

# Texturbasierte Volumenvisualisierung

Christof Rezk-Salama

Visualisierung WS 03/04, 13.01.2004

computergraphik und multimedia systeme  
universität siegen

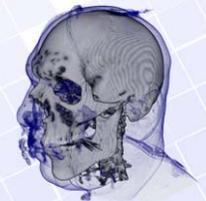
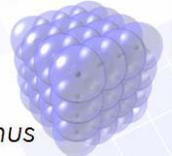
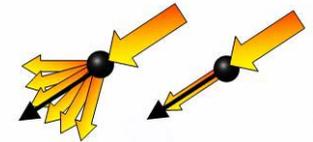


2

## Einführung

### Direkte Volumenvisualisierung

- *Letzte Stunde: Standardverfahren zur Direkten Volumenvisualisierung*
  - Physikalisch basierte Bildsynthese.
  - Bildraum- und Objektraumverfahren
  - *Beispielalgorithmen:*  
*Ray-Casting, Splatting, Shear-Warp-Algorithmus*
- **Fazit:** Direkte Volumenvisualisierung bedeutet hohen Rechenaufwand (Interpolation und numerische Integration)



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Einführung

3

### Direkte Volumenvisualisierung

- *Heute: Hardwarebeschleunigte Verfahren der Volumenvisualisierung*
  - Interaktive bzw. Echtzeitfähige Algorithmen
  - Texturbasierte Verfahren
  - Beispielimplementierungen mit 2D Texturen, 3D Texturen und 2D Multitexturen
  - Keine Spezialhardware (warum nicht?)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Überblick

4

### Hardwarebeschleunigtes Volume Rendering

- **Einführung:** Was kann die Grafik-Hardware?
- **Texturen:** Wie verwende ich Texturen zum Volume Rendering?
  - 2D Texturbasierte Verfahren
  - 3D Texturbasierte Verfahren
  - 2D Multitexturbasierte Verfahren

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

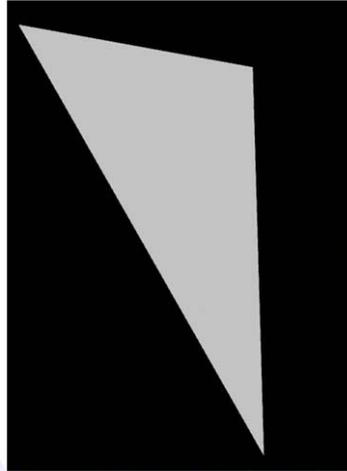
## Was kann die Hardware?

5

Gegeben:

Eckpunkte der Dreiecke

- *Dreiecke zeichnen*
  - Pixel bestimmen



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

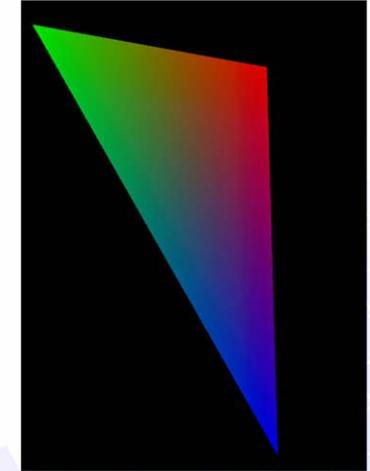
## Was kann die Hardware?

5

Gegeben:

Eckpunkte der Dreiecke

- *Dreiecke zeichnen*
  - Pixel bestimmen
  - Farbe interpolieren



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Was kann die Hardware?

5

Gegeben:

Eckpunkte der Dreiecke

- *Dreiecke zeichnen*
  - Pixel bestimmen
  - Farbe interpolieren
  - Texturen „draufkleben“



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Was kann die Hardware?

5

Gegeben:

Eckpunkte der Dreiecke

- *Dreiecke zeichnen*
  - Pixel bestimmen
  - Farbe interpolieren
  - Texturen „draufkleben“
- *Compositing*
  - Primär-Farbe mit Textur

