

Volumenvisualisierung: Klassifikation und Shading

Christof Rezk-Salama

Visualisierung WS 03/04, 20.01.2004

computergraphik und multimedia systeme
universität siegen



Letzte Stunde

Texturbasierte Volumenvisualisierung

- Ausnutzung der Standard-Grafik-Hardware

2D Texturen:

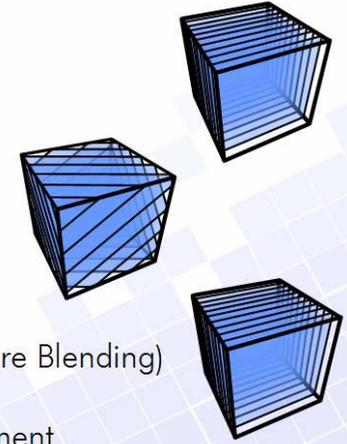
- bilineare Interpolation
- achsenparallele Schichten

3D Texturen:

- trilineare Interpolation
- Viewport-parallele Schichten
- ein Texturblock, Bricking

2D Multitexturen

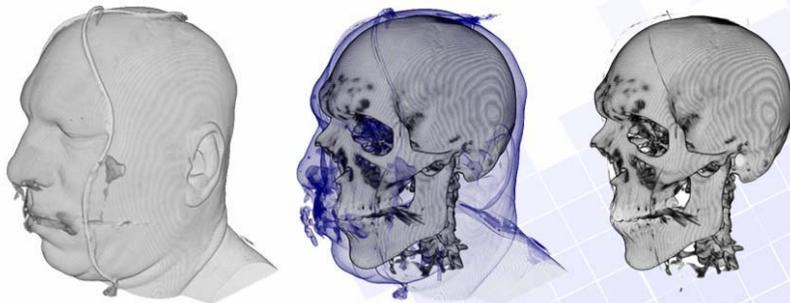
- trilineare Interpolation (Texture Blending)
- achsenparallele Schichten
- Effizientes Memory Management.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Klassifikation

- Im Klassifikationsprozess spezifiziert der Benutzer das „*Aussehen*“ seiner Daten.
 - Welche Bereiche sollen transparent sein?
 - Welche Bereiche sollen welche Farbe haben?



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Klassifikation

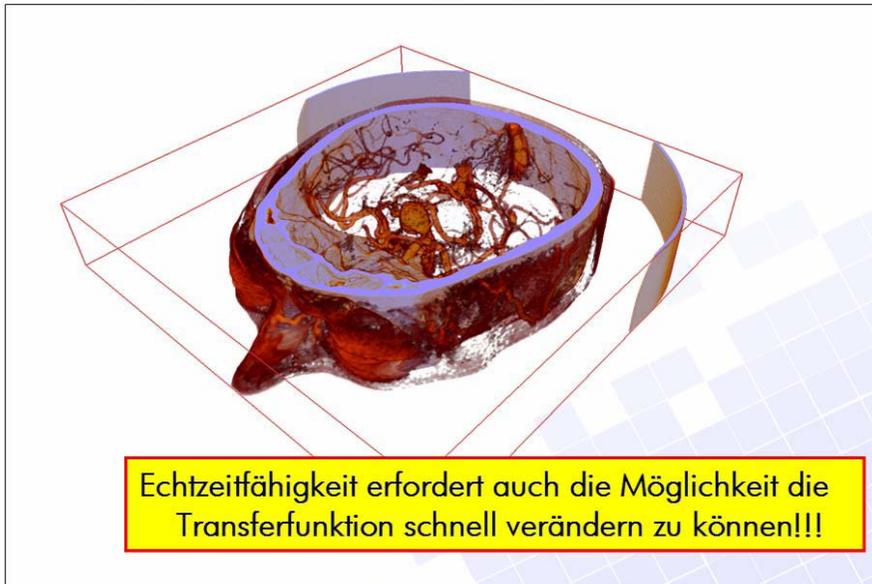
- Im Klassifikationsprozess spezifiziert der Benutzer das „*Aussehen*“ seiner Daten.
 - Welche Bereiche sollen transparent sein?
 - Welche Bereiche sollen welche Farbe haben?
- Der Benutzer stellt eine *Transferfunktion* ein.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Beispiel: CT Angiographie

6



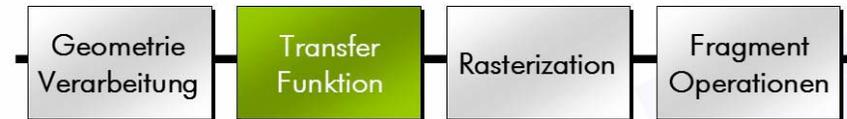
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Klassifikation – the easy way

7

● Präklassifikation:

Farbtabelle wird *vor* der Interpolation angewendet.
(präinterpolative Transferfunktion)



- Ein Farbwert wird in der Tabelle nachgeschaut für jeden Voxel
- Ein RGBA Wert wird bestimmt für jeden Voxel

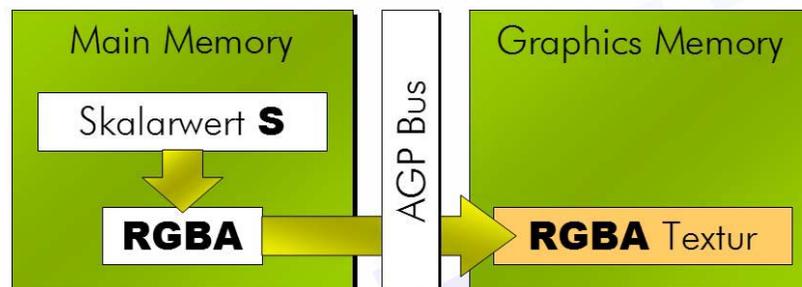
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Mögliche Implementierungen

9

● Der naive Ansatz:

Speichere die Emissions- und Absorptionsterme direkt in der Textur.



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Mögliche Implementierungen

9

● Der naive Ansatz:

Speichere die Emissions- und Absorptionsterme direkt in der Textur.

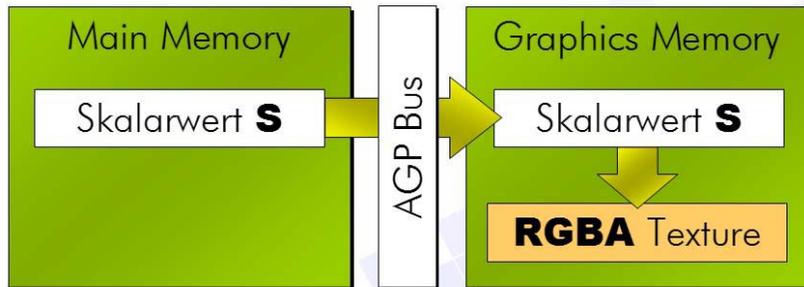
- Sehr hoher Speicherbedarf
 - Main Memory (RGBA und Skalarvolumen)
 - Graphics Memory (RGBA Volumen)
- Hohe Last für den Speicherbus
RGBA Volumen muss transferiert werden.
- Upload notwendig, bei Änderung der TF !!!

