Computer Graphics I

Lecture 1: Introduction

Xiaopei LIU

School of Information Science and Technology
ShanghaiTech University

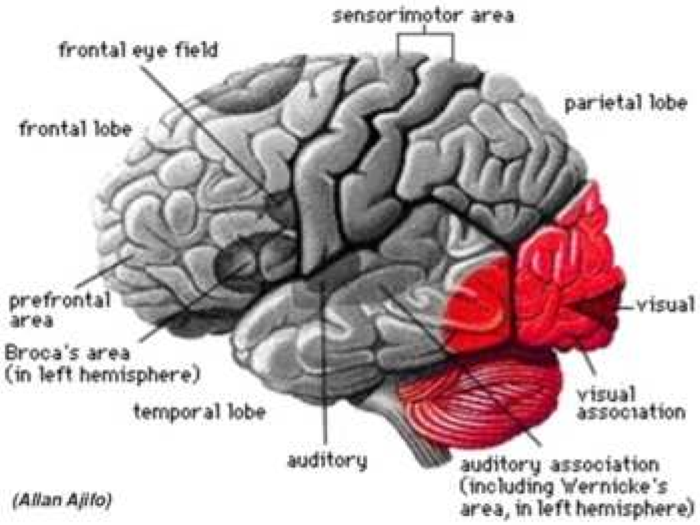
Teaching computer graphics course

- **Lecturer**
  - *Liu Xiaopei*, liuxp@shanghaitech.edu.cn;
    - office hour: Fri: 17:00~18:00
    - address: SIST Building No. 2, Rm 202.J

- **Teaching assistants**
  - *Zhonghan Zhang*, zhangzhh2022@shanghaitech.edu.cn;
    - office hour: Mon: 20:30-21:30, address: SIST Building No. 2, Room 213
  - *Yanxi Huang*, huangyx12024@shanghaitech.edu.cn;
    - office hour: Fri: 13:30 - 14:30, address: SIST Building No. 2, Room 213

Visual information

- **About 30% of brain is dedicated to visual processing**



(Allan Ajifo)

(Petar Milošević)

...eyes are highest-bandwidth port into the head!

What is computer graphics?

- **Common definition**
  - Pictures and films created using computers
- **Broader definition**
  - Almost everything on computers that are not text or sound

From movie “Life of Pi”

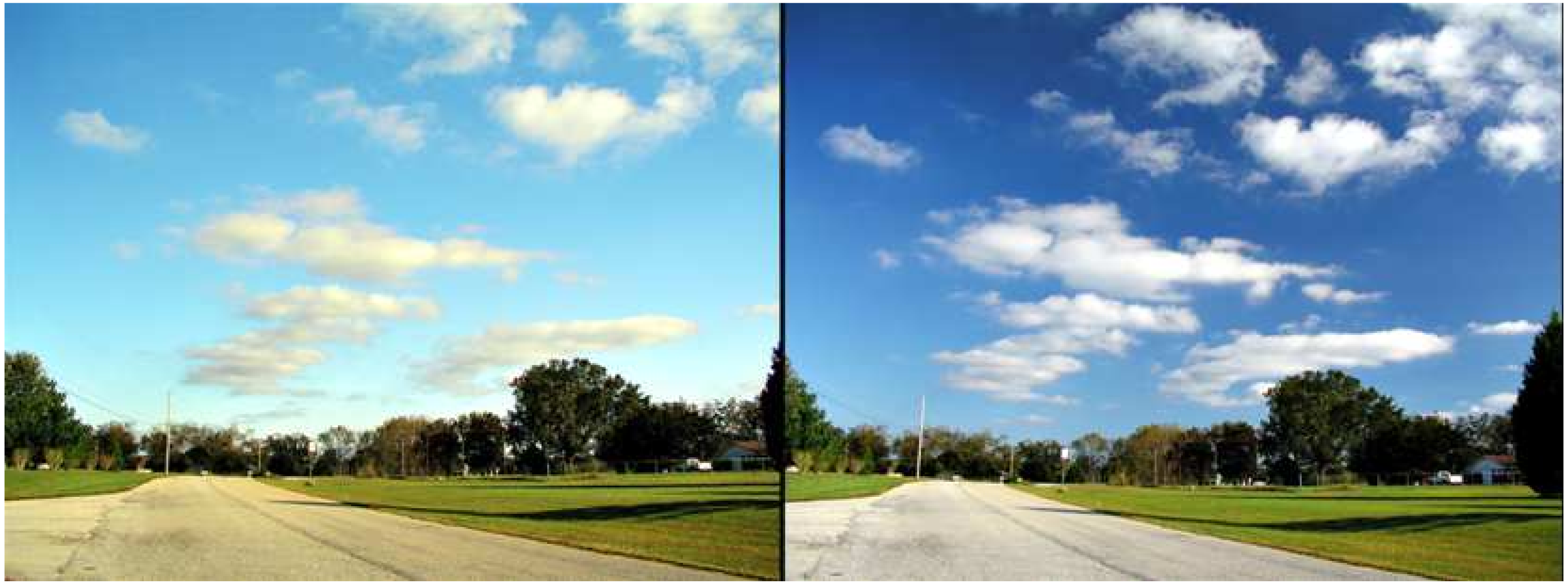
What is computer graphics

What is computer graphics?

- **Computer graphics is**
  - A discipline in applied computer science
  - An interdisciplinary science involving computer science, mathematics and physics

Difference from other visual disciplines

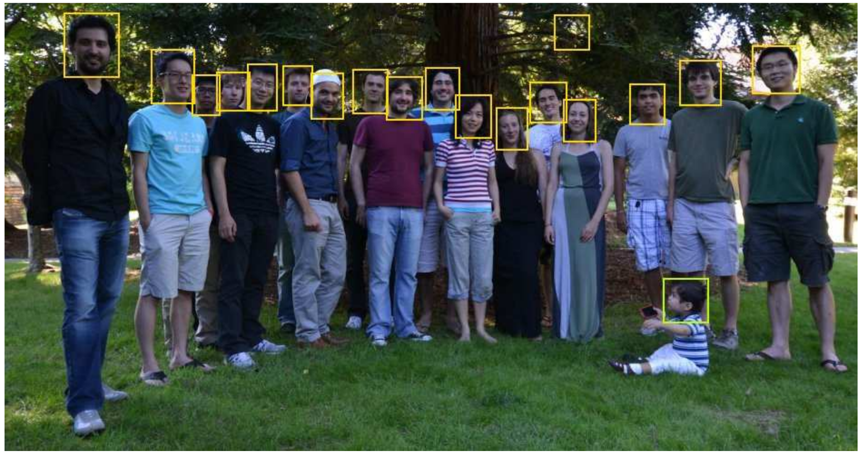
- **Difference from image processing**
  - Synthesize new images from input images

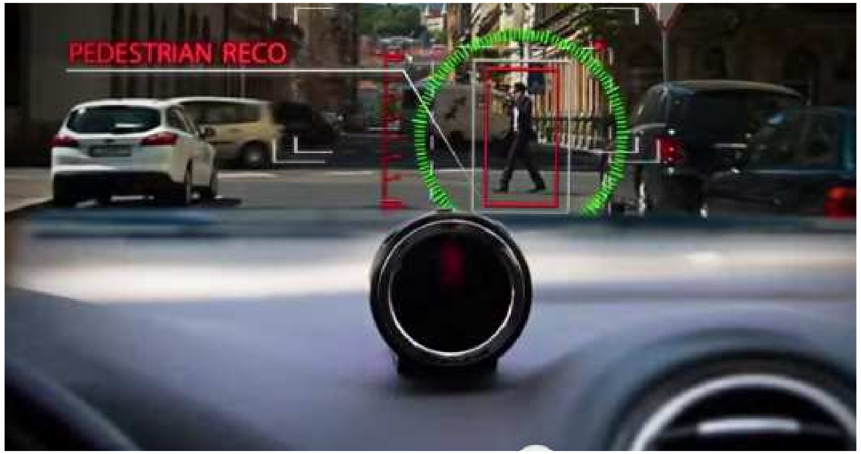


Emulating optical polarizing filter

Difference from other visual disciplines

- **Difference from computer vision**
  - From image to abstract knowledge (understanding)

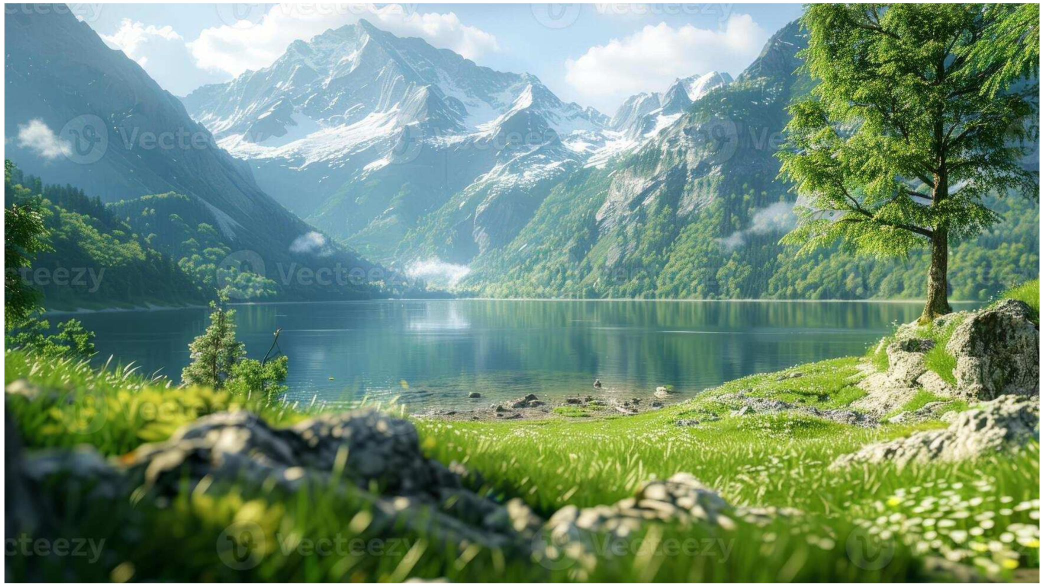




Face and object detection

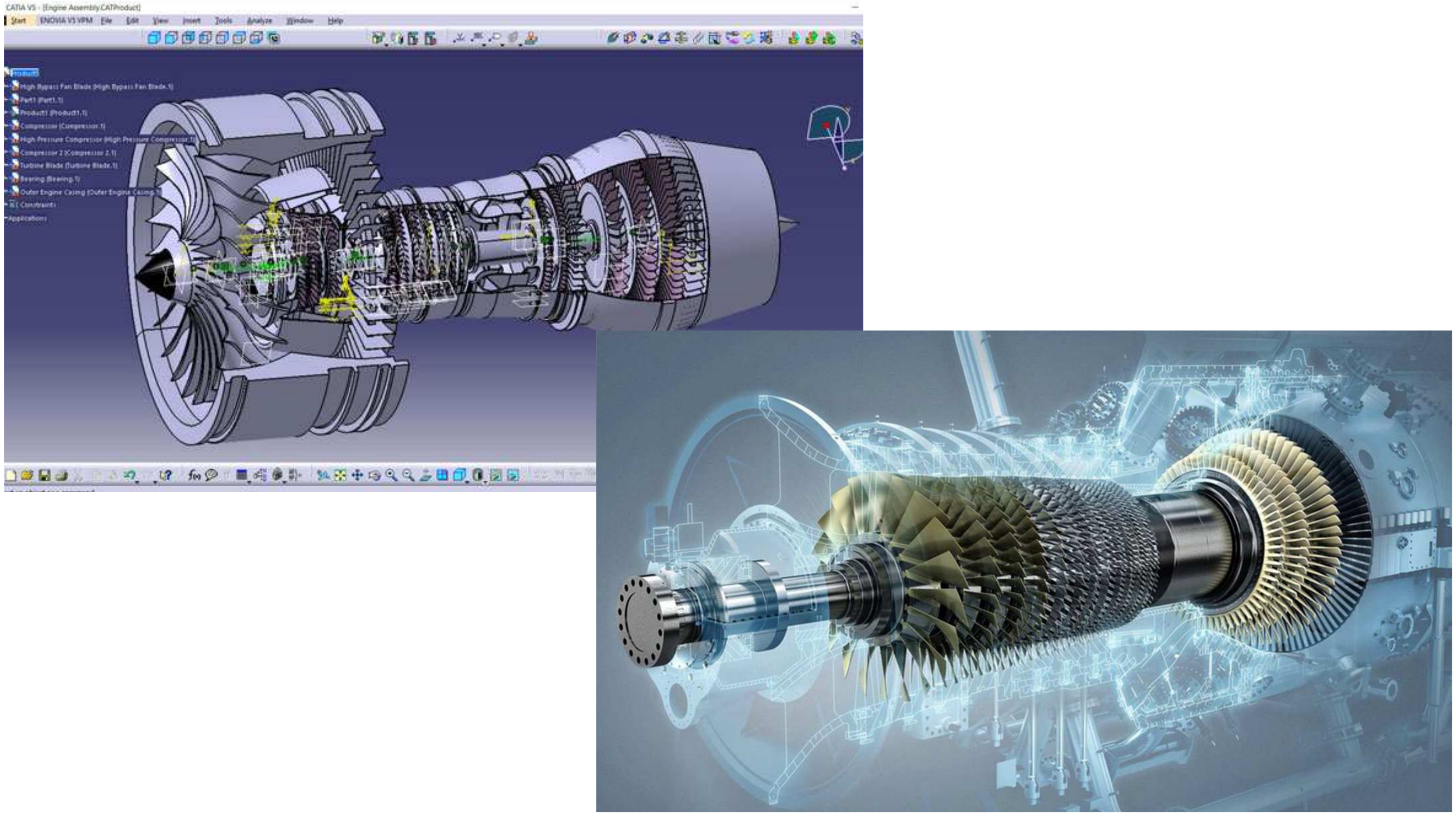
Computer graphics in the age of AI

- **Nowadays, large-language models (LLM) can generate complex graphics**
  - Based on a transformer architecture
  - Trained on large dataset
  - But where do the dataset come from?