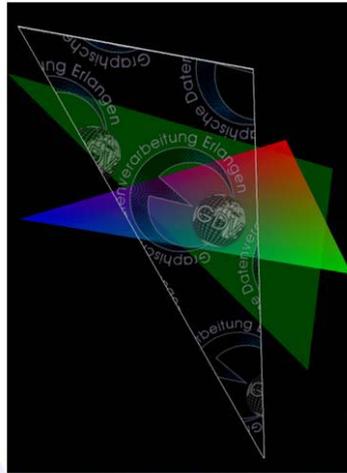


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

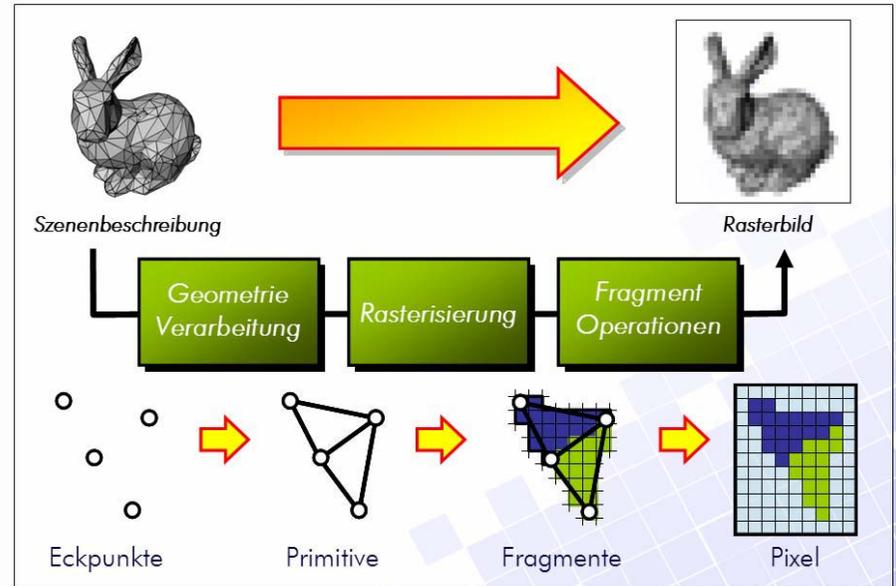
# Was kann die Hardware?

Gegeben:  
Eckpunkte der Dreiecke

- **Dreiecke zeichnen**
  - Pixel bestimmen
  - Farbe interpolieren
  - Texturen „draufkleben“
- **Compositing**
  - Primär-Farbe mit Textur
  - Verdeckung
  - Transparenz

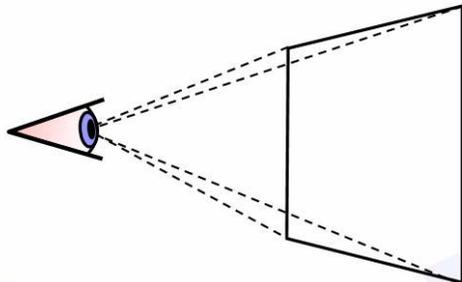


# Rendering Pipeline



# Der Frame-Buffer

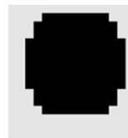
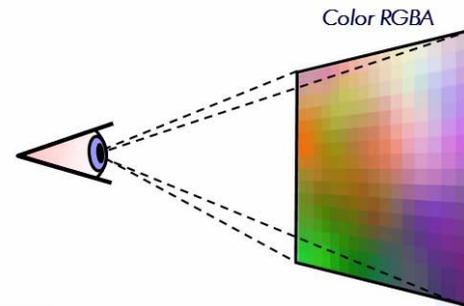
- **Speicherbereich** auf der Grafikkarte, der das darzustellende Bild enthält.



**Color Buffer:** Enthält für jeden Pixel den Farbwert *RGB* und die Opazität *A*.

# Der Frame-Buffer

- **Speicherbereich** auf der Grafikkarte, der das darzustellende Bild enthält.

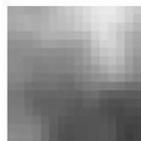
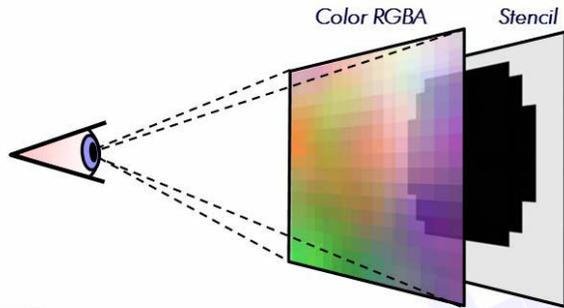


**Stencil Buffer (optional):** „Schablone“, enthält für jeden Pixel ein Flag, ob der Pixel verändert werden darf oder nicht („Schreibschutz“).

# Der Frame-Buffer

7

- *Speicherbereich* auf der Grafikkarte, der das darzustellende Bild enthält.

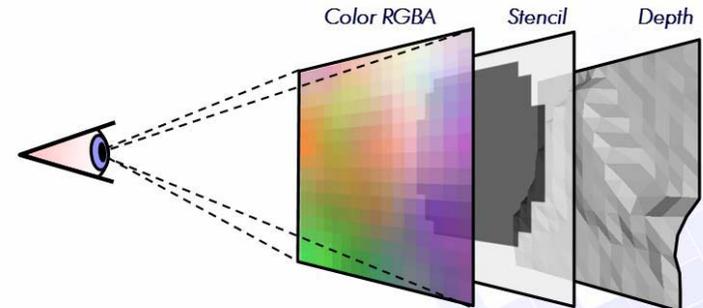


*Depth Buffer:* (auch z-Buffer), speichert für jeden den Tiefenwert, der bei der Projektion auf die Ebene verloren geht.

