bei senkrechter
Einstrahlung erzeugte
Bestrahlungsstärke



Diffuse
Beleuchtung
(Quelle: [1])

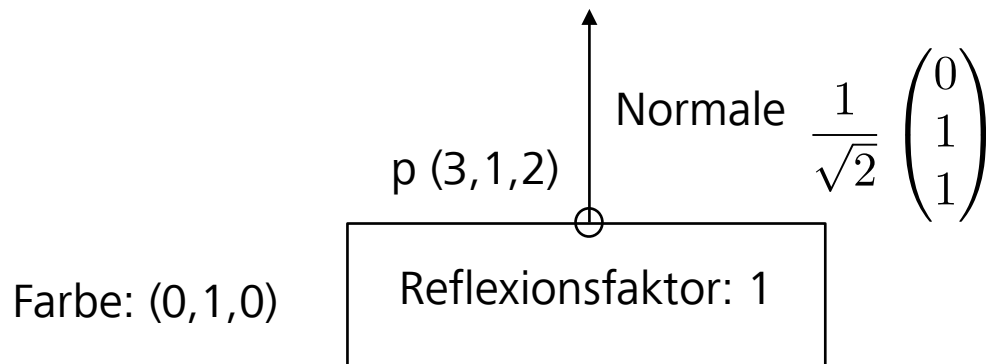
Übung: Diffuses Licht

- Berechnen Sie die Leuchtdichte als Farbe, die in folgender Szene von Lichtquelle 1 zum Beobachter an der Position p reflektiert wird

$$L_{diff} = \begin{cases} r_d \cdot L_d \cdot \cos \theta = r_d \cdot L_d \cdot (\vec{N} \cdot \vec{L}), & \text{falls } |\theta| < \frac{\pi}{2} \text{ bzw. } (\vec{N} \cdot \vec{L}) > 0 \\ 0 & \text{sonst} \end{cases}$$

○ Beobachter
Position: (0,5,1)

Lichtquelle
○ Position: (6,1,6)



Spekulares Licht

- in der Natur: ideale Reflexionsarten spiegelnd und diffus selten
- häufig
 - deutliches Maximum in Richtung der spiegelnden Reflexion
 - umso kleiner, je weiter man sich von dieser Richtung entfernt
- daher
 - richtungsunabhängiger, diffuser Anteil (Index d)
 - richtungsabhängiger Anteil (Index s)



Lokale Beleuchtungsmodelle

spekular
reflektierte
Leuchtdichte

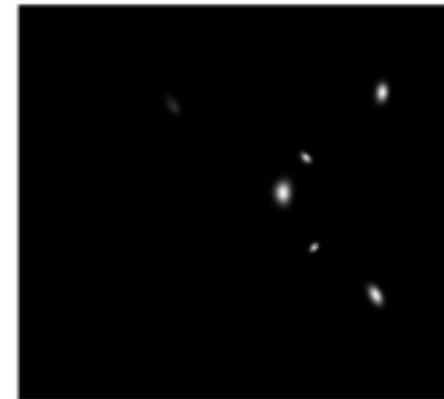
Bestrahlungs-
stärke der
Lichtquelle

Bündelung des
reflektierten Lichts

$$L_{spec} = \begin{cases} r_s \cdot L_s \cdot \cos^m \gamma = r_s \cdot L_s \cdot (\vec{R} \cdot \vec{E})^m, & \text{falls } |\gamma| < \frac{\pi}{2} \text{ bzw. } (\vec{R} \cdot \vec{E}) > 0 \\ 0 & \text{sonst} \end{cases}$$

spekularer
reflektionsgrad

Winkel zwischen
reflektiertem Strahl (R)
und Beobachter-richtung
(E)



Diffuse Beleuchtung
(Quelle: [1])

Lokale Beleuchtungsmodelle

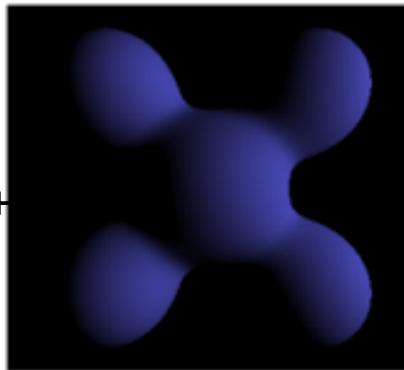
- Phong Modell
- für n Lichtquellen erhält man nach dem Phong'schen Beleuchtungsmodell insgesamt

$$L_{Phong} = L_{amb} + \sum_{j=1}^n (L_{diff} + L_{spec})$$



ambient

+



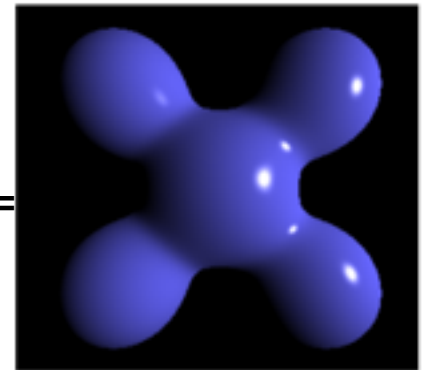
diffus

+



spekular

=



Phong

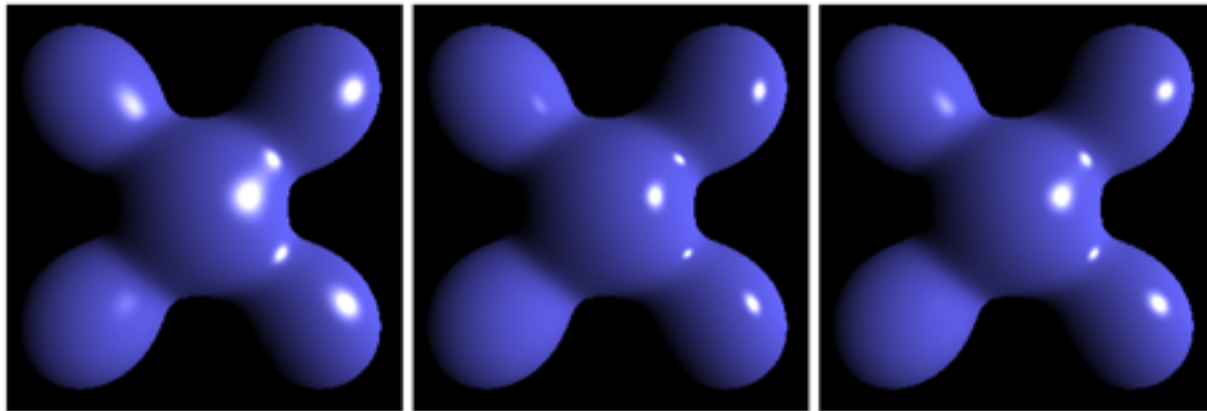
Quelle: [1]

Farbe der Lichtquelle

- bisher haben wir angenommen, dass die Farbe der Lichtquelle keinen Einfluss auf die berechnete Farbe hat.
- also Lichtquelle = weißes Licht (1,1,1)
- falls die Lichtquelle einen anderen Farbwert hat:
 - Multiplikation der Farben LD (diffuse Leuchtdichte) und Lichtquellenfarbe
 - analog bei spekularem Anteil
 - Achtung: komponentenweise Multiplikation der Koordinaten, Ergebnis wieder ein Vektor
 - Beispiel: $(1,2,3) * (2,3,1) = (2,6,3)$
- ambiente Leuchtdichte ist unabhängig von der Lichtquelle und deren Farbe

