Grafikpipeline

Visual Computing

Winter Semester 2023-2024

TECHNISCHE UNIVERSITÄT DARMSTADT

Prof. Dr. A. Kuijper

Mathematical and Applied Visual Computing (MAVC)
Graphisch-Interaktive Systeme (GRIS)
Fraunhofer IGD
Fraunhoferstrasse 5
D - 64283 Darmstadt

E-Mail: office@gris.tu-darmstadt.de
http://www.gris.tu-darmstadt.de
https://www.mavc.tu-darmstadt.de

Lena revisited

TECHNISCHE UNIVERSITÄT DARMSTADT

https://www.wired.com/story/finding-lena-the-patron-saint-of-jpegs/#

https://onezero.medium.com/a-nude-playboy-photo-has-been-a-mainstay-in-testing-tech-for-decades-b8cdb434dce1

(Graustufen-) Bild

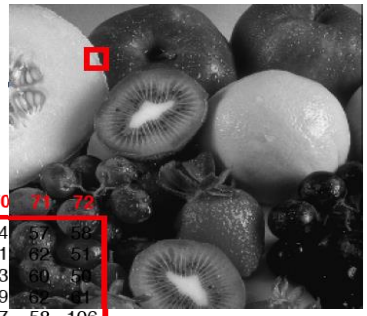
TECHNISCHE UNIVERSITÄT DARMSTADT

Ziele der Computer Vision

- Wie kann man Obst in einem Raster von (graustufen) Zahlen erkennen?
- Wie kann man Tiefen-informationen aus einem Raster von (graustufen) Zahlen herleiten?
- …

Ziele der Computergrafik

- Wie kann man ein Raster von (graustufen) Zahlen erzeugen, das wie Obst aussieht?
- Wie kann man ein Raster von (graustufen) Zahlen erzeugen, sodass der menschliche Betrachter Tiefe wahrnimmt?
- …



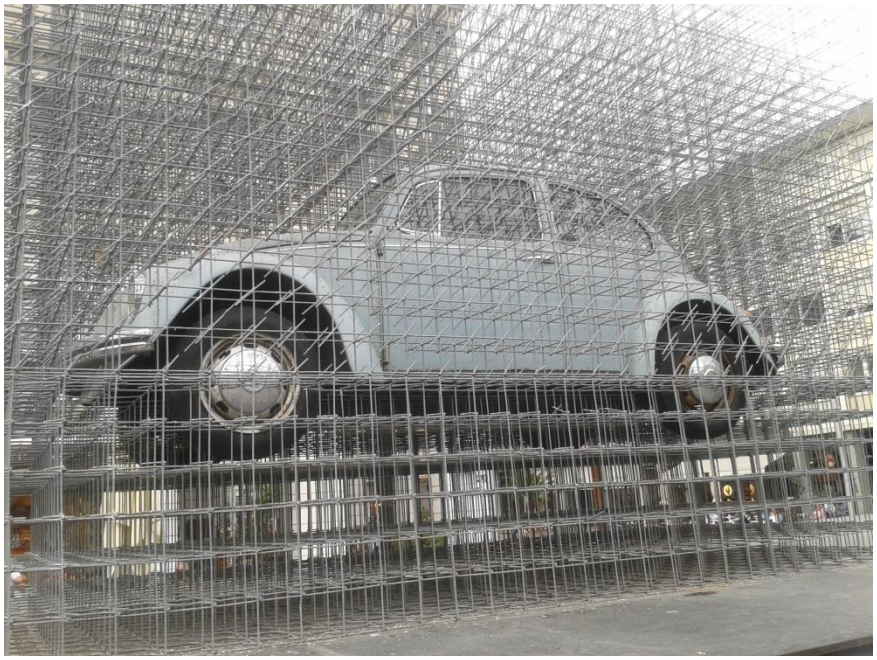
y =	x = 58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
41	210	209	204	202	197	247	143	71	64	80	84	54	54	57	58
42	206	196	203	197	195	210	207	56	63	58	53	53	61	62	51
43	201	207	192	201	198	213	156	69	65	57	55	52	53	60	54
44	216	206	211	193	202	207	208	57	69	60	55	77	49	67	41
45	221	206	211	194	196	197	220	56	63	60	55	46	97	58	106
46	209	214	224	199	194	193	204	173	64	60	59	51	62	56	48
47	204	212	213	208	191	190	191	214	60	62	66	76	51	49	55
48	214	215	215	207	208	180	172	188	69	72	55	49	56	52	56
49	209	205	214	205	204	196	187	196	86	62	66	87	57	60	48
50	208	209	205	203	202	186	174	185	149	71	63	55	55	45	56
51	207	210	211	199	217	194	183	177	209	90	62	64	52	93	52
52	208	205	209	209	197	194	183	187	187	239	58	68	61	51	56
53	204	206	203	209	195	203	188	185	183	221	75	61	58	60	60
54	200	203	199	236	188	197	183	190	183	196	122	63	58	64	66
55	205	210	202	203	199	197	196	181	173	186	105	62	57	64	63

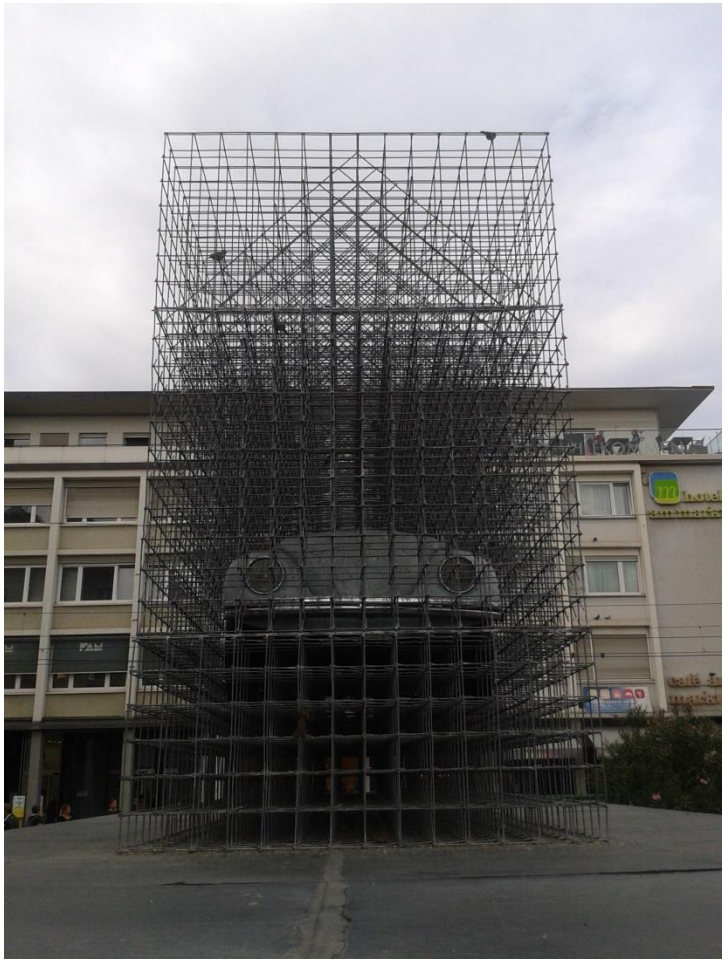
computer vision =
das Problem der
„umgekehrten Grafik“?

MATHEMATICAL AND APPLIED VISUAL COMPUTING

Real <-> Virtuell

TECHNISCHE UNIVERSITÄT DARMSTADT





Auah!

NO TIME TO DIE Trailer

https://www.youtube.com/watch?v=BIhNsAtPbPI

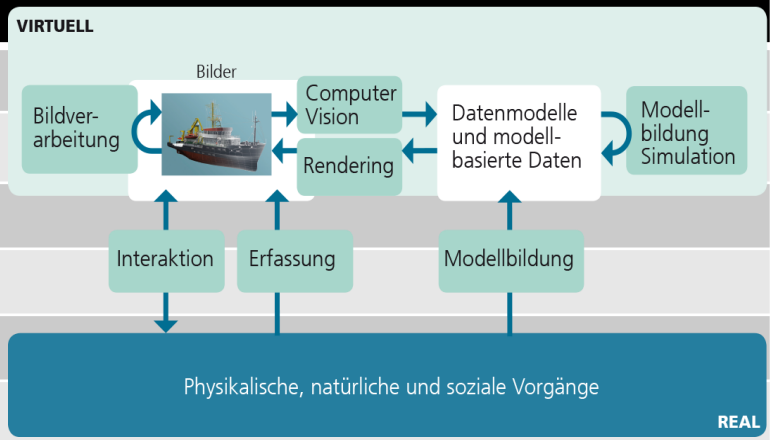
TECHNISCHE UNIVERSITÄT DARMSTADT

MATHEMATICAL AND APPLIED VISUAL COMPUTING

Semesterplan

TECHNISCHE UNIVERSITÄT DARMSTADT

Datum	
20. Okt	Einführung + Visual Computing
27. Okt	Wahrnehmung
03. Nov	Objekterkennung und Bayes
10. Nov	Fourier Theorie
17. Nov	Bilder
24. Nov	Bildverarbeitung
01. Dez	Grafikpipeline & Eingabemodalitäten & VR+AR
08. Dez	Transformationen & 2D/3D Ausgabe
15. Dez	3D-Visualisierung
12. Jan	X3D – 3D in HTML
19. Jan	Informationsvisualisierung
26. Jan	Farbe
02. Feb	User Interfaces + Multimedia Retrieval
11. Feb	Biometrie



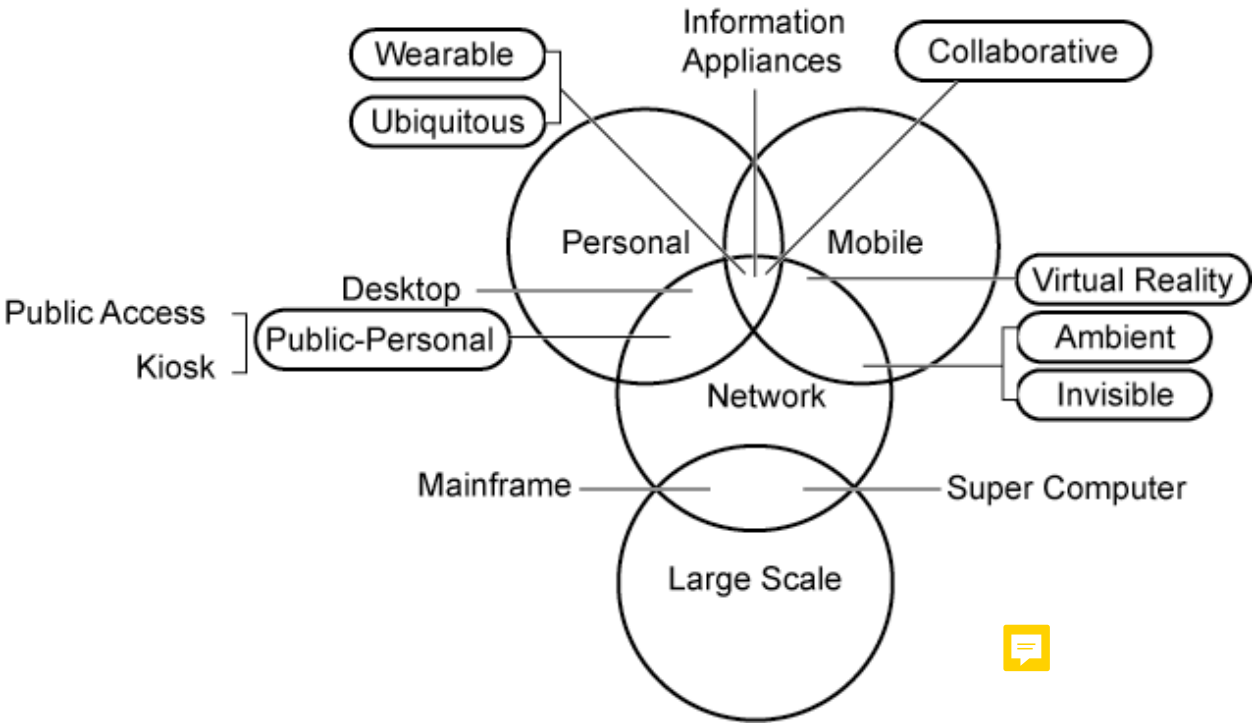
MATHEMATICAL AND APPLIED VISUAL COMPUTING

Überblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Hardware**
- Computergrafik
  - Geschichte
- Grafikpipeline
  - Anwendung
  - Geometrieverarbeitung
  - Rasterisierung
  - Ausgabe

Computing: Hardware & Paradigmen



Kreise

- **Grundlegende Paradigmen.**

Längliche Formen

- Zusammenlaufende Paradigmen.

Nicht umrandete Ausdrücke

- Konkrete Systemarchitekturen (manchmal als Verweis auf ein Paradigma verwendet, wie z.B. in „Desktop Computing“, das für „Personal Computing“ steht).

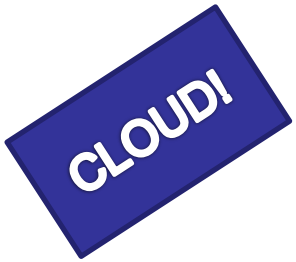
P1: Large-Scale-Computing

TECHNISCHE UNIVERSITÄT DARMSTADT

Die ursprünglichen Mainframe-Computer waren riesige Rechenmaschinen, die als Hosts angesteuert wurden. Sie befanden sich fest in Rechenzentren.

Der Zugriff fand über externe, mit Tastaturen ausgerüstete, alphanumerische Terminals statt („Dumb Terminals“)

→ Host- und Terminal-Systeme



4381 Processor, IBM (1983)

zSeries 990, IBM (2003)

MATHEMATICAL AND APPLIED VISUAL COMPUTING

P2: Personal Computing / Desktop Computing

Xerox Alto

- Entwickelt 1973 im Xerox Palo Alto Research Center
- Erster GUI-basierter Computer, der die „Desktop-Metapher“ verwendete
  - Popup-Menüs
  - Fenster
  - Icons
  - ...

Dell XPS One (2008)

P3: Networked Computing

TECHNISCHE UNIVERSITÄT DARMSTADT

J.C.R. Licklider – Intergalactic Computer Network

ARPAnet – 22:30 am 29. Oktober, 1969

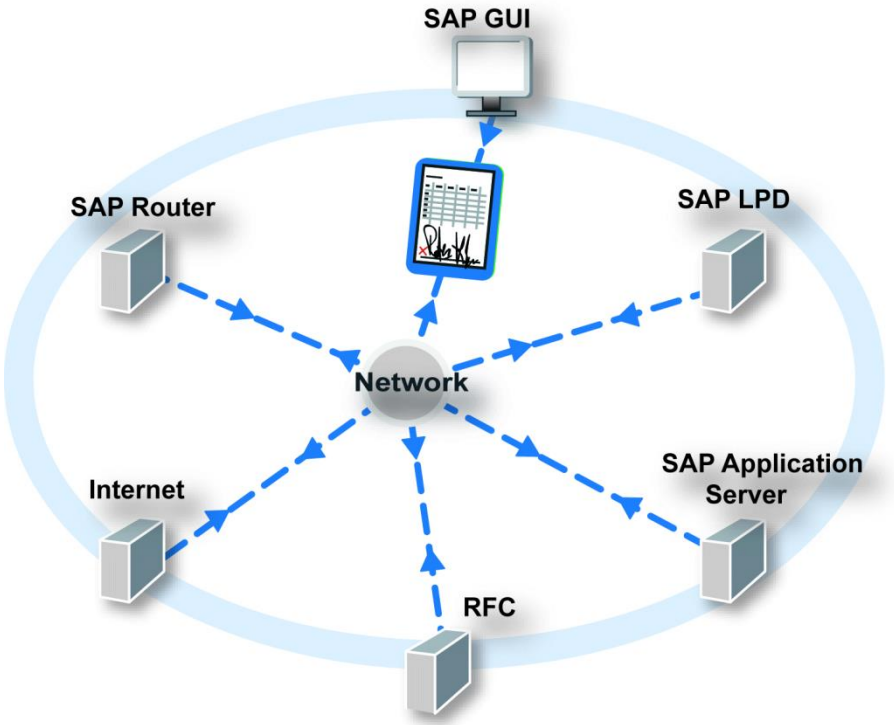
Geltungsbereiche

- WAN – Wide Area Network
- MAN – Metropolitan Area Network
- LAN – Local Area Network
- PAN – Personal Area Network

Wired - Wireless

- Wi-Fi (IEEE 802.11x)
- Bluetooth
- xG

Virtual Network Computing



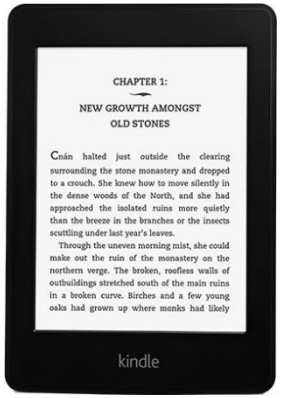
MATHEMATICAL AND APPLIED VISUAL COMPUTING

P4: Mobile Computing

TECHNISCHE UNIVERSITÄT DARMSTADT

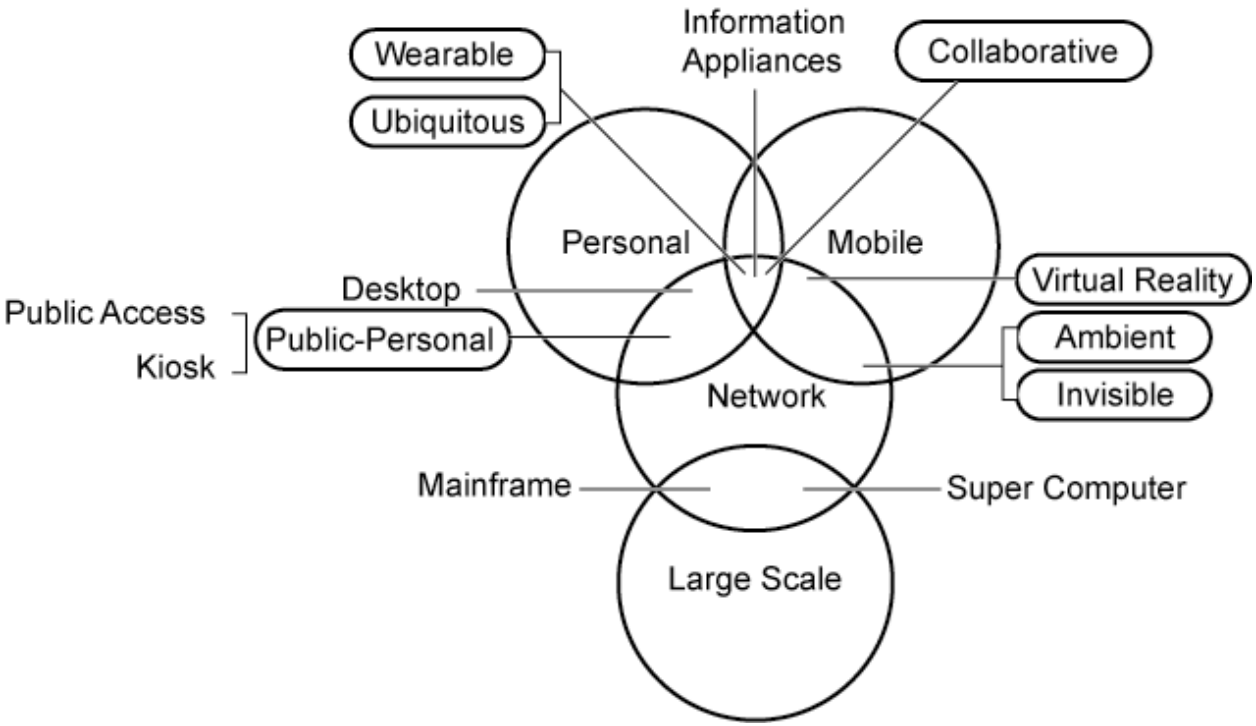
Es gibt viele verschiedene Arten von mobilen Geräten:

- Laptops
- Tablets
- Smartphones
- Spielekonsolen
- Musikplayer
- E-Reader
- ...



MATHEMATICAL AND APPLIED VISUAL COMPUTING

Computing: Hardware & Paradigmen



Kreise

- Grundlegende Paradigmen.

Längliche Formen

- **Zusammenlaufende Paradigmen.**

Nicht umrandete Ausdrücke

- Konkrete Systemarchitekturen (manchmal als Verweis auf ein Paradigma verwendet, wie z.B. in „Desktop Computing“, das für „Personal Computing“ steht).

ZP1: Collaborative computing

TECHNISCHE UNIVERSITÄT DARMSTADT

Multi-touch tables

…

Startseite > Region > Darmstadt

DARMSTADT

| TT Schrift: - + |

26. Juni 2013 | von Daniel Klose

Kompromiss am Flachbildschirm

Fraunhofer-IGD – Institut für Grafische Datenverarbeitung entwickelt auch Techniken zur digitalen Bürgerbeteiligung

Wo können wir bauen? Staatssekretär Ingmar Jung vom Hessischen Ministerium für Wissenschaft und Kunst (Mitte) testet den Multitouch-Tisch, den das Fraunhofer-IGD zur besseren Einbindung der Bürgerschaft bei Bauprojekten entwickelt hat. Foto: Claus Völker

Windkanal-Simulationen für die Autoproduktion, Visualisierung von Bauprojekten zur Bürgerbeteiligung oder digitale Archivierung kostbarer Kulturschätze: Das Fraunhofer-Institut für Grafische Datenverarbeitung arbeitet an visuellen Lösungen für die Zukunft.

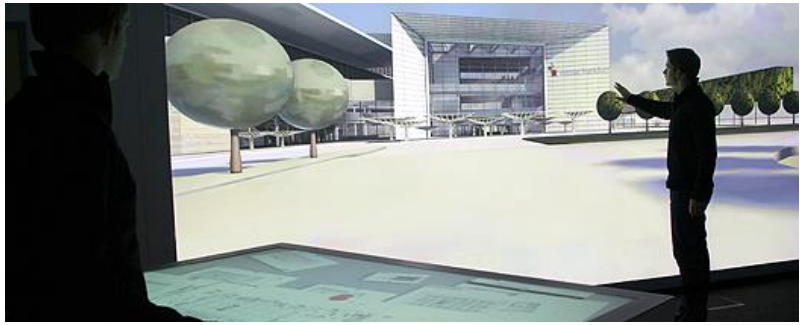
MATHEMATICAL AND APPLIED VISUAL COMPUTING

ZP2: Virtual Reality

Virtuelle Realität soll Menschen in virtuelle Welten eintauchen lassen.

Dabei gibt es zwei Arten:

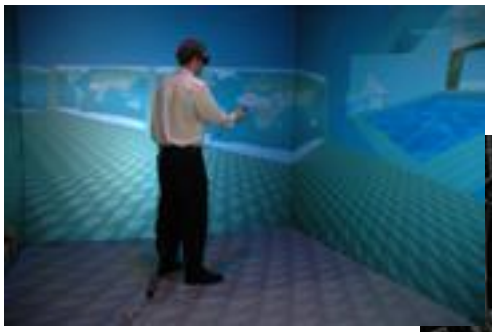
- Nicht-immersive Umgebungen
  - Bildschirm- und zeigerbasiert
  - 3D-Anzeige, eventuell haptisches Feedback
- Immersive Umgebungen
  - Es wird der Eindruck erweckt, tatsächlich in einer Welt aus virtuellen Objekten zu sein

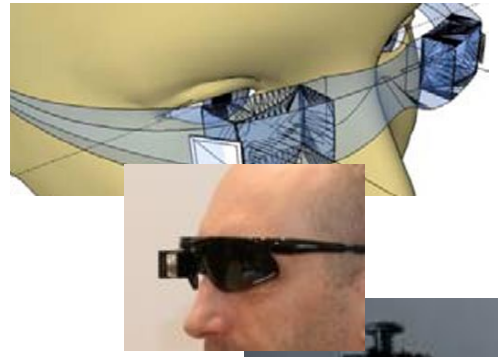


Virtual Reality

Ein- und Ausgabegeräte:

- Head Mounted Display (HMD)
- Spatial Immersive Display (SID)
- Cave Automated Virtual Environment (CAVE)
- Head-Movement-Tracking-Systeme



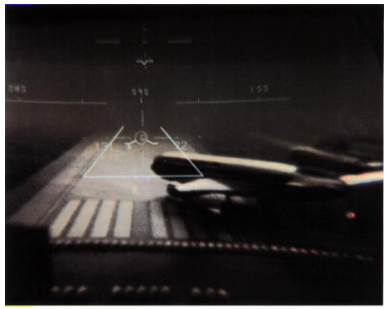




Augmented Reality

Augmented Reality beschreibt die nahtlose Integration zwischen virtuellen und realen Objekten. Die Wahrnehmung des Benutzers soll so erweitert und verbessert werden.

Kriterien für Augmented Reality Umgebungen:

- Die virtuellen Informationen sollten relevant bezüglich und synchron mit der echten Welt sein.