# Der Frame-Buffer

7

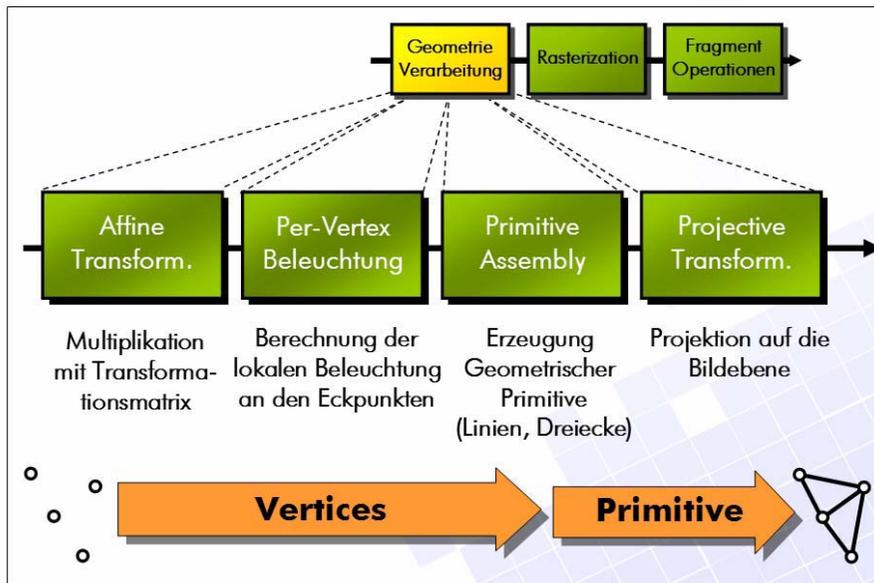
- *Speicherbereich* auf der Grafikkarte, der das darzustellende Bild enthält.



Alpha-, Stencil- und Depth-Buffer werden nur während des Renderings benötigt. Dargestellt wird natürlich nur der RGB-Color Buffer.