Alternativen zum Phong-Modell

- es gibt alternative Modelle
- Beispiele
 - Blinn-Phong
 - Torrance-Sparrow
- Unterschiede aber häufig nicht wahrnehmbar
- Phong-Modell ist de-fakto Standard



Blinn-Phong
(Quelle [4])

Phong Quelle: [1]

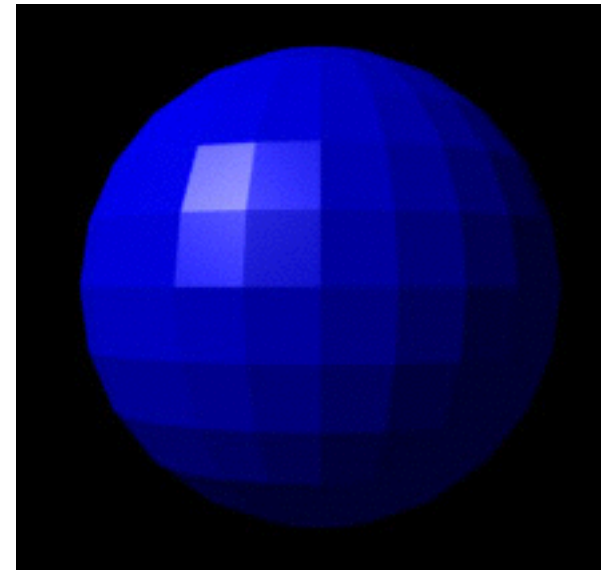
Blinn-Phong
(kleinerer Exponent,
Quelle: [4])



Shading

Flat Shading

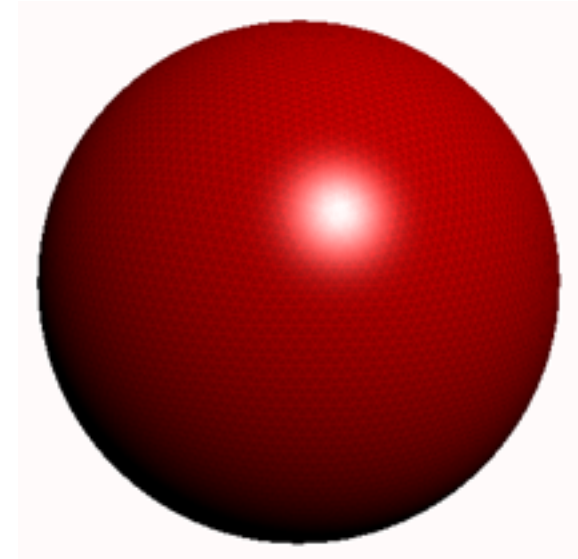
- Konstante Beleuchtung (flat shading)
- einfachster Algorithmus zur Farbgebung polygonal begrenzter Körper
- belegt ganzes Polygon gleichmäßig mit einer Farbe
 - abhängig von der Beleuchtung der Szene
- nur einmal pro Polygon Farbwert berechnen
- Farbe als konstantes Attribut der Fläche



Quelle: [3]

Gouraud Shading

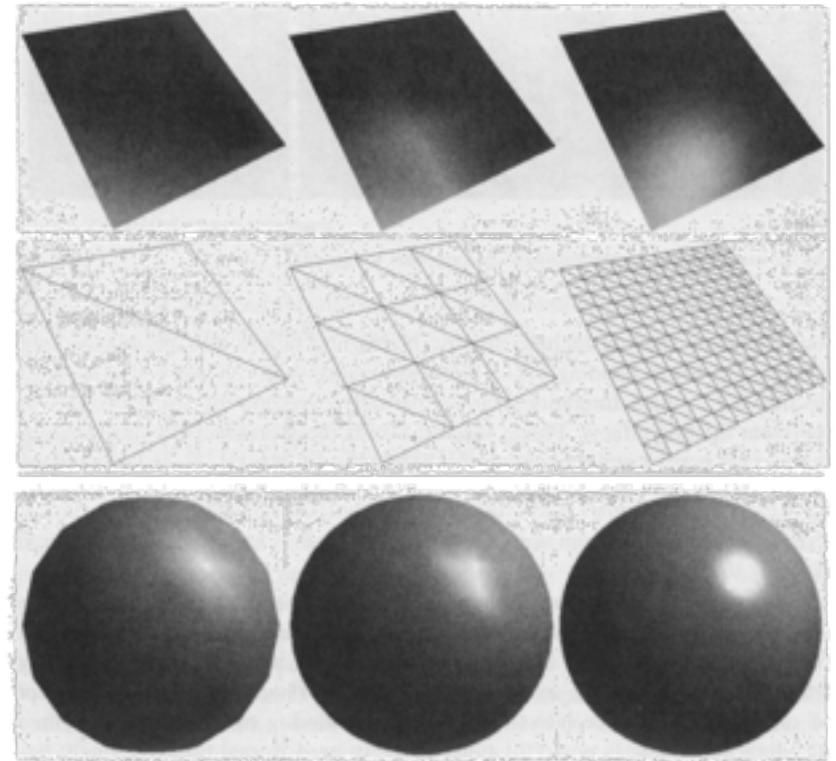
- Gouraud Shading
- 1971 von Henri Gouraud vorgestellt
- basiert auf dem Scanline-Algorithmus (Rasterisierung) von Watkins
- Interpolation der an den Ecken eines Dreiecks berechneten Leuchtdichten
 - entlang der Kanten des Dreiecks
 - anschließend entlang einer Rasterzeile
- Integration in die Rasterisierung
- inkrementelles Verfahren, sehr effizient



Quelle: [5]

Gouraud Shading

- Gouraud Shading
- Einfluss der Triangulierung



Dreiecke

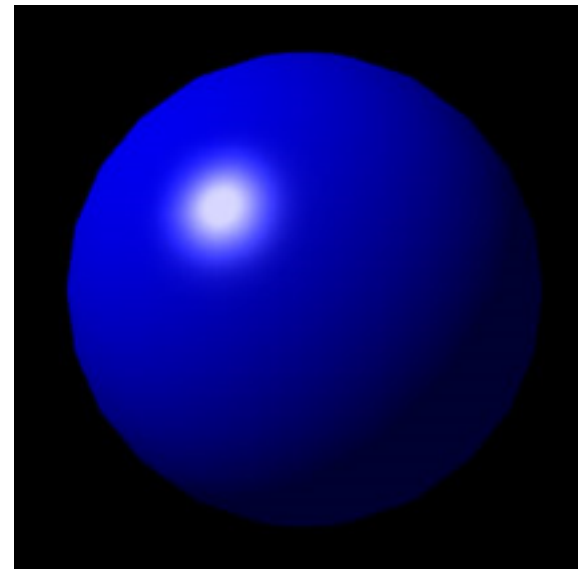
256

1024

16384

Phong Shading

- Phong Shading
- erfasst auch spiegelnde Effekte
- Normalen nötig
- daher: Interpolation der Normalen-Vektoren (nicht der Leuchtdichten) in den Eckpunkten der Dreiecke
- Berechnung ebenfalls inkrementell
 - Integration in Rasterung
 - aber: Normierung der Normalen-Vektoren zur Auswertung des Beleuchtungsmodells
- Berechnung des Farbwerts in jedem Punkt
- nicht zu verwechseln mit Phong-Beleuchtungsmodell!