Ein- und Ausgabegeräte:

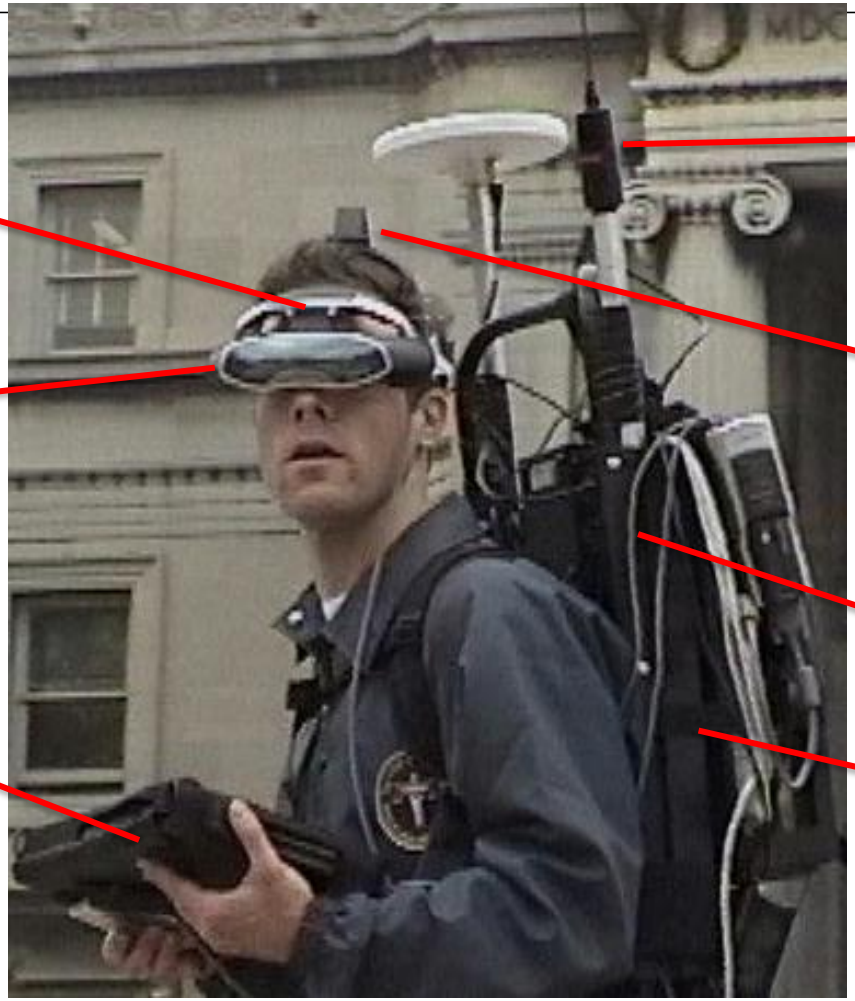
Heads Up Displays (HUD)

- Optisch durchlässig
- Halbdurchlässige Anzeige

zB Windschutzscheibe

Augmented Reality platform 1999

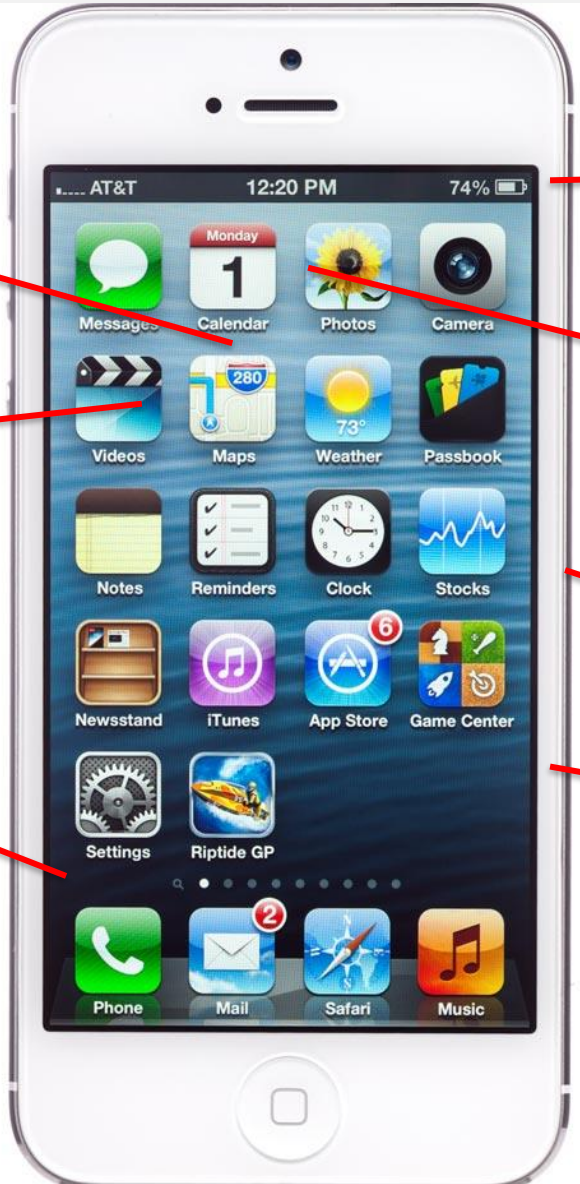
Mobile Augmented Reality



MARS, Höllerer et al. ~1999

Augmented Reality platform 2014

Mobile Augmented Reality



video camera

display

Mobile computer

GPS receiver

inertial sensor

Compass Sensor

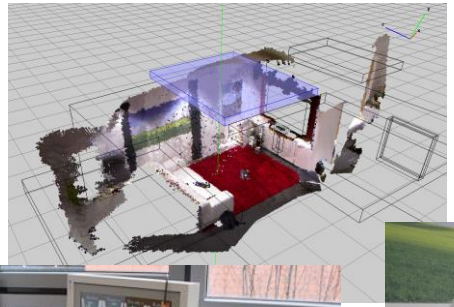
battery

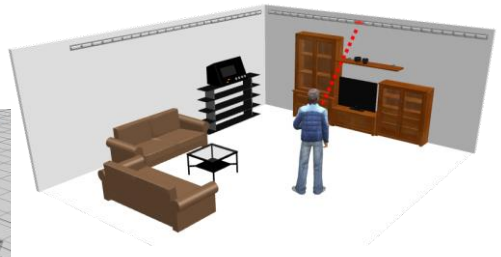
ZP3: Ambient / invisible

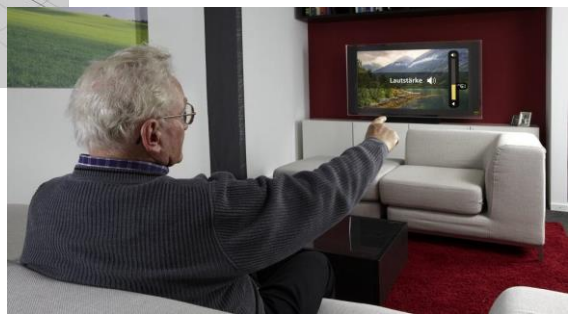
Lösungen, die das Leben erleichtern: zukunftsorientierten Ambient Intelligence / Ambient Assisted Living / Benutzerinteraktion in intelligenten Umgebungen

zB berührungslosen Freiraum-Gestenerkennung

- Kinect
- Kapazitive Sensoren
- http://www.opencapsense.org







ZP: Wearable / Ubiquitous

TECHNISCHE UNIVERSITÄT DARMSTADT

“Nach dem Smartphone beginnt die Smartwatch auf dem digitalen Markt Fuß zu fassen. Ihre Vorteile: man muss sie nicht erst aus der Tasche holen und hat die Hände frei. “

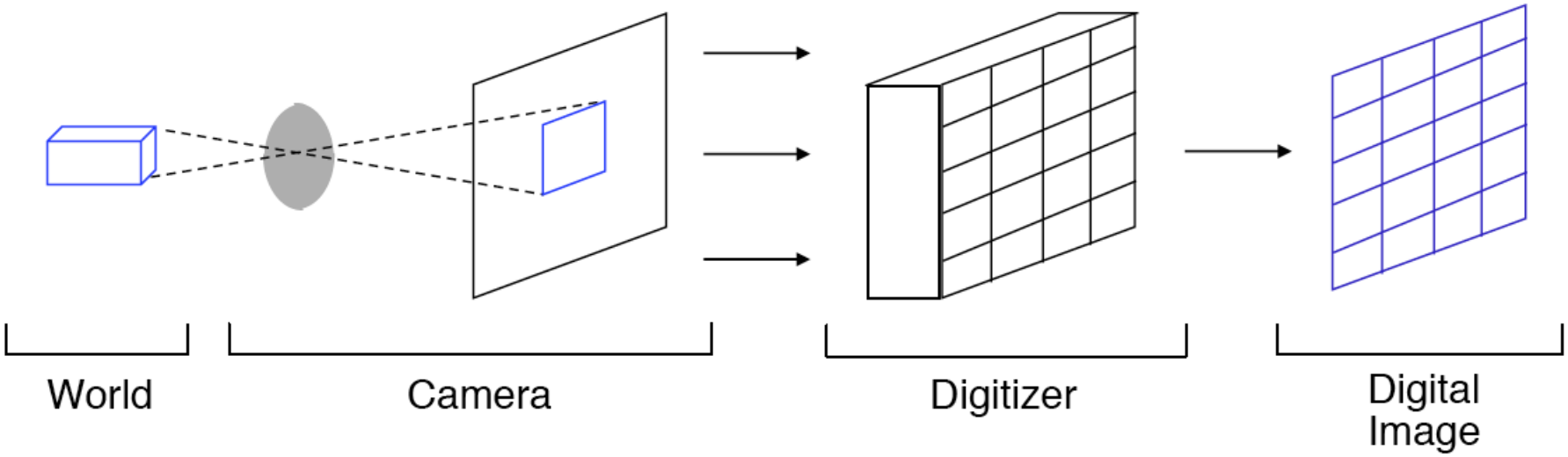
„Smartwatches bieten viel mehr als nur die Uhrzeit. Drahtlos mit einem Smartphone verbunden, zeigen sie E-Mails oder Kurzmitteilungen an. Der Träger sieht wer gerade anruft, noch ehe er in die Tasche nach dem Handy greift. Smartwatches können sich mit anderen Geräten verbinden, haben einen Vibrationsalarm und verfügen über Sensoren die Bewegungen erkennen. “

Überblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Hardware
- **Computergrafik**
  - **Geschichte**
- Grafikpipeline
  - Anwendung
  - Geometrieverarbeitung
  - Rasterisierung
  - Ausgabe

Computergrafik



Computergrafik: Prozess von links nach rechts

Computer Vision: Prozess von rechts nach links

Computergrafik

TECHNISCHE UNIVERSITÄT DARMSTADT

Modellierung

Darstellung

3D Objekte

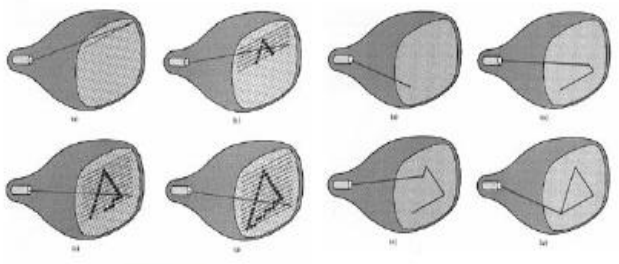
3D-Modelle
- Szene
- Geometrie
  - Polygon-Netz
- Material
- Beleuchtung

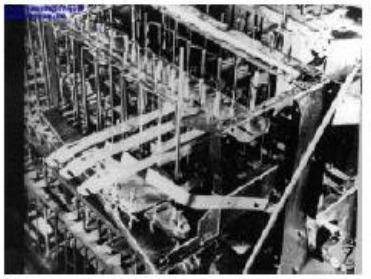
Notwendig für Darstellung:
- Rasterisierung

Bilder
- Interaktion
- Animation

Die Anfänge

- 1885: Erfindung des CRT (Cathode Ray Tube)
- 1938: Erster mechanischer Computer, Z1, von Konrad Zuse
- 1946: ENIAC
  - Electronic Numerical Integrator and Computer
  - Zahlenrepräsentation durch Elektronenröhren





Quelle: Marc Pauly, ETH Zurich

Die Anfänge

TECHNISCHE UNIVERSITÄT DARMSTADT

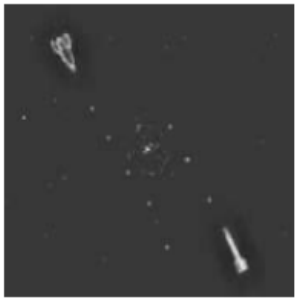
- 1946: MIT Whirlwind computer
  - Erster Computer mit Echtzeitdisplay
  - Flugsimulator für US-Marine
  - CRT-Ausgabe, Lightpen-Eingabe
- 1947: Erfindung des Transistors
- 1959: TX-2, entwickelt am MIT
  - Erster transistorbasierter Computer, 9 Zoll CRT und Lightpen

Quelle: Marc Pauly, ETH Zurich

60er Jahre

TECHNISCHE UNIVERSITÄT DARMSTADT

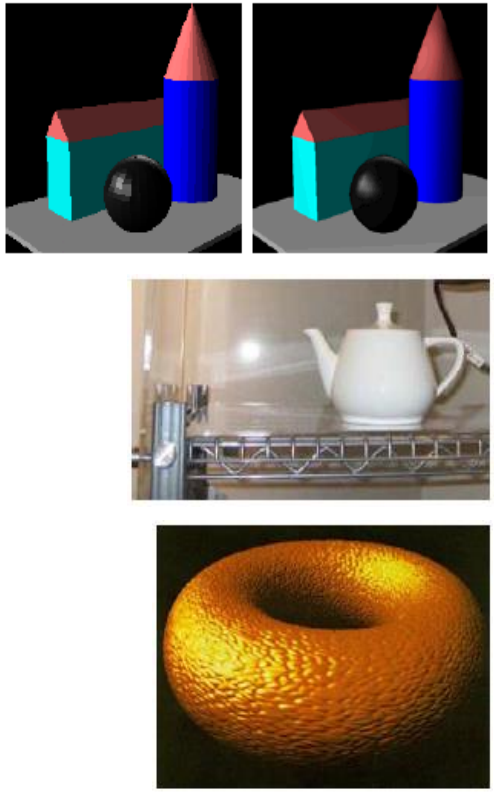
- 1961: SpaceWar
  - Erstes Computerspiel am MIT
- 1963: Sketchpad auf dem TX-2
  - Von Ivan Sutherland, dem „Großvater“ der interaktiven Computergrafik
- 1968: Maus (Douglas Engelbart)
- 1969: ACM SIGGRAPH
- 1969: Erster Frame Buffer
- 1969: Erste GUI von Alan Kay (Xerox)



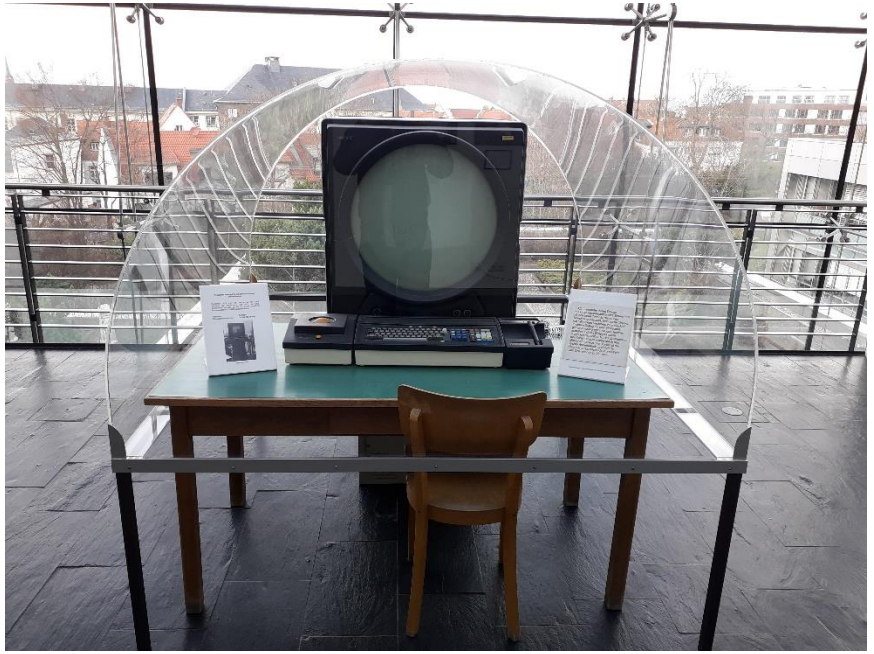
Source: Marc Pauly, ETH Zurich

70er Jahre

- 1971: Henri Gouraud:
  - Interpoliertes Shading
- 1974: Ed Catmull:
  - Texture Mapping, Z-Buffer
- 1975: Bui-Tuong Phong:
  - Normal Interpolation Shading
- 1975: Utah Teapot
- 1977: Jack Bresenham:
  - Scan Conversion Algorithmen
- 1978: Jim Blinn:
  - Bump Mapping

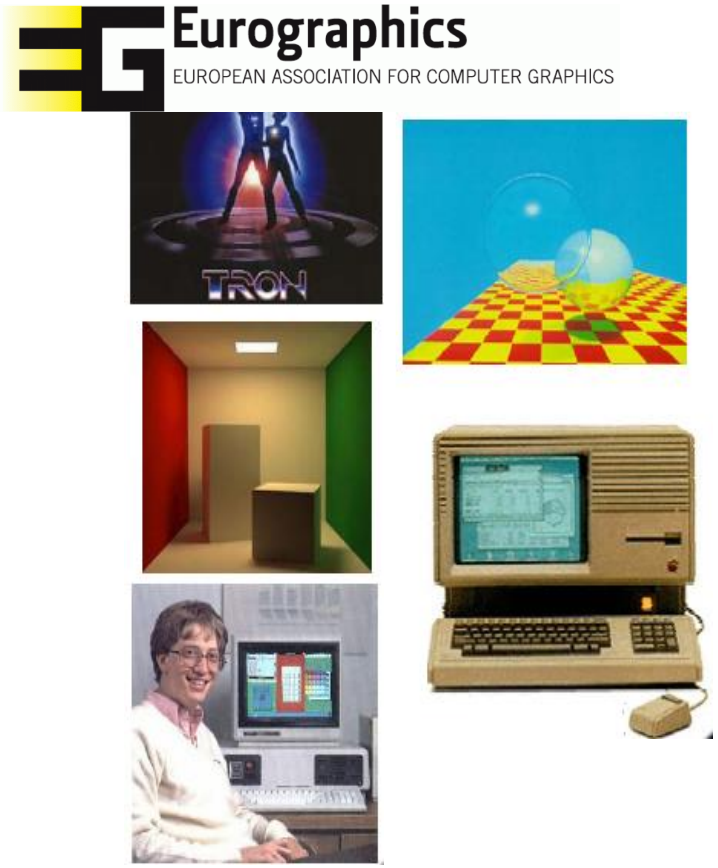


Source: Marc Pauly, ETH Zurich



80er Jahre

- 1980: Eurographics
  - Mitglied Nr. 1: José Encarnação (TU Darmstadt und Fraunhofer IGD)
- 1980: Turner Whitted: Ray Tracing
- 1982: Silicon Graphics (SGI)
- 1982: Marr: Primal Sketch
- 1982: TRON (Disney):
  - 15-minütiger Anteil an computergenerierten Bildern
- 1983: Apple Lisa: Erster PC mit GUI
- 1983: Witkin 1D Scale space
- 1984: Koenderink: 2D Scale Space
- 1984: Goral et al.: Radiosity
- 1985: Microsoft Windows 1.01
- 1986: MIT: X-Window System



Source: Marc Pauly, ETH Zurich

90er Jahre

TECHNISCHE UNIVERSITÄT DARMSTADT

- 1990: Perona - Malik
- 1990: Total Variation (Rudin)
- Anfang 1990er Jahre: Prophezeiung
  „Computer sich verteilen sich auf das Netz “
- 1992: OpenGL von SGI veröffentlicht
- 1995: GMD (heute Fraunhofer FIT)
  - BSCW (heute „Cloud“)
- 1995: Toy Story:
  - Erster Film in voller Länge, der vollständig computeranimiert ist
- 1996: 3Dfx Voodoo:
  - Erster 3D-Beschleuniger für PCs, texturierte Dreiecke
- 1998: Schneiderman and Kanade:
  - Gesichtsdetektion
- 1999: GeForce256:
  - Transformationen und Beleuchtung
- 1999: PC-Grafik erreicht höhere Performance als SGI-Grafik-Workstations

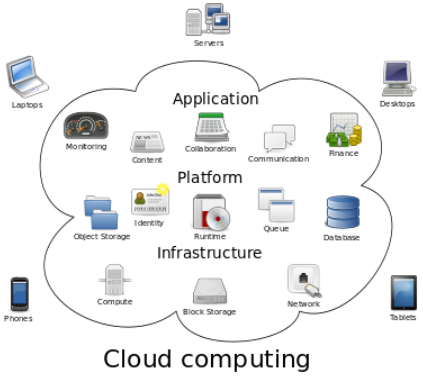
Source: Marc Pauly, ETH Zurich

2000


Daily Mail, Tuesday, December 5, 2000
Page 33
Internet 'may be just a passing fad as millions give up on it'
By James Chapman
Science Correspondent
THE Internet may be only a passing fad for many users, according to a report.
Researchers found that millions were turning their back on the world wide web, frustrated by its limitations and unwilling to pay high access charges.
They say that e-mail, far from replacing other forms of communication, is adding to an overload of information.
Experts from the Virtual Society project, which published the report, say predictions that the Internet would revolutionise the way society works have proved wildly
Many teenagers are using the Internet less now than previously, they conclude, and the future of online shopping is limited. Steve
Net loss: Two million Britons have logged off the Internet
EXCLUSIVE!
NOW THERE'S ANOTHER WAY INTELLIGENT FINANCE COULD MAKE YOU BETTER OFF.

2000 – 2005

- 2001: Viola & Jones:
  - Real-time Gesichtsdetektion
- 2001: GeForce 3
  - Programmierbare Transformationen und Beleuchtung
- 2001: Final Fanasy
  - Menschliche Darsteller werden durch computergenerierte Modelle ersetzt
- 2004: Facebook
- 2004: GeForce FX, ATI Radeon 9800XT
  - ~ 4000 Mio. Texel pro Sekunde
  - ~ 400 Mio. Vertizes pro Sekunde
- 2005: XBOX 360
  - ~ 8000 Mio. Texel pro Sekunde
  - ~ 1200 Mio. Vertizes pro Sekunde



Source: Marc Pauly, ETH Zurich

2006 – 2023

TECHNISCHE UNIVERSITÄT DARMSTADT

- 2006: Amazon: (serviceorientierte) Architektur
- 2007: iPhone.
  - Berührungsempfindlich, iOS 1.0,
    480 × 320 Pixel, 2MP Kamera

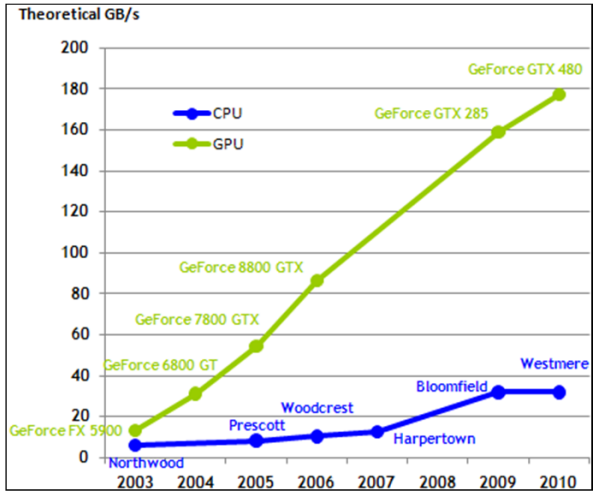
- 2009: Big Bang in Machine Learning (NVIDIA)
- 2011: Deep-Learning-basierte Bilderkennung ist "übermenschlich" geworden
  - Liefert genauere Ergebnisse als menschliche Teilnehmer
- 2015: AlphaGo
- 2020: NeRF: Neural Radiance Fields
- 2022: Stable Diffusion

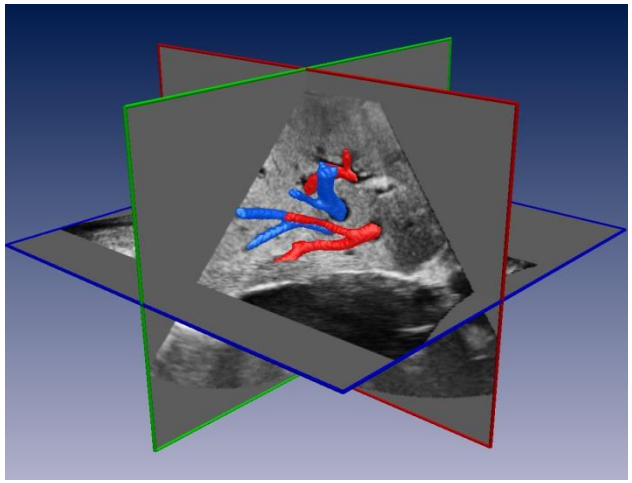
MATHEMATICAL AND APPLIED VISUAL COMPUTING

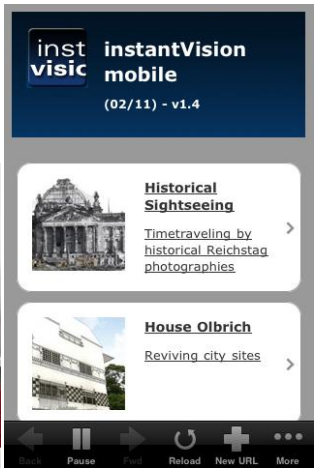
2024 –

TECHNISCHE UNIVERSITÄT DARMSTADT

- Cloud
- Immersiv
- 3D
- Mobil
- Interaktiv
- Deep(-er) learning
- Quantum...

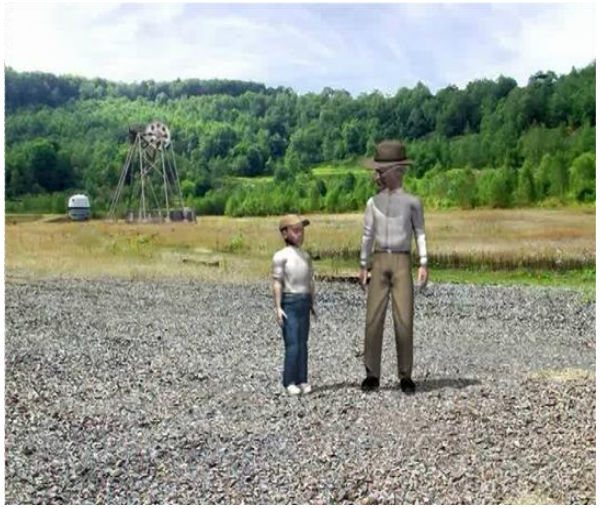






MATHEMATICAL AND APPLIED VISUAL COMPUTING

Beispiel: Virtuelle Charakter



Virtuelle Charakter

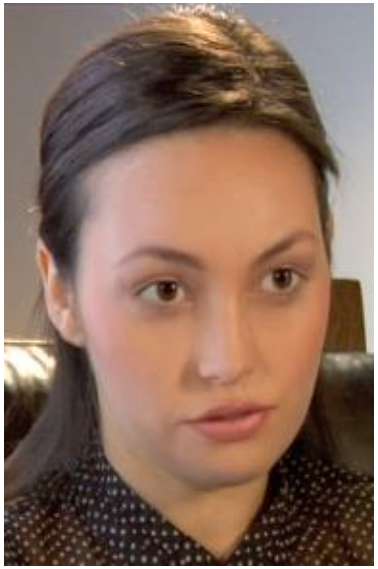
TECHNISCHE UNIVERSITÄT DARMSTADT

Unterschiedliche Darstellung

Sketches

Comic-Style

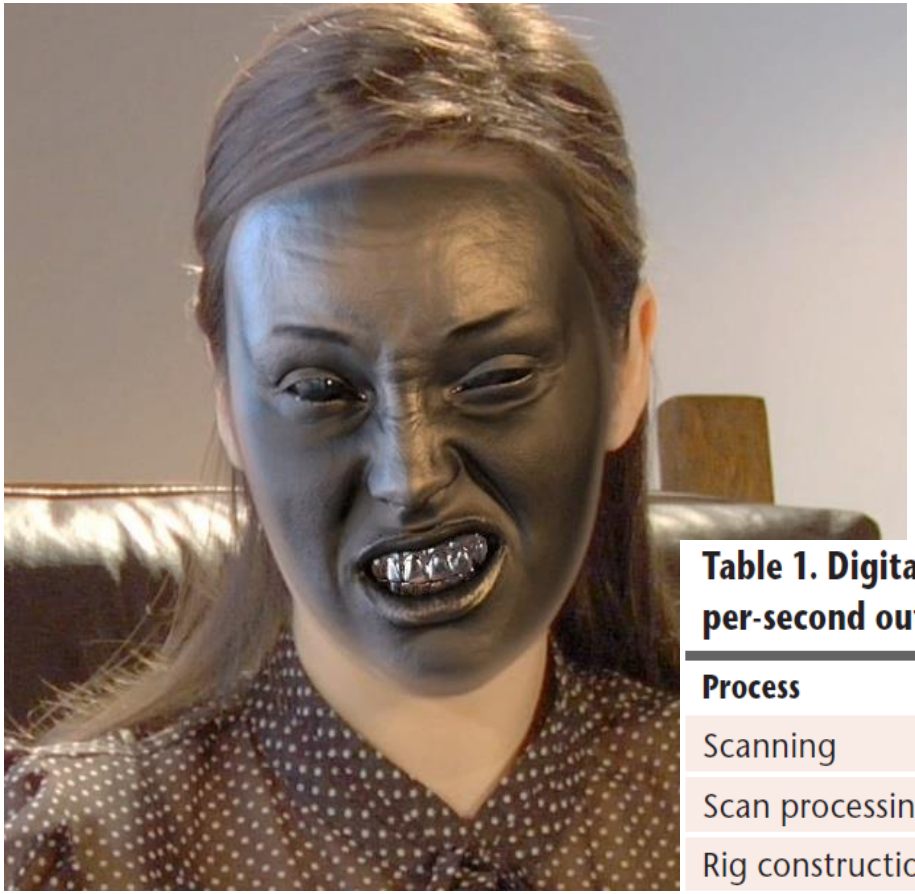
Artificial



Human-like

MATHEMATICAL AND APPLIED VISUAL COMPUTING

Digital Emily



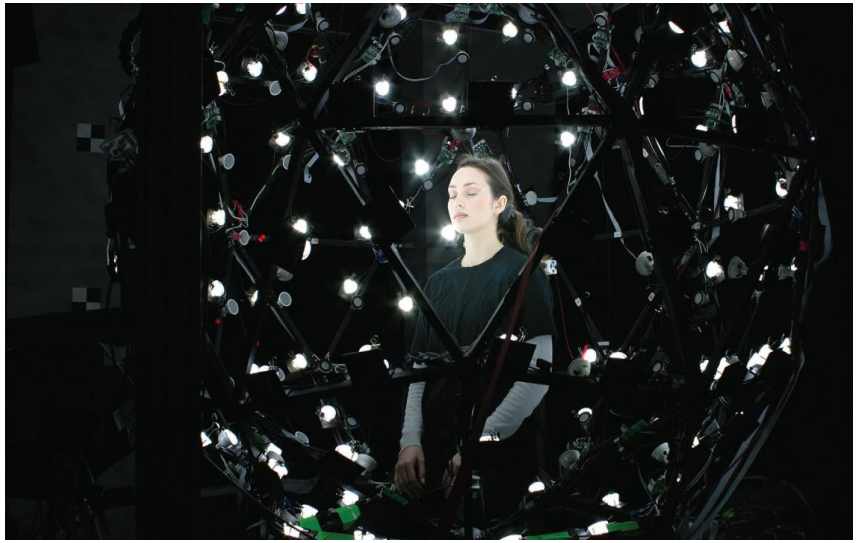
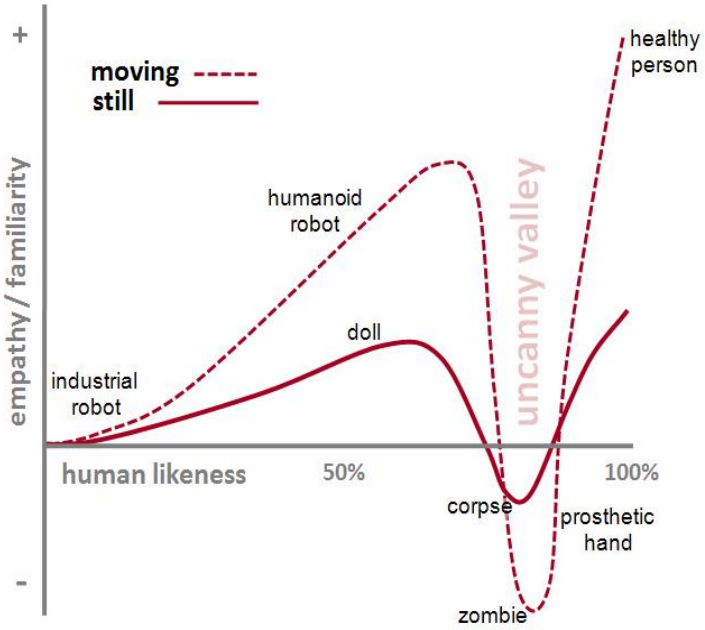


