Echtzeit–Videoverarbeitung

Erzeugung von 3D Darstellungen

Herbert Thoma

Gliederung

3D Transformationen

Sichtbarkeit von 3D Objekten

Beleuchtung

Texturen

Beispiele: SGI Infinite Reality 2, PC 3D Hardware

3D Transformationen 1

Darstellung von 3D Koordinaten durch 4D homogene Koordinaten, Manipulation durch 4×4 Matrizen

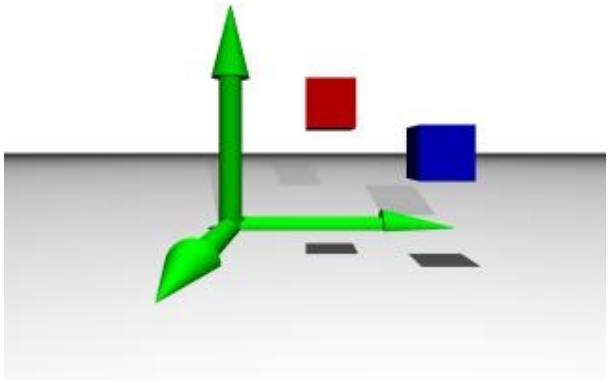
$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \text{ und } \begin{bmatrix} W \cdot x \\ W \cdot y \\ W \cdot z \\ W \end{bmatrix} \text{ repräsentieren den selben Punkt}$$

3D Transformationen 2

Translation

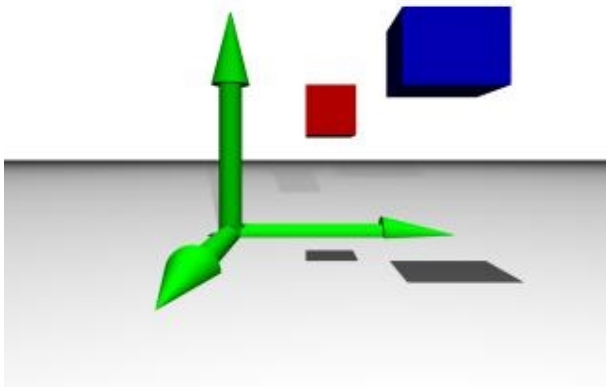
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+t_x \\ y+t_y \\ z+t_z \\ 1 \end{bmatrix}$$



3D Transformationen 3

Skalierung

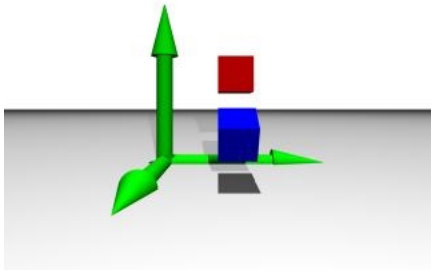
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \cdot s_x \\ y \cdot s_y \\ z \cdot s_z \\ 1 \end{bmatrix}$$

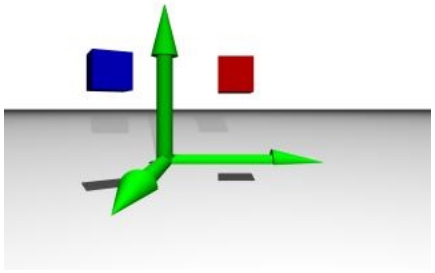


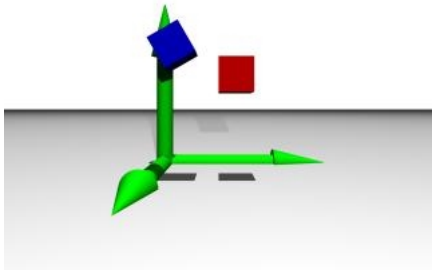
3D Transformationen 4

Rotation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



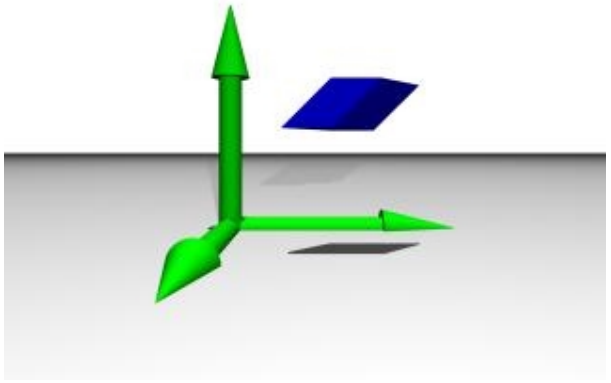




3D Transformationen 5

Scherung

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + z \cdot sh_x \\ y + z \cdot sh_y \\ z \\ 1 \end{bmatrix}$$