Quelle: [3]

Übung: Shading

- Wie häufig muss das Phong-Beleuchtungsmodell pro Dreieck ausgewertet werden
 - a) beim Flat Shading?
 - b) beim Gouraud-Shading?
 - c) beim Phong-Shading?



Texturen

Realität

- großes Spektrum geometrischer Formen und physikalischer Materialien
 - Maserungen von Holz, Marmor, Tapeten
 - Wolken
 - Strukturen in unebenen Oberflächen (z.B. Rauputz, Apfelsinen, Baumstämme)
 - im Hintergrund: Häuser, Maschinen, Pflanzen, Personen
- exakte Nachbildung der geometrischen Formen?
 - viel zu aufwändig



Einführung

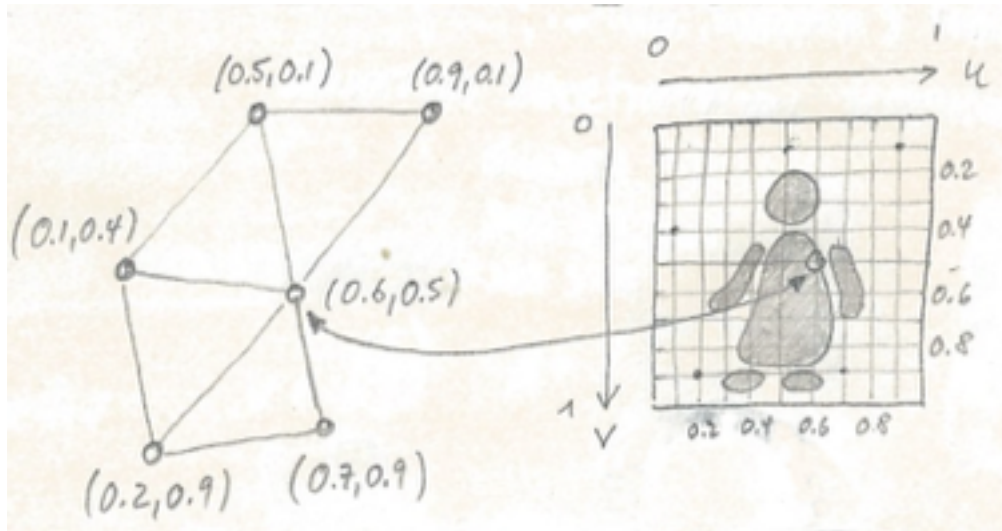
- Idee: Texturierung
 - komplexe Gestaltung des Erscheinungsbildes von Objekten
 - Modellierung der Wand als planare Oberfläche (genauso: Spiegel, Fenster, ...)
 - Tapezieren der Fläche durch ein Bild