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Mögliche Implementierungen ¹⁰

● Eine bessere Lösung:

Anwendung der Farbtabelle während des Textur Transfers vom Main Memory auf die Graphikkarte (standard OpenGL feature)

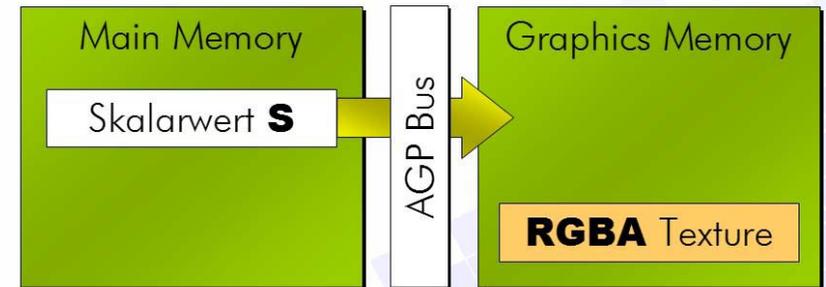


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Mögliche Implementierungen ¹⁰

● Eine bessere Lösung:

Anwendung der Farbtabelle während des Textur Transfers vom Main Memory auf die Graphikkarte (standard OpenGL feature)



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Mögliche Implementierungen ¹¹

● Eine bessere Lösung:

Anwendung der Farbtabelle während des Textur Transfers vom Main Memory auf die Graphikkarte (standard OpenGL feature)

● Hoher Speicherbedarf

- Main Memory (nur Skalarvolumen)
- Graphics Memory (RGBA Volumen)

● Verringerte Last auf dem Speicherbus

- Nur das Skalarvolumen wird transferiert.

● Upload notwendig bei Änderung der TF !!!

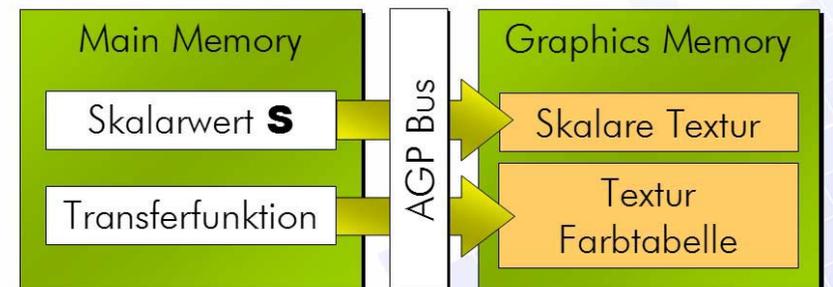
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Mögliche Implementierungen ¹²

● Der beste Ansatz: Texturpaletten

Speichere das Skalarvolumen zusammen mit der Farbtabelle direkt im Grafikspeicher.

● Hardwareunterstützung dafür ist notwendig!



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Mögliche Implementierungen ¹³

- Der beste Ansatz: Texturpaletten
Speichere das Skalarvolumen zusammen mit der Farbtabelle direkt im Grafikspeicher.
- Hardwareunterstützung dafür ist notwendig!
- Geringer Speicherbedarf
 - Main Memory (Skalarvolumen kann gelöscht werden!)
 - Graphics Memory (Skalarvolumen + TF)
- Geringe Last auf dem Speicherbus
 - Skalarvolumen muss **nur einmal** transferiert werden!
- Nur die Farbtabelle muß bei Änderungen der TF hochgeladen werden!

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Klassifikation – the easy way ¹⁴

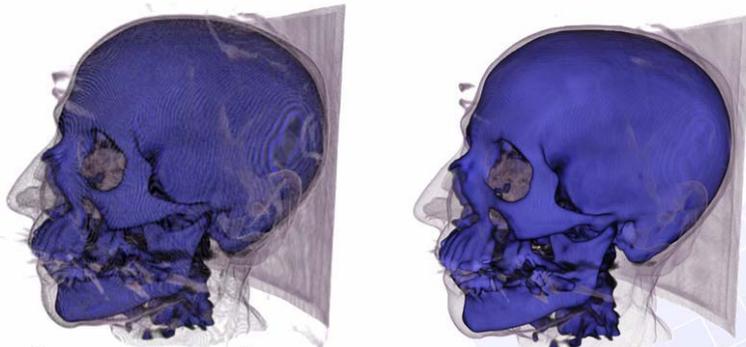
- Zusammenfassung Präklassifikation
 - Anwendung der Transferfunktion vor der Rasterisierung
 - Ein RGBA Lookup für jeden Voxel
 - Unterschiedliche Implementierungen:
 - Texture Transfer
 - Texture Color Tables (paletted textures)
 - Einfach und Effizient

... aber Präklassifikation ist nicht wirklich gut!

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Warum Präklassifikation nicht gut ist. ¹⁵

- Vergleiche die Bildqualität mit Raycasting



Texturbasiert mit Präklassifikation

Ray casting (software)

Gleiche TF, gleiche Auflösung, gleiche Abtastrate

Was haben wir falsch gemacht?

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Abtasttheorem ¹⁶

Das Abtasttheorem verlangt:

- Das kontinuierliche Skalarfeld ist bandbegrenzt:
Ein kontinuierliches Signal $f(x)$, dessen Frequenzspektrum $F(t)$ bandbegrenzt ist, d.h.

$$F(t) = 0 \quad \text{für} \quad |t| > t_{max}$$

läßt sich **exakt rekonstruieren** aus einer diskreten Abtastung mit einer Schrittweite:

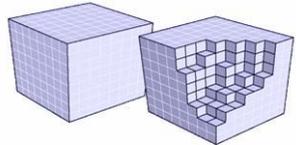
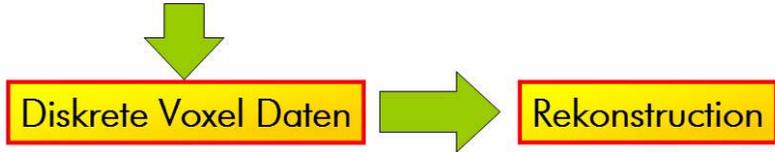
$$\tau > \frac{1}{2t_{max}}$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

