Visual Computing
University of Magdeburg

Grundlagen der Computergraphik (Graphik 1)

Holger Theisel

Daniel Stelter

Otto-von-Guericke-Universität

Magdeburg, SS 2024

Allgemein:

Prof. Holger Theisel
theisel@ovgu.de,
Büro: G29-217 , Tel: +49-391-6758773

Daniel Stelter
daniel.stelter@ovgu.de

Homepage:

http://vc.cs.ovgu.de/index.php?article_id=792&clang=1
https://elearning.ovgu.de/course/view.php?id=16588

Registrierung im e-learning Portal
- Übungen: Registerung e-learning Portal:
- Einschreibeschlüssel : raytracer

Im e-learning Portal:
- Aufzeichung der Vorlesungen (vom SS 2021)
- Folien der Vorlesungen
- Übungsaufgaben, Programmieraufgaben (Abgabe der Übungsblätter erfolgt nur elektronisch über das Elearning)

Einordnung:

Studiengänge

Studiengang	Abschluss	Empfohlenes Semester	Kategorie	ECTS	Modul
Computervisualistik (82152)	Bachelor	2 - 3	PF		
Digital Engineering (86158)	Master	1 - 3	WPF		
Informatik (82150)	Bachelor	4 - 6	WPF		
Ingenieurinformatik (82157)	Bachelor	4 - 6	WPF		
Medienbildung AVKK (82735)	Bachelor	4 - 5	WPF		14
Sport+Technik (86720)	Master	2 - 2	WPF		
Wirtschaftsinformatik (82159)	Bachelor	4 - 6	WPF		

Einordnung:

Pflicht Informatik Profilrichtung Computergames

Computer Games

Du willst mehr als nur spielen? Lernen, was sich im Inneren von Computerspielen verbirgt? Du willst wissen, welche Algorithmen hinter den tollen Graphiken stecken? Du willst selbst eigene Spiele entwickeln oder später in der Computerspieleindustrie arbeiten?

Mit der Profillinie „Computer Games" innerhalb des Bachelorstudiengangs Informatik lernst Du, wie Spiele entwickelt werden. Zusätzlich kannst Du Dich beim an der Uni tätigen Verein „Acagamics e.V." mit Gleichgesinnten austauschen und mehr über Industrie und Forschung im Bereich Computerspiele erfahren.

Ergänzend zu den Pflichtveranstaltungen im Informatikstudium belegst Du folgende vier Lehrveranstaltungen: Introduction to Computer Games, Introduction to Computer Graphics, Introduction to Simulation, Algorithmische Geometrie. Dazu wählst Du weitere vier Vorlesungen aus einem größeren Pool von passenden Veranstaltungen. Zum Studium gehörende Seminare, Projekte und Praktika kannst Du ebenfalls zum Thema Computerspiele auswählen.

Profilmodule

Die Übersicht über die angebotenen Profilmodule kann hier abgerufen werden:
› Modulübersicht Profil "ComputerGames". Zur besseren Übersicht gibt es auch einen entsprechenden
› Regelstudienplan

Verantwortliche

Für das Studienprofil ist verantwortlich:

- Prof. Dr. Holger Theisel

Allgemein:

Vorlesung Mo, 15:00 - 17:00 Uhr
Beginn 08.04.2024

Allgemein:

Übungen:
Gruppe 2 (Stelter) Mi, 15:00 -17:00 Uhr, G29-336
Gruppe 3 (Stelteri) Mi, 13:00 -15:00 Uhr, G29-K053
Beginn: 10.04.2024

- Besprechen der Übungsaufgaben

Abschluss:

(1) 2/3 der möglichen Punkte für Übungsaufgaben
(2) 2/3 der möglichen Punkte für Programmieraufgaben

(3) schriftliche Prüfung (Klausur) am Semesterende

(1) und (2) sind Voraussetzungen zur Zulassung für (3)

Keine Anwesenheitspflicht in Vorlesung oder Übung!
Schein: (3) mit mind. Note 4

Nicht-FIN Studiengänge: nur (1) und (3)

Übungsaufgaben:

- **Ausgabe:** jede Woche Mo ab 18 Uhr im eLearning
- **Abgabe der Lösungen:**
  - bis Mo 15 Uhr (vor der Vorlesung) im E-learing system

  Besprechung: nächste Übung nach der Abgabe

- **Ausgabe der ersten Serie:** 08.04.2024

Programmieraufgaben:

Raytracer in C++,

Ausgabe: alle 2 -3 Wochen mit Übungsaufgaben
Abgabe der Lösungen:

- Mo 15 Uhr (vor der Vorlesung) im E-learning system

Besprechung: nächste Übung nach der Abgabe

- Ausgabe der ersten Serie: 22.04.2024.

Plan der nächsten Wochen:

- Mo, 08.04.: 1. Vorlesung
- Mo, 08.04. Ausgabe Übungsblatt 1
- Mi, 10.04.: 1. Übung: Einführung C++

- Mo 15.04.: 2. Vorlesung, davor Abgabe Übungsblatt 1
- Mo 15.04.: Ausgabe Übungsblatt 2
- Mi 17.04.: 2. Übung: Besprechen Übungsblatt 1

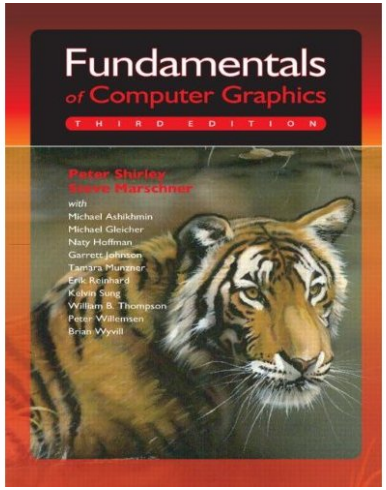
- Mo, 22.04.: 3. Vorlesung, davor Abgabe Übungsblatt 2
- Mo, 22.04.: Ausgabe Übungsblatt 3 mit Programmierblatt 1
- Mi 24.04. 3. Übung: Besprechen Übungsblatt 2

- Mo, 29.04.: 4. Vorlesung, davor Abgabe Übungsblatt 3
- Mo, 29.04.: Ausgabe Übungsblatt 4
- Mi 01.05.: keine Übung (1.Mai)

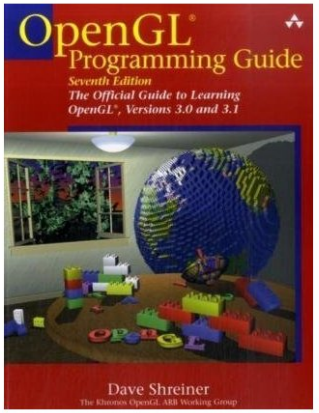
Für Übungsaufgaben und Programmieraufgaben

- Gruppen 1-3 Personen möglich
- nach erster gemeinsamer Einreichung können keine neuen Mitglieder in eine Gruppe kommen
- erlaubt: Gruppenverkleinerungen und Gruppensplits
- bitte immer alle Gruppenmitglieder und Matrikelnummer auf der Abgabe vermerken!

Literatur



Peter Shirley
Fundamentals of Computer Graphics
AK Peters, 3. Auflage



Dave Shreiner
OpenGL Programming Guide
Morgan Kaufmann, 7. Auflage

Beide Bücher sind mehrfach in der Bibliothek als Präsenz – und Ausleihexemplar vorhanden

Literatur

- Bücher
  - J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes: Computer Graphics - Principles and Practice (second Edition). Addison-Wesley Publishing Company, Inc., 1996
  - J. Encarnacao, W. Straßer, R. Klein: Gerätetechnik, Programmierung und Anwendung graphischer Systeme Teil I und II. Oldenbourg, München, Wien, 1996, 1997
  - D. Salomon: Computer Graphics Geometric Modeling, Springer, 1999
  - A. Watt: 3D Computer Graphics. Addison-Wesley Publishing Company, Inc., 2000
- Zeitschriften
  - Computer Graphics Forum
  - IEEE CG & Applications
  - ACM Transactions on Graphics
  - …

Acknowledgements

Diese Vorlesungsskripte basieren teilweise auf Skripten von

- Prof. Heidrun Schumann (Universität Rostock)
- Prof. Marcus Magnor (Universität Braunschweig)
- Prof. Thorsten Grosch (Universität Clausthal-Zellerfeld)

und weiteren KollegInnen.

Dankeschön!

1. Gliederung

- 1. Einführung
- 2. Modellierung und Akquisition graphischer Daten
- 3. Rendering
- 4. Raytracing
- 5. Aktuelle Themen der Computergraphik im Überblick

Visual Computing
University of Magdeburg

1. Einführung

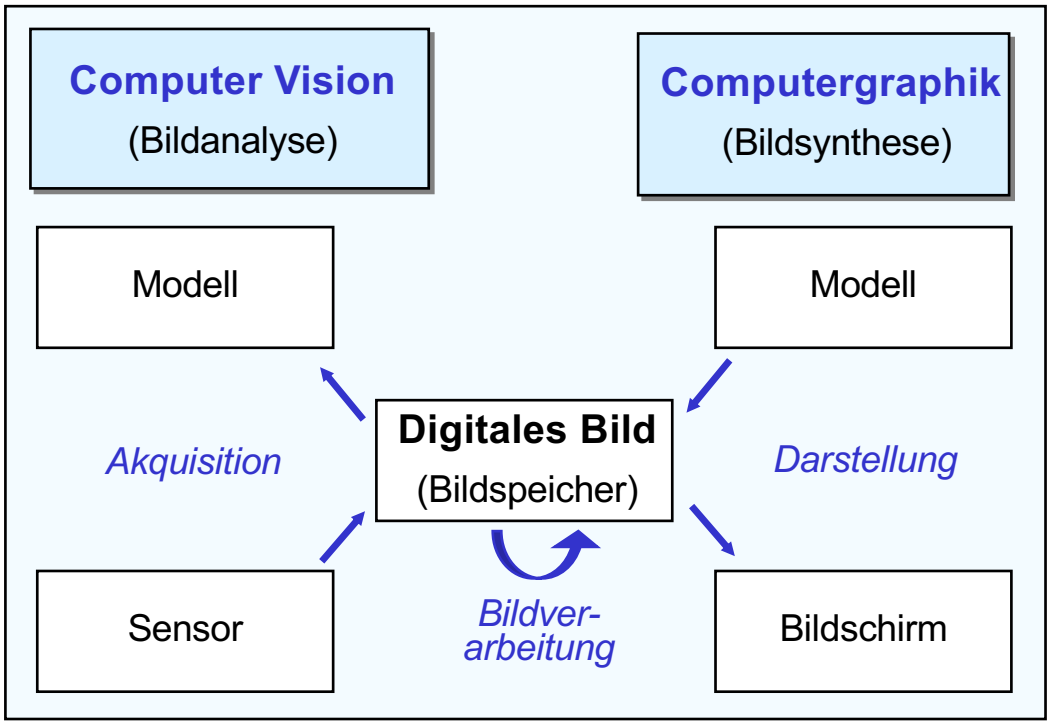
Visual Computing
University of Magdeburg

1. Einführung
1.1 Einordnung und Begriffsbestimmung

Die graphische Datenverarbeitung ist in der ISO-Norm wie folgt definiert (ISO DIS 2382/13, 1982):

> *"Methods and techniques for converting data*
> ***to and from***
> *graphics displays via computer"*

Mit "to" und "from" sind zwei Wege aufgezeichnet, einmal von der Beschreibung (Modell) zum Bild und einmal vom Bild zur Beschreibung.



Dualität von Computergraphik und Computervision

1. Einführung
1.1 Einordnung und Begriffsbestimmung

- Die Computergraphik behandelt die Erzeugung und

Manipulation künstlicher Bilder mit dem Computer, die in

Form von Bildbeschreibungen (Modell) vorliegen.