Abbildung zwischen Objektraum und Texturraum

Objektraum

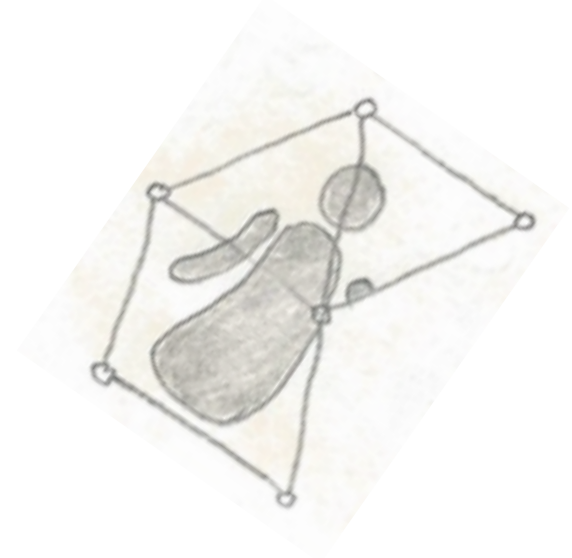
Texturraum



Dreiecksnetz mit
Texturkoordinaten (u,v) für
jeden Vertex

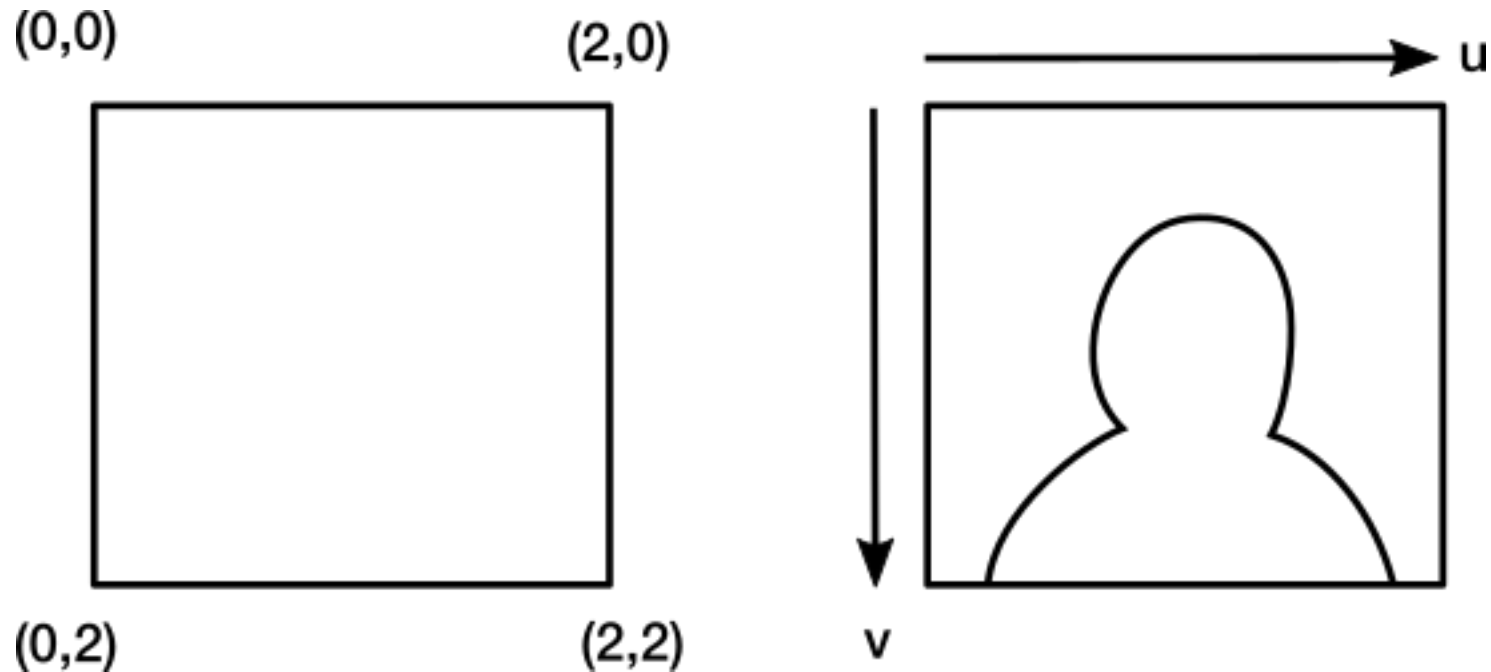
Textur mit
Koordinatenachsen u
und v

Textur auf Dreiecksnetz
übertragen



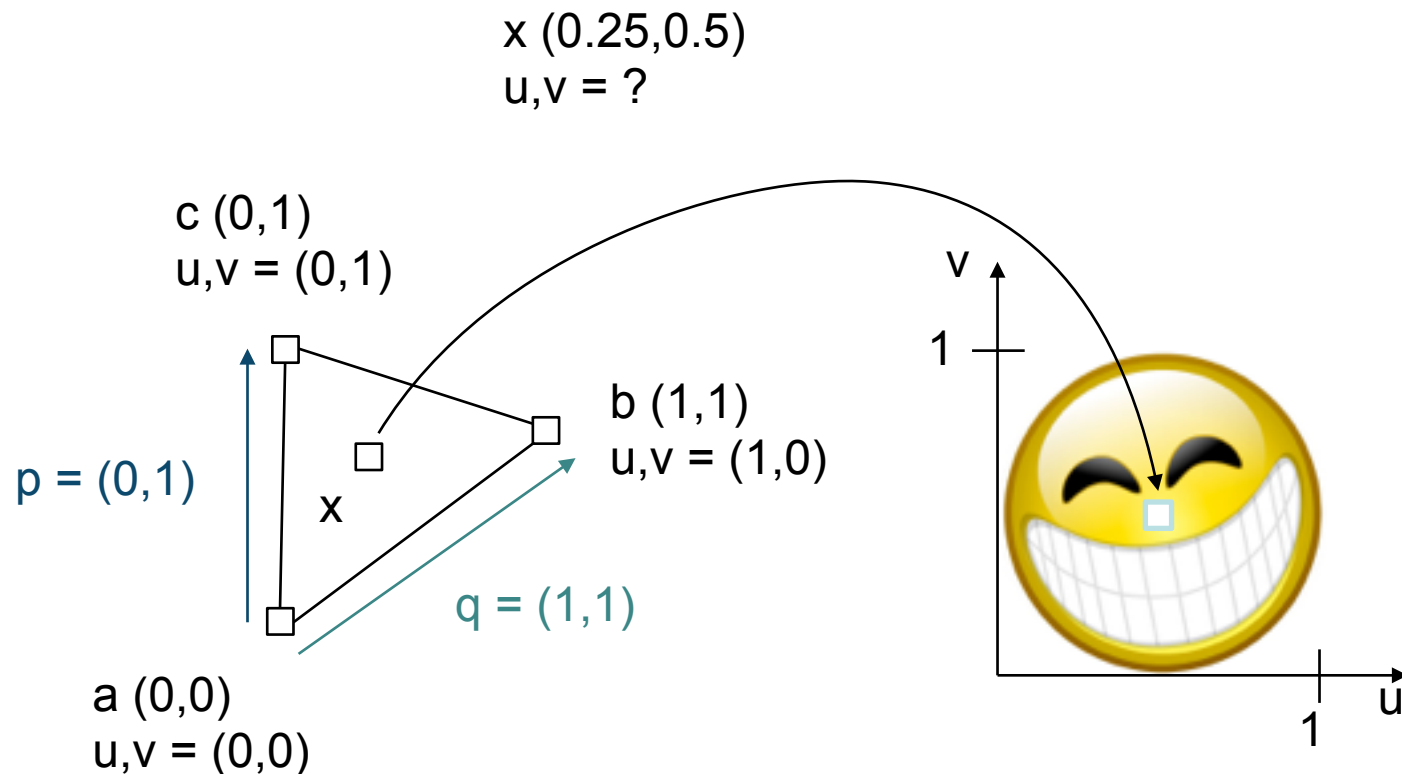
Übung: Texturierung

- Skizzieren Sie das Ergebnis der Texturierung des Rechtecks links mit der Textur rechts. An den Ecken sind die Texturkoordinaten angegeben.



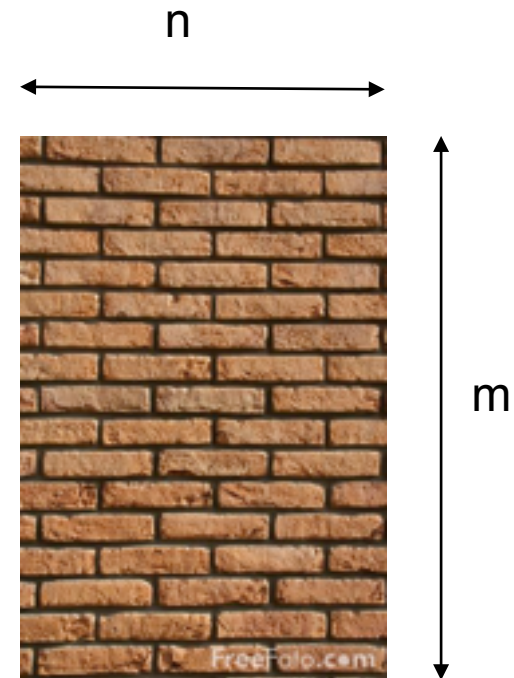
Übung: Berechnung Texturkoordinaten

- Gegeben ist ein Dreieck in der Bildebene (2D) mit den Eckpunkten a , b , c und den zugehörigen Texturkoordinaten.
- Gesucht sind die Texturkoordinaten am Punkt x .



Einführung

- Aufbau von Texturen
- Raster von Farbwerten
- 2D: Bild mit Breite n und Höhe m
 - $\{ C[i,j] \mid 0 \leq i \leq n, 0 \leq j \leq m \}$
- jeder Eintrag
 - Vektor mit drei Komponenten (r,g,b)
 - wird als Texel bezeichnet



Einführung

- Erzeugen von Texturen
- Vorrat nahezu unerschöpflich
 - Fotoapparate
 - Scanner
 - Speichermedien
- Generieren von komplexen 2D-Texturen
 - einfach
 - photorealistische Ergebnisse

Photorealistische Texturen (Quelle: [8])



Einführung

- Herausforderungen
- große Bilder haben hohen Speicherbedarf
- beim Vergrößern kleiner Bilder treten Artefakte auf
- Wiederholung (Fortführung von Texturen oft kompliziert)



Wiederholbare
Textur
(Quelle: [9])

- Kontext der Bilder passt oft in 3D-Anwendung nicht mehr
 - Sonnenstand
 - Schattenwurf
 - ...

Texturüberlauf

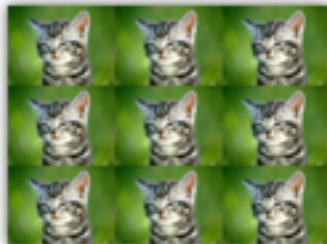
- Texturkoordinaten außerhalb des gültigen Bereichs [0;1]; was dann?
- Möglichkeiten

- periodisch wiederholen
- letzten Farbwert wiederholen

- in OpenGL

- `glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);`
- `glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);`

Texturkoordinate
(u = s, v = t)