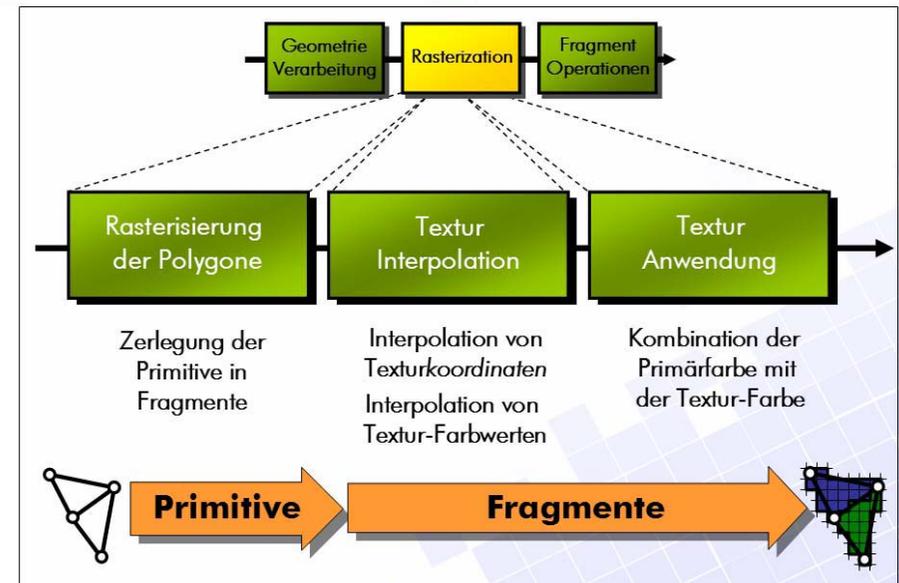
# Geometrieverarbeitung

8



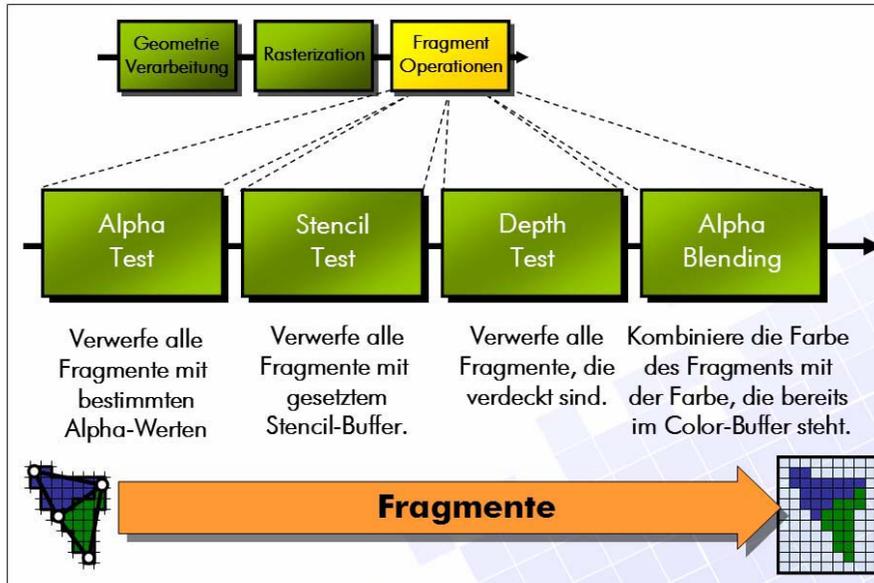
# Rasterisierung

9



# Fragment Operationen

10

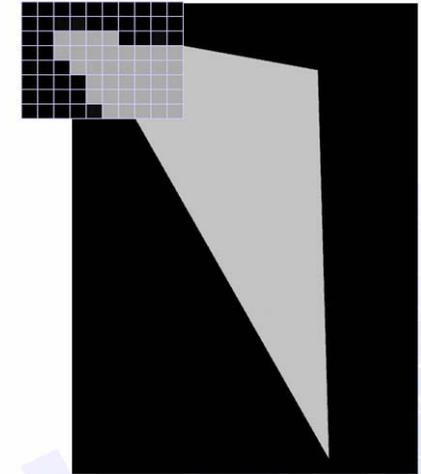


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Was kann die Hardware?

11

- Rasterisierung
  - Zerlegung in Fragmente

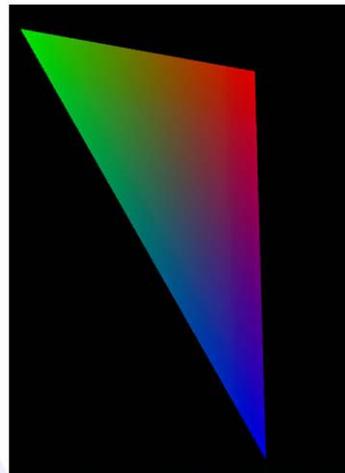


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Was kann die Hardware?

11

- Rasterisierung
  - Zerlegung in Fragmente
  - Interpolation der Farbe



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

# Was kann die Hardware?

11

- Rasterisierung
  - Zerlegung in Fragmente
  - Interpolation der Farbe
  - Texturierung
    - Interpolation (Textur Interpolation)



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Was kann die Hardware?

11

- **Rasterisierung**

- Zerlegung in Fragmente
- Interpolation der Farbe
- Texturierung
  - Interpolation (Textur Interpolation)
  - Kombination (Textur-Anwendung)



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Was kann die Hardware?

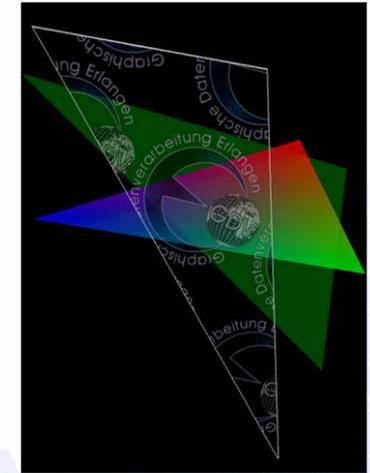
11

- **Rasterisierung**

- Zerlegung in Fragmente
- Interpolation der Farbe
- Texturierung
  - Interpolation (Textur Interpolation)
  - Kombination (Textur-Anwendung)

- **Fragment Operationen**

- Tiefentest (Z-Test)
- Alpha Blending (Compositing)



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Literatur Grafik-Hardware

12

J. Foley, A. van Dam, S. Feiner, and J. Hughes.  
**Computer Graphics, Principle And Practice.**

Addison-Wesley, 1993.

OpenGL ARB.

**The OpenGL Programming Guide (Red Book)**

Addison-Wesley.

