



Computergraphik I

Die Graphik-Pipeline

G. Zachmann
 Clausthal University, Germany
zach@tu-clausthal.de

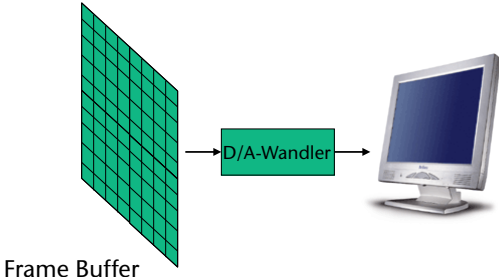
Der Input: Graphische Primitive

- Alle Geometrie wird aus folgenden **graphischen Primitiven** zusammengesetzt:
 - Punkte
 - Einzelne Strecken (= "Geraden" oder "Linien")
 - Linienzug (= Folge von zusammenhängenden Linien)
 - Dreiecke (evtl. konvexe Polygone)
 - Buchstaben (vordefinierte Zeichen)
- Das sind alle!
 - Kurven? → werden aus einer Folge von Linien approximiert
 - Allgemeine Polygone? → werden i.A. in Dreiecke unterteilt
 - Gekrümmte Flächen? → werden i.A. durch Dreiecke approximiert
- Der Trend geht in Richtung einfacher Primitive
 - Einfach, einheitlich, sich oft wiederholend: gut für Parallel-Verarbeitung

G. Zachmann Computer-Graphik 1 – WS 10/11 Die Graphik-Pipeline 2

Der Output: ein Array von Farbwerten

- Auf der Graphikkarte haben wir einen eigenen Bildspeicher
- Hier schreiben wir letzten Endes einzelne Pixel
- Ein Digital/Analog-Wandler liest diesen periodisch aus und wandelt den Inhalt in das Bildsignal für den Monitor um
- Typisches Format: 3 Bytes pro Pixel (r,g,b)



Frame Buffer

D/A-Wandler

G. Zachmann Computer-Graphik 1 – WS 10/11 Die Graphik-Pipeline 3

Übersicht über die Pipeline

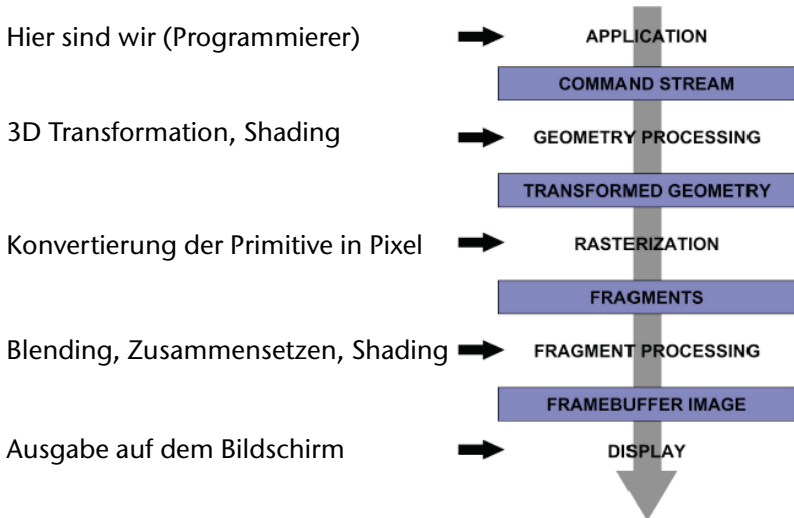
Hier sind wir (Programmierer) → APPLICATION