3D Transformationen 6

Hintereinanderausführung von Transformationen
➔Multiplikation der Transformationsmatrizen

Zu beachten:
- Transformationen beziehen sich auf den Ursprung
- Reihenfolge der Transformationen

Allgemeine Form der Transformationsmatrix

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D Transformationen 7

Echtzeitimplementierungen von 3D Transformationen

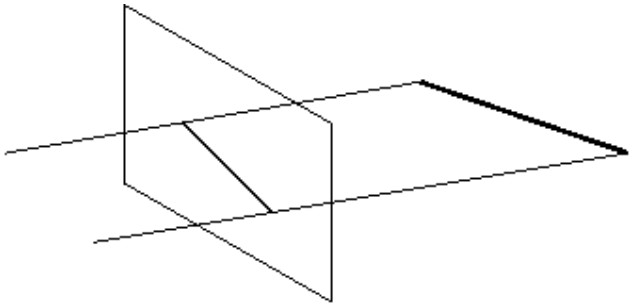
- Fließkommaberechnungen
- Grundoperation: Skalarprodukt von 4D Vektoren

➔SIMD Architekturen mit 4 parallelen Fließkommaoperationen
  - ➔ ISSE
  - ➔ AltiVec

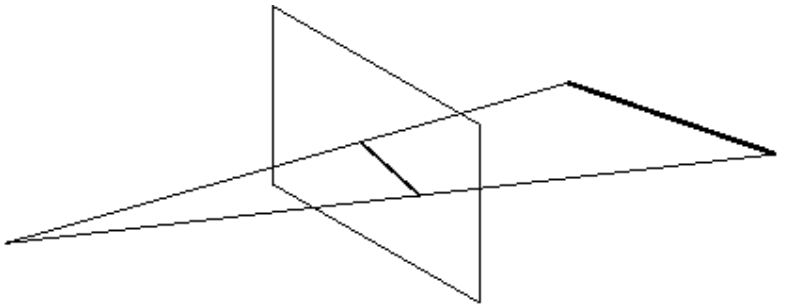
Perspektivische Projektion 1

Abbildung der 3D Welt auf eine Ebene

Parallelprojektion



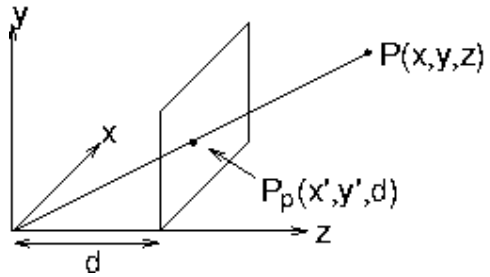
Zentralprojektion

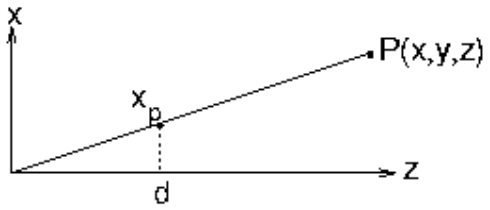


Perspektivische Projektion 2

Mathematische Beschreibung der Zentralprojektion

- Projektionszentrum im Ursprung
- Projektionsebene senkrecht zur z−Achse, Abstand d vom Ursprung