- Die klassischen Hauptrichtungen der Computergraphik sind:
  - Modellierung/Akquisition

    Generierung der Bildbeschreibung in maschinell-interpretierbarer Form als Voraussetzung der Computergraphik
  - Bearbeitung (Processing)

    Hierarchien, Segmentierungen, Komprimierungen, Umwandlungen
  - Rendering (Darstellung)

    Erzeugung des Bildes aus der Bildbeschreibung
  - Interaktion

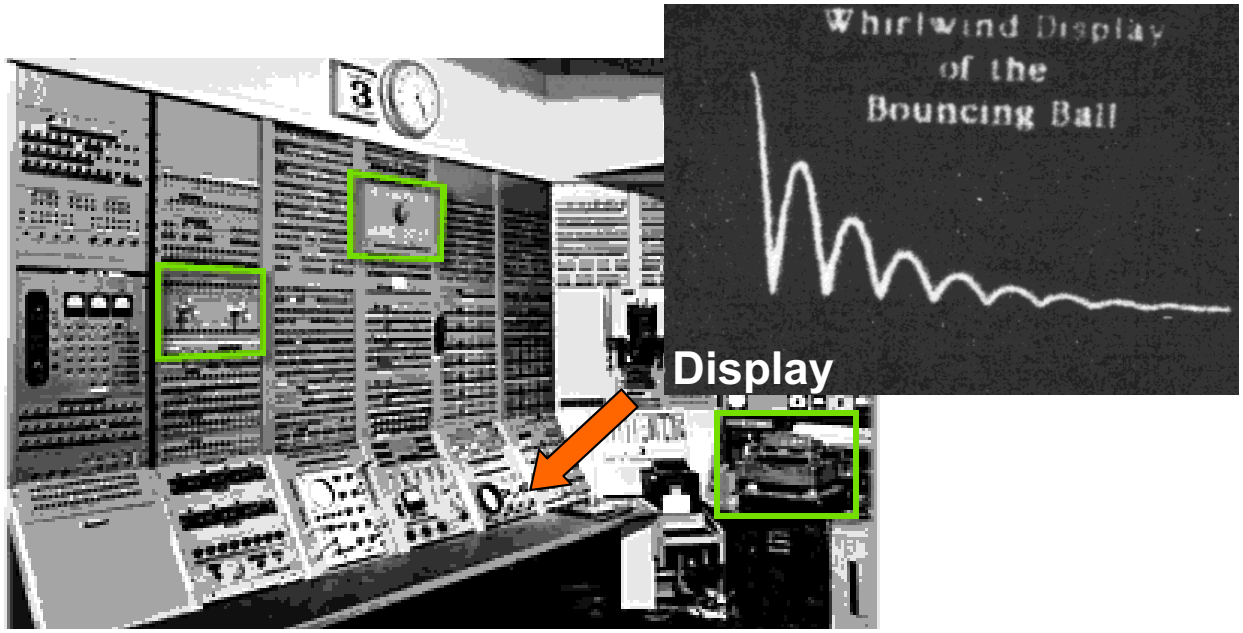
    Manipulation des Bildes bzw. der Bildbeschreibung

Visual Computing
University of Magdeburg

1. Einführung
1.2 Geschichte der Computergraphik

Geschichte der Computergraphik

1949 Erste Computergraphik auf dem Whirlwind Computer des MIT (**bouncing Ball Program** von C. Adams);



1. Einführung
1.2 Geschichte der Computergraphik

1952 Einsatz der Computergraphik zur Kennzeichnung von Flugobjekten auf Radarbildschirmen

(SAGE Computer mit 82 Graphikkonsolen zur Luftüberwachung, erster Einsatz eines Lichtgriffels);

Sage Consolen

1. Einführung
1.2 Geschichte der Computergraphik

- **1962** Erste **3D Computergraphiken** von L.G. Roberts auf dem TX2 des MIT;

TX2 Computer

1. Einführung
1.2 Geschichte der Computergraphik

- **1963** **Sketchpad** - erstes interaktives Computergraphiksystem von Sutherland mit
  - Bildkomposition aus graphischen Standardelementen,
  - Interaktion mit Tastatur und Lichtgriffel zur Arbeit mit Menüs,
  - entsprechenden Datenstrukturen zur Verwaltung graphischer Daten;

Ivan Sutherland an der Konsole des TX-2 Computers

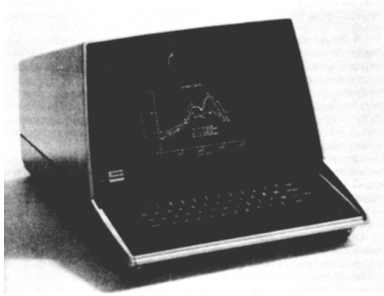
1. Einführung
1.2 Geschichte der Computergraphik

Mitte der 60er-Jahre: Beginn einer Vielzahl von CG-Forschungs-projekten (am MIT, General Motors, Bell Telephone Lab., Lockheed Aircraft usw.)

1965: Erstes **kommerzielles Vektor-Display** von IBM

(Preis ca. 100.000 US$);

1967: Erster Bildspeicher-Display von Tektronix (bietet einem breiten Interessentenkreis Zugang zur Computergraphik, Preis ~10.000 USD);



1. Einführung
1.2 Geschichte der Computergraphik

Anfang der 70er Jahre: erste kommerzielle CAD/CAM-Systeme kommen auf den Markt.

1971: **Raster-Scan-Prinzip** von M. Noll (Bell Lab.) vorgeschlagen;

1972: Erster Flugsimulator (General Electronics);

Erster **Heimcomputer** als Bausatz unter dem Namen „Altair 8800“ auf dem Markt

1. Einführung
1.2 Geschichte der Computergraphik

Anfang der 70er Jahre: erste kommerzielle CAD/CAM-Systeme kommen auf den Markt.

- **1973:** Erste Konferenz der **SIGGRAPH** (**S**pecial **I**nterest **G**roup on Computer **Graph**ics) der ACM (Association of Computing Machinery), die sich ausschließlich mit Computer Graphik beschäftigt (ca. 1200 Teilnehmer);

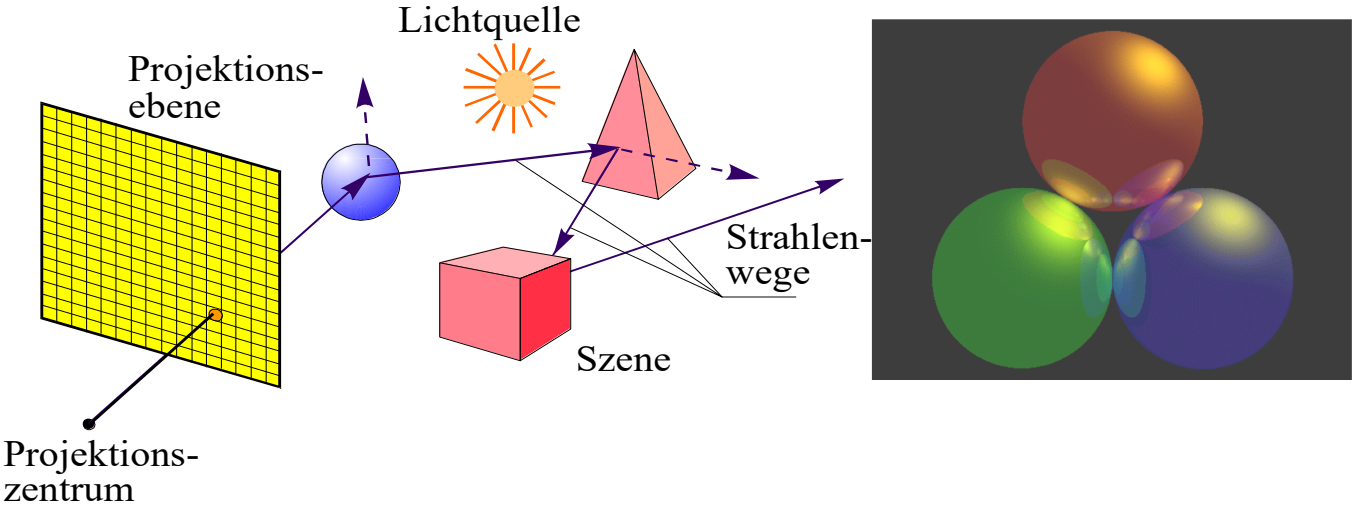
1. Einführung
1.2 Geschichte der Computergraphik

- Ab Mitte der 70er Jahre:
  - Graphische Unterprogrammpakete (PLOT10, CAL-Comp),
  - Graphische Programmiersprachen (DIGRA 73),
  - Erste **kommerzielle Raster-Displays** (max. Auflösung 512 x 512 Pixel, 8 Bit pro Pixel, Preis ~100.000DM);
  - Erste Verfahren zur schattierten Objektdarstellung:
    - Beleuchtungsverfahren (Phong 1975, Blinn 1977),
    - Schattierungsverfahren (Gouraud 1971, Phong 1975),
    - Texturierung (Catmull 1974),
    - Schattenwurf (Crow 1977);

*M. Newell (Univ. of Utah) modelliert 1975 den **Utah tea pot** – eine Ikone der Computer Graphik.*

1. Einführung
1.2 Geschichte der Computergraphik

- **1977** Erste Vorschläge zur Standardisierung von Graphiksoftware - **CORE**;
- **1979** Erstmalige Darstellung von spiegelnder Reflexion und Transparenz mit Hilfe des **Raytracing** (Kay);



1. Einführung
1.2 Geschichte der Computergraphik

- **1980:** Vorführung des Films Vol Libre (von L. Carpenter, Boeing) auf der SIGGRAPH'80 (in dem Film wird der Flug durch eine **fraktale Landschaft** gezeigt);

Carpenter's Kunst-Gebirge wurde bei der Prämierung von der SIGGRAPH-Jury mit der Begründung ausgeschlossen, weil

„... es nicht wie eine Computer Graphik aussieht !"

1. Einführung
1.2 Geschichte der Computergraphik

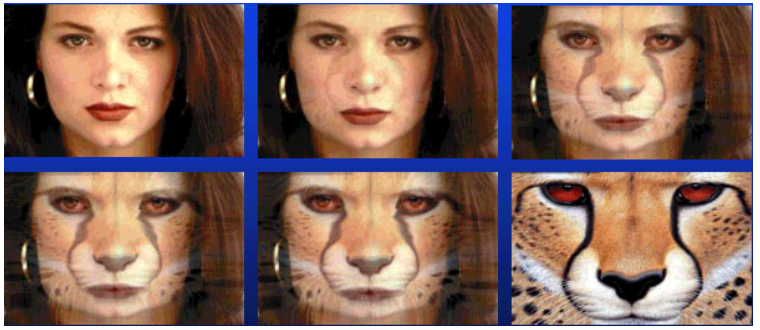
1980: ~30 Min. Computeranimationen im Film **Tron,** (Film floppt, große Hollywood Filmstudios reagieren mit Zurückhaltung gegenüber Computer Graphik);

1981: Erstes „Rendering"-System **REYES** („Rendering everything you ever saw", von L. Carpenter für LucasFilm – wird später zu **Renderman** weiterentwickelt);

Beginn der Entwicklung des **Volume-Rendering**;

1982: Erste Filmsequenz, in der sich Frau in Luchs verwandelt (T.Brigham, SIGGRAPH´82);

Diese Technik wird später **Morphing** genannt -
(wurde bis 1987 nicht weiter beachtet, bis LucasFilm sie in dem Film „Willow" einsetzt);



1. Einführung
1.2 Geschichte der Computergraphik

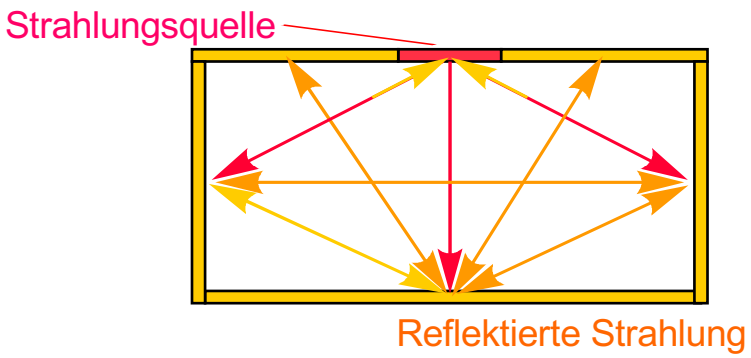
- **1982:** Gründung **Silicon Graphics Inc.** (**SGI)**, J. Clark, (Entwicklung von Hochleistungsrechnern für graphische Anwendungen);

- **1983:** J. Lanier (Atari Research Center) entwickelt **Datenhandschuh;**

1. Einführung
1.2 Geschichte der Computergraphik

- **1984:** Globale Beleuchtungs-Simulation
  mit **Radiosity** (Goral u.a., Nishita);

  Gründung Wavefront Technologies
  für **Animations-Software**;



Radiosity-Bild

1. Einführung
1.2 Geschichte der Computergraphik

- **1985:** Gründung der Fa. VLP durch J. Lanier, um die ersten kommerziellen **Virtual Reality** Produkte zu entwickeln;

  GKS - erster graphischer ISO-Standard

  (1988 GKS-3D);

1. Einführung
1.2 Geschichte der Computergraphik

1986: Gründung der Fa. Pixar durch Ed Catmull und A.R. Smith nach Abspaltung von Lucas Film;

- Pixars **RenderMan** wird Industrie-Standard;
- **1988:** Film **The Abyss**, James Cameron

(ILM stellt dabei die Szene mit der Wasser-Kreatur her, die die Gesichter der Mannschaft imitiert);

aus J.D. Foley, et al
Computer Graphics – Principles and Practice

1. Einführung
1.2 Geschichte der Computergraphik

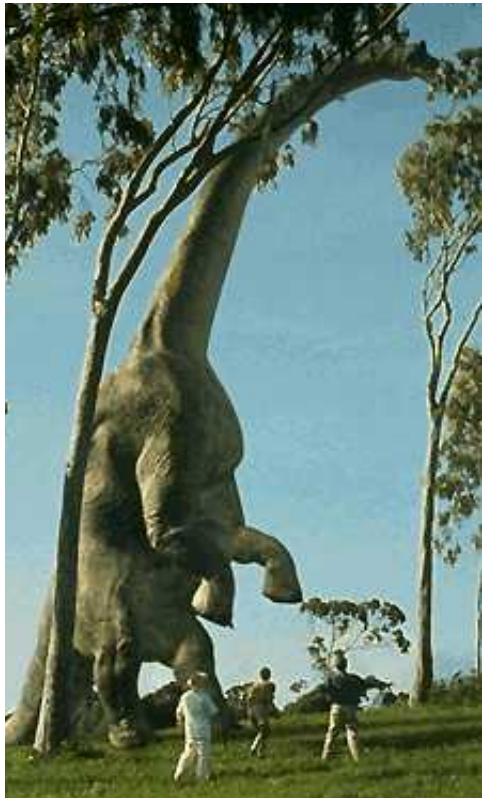
1989: Einführung von **Motion Capture**, Nutzung mechanischer Eingabegeräte für Computeranimation;

1992: Neue Maßstäbe bei computer-generierten Spezialeffekten, (Animationen des „T1000"-Roboters in J. Cameron's Film **Terminator 2**);

1. Einführung
1.2 Geschichte der Computergraphik

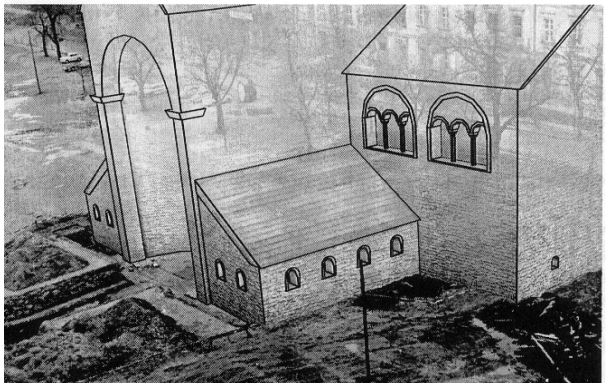
1993: Steven Spielberg's Film
Jurassic Park

(anstelle der ursprünglich geplanten Puppenanimationen werden Computer-animationen für die Dinosaurierszenen eingesetzt);



1. Einführung
1.2 Geschichte der Computergraphik

→ Anfang- Mitte der 90er Jahre:
- Verbindung moderner Kommunikationstechnologien mit Graphik:
  - ⇨ Multimedia in verteilten Umgebungen,
  - ⇨ CSCW Computer supported cooperative work,
  - ⇨ Graphik im Internet,
  - ⇨ Standards zur Bild- und Bewegt-bildübertragung...,
- Daten- und Informations-Visualisierung,
- Methoden des Non-Photorealistic Rendering;



1. Einführung
1.2 Geschichte der Computergraphik

→ **1995**: **Toy Story** kommt in die Kinos; erster vollständig computer-animierte Film von Pixar;

→ Rendering: 800 000 Std. Berechnungs-zeit für 70 min. Film auf 177 Sun Sparc 20

→ Der Film ist ein großer Kassenerfolg, wird für drei Oskars nominiert, erhält jedoch keinen davon ...