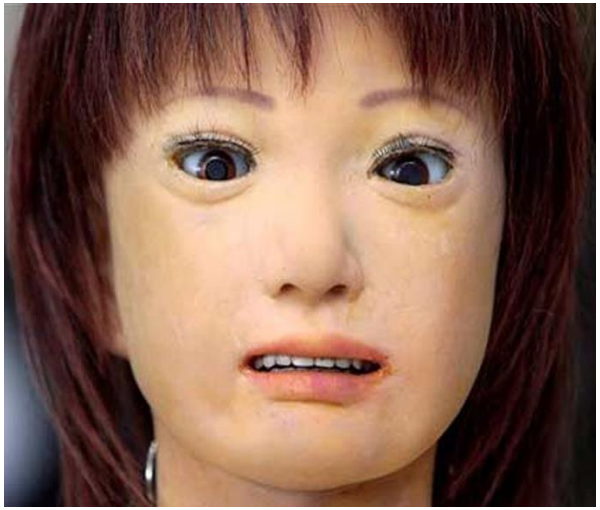
Table 1. Digital Emily process time frames and resources for 24 frames-per-second output at 1,920 × 1,080 pixels.

Process	Resources
Scanning	1.5 hours total, 3 seconds per scan, 3 technicians
Scan processing	10 days, 37 processed scans, 1 artist
Rig construction	3 months, 75 blendshapes, 1 artist
Animation	2 weeks, 90 seconds of animation, 2 animators
Rendering/compositing	3 months, 1 artist

The “uncanny valley”

Eine hypothetische Beziehung zwischen dem Grad der Ähnlichkeit eines Objekts mit einem Menschen und der emotionalen Reaktion auf ein solches Objekt.





https://www.youtube.com/watch?v=CNdAIPoh8a4

(Mehr: https://de.wikipedia.org/wiki/Uncanny_Valley & Google Bilder, Youtube)

Rendering

TECHNISCHE UNIVERSITÄT DARMSTADT

Blau?

Youtube:

Avatar Motion Capture Mirrors Emotions (High).flv

Making-off «Tintin», Nick Frost and Simon Pegg (Low).flv

MATHEMATICAL AND APPLIED VISUAL COMPUTING

Überblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Hardware
- Computergrafik
  - Geschichte
- **Grafikpipeline**
  - **Anwendung**
  - **Geometrieverarbeitung**
  - **Rasterisierung**
  - **Ausgabe**

Computergrafik

TECHNISCHE UNIVERSITÄT DARMSTADT

Modellierung

Darstellung

3D Objekte

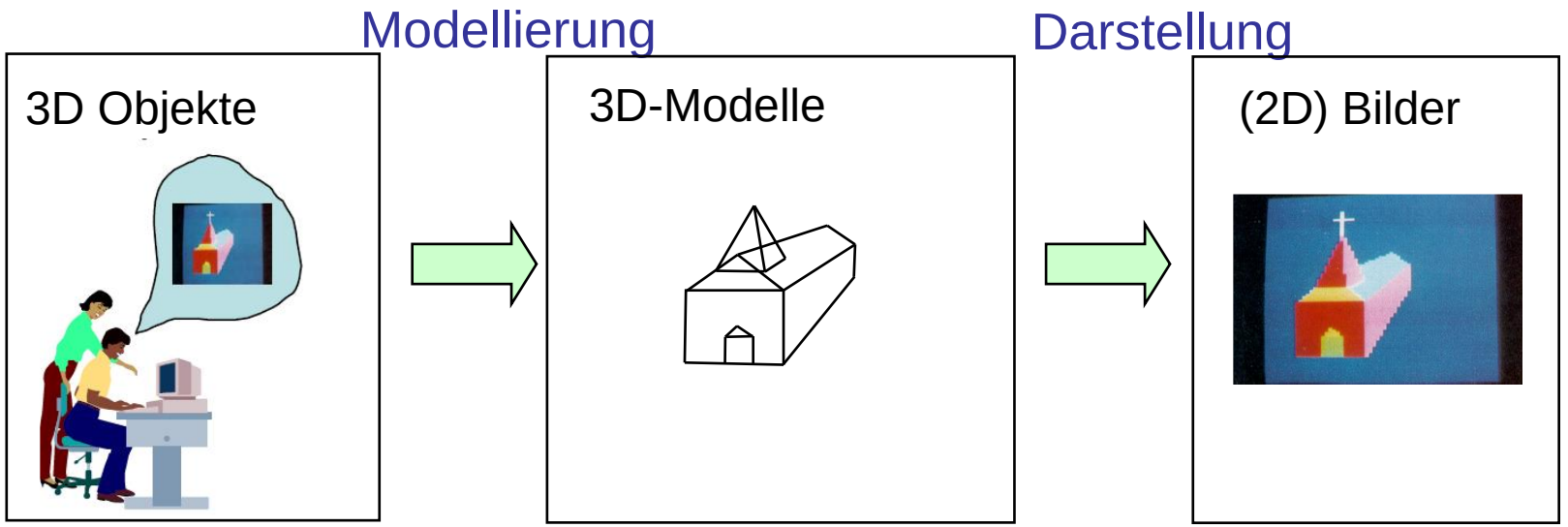
3D-Modelle
- Szene
- Geometrie
  - Polygon-Netz
- Material
- Beleuchtung

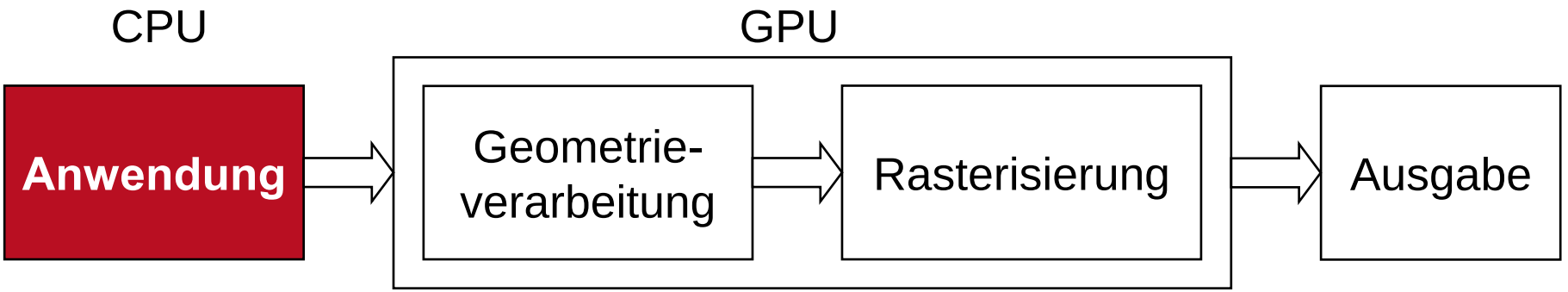
Notwendig für Darstellung:
- Rasterisierung

Bilder
- Interaktion
- Animation

3D-Grafik-Pipeline - Schematisch

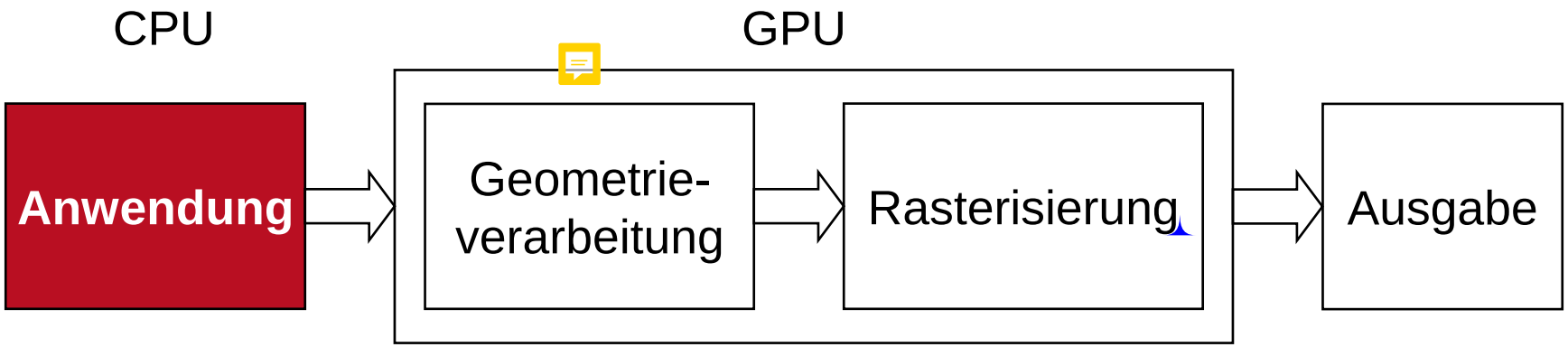
TECHNISCHE UNIVERSITÄT DARMSTADT



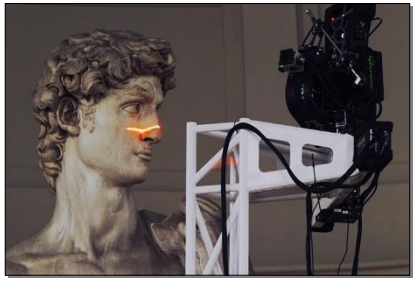


MATHEMATICAL AND APPLIED VISUAL COMPUTING

Grafikpipeline: Anwendung

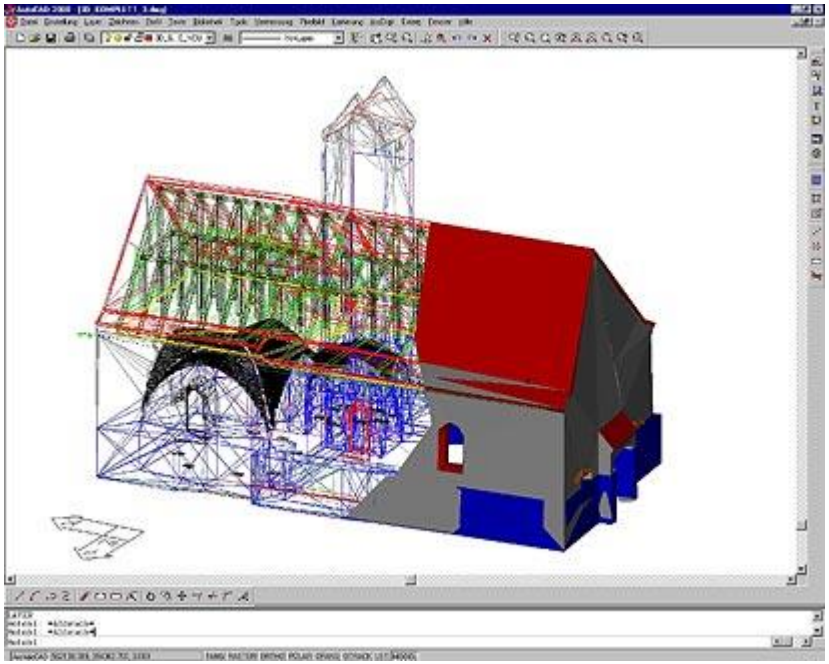


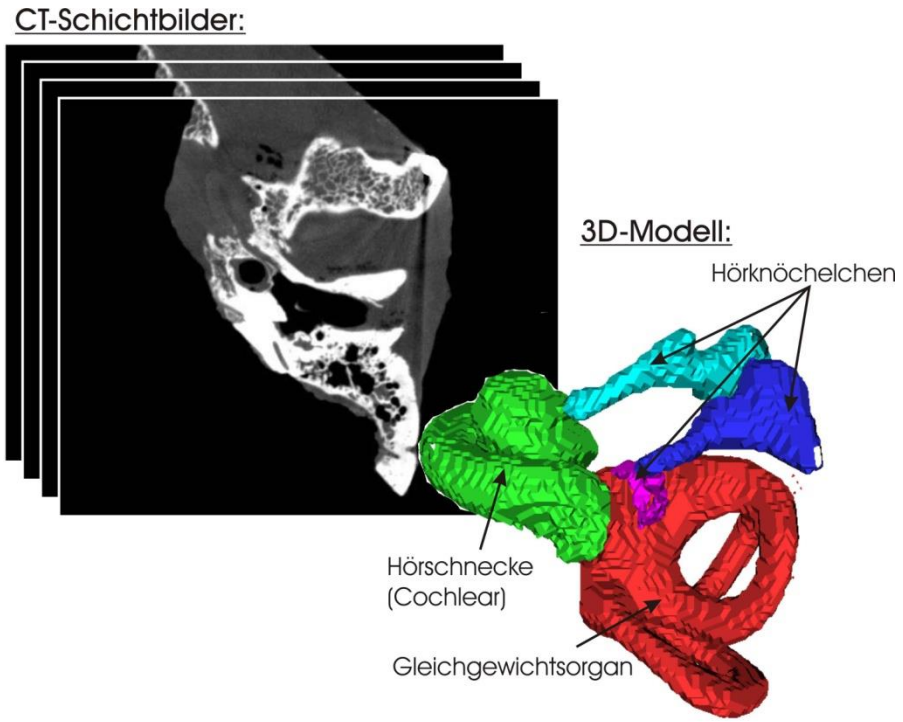
- **Eingabe grafischer Daten**
- Repräsentation von 3D-Daten
  - Grafische Primitive
  - Transformationen
  - Räumliche Datenstrukturen



Generierung von 3D-Modellen

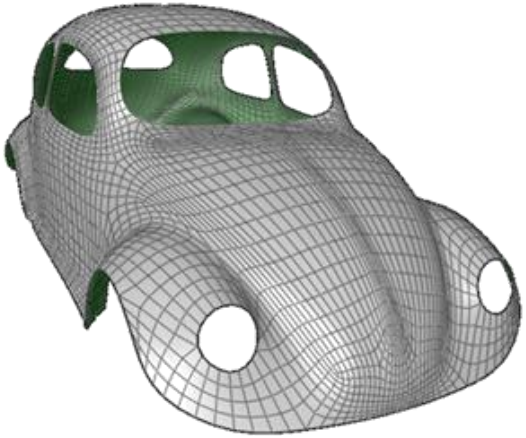
TECHNISCHE UNIVERSITÄT DARMSTADT

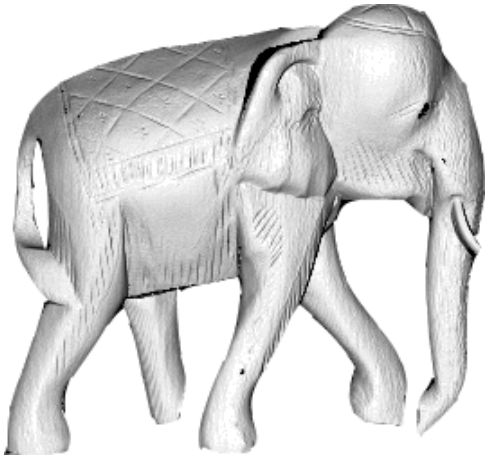


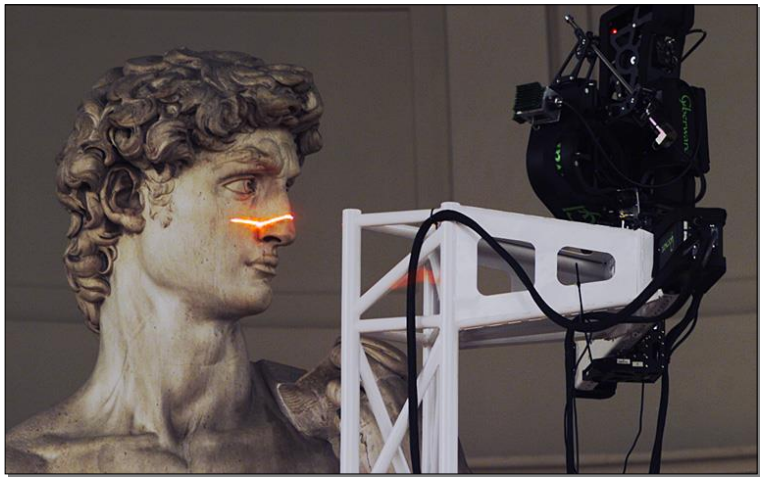


MATHEMATICAL AND APPLIED VISUAL COMPUTING

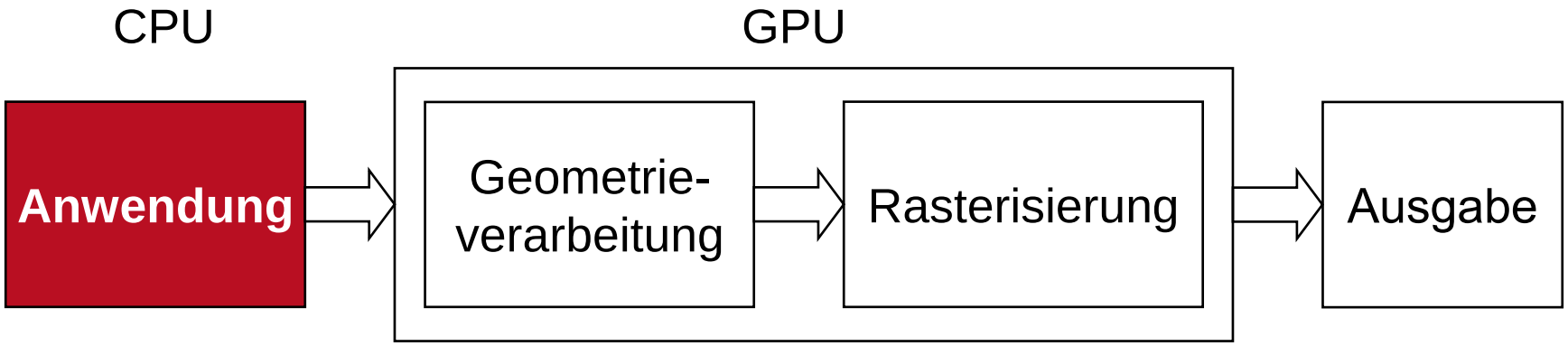
Abtastung realer Objekte



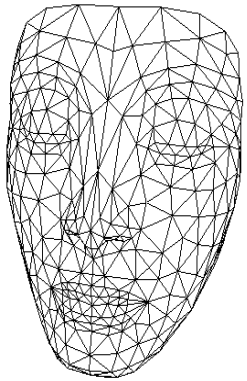


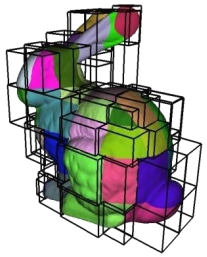


Grafikpipeline: Anwendung

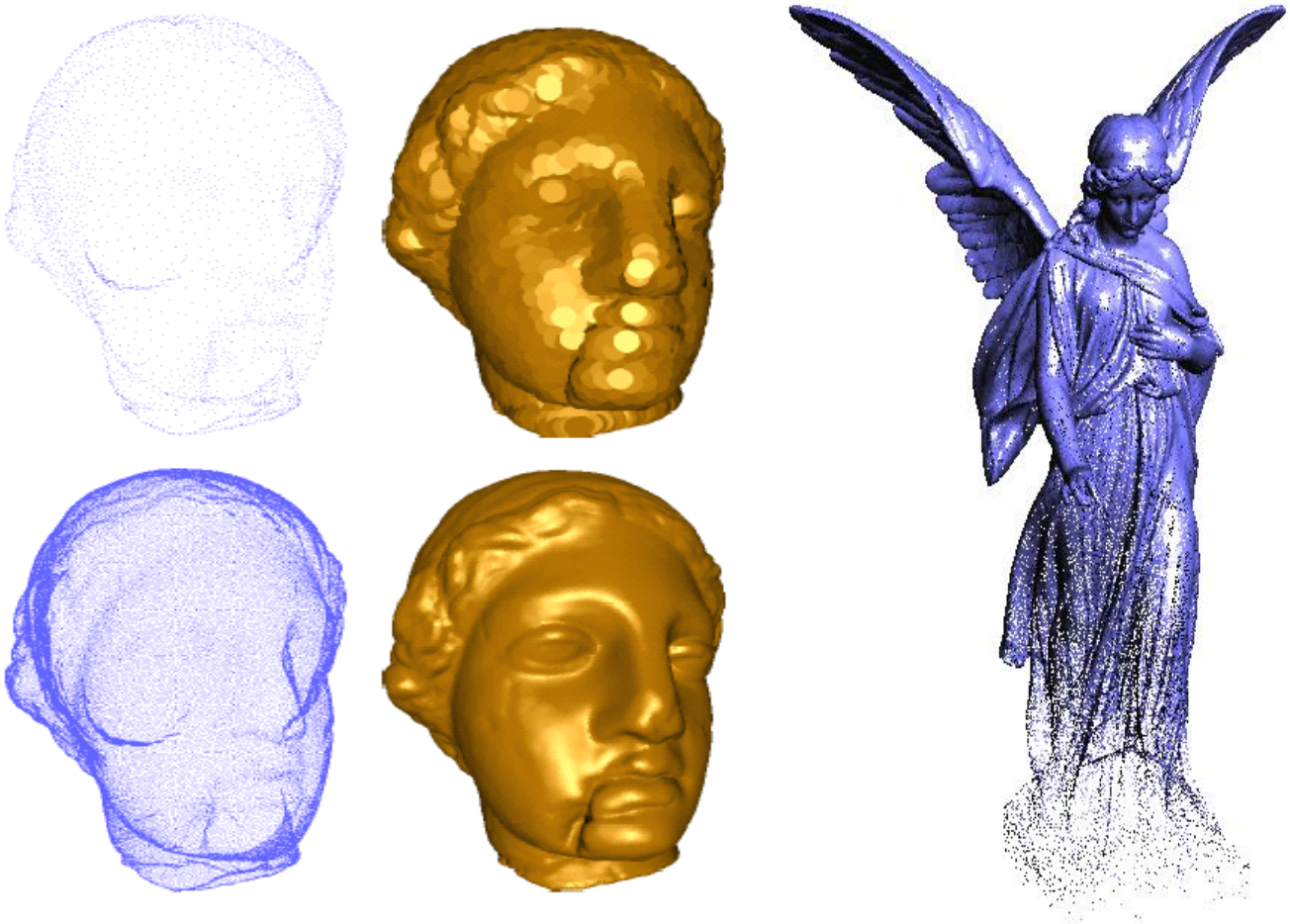


- Eingabe grafischer Daten
- **Repräsentation von 3D-Daten**
  - **Grafische Primitive**
  - Transformationen
  - Räumliche Datenstrukturen