GL_REPEAT



GL_MIRRORED_REPEAT



GL_CLAMP_TO_EDGE



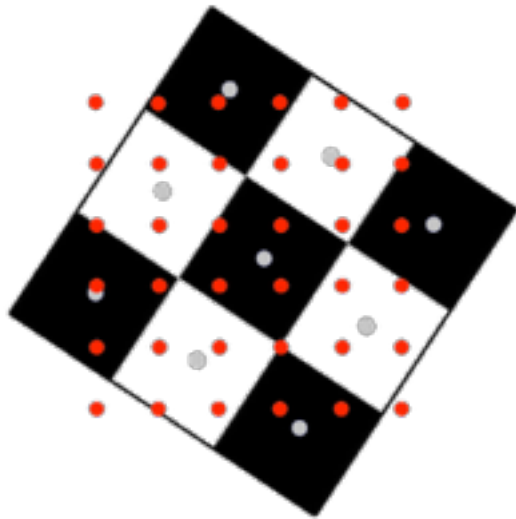
GL_CLAMP_TO_BORDER



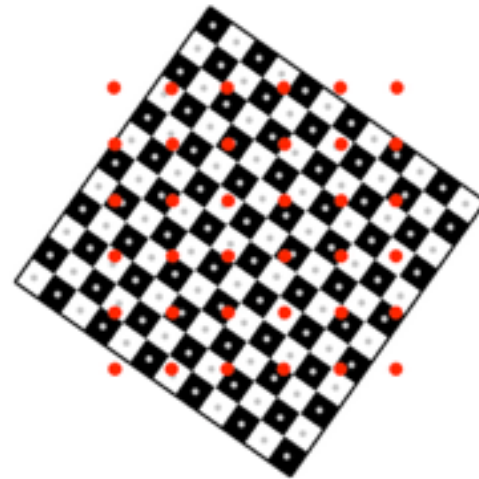
Aliasing

Aliasing

- Herausforderung
 - zu viel/zu wenig Information in Textur für Bildschirmdarstellung
 - Textur: Schachbrettmuster im Hintergrund
 - Pixel auf dem Bildschirm: rote Punkte



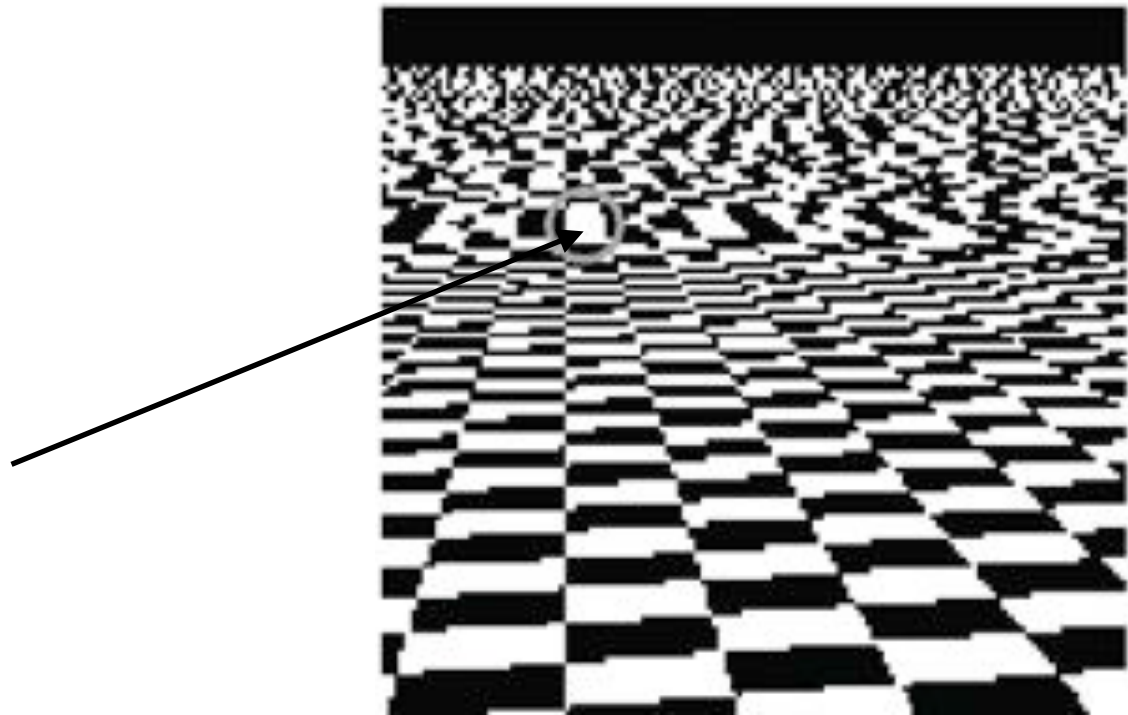
Vergrößerung
(zu wenig Texturinformation)



Verkleinerung
(zu viel Texturinformation)

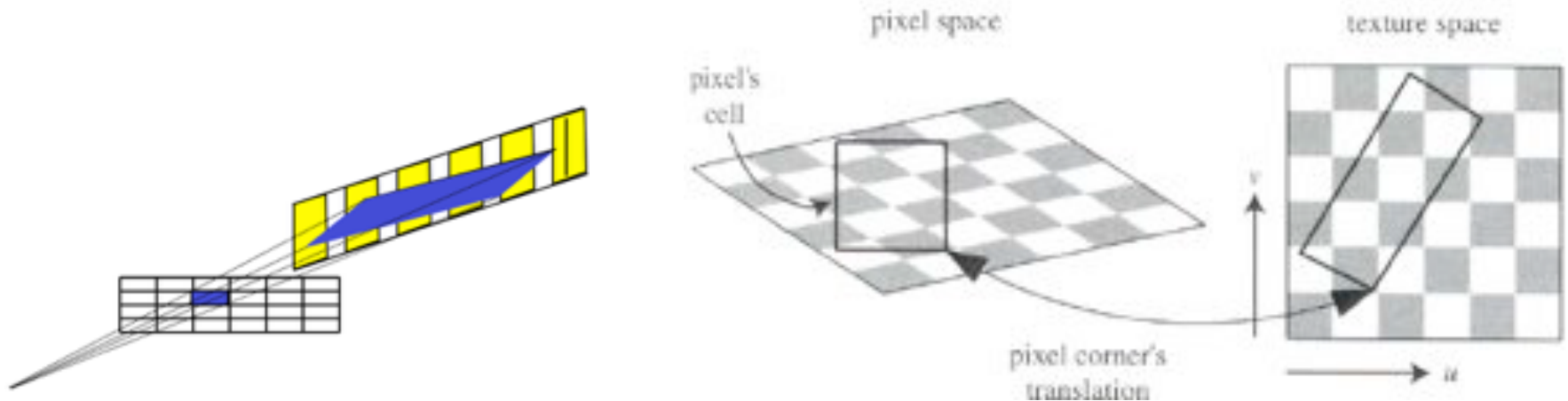
Aliasing

- Aliasing
- Abbildung auf Bildschirmpixel für ästhetische Darstellung nicht ausreichen
- Entstehung: Rundungsartefakte