Geschichte der Computergraphik

1998: High Dynamic Range (**HDR**), Paul Debevec
8 Bit → 32 Bit float pro Pixel
Digitale Kamera nimmt echte Helligkeiten auf

HDR Light Probe

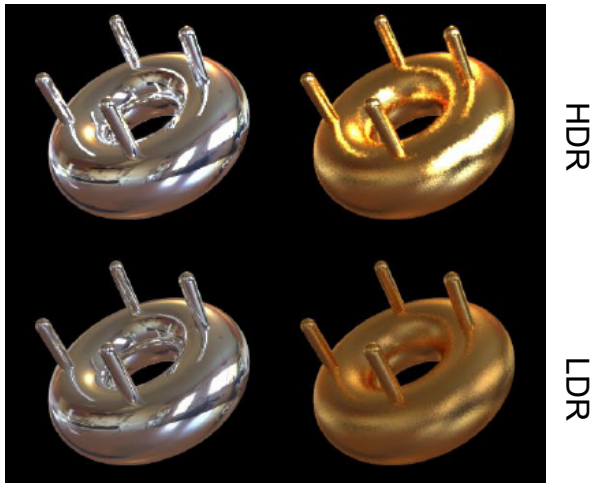


Image-based Lighting

Geschichte der Computergraphik

Grafikkarten

1999 - heute: Graphics Processing Unit (GPU)

SGI Grafik-Workstation wird immer mehr von PC mit Grafikkarte abgelöst

Starke Weiterentwicklung durch Spieleindustrie

GPU wird zum schnellen Coprozessor, bisherige Offline-Techniken werden Echtzeitfähig

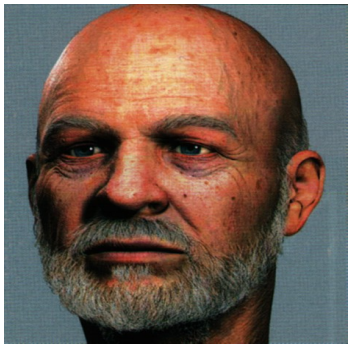
1. Einführung
1.2 Geschichte der Computergraphik

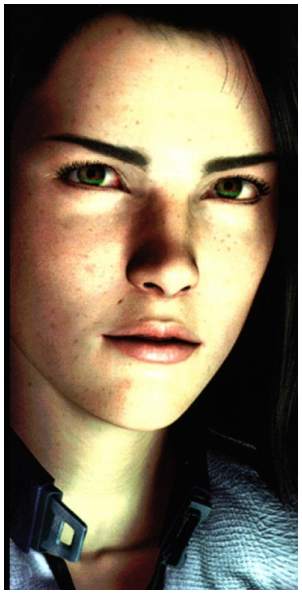
2001: sehr gute Gesichtsanimationen von virtuellen Charakteren im Film **Shrek** ;

Simulation von Haaren: **Monster AG**

1. Einführung
1.2 Geschichte der Computergraphik

2001: Final Fantasy, erster komplett computer-animierter Spielfilm mit realistischen virtuellen Charakteren;
die Produktion des Films dauerte ~4 Jahre, wobei etwa 170 Computeranimatoren tätig waren





aus: digital production, Heft 3/01

Geschichte der Computergraphik

Sehr große, virtuelle Welten…

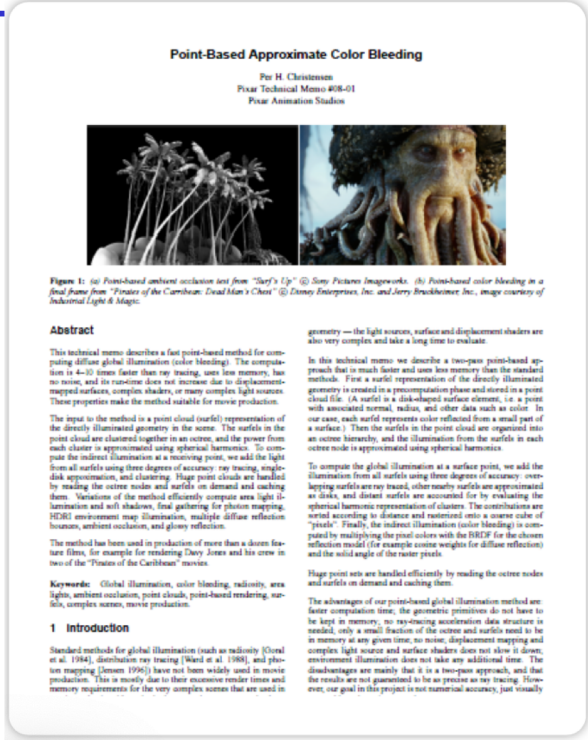
2003: Herr der Ringe

2009: Avatar

2019… ?

“Point-based Global Illumination”

Pixar, Industrial Light & Magic



Point-Based Approximate Color Bleeding

Per H. Christensen
Pixar Technical Memo #08-01
Pixar Animation Studios

Abstract

This technical memo describes a fast point-based method for computing diffuse global illumination (color bleeding). The computation is 4–10 times faster than ray tracing, uses less memory, has no noise, and its run-time does not increase due to displacement-mapped surfaces, complex shaders, or many complex light sources. These properties make the method suitable for movie production.

Keywords: Global illumination, color blending, radiosity, area lights, ambient occlusion, point clouds, point-based rendering, surfels, complex scenes, movie production.

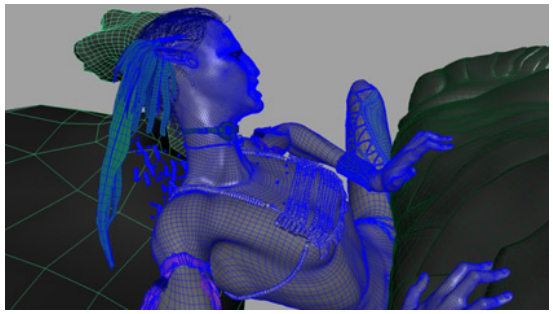
1 Introduction

[Fluch der Karibik 2, 2006]

“PantaRay – Visibility Precomputing”

Nvidia, Weta Digital

[Avatar, 2009]



PantaRay: Fast Ray-traced Occlusion Caching of Massive Scenes

Jacopo Pantaleoni* NVIDIA Research · Luca Fascione† Weta Digital · Martin Hill† Weta Digital · Timo Aila* NVIDIA Research

Figure 1: *The geometric complexity of scenes rendered in the movie Avatar often exceeds a billion polygons and varies widely: distant rocks and vegetation are tessellated to a level of meters and centimeters, while the faces of even distant characters are modeled to over 40,000 polygons from forehead to chin. The spatial resolution of occlusion caches precomputed by our system also spans several orders of magnitude.*

Abstract

We describe the architecture of a novel system for precomputing sparse directional occlusion caches. These caches are used for accelerating a fast cinematic lighting pipeline that works in the spherical harmonics domain. The system was used as a primary lighting technology in the movie Avatar, and is able to efficiently handle massive scenes of unprecedented complexity through the use of a flexible, stream-based geometry processing architecture, a novel out-of-core algorithm for creating efficient ray tracing acceleration structures, and a novel out-of-core GPU ray tracing algorithm for the computation of directional occlusion and spherical integrals at arbitrary points.

1 Introduction

The movie Avatar featured unprecedented geometric complexity (Figure 1), with production shots containing anywhere from ten million to over one billion polygons.

To make the rendering of such complex scenes manageable while satisfying the need to provide fast lighting iterations for lighting artists and the director, modern relighting methods based on spherical harmonics (SH) [Ramamoorthi and Hanrahan 2001] and image-based lighting [Debevec 1998] were used. These methods can speed up the lighting iterations significantly, but unfortunately require an extremely compute and resource intensive precomputation of directional occlusion information. Directional occlusion encodes the visibility term used for lighting modulation as a function of direction, and is typically computed using ray tracing.

We describe PantaRay[1], a system designed to make this precomputation practical by leveraging the development of modern ray tracing algorithms for massively parallel GPU architectures [Aila and Laine 2009] and combining them with new out-of-core and level of detail rendering techniques.

The PantaRay engine is an out-of-core, massively parallel ray tracer designed to handle scenes that are roughly an order of magnitude bigger than available system memory, and that require baking spherical harmonics-encoded directional occlusion (SH occlusion) and indirect lighting information for billions of points with highly varying spatial density.

Our key contributions are the introduction of a flexible, stream-based geometry processing architecture, a novel out-of-core algorithm for constructing efficient ray tracing acceleration structures and a novel out-of-core GPU ray tracing algorithm for the computation of directional occlusion and spherical integrals. These are

“Volumetric Lighting”

Disney Research

[Rapunzel, 2010]

A Programmable System for Artistic Volumetric Lighting

Derek Nowrouzezahrai[1] Jared Johnson[2] Andrew Selle[2] Dylan Lacewell[2] Michael Kaschalk[2] Wojciech Jarosz[1]

[1]Disney Research Zürich [2]Walt Disney Animation Studios

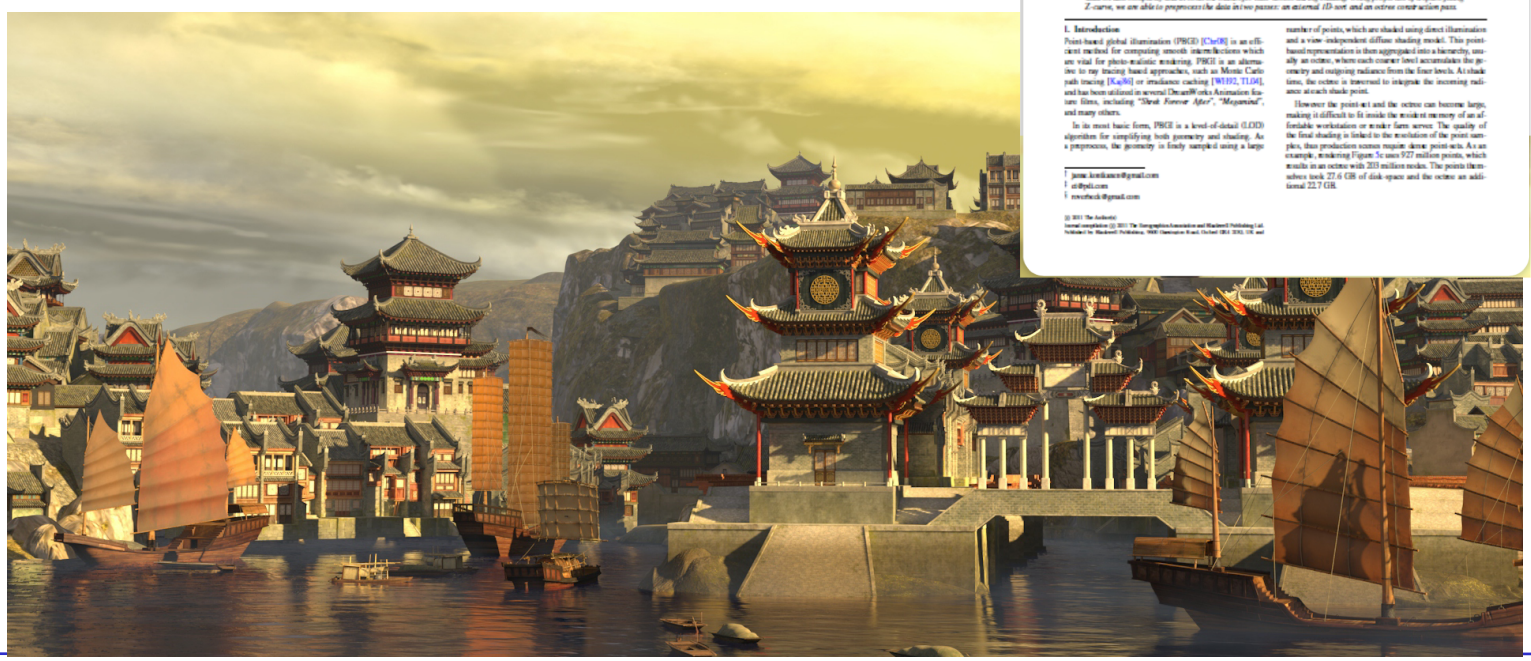
Abstract

1 Introduction

“Out-of-Core Global Illumination”

DreamWorks Animation

[Kung Fu Panda 2, 2011]



Coherent Out-of-Core Point-Based Global Illumination

Jacob Koskanen[1] and Eric Tabellion[2] and Ryan S. Overbeck[3]

DreamWorks Animation

“Artistic Simulation of Curly Hair”

Disney, Pixar

[Merida, 2012]

Artistic Simulation of Curly Hair

Hayley Iben, Mark Meyer, Lena Petrovic,
Olivier Soares, John Anderson and Andrew Witkin
Pixar Animation Studies
Pixar Technical Memo #12-03b

Figure 1: *Example of stylized curly hair simulated with our method. © Disney/Pixar*

Abstract

Artistic simulation of hair presents many challenges - ranging from supporting artistic control to dealing with extreme motions of characters. Additionally, in a production environment, the simulation needs to be fast and results need to be usable "out of the box" (without extensive parameter modifications) in order to produce content efficiently. These challenges are only increased when simulating curly, stylized hair.