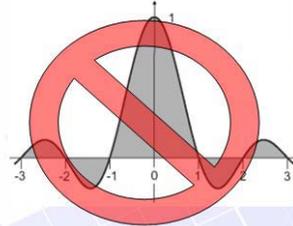
Abtasttheorem

17

- Kontinuierliches 3D Skalarfeld



ideal: SINC-Filter



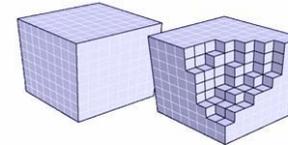
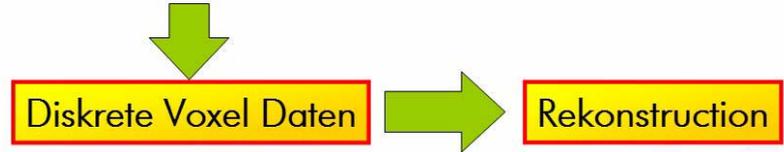
Der diskrete Datensatz enthält die gesamte Information des kontinuierlichen Skalarfeldes

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

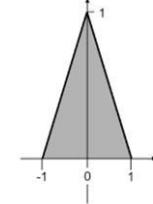
Abtasttheorem

17

- Kontinuierliches 3D Skalarfeld



real: Dreiecksfilter



Der diskrete Datensatz enthält die gesamte Information des kontinuierlichen Skalarfeldes

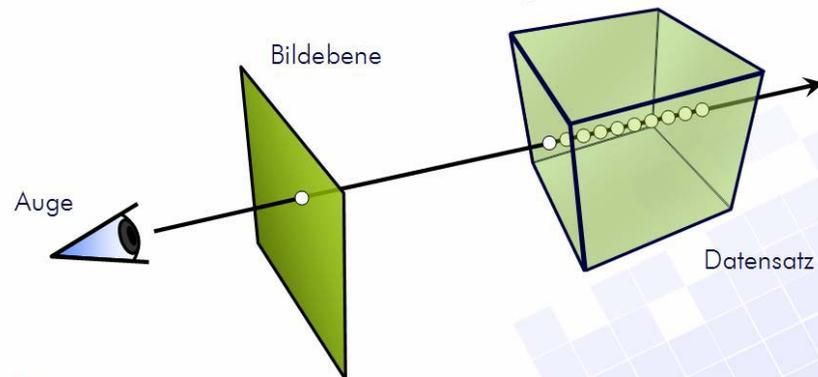
trilineare Interpolation

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Rekonstruktion

18

- Wir visualisieren das kontinuierliche Skalarfeld!
Also wann wird das kontinuierliche Signal rekonstruiert?



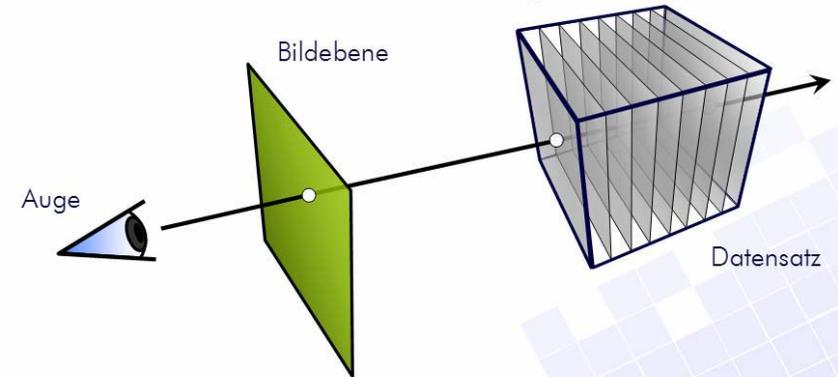
- Antwort für Raycasting:
Rekonstruktion während der Integration (beim Resampling)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Rekonstruktion

19

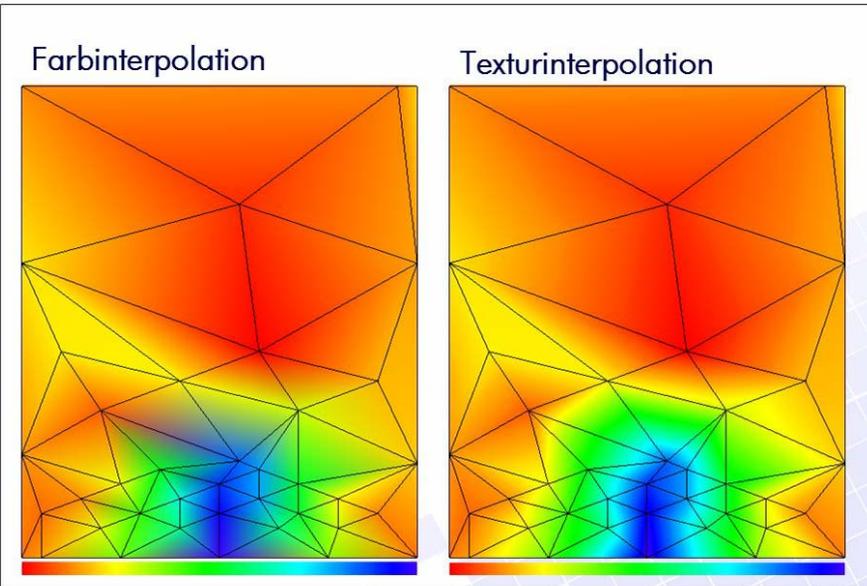
- Wir visualisieren das kontinuierliche Skalarfeld!
Also wann wird das kontinuierliche Signal rekonstruiert?



- Antwort für texturbasierte Verfahren:
Rekonstruktion während Texture Mapping/Rasterization

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

2D Skalarfelder



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Warum Präklassifikation nicht gut ist.

Abtasttheorem erfüllt für das Skalarfeld S

↓

Ersetze S durch $T(S)$ für jeden Voxel

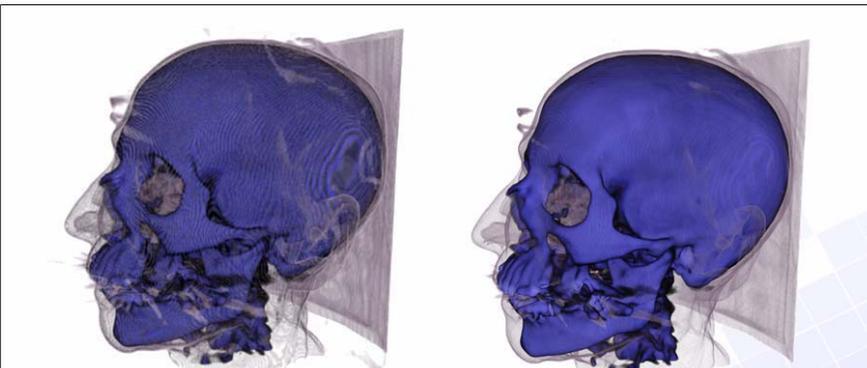
↓

Das Abtasttheorem ist nicht automatisch auch für $T(S)$ erfüllt !!

- Die Transferfunktion $T(S)$ ändert das Frequenzspektrum des Signals. Es werden Frequenzen hinzugefügt, die durch die ursprüngliche Schrittweite nicht erfasst werden.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Warum Präklassifikation nicht gut ist.