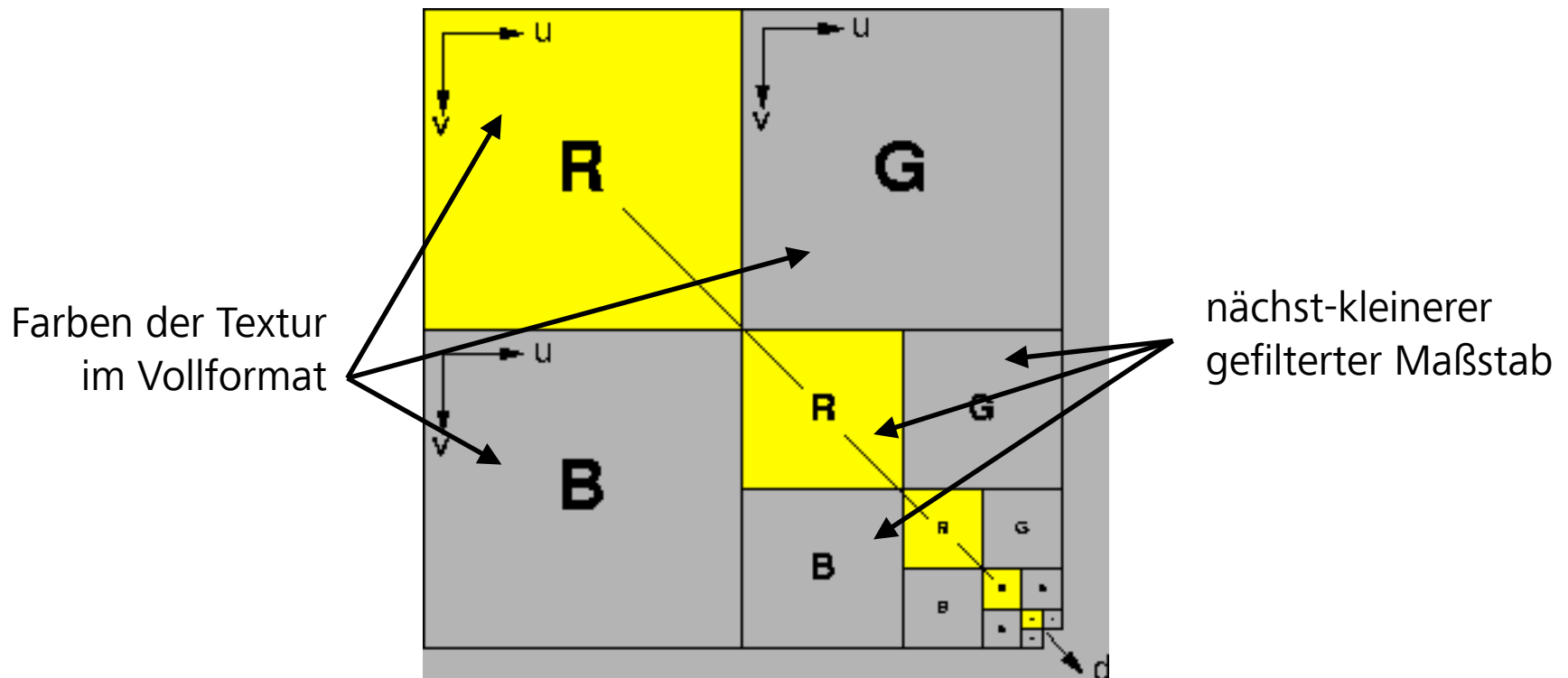
Aliasing

- Lösungsidee
- Berechnung des Footprints
 - Abbildung des Parallelogramms, das der Pixel auf der Textur bedeckt
 - Einsammeln der Texturwerte innerhalb des Footprints
- Interaktive Anwendungen
 - Vorberechnung durch Reduktion auf einfache Flächen
 - Verwenden von Mip-Maps



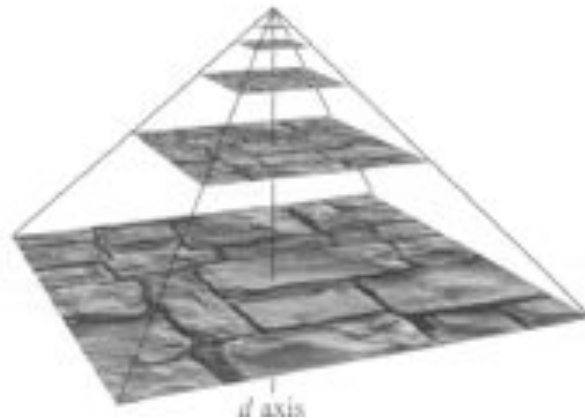
Aliasing

- Mip-Map
- speichert quadratische Textur der Größe $n \times n$, $n = 2^k$
- jeweils in halbierter (gefilterter) Auflösungsstufe



Aliasing

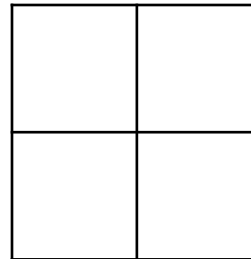
- Mip-Map
- Stufe 0
 - Übernehmen der Texturwerte
 - $C_{mip_0}[i,j] = C[i,j]$
- übrige Stufen
 - Filterung der vorangegangenen Stufe
 - $C_{mip_d}[i,j] = 0.25 * (C_{mip_{d-1}}[2i,2j] + C_{mip_{d-1}}[2i+1,2j] + C_{mip_{d-1}}[2i,2j+1] + C_{mip_{d-1}}[2i+1,2j+1])$



Übung: Mip-Map

- Gegeben ist ein Ausschnitt aus dem Rot-Kanal (Wertebereich 0 ... 255) einer Mip-Map-Stufe der 0. Geben Sie die Werte der restlichen Stufen an.

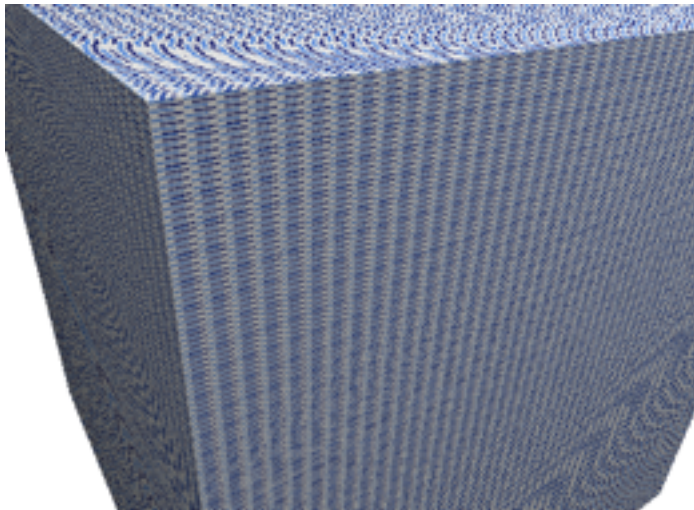
159	160	133	133
157	156	133	133
110	113	8	107
116	113	208	109



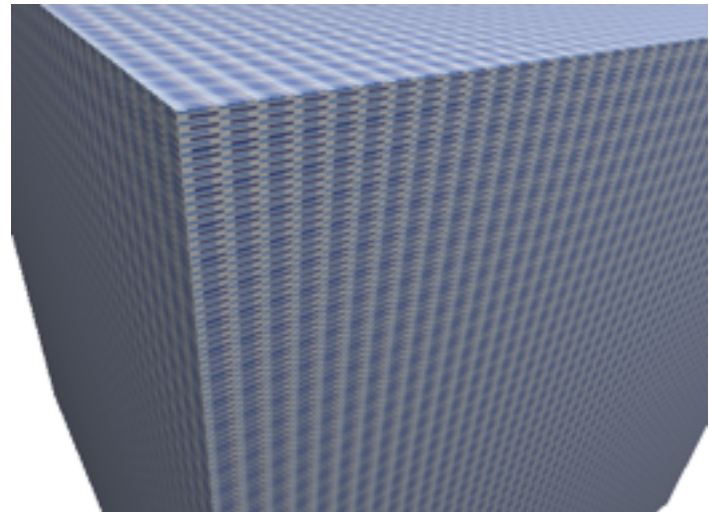
Mip-Mapping in OpenGL

- Erzeugen der Mip-Maps
 - `glGenerateMipmap(GL_TEXTURE_2D);`
 - oder im Texture-Konstruktor von JOGL
- Interpolationsformat wählen

```
gl.glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL2.GL_LINEAR);  
gl.glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
```