We present a method for stably simulating stylized curly hair that addresses these artistic needs and performance demands. To satisfy the artistic requirement of maintaining the curl's helical shape during motion, we propose a hair model based upon an extensible elastic rod. We introduce a method for stably computing a frame along the hair curve, essential for stable simulation of curly hair. Our hair model uses a spring for controlling the bending of the curl and another for maintaining the helical shape during extension. We also address performance concerns often associated with handling hair-hair contact interactions by efficiently parallelizing the simulation. To do so, we present a technique for pruning both hair-hair contact pairs and hair particles.

Our method has been used on two full length feature films and has been shown to be robust and stable over a wide range of animated motion as well as on a variety of hair styles, from straight to wavy to curly. It has proven invaluable in providing controllable, stable and efficient simulation allowing our artists to achieve their desired performance goals when facing strict scheduling demands.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: Hair simulation, Mass-spring models

1 Introduction

Robustly simulating stylized hair in a production environment is a challenging problem. Hair styles can range from straight to curly and often consist of an enormous number of hair-hair interactions leading to performance and memory concerns. In an environment driven by artistic expression, such as feature films, the shape and motion of the hairs resulting from the simulator are critical, as is the overall simulation time.

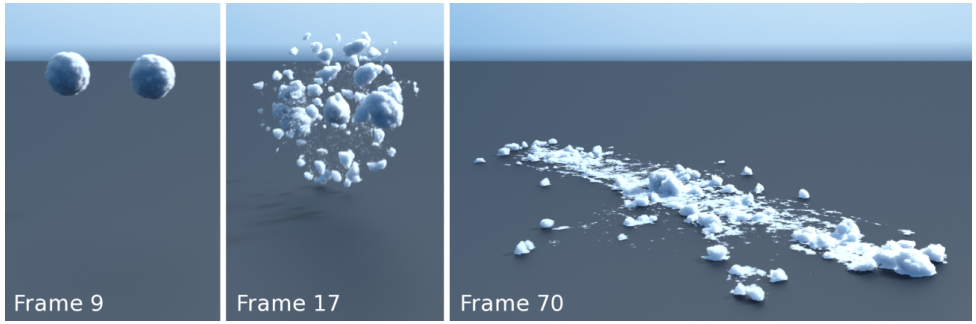
We present a hair model designed for creating specific visual looks of curly hair that are often non-physical. For example, our artists want to preserve the helical shape of the hair, regardless of the extensive forces caused by extreme motion only possible in animation. With a physical model for hair, such as inextensible elastic rods [Bergou et al. 2008], the helical shape would naturally straighten under such motion unless the material properties are extended to maintain the shape, making the hair wire-like. This natural straightening is an example of a physical motion undesired by artists. In addition to behavioral requirements, our hair model must be robust and stable, able to handle fairly arbitrary hair shapes created by artists as depicted in Figure 1. Given the large range of motion present in an animated feature film, our simulator must be able to give dependable results without per-shot parameter tuning while preserving the overall look of the hair.

Given these requirements, we chose to represent hair as a mass-spring system defined by a piecewise linear curve that captures the deformation of the hair over time. Using a point representation for hair has several advantages. It gives artists an intuitive interface to define and modify finely detailed, complex geometry, as well as providing straightforward simulation targeting and an direct and easy integration of external forces. However, mass-spring systems have been known to suffer from several well-documented

“Simulation of Snow”

Disney, Pixar

[Eiskönigin, 2013]



A material point method for snow simulation

Alexey Stomakhin[1] Craig Schroeder[1] Lawrence Chai[1] Joseph Teran[1,2] Andrew Selle[2]

[1]University of California Los Angeles [2]Walt Disney Animation Studios

Abstract

Snow is a challenging natural phenomenon to visually simulate. While the graphics community has previously considered accumulation and rendering of snow, animation of snow dynamics has not been fully addressed. Additionally, existing techniques for solids and fluids have difficulty producing convincing snow results. Specifically, wet or dense snow that has both solid- and fluid-like properties is difficult to handle. Consequently, this paper presents a novel snow simulation method utilizing a user-controllable elasto-plastic constitutive model integrated with a hybrid Eulerian/Lagrangian Material Point Method. The method is continuum based and its hybrid nature allows us to use a regular Cartesian grid to automate treatment of self-collision and fracture. It also naturally allows us to derive a grid-based semi-implicit integration scheme that has conditioning independent of the number of Lagrangian particles. We demonstrate the power of our method with a variety of snow phenomena including complex character interactions.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

Keywords: material point, snow simulation, physically-based modeling

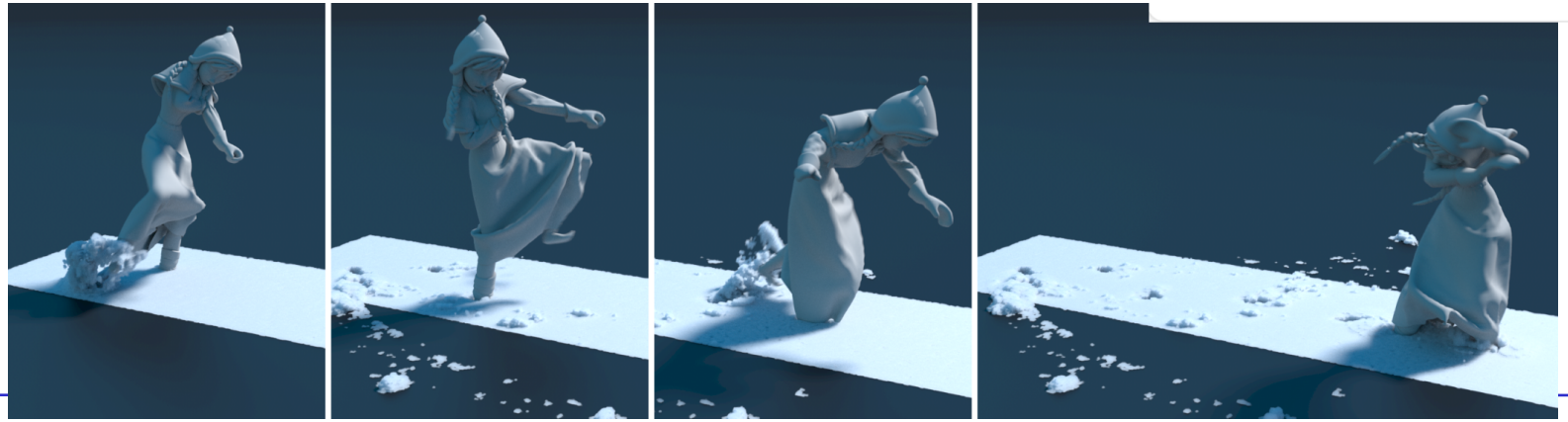
Links: DL PDF Web

1 Introduction

Snow dynamics are amazingly beautiful yet varied. Whether it is *powder snow* flattering in a skier's wake, foot steps shattering an icy *snow crust* or even *packing snow* rolled into balls to make a snowman, it is snow's rich repertoire that makes it simultaneously compelling for storytelling and infuriatingly difficult to model on a computer. Artists typically use simpler techniques combined in various ways to achieve snow effects [Kim and Flores 2008; Coony et al. 2010; Klohs et al. 2012], but these approaches are often tweaked for one type of snow. This suggests the need for a specialized solver that handles difficult snow behaviors in a single solver.

Specialized solvers for specific phenomena are frequently used in graphics and computational physics because achieving maximum resolution (and thus visual quality) requires efficiency. While a fluid simulator can produce solid-like elastic effects (and vice versa), it is not the most optimal strategy. When solids and fluids are needed simultaneously, researchers have developed two-way coupled systems to get good accuracy and performance for both phenomena. Unfortunately, snow has continuously varying phase effects, sometimes behaving as a rigid/deforming solid and sometimes behaving as a fluid. Thus, instead of discrete coupling we must simultaneously handle a continuum of material properties efficiently in the same domain, even though such a solver may not be most efficient for a single discrete phenomenon.

We present two main contributions that achieve these aims. First, we develop a semi-implicit Material Point Method (MPM) [Sulsky et al. 1995] specifically designed to efficiently treat the wide range of material stiffnesses, collisions and topological changes arising in complex snow scenes. To our knowledge, this is the first time MPM has been used in graphics. MPM methods combine Lagrangian material particles (points) with Eulerian Cartesian grids. Notably, there is no inherent need for Lagrangian mesh connectivity. Many researchers in graphics have experimented with hybrid grid and particle methods. For example, [Zhu and Yang 2010] simulate sand as a fluid using a PIC/FLIP incompressible fluid technique. In fact, MPMs were designed as a generalization of the PIC/FLIP solvers to computational solids. As with PIC/FLIP, MPMs implicitly handle self-collision and fracture with the use of the background Eulerian grid. This is essential given the many topological changes exhibited by practical snow dynamics. Our second contribution is a novel snow constitutive model designed for intuitive user control of practical snow behavior. This is also designed to achieve our goal of



Visual Computing
University of Magdeburg

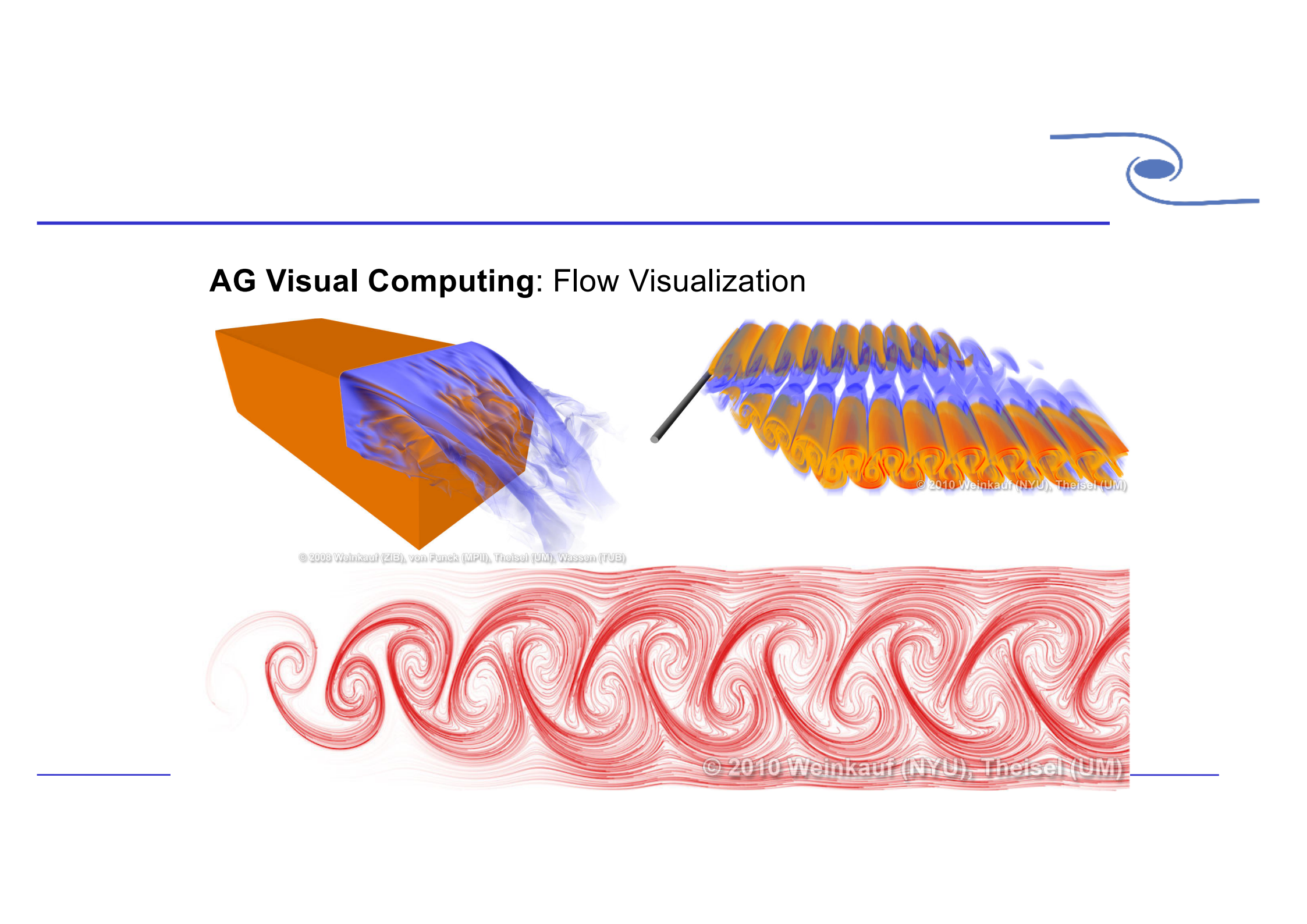
Computergraphik in Magdeburg

AG Visual Computing: Prof. Holger Theisel
AG Visualisierung: Prof. Bernhard Preim
AG Computer-Assisted Surgery: Jun.-Prof. Christian Hansen