$$\frac{x_p}{d} = \frac{x}{z} \qquad \frac{y_p}{d} = \frac{y}{z}$$

$$x_p = \frac{x}{z/d} \qquad y_p = \frac{y}{z/d}$$

Perspektivische Projektion 3

Matrixdarstellung der Zentralprojektion

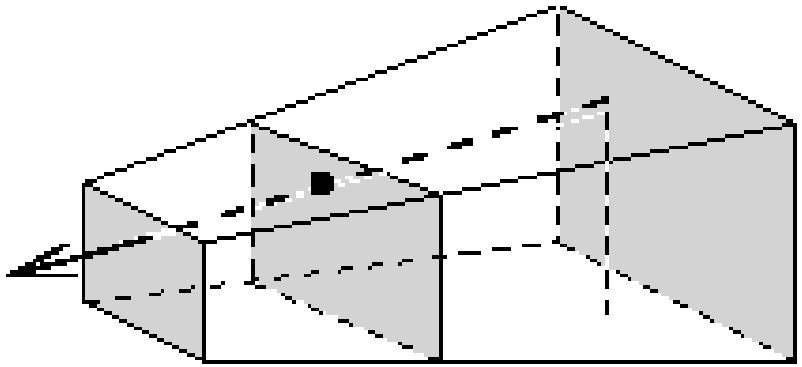
$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \frac{z}{d} \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \\ 1 \end{bmatrix}$$

Sichtbarkeit 1

Bildraum

Parallelprojektion: Quader

Zentralprojektion: Pyramidenstumpf



Objekte außerhalb des Bildraums sind nicht sichtbar

Sichtbarkeit 2

Bestimmung sichtbarer Linien / Flächen

- Backface Culling: Flächen, deren Normalen vom Betrachter wegzeigen, sind nicht sichtbar
- Depth Sort: Objekte werden nach ihrer Entfernung vom Betrachter sortiert und von hinten nach vorne gezeichnet
- Visible Surface Ray−Tracing: Für jedes Pixel wird der Schnittpunkt der Gerade vom Projektionszentrum duch den entsprechenden Punkt der Projektions−ebene mit jedem Objekt berechnet. Der dem Betrachter am nächsten liegende Punkt wird gezeichnet
- Z−Buffer

Sichtbarkeit 3

Z–Buffer

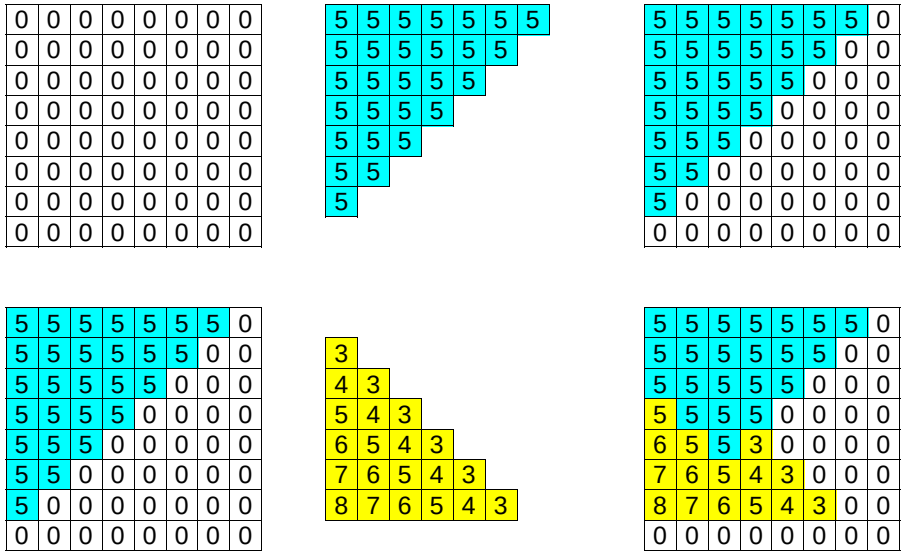
- Tiefeninformation pro Pixel wird gespeichert
- Pixel wird überschrieben, wenn aktuelles Objekt näher am Betrachter ist als schon gezeichnetes Pixel

Vorteile:

- Einfache Implementierung, auch in Hardware
- Unabhängig von Zeichenreihenfolge

Nachteile:

- Zusätzlicher Speicherbedarf
- Pixel werden oft mehrmals überschrieben (Overdraw)



