

# *An Introduction to Computer Graphics*

---

Joaquim Madeira

March 2022

# Overview

- What is CG ?
- Application areas
- Evolution and trends
- Main tasks
- CG APIs

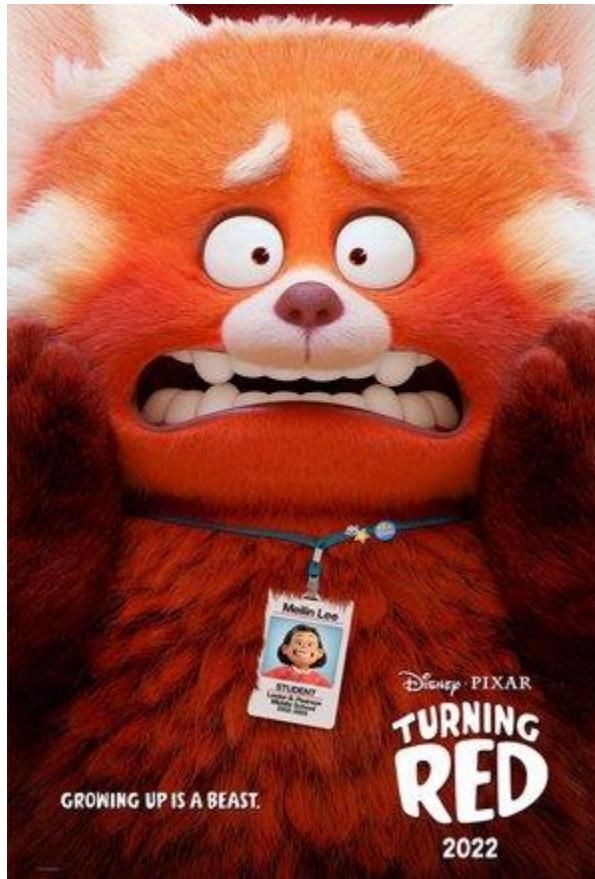


---

# **PIXAR'S COMPUTER-ANIMATED FILMS**

---

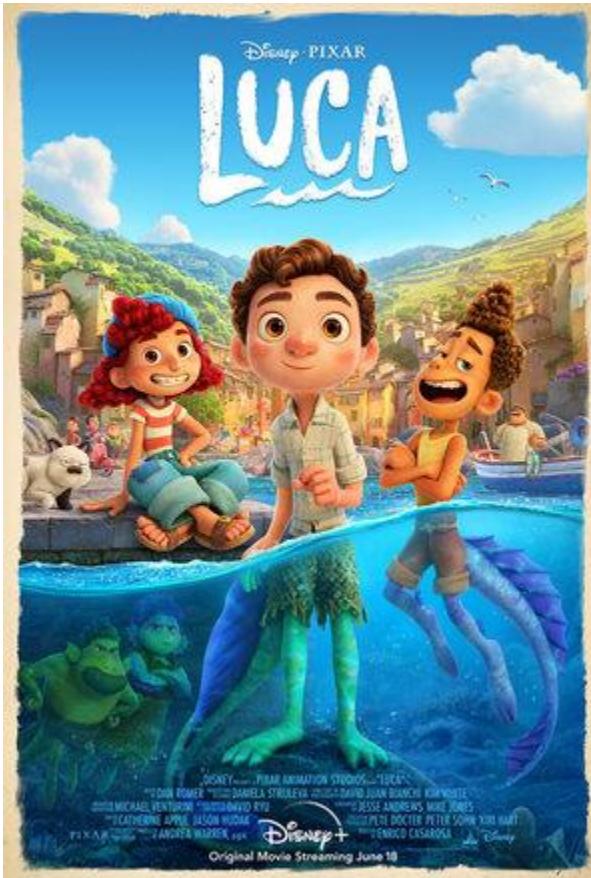
# Turning Red (2022)



Trailer at  
YouTube

- Pixar's latest computer-animated film

# Luca (2021)



Trailer at  
YouTube

- Pixar's 2021 computer-animated film

# Soul (2020)



[Trailer at  
YouTube](#)