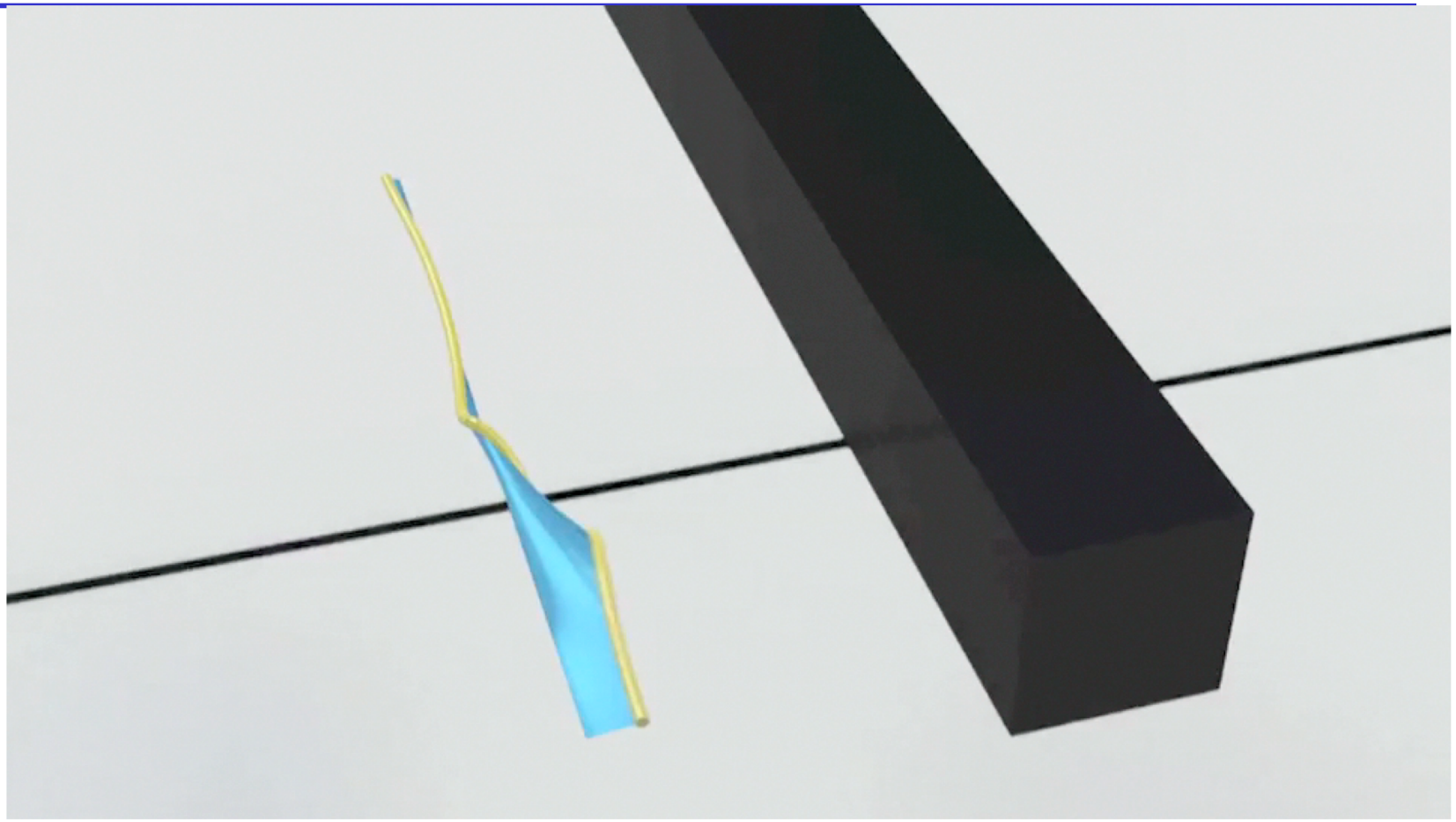
Computergraphik in Magdeburg

AG Visual Computing
Prof. Holger Theisel

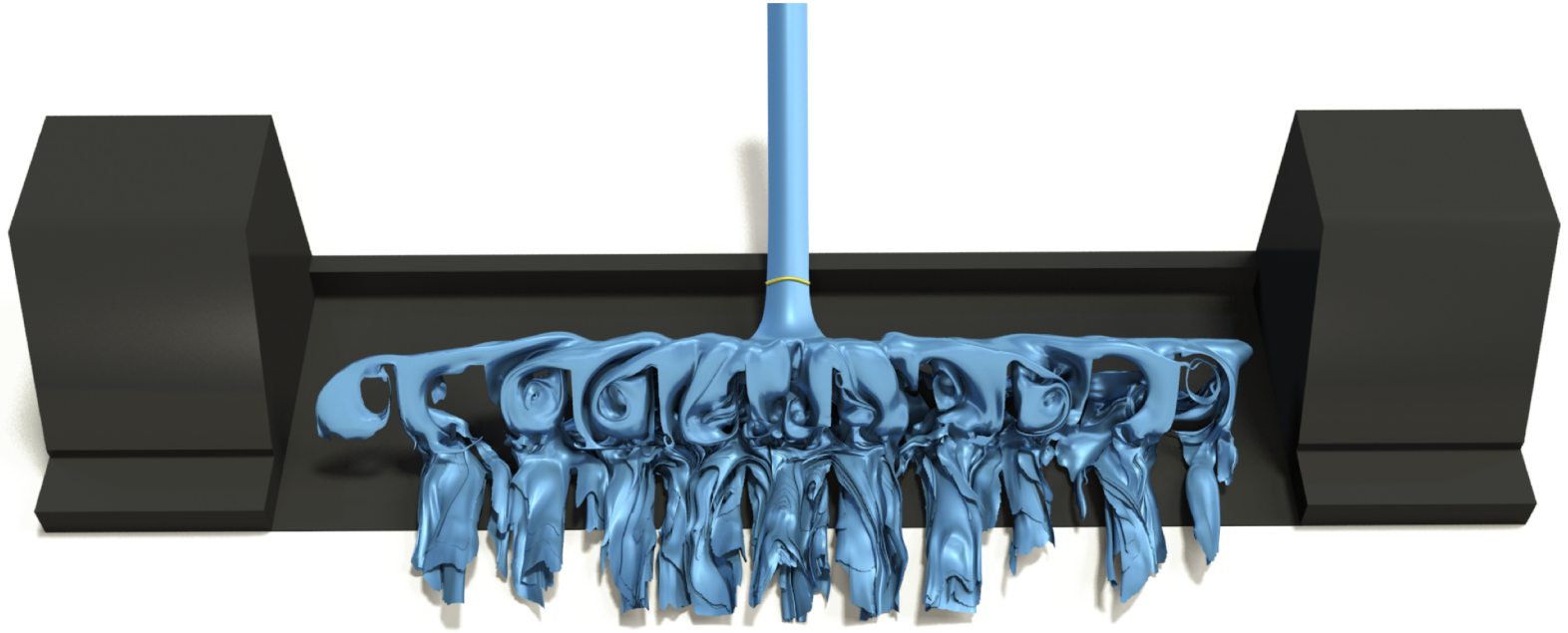
- Visualisierung
  - speziell Strömungsdaten
- Information Visualization
- Visual Analytics
- Geometric Modelling, geometric Design

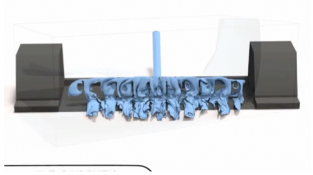

AG Visual Computing: Flow Visualization
© 2008 Weinkauf (ZIB), von Funck (MPII), Theisel (UM), Wassen (TUB)
© 2010 Weinkauf (NYU), Theisel (UM)
© 2010 Weinkauf (NYU), Theisel (UM)