Außerdem:

Vorlesung „Computergrafik“

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Volume Rendering in Hardware

13

- **Volume Rendering** (z.B. Ray-Casting) bedeutet eine hohe Anzahl an Interpolationen.
- **Idee:** Nutze die beschleunigte Interpolation der Grafik-Hardware
- **Interpolation** findet (u.a.) in der Texture-Unit statt
- **Problem:** Die Hardware unterstützt keine volumetrischen Primitive (nur ebene Polygone bzw. Dreiecke)

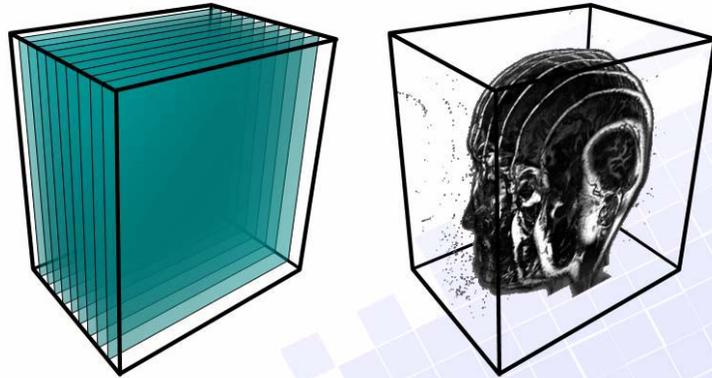
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Texturbasierte Ansätze

14

Hardware unterstützt keine volumetrischen Primitive

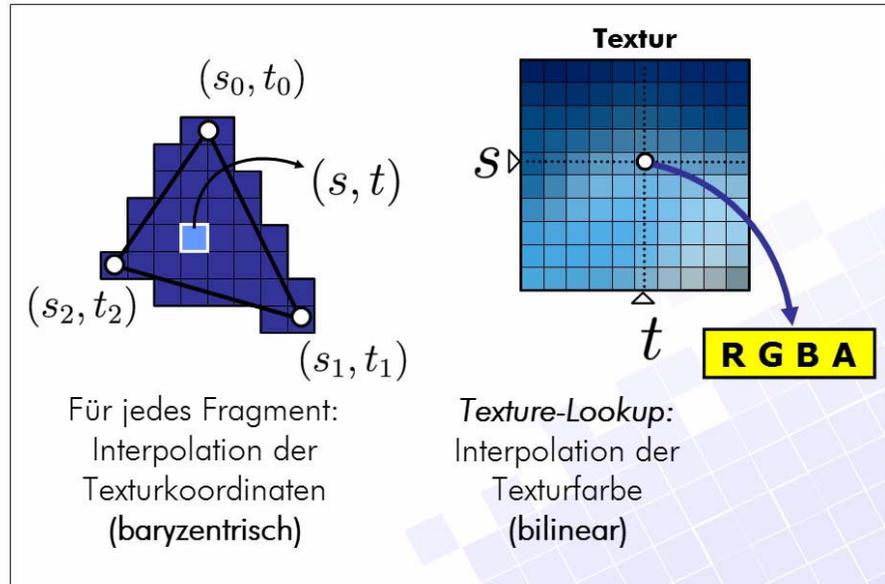
➔ Hilfsgeometrie (*Polygonale Schichten*)



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Wie funktionieren Texturen?

15



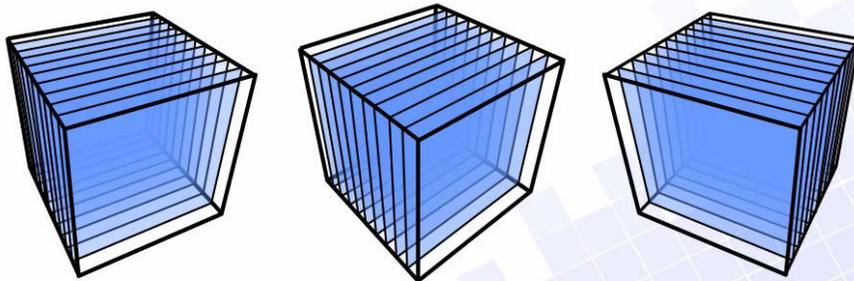
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 2D Texturen

16

● Bilineare Interpolation in Hardware

➔ Zerlegung des Volumens in achsen-parallele Schichten (*analog Shear Warp*)



● 3 Kopien des Datensatzes im Speicher!

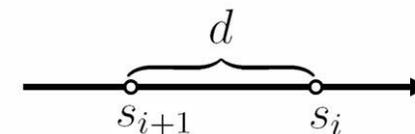
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 2D Texturen

17

● Was wird in der Textur gespeichert?