3D Transformation, Shading → GEOMETRY PROCESSING

Konvertierung der Primitive in Pixel → RASTERIZATION

Blending, Zusammensetzen, Shading → FRAGMENT PROCESSING

Ausgabe auf dem Bildschirm → DISPLAY



APPLICATION

COMMAND STREAM

GEOMETRY PROCESSING

TRANSFORMED GEOMETRY

RASTERIZATION

FRAGMENTS


FRAGMENT PROCESSING

FRAMEBUFFER IMAGE

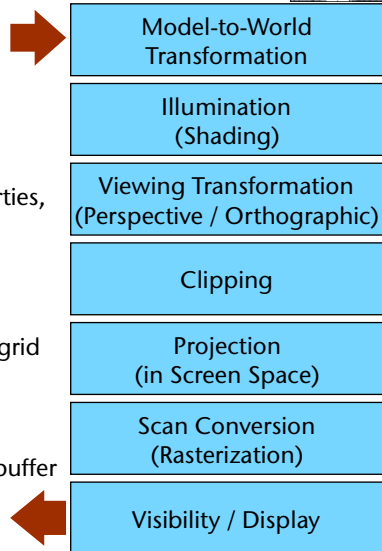
DISPLAY

G. Zachmann Computer-Graphik 1 – WS 10/11 Die Graphik-Pipeline 4

"A Trip Down the Graphics Pipeline"



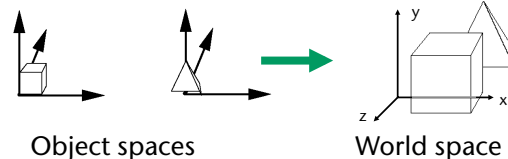
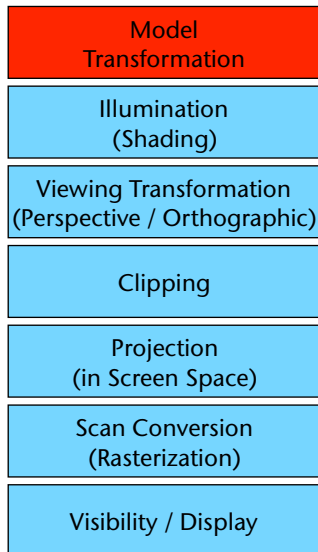
- Input:
 - Geometric model: description of all objects, surface, and light source geometry and transformations
 - Lighting model: Computational description of object and light properties, interaction (reflection)
 - Synthetic Viewpoint (or Camera): Eye position and viewing frustum
 - Raster Viewport (Frame Buffer): Pixel grid onto which image plane is mapped
- Output:
 - Colors / Intensities suitable for framebuffer display (for example, 24 Bits RGB values at each pixel)



G. Zachmann Computer-Graphik 1 – WS 10/11 Die Graphik-Pipeline 5

Model Transformation (model-to-world)

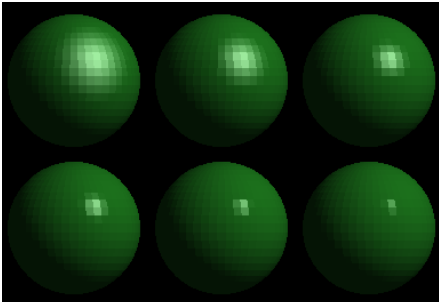
- 3D Modell wird im eigenen Koordinatensystem definiert (**object space**)
- Model Transformation** positioniert die Objekte in einem allg. Koordinatensystem (**world space**)

G. Zachmann Computer-Graphik 1 – WS 10/11 Die Graphik-Pipeline 6

Illumination (Beleuchtung, Schattierung)

- Beleuchten von Dreiecken (für Schattierung und Highlights) gemäß der Material-Eigenschaften, Oberflächeneigenschaften und Lichtquellen
- Lokale Beleuchtungsmodelle (Diffuse, Ambient, Phong, etc.)