ohne Mip-Mapping



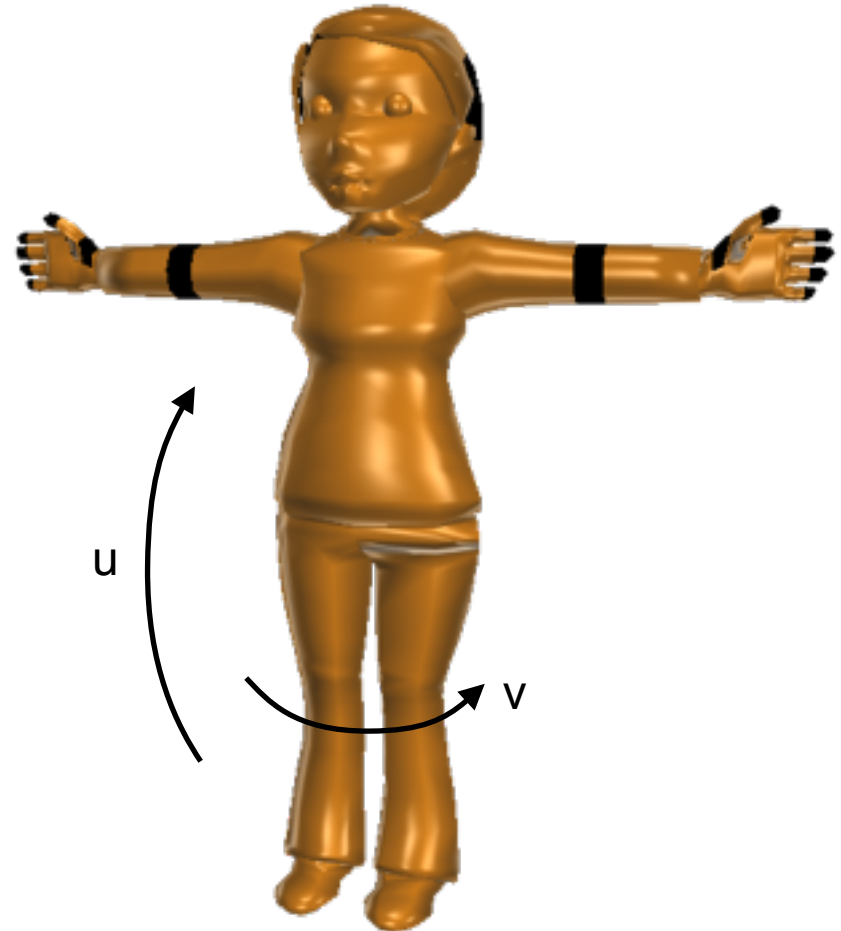
mit Mip-Mapping



Erzeugen von Texturkoordinaten

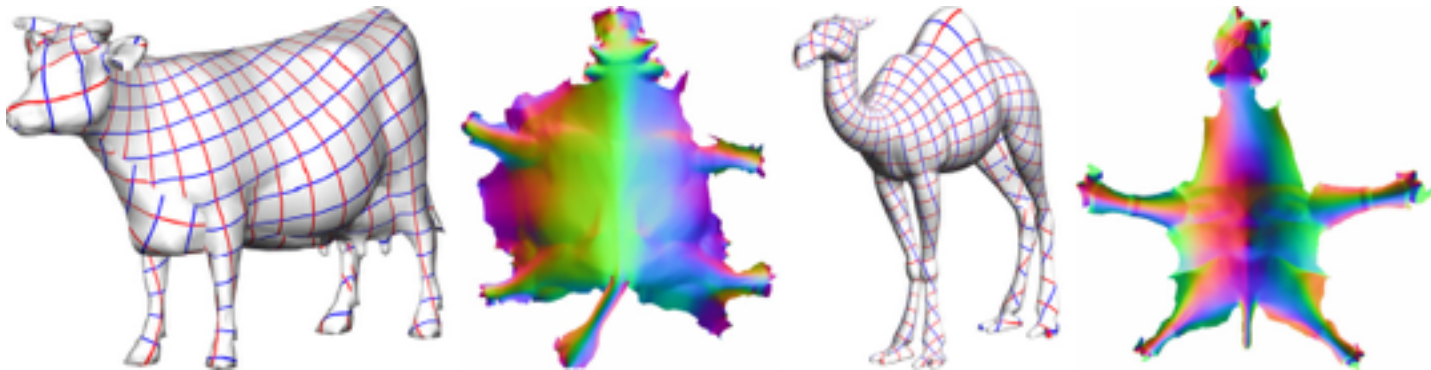
Erzeugen von Texturkoordinaten

- Aufgabe: Parametrisierung
 - für jeden Vertex: (u,v) -Koordinaten
- manuelle Zuweisung
 - uv-Mapping
 - sehr aufwändig



Erzeugen von Texturkoordinaten

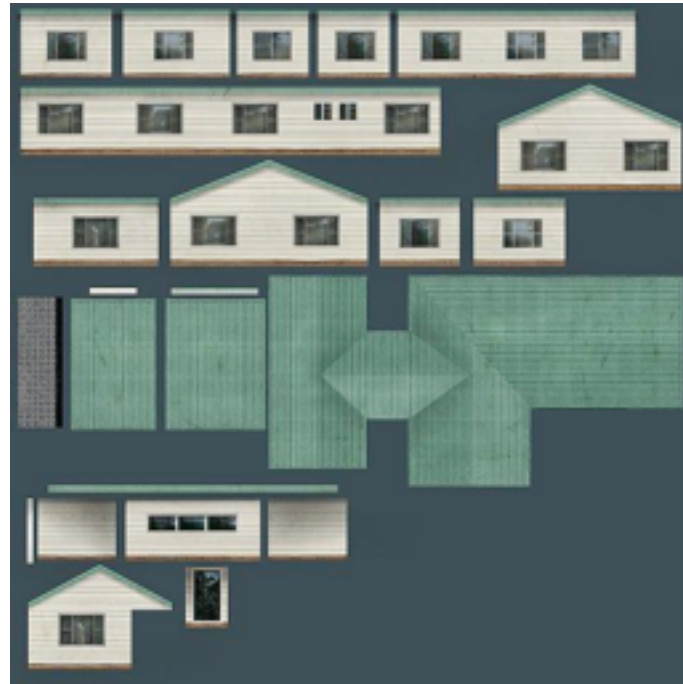
- Aktives Forschungsgebiet: Entwicklung von Algorithmen zur automatischen Parametrisierung von Oberflächen



Parametrisierung von Oberflächen (Quelle [11])

Textur-Atlas

- Sammeln mehrere Einzeltexturen in einem Texturbild
 - sogenannter Texturatlas



Texturatlas
(Quelle [10])



Weitere Anwendungen für Texturen

Weitere Anwendungen für Texturen

- Texturen können weitere Oberflächeneigenschaften darstellen
 - Light-Mapping
 - Environment-Mapping
 - Bump-Mapping