Texturbasiert mit Präklassifikation

Ray casting (software)

- Die Rekonstruktion muss das modifizierte Frequenzspektrum berücksichtigen, das durch die Transferfunktion entsteht.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Klassifikation - the hard way

- Postklassifikation:**
Die Farbtabelle wird erst nach der Interpolation angewendet (*post-interpolative Transferfunktion*).

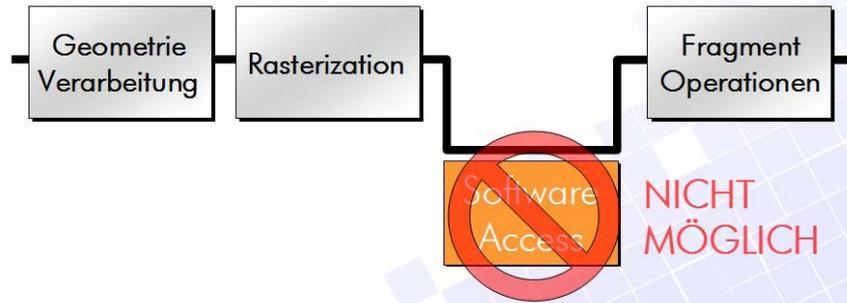
```
graph LR; A[Geometrie Verarbeitung] --> B[Rasterization]; B --> C[Transfer Funktion]; C --> D[Fragment Operationen];
```

- Eine Farbwert wird in der Tabelle nachgeschaut für jedes Fragment !
- Ein RGBA Wert wird bestimmt für jedes Fragment !

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Implementierung

- Die Implementierung ist schwierig: Die Hardware erlaubt es nicht Werte aus der Pipeline zu extrahieren, in Software zu modifizieren und sie wieder einzufügen!



Implementations

- Wir benötigen einen Mechanismus, der es uns erlaubt eine Farbtabelle anzuwenden, nachdem die Fragmente interpoliert wurden

➔ **Solution:** Moderne Graphikhardware

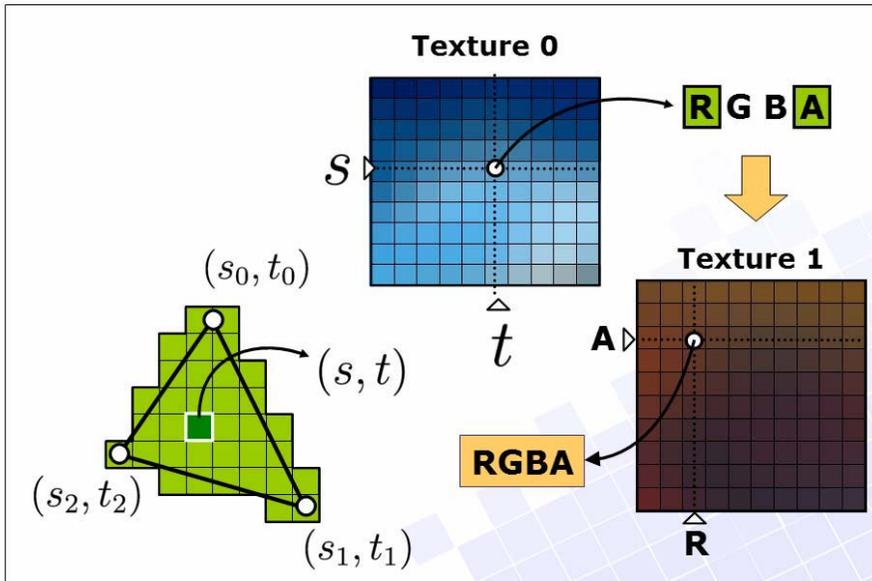
Traditionelle Fixed-Function Pipeline:



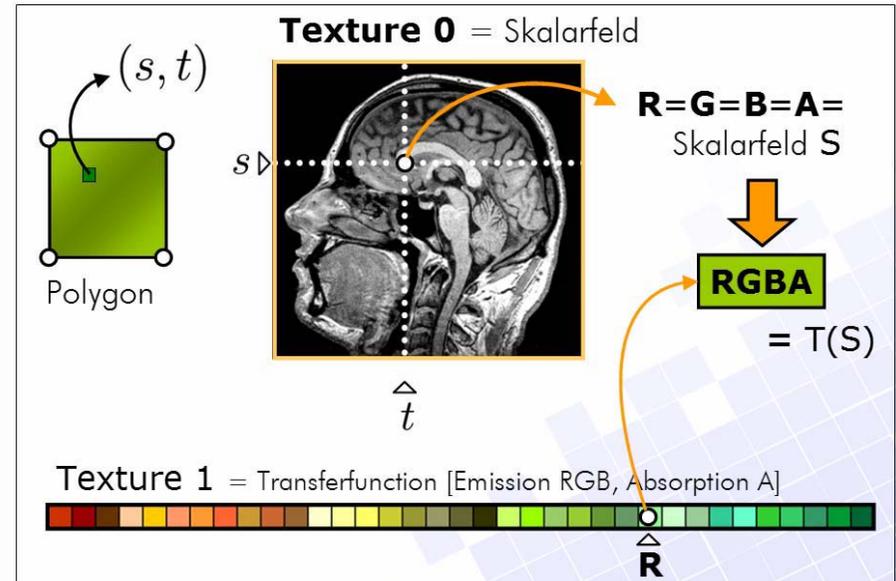
Moderne GPU mit Vertex Shadern und Pixel Shadern:



Dependent Textures

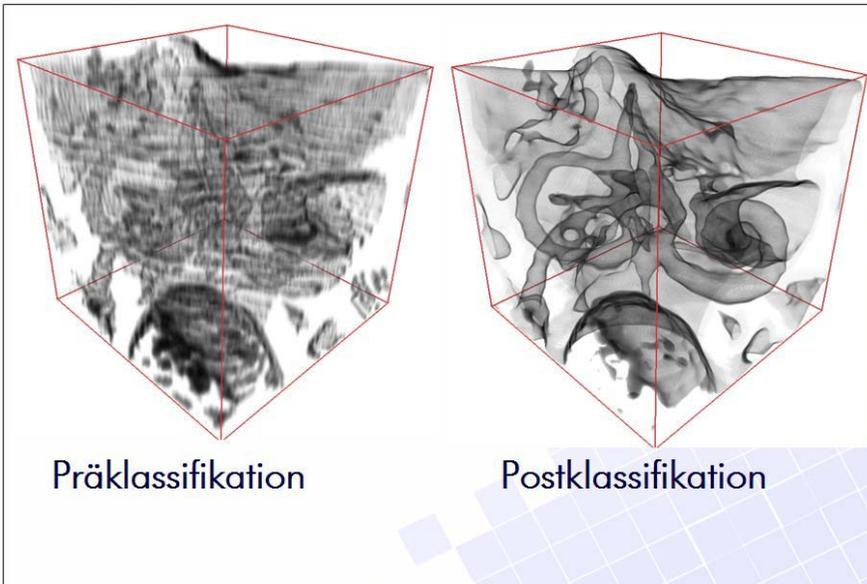


Postklassifikation



Vergleich

28



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Klassifikation - the hard way