**RGB A** Emissionsterm  $RGB = \int_{s_i}^{s_i+d} q(s) ds$   
Absorptionsterm  $A = e^{-\tau(s_i, s_i+d)}$



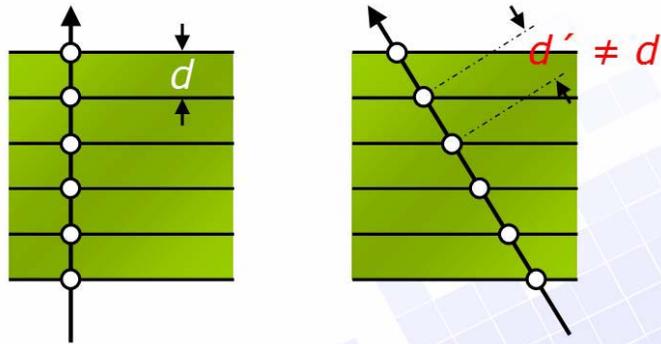
Die RGBA Werte, die in durch, die Transferfunktion spezifiziert und in die Textur geschrieben werden, beziehen sich auf eine bestimmten Abstand  $d$  der Abtastpunkte.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 2D Texturen

18

Für die numerische Integration (over-operator) muß der Abstand  $d$  konstant sein.



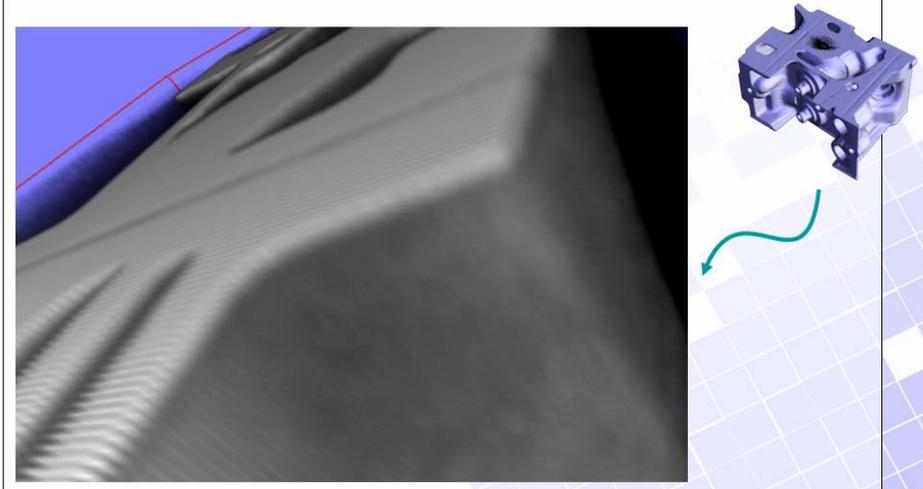
- Inkorrekte (diskrete) Emission/Absorption
- Keine Überabtastung (Erhöhung der Schichtenzahl) möglich!

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 2D Texturen

19

- Nachteil – Visuelle Bildartefakte (bei starkem Zoom)