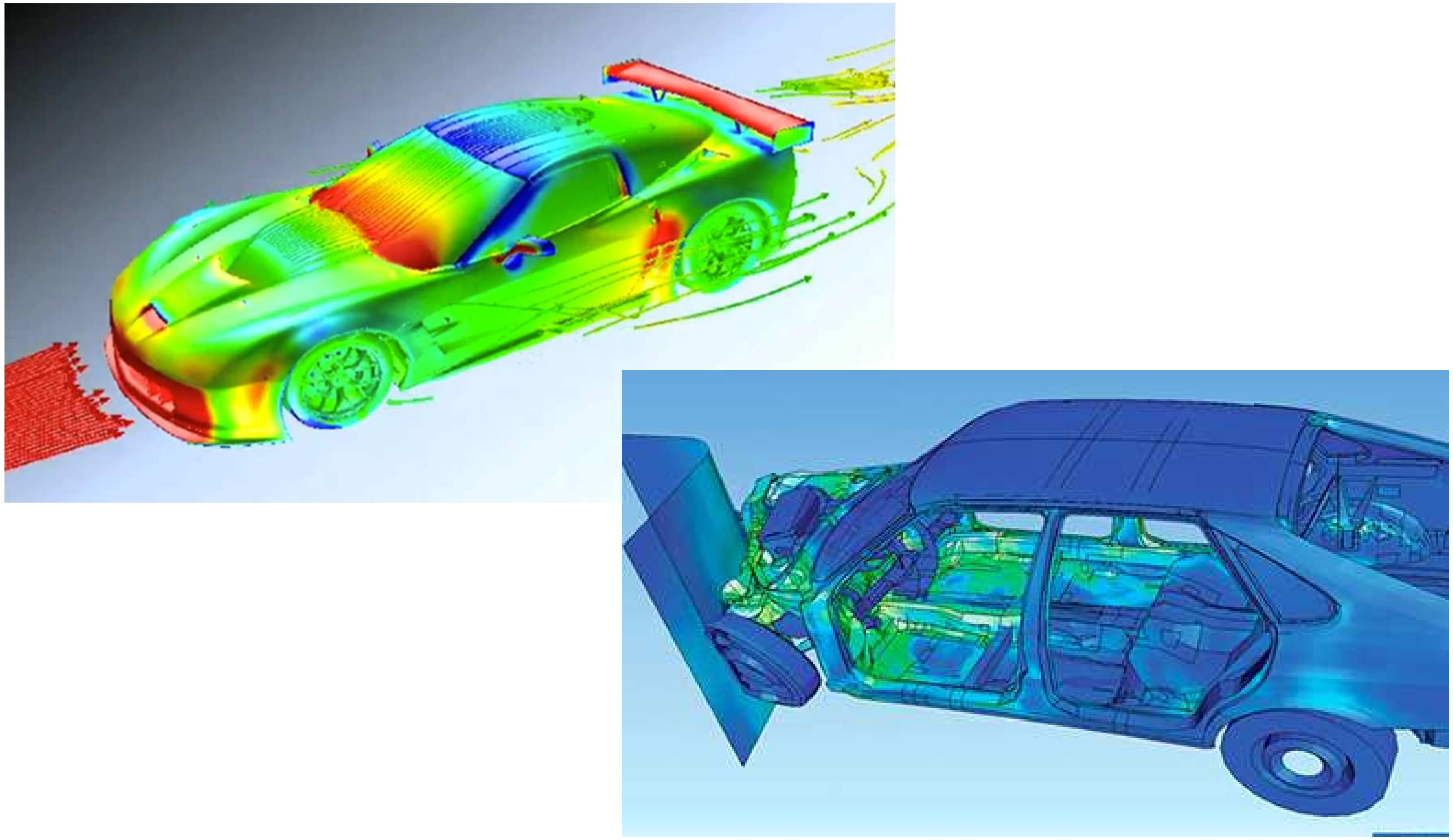
Applications of computer graphics

- **Computer-aided design (CAD)**


CATIA V5 - [Engine Assembly.CATProduct]
Start
ENOVIA V5 VPM
File
Edit
View
Insert
Tools
Analyze
Window
Help
Product1
High Bypass Fan Blade (High Bypass Fan Blade.1)
Part1 (Part1.1)
Product1 (Product1.1)
Compressor (Compressor.1)
High Pressure Compressor (High Pressure Compressor.1)
Compressor 2 (Compressor 2.1)
Turbine Blade (Turbine Blade.1)
Bearing (Bearing.1)
Outer Engine Casing (Outer Engine Casing.1)
Constraints
Applications