29

- *Zusammenfassung Postklassifikation*
 - Transferfunktion wird nach der Rasterisierung angewendet
 - Ein RGBA lookup für jedes Fragment
 - Dependent Textures:
 - Per Fragment Interpolation
 - Per-Fragment Color Lookup
 - Berücksichtigt Transferfunktionen mit hohen Frequenzen
 - einfach durch erhöhen der Abtastrate (Anzahl der Schichten)
 - Effiziente Implementation

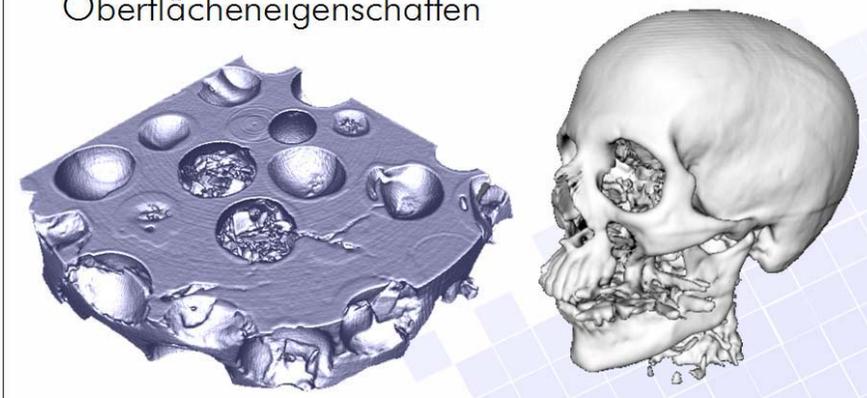
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Lokale Beleuchtung

32

Wozu brauche ich lokale Beleuchtung?

- Verbessert die Tiefenwahrnehmung
- Verbessert die Wahrnehmung von Oberflächeneigenschaften



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Lokale Beleuchtung

33

- *Berücksichtige nur die lokale Form des Objekts*
 - kein Schatten
 - keine „Self-Shadowing“
- *Keine globalen Beleuchtungseffekte*
 - Indirekte Beleuchtung
 - Kaustiken
- *Wir wollen hier das (Blinn-)Phong-Beleuchtungsmodell für Oberflächen betrachten und versuchen, es auf Volumendaten zu übertragen.*

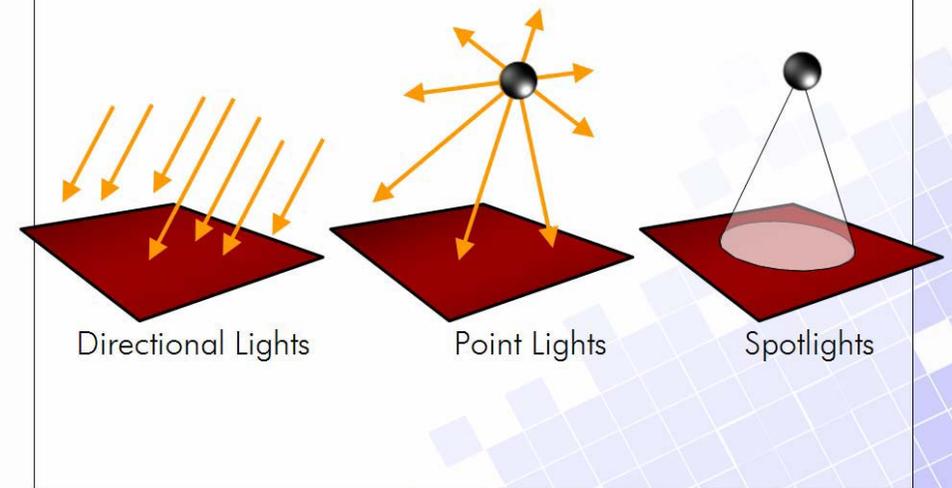
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Globale Beleuchtung



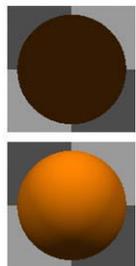
Lichtquellen

● Unterschiedliche Typen von Lichtquellen



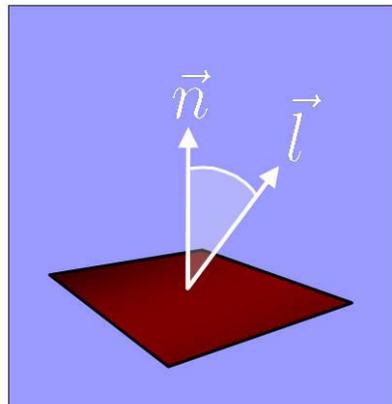
Lokale Beleuchtung

Blinn-Phong Model für Oberflächen



ambient

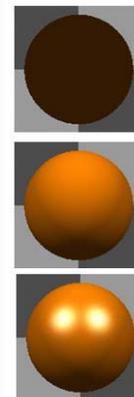
diffuse



$$I = I_a + I_d (\vec{n} \circ \vec{l})$$

Lokale Beleuchtung

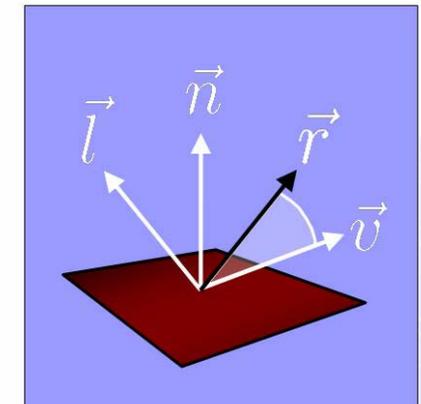
Blinn-Phong Model für Oberflächen



ambient

diffuse

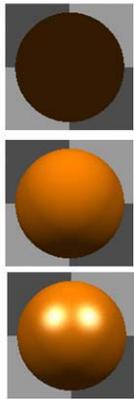
specular



$$I = I_a + I_d (\vec{n} \circ \vec{l}) + I_s (\vec{n} \circ \vec{h})^r$$

Lokale Beleuchtung

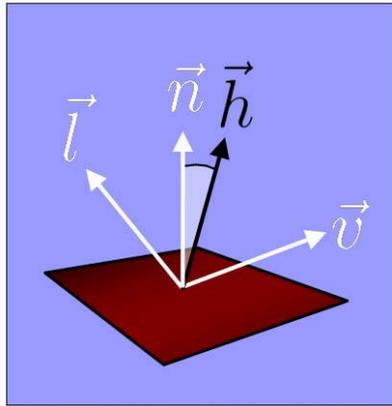
Blinn-Phong Model für Oberflächen



ambient

diffuse

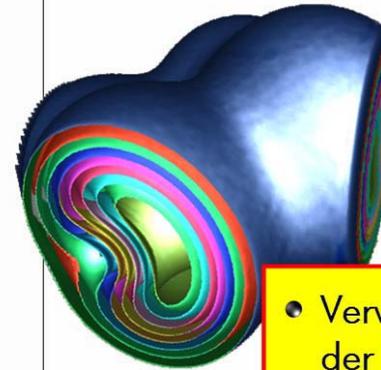
specular



$$I = I_a + I_d (\vec{n} \circ \vec{l}) + I_s (\vec{n} \circ \vec{h})^r$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Beleuchtung für Voxeldaten