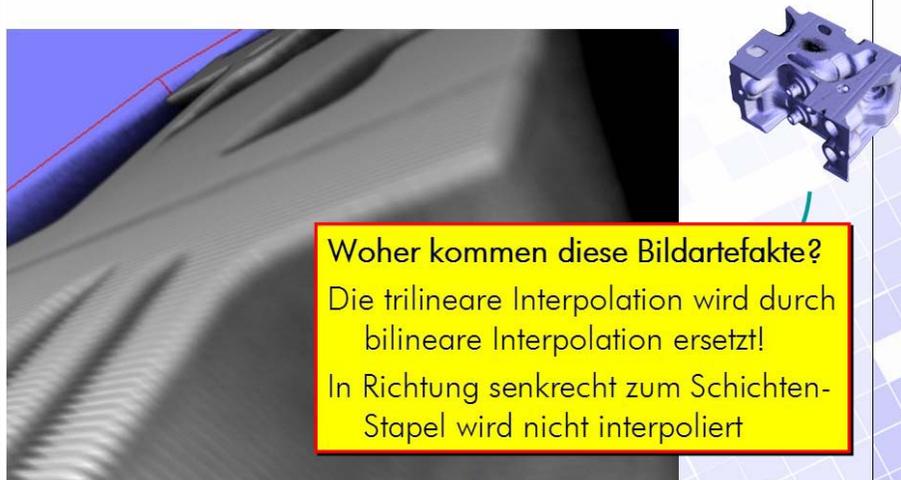


christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 2D Texturen

19

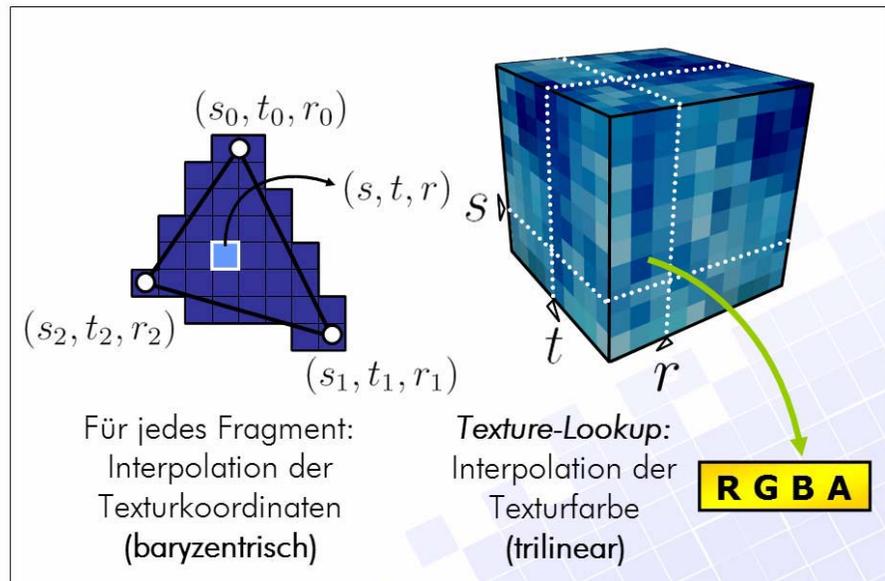
- Nachteil – Visuelle Bildartefakte (bei starkem Zoom)



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 3D Texturen

20



christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 3D Texturen

21

- 3D Textur: Volumetrisches Textur-Objekt
- Aber kein volumetrisches Rendering-Primitiv!

➔ Schichten parallel zur Bildebene



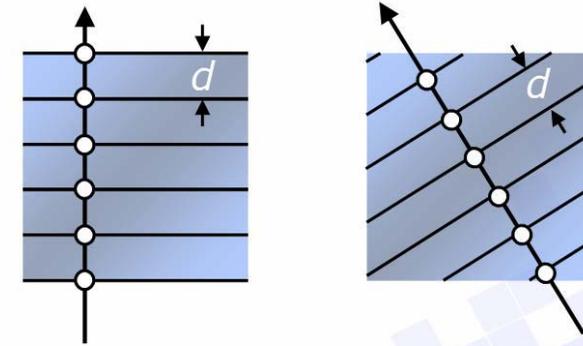
- Das Volumen ist als ein großer Texturblock im Speicher

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Resampling bei 3D Texturen

22

- Abtastrate ist konstant (für Parallelprojektion)



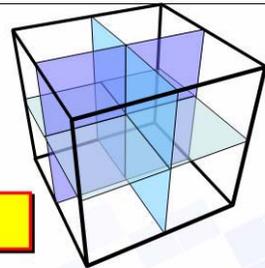
- Überabtastung einfach durch Erhöhung der Schichtenzahl.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Bricking

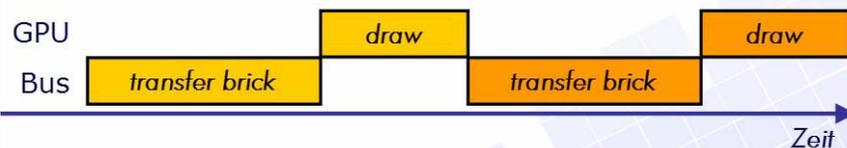
23

- Was ist, wenn der Datensatz zu groß ist für den Grafik-Speicher?
- ➔ Teile den Datensatz in kleinere Subvolumina (Bricks)



### Problem: Bus-Bandbreite

- Schlechte Auslastung von GPU und Bus



- GPU und Bus warten aufeinander (idle wait)

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

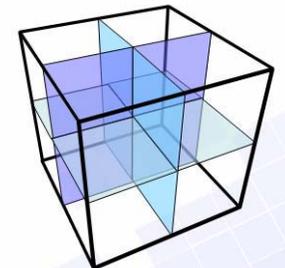
## Bricking

24

- Schlechte Auslastung von GPU und Bus

### Mögliche Lösungen:

- Mache die Bricks klein genug! Mehr als ein Brick muß in den Grafikspeicher passen!
- Transfer und Rendering können parallel ausgeführt werden.
- Erhöhte CPU Last für Schnittberechnung!
- Effektives Load Balancing ist sehr schwierig!



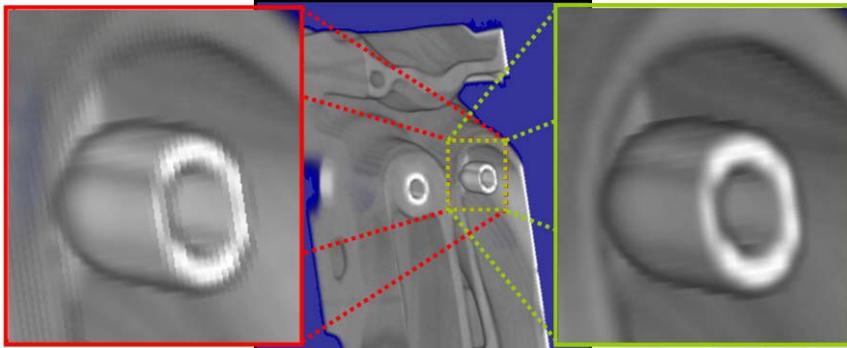
### Verwende 2D Multi-Texturen anstatt 3D Texturen!