Beleuchtung 1

Beleuchtungsgleichung

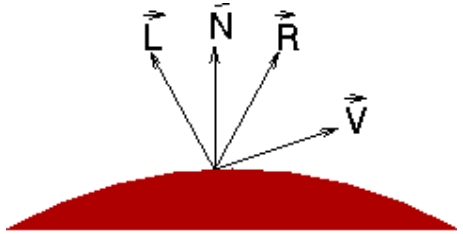
$$I_\lambda = I_{a\lambda} k_a O_{a\lambda} + \sum_{i=1}^{m} f_{att_i} I_{p\lambda_i} \left[k_d O_{d\lambda} \left(\vec{N} \cdot \vec{L}_i \right) + k_s O_{s\lambda} \left(\vec{R}_i \cdot \vec{V} \right)^n \right]$$

$\vec{L}_i$ Vektor zur Lichtquelle

$\vec{N}$ Oberflächennormale

$\vec{R}_i$ Vektor des reflektierten Lichtstrahls

$\vec{V}$ Vektor zum Betrachter



I_λ Lichtintensität (wellenlängenabhängig (RGB))

$I_{a\lambda} k_a O_{a\lambda}$ Ambient Light (Grundhelligkeit)

$f_{att_i} I_{p\lambda_i} k_d O_{d\lambda}$ Diffuse Reflection (Oberfläche als Lambertstrahler)

$f_{att_i} I_{p\lambda_i} k_s O_{s\lambda}$ Specular Reflection (Glanzlichter)

Nicht berücksichtigt:

- Schatten
- Spiegelungen
- Lichtbrechung bei transparenten Körpern

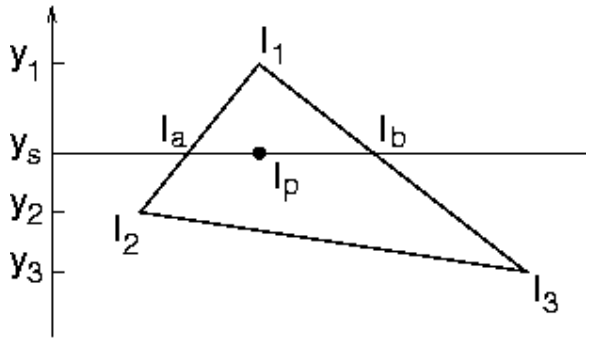
Beleuchtung 2

Polygonshading

- Objektoberflächen werden duch Polygone (meist Dreiecke) angenähert

Beleuchtungsmodelle

- Flat Shading: Eine Beleuchtungsberechnung pro Polygon
- Gouraud Shading: Beleuchtungsberechnung an den Ecken der Polygone, Interpolation im Inneren der Polygone
- Phong Shading: Oberflächennormalen werden über die Polygone interpoliert, Beleuchtungsberechnung an jedem Pixel



$$I_a = I_1 - \left(I_1 - I_2\right)\frac{y_1 - y_s}{y_1 - y_2}$$

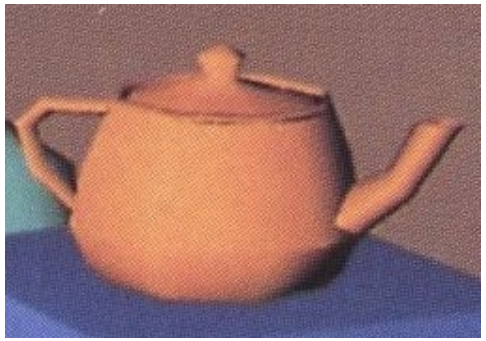
$$I_b = I_1 - \left(I_1 - I_3\right)\frac{y_1 - y_s}{y_1 - y_3}$$