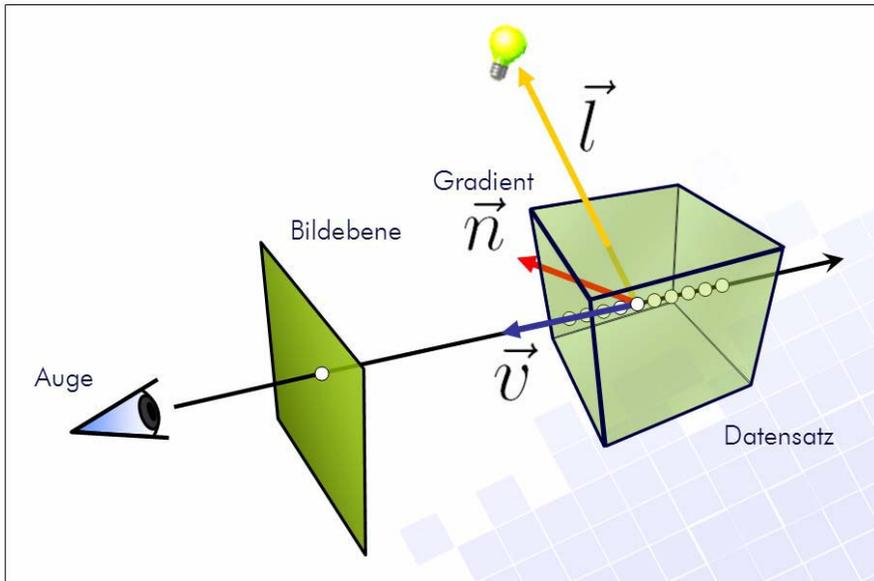
- Das Phong-Model benötigt die Oberflächennormale
- Wie sieht die Oberflächennormale eines Voxels aus?

- Verwende die Oberflächennormale der Isofläche die den Voxel schneidet.

= Gradientenvektor

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Raycasting



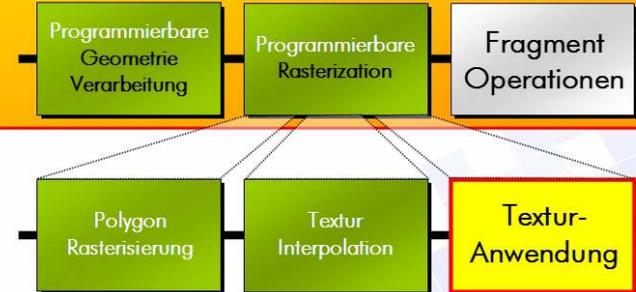
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Texturbasierte Ansätze

- Wir benötigen einen Mechanismus, der es uns erlaubt Skalarprodukte zu berechnen, nachdem die Fragmente interpoliert wurden.

➔ **Solution:** Moderne Graphikhardware

Moderne GPU mit Vertex Shaders und Pixel Shaders:



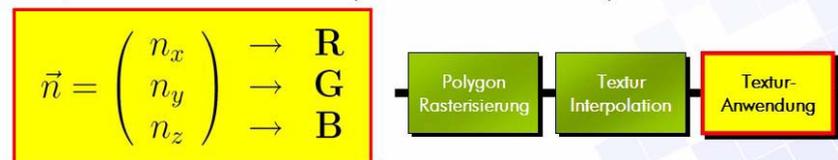
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Beleuchtung für Voxeldaten

41

Grundprinzip für Beleuchtung bei Texturbasierten Verfahren:

- Berechne den Gradientenvektor jedes Voxels vor,
- normalisiere ihn und
- speichere ihn in einer (zusätzlichen Multi-) Textur:



- Bei der *Textur-Interpolation* wird für jedes Fragment ein Normalenvektor interpoliert
- Die Beleuchtungsberechnung (dot products) wird in der *Textur-Anwendung* berechnet.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Einfache Implementierung

42

- 1 Directional Light (Lichtrichtung \vec{l} ist konstant)
- Nur diffuse Reflexion (kein spekularer Anteil)

Dies reduziert das Phong-Modell auf

$$I = I_d (\vec{n} \circ \vec{l})$$

Ausserdem setzen wir noch:

$$I_d = 1$$

Vereinfachter Beleuchtungsterm:

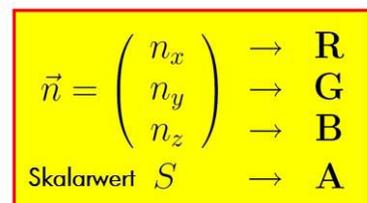
$$I = (\vec{n} \circ \vec{l})$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Implementierungsbeispiel

43

- Darstellung einer beleuchteten Isofläche *ohne vorher die polygonale Oberfläche zu bestimmen.* (Non-polygonal Isosurfaces)

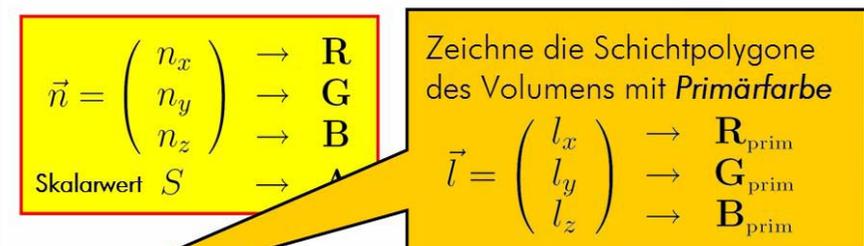


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Implementierungsbeispiel

43

- Darstellung einer beleuchteten Isofläche *ohne vorher die polygonale Oberfläche zu bestimmen.* (Non-polygonal Isosurfaces)