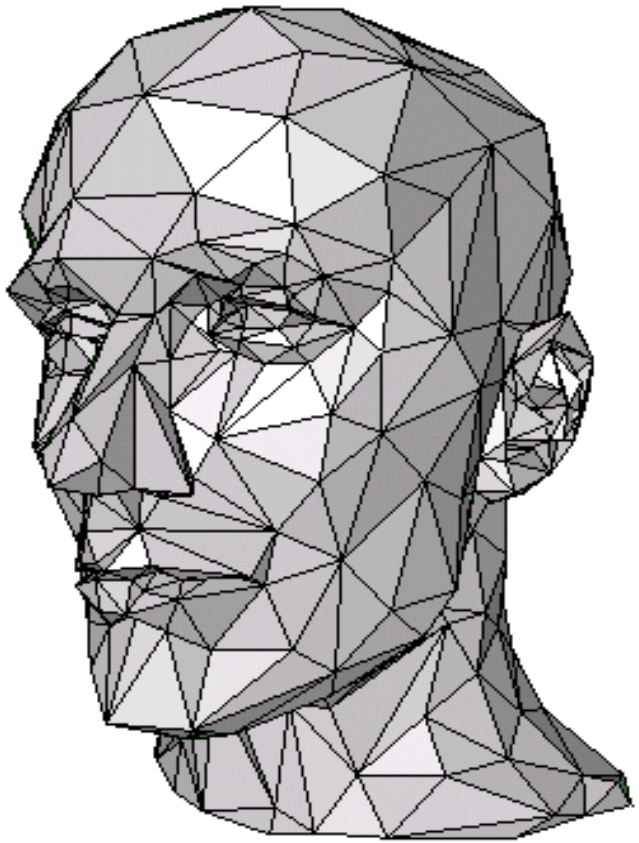


Primitive – Punkte

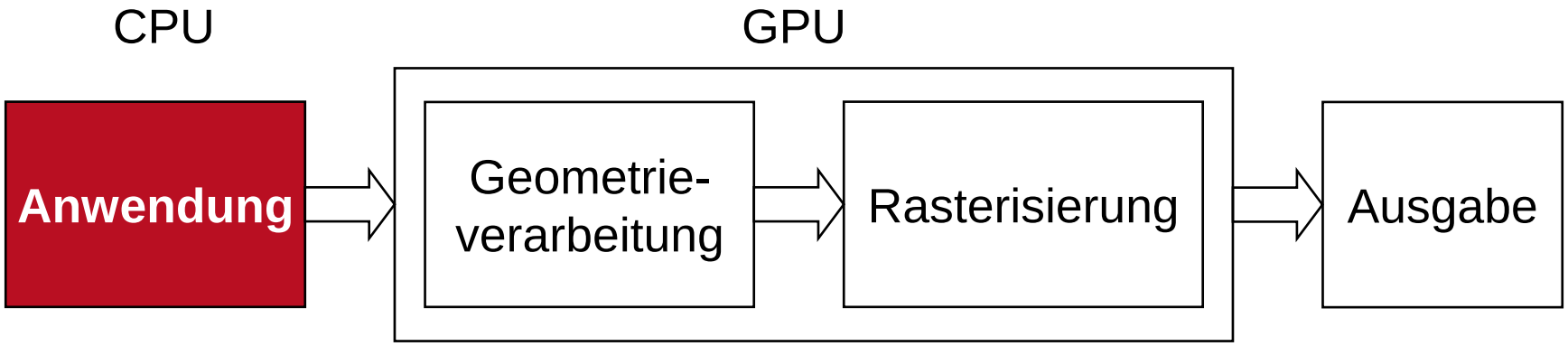


Primitive – Linien

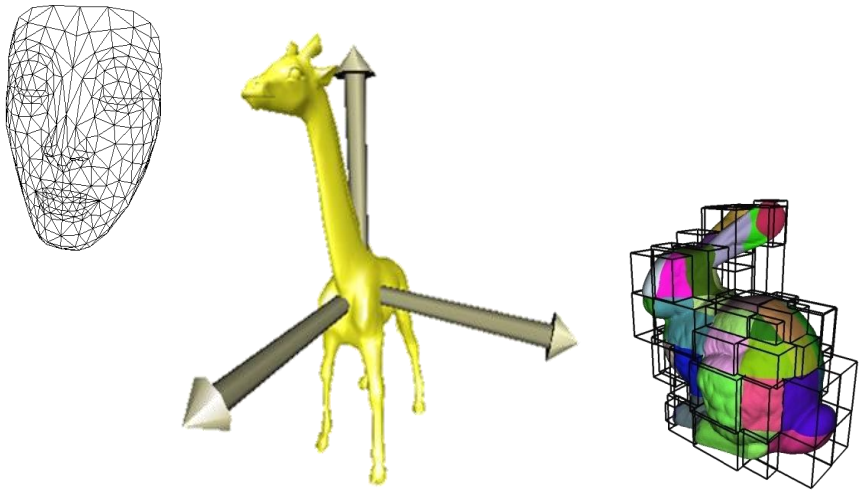
Primitive – Dreiecke



Grafikpipeline: Anwendung



- Eingabe grafischer Daten
- **Repräsentation von 3D-Daten**
  - Grafische Primitive
  - **Transformationen**
  - Räumliche Datenstrukturen



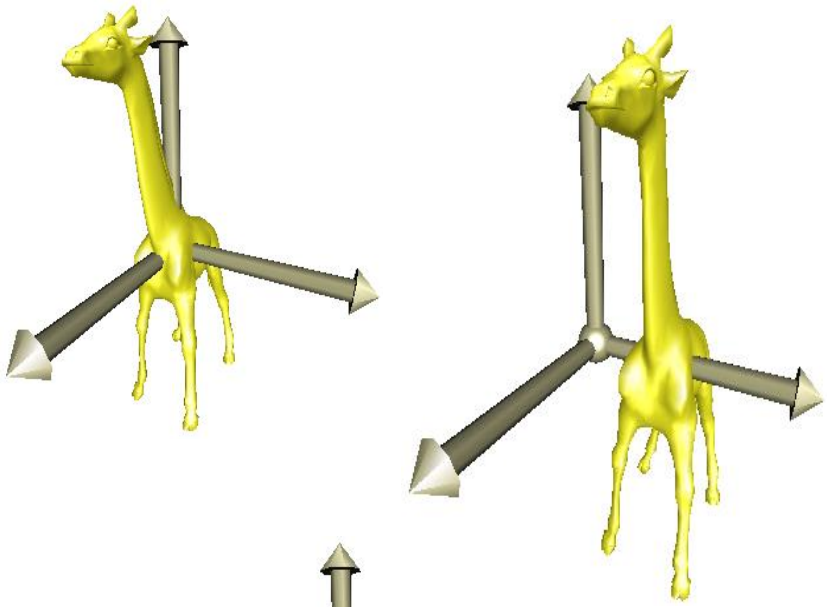
Primitive – Positionierung

TECHNISCHE UNIVERSITÄT DARMSTADT

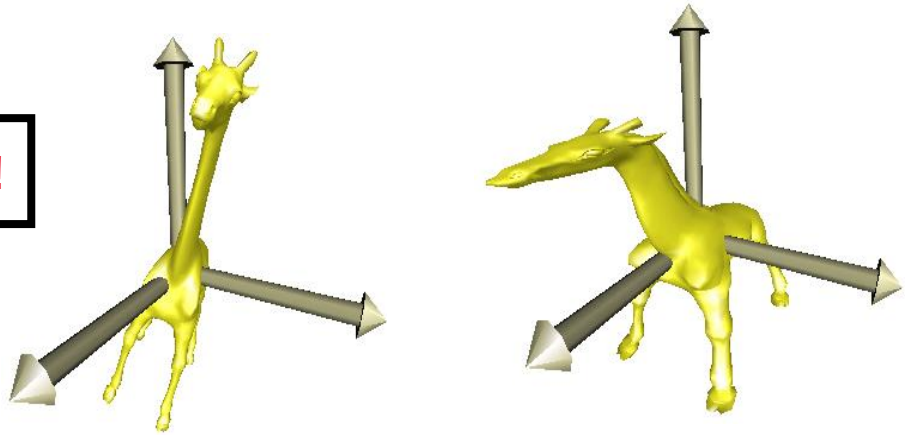
Transformation + relative Koordinaten

- Translation
- Rotation
- Skalierung
- Scherung

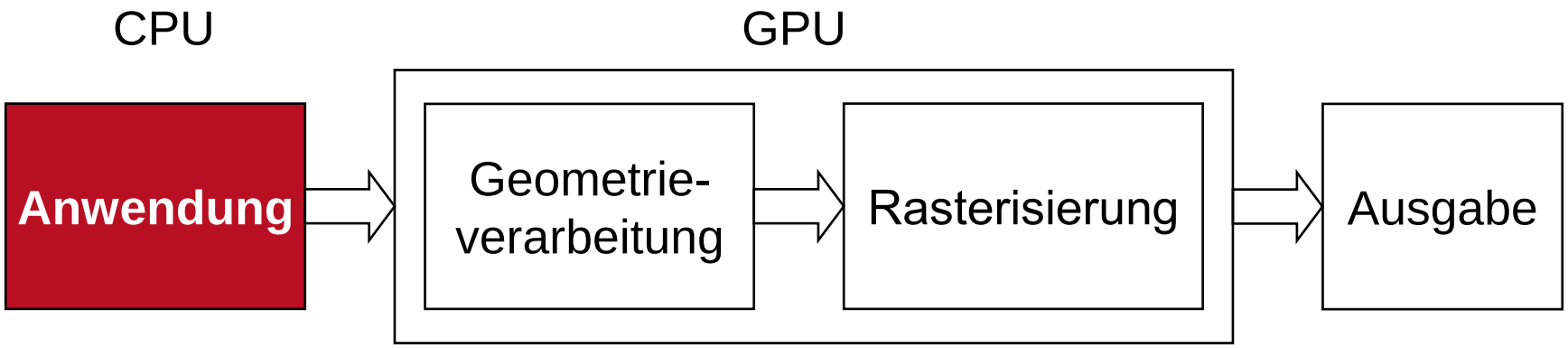
affine Abbildungen / Transformationen!



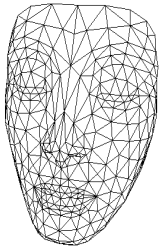
Nächste Vorlesung!

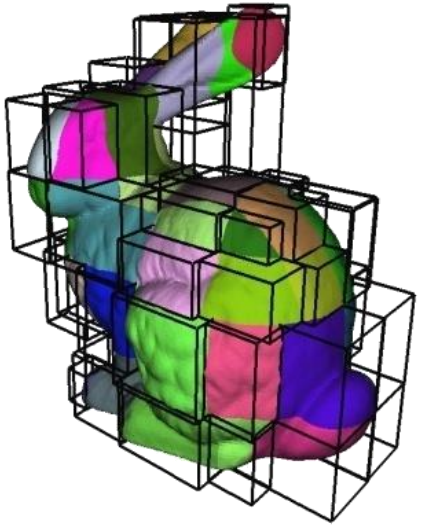


Grafikpipeline: Anwendung



- Eingabe grafischer Daten
- **Repräsentation von 3D-Daten**
  - Grafische Primitive
  - Transformationen
  - **Räumliche Datenstrukturen**

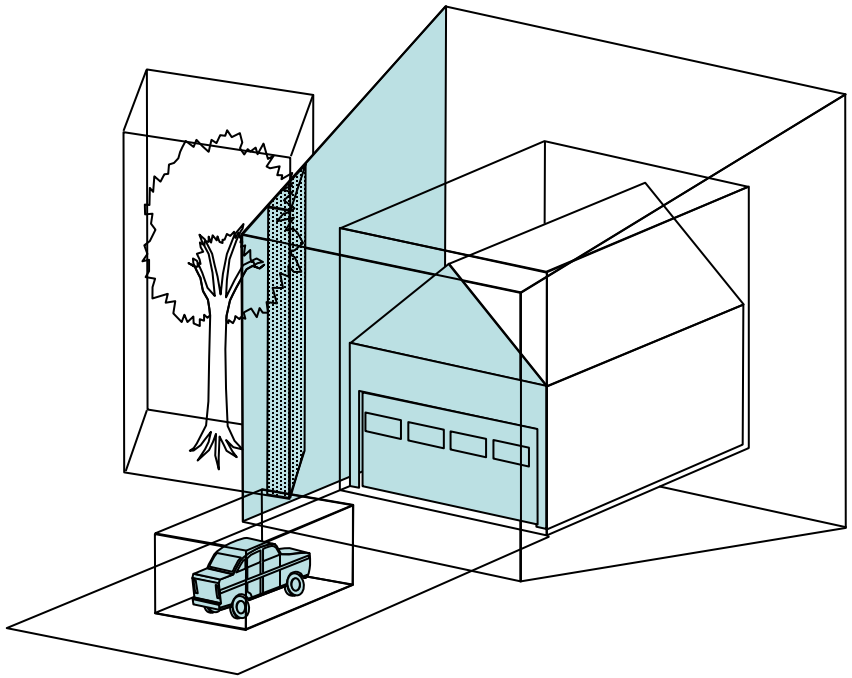




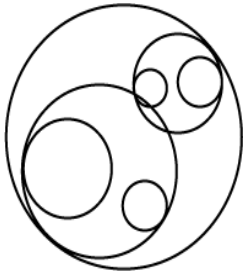
Räumliche Datenstrukturen - Anwendungen

TECHNISCHE UNIVERSITÄT DARMSTADT

View Frustum Culling
   (Objekt in Sichtvolumen sichtbar?)
Occlusion Culling / z-Buffer
   (Verdeckung)
Kollisionserkennung



Räumliche Datenstrukturen - Klassifikation

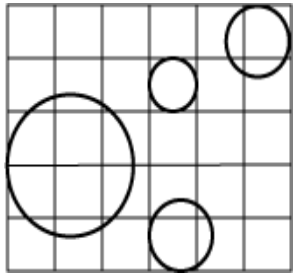


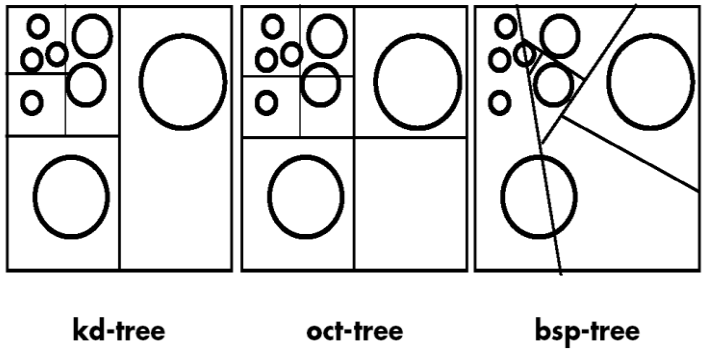
Hüllkörperhierarchie

- Bsp.: Bounding Sphere Hierarchy

Raumunterteilung

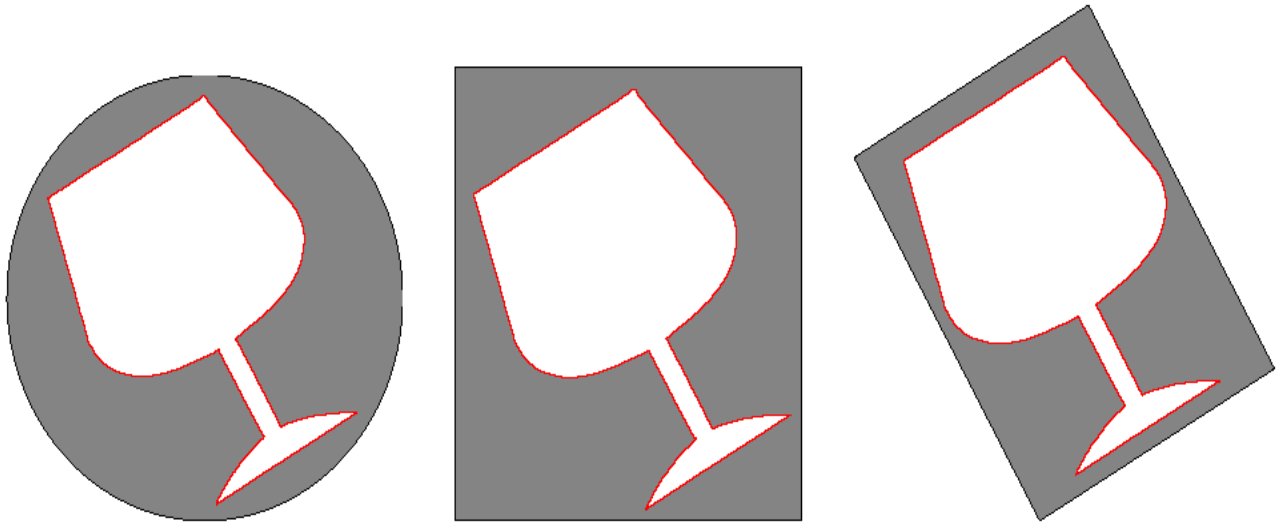
- Regulär: Gitter
- Irregulär / Hierarchisch
  - k-d Tree (achsenparallele BSP-Bäume)
  - Uniform: Quadtree (2D) / Octree (3D)
  - BSP („Binary Space Partitioning“) Tree





Hüllkörper (Bounding Volumes)

Hüllkörper müssen einfach sein, d.h. Schnitttests mit anderen Primitiven (Sichtvolumen, Sehstrahl) müssen sich einfach berechnen lassen.



Kugel

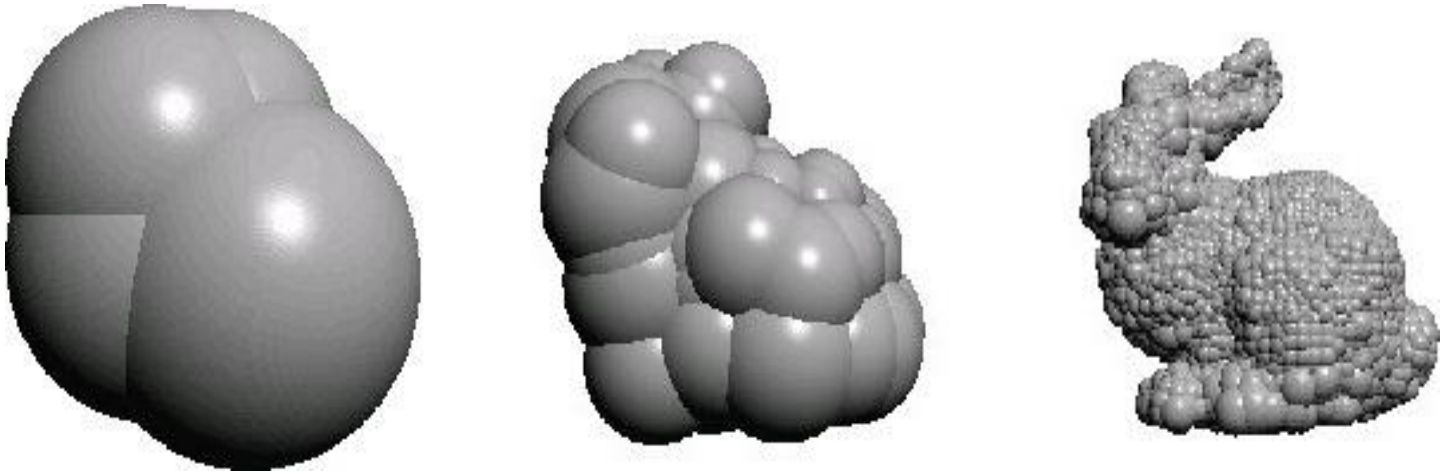
Achsenparallele Bounding Box (BB)

Orientierte BB

Hüllkörperhierarchien - Konstruktion

TECHNISCHE UNIVERSITÄT DARMSTADT

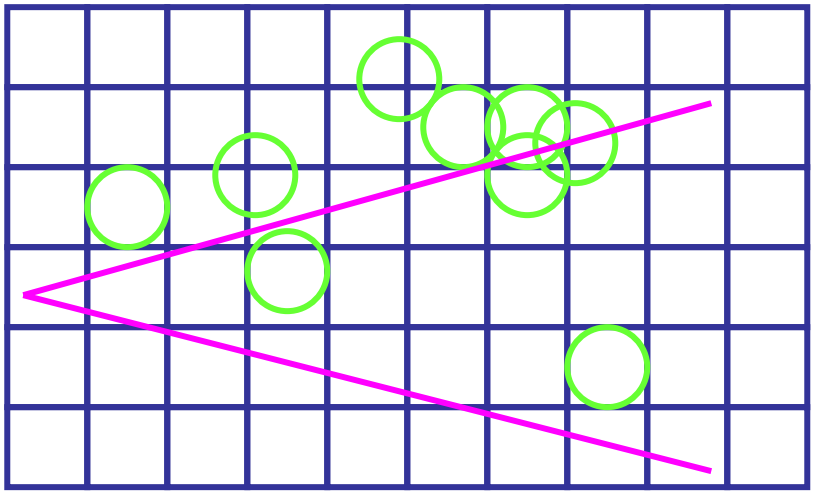
Beispiel: Bounding Sphere Hierarchy



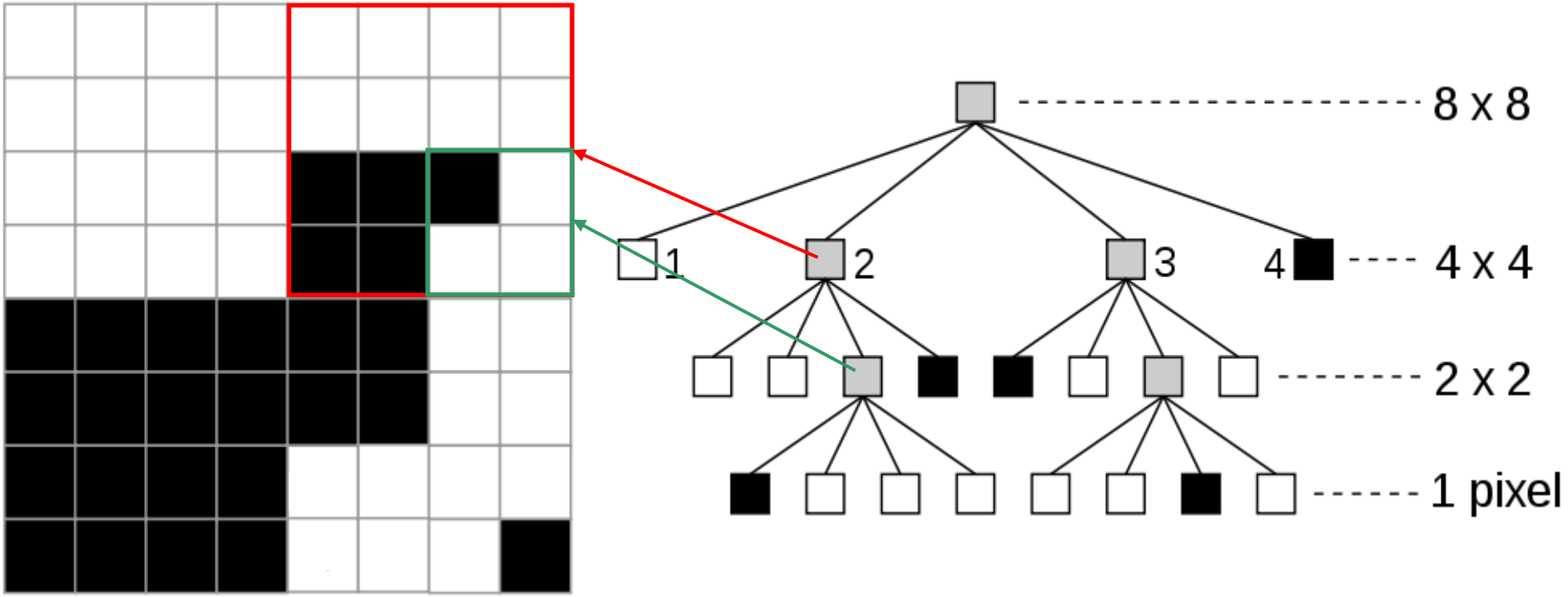
Raumunterteilungen
Achsenparallele Gitter (Grids)

TECHNISCHE UNIVERSITÄT DARMSTADT

- einfach
- Objekt ist in mehreren Zellen enthalten (Objektredundanz)
- kann sich der Geometrie nicht anpassen
- Sehr speicheraufwändig
- Effizient traversierbar
- Schneller Zugriff auf Nachbarn ist möglich
- Voxel- / Volumendarstellung



Raumunterteilungen
Quadtree/Octree

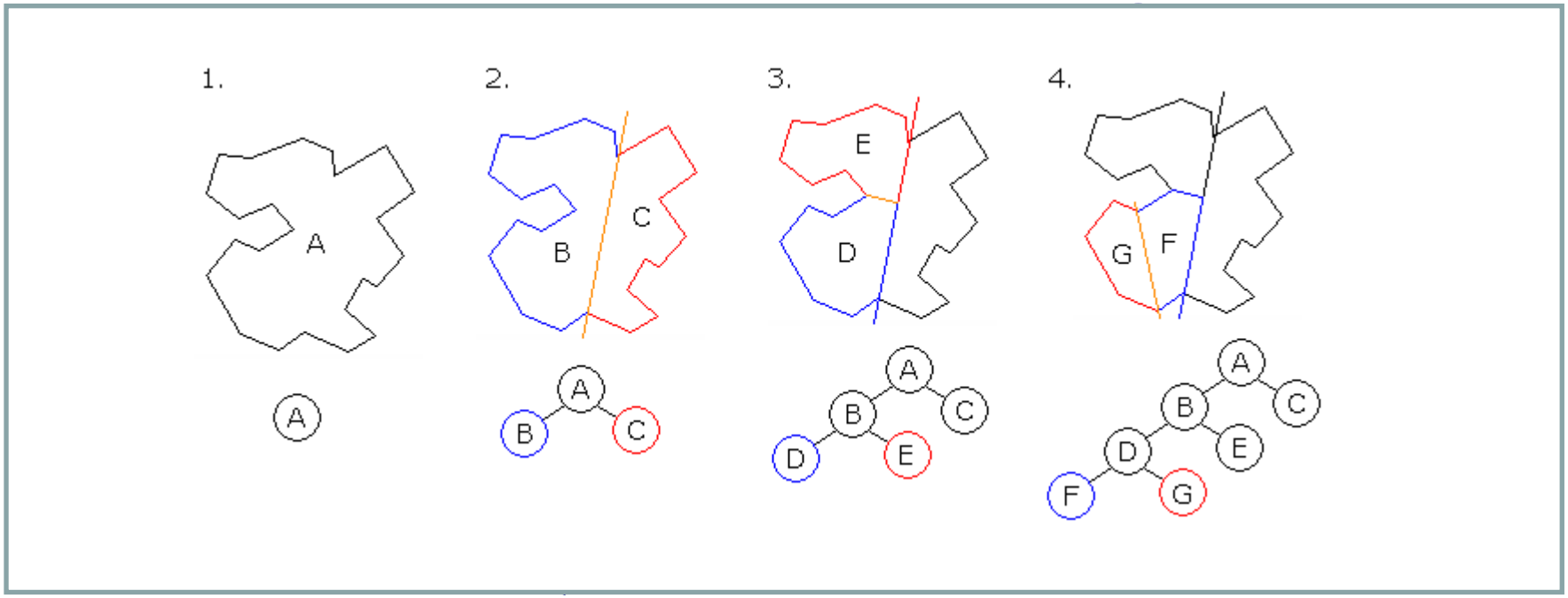


Raumunterteilungen
Binary Space Partition (BSP)

TECHNISCHE UNIVERSITÄT DARMSTADT

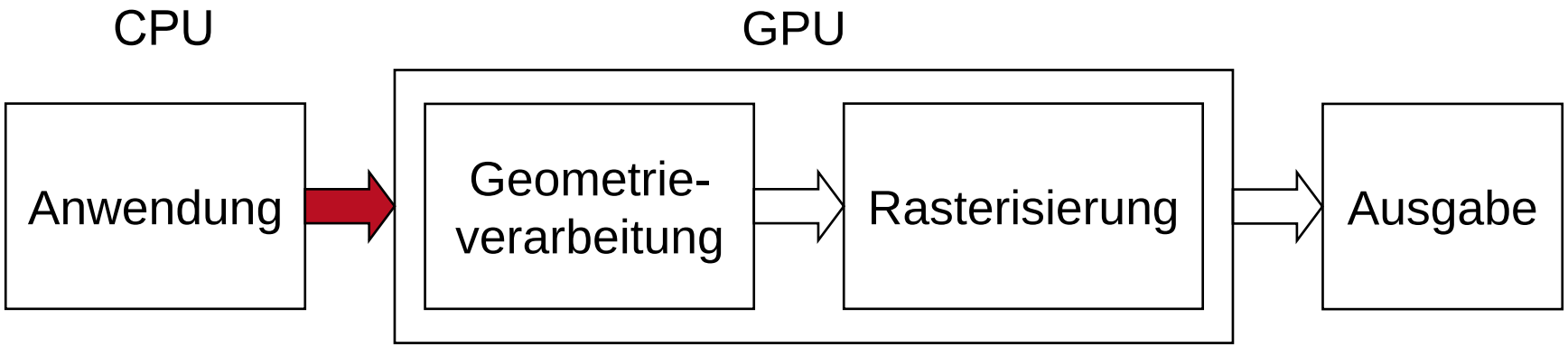
BSP Bäume: Raum wird binär unterteilt

- Jeder Knoten entspricht einer Unterteilungsebene, welche den Raum in zwei Halbräume unterteilt.
- Man teilt an den durch Polygone induzierten Ebenen