Real Data – squared cylinder



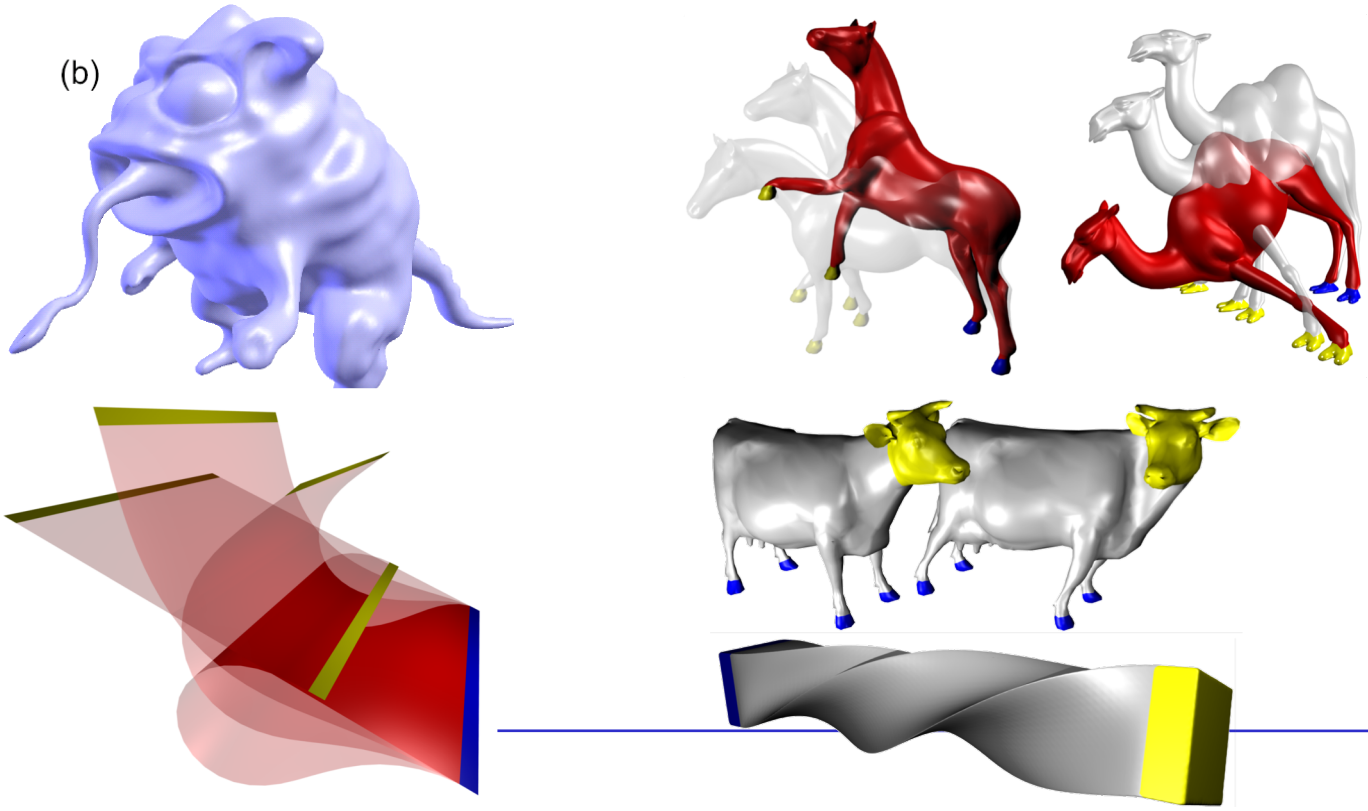
Results: Air Conditioner Outlet Box



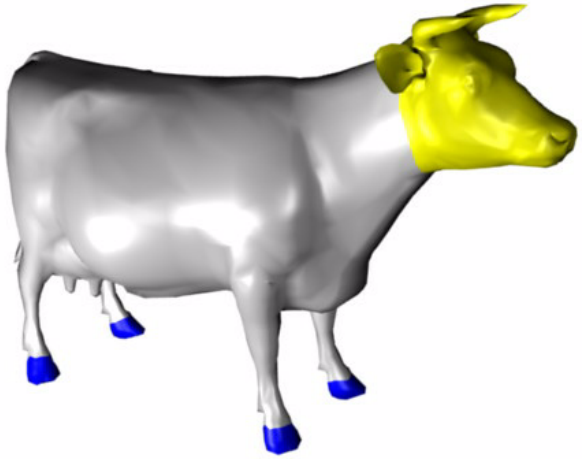


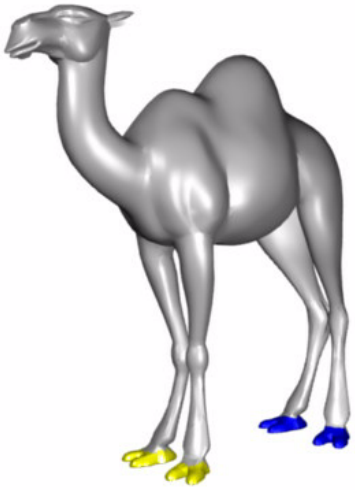
Computergraphik in Magdeburg

AG Visual Computing:
Geometric Modelling, Deformations



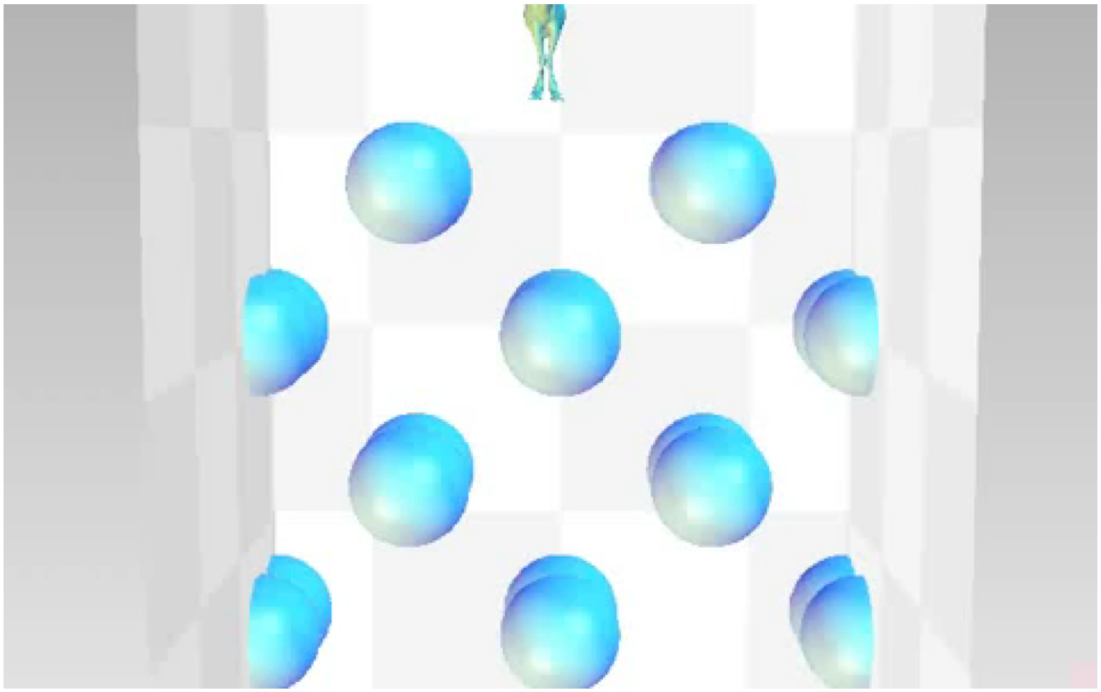
Introduction to Computer Graphics





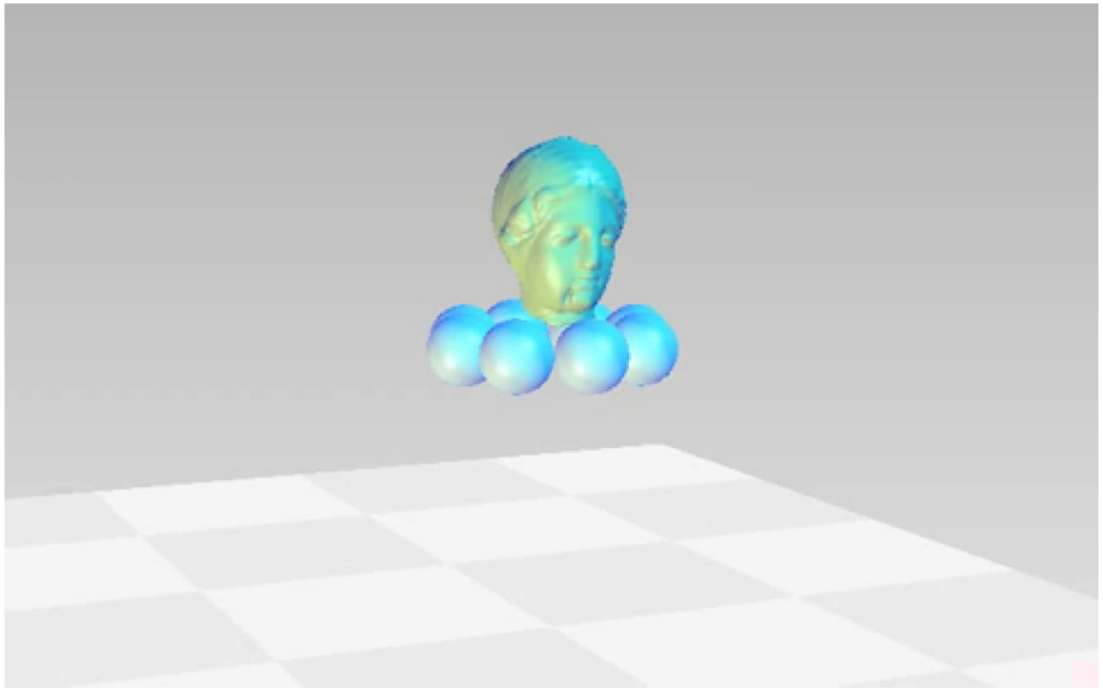
Computergraphik in Magdeburg

AG Visual Computing:
Geometric Modelling, Deformations



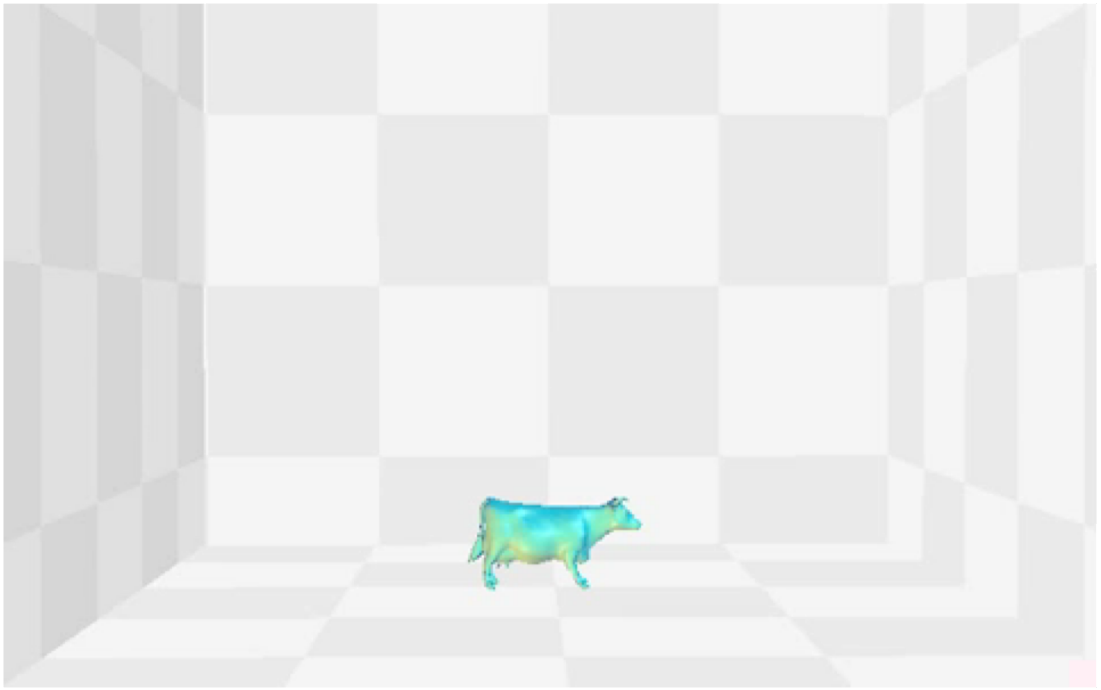
Computergraphik in Magdeburg

AG Visual Computing:
Geometric Modelling, Deformations



Computergraphik in Magdeburg

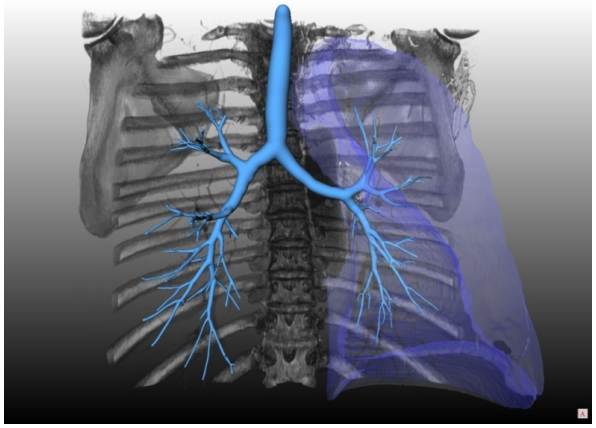
AG Visual Computing:
Geometric Modelling, Deformations



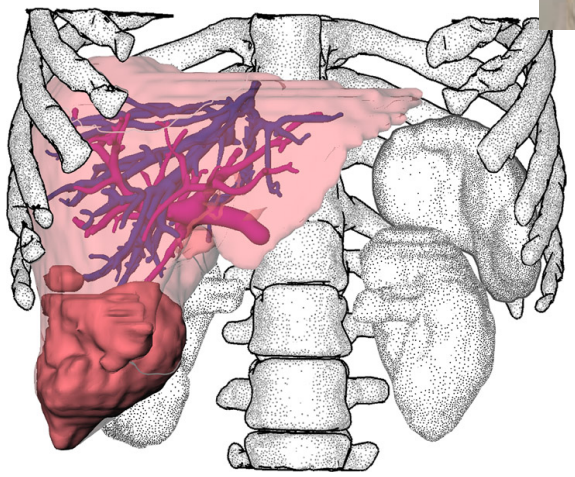
Computergraphik in Magdeburg:

AG Visualisierung (Prof. Preim)

Gefäßvisualisierung und illustrative Visualisierung



Ausgusspräparat der Bronchialgefäße einer menschlichen Lunge. Der linke Lungenlappen ist transparent dargestellt zusammen mit einem Volume Rendering weiterer Gefäße und umgebender Knochen.

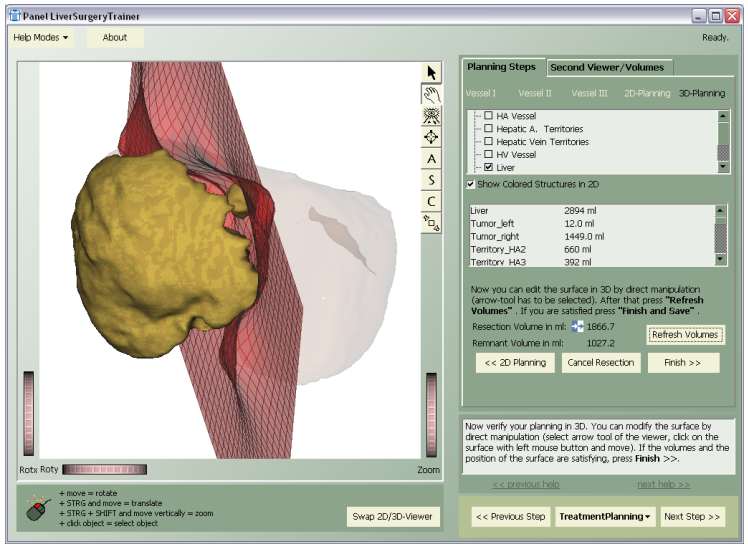


Fokus- und Kontextdarstellung durch farblich hervorgehobene Fokusobjekte und Repräsentation von Kontextstrukturen mittels Stippling und Silhouetten.

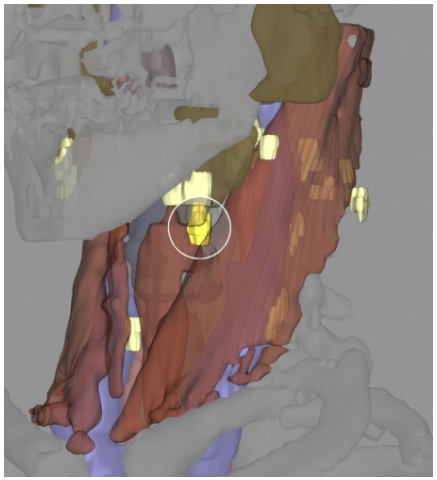
Computergraphik in Magdeburg:

AG Visualisierung (Prof. Preim)

Planung und Training von chirurgischen Eingriffen



Der Nutzer kann in einer 3D-Darstellung eine virtuelle Resektion vornehmen.



Kombination von Farbe, Transparenz-Mapping und lokaler Transparenz um einen einzelnen Hals-Lymphknoten hervorzuheben.