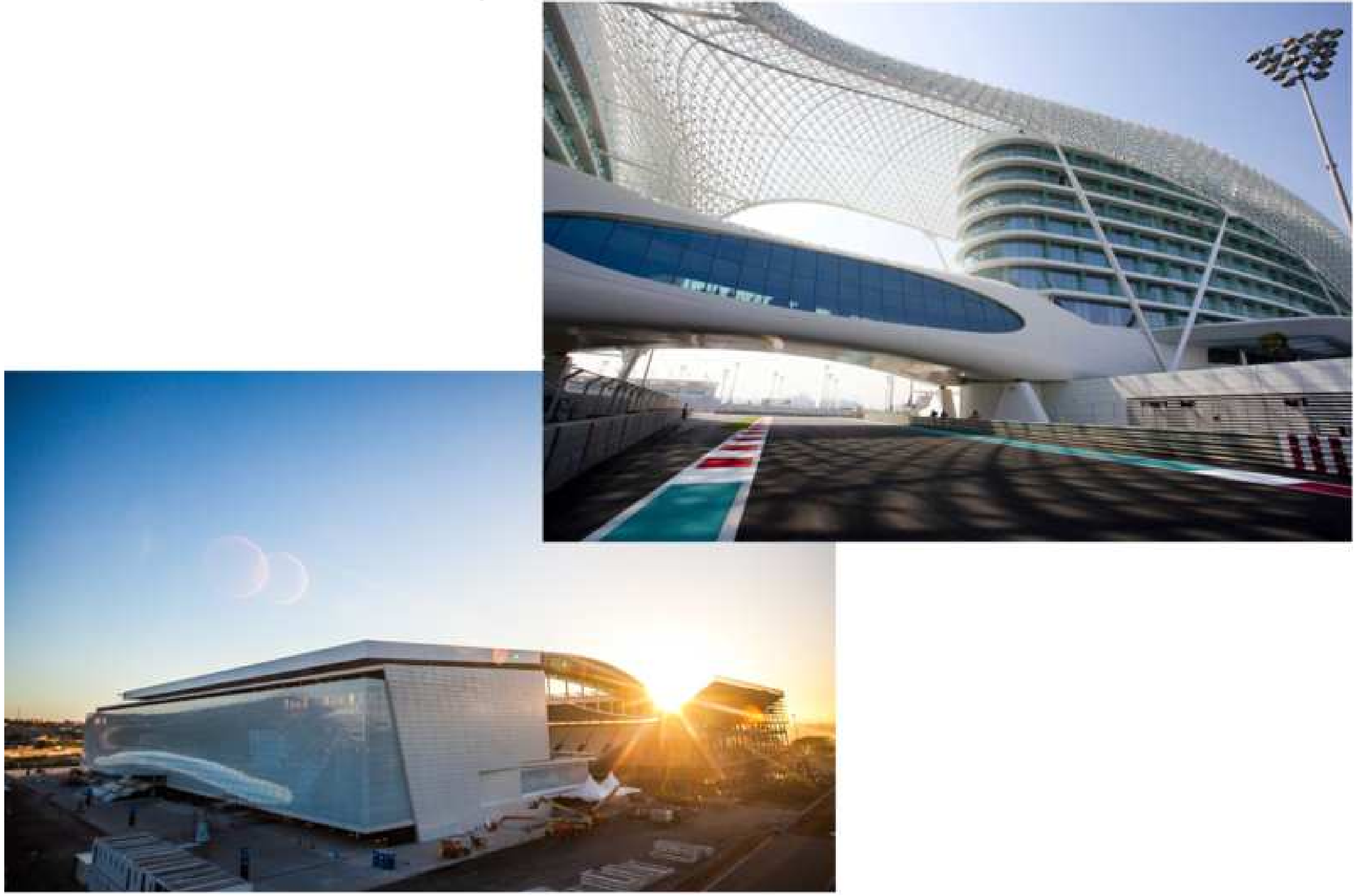
Applications of computer graphics

- **Computer-aided engineering (CAE)**



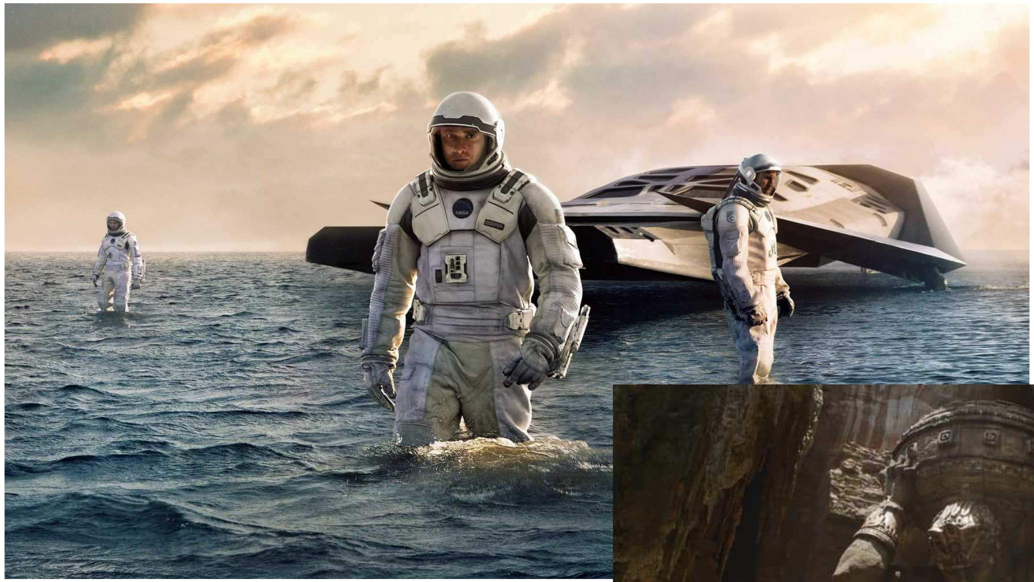
Applications of computer graphics

- **Architecture design**



Applications of computer graphics

- **Entertainment**

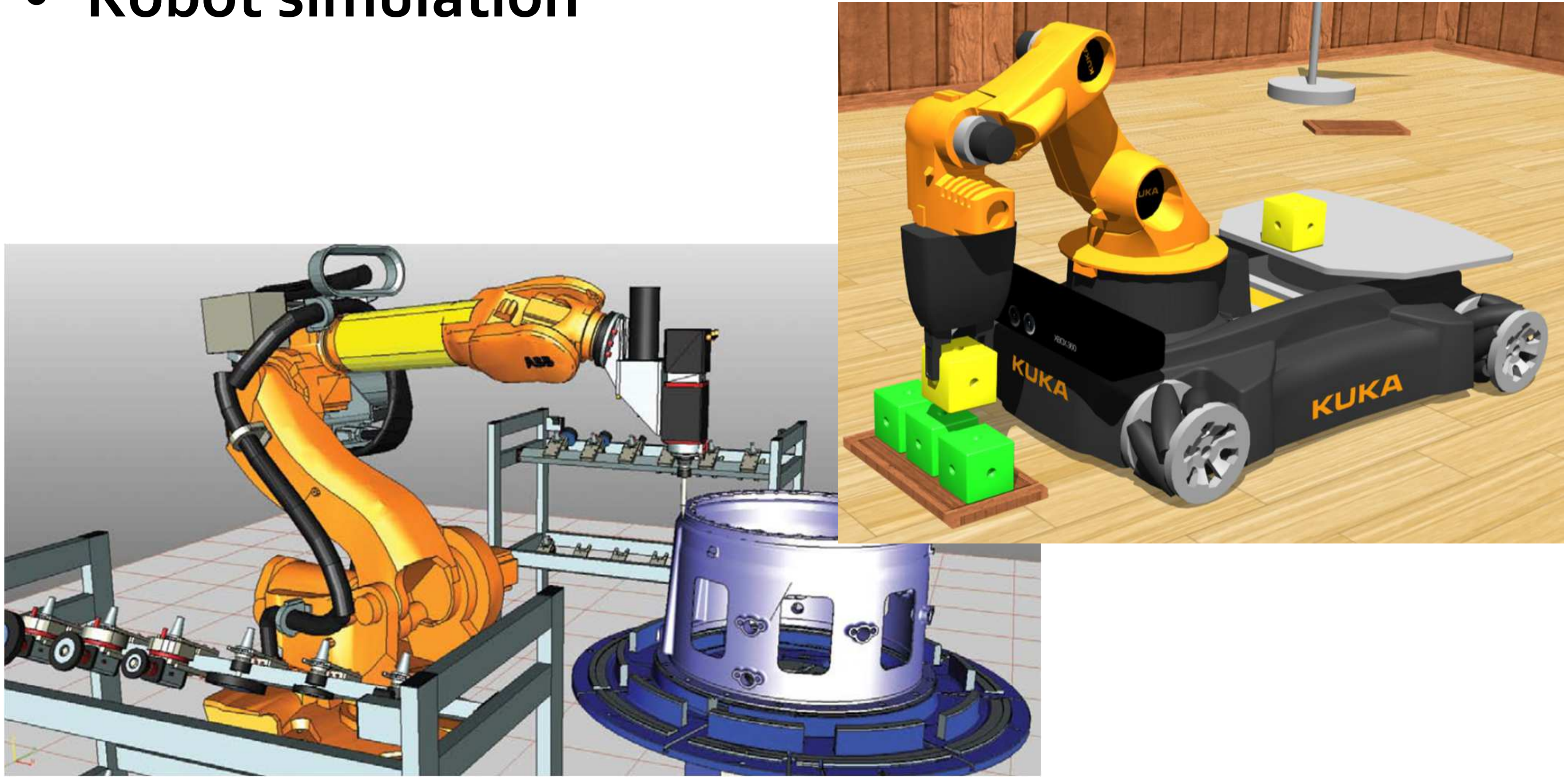


Applications of computer graphics

- **Art and design**

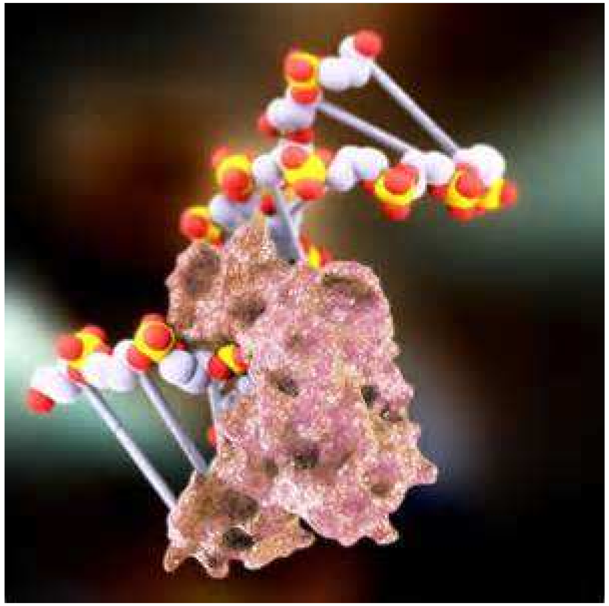
Applications of computer graphics

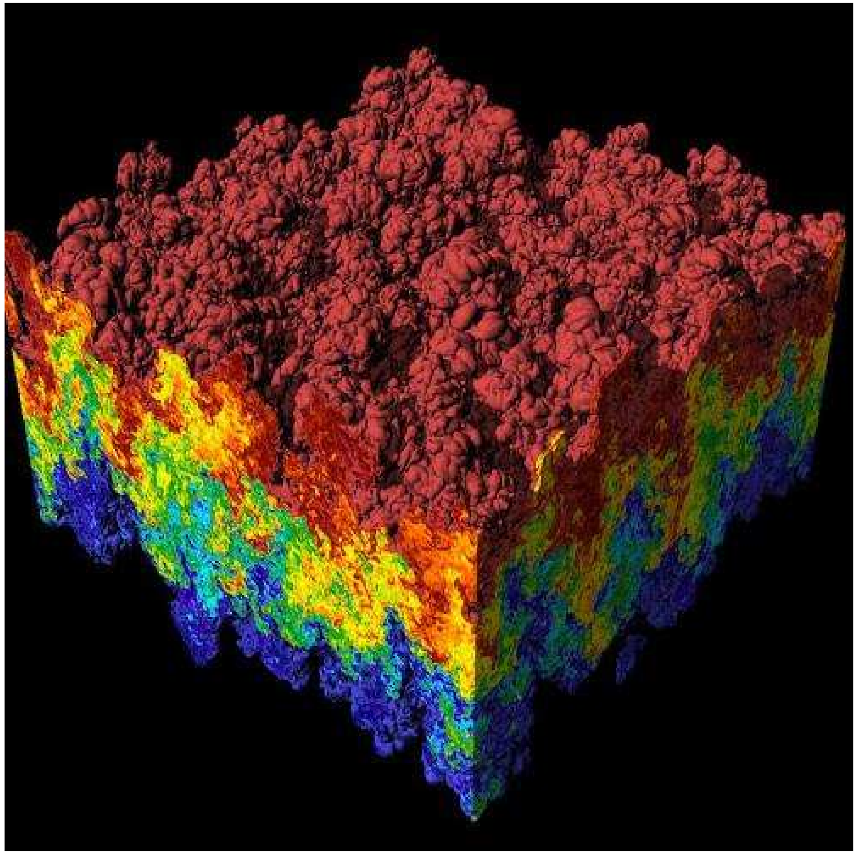
- **Robot simulation**



Applications of computer graphics

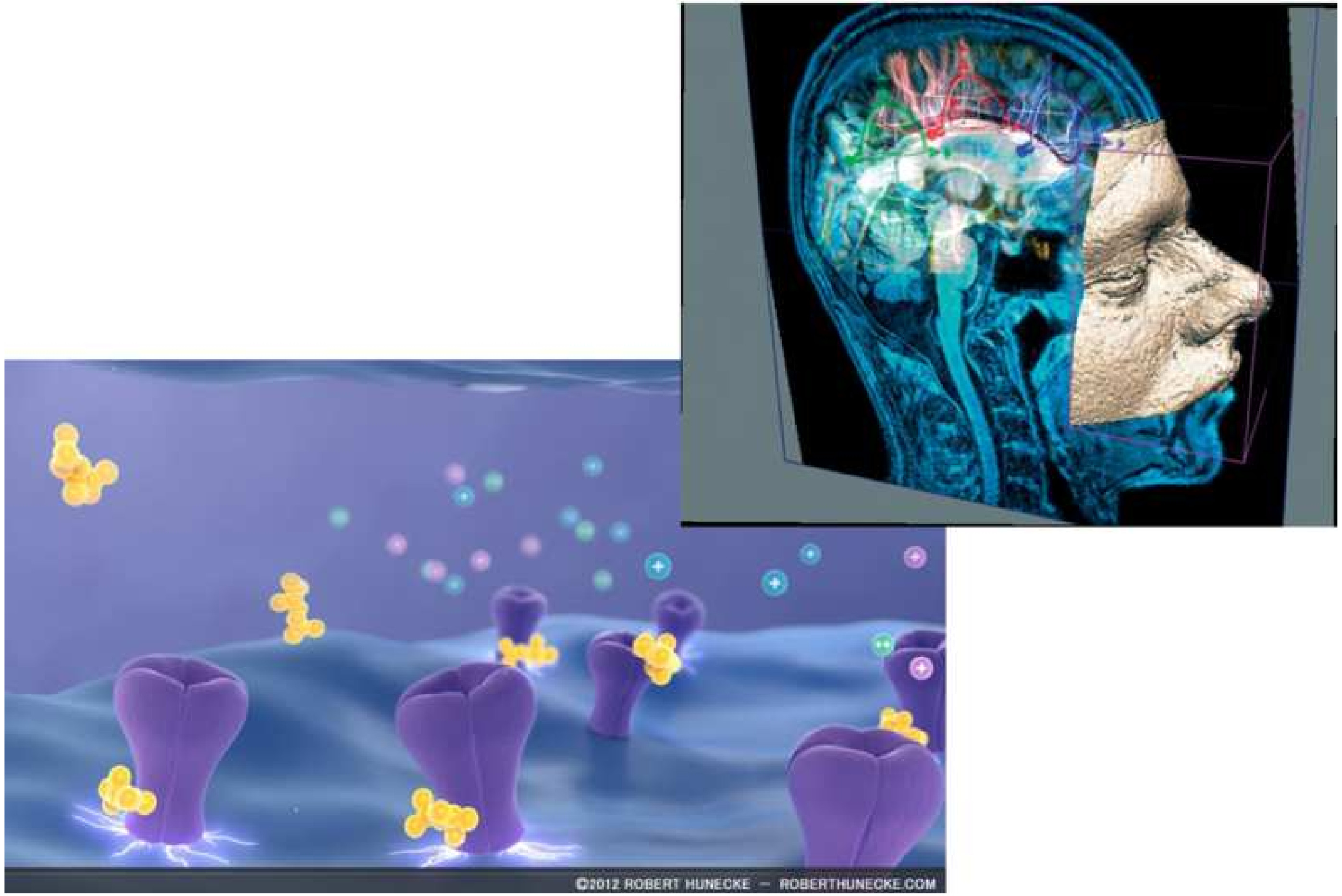
- **Scientific & mathematical visualization**





Applications of computer graphics

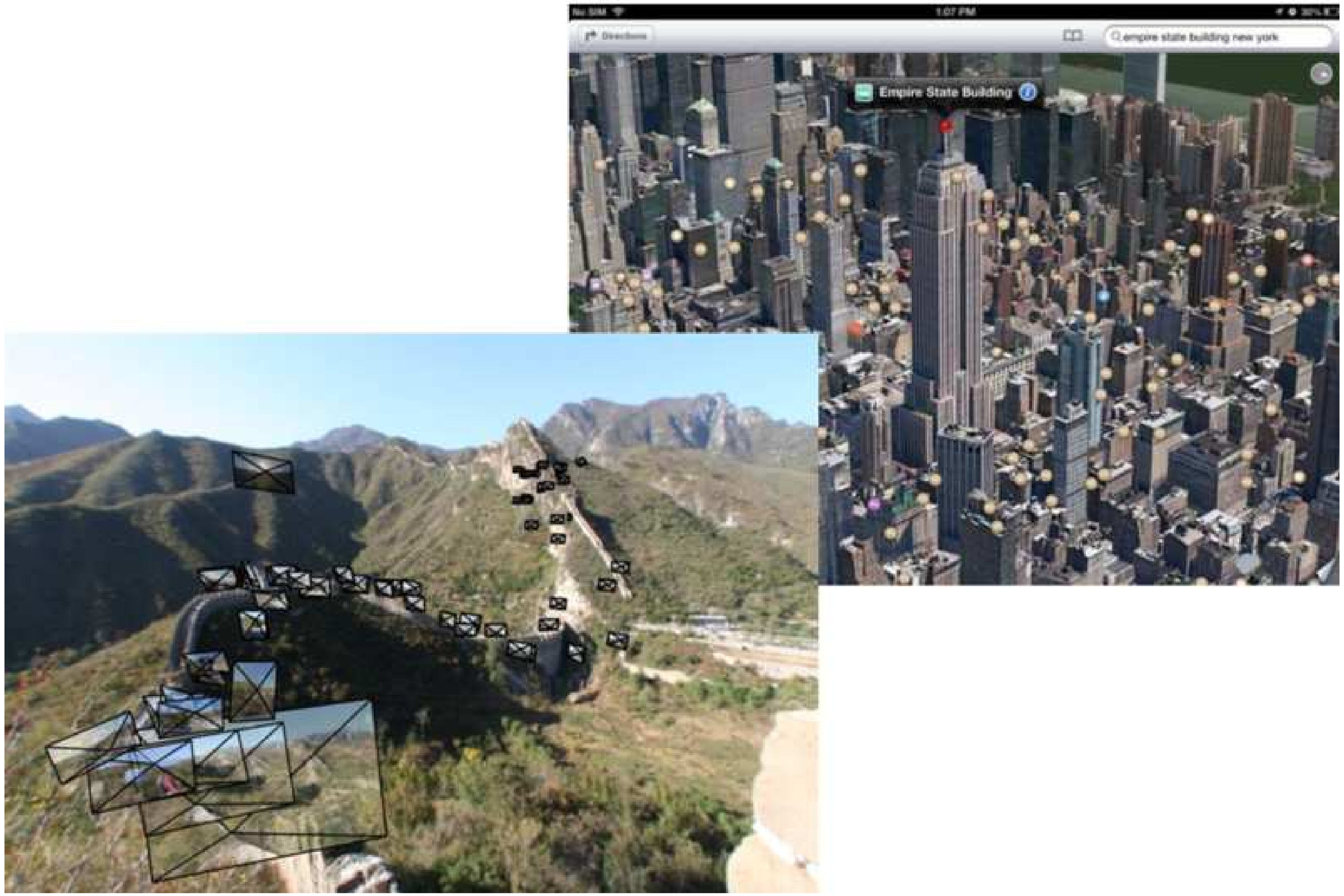
- **Medical & anatomical visualization**



©2012 ROBERT HUNECKE — ROBERTHUNECKE.COM

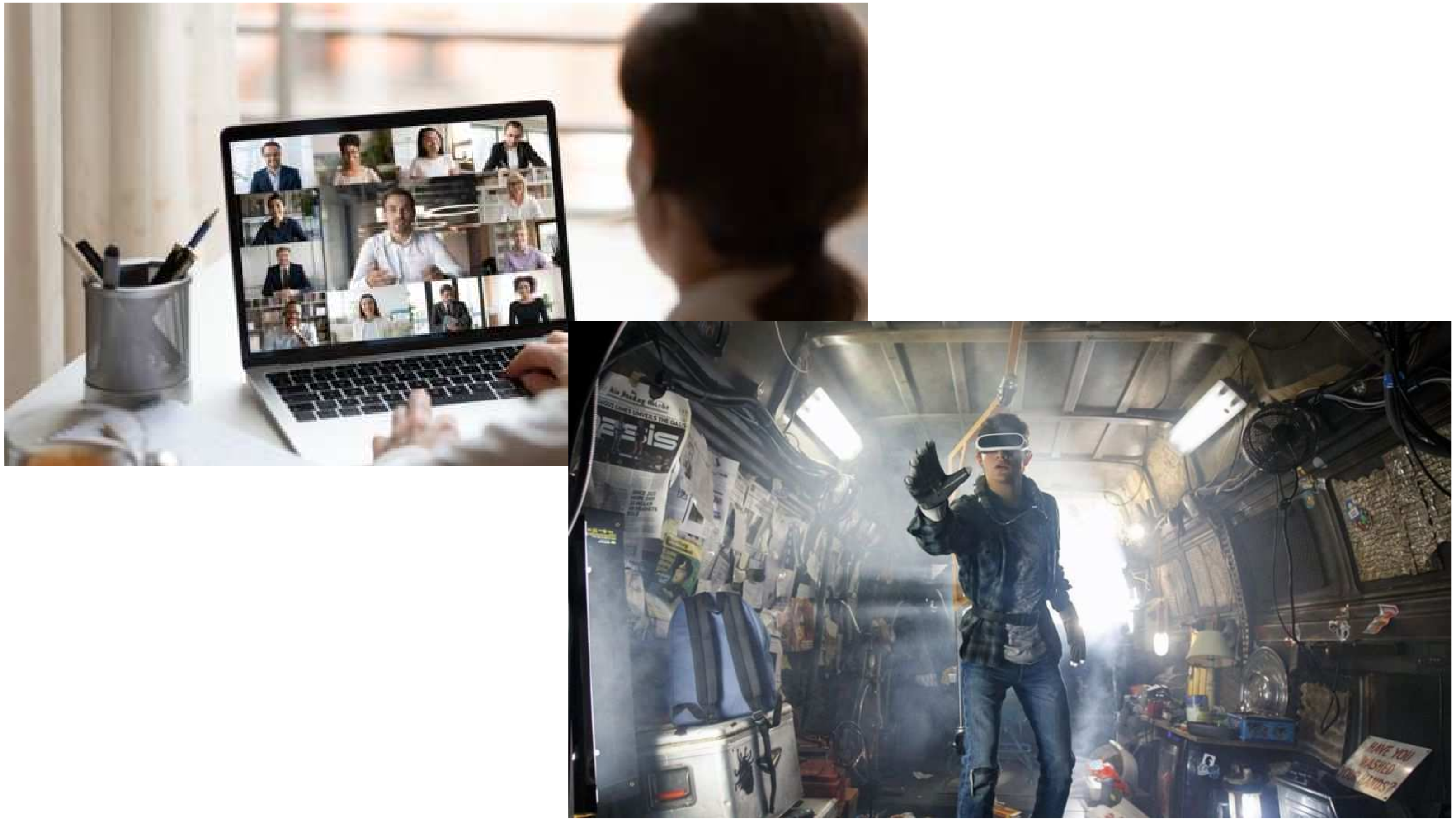
Applications of computer graphics

- **Visual navigation**



Applications of computer graphics

- **Visual communication**



Application of computer graphics

Topics in this course

- **Imaging**
  - Geometrical elements
  - Coordinate spaces and transformations
  - Projection-based rendering
  - Ray-based rendering
  - Texture mapping
  - Antialiasing

- **Geometric modeling and processing**
  - Curves, surfaces, meshes
  - Mathematical modeling
  - Geometric processing

Topics in this course

- **Material and reflectance modeling**
  - BRDF/BSSRDF
  - Texturing
- **Global illumination**
  - Monte-Carlo integration
  - Path tracing & photon mapping
- **Volume rendering**
  - Volume rendering equation
  - Transfer function

Topics in this course

- **Computer animation**
  - Keyframe-based animation
  - Rigid-body simulation
  - Soft-body simulation
  - Fluid simulation

- **Summary and final project**

How the course will be taught?