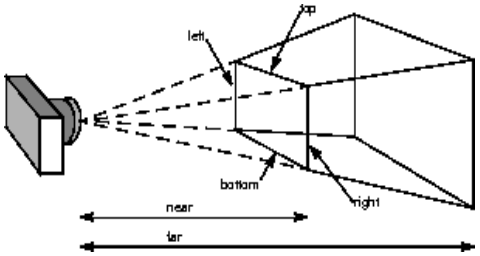
MATHEMATICAL AND APPLIED VISUAL COMPUTING

Grafikpipeline

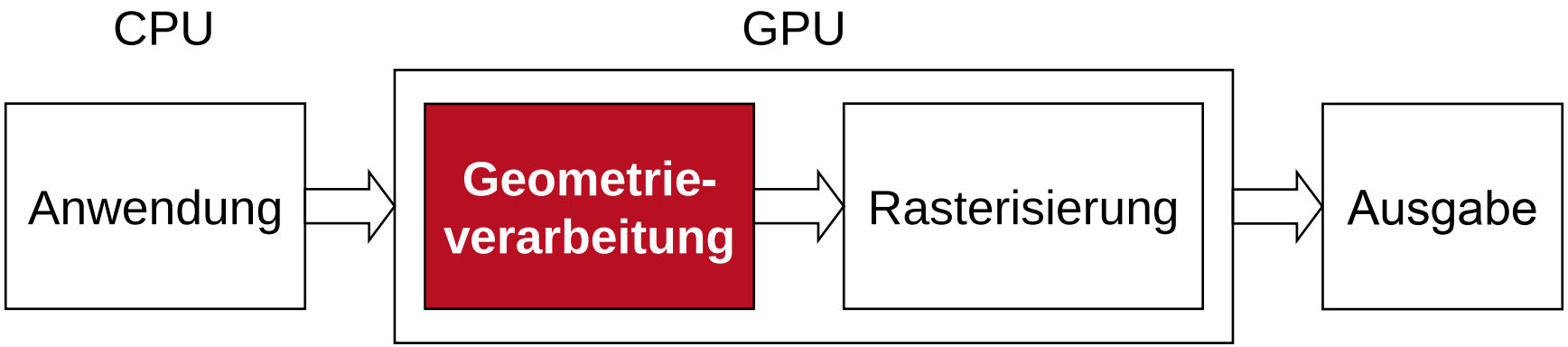


Modellierung:

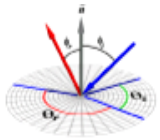
- Betrachter (Kamera)
- Lichtquellen

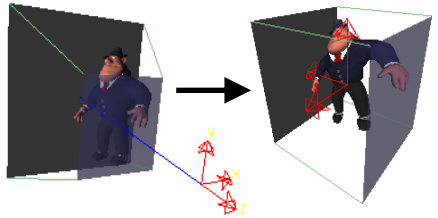


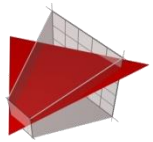
Grafikpipeline: Geometrieverarbeitung

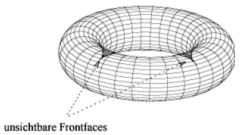


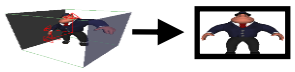
- **Modell Transformation (kanonische Orientierung + Position)**
- Simulation der Beleuchtung (in den Knoten)
- Perspektivische Transformation (kanonisches Sichtvolumen)
- Clipping (Abschneiden)
- Culling (Verdeckungsrechnung im Objektraum)
- Projektion



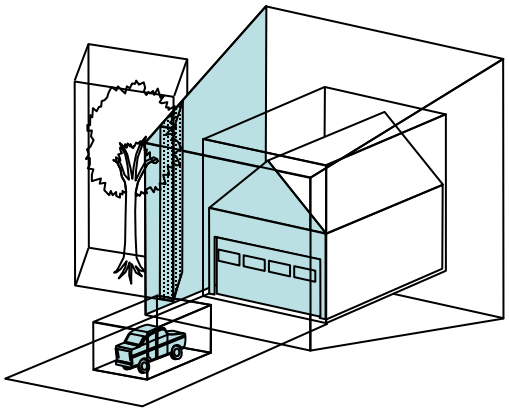






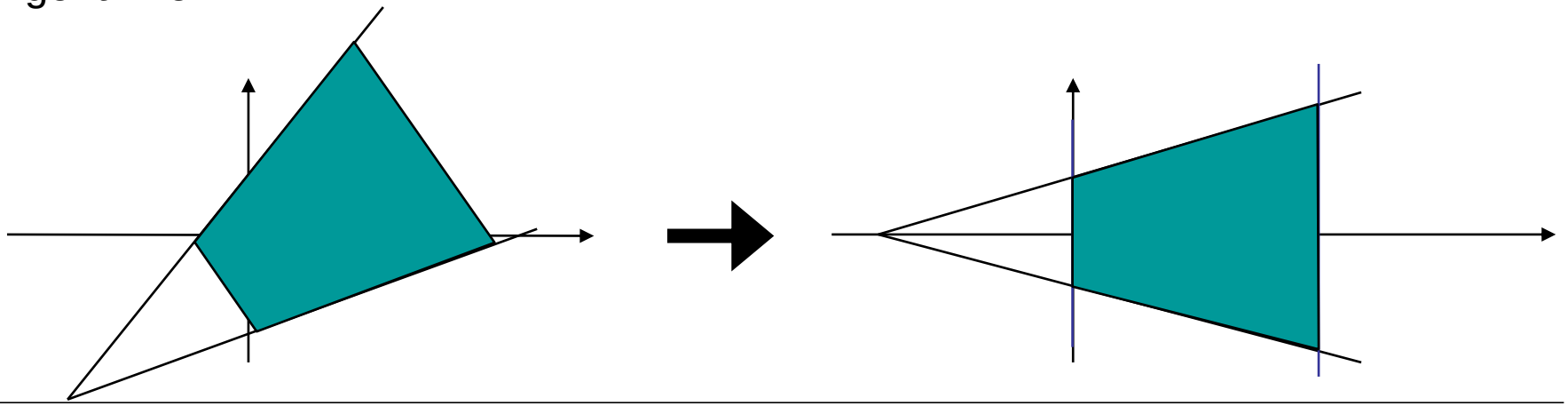


Kanonische Position und Orientierung

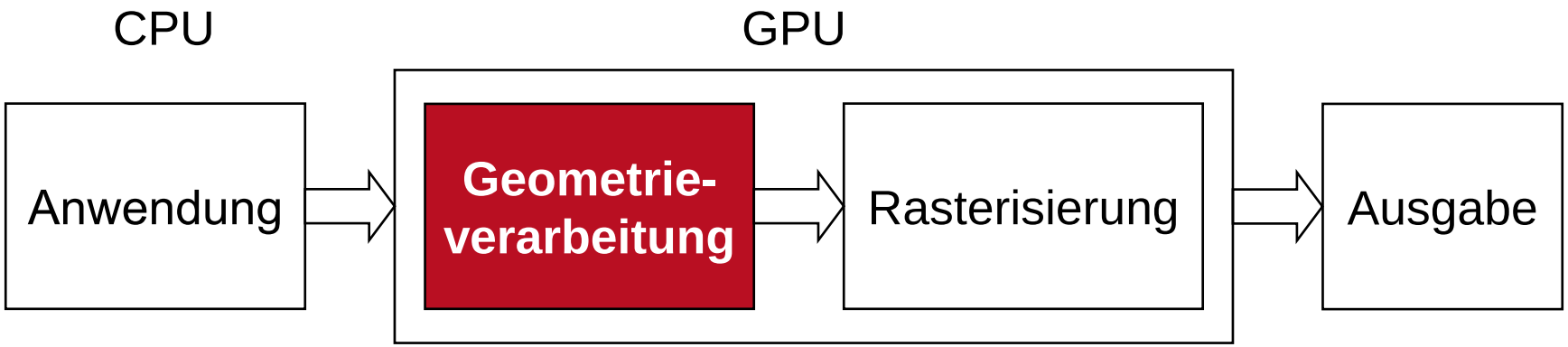


Sichtvolumen (und Auge) wird nach dem Koordinatensystem ausgerichtet

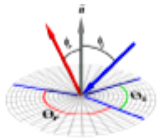
- Parallelprojektion (Kamera im Unendlichen): Sichtvolumen = Einheitswürfel
- Perspektive: Sichtvolumen = Pyramide
- Ziel: Einheitliches Sichtvolumen für die nachfolgenden Algorithmen

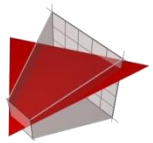


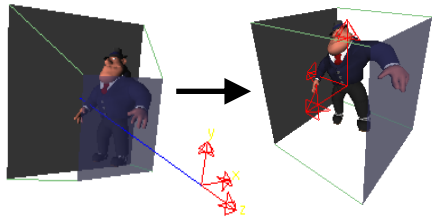
Grafikpipeline: Geometrieverarbeitung

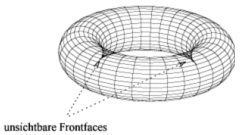


- Modell Transformation (kanonische Orientierung + Position)
- **Simulation der Beleuchtung (in den Knoten)**
- Perspektivische Transformation (kanonisches Sichtvolumen)
- Clipping (Abschneiden)
- Culling (Verdeckungsrechnung im Objektraum)
- Projektion







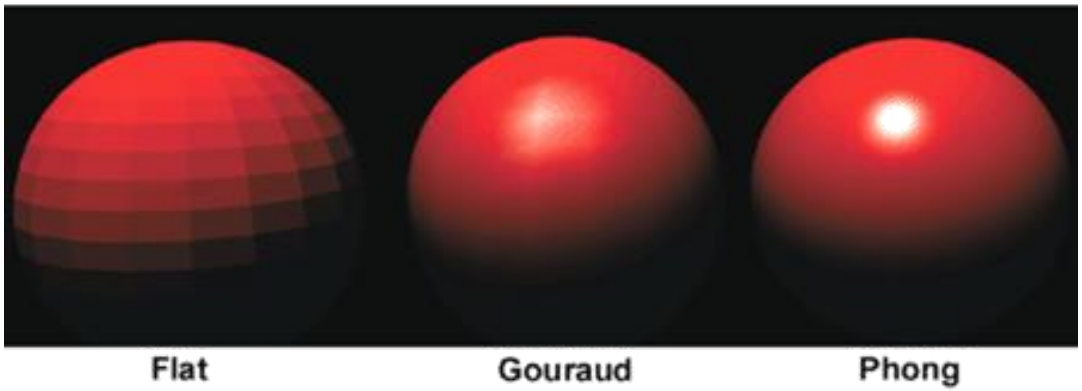


Beleuchtung eines Primitivs

TECHNISCHE UNIVERSITÄT DARMSTADT

Zunächst: Bestimmung der Leuchtdichten des Primitivs

Die Bestimmung der Leuchtdichten pro Pixel findet in der Praxis erst während der Rasterisierung statt



Flat Shading

- Normale des Primitivs ergibt einheitliche Helligkeit

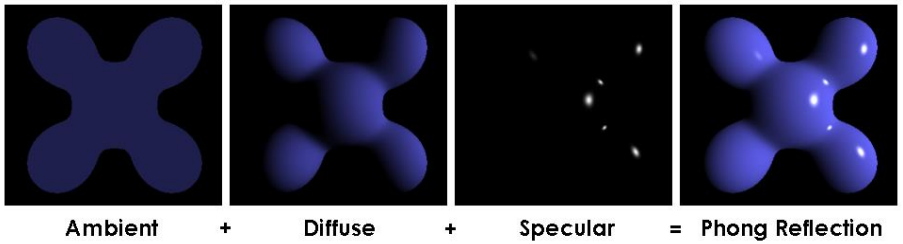
Gouraud Shading

- Normale in den Eckpunkten ergibt Helligkeitswerte für die Eckpunkte
- Helligkeitswerte der Eckpunkte werden linear interpoliert

Phong Shading

- Eckpunkt-Normalen werden für jeden Punkt linear interpoliert und normiert
- Helligkeitswert ergibt sich aus interpolierter Normale

Phong-Beleuchtungsmodell



Physik: Welcher Anteil des Lichts, das an der Position L emittiert und am Objekt reflektiert wurde, erreicht den Beobachter an der Position V ?

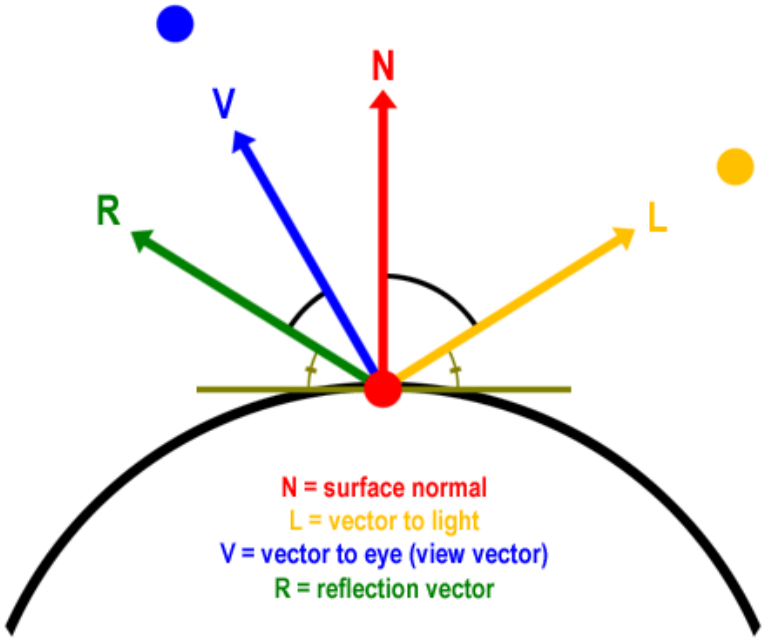
- Lichtquellen sind punktförmig
- Die Geometrie der Oberflächen, außer den Oberflächennormalen, wird ignoriert

$$I_{total} = I_{amb} + I_{diff} + I_{spec}$$

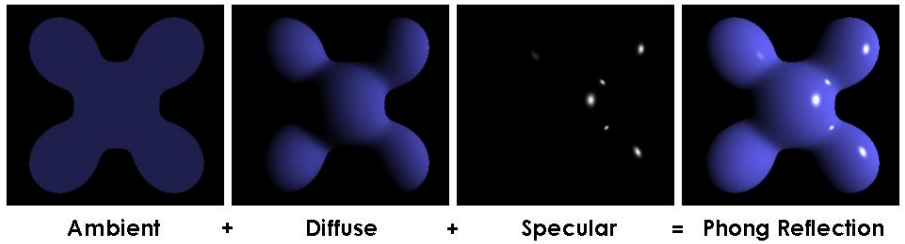
Diffuse I_{diff} und

Spiegelnde Reflexion I_{spec} werden nur lokal modelliert

Ambiente Reflexion I_{amb} wird global modelliert



Phong-Beleuchtungsmodell



Ambiente Komponente: Umgebungslicht C, mat. Konst. k; richtungsunabhängig

$$I_{amb} = k_{amb} C_{amb}$$

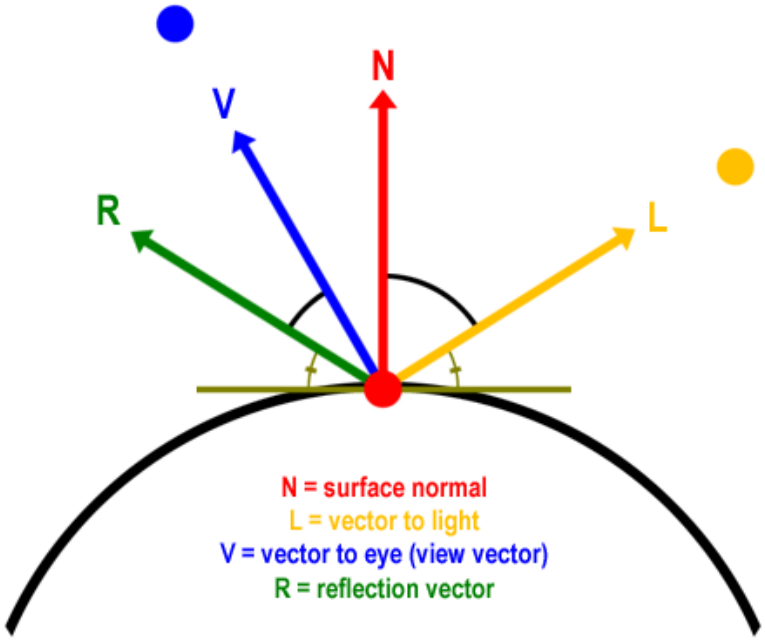
Diffuse Reflektion: abhängig von der Richtung des Lichts und Oberflächennormalen

$$I_{diff} = k_{diff} C_{light} (\underline{N} \cdot \underline{L})$$

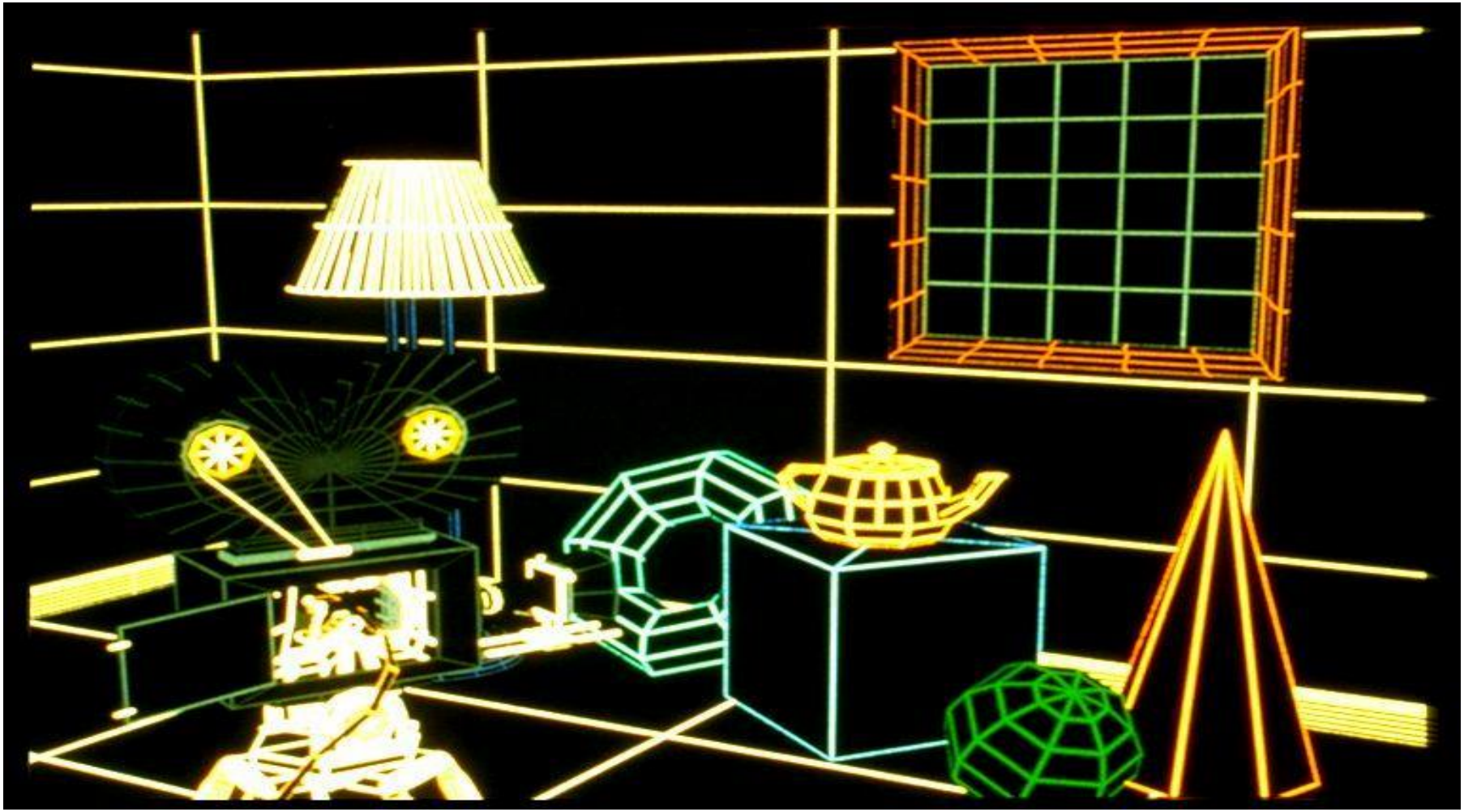
Spiegelnde Reflektion: abhängig von der Richtung der Reflektion und des Betrachters (m = „Rauhigkeit“)

$$I_{spec} = k_{spec} C_{light} (\underline{R} \cdot \underline{V})^m$$

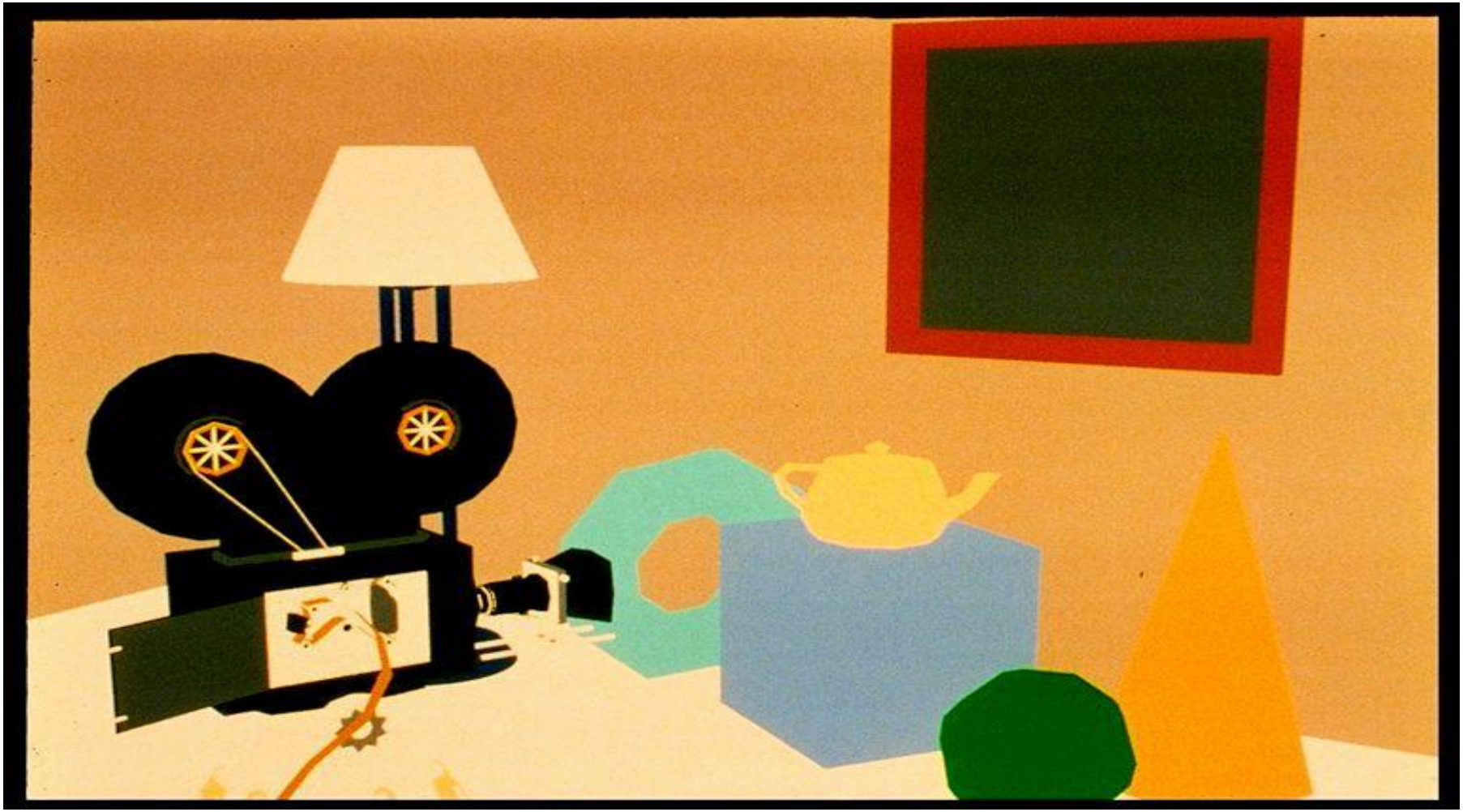
$$I_{total} = I_{amb} + I_{diff} + I_{spec}$$