$$I_p = I_b - \left(I_b - I_a\right)\frac{x_b - x_p}{x_b - x_a}$$

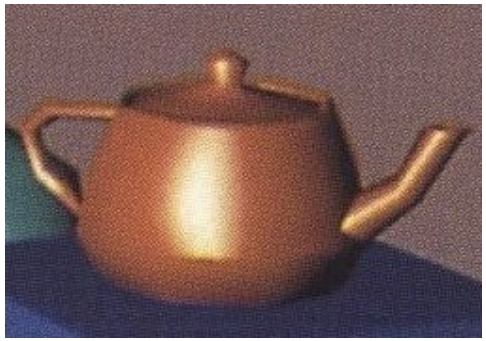
Beleuchtung 3

Polygonshading

flat



gouraud

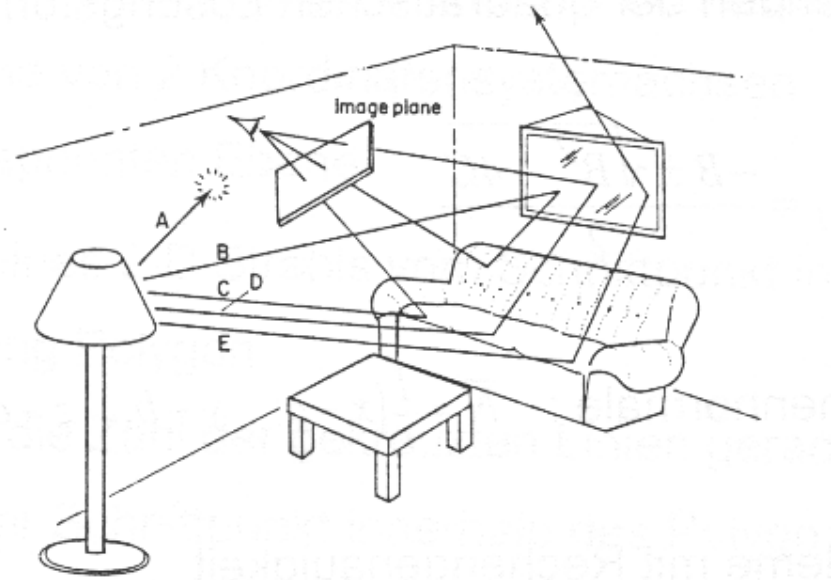


phong

Beleuchtung 4

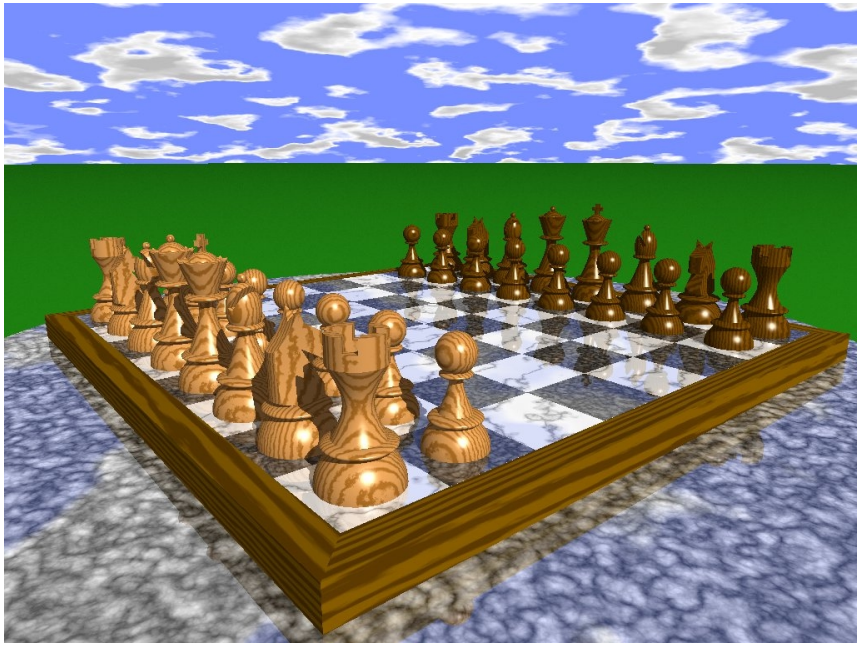
Raytracing

- Aussendung von Strahlen vom Projektionszentrum durch die Bildebene
- Ein Augenstrahl pro Pixel
- Jeden Strahl mit jedem Objekt auf einen Schnitt testen
- Aussendung von Sekundärstrahlen
  - Spiegelungsstrahlen
  - Brechungsstrahlen
  - Schattenstrahlen
- Sehr realistische Darstellungen
- Sehr großer Rechenleistungsbedarf