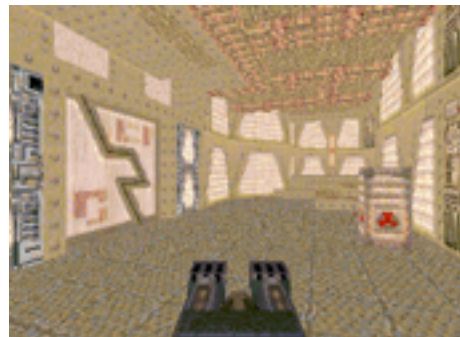
Light-Mapping

- Beleuchtungsmodell vorberechnen
 - z.B. Phong-Modell
- Die Leuchtdichte L_{phong} ist wellenlängenabhängig
 - muss zur Ausgabe in darstellbare C_{phong} umgerechnet werden
 - Modulierung der Texturfarbe mit Lightmaps



Light-Map

+



Texturen

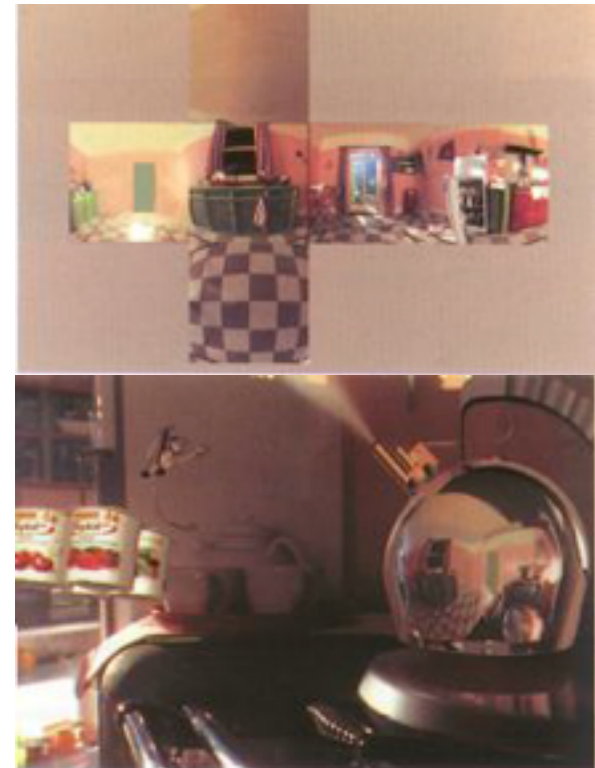
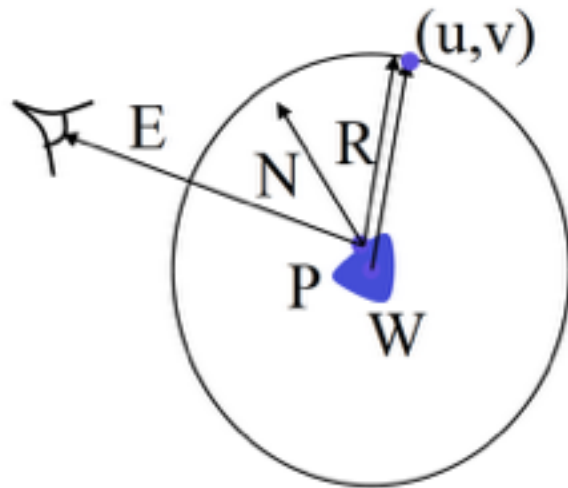
=



Kombination
(Quelle: [12])

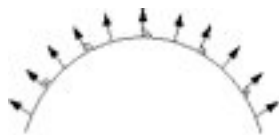
Environment-Mapping

- Approximation von Reflexionen über Texturen
- Idee
 - Umgebungstextur repräsentiert "Panorama"



Bump-Mapping

- Material und Farbe allein erzeugen oft zu glatte Oberflächen
- realistische Oberflächen brauchen Struktur
- Idee
 - Modellierung der geometrischen Struktur als Offset-Fläche
 - $P'(u,v) = P(u,v) + h(u,v) N(u,v) / |N(u,v)|$



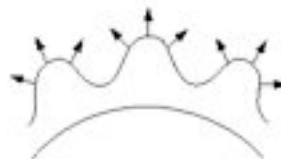
(a) Originalfläche $P(u)$
mit Normalen $N(u)$



(b) Bump Map $h(u)$



(c) Offsetfläche $P'(u)$



(d) Perturbierte Normalen $N'(u)$



Oberfläche mit Bump-
Map

Bump-Mapping

- Geometrische Positionen $P'(u,v)$ im Allgemeinen nicht notwendig
- ausreichend, die angepassten Normalen in der Beleuchtungsrechnung zu verwenden
 - Gouraud-Shading nicht ausreichend
 - Phong-Shading
 - Umsetzung über Fragment-Shader



Oberfläche mit Bump-Map
(Quelle: [13])

Zusammenfassung

- Beleuchtung
 - Einführung
 - Lokale Beleuchtungsmodelle
 - Shading
- Texturen
 - Einführung
 - Aliasing
 - Erzeugen von Texturkoordinaten

Quellen

- Die Folien basieren u.a. auf Vorlesungsfolien von Prof. Dr. Wolfgang Straßer (Universität Tübingen) und Prof. Dr. Stefan Gumhold (Technische Universität Dresden) und Prof. Dr. Marc Alexa, Technische Universität Berlin, 2008
- [1] Wikipedia: Phong Beleuchtungsmodell, <http://de.wikipedia.org/wiki/Phong-Beleuchtungsmodell>, abgerufen am 27.10.2013
- [2] S. R. Marschner, E. P. F. Lafortune, S. H. Westin, K. E. Torrance, D. P. Greenberg: Image-Based BRDF Measurement, 1999
- [3] Wikipedia: Phong Shading, http://de.wikipedia.org/wiki/Phong_Shading, abgerufen am 27.10.2013
- [4] Wikipedia: Blinn-Beleuchtungsmodell, <http://de.wikipedia.org/wiki/Blinn-Beleuchtungsmodell>, abgerufen am 27.10.2013
- [5] Wikipedia: Gouraud-Shading, http://de.wikipedia.org/wiki/Gouraud_Shading, abgerufen am 27.10.2013
- [6] <http://emileeknits.wordpress.com/category/swatch-sunday/>, abgerufen am 8.11.13
- [7] <http://healthtoken.com/human-body-outline-for-kids-and-adult/human-body-outline-for-kids-459/>, abgerufen am 8.11.13
- [8] http://cdn.overclock.net/f/ff/fff1d246_2012-04-26_00003.jpeg, abgerufen am 8.11.13
- [9] 123RF Bildnummer: 11812779, de.123rf.com, abgerufen am 8.11.13
- [10] http://wiki.simigon.com/wiki/index.php?title=Terrain_texture_atlas, abgerufen am 8.11.13
- [11] Alla Sheffer, Emil Praun and Kenneth Rose: Mesh Parameterization Methods and Their Applications, Foundations and Trends in Computer Graphics and Vision, Vol. 2, No 2 (2006) 105–171
- [12] Quake I, ID Software
- [13] http://www.chromesphere.com/tutorials/vue6/Optics_Basics_Print.html, abgerufen am 8.11.13
- [14] <http://www.adobe.com/devnet/flashplayer/articles/how-stage3d-works.html>, abgerufen am 8.11.13
- [15] T. Akenine-Möller, E. Haines, N. Hoffman: Real-Time Rendering, 3rd edition, CRC Press, 2008
- [16] <http://prideout.net/blog/?cat=19>, abgerufen am 8.11.13
- [17] <https://open.gl/textures>, abgerufen am 25.08.2015