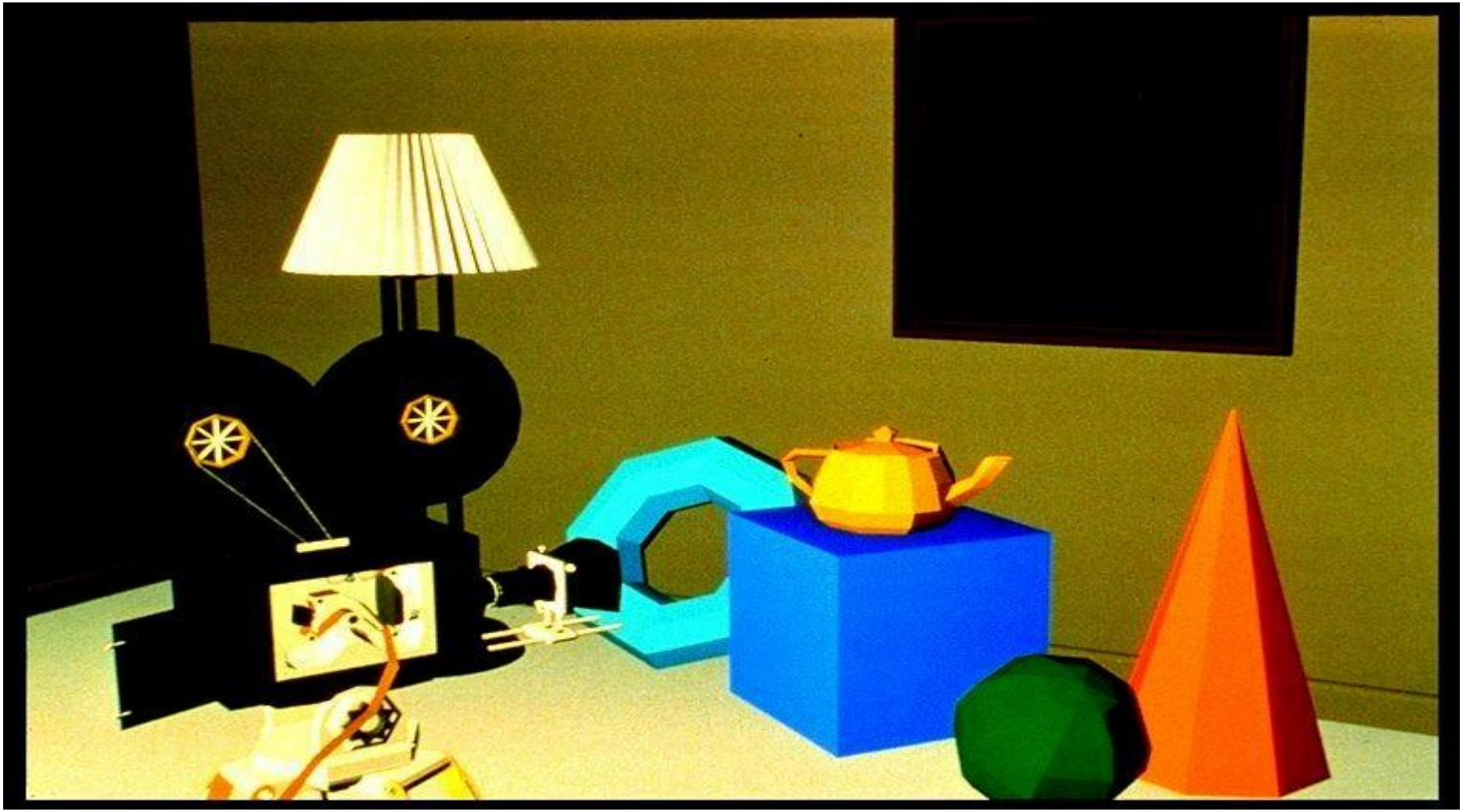
Schattierung von Primitiven
Wireframe



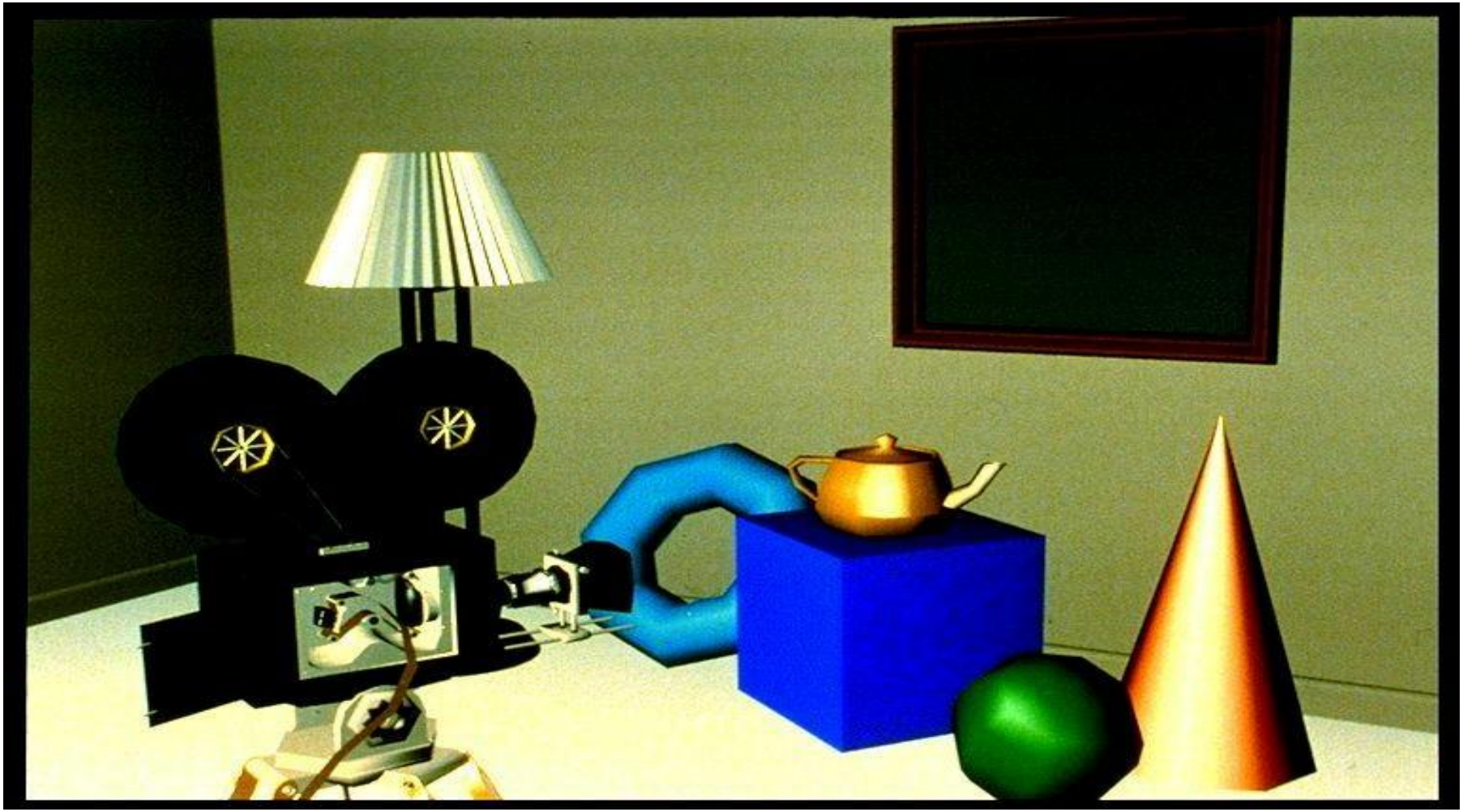
Schattierung von Primitiven
Constant Shading



Schattierung von Primitiven
Flat Shading

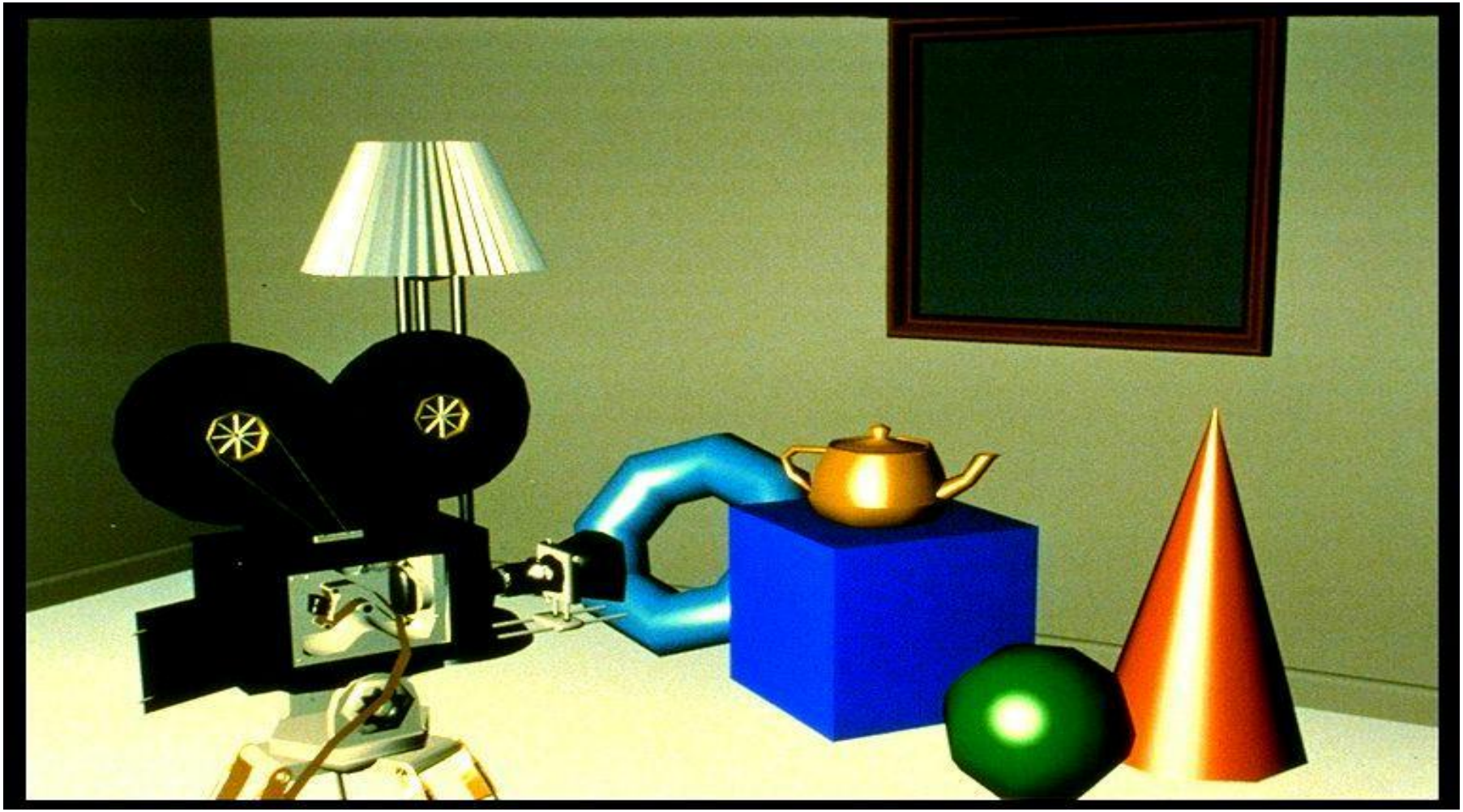


Schattierung von Primitiven
Gouraud Shading



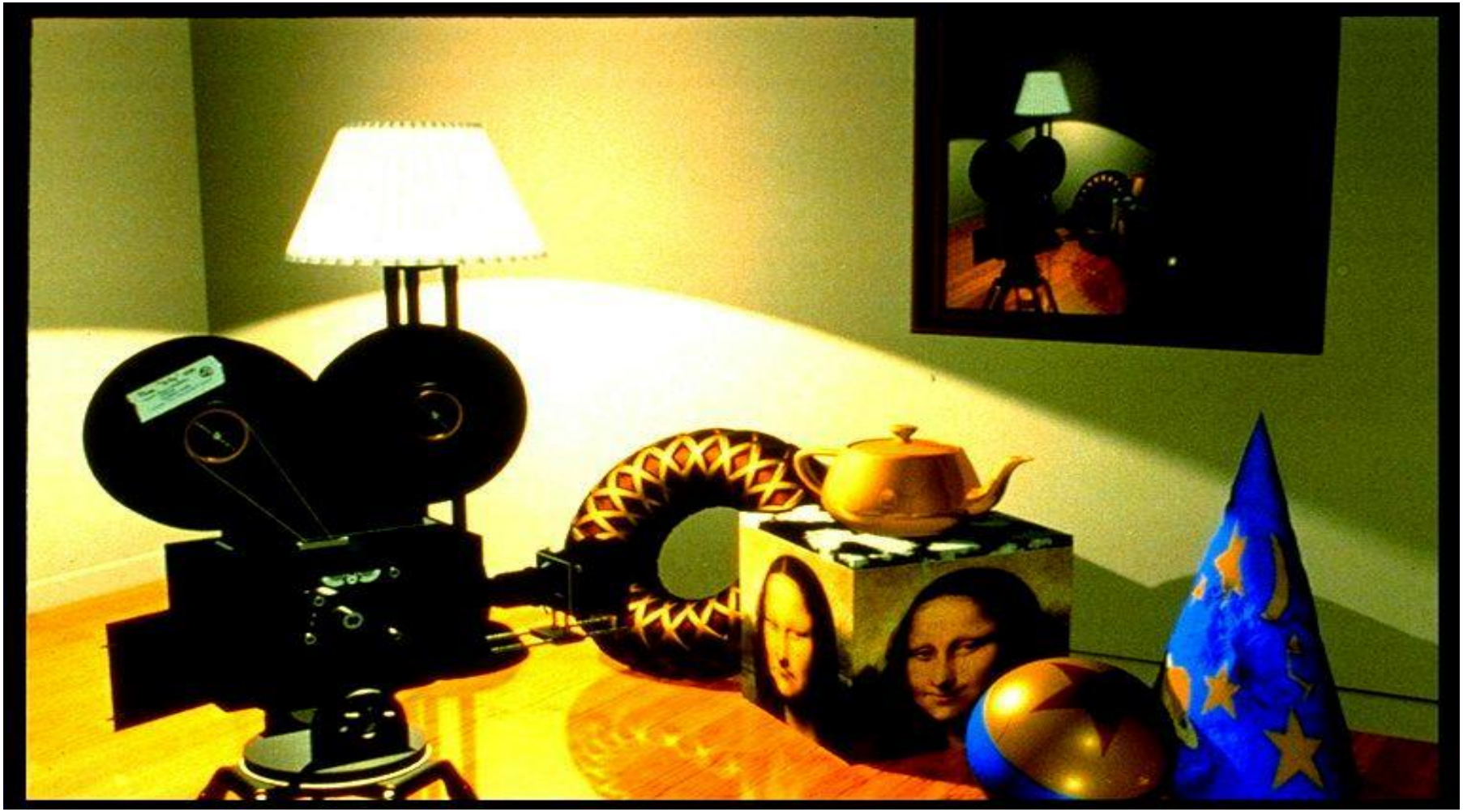
Schattierung von Primitiven
Phong Shading

TECHNISCHE UNIVERSITÄT DARMSTADT



Beleuchtung eines Primitivs
Subdivision, Texture/Bump/Reflection Mapping

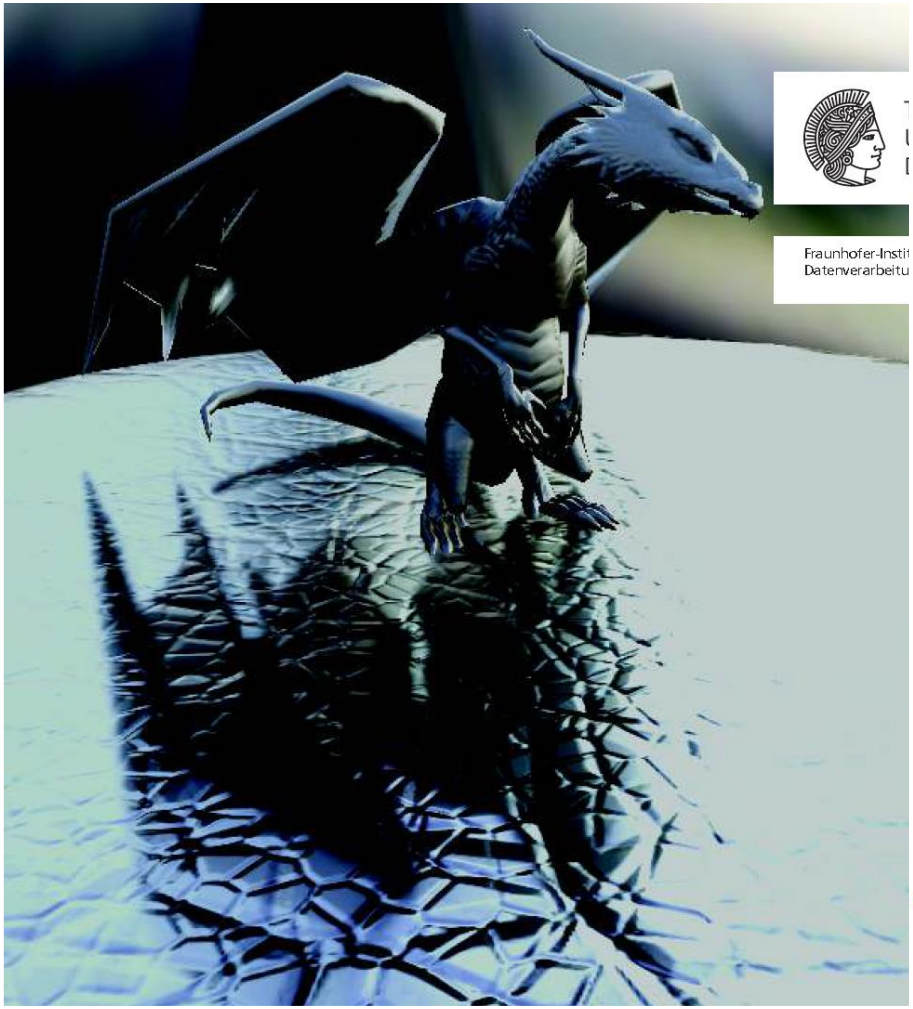
TECHNISCHE UNIVERSITÄT DARMSTADT

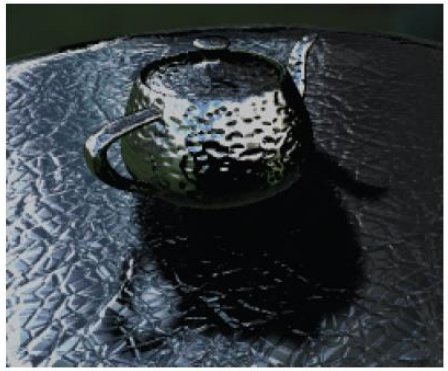


MATHEMATICAL AND APPLIED VISUAL COMPUTING

MSc Thesis: Efficient Self-Shadowing using Image Based Lighting on Glossy Surfaces

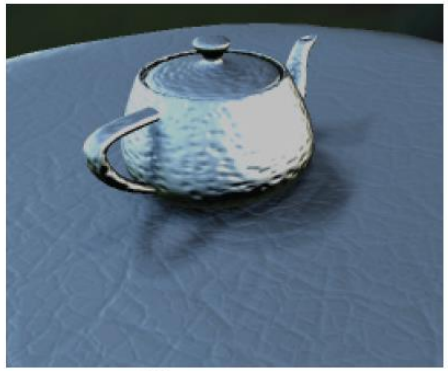
TECHNISCHE UNIVERSITÄT DARMSTADT



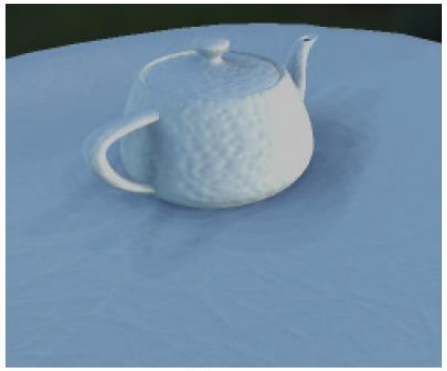


(a) Spiegelnd

(b) Hoch glänzend



(c) Leicht glänzend



(d) Matt

Abbildung 6.3: Verschiedene Reflektionsgrade von komplett spiegelnd bis diffus. (a) bis (d) zeigen die dynamische Anpassung der Reflexivität in Echtzeit. Dabei verändert sich sowohl die Farbgebung der Objekte als auch die Charakteristika des Schattenwurfs.

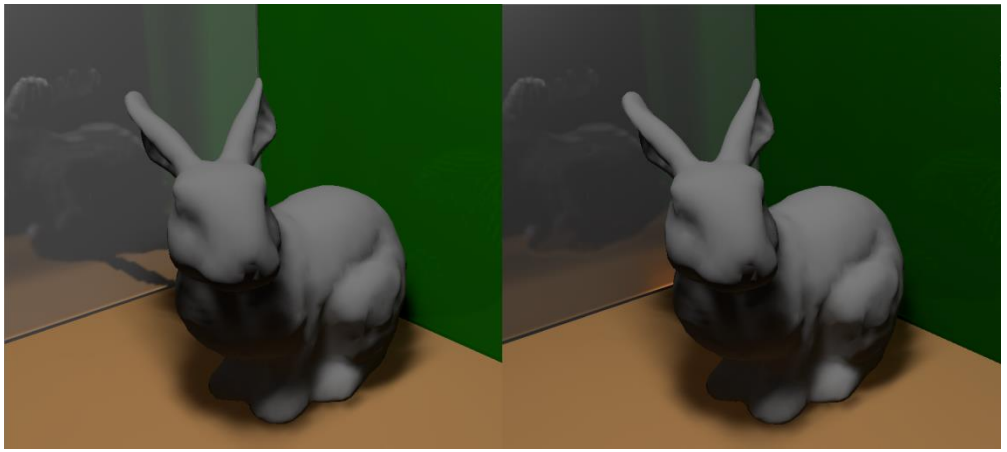
MATHEMATICAL AND APPLIED VISUAL COMPUTING

Msc Thesis:
Delta Global Illumination für Mixed Reality

The effect of lighting in Mixed Reality.

In this image a virtual object is just pasted above the background image.

No realistic perception can be achieved.



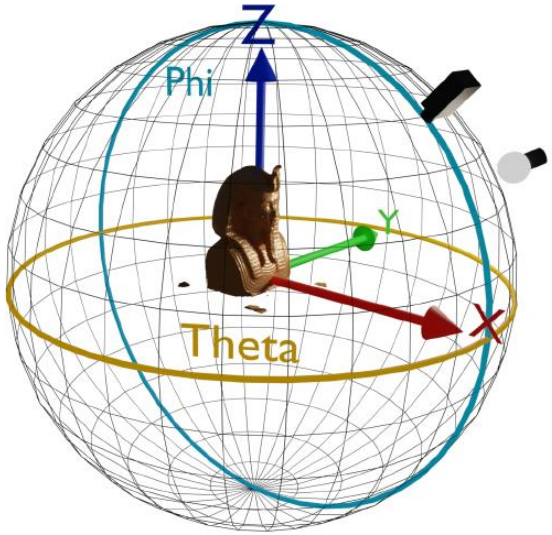
Left: The reference image.
The background scene and the bunny were merged before the lighting took place.

Right: The Relighting result.
The Bunny is added to an already lit scene and is relit with Differential Rendering.

Ba-Thesis: Erfassung und Rendering

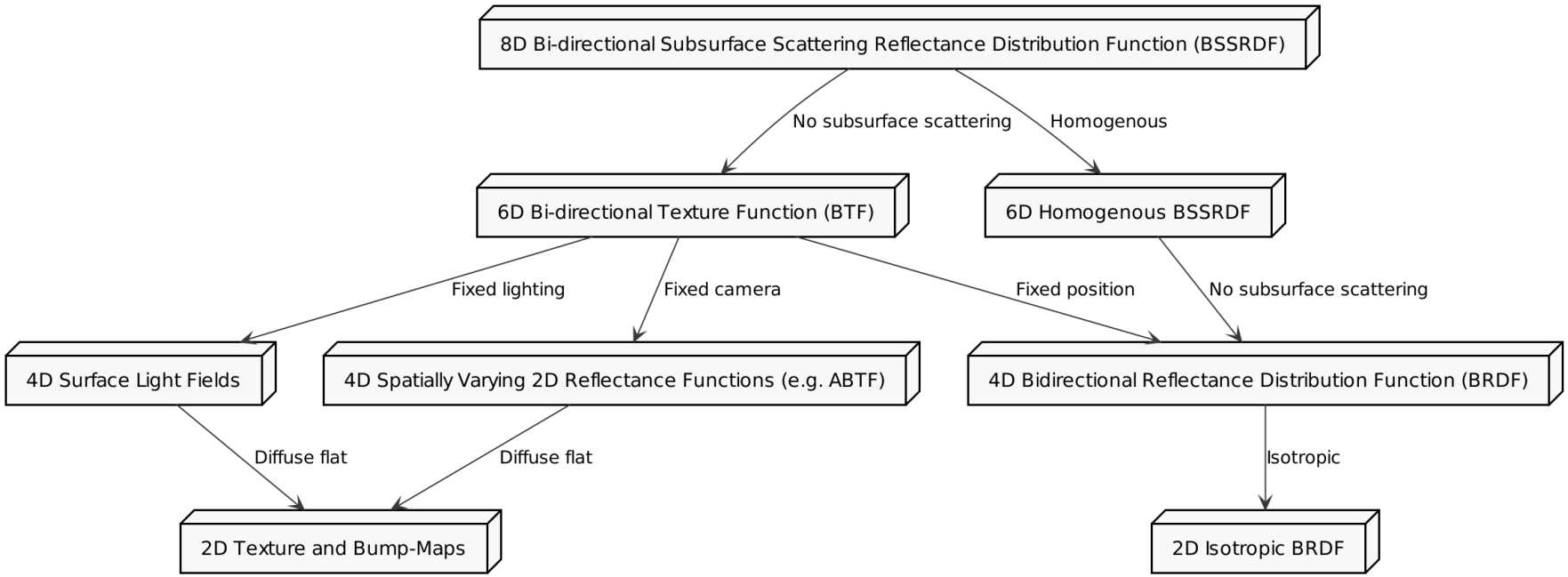
TECHNISCHE UNIVERSITÄT DARMSTADT

Phänomenologische Erfassung und Rendering des Optischen Materialverhaltens ganzheitlicher 3D-Objekte

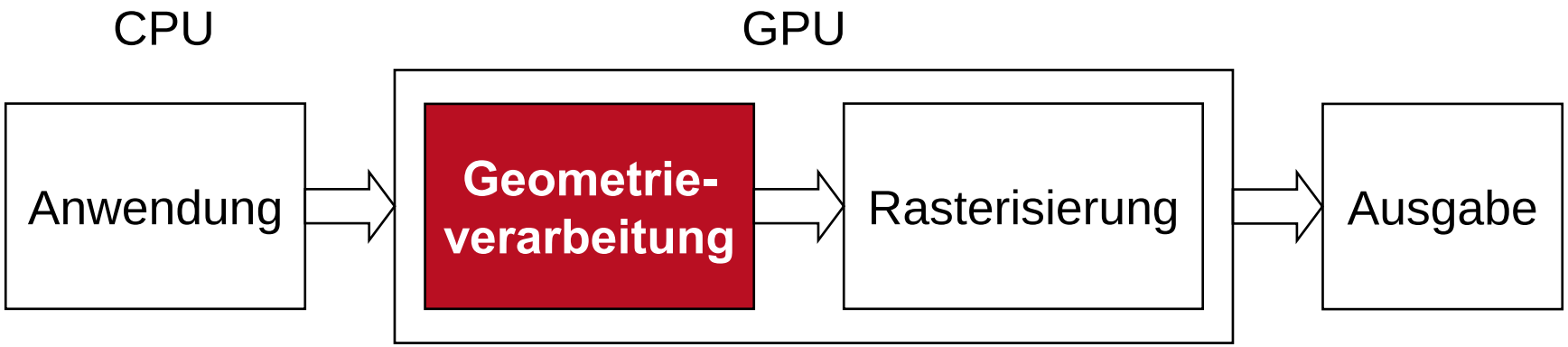


MATHEMATICAL AND APPLIED VISUAL COMPUTING

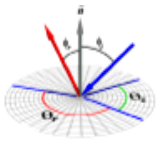
Reflectance

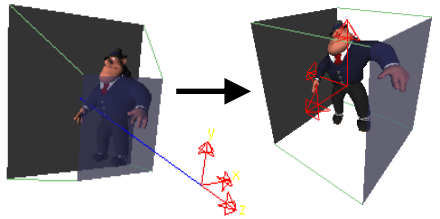


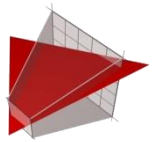
Grafikpipeline: Geometrieverarbeitung

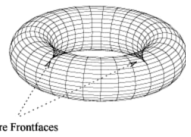


- Modell Transformation (kanonische Orientierung + Position)
- Simulation der Beleuchtung (in den Knoten)
- **Perspektivische Transformation (kanonisches Sichtvolumen)**
- **Clipping (Abschneiden)**
- Culling (Verdeckungsrechnung im Objektraum)
- Projektion