Beleuchtung 5

Raytracing Beispiele



Beleuchtung 5

Radiosity

- Radiosity: Energiedichte [W/m^2]
- Voraussetzung:
  - Alle Flächen reflektieren lambert−diffus
  - Alle Lichtquellen sind Lambertstrahler
- Zerlegung aller Objekte in Flächenstücke, sog. Patches
- Aufstellen des Energiegleichgewichts für jedes Patch

$$B_i = E_i + \rho_i \sum_{j=1}^{N} B_i F_{ij}$$

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1N} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_2 F_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_N F_{N1} & -\rho_N F_{N2} & \cdots & 1-\rho_N F_{NN} \end{bmatrix} \cdot \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix}$$

Texturen 1

Prozedurale Texturen

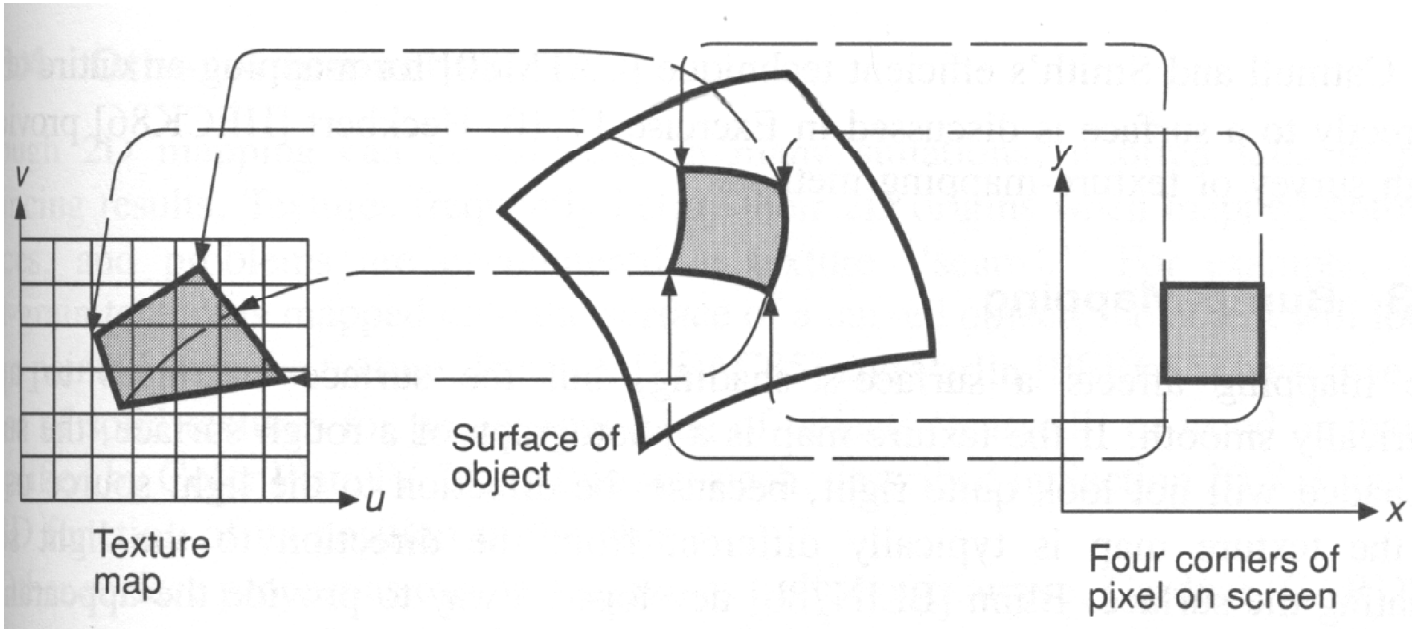
Funktion der 3D Koordinaten bestimmt die Oberflächen–struktur eines Objekts