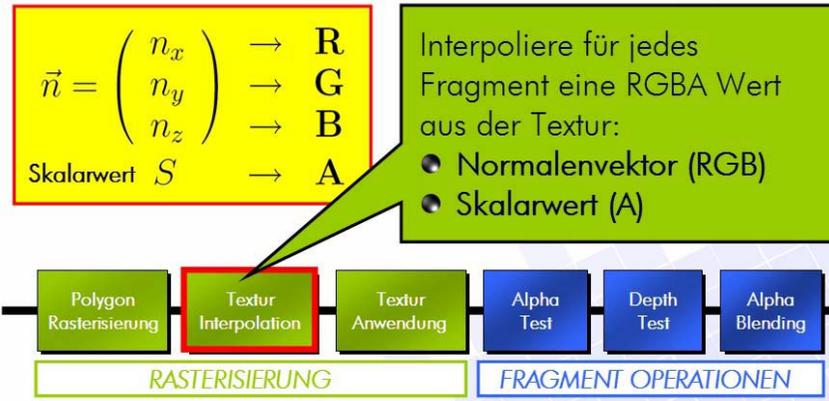


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Implementierungsbeispiel

43

- Darstellung einer beleuchteten Isofläche *ohne* vorher die *polygonale* Oberfläche zu bestimmen. (Non-polygonal Isosurfaces)

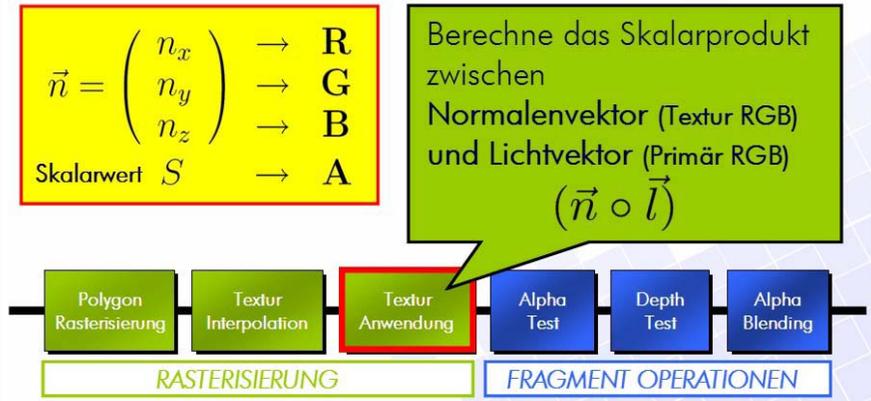


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Implementierungsbeispiel

43

- Darstellung einer beleuchteten Isofläche *ohne* vorher die *polygonale* Oberfläche zu bestimmen. (Non-polygonal Isosurfaces)

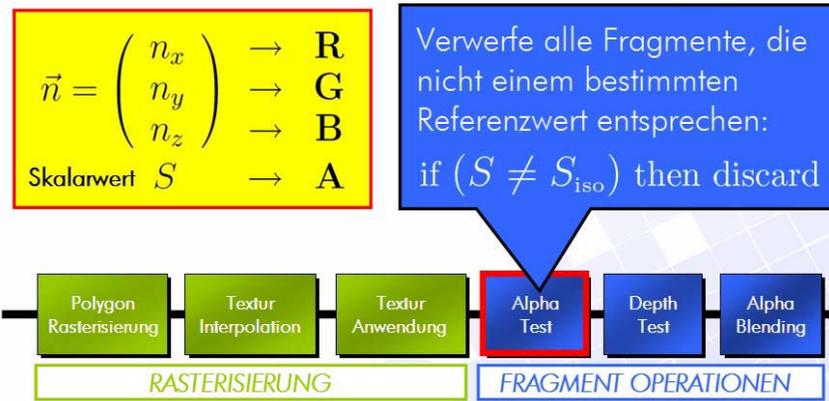


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Implementierungsbeispiel

43

- Darstellung einer beleuchteten Isofläche *ohne* vorher die *polygonale* Oberfläche zu bestimmen. (Non-polygonal Isosurfaces)

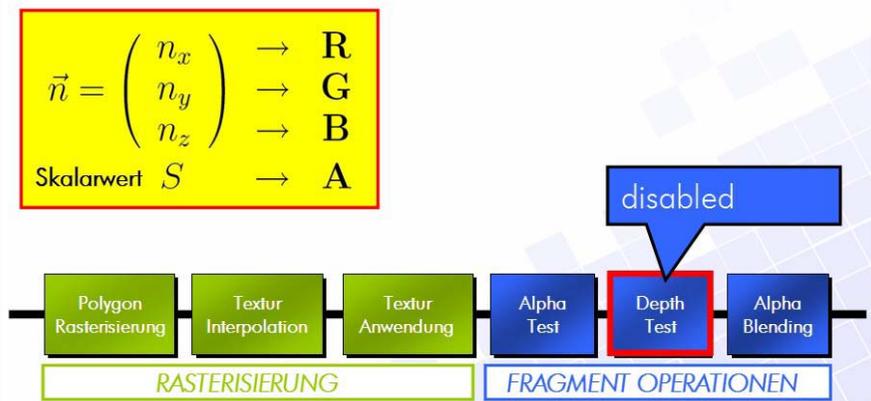


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Implementierungsbeispiel

43

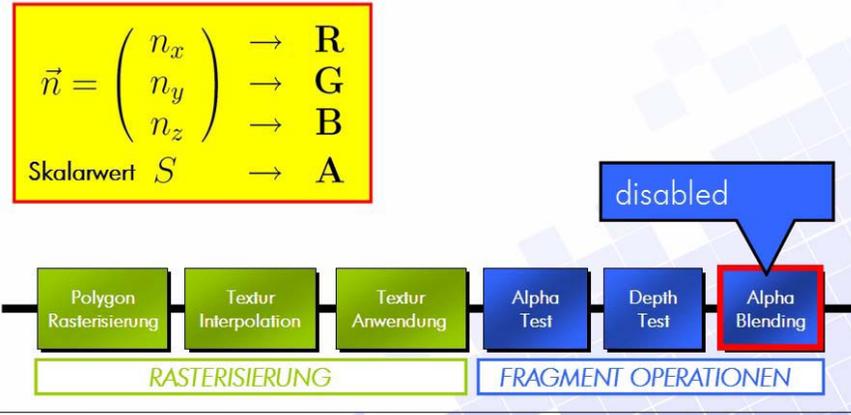
- Darstellung einer beleuchteten Isofläche *ohne* vorher die *polygonale* Oberfläche zu bestimmen. (Non-polygonal Isosurfaces)