Arbeitsgruppe Computerassistierte Chirurgie

- Augmented Reality zur Unterstützung von Operationen
- Visualisierungstechniken zur besseren Navigation chirurgischer Instrumente

→ Viele Analogien zur Navigation in 3D Computerspielen

Navigierte Leberoperation, Asklepios Klinik Hamburg

Jun. Prof. Dr. Christian Hansen

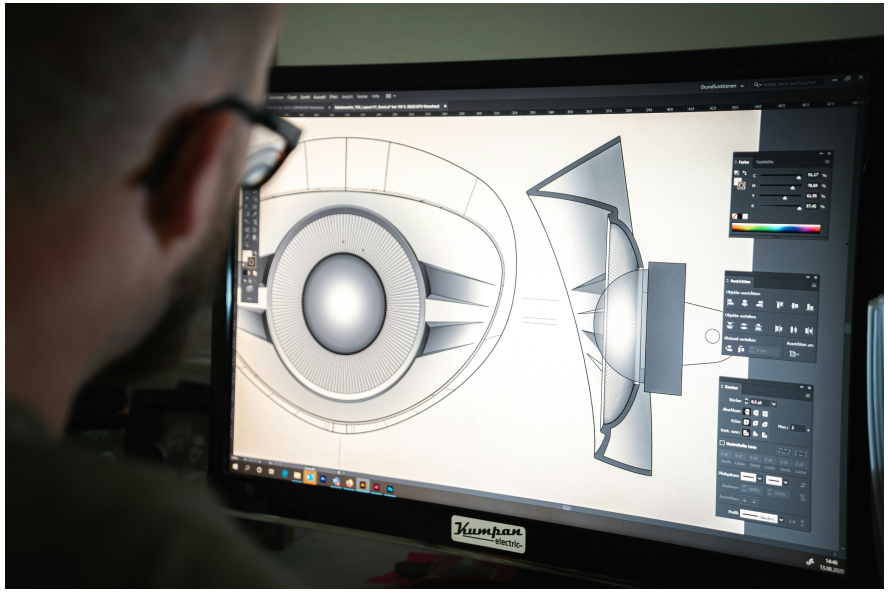
Computergraphik in Magdeburg:

Fraunhofer-Institut für Fabrikbetrieb
und -automatisierung

Visual Computing
University of Magdeburg

1. Einführung
1.3 Anwendungen der Computergraphik

- Rechnergestützter Entwurf, Konstruktion und Manipulation (CAE)
  - Entwurf und Konstruktion von Produkten - Flugzeuge, Automobile, Häuser, Produktions- / Energieanlagen, etc. (CAD),



- Graphik in Steuerungs-, Überwachungs- u. Informations-systemen
  - Leitstände, Prozeßüberwachung und - darstellung,
  - Meßwertaufnahme u. –aufbereitung,

- Ausbildung und Training
  - Flug- und Fahrsimulatoren,
  - militärische Anwendungen;

- Plotsysteme, Zeichnungserstellung, Kartographie
  - Geschäfts-, Technik- und Ingenieurgraphiken,
  - Darstellung geologischer und geographischer Daten, z.B. Landkarten Liegenschaftskarten, Reliefkarten, Wetterkarten, etc.
- Visualisierung von Daten und Informationen
  - Physikalische / chemische Prozesse, Strömungsmechanik,
  - Wetter- / Klimadaten,
  - Medizinische Daten;

- Kunst,Werbung, Design, Unterrichtsfilme;
  Unterhaltungsfilme
  Graphik für Multimedia und Internet-Anwendungen,
- Virtual und Augmented Reality,
- Computerspiele !

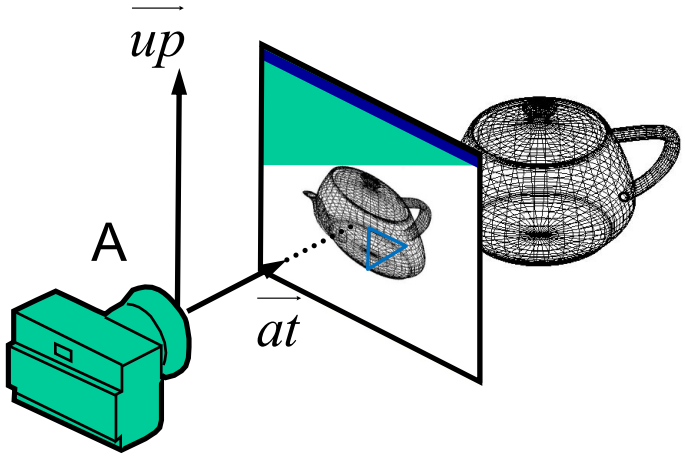
Visual Computing
University of Magdeburg

Raytracing

Eine ganz kurze Einführung

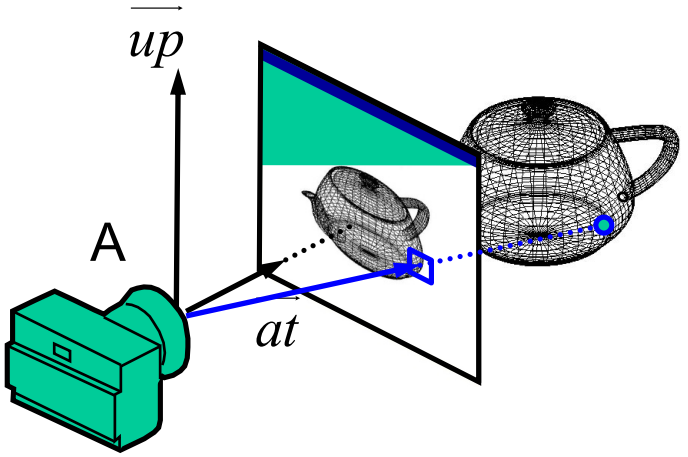
- In der Computergraphik gibt es 2 grosse Ansätze zur Bilderzeugung (rendering)
  - Projektion und Rasterisierung
  - Raytracing

Projektion/Rasterisierung



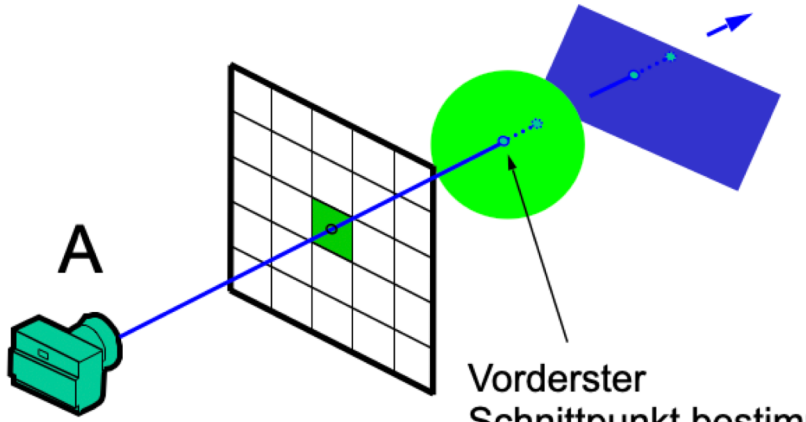
Die Polygone werden der Reihe nach rasterisiert

Ray Tracing



Das Bild wird durch Sehstrahlen abgetastet

- … steht für Strahlverfolgung.
- Man geht in einer Schleife über alle Pixel und berechnet einen Strahl vom Augpunkt über das jeweilige Pixel hinweg.
- Dieser Strahl wird mit der Geometrie der Szene geschnitten und der vorderste Schnittpunkt (das sichtbare Objekt) ermittelt
- Um sicherzustellen, dass es der vorderste Schnittpunkt ist, müssen im Prinzip alle Objekte der Szene getestet werden

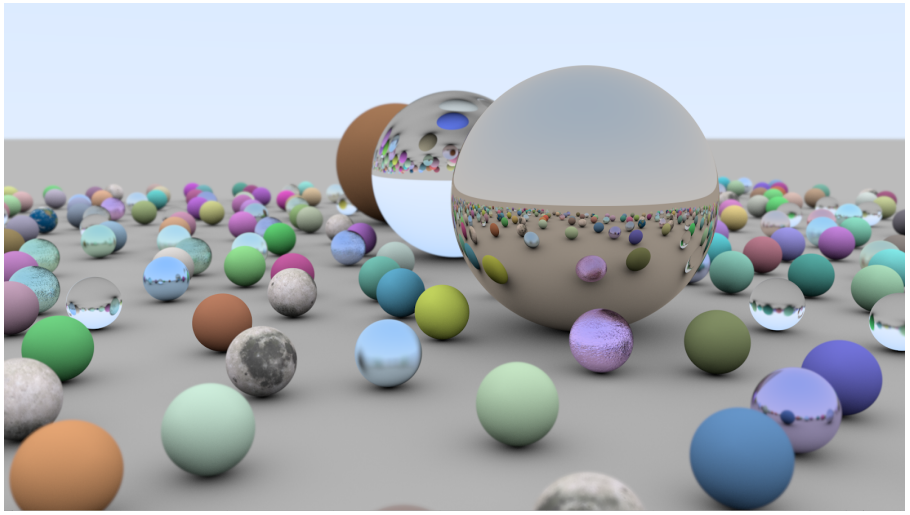


Schnitttest Strahl -Kugel

- Kugel in impliziter Beschreibung: $(\mathbf{x}\text{-}\mathbf{c})^2 - r^2 = 0$
  - $\mathbf{c}$: Kugelmittelpunkt
  - r : radius
- Strahl in Punkt-Richtungsgleichung: $\underline{\mathbf{r}}(t) = \mathbf{o} + t\,\mathbf{d}$
  - $\mathbf{o}=(o_x, o_y, o_z)$
  - $\mathbf{d}=(d_x, d_y, d_z)$
- zu lösen: $(\mathbf{o} + t\,\mathbf{d} - \mathbf{c})^2 - r^2 = 0$

  -> quadratische Gleichung in t

Leicht lösbar mit Standard-Techniken



Schnitttest Strahl Dreieck

- Strahl: $\underline{\mathbf{r}}(t) = \mathbf{o} + t\,\mathbf{d}$
- Dreieck: 3 punkte **a,b,c**
  - **x** ist innerhalb des Dreiecks wenn es drei Zahlen u,v,w gibt mit $\mathbf{x} = u\,\mathbf{a} + v\,\mathbf{b} + w\,\mathbf{c}$ und $u+v+w = 1$ und $0 <= u,v,w <= 1$
  - u,v,w : baryzentrische Koordinaten of **x**

- -> löse [$\mathbf{o} + t\,\mathbf{d} = u\,\mathbf{a} + v\,\mathbf{b} + w\,\mathbf{c}$, $u+v+w = 1$]
- -> lineares Gleichungssystem mit den 4 Unbekannten u,v,w,t
- -> i.a. eindeutige Lösung
- -> Strahl trifft Dreieck falls $0 <= u,v,w <= 1$ und $0 < t$.

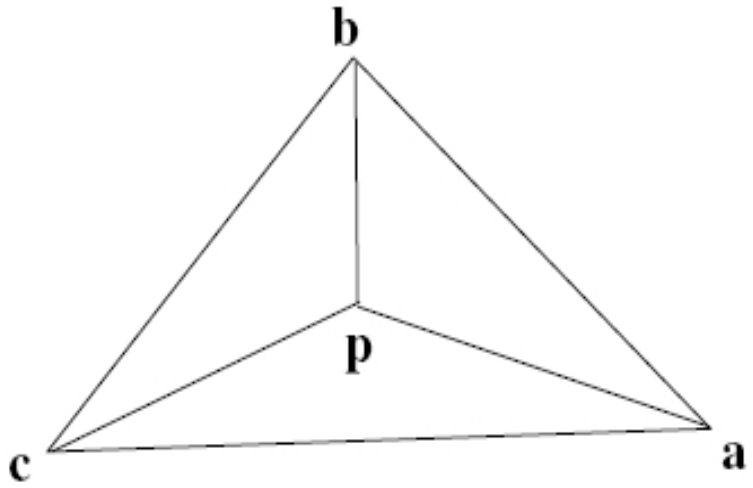
- **Baryzentrische Koordinaten**
  - gegeben: 3 Punkte $\mathbf{a}, \mathbf{b}, \mathbf{c}$ (nicht-kollinear). Jeder weitere Punkt $\mathbf{p}$ in der Ebene $\mathbf{a}, \mathbf{b}, \mathbf{c}$ kann als Linearkombination von $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ beschrieben werden:

$$\mathbf{p} = u \cdot \mathbf{a} + v \cdot \mathbf{b} + w \cdot \mathbf{c}$$

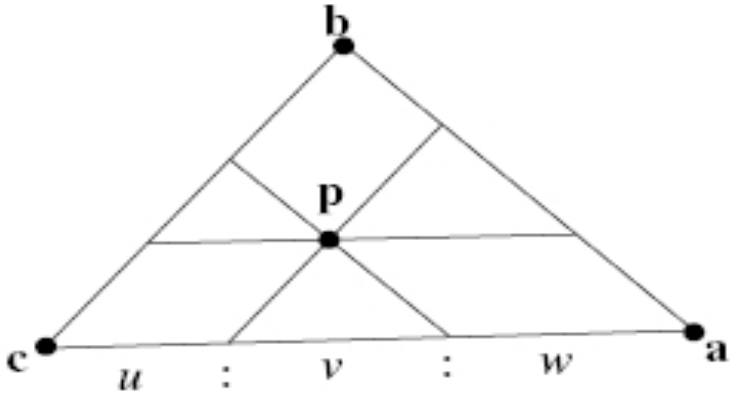
mit $u + v + w = 1$.