- **Lectures**
  - Each lecture will cover a specific topic
  - Contents in lectures can be found in course materials
- **After-class reading and tutorials**
  - We will give you materials for selective reading
  - Discussions on Piazza (we will add you all in) and tutorials
- **Programming assignments**
  - Small programming projects, from easier to more difficult ones
- **Written exams (mid-term, quizzes)**
- **Paper presentation**
- **Final Project**
  - A group project with a formal presentation & a technical report

Grading criterion

- **Assignments (x5) 50%**
  - Assignment 1: Creating your first OpenGL program （6%）
  - Assignment 2: 3D modeling with OpenGL rendering （10%）
  - Assignment 3: Ray tracing - basics（10%）
  - Assignment 4: Ray tracing - advanced（15%）
  - Assignment 5: Volume rendering（9%）
- **Oral presentation 5%**
- **Mid-term exam (written) (mid-term 15% + quizzes 2% x 5)**
- **Final project (20%)**
  - A group project (2~3 students) selected from a candidate project list or proposed by your own with permission from the instructor (15%)
  - Formal presentation with a technical report (presentation 3% + report 2%)
- **Piazza course:** https://piazza.com/shanghaitech.edu.cn/fall2025/cs17101
- **Checkout the specifics at** http://faculty.sist.shanghaitech.edu.cn/faculty/liuxp/course/cs171/

Reading materials

- **Reference books**
  - Physically Based Rendering - From Theory To Implementation
  - Advanced Global Illumination
  - Real-Time Volume Graphics
- **Other important materials**
  - Documentations, tutorial notes and websites
  - SIGGRAPH course notes
- **Some research papers**

How to learn computer graphics

- **Concepts and ideas are more important**
  - Try to grasp the general idea first
  - Progressively go into mathematical details

- **Implementation (C++) is an important practice**
  - Write specific graphics programs after class may help a lot
  - Starting from black-box framework (such as OpenGL)
  - Write up your own program from scratch (ray tracer)

- **Read advanced books/materials and research papers**
  - To obtain richer and more advanced knowledge

History of Computer Graphics

History of computer graphics

- **1950s**
  - Initial development of graphics hardware (CRT display, light pen, electronic gun, etc.);
  - Very small scattered graphics programs
- **1960s**
  - Formal terminology by William Fetter, a graphic designer for Boeing
  - Small video games (Star War); Boeing Aircraft/Renault automobile produced simulations for product design
  - Ivan Sutherland created stereoscopic 3D head-mounted display, rediscovered for virtual reality 20 years later
  - The basis algorithm for ray tracing; ACM initiated SIGGRAPH

History of computer graphics

- **1970s**
  - Many breakthroughs; Ivan Sutherland and David C. Evans (University of Utah) contributed a lot to the founding research;
  - Their students founded several of the industry's most important companies: Pixar, Silicon Graphics, and Adobe Systems
- **1980s**
  - Modernization and commercialization of computer graphics
  - Japan's Osaka University developed a supercomputer system for realistic graphical rendering
  - Computer graphics in movie industry started to grow: Lucasfilm and Industrial Light & Magic became known
  - In 1988, the first shaders developed by Pixar
  - Also the golden era of videogames

History of computer graphics

- **1990s**
  - Emergence of 3D modeling on a mass scale; an impressive rise in the quality of CGI generally
  - Microsoft Windows and Apple Macintosh started to popularize for home rendering software
  - Pixar began its serious commercial rise in this era (Toy Story in 1995)
  - Real-time 3D graphics, GPU technology developed; NVIDIA released the first home video card
  - OpenGL and DirectX

History of computer graphics

- **2000s**
  - CGI became ubiquitous; video games and CGI cinema had spread the reach of computer graphics
  - Increasing sophistication of GPU; 3D rendering capabilities became a standard feature
  - Shaders were introduced into GPU; GPGPU started
  - CUDA emerged at the end of the decade, and started to be applied to other domains, such as computer vision
  - Computer graphics used in films and video games gradually began to be realistic

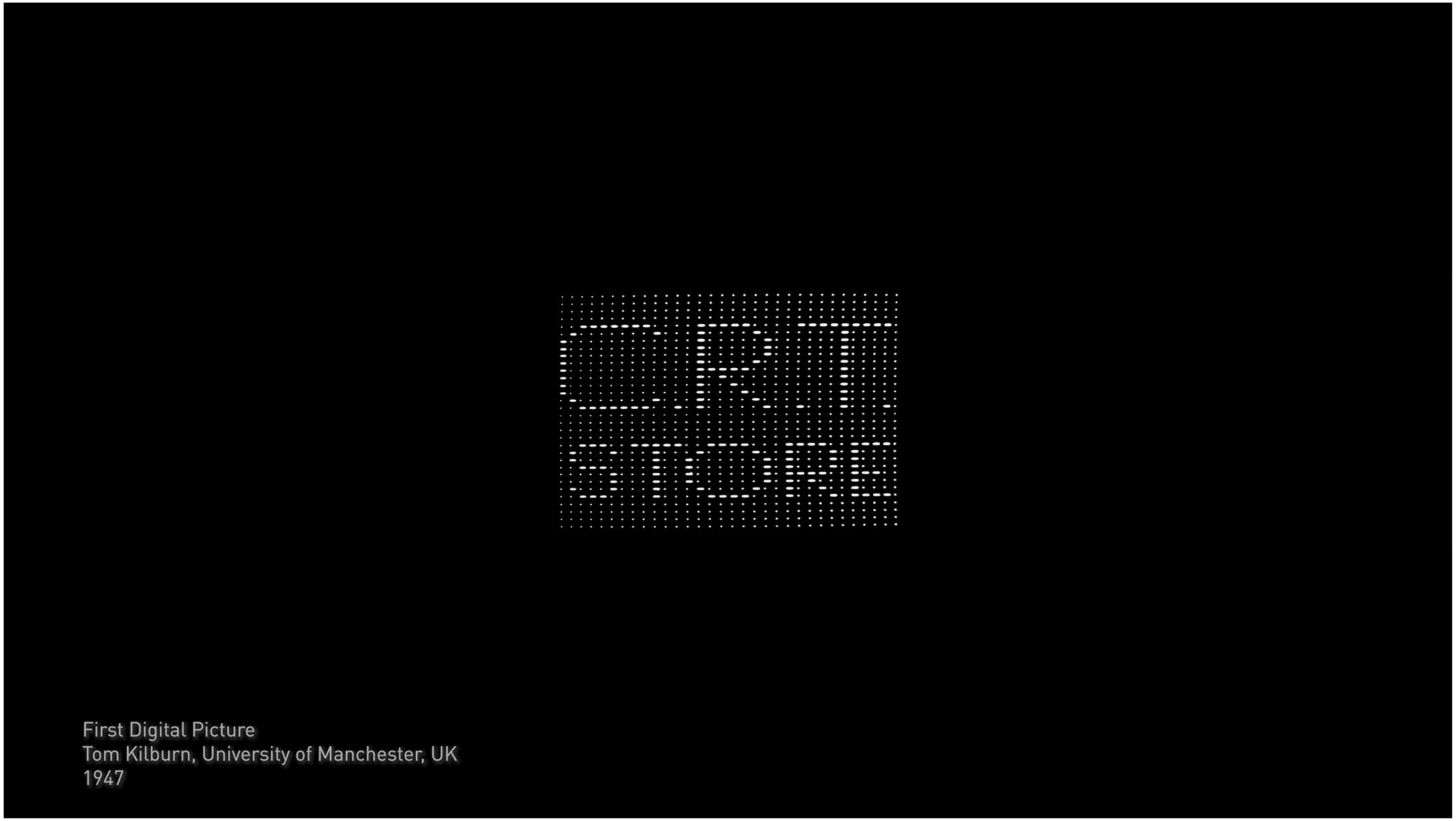
History of computer graphics

- **2010s**
  - CGI is nearly ubiquitous in video; pre-rendered graphics are nearly scientifically photorealistic
  - Real-time graphics on a suitably high-end system may simulate photorealism to the untrained eye
  - Advanced optics areas are put into rendering
  - Real time graphics at ultra-high-resolution modes began
  - Most animated movies are CGI now; a great many animated CGI films are made per year
  - Microsoft Xbox One, Sony Playstation 4, and Nintendo Switch currently dominate the home space
  - Windows PC is still one of the most active gaming platforms

History of computer graphics

- **2020s**
  - Advances in ray-tracing technology allowed it to be used for real-time rendering
  - AI-powered graphics for generating or upscaling frames.
  - Nvidia was the first to push for ray-tracing with ray-tracing cores
  - AI with DLSS and Tensor cores
  - Generative AI for graphics

History of computer graphics



First Digital Picture
Tom Kilburn, University of Manchester, UK
1947

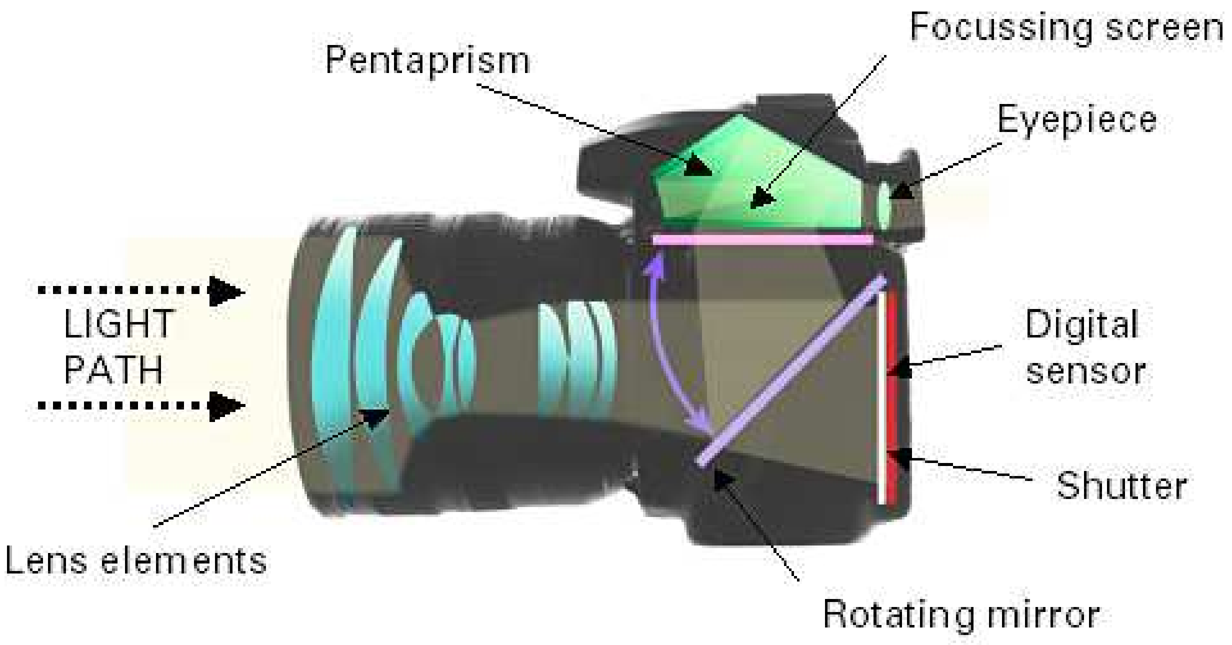
Digital Image and Imaging Basics

What is a (digital) image?

- **Image**
  - An artifact that depicts visual perception
- **Digital image**
  - A numeric representation of a (2D) image (rasterization)
  - Raster images have a finite set of picture elements or pixels

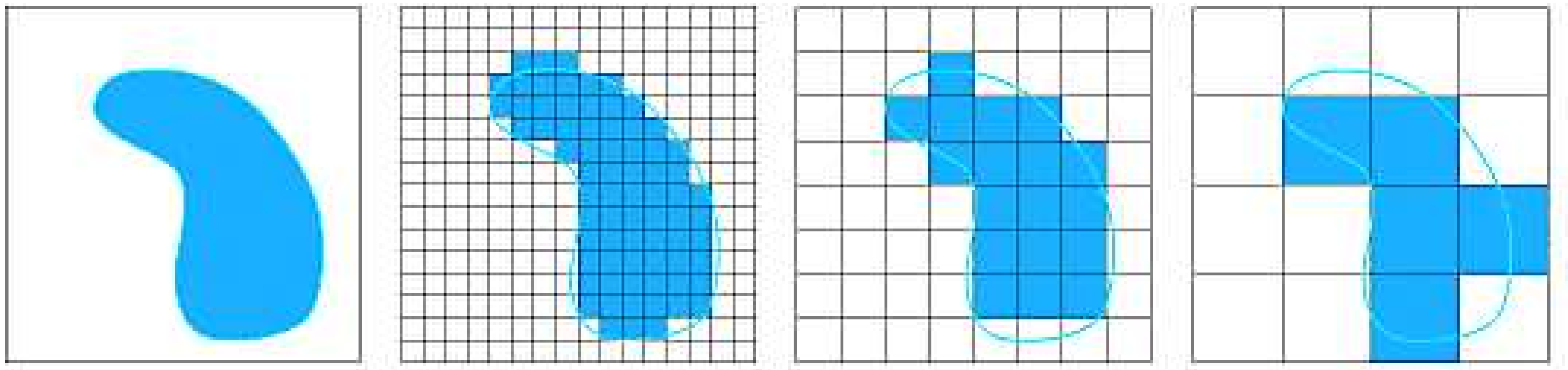
How to obtain a digital image?

- **Imaging device (camera)**



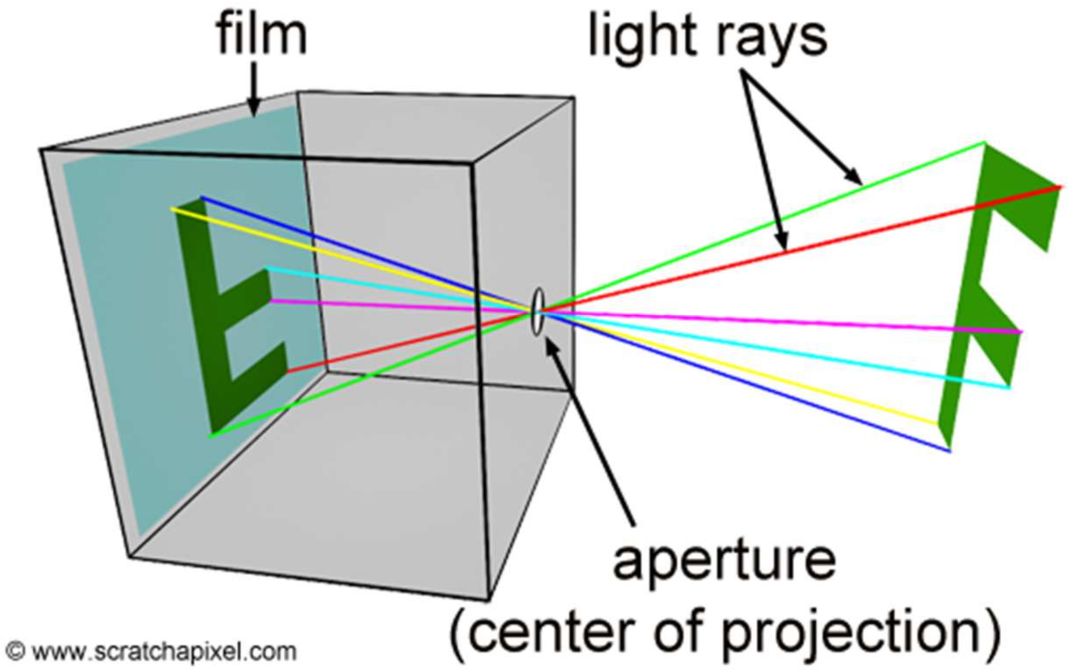
Rasterization

- **The goal:**
  - The task of taking an image described in continuous signals and converting it into a raster image (pixels)
  - Pixel: a cell of uniform grid over the imaging plane