- Pixar's 2020 computer-animated film

# Onward (2020)



[Trailer at  
YouTube](#)

- Pixar's 2020 computer-animated film

# Toy Story 4 (2019)



Trailer at  
YouTube

- Pixar's 2019 computer-animated film

# Incredibles 2 (2018)



Trailer at  
YouTube

- Pixar's 2018 computer-animated film

# Toy Story (1995)



[Trailer at  
YouTube](#)

- Pixar's first **entirely** computer-animated feature film

# Pixar's Luxo Jr. (1986)



[Trailer at YouTube](#)

- Pixar's second computer-animated short-film

# **WHAT IS COMPUTER GRAPHICS?**

# What is Computer Graphics?

- The technology with which **pictures**, in the broadest sense of the word, are
  - Captured or generated, and presented
  - Manipulated and / or processed
  - Merged with other, non-graphical application data
- It includes:
  - Integration with other kinds of data – **Multimedia**
  - Advanced dialogue and **interactive technologies**

[CG Topics – Darmstadt]

# What is Computer Graphics?

- Computer graphics generally means **creation**, **storage** and **manipulation** of **models** and **images**
- Such models come from a diverse and expanding set of fields
  - Physical, biological, mathematical, artistic, and conceptual / abstract structures

[Andy van Dam]

# What is Computer Graphics?

- Computer Graphics deals with all aspects of **creating images with a computer**
  - Hardware
  - Software
  - Applications
- How was this image produced?



[Angel]

# APPLICATION AREAS

# CG – Application areas

- Entertainment
  - Computer games
  - Animation films
  - Special effects
- Engineering / Architecture
  - Computer-Aided Design (CAD)
  - Data Visualization
- Medicine
  - Visualization
- ...

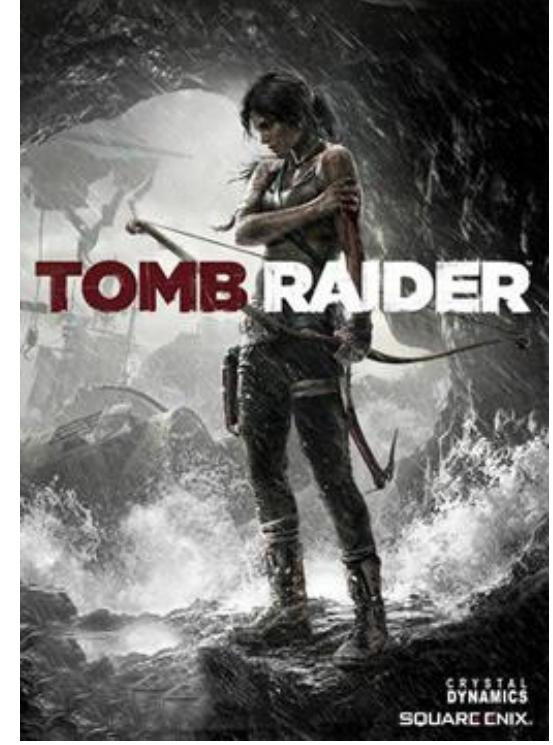
# Games – *Lara Croft*



1996



2007



2013

[Wikipedia]

# The Games With The Most Impressive Graphics, Ranked

Graphics can make a good game look great. Regardless of the actual gameplay, here are the games with the most undeniably impressive graphics

BY JAMES CARR

UPDATED FEB 05, 2022



<https://gamerant.com/video-games-best-graphics/>

# 10 Older Video Games That Still Boast Amazing Graphics

Despite how old some of these classic games are, their graphics still stand the test of time and were amazing in their day. These are the standouts.

BY DEREK DRAVEN

PUBLISHED APR 22, 2021



<https://screenrant.com/classic-older-video-games-that-best-graphics/>

# 10 Games With The Best Graphics Ever (At Their Time Of Release)

Gorgeous graphics tend to make an average game ten times better, and these games in particular wowed players with incredible visuals.