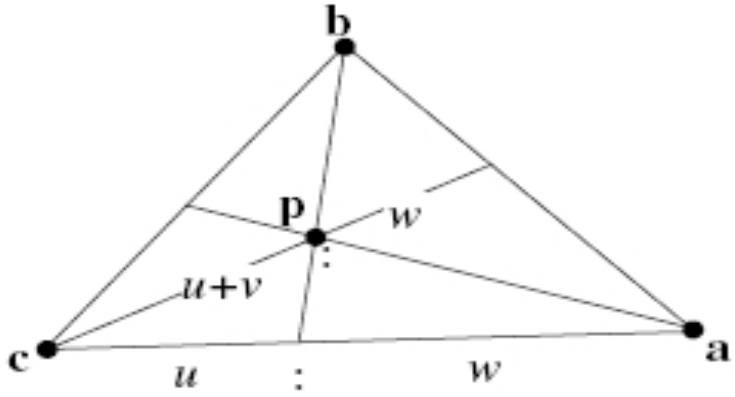
Dann sind, (u, v, w) die baryentrischen Koordinaten $\mathbf{p}$ relativ zu $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

- $0 \leq u, v, w \leq 1 \Leftrightarrow \mathbf{p}$ ist innerhalb des Dreiecks **a, b, c**

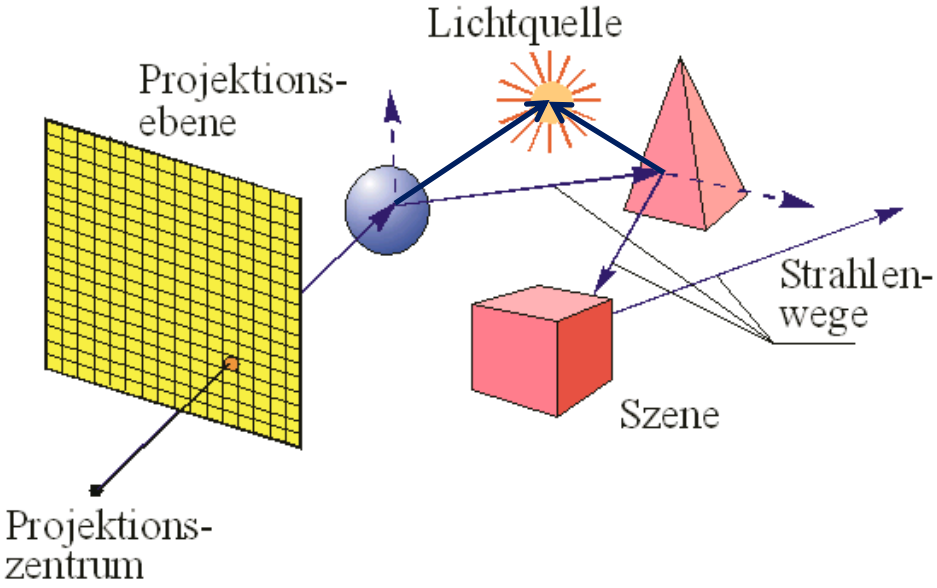


$$u = \frac{\text{area}(\mathbf{p}, \mathbf{b}, \mathbf{c})}{\text{area}(\mathbf{a}, \mathbf{b}, \mathbf{c})} \;\; ; \;\; v = \frac{\text{area}(\mathbf{p}, \mathbf{a}, \mathbf{c})}{\text{area}(\mathbf{a}, \mathbf{b}, \mathbf{c})} \;\; ; \;\; w = \frac{\text{area}(\mathbf{p}, \mathbf{a}, \mathbf{b})}{\text{area}(\mathbf{a}, \mathbf{b}, \mathbf{c})}$$

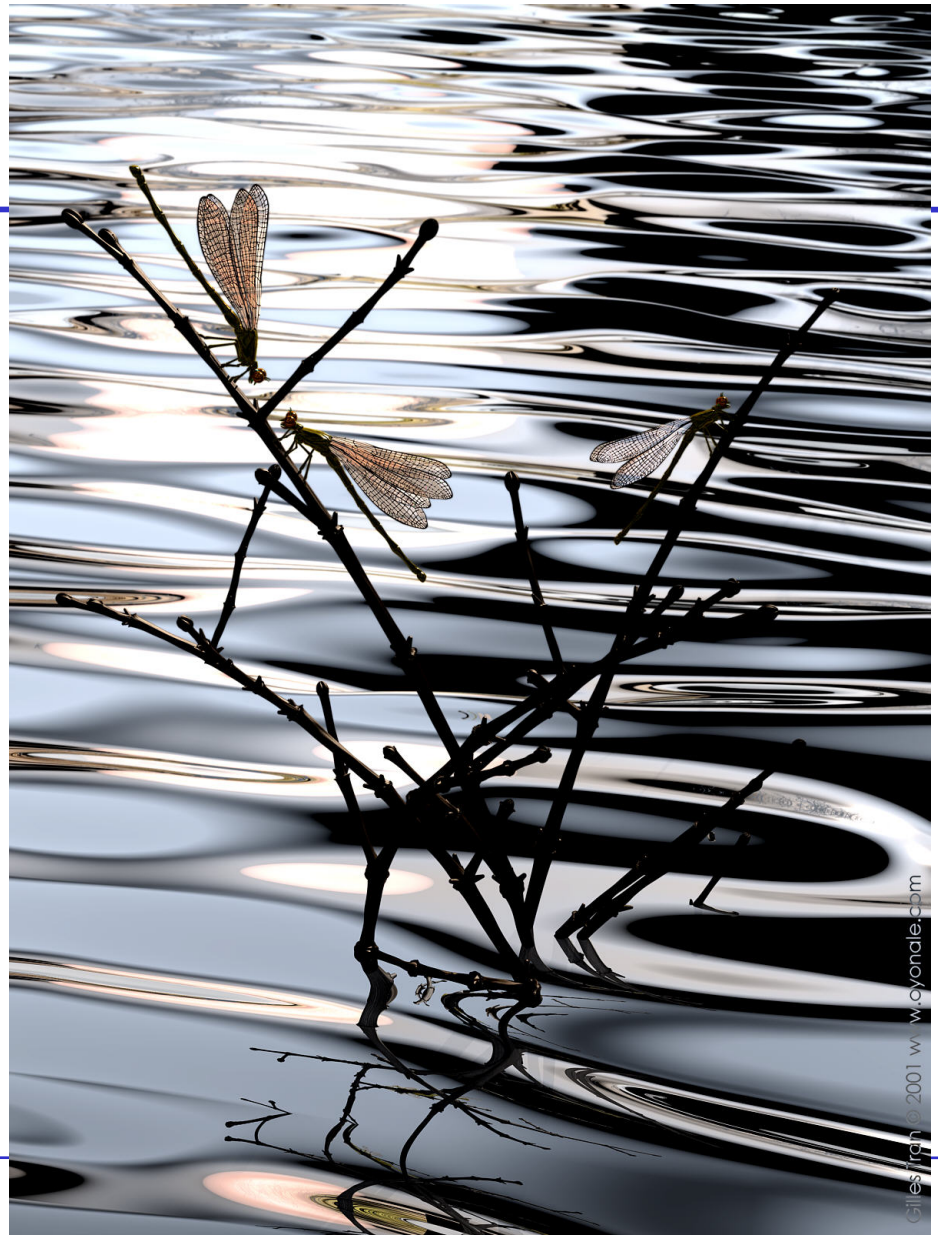

b
p
c
a
u
:
v
:
w

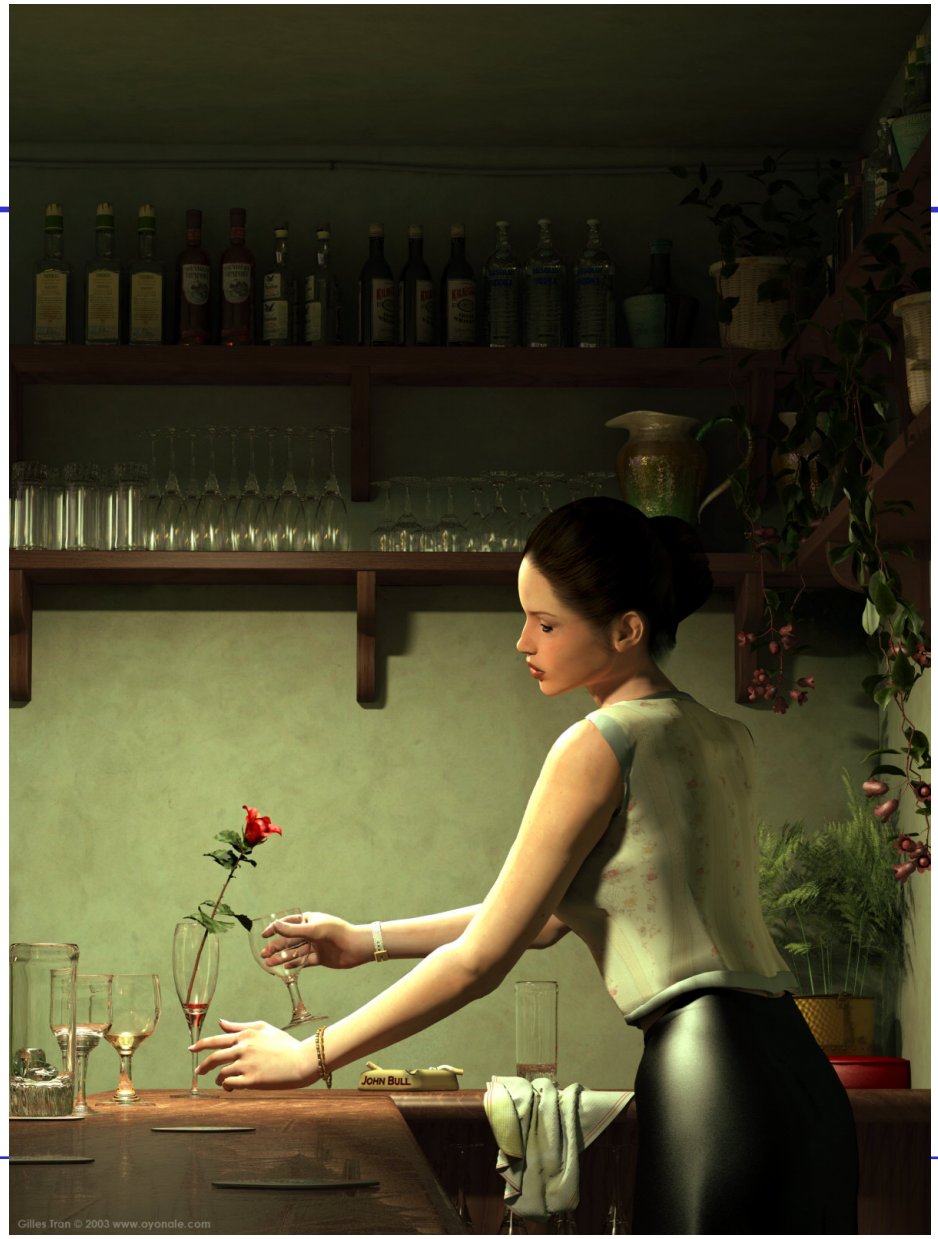

b
p
w
:
u+v
c
a
u
:
w

Schatten
Reflexionen
Brechungen
Sichtbarkeit
Schattierung



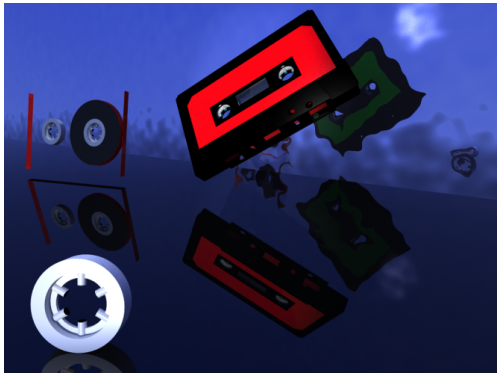
Raytraycing-Bild,
Rendered mit PovRay

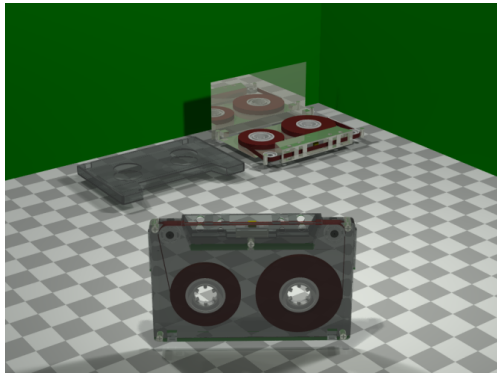

Gilles Iran © 2001 www.oyonale.com

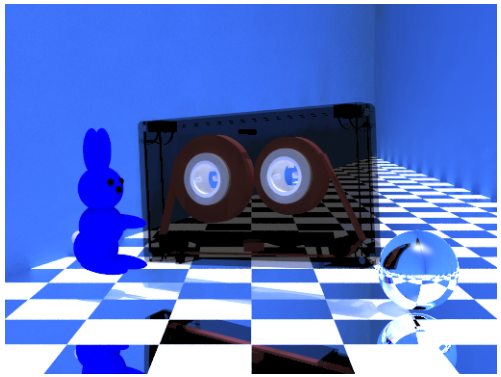
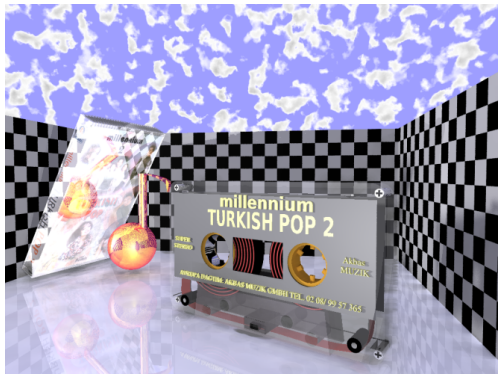

JOHN BULL
Gilles Tran © 2003 www.oyonale.com


© 2003 Hildur Kolbrún


TOW
AWAY
ZONE
S 403ST
ONE WAY
DONT
WALK
Gilles Tran © 2000 www.oyonale.com







millennium
TURKISH POP 2