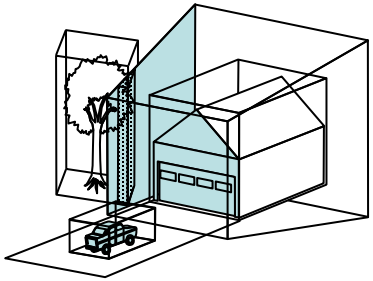


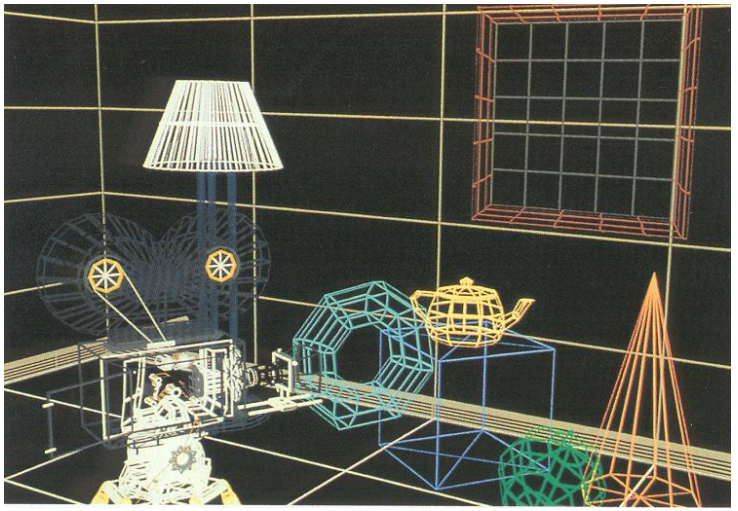


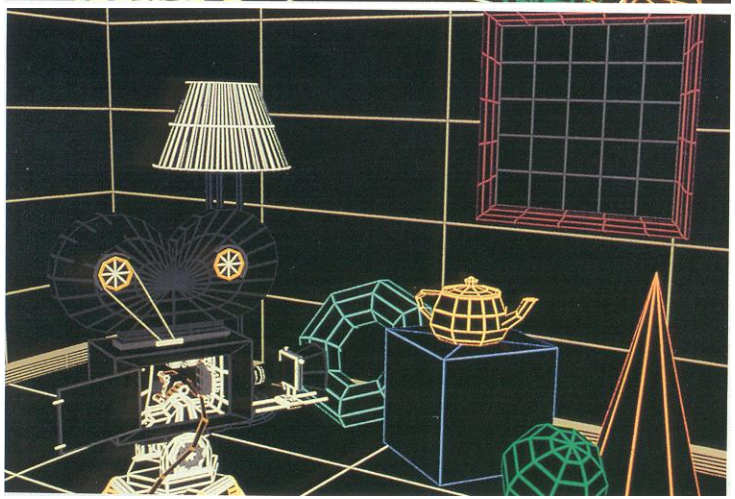
Sichtbarkeit

TECHNISCHE UNIVERSITÄT DARMSTADT

Verdeckung







- Sichtbar ist der dem Auge am nächsten liegende Punkt
- Ist das Objekt durchsichtig (transparent), wird der dahinterliegende Punkt auch sichtbar, usw.
- Clipping: Abschneiden von Objekten am Rand eines gewünschten Bildschirmausschnittes

Painters Algorithmus

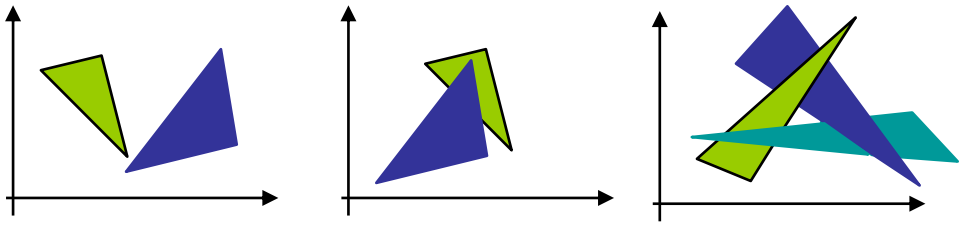
TECHNISCHE UNIVERSITÄT DARMSTADT

Zeichne Polygone wie ein Maler

- Tiefe: z-Wert, $z \in [z_{min}, z_{max}]$
- Das am weitesten entfernte zuerst (z_{max})

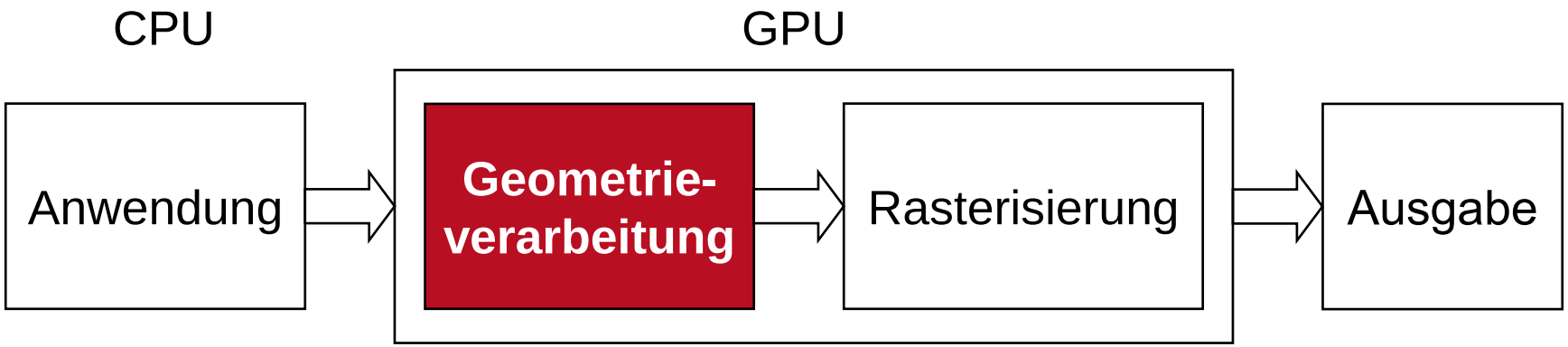
Algorithmus:

- Sortiere Polygone nach z-Wert
- Falls z-Intervalle überlappen müssen Schnittpolygone berechnet werden
- Beginne das Zeichnen mit dem Polygon mit größtem z-Wert

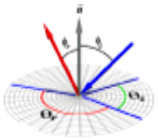


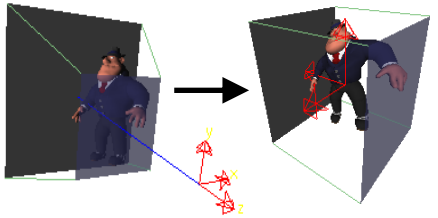
Komplexität ist $O(n^2)$ (n =Anzahl Dreiecke)

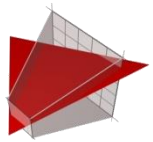
Grafikpipeline: Geometrieverarbeitung

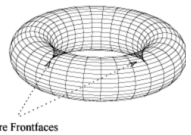


- Modell Transformation (kanonische Orientierung + Position)
- Simulation der Beleuchtung (in den Knoten)
- Perspektivische Transformation (kanonisches Sichtvolumen)
- Clipping (Abschneiden)
- **Culling (Verdeckungsrechnung im Objektraum)**
- Projektion



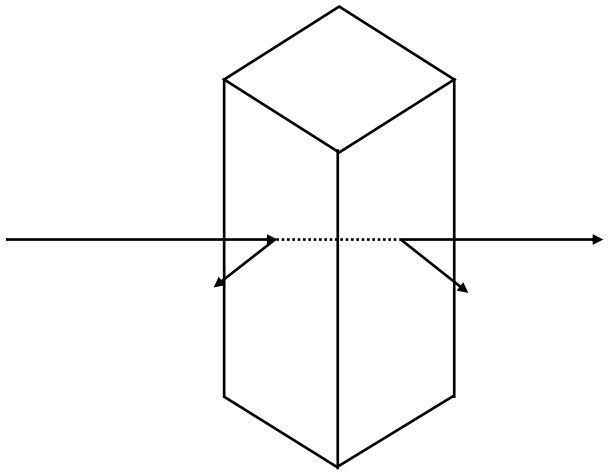






Culling

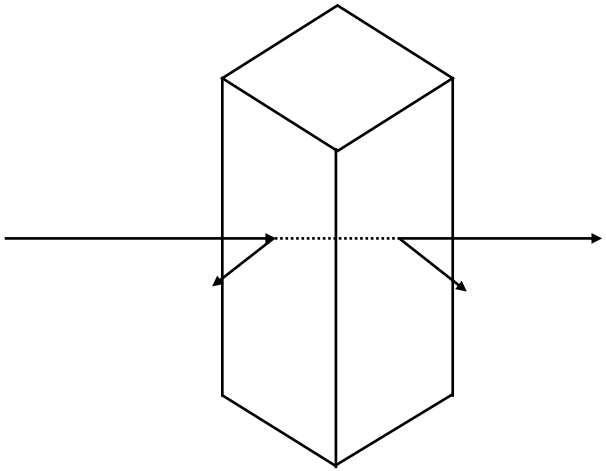
- Rückseiten machen etwa die Hälfte der in der Szene vorkommenden Flächen aus
- Sie werden von der weiteren Betrachtung ausgeschlossen
- Bei konvexen Körpern sind alle verdeckten Flächen Rückseiten
  - Bestimmung der Rückseiten beseitigt genau alle unsichtbaren Bildteile für einen konvexen Körper

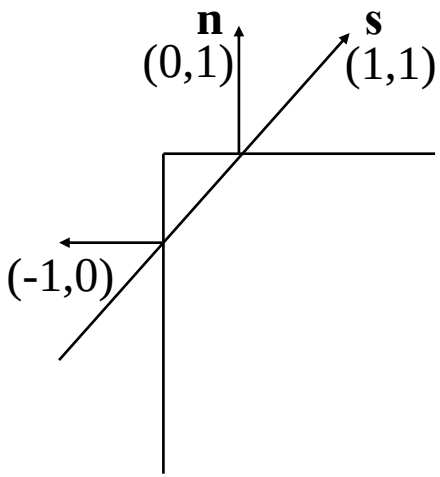


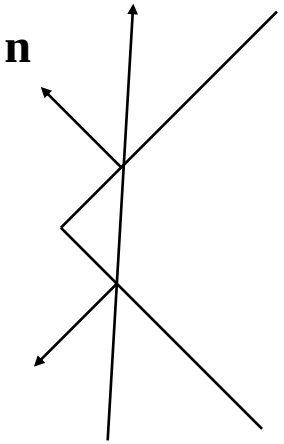
Culling

Erkennen von Rückseiten:

- Das Skalarprodukt aus Sehstrahl und Normale ist positiv ($\mathbf{n} \cdot \mathbf{s} > 0$)







Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

Spatial Data Structures for Accelerated Visibility Determination on the Web (Web3D 2014 / Ma-Thesis)

→ Sehe VL X3DOM

Figure 9: *Different points of view during a walkthrough: outside (left) and inside (middle). Bounding volumes reaching out to far, thus interfering with occlusion culling.(right)*

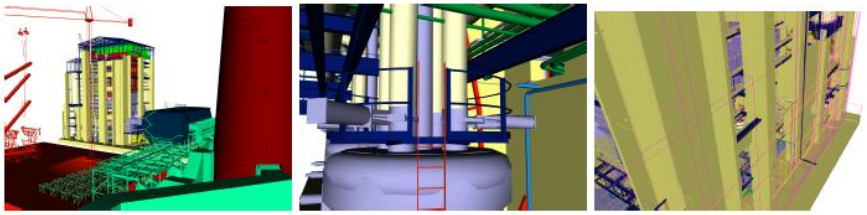
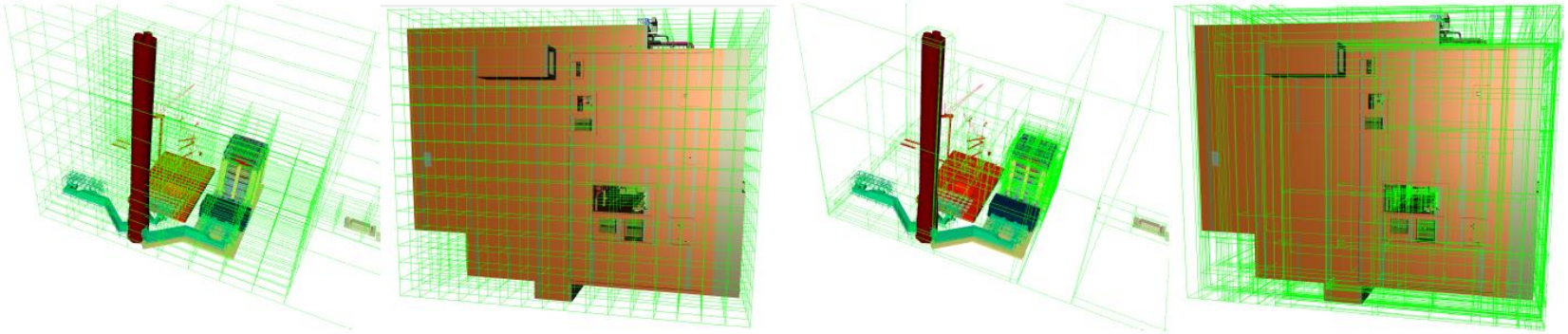
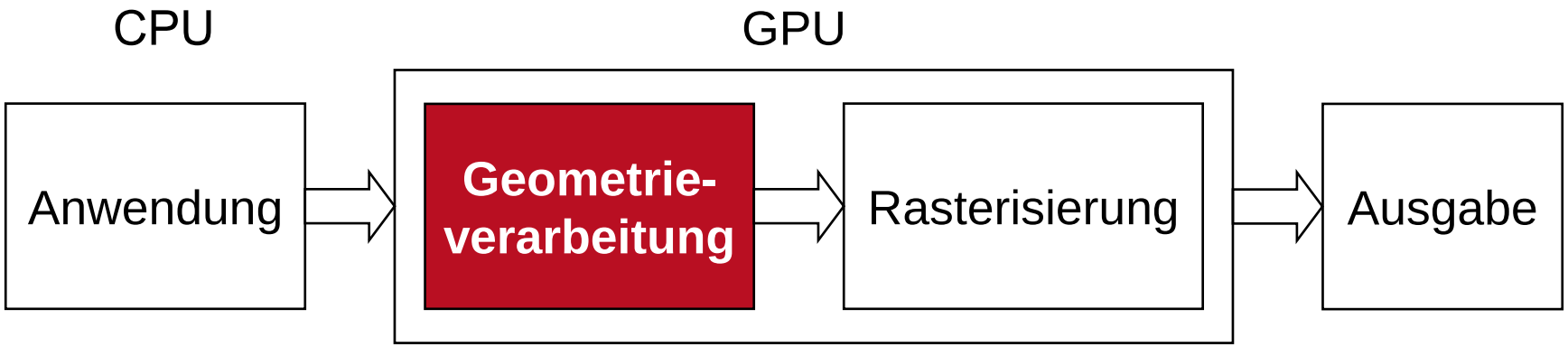


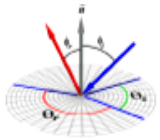
Figure 1: *Visualization of octree (left) and bounding interval hierarchy (right) for two powerplant models of ∼13/ ∼57 millions of triangles. Both structures are computed directly inside the browser. By accelerating the visibility determination, the data structures are the key to an interactive experience when rendering CAD data of such magnitude. Efficient hierarchies can be constructed in ∼20-30 ms already.*

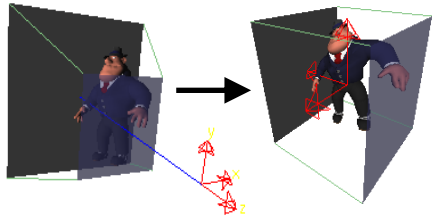


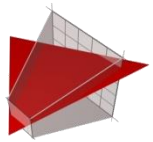
Grafikpipeline: Geometrieverarbeitung

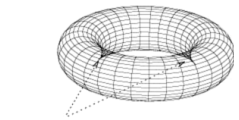


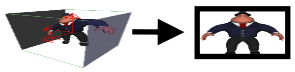
- Modell Transformation (kanonische Orientierung + Position)
- Simulation der Beleuchtung (in den Knoten)
- Perspektivische Transformation (kanonisches Sichtvolumen)
- Clipping (Abschneiden)
- Culling (Verdeckungsrechnung im Objektraum)
- **Projektion**



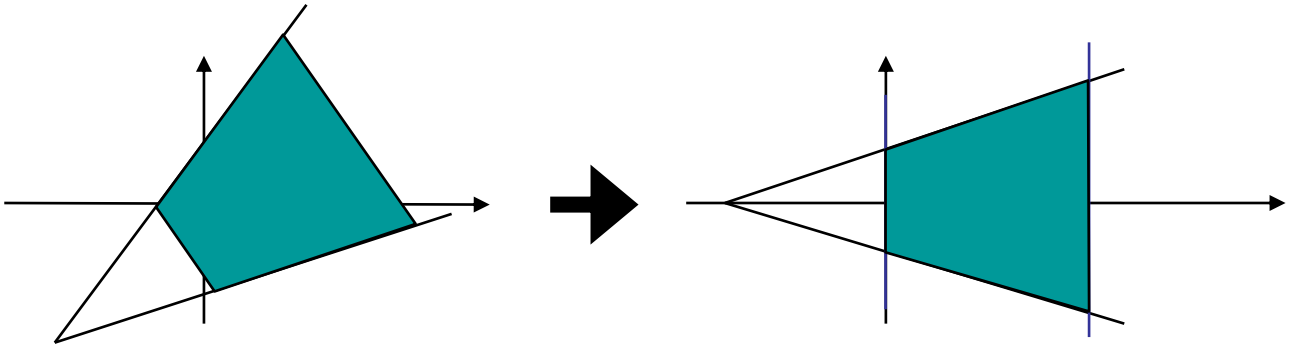




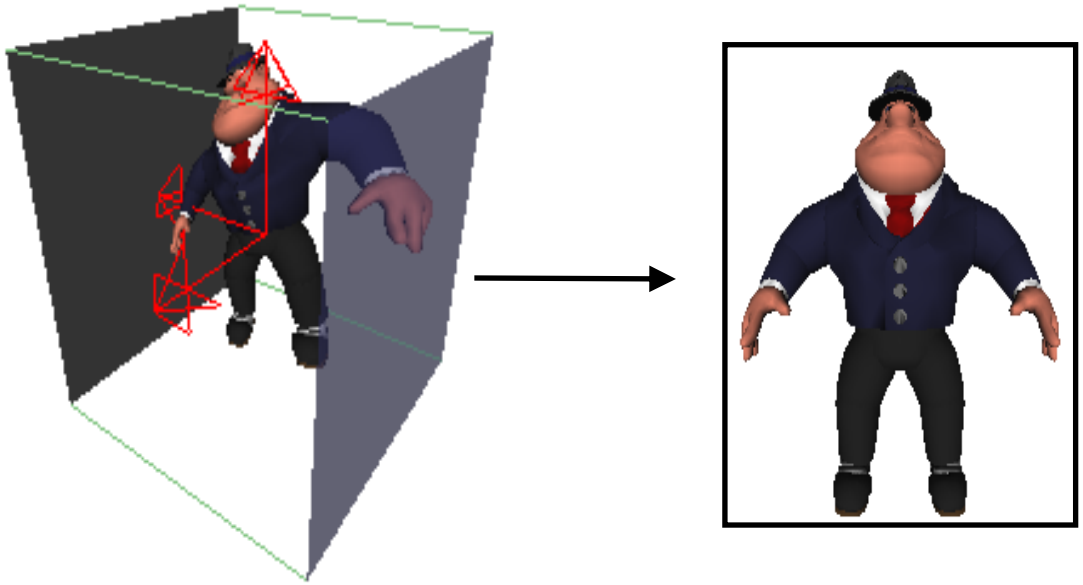




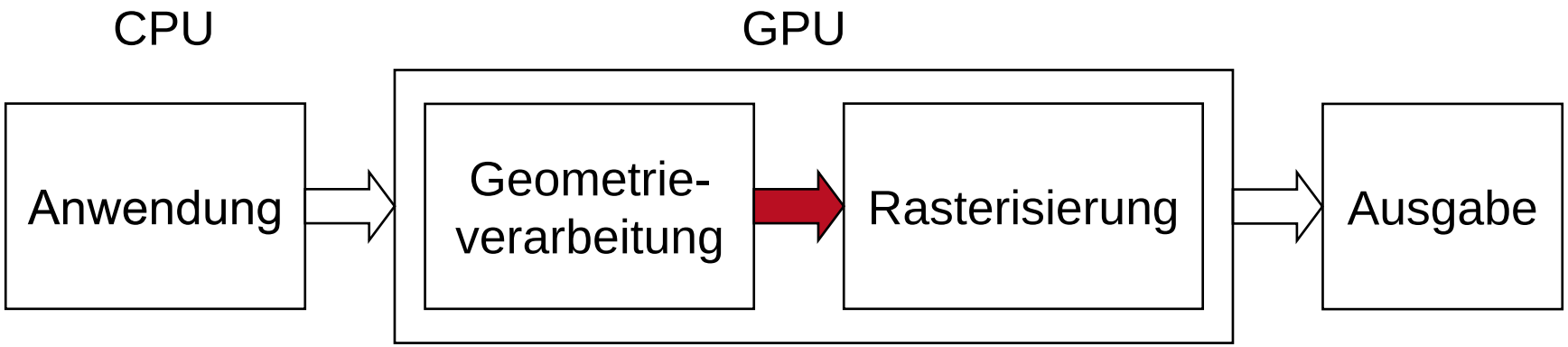
Projektion



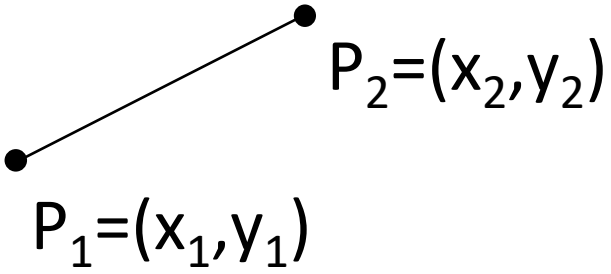
Nächste Vorlesung!



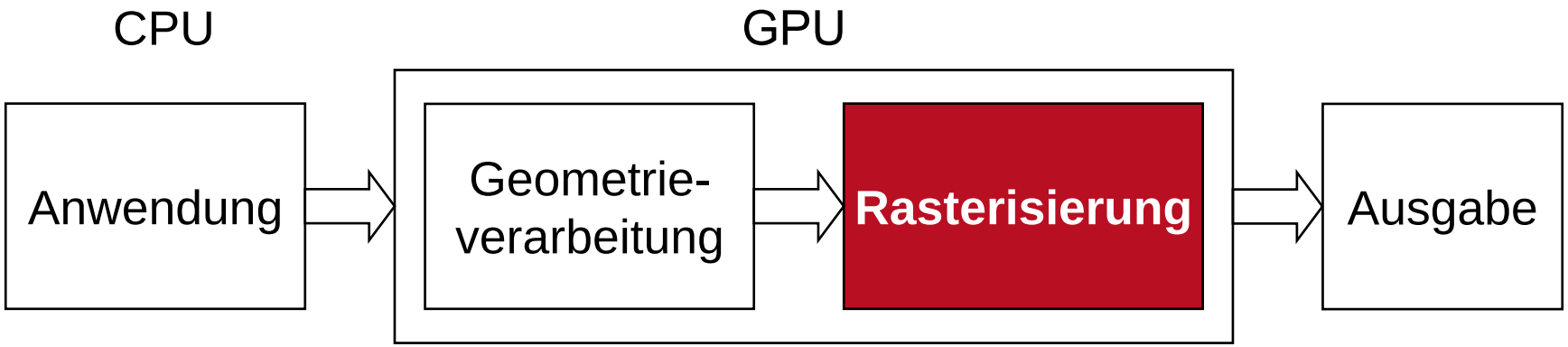
Grafikpipeline



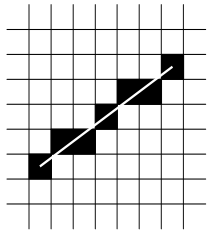
- 2D-Koordinaten und zusätzliche Daten:
  - z-Buffer Werte
  - Farbwerte pro Knoten / Primitiv

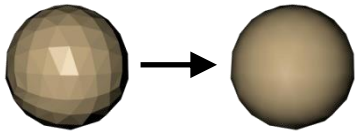


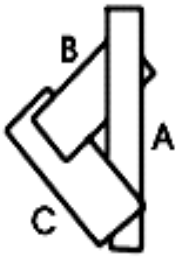
Grafikpipeline: Rasterisierung

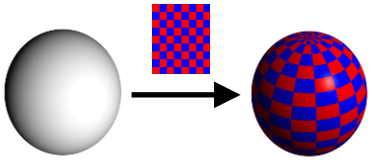


- **Rasterisierung der Primitive nacheinander:**
  - **Scan-Konvertierung**
  - Verdeckungsrechnung
  - (Farbwertinterpolation)





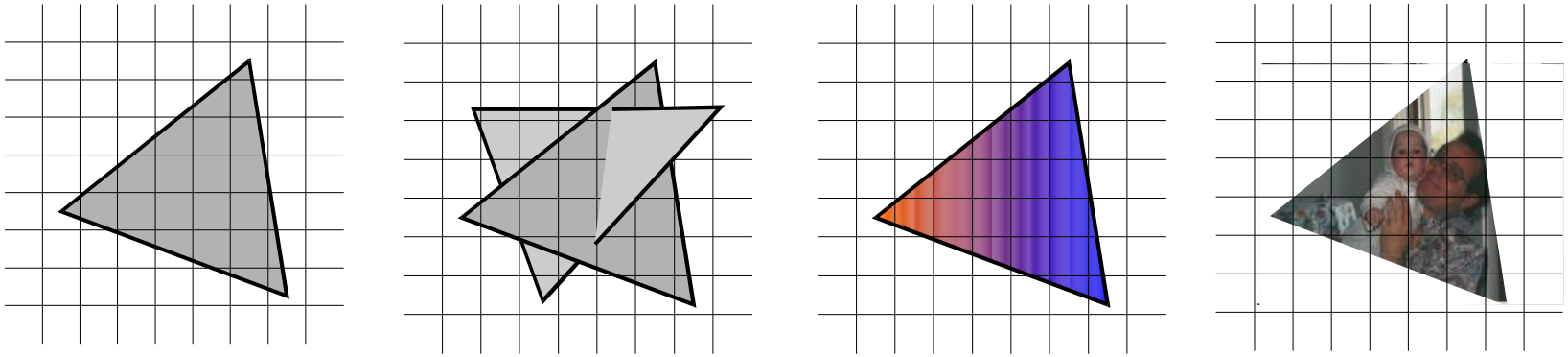




Rasteralgorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Rasterisierung: Primitive (Linien, Polygone) werden in Pixel zerlegt



Zusätzlich Operationen pro Pixel:

- Verdeckungsrechnung
- Shading

Rasterisierung von Linien
Algorithmus von Bresenham

TECHNISCHE UNIVERSITÄT DARMSTADT

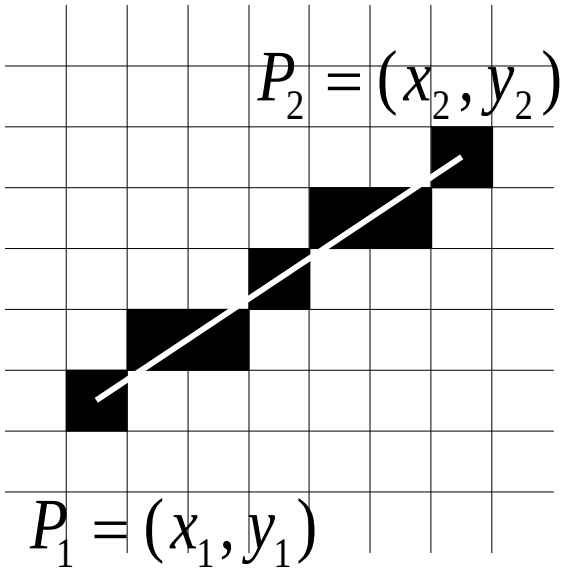
Der erste Integer-Algorithmus zum Zeichnen von Linien

- Bresenham (1965).