- Sehr hoher Realismus möglich
- Großer Rechenleistungsbedarf

Texturen 2

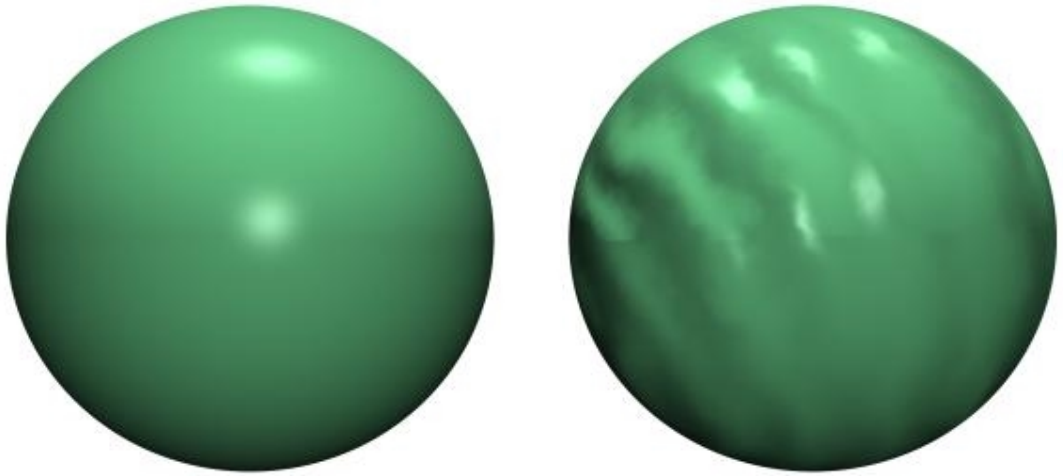
Texture Mapping

- Projektion eines Bildes auf die Oberfläche eines Objektes
- Texturfilterung
  - bilinear
  - MIP mapping: Verschiedene Auflösungsstufen
  - Perspektivische Verzerrung



Texturen 3

Bump Mapping: Variation des Normalenvektors zur Simulation von Oberflächenstruktur



SGI Infinite Reality 2

High End Grafiksubsystem für SGI Onyx2 Rechner

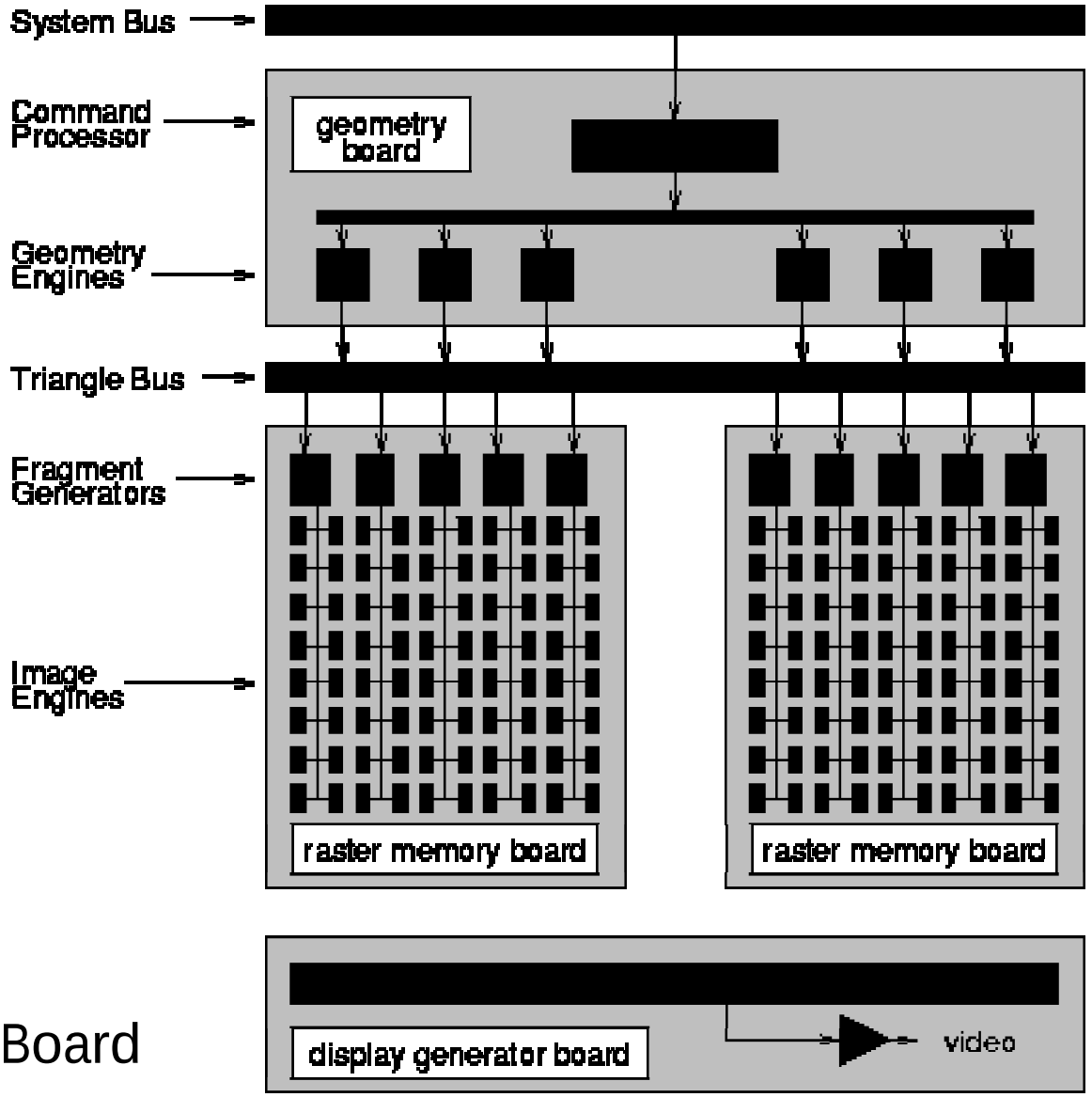
Anwendungen
- Visualisierung
- Simulationen
- TU Ilmenau: Virtuelles Studio