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## Zurück zu 2D Texturen

25

Schneller als 3D Texturen, aber mindere Bildqualität



Visuelle Artefakte wegen fester Schichtenzahl

Bei höhere Abstrakte verschwinden diese Artefakte

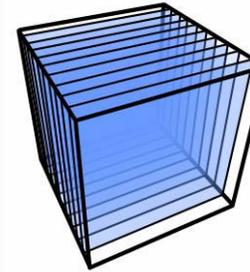
christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 2D Multi-Texturen

26

- Idee: Verwende achsenparallele Schichten
- ➔ Zerlegung des Volumens in achsenparallele Schichten

an beliebigen Positionen



- Interpoliere die Textur für diese Schicht aus zwei benachbarten Texturen

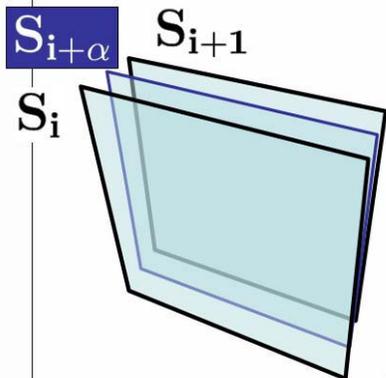
- 3 Kopien des Datensatzes im Speicher!

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 2D Multi-Textures

27

Achsenparallele Schichten



- Bilineare Interpolation durch 2D Textur

- Blending zweier Schichten (Texturkombination)

$$S_{i+\alpha} = (1 - \alpha)S_i + \alpha \cdot S_{i+1}$$

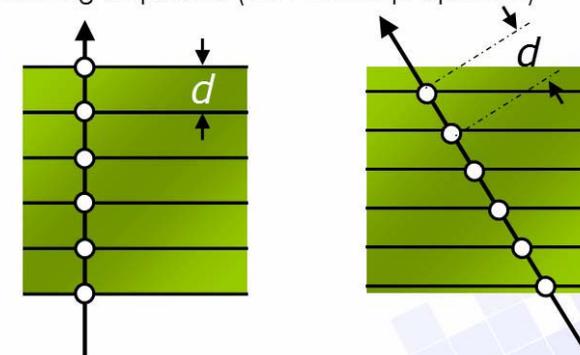
- Ergebnis: *Trilineare Interpolation*

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## 2D Multi-Textures

28

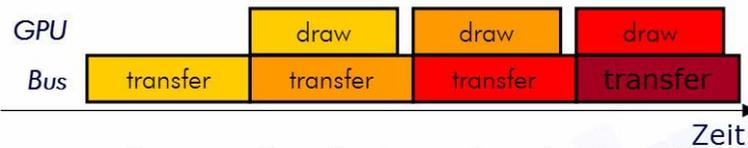
- Abtastrate ist konstant wenn ich den Schichtabstand an die Blickrichtung anpasse! (für Parallelprojektion)



- Überabtastung ist einfach durch Erhöhung der Schichtenzahl. Textur wird on-the-fly interpoliert.

christof rezk-salama, computergraphik und multimediasysteme, universität siegen

## ● Effizienteres Load-Balancing:



- Nutze GPU und verfügbare Speicherbandbreite parallel!
- Transferiere pro Schicht die minimale Information die nötig ist um diese Schicht zu zeichnen (feine Granularität)!
- Erheblich höhere Performanz, obwohl 3 Kopien des Datensatzes im Speicher sind!

B. Cabral, N. Cam, and J. Foran.  
*Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware.*  
*ACM Symposium on Volume Visualization, 1994.*

O. Wilson, A. Van Gelder, and J. Wilhelms.  
*Direct Volume Rendering via 3D Textures.*  
Technical Report UCSC-CRL-94-19, Univ. of California, Santa Cruz, 1994.

C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl.  
*Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Textures and Multi-Stage Rasterization.*  
*Proceedings of the SIGGRAPH/Eurographics Workshop on Graphics Hardware, 2000.*

## Grafik-Hardware

- Geometrieverarbeitung
- Rasterisierung
- Fragment Operationen

## Texturbasierte Verfahren

- Zerlege das Volumen in Schichten (Hilfsgeometrie)
- 2D Texturen (schnell, aber geringe Qualität)
- 3D Texturen (schnell, gute Qualität, aber Probleme mit großen Datensätzen!)
- 2D Multi-Texturen (schnell, gute Qualität)

## Klassifikation und Shading

- Präklassifikation
- Postklassifikation
- Lokale Beleuchtung für Volumina