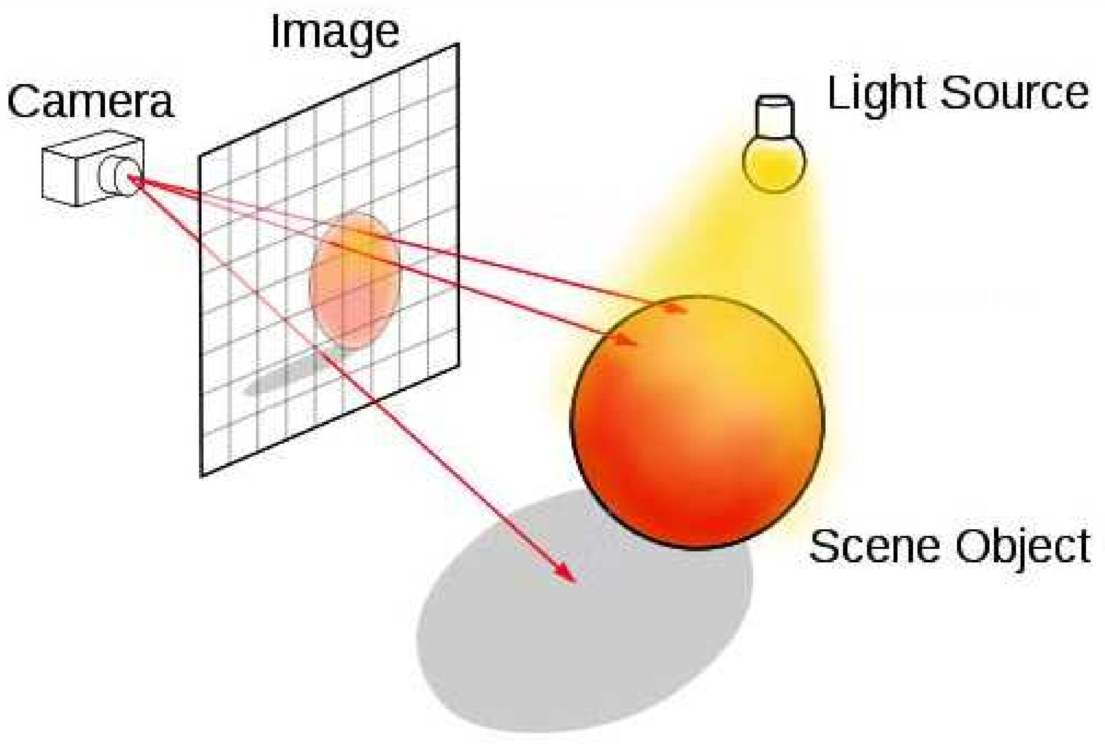
Camera model

- **Pin-hole camera**



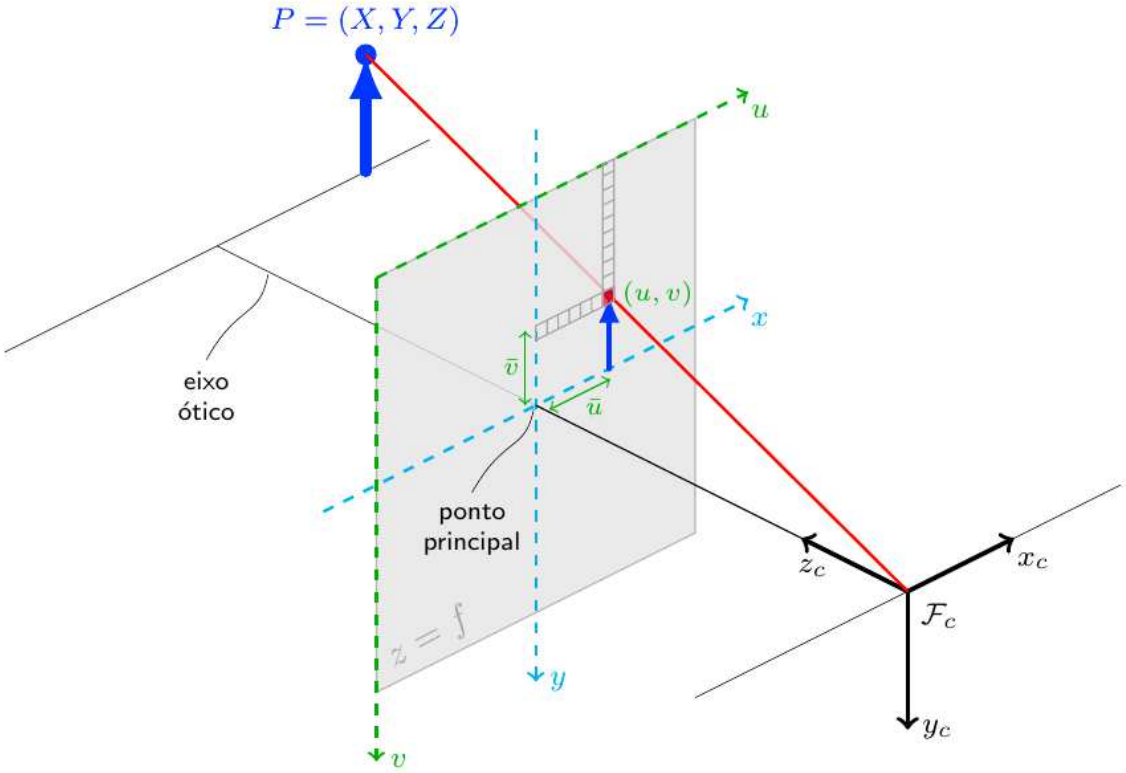
Camera model

- **Pin-hole camera model**



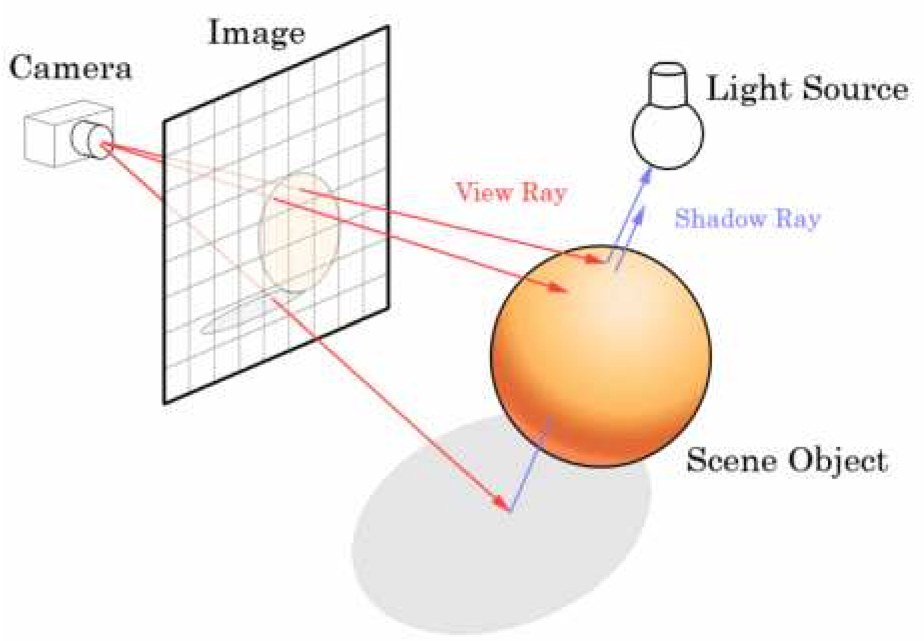
3D projection

- **Project 3D world coordinates onto a 2D image plane**
  - For 2D display purpose



Ray tracing

- **An image synthesis technique**
  - Shoot rays from camera pixels



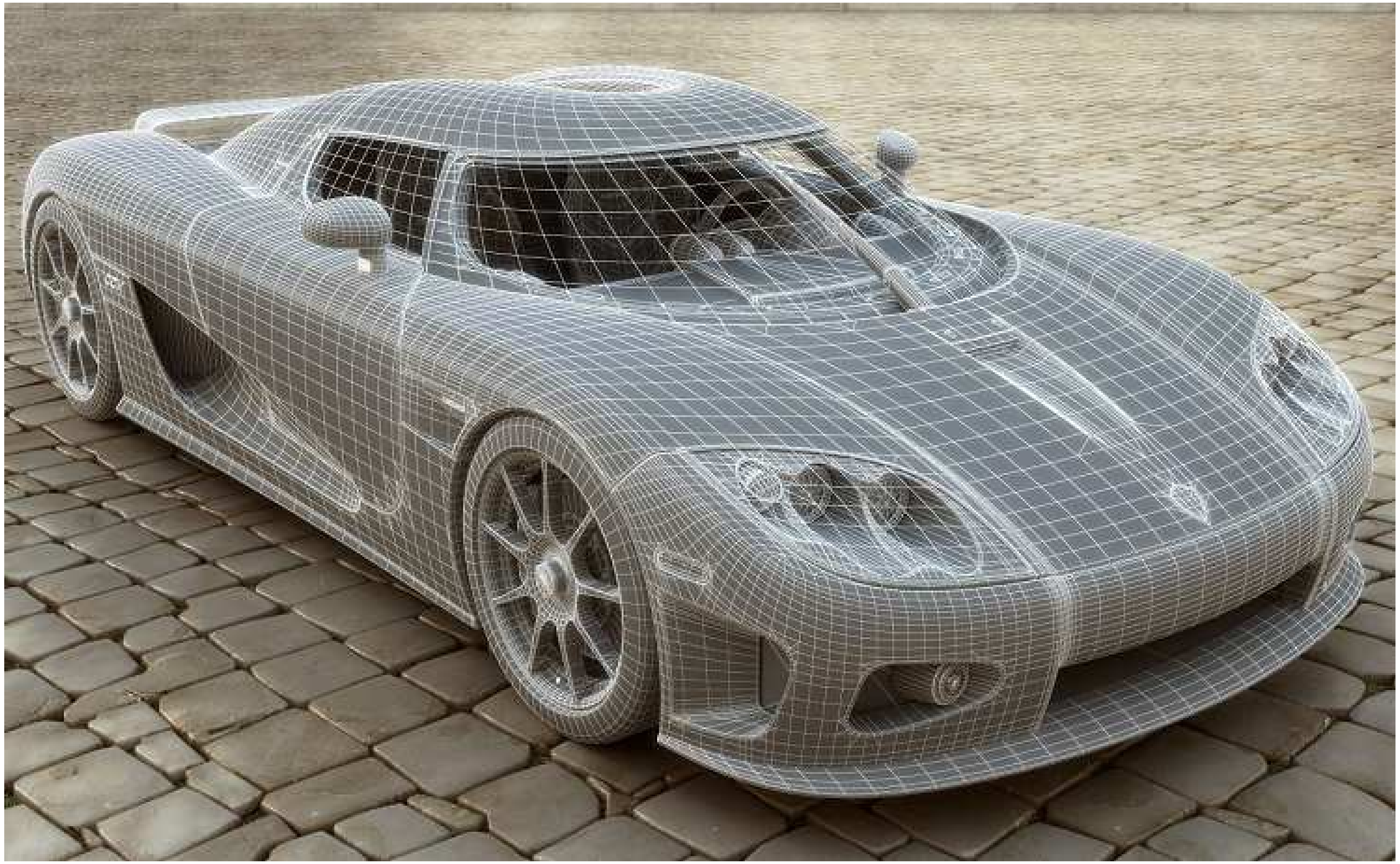
Geometric representation and modeling

Geometric representation

- **How objects are represented?**
  - **Continuous surface**
    - Pure mathematical representation
    - Bezier, B-Spline, NURBS surfaces
  - **Meshes**
    - A collection of triangles or quadrilaterals with connectivity
    - Discretized from continuous surfaces
    - Constructed from scanned data
      - Depth-sensing devices

Geometric representation

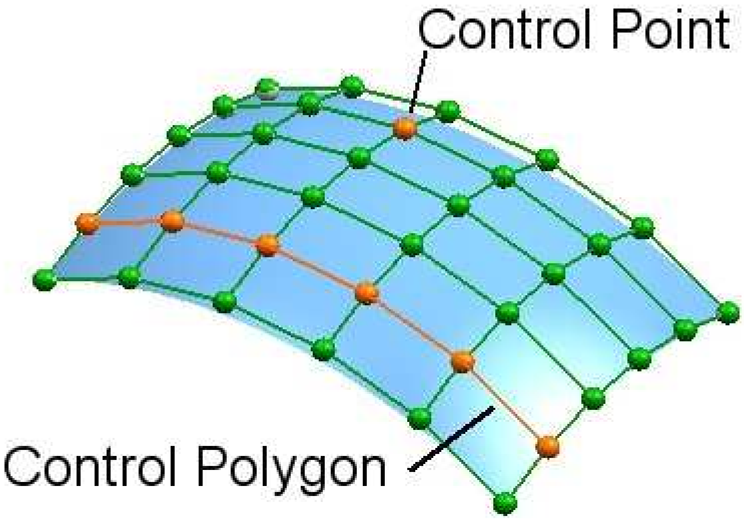
- **Geometric representation by meshes**

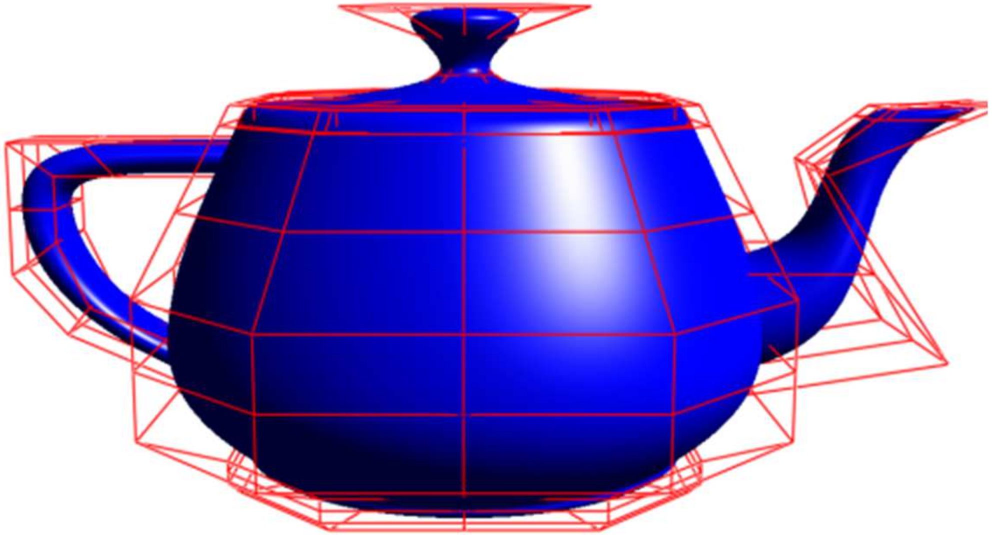


Geometry from free-form surfaces

- **Bezier surface**

$$\mathbf{p}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^n(u)\ B_j^m(v)\ \mathbf{k}_{i,j}$$

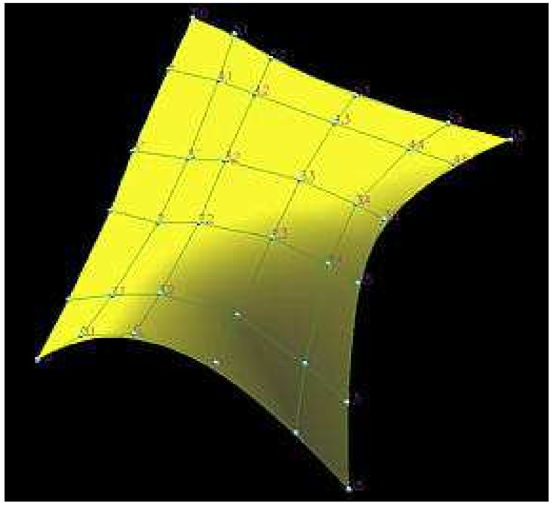


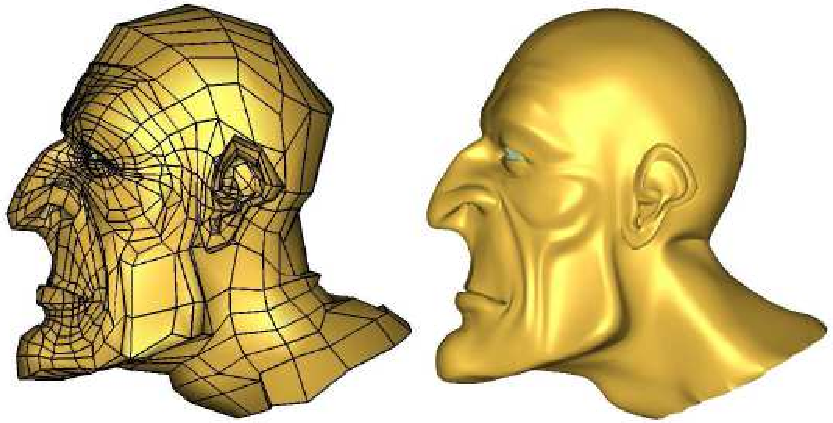


Geometry from free-form surfaces

- **B-spline surfaces**
  - Local shape adjustment

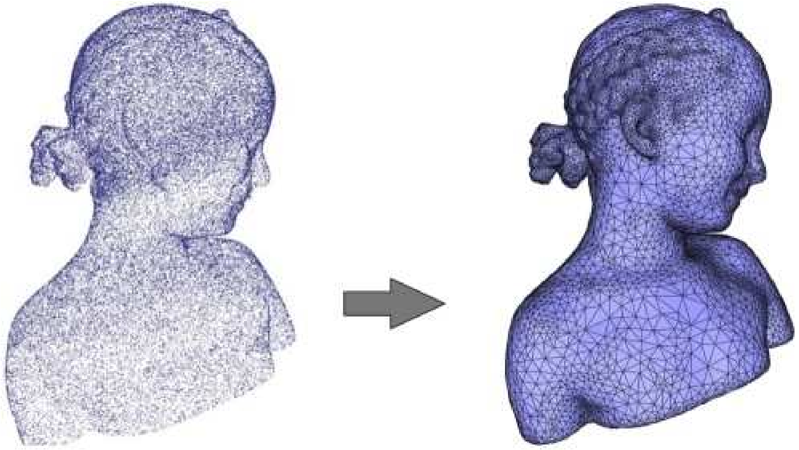
$$Q(u,w)=\sum_{i=1}^{n+1}\sum_{j=1}^{m+1}B_{i,j}N_{i,k}(u)M_{j,l}(w)$$