SGI Infinite Reality 2

1 Geometry Board

1, 2 oder 4 Raster Memory Boards

1 Display Generator Board



SGI Infinite Reality 2

Geometry Board

- Direkte Verarbeitung von OpenGL Anweisungen
- Transformationen
- Beleuchtungsberechnung

- 4 ASICs mit je 3 SIMD Einheiten

SGI Infinite Reality 2

Raster Memory Board

- Gouraud Shading
- Z–Buffer
- Texture Mapping
- Antialiasing

- 80 MB Framebufferspeicher
  - 256, 512 oder 1024 bits pro Pixel
  - 1, 2 oder 4 Pixelbuffer mit je 8 oder 12 bit für RGBA
  - 24 bit oder 32 bit Z–Buffer
  - 4, 8 oder 16 Subsamples für Antialiasing
- 16 MB oder 64 MB Texturspeicher

SGI Infinite Reality 2

Display Generator Board

- Digital / Analog Umsetzung
- Bis zu 8 unabhängige Ausgänge
- Maximal 1920 x 1200 @ 66 Hz
- Genlock
- Video Input für Video to Texture

SGI Infinite Reality 2

Performance

13.1 Mio Polygone/sec

224 Mio bis 896 Mio Pixel/sec, gouraud shaded, z−buffer

192 Mio bis 768 Mio Pixel/sec, textured, anti−aliased, z−buffer

PC 3D Grafikhardware

1. Generation

- Gouraud Shading
- Z−Buffer
- Texturemapping
- z. B.: S3 Virge 3D, Voodoo1

2. Generation

- Triangle setup
- Mehrfachtexturen
- z. B.: nVidia TNT, Voodoo2,3,5

3. Generation

- Transformation und Beleuchtungsberechnung (T&L)
- z. B.: nVidia GeForce, ATI Radeon

PC 3D Grafikhardware

Speicheranforderungen

1024 x 768 Pixel, 32 bit, 32 bit Z−Buffer, 32 bit Texturen MIPmapped,
Framerate 40 fps, Overdraw 2.5, Bildwiederholfrequenz 75 Hz

- Framebuffer schreiben:
  1024 x 768 x 32 bit = 3 MByte x 2.5 x 40 fps = 300 MByte/s
- Z−Buffer lesen/schreiben:
  1024 x 768 x 32 bit = 3 MByte x (2 x 2.5) x 40 fps = 600 MByte/s
- Texturen lesen:
  8 Texel/Pixel = 24 MByte x 2.5 x 40 fps = 2400 MByte/s
- Framebuffer auslesen:
  1024 x 768 x 32 bit = 3 MByte x 75 Hz = 225 MByte/s

- Summe: **3525 MByte/s**

- Vergleich: 133 MHz DDR SDRAM, 128 bit Busbreite
  133 MHz x 2 x 16 Byte = 4256 Mbyte/s