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

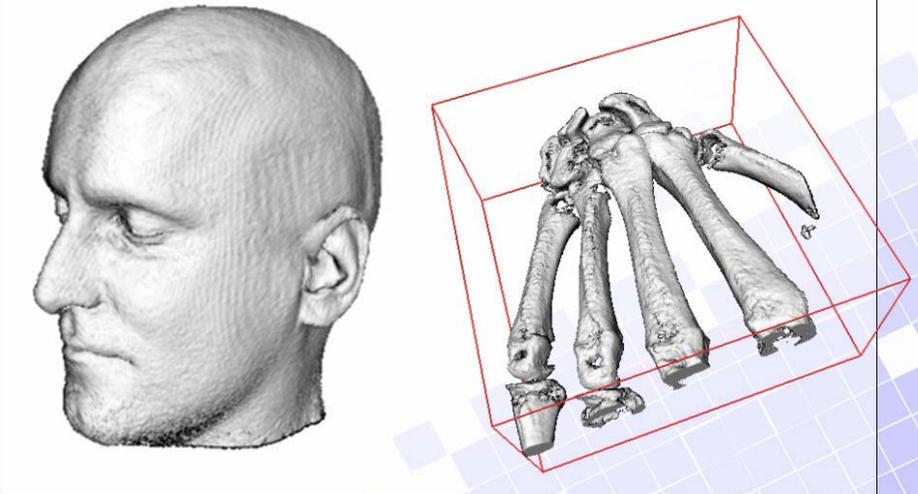
Implementierungsbeispiel

- Darstellung einer beleuchteten Isofläche *ohne* vorher die *polygonale* Oberfläche zu bestimmen. (Non-polygonal Isosurfaces)



Beispiele

Non-polygonal Isosurface (mit diffuser Beleuchtung)



Weitere Möglichkeiten

Emission/Absorption mit lokaler Beleuchtung:

Multitexturen

Jedes Schichtpolygon hat 2 Texturen:

1 Textur für Emission/Absorption $RGBA_{tex0}$

1 Textur für Beleuchtung $RGBA_{tex1}$

Emission

Weitere Möglichkeiten

Emission/Absorption mit lokaler Beleuchtung:

Multitexturen

Jedes Schichtpolygon hat 2 Texturen:

1 Textur für Emission/Absorption $RGBA_{tex0}$

1 Textur für Beleuchtung $RGBA_{tex1}$

Absorption

Weitere Möglichkeiten

45

Emission/Absorption mit lokaler Beleuchtung:

Multitexturen

Jedes Schichtpolygon hat 2 Texturen:

1 Textur für Emission/Absorption $RGBA_{tex0}$

1 Textur für Beleuchtung $RGBA_{tex1}$

Normale

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Weitere Möglichkeiten

45

Emission/Absorption mit lokaler Beleuchtung:

Multitexturen

Jedes Schichtpolygon hat 2 Texturen:

1 Textur für Emission/Absorption $RGBA_{tex0}$

1 Textur für Beleuchtung $RGBA_{tex1}$

unbenutzt

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Weitere Möglichkeiten

45

Emission/Absorption mit lokaler Beleuchtung:

Multitexturen

Jedes Schichtpolygon hat 2 Texturen:

1 Textur für Emission/Absorption $RGBA_{tex0}$

1 Textur für Beleuchtung $RGBA_{tex1}$

Textur-Anwendung:

$$RGB_{final} = RGB_{tex0} + \langle RGB_{prim} \circ RGB_{tex1} \rangle$$

(Emission) (Lichttrichtung) (Normale)

$$A_{final} = A_{tex0} \text{ (Absorption)}$$

Fragment-Operationen:

Alpha-Blending (OVER):

$$RGB_{new} = RGB_{final} \cdot A_{final} + RGB_{old} (1 - A_{final})$$

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Beispiele

46

Emission/Absorption Model mit diffuser Beleuchtung



Emission/Absorption Model
ohne Beleuchtung

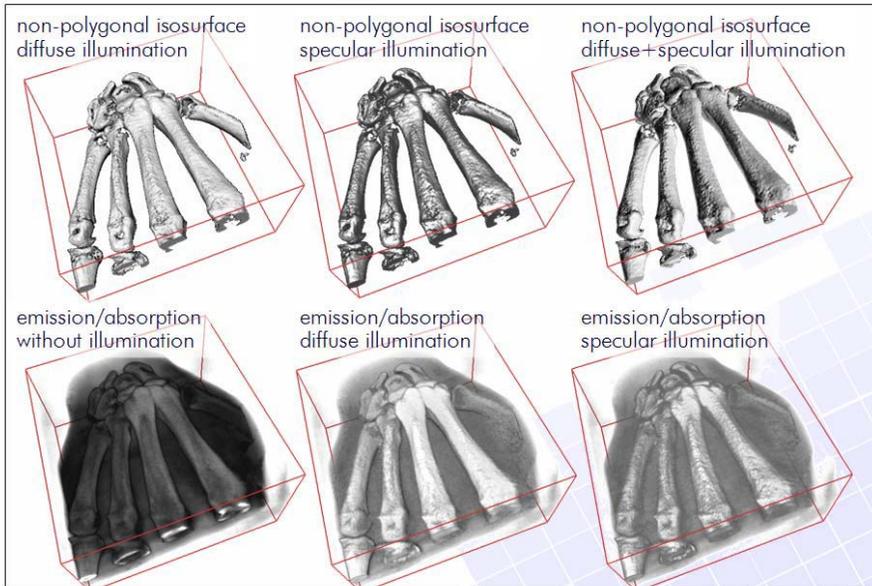


Emission/Absorption Model
mit diffuser Beleuchtung

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Weitere Möglichkeiten

47



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Weitere Möglichkeiten

48

Reflection Mapping:

- Komplexe Beleuchtungsumgebungen vorberechnet



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zusammenfassung

49

Texturbasierte Volumenvisualisierung

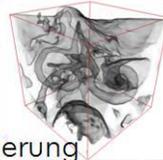
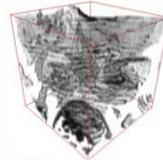
- Klassifikation (Transferfunktionen)

Präklassifikation:

- Farbtabelle vor der Interpolation/Rasterisierung
- Aliasing Artefakte durch hohe Frequenzen in der Transferfunktion.
- Einfache Implementierung

Postklassifikation:

- Farbtabelle nach der Interpolation/Rasterisierung
- Anpassung an hohe Frequenzen in der Transferfunktion durch Erhöhung der Schichtenzahl (Überabtastung).
- Aufwändigere Implementierung



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

Zusammenfassung

50

Texturbasierte Volumenvisualisierung

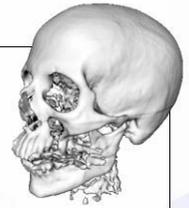
- Lokale Beleuchtung (Shading)

Grundprinzip:

- Vorberechnung des Gradientenvektors als Ersatz für die Oberflächennormale (Blinn-Phong-Modell)
- Speichern in zusätzlicher Textur.
- Beleuchtungsberechnung in Textur-Anwendung

Implementierungsbeispiele:

- Non-polygonal Isosurfaces
- Emission/Absorption mit Beleuchtung



christof rezk-salama, computergraphik und multimediasysteme, universität siegen