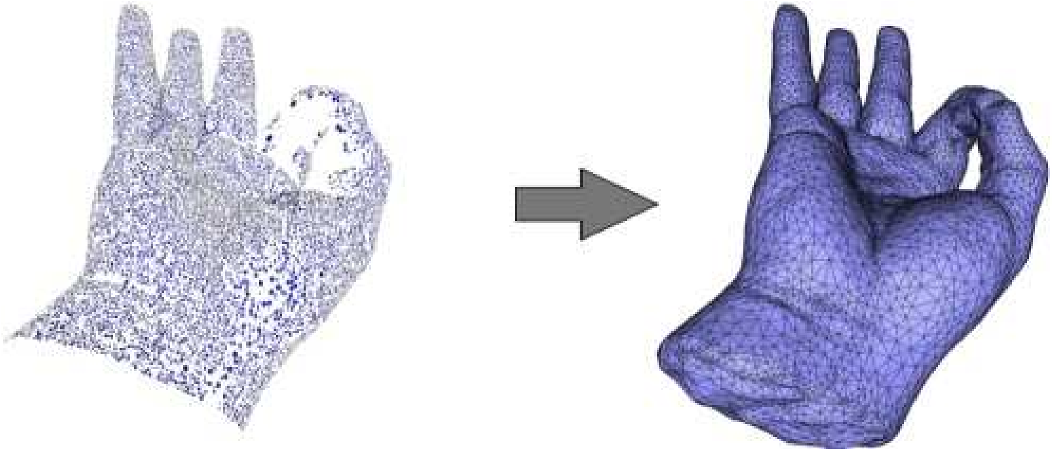




Geometry from point cloud

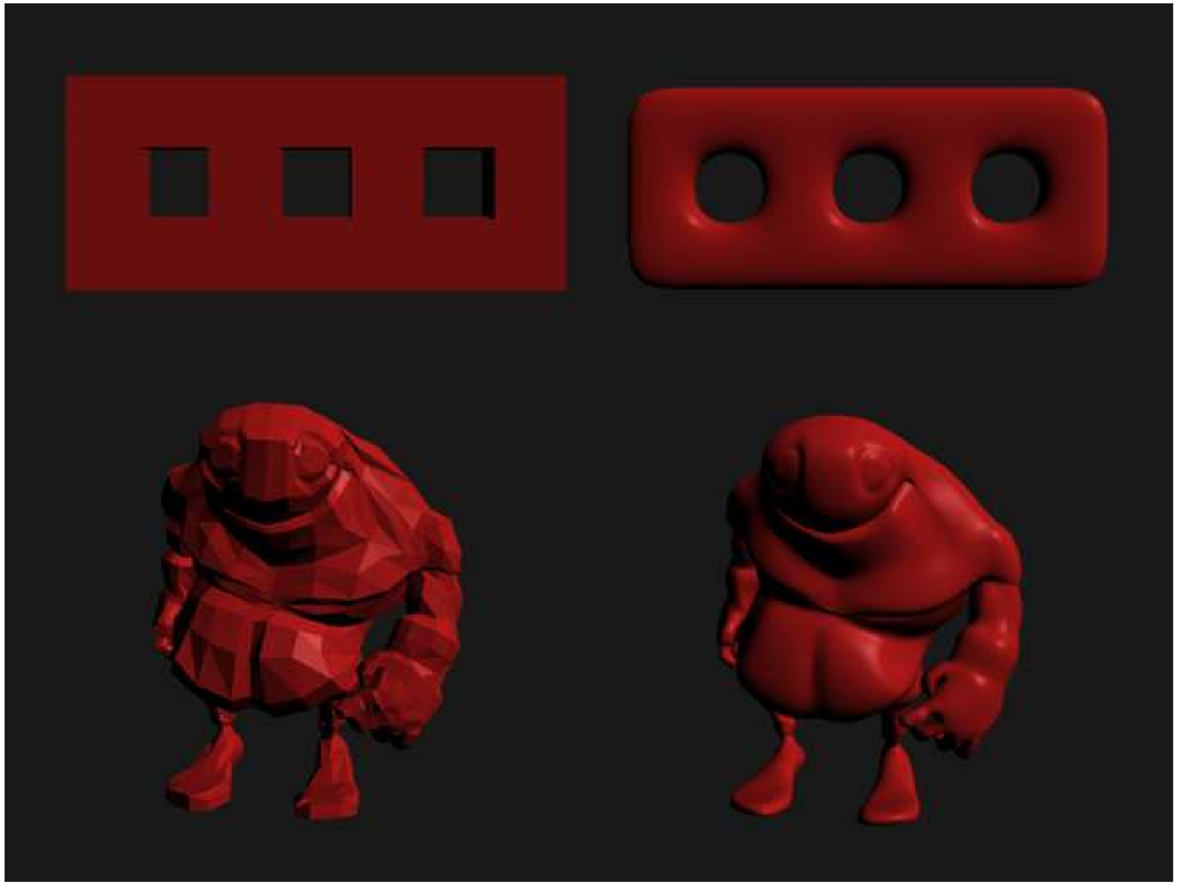
- **Mesh reconstruction**
  - Create meshes from scattered point clouds





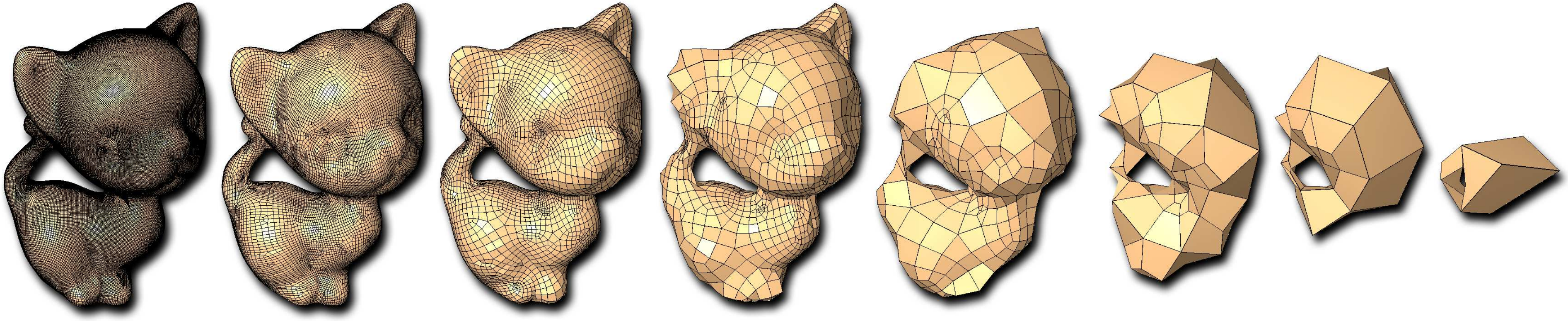
Mesh subdivision

- **Subdivide mesh faces progressively**
  - According to a specific division rule



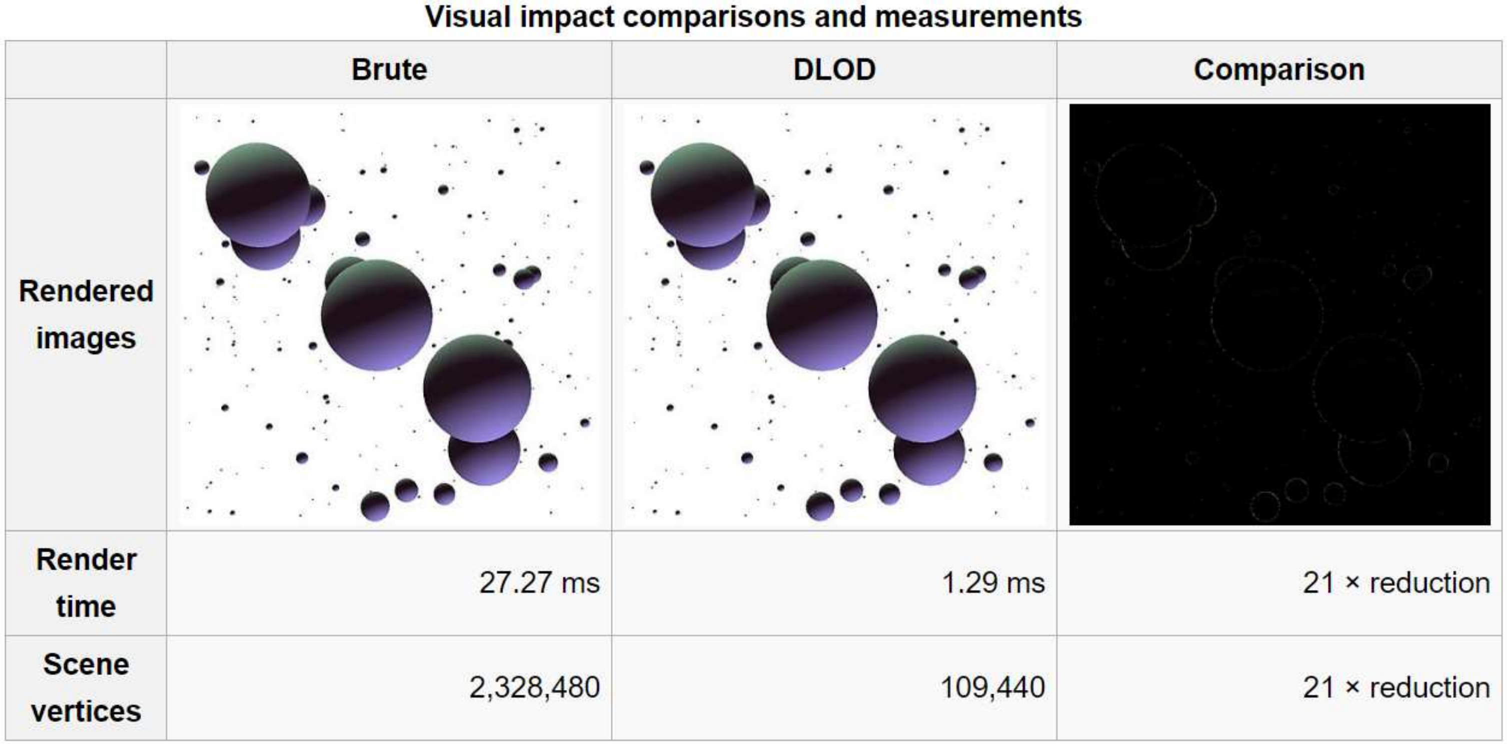
Mesh simplification

- **Inverse process of mesh subdivision**
  - Simplify a mesh representation with fewer vertices and faces



Level-of-detail (LoD)

- **Increase or decrease the complexity of a 3D model adaptively**
  - According to the viewing distance



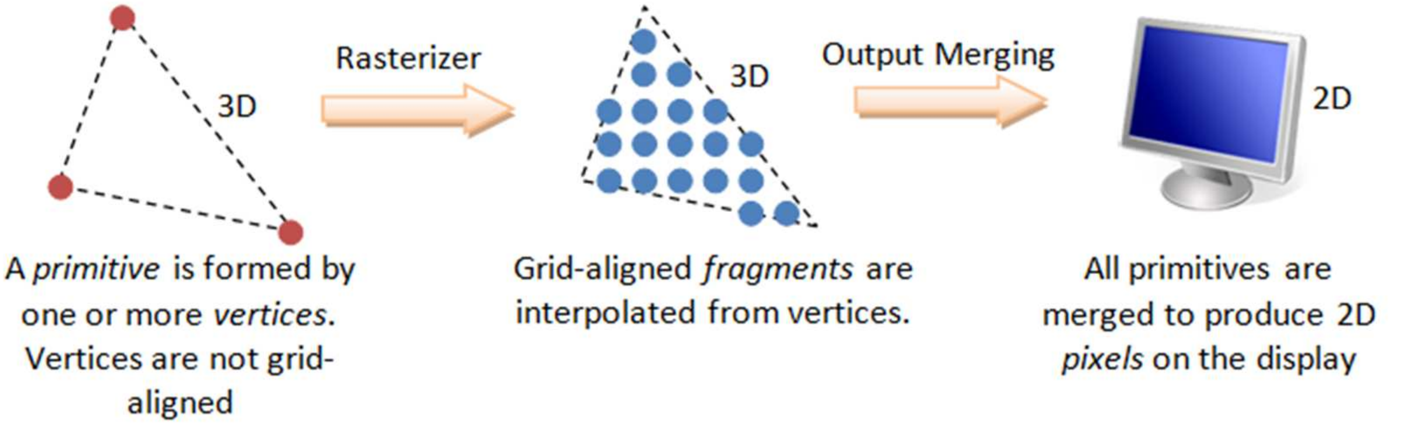
Visual impact comparisons and measurements

	Brute	DLOD	Comparison
Rendered images			
Render time	27.27 ms	1.29 ms	21 × reduction
Scene vertices	2,328,480	109,440	21 × reduction