Herleitung:

- Anfangs- und Endpunkt auf dem Raster
- Steigung zwischen 0 und 1

$$\Delta x = x_2 - x_1 \geq 0,$$
$$\Delta y = y_2 - y_1 \geq 0,$$
$$\Delta x \geq \Delta y.$$



MATHEMATICAL AND APPLIED VISUAL COMPUTING

Rasterisierung von Linien
Algorithmus von Bresenham

TECHNISCHE UNIVERSITÄT DARMSTADT

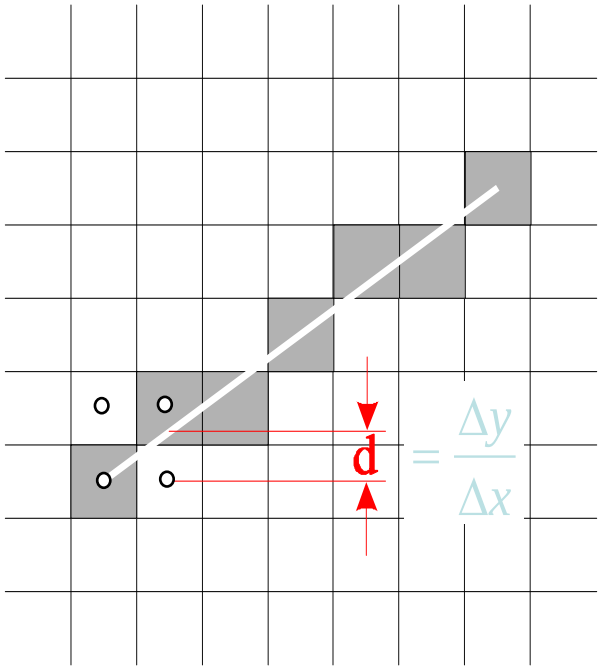
REM Bresenham-Algorithmus für eine Linie im ersten Oktanten
dx = xend-xstart
dy = yend-ystart
REM im ersten Oktanten muss 0 < dy <= dx sein, sonst vertauschen
REM Initialisierungen

- x = xstart
- y = ystart
- SETPIXEL x,y
- fehler = dx/2

REM Pixelschleife: immer ein Schritt in schnelle Richtung, wenn nötig in die langsame
WHILE x < xend

- REM Schritt in schnelle Richtung
- x = x + 1
- fehler = fehler-dy
- IF fehler < 0 THEN
  - REM Schritt in langsame Richtung
  - y = y + 1
  - fehler = fehler + dx
  - END IF
- SETPIXEL x,y

WEND



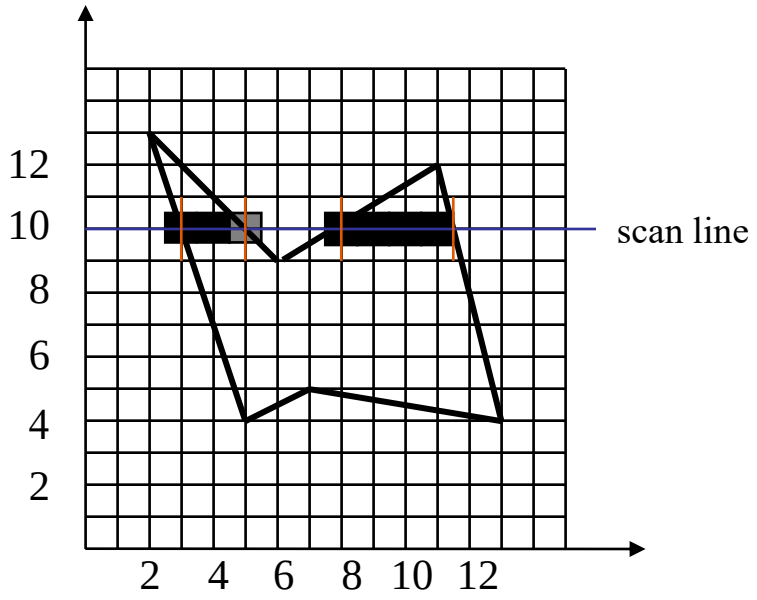
MATHEMATICAL AND APPLIED VISUAL COMPUTING

Rasterisieren von Polygonen

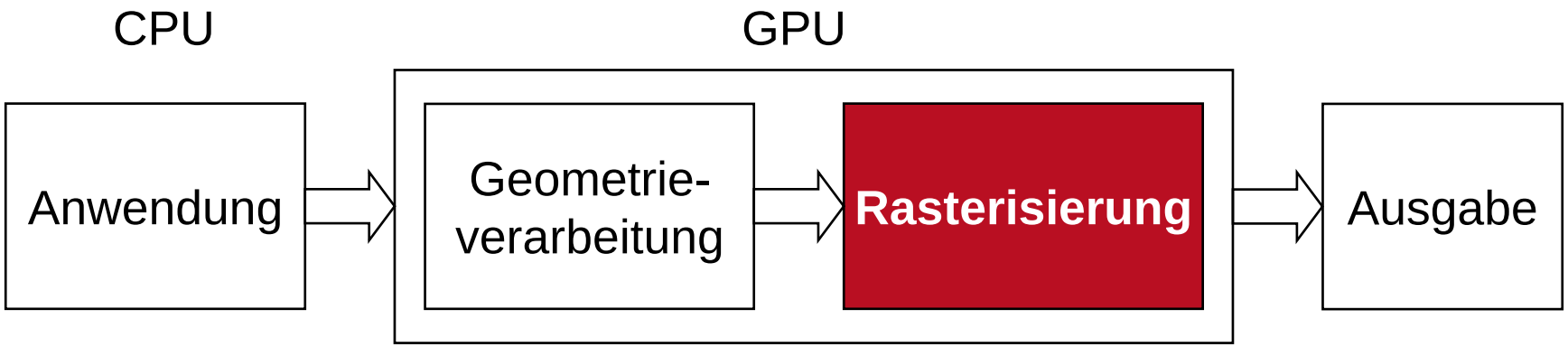
TECHNISCHE UNIVERSITÄT DARMSTADT

Idee: (Scanline Algorithmus)

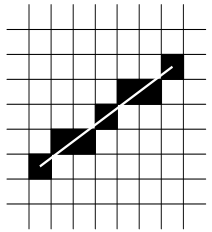
- Finde Schnittpunkt der scan line mit allen Kanten des Polygons
- Sortiere Schnittpunkte nach wachsender x-Koordinate
- Fülle Pixel zwischen Paaren aufeinanderfolgender Schnittpunkte
  - Regel von der ungeraden Parität
  - Parität ist am Anfang 0
  - mit jedem Schnittpunkt um eins inkrementiert
  - Pixel wird gesetzt falls Parität ungerade

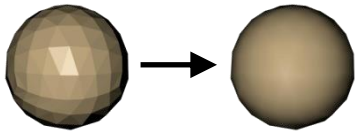


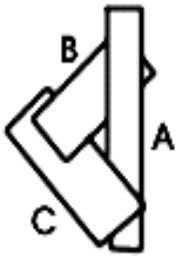
Grafikpipeline: Rasterisierung

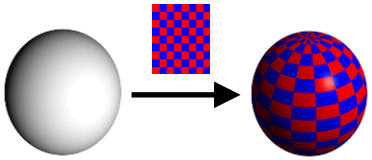


- **Rasterisierung der Primitive nacheinander:**
  - Scan-Konvertierung
  - **Verdeckungsrechnung**
  - (Farbwertinterpolation)





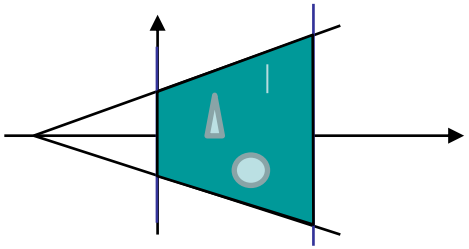




z-Buffer-Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

- Der Tiefenspeicher wurde bereits 1974 vorgeschlagen, aber aus Kostengründen damals nicht realisiert
- Für jeden Bildpunkt wird auch ein z-Wert gespeichert
  - Tiefenbild
- Initialisierung
  - Der Bildspeicher ➔ Hintergrundfarbe
  - z-Speicher ➔ maximaler z-Wert
- Alle Objekte der Szene werden nacheinander gerastert
  - Eine besondere Reihenfolge ist nicht erforderlich



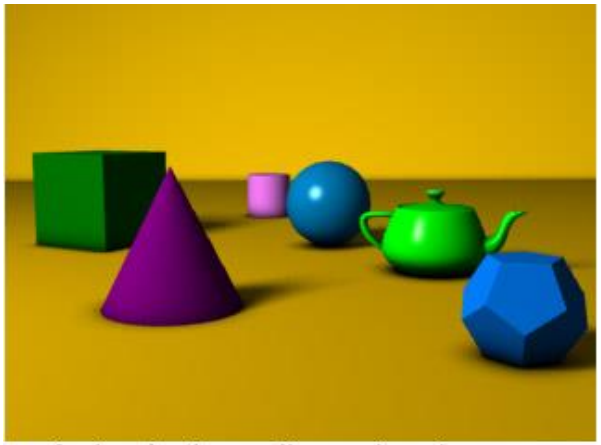
z-Buffer-Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Für jeden Punkt (x,y) eines darzustellenden Polygons

1. Berechne z(x,y)
   - Perspektivische Transformation erlaubt keine lineare Interpolation mehr!
2. Ist z(x,y) kleiner als der unter (x,y) bereits gespeicherte Wert
   - Schreibe z(x,y) in den z-Speicher und den zugehörigen Farbwert an der Stelle (x,y) in den Bildspeicher.

Nach der Behandlung aller Objekte steht im Bildspeicher das gewünschte Bild der sichtbaren (Teil-)flächen



A simple three dimensional scene

Z-buffer representation

z-Buffer-Algorithmus
Vorteile

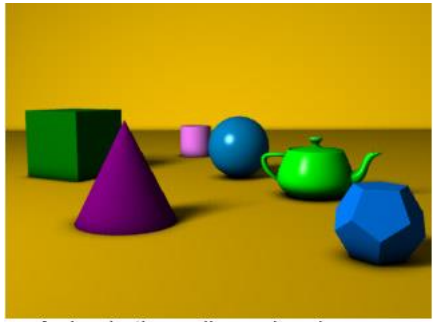
TECHNISCHE UNIVERSITÄT DARMSTADT

- Jede Szene mit jeder Art von Objekten kann behandelt werden
- Komplexität ist unabhängig von der Tiefenkomplexität
- In eine fertige Szene können nachträglich Objekte eingefügt werden
    - Dies ist besonders für die Kombination mit Kameraaufnahmen interessant
- Spezielle Objekte, wie z.B. ein 3D-Cursor, können in der Szene mit korrekter Verdeckung dargestellt werden
- Leicht in Hardware zu realisieren.

z-Buffer-Algorithmus
Nachteile

TECHNISCHE UNIVERSITÄT DARMSTADT

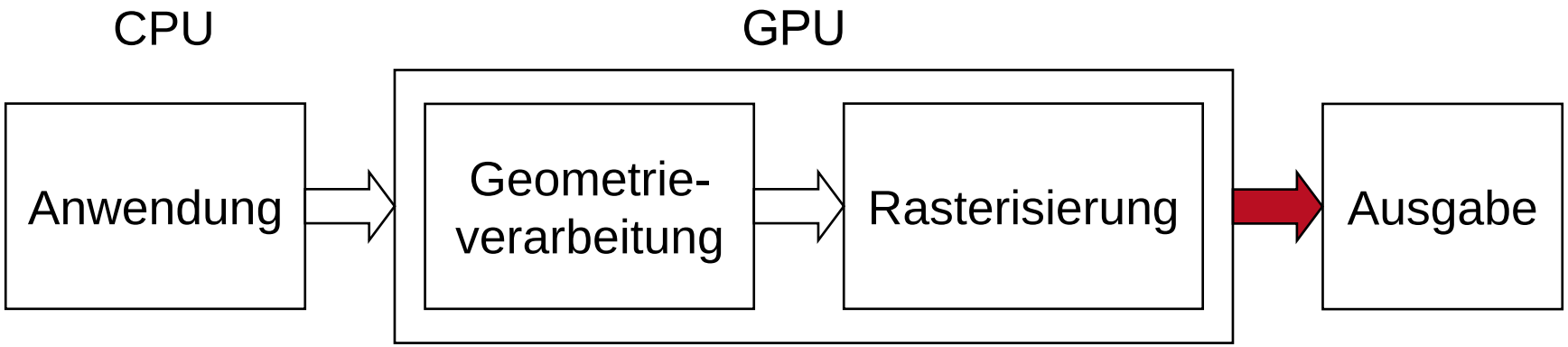
- Pro Bildpunkt wird nur ein Objekt gespeichert
  - Daraus resultieren Abtastfehler, die durch Supersampling [höhere Auflösung] verkleinert aber nicht beseitigt werden können
- Transparenz ist prinzipiell nicht realisierbar
- Die Genauigkeit des z-Buffers ist beschränkt
  - Getrennte Objekte erhalten denselben z-Wert
  - Die Farbe wird dann von der Objektreihenfolge bei der Rasterung bestimmt



A simple three dimensional scene

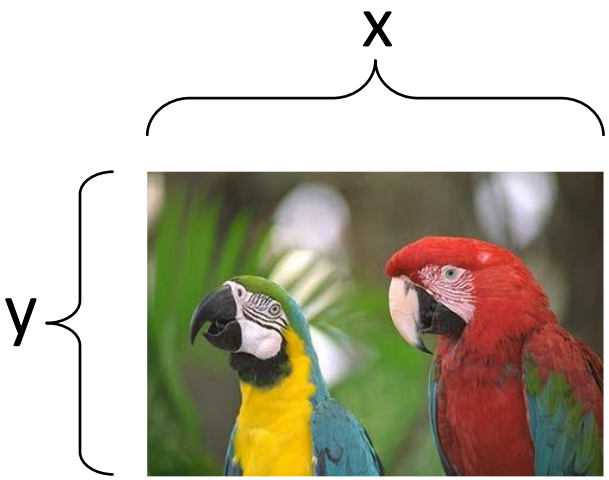
Z-buffer representation

Grafikpipeline



Darstellung:

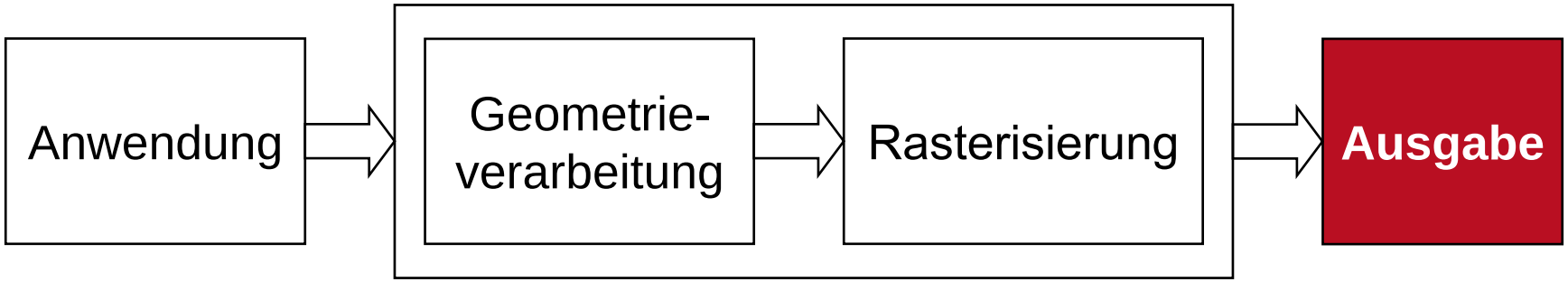
- Rasterbild (Bildspeicher)



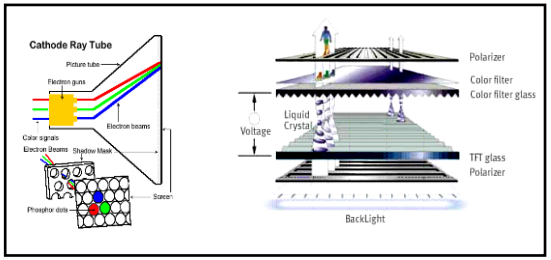
Grafikpipeline: Ausgabe

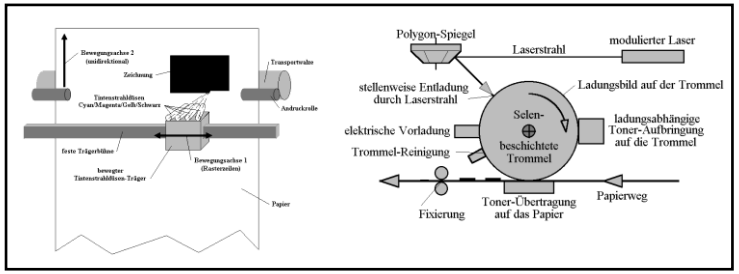
TECHNISCHE UNIVERSITÄT DARMSTADT

CPU GPU



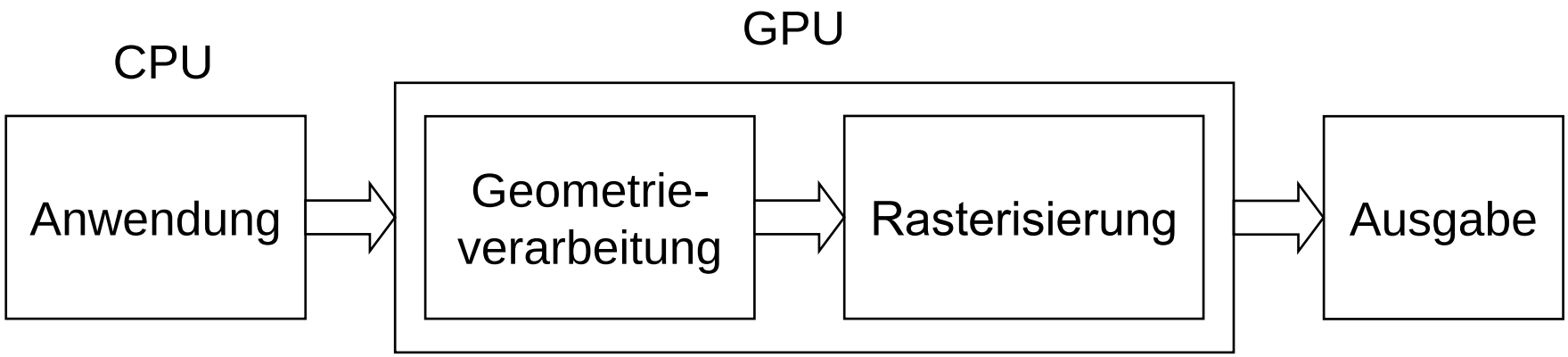
- Speichern des Bildes
- Display
- Hardcopy





MATHEMATICAL AND APPLIED VISUAL COMPUTING

3D-Grafikpipeline



- Eingabe Grafischer Daten
- Repräsentation von 3D Daten (Primitive, Transformationen, räumliche Datenstrukturen)

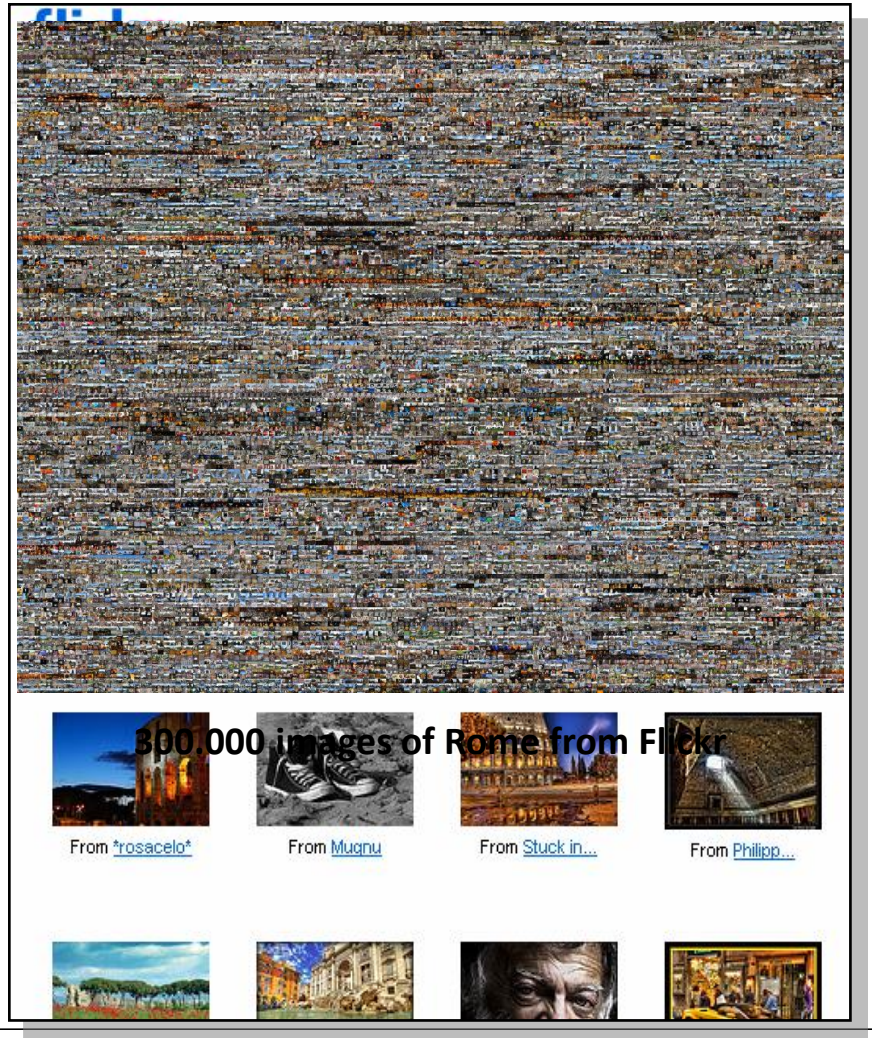
- Modell-Transformation
- Simulation der Beleuchtung
- Perspektivische Transformation
- Clipping
- Culling
- Projektion

- Rasterisierung der Primitive nacheinander:
  - Scan-Konvertierung
  - Verdeckungsrechnung
  - Farbwertinterpolation

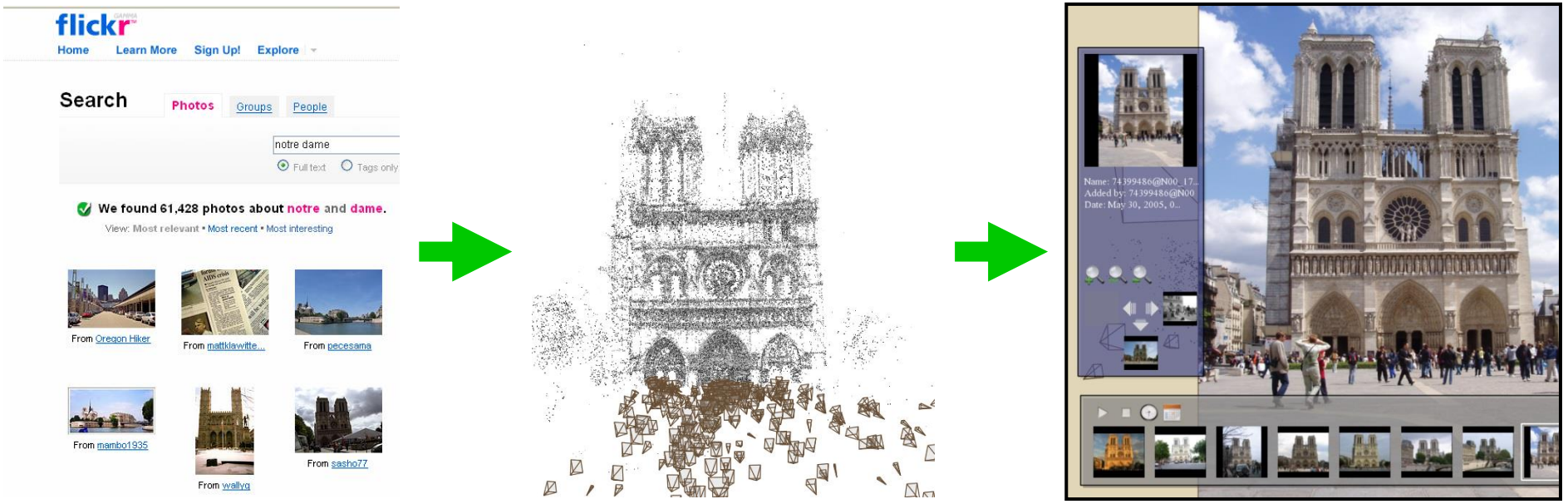
- Speichern des Bildes
- Display
- Hardcopy

Beispiel:
Community-basierter Ansatz: Bilddaten und das soziale Web

- Bildportal Flickr (www.flickr.com)
  - Mehr als 6 Milliarden Bilder bis zum August 2011 hochgeladen
  - 3.457.833 Suchergebnisse für „Rome“ (05.06.2012)
  - „Community Fotosammlungen“
- Diese Bilddaten werden verwendet für Inpainting, Kamerakalibrierung, bildbasiertes Rendering, …

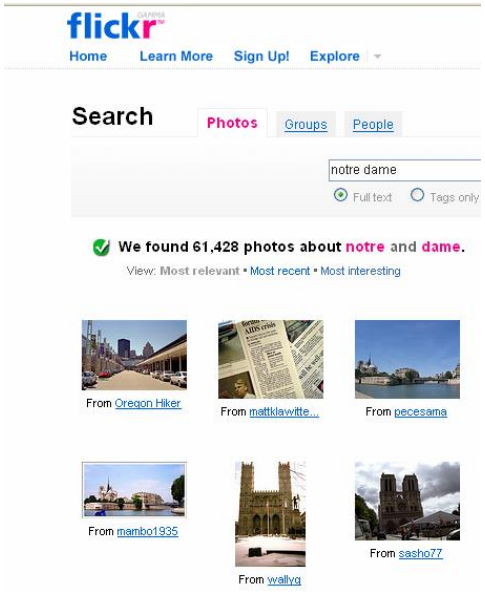


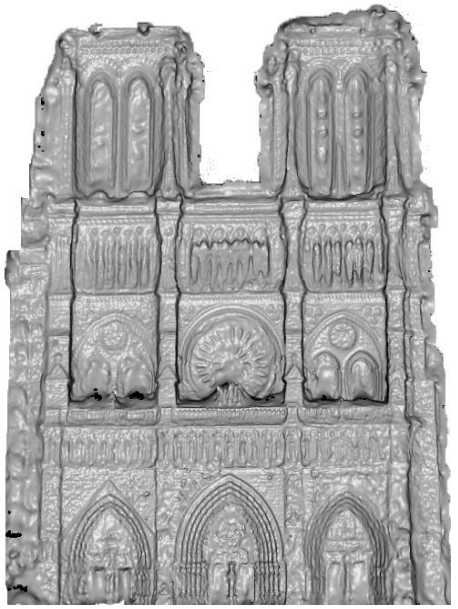
Fototourismus



- Wiederherstellung von Kameraposition und Szenenpunkten aus Flickr-Bildsammlungen durch Benutzung von Structure-From-Motion
- User Interface um durch Fotosammlungen zu navigieren
- Kerntechnologie von Microsoft Live Labs Photosynth (seit August 2008)
- Google StreetView!

Multi-View Stereo for Community Photo Collections





- Rekonstruktion von geometrischen Modellen mithilfe von Multi-View-Stereo-Techniken
- Beispiel: Front der Notre-Dame wurde aus 653 Bildern rekonstruiert, die von 313 Fotografen auf Flickr hochgeladen wurden: http://grail.cs.washington.edu/projects/mvscpc/
- Multi-View Environment: https://www.gcc.tu-darmstadt.de/home/proj/mve/

Weiterführende Vorlesung @ YouTube

TECHNISCHE UNIVERSITÄT DARMSTADT

Computer Graphics 2011, Lect. 8 – Perspective projection, clipping, culling, and Z-buffer (32:40)

Recordings from an introductory lecture about computer graphics given by Wolfgang Hürst, Utrecht University, The Netherlands, from April 2011 till June 2011.

https://www.youtube.com/watch?v=stCELsaDJRw

Part 2 (39:17):

https://www.youtube.com/watch?v=3IcBpKqgFBQ

TECHNISCHE UNIVERSITÄT DARMSTADT

Vielen Dank für die Aufmerksamkeit

(und gleich zu den Übungen!)