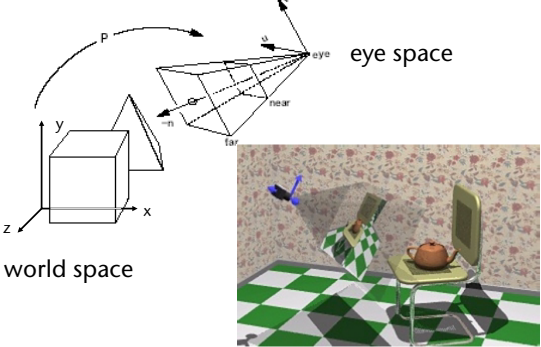


Model Transformation
Illumination (Shading)
Viewing Transformation (Perspective / Orthographic)
Clipping
Projektion (in Screen Space)
Scan Conversion (Rasterization)
Visibility / Display

G. Zachmann Computer-Graphik 1 – WS 10/11
Die Graphik-Pipeline 7

Viewing Transformation

- Umwandeln von Welt-Koord. nach Kamera-Koord.
- Bestimme Transformation für komplette Szene so, daß Betrachter-Position in den Ursprung verschoben wird und Blickrichtung entlang (-Z)-Achse ist



Model Transformation
Illumination (Shading)
Viewing Transformation (Perspective / Orthographic)
Clipping
Projection (in Screen Space)
Scan Conversion (Rasterization)
Visibility / Display

G. Zachmann Computer-Graphik 1 – WS 10/11
Die Graphik-Pipeline 8

Clipping

- Abschneiden der Polygone, die außerhalb des sichtbaren Bereiches liegen (*view frustum*)
- Transformiere Szene in **normalisierte Koordinaten (NDC)**

The diagram illustrates the clipping process. On the left, a purple cow is shown within a 3D view frustum. The frustum is defined by 'near' and 'far' planes. A green arrow points to the right, where the cow is shown within a normalized coordinate system (NDC) box. The NDC box is a unit cube with axes x, y, and z.

eye space

NDC

Modell Transformation

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

Clipping

Projektion (in Screen Space)

Scan Conversion (Rasterization)

Visibility / Display

G. Zachmann Computer-Graphik 1 – WS 10/11 Die Graphik-Pipeline 9

Projektion

- Das Objekt wird in ein 2D Bild (*space*) projiziert

The diagram illustrates the projection process. On the left, a purple cow is shown within a 3D NDC box. A red arrow points to the right, where the cow is shown within a 2D screen space grid. The screen space grid has axes 'width' and 'height'. Below the NDC box, a detailed view shows the cow within a 3D eye space frustum, with axes labeled 'top', 'bottom', 'left', 'right', 'near', and 'far'.

NDC

Screen Space

screen space

Modell Transformation

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

Clipping

Projection (in Screen Space)

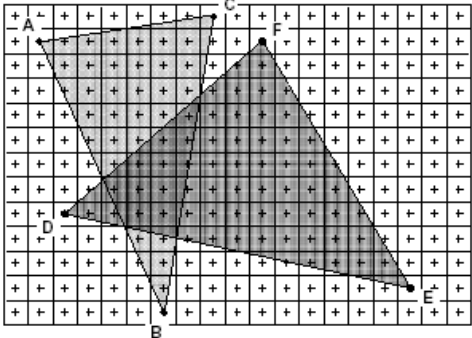
Scan Conversion (Rasterization)

Visibility / Display

G. Zachmann Computer-Graphik 1 – WS 10/11 Die Graphik-Pipeline 10

Scan Conversion (Rasterisierung)

- Rasterisierung der Polygone in Pixel
- Ecken-Werte interpolieren (Farbe, Tiefenwert, ...)



Modell Transformation

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

Clipping

Projektion (in Screen Space)

Scan Conversion (Rasterization)

Visibility / Display

G. Zachmann Computer-Graphik 1 – WS 10/11
Die Graphik-Pipeline 11

Visibility (Sichtbarkeit) u.a. Tests

- Verdeckungen bestimmen
- Evtl. weitere Pixel-Operationen:
 - Blending mit vorhandenem Frame-Buffer-Inhalt
 - Maskierung (z.B. wegen Verdeckung durch andere Fenster)
 - Farb-Transfer

Modell Transformation

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

Clipping

Projektion (in Screen Space)

Scan Conversion (Rasterization)

Visibility / Display

G. Zachmann Computer-Graphik 1 – WS 10/11
Die Graphik-Pipeline 12