Illumination and Rendering

Scanline rendering

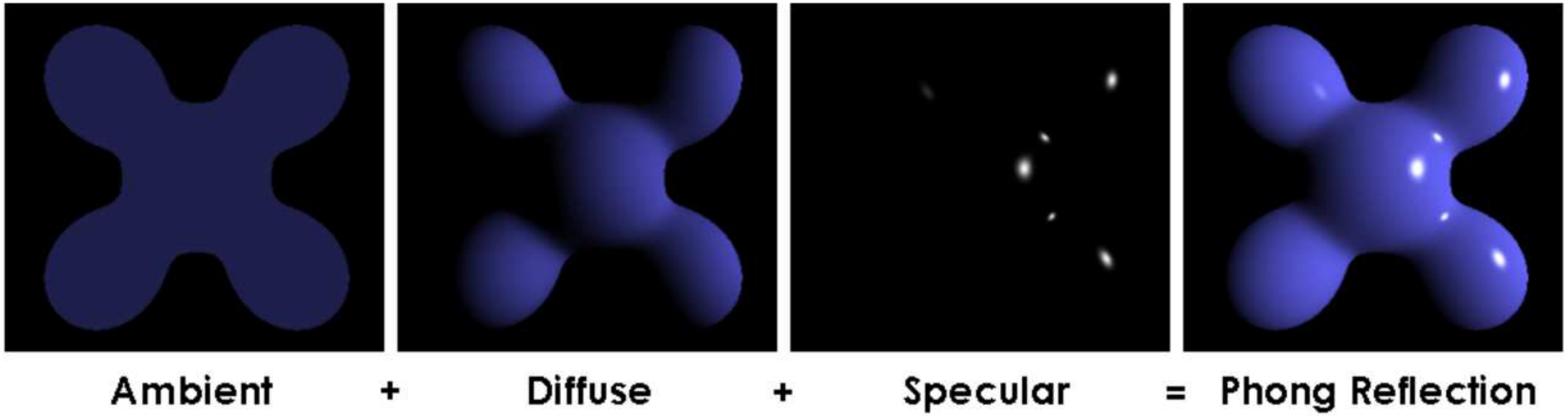
- **Projection-space rendering**
  - Geometries are first projected onto the image plane
  - A scanline algorithm is used to render the images


3D
Rasterizer
3D
Output Merging
2D
A *primitive* is formed by one or more *vertices*. Vertices are not grid-aligned
Grid-aligned *fragments* are interpolated from vertices.
All primitives are merged to produce 2D *pixels* on the display

Vertex, Primitives, Fragment and Pixel

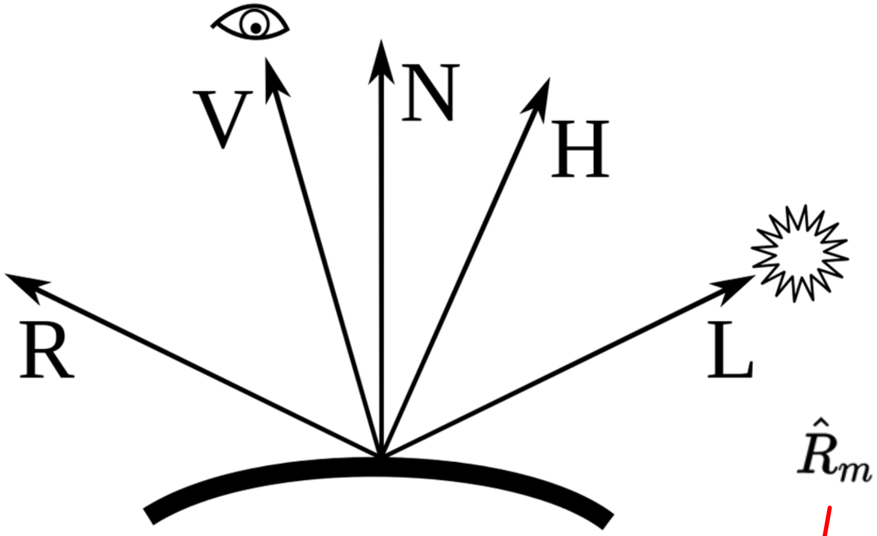
Reflection

- **Local illumination**
  - Phong model



Reflection

- **Local illumination**
  - Phong model



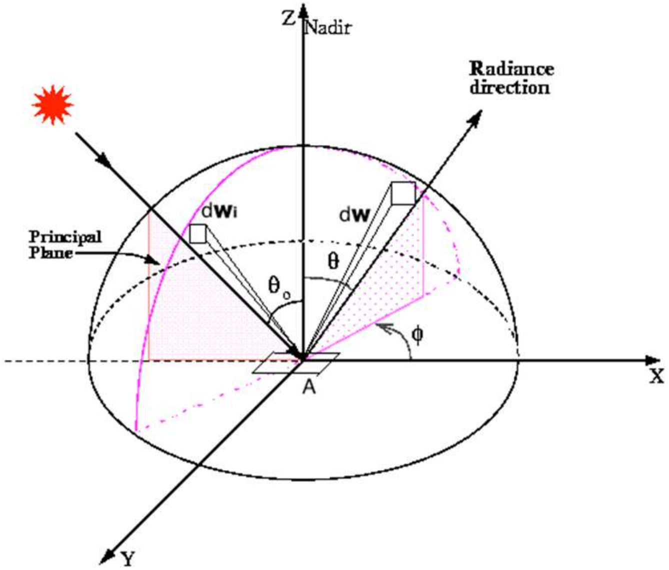
$$\hat{R}_m = 2(\hat{L}_m \cdot \hat{N})\hat{N} - \hat{L}_m$$

$$I_\mathrm{p} = k_\mathrm{a} i_\mathrm{a} + \sum_{m \,\in\, \mathrm{lights}} (k_\mathrm{d} (\hat{L}_m \cdot \hat{N}) i_{m,\mathrm{d}} + k_\mathrm{s} (\hat{R}_m \cdot \hat{V})^\alpha i_{m,\mathrm{s}})$$

Reflection

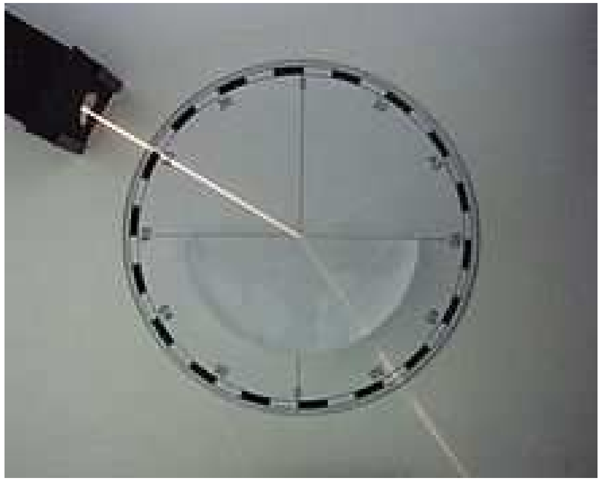
- **Local illumination**
  - Bidirectional reflectance distribution function (BRDF)

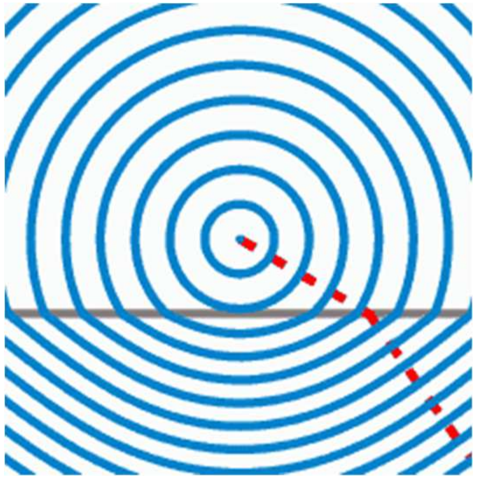
$$f_r(\omega_i,\, \omega_r) = \frac{\mathrm{d}\, L_r(\omega_r)}{\mathrm{d}\, E_i(\omega_i)} = \frac{\mathrm{d}\, L_r(\omega_r)}{L_i(\omega_i) \cos\theta_i \ \mathrm{d}\, \omega_i}$$



Refraction

- **Refraction**



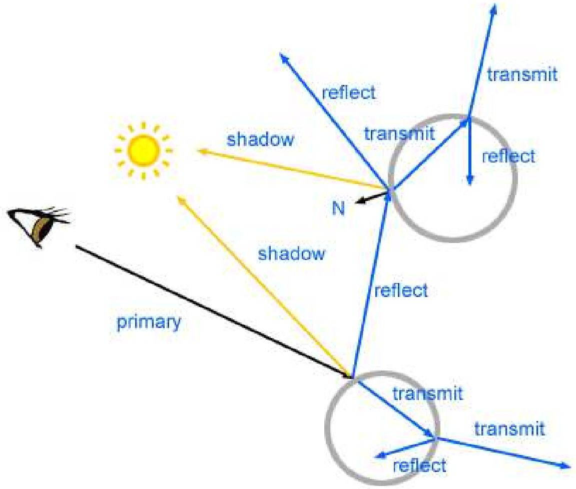


  - Fresnel's law
    - The amount of reflection vs. refraction
  - Snell's law

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{v_1}{v_2} = \frac{\lambda_1}{\lambda_2} = \frac{n_2}{n_1}$$

Refraction

- **Refraction**
  - Trace rays through transparent surfaces
    - Both reflective and refractive rays



Scattering

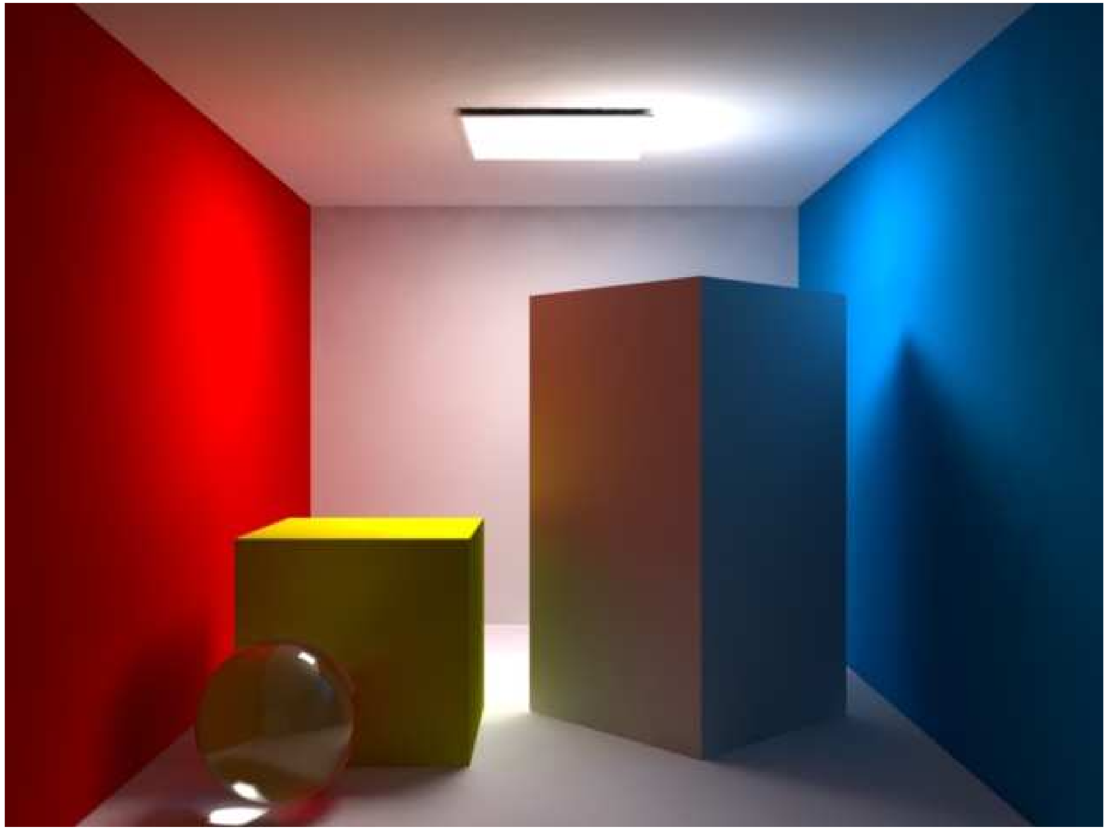
- **Subsurface scattering**
  - A mechanism of light transport in which light penetrates the surface of a translucent object
  - The realistic rendering of materials such as marble, skin, leaves, wax and milk

Scattering

- **Subsurface scattering**
  - Example: sub-surface scattering of light through hand

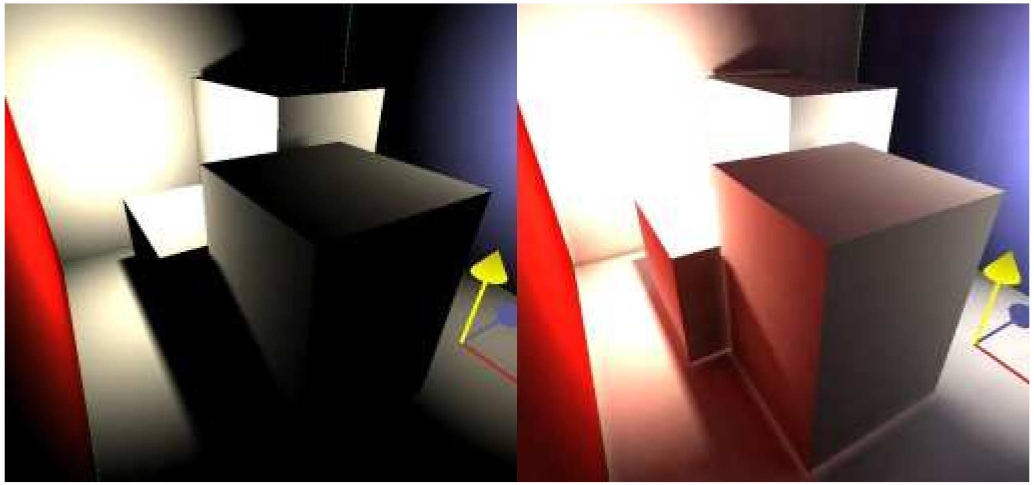
Global illumination

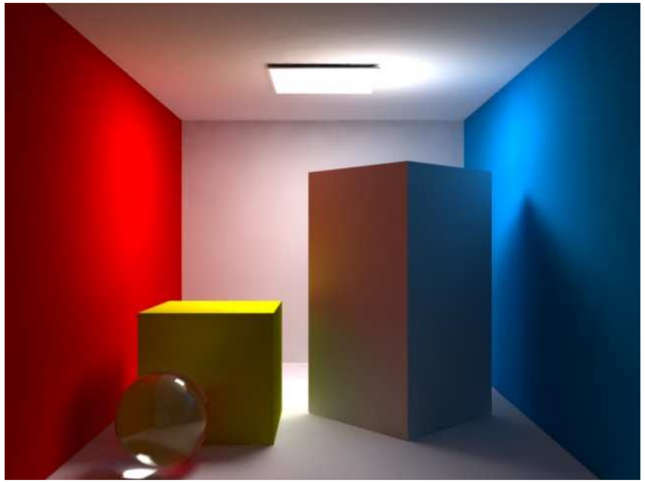
- **Global illumination**
  - Inter-reflections, soft shadow, caustics, etc.
  - Environment lighting



Global illumination

- **Examples**





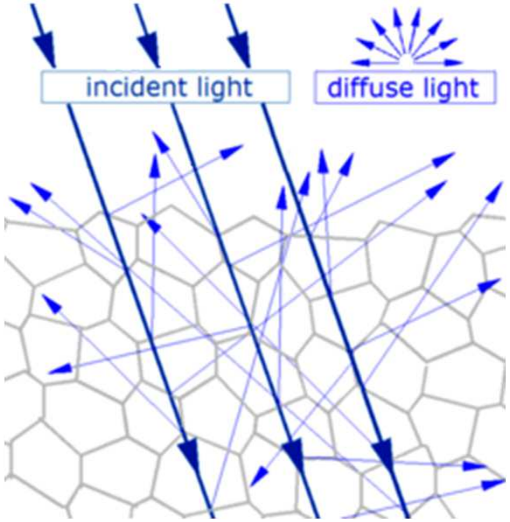
Volumetric illumination

- **Volumetric illumination effects**
  - Light coming from a whole volume
  - A certain degree of transparency

Courtesy of Mr. X

Volumetric scattering

- **Volumetric light scattering**
  - Can be thought of as the deflection of a ray from a straight path



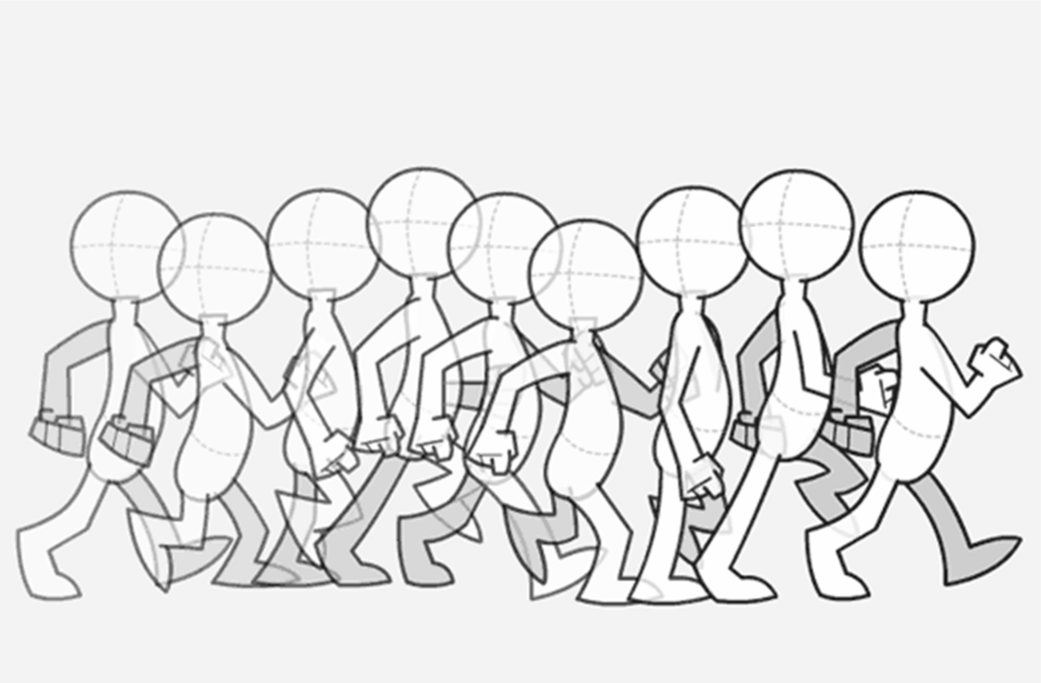
Computer Animation

Computer animation

- **The process used for generating animated images**
  - Key-framing with interpolation
  - Warping
  - Motion capture
  - Physically-based modeling
- **Pixar Animation Studios**

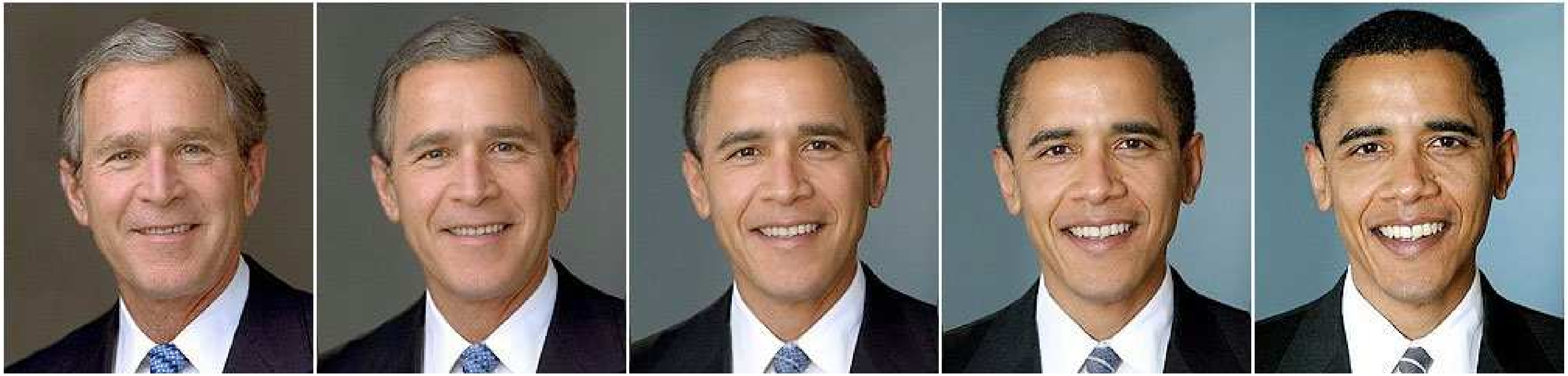
Key-framing

- **Smooth transition**
  - – Define the starting and ending points
  - – A sequence of key frames define the movement



Morphing

- **A special effect in motion pictures**
  - Most often used to depict one person turning into another object
  - Feature matching with image warping/blending

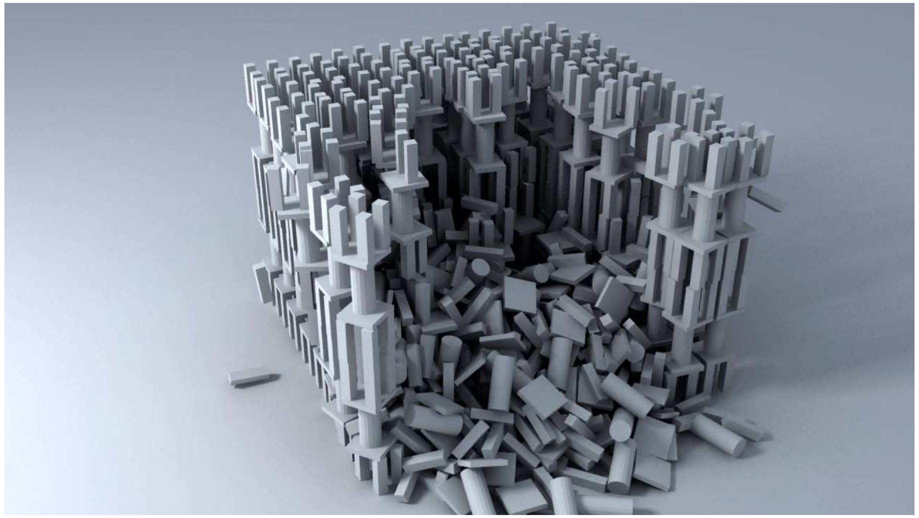


Motion capture

- **The process of recording movement**
  - Based on feature points
  - Based on mesh deformation

Rigid body animation

- **Simulation for rigid body dynamics**
  - Rendering with global illumination

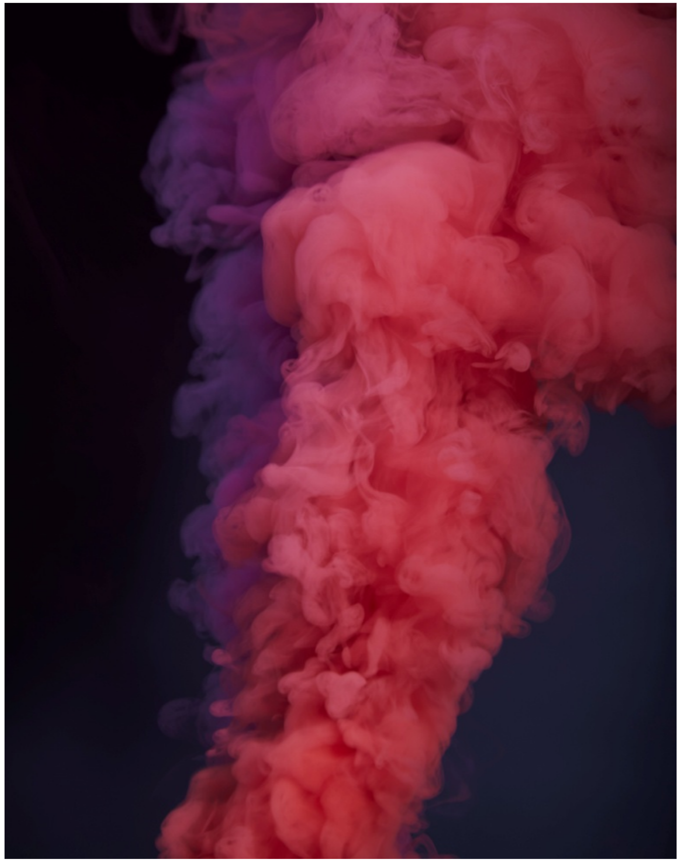


Soft-body animation

- **Simulation of soft-body**
  - Cloth, hair, paper, etc.

Flow animation

- **Simulation and rendering of flow phenomena**



Parallel computing in computer graphics

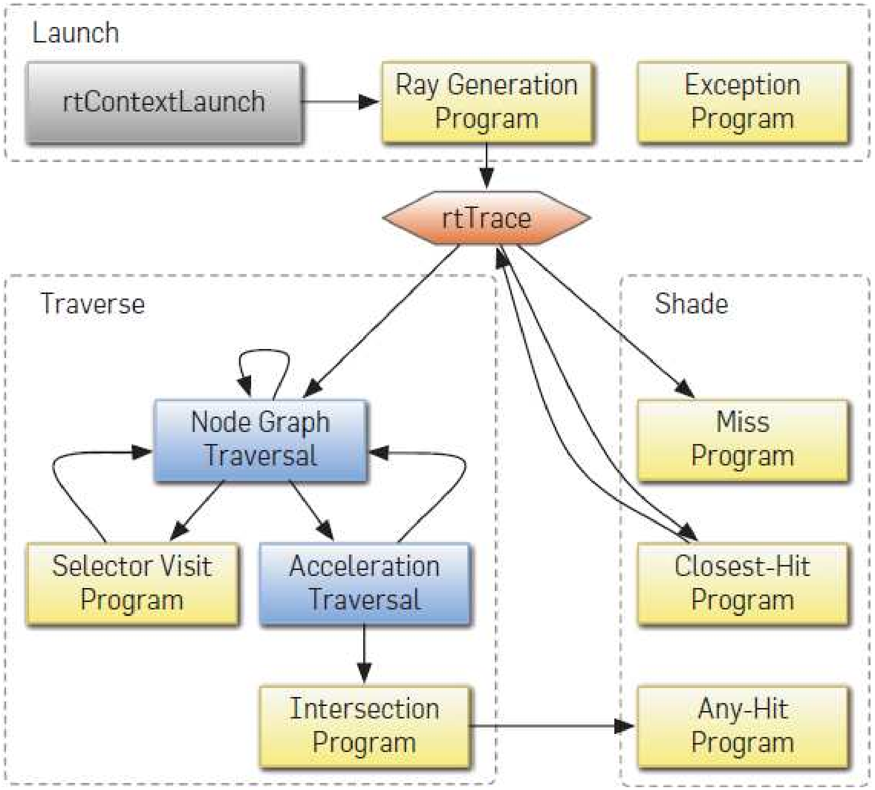
- **Many algorithms in graphics rendering can be parallelizable**
  - Vertex processing
  - Projection
  - Rasterization
  - Pixel (fragment) processing
  - Texturing
  - etc.

Graphics card

- **Graphics processing unit (GPU)**
  - A specialized electronic circuit designed to accelerate the creation of images in a framebuffer
  - Parallel processing for graphics operations
    - Vertex, projection, rasterization, pixel fill, etc.

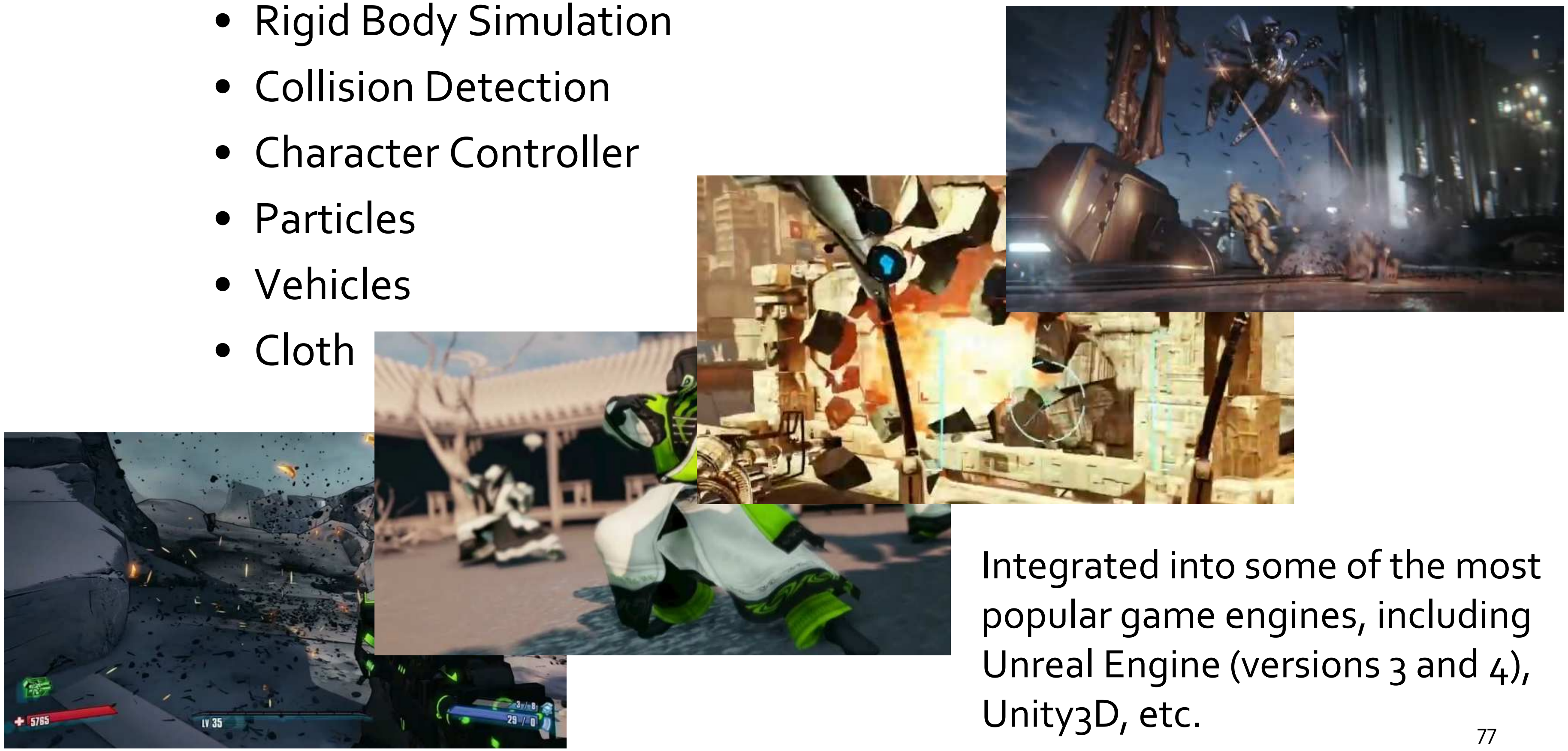
CUDA programming

- **CUDA for computer graphics (rendering)**
  - Better integrated with GPU
  - OptiX – CUDA-based programmable ray-tracing engine



CUDA programming

- **CUDA for computer graphics (physical simulation)**
  - PhysX
    - Rigid Body Simulation
    - Collision Detection
    - Character Controller
    - Particles
    - Vehicles
    - Cloth



Integrated into some of the most popular game engines, including Unreal Engine (versions 3 and 4), Unity3D, etc.

Next Lecture :

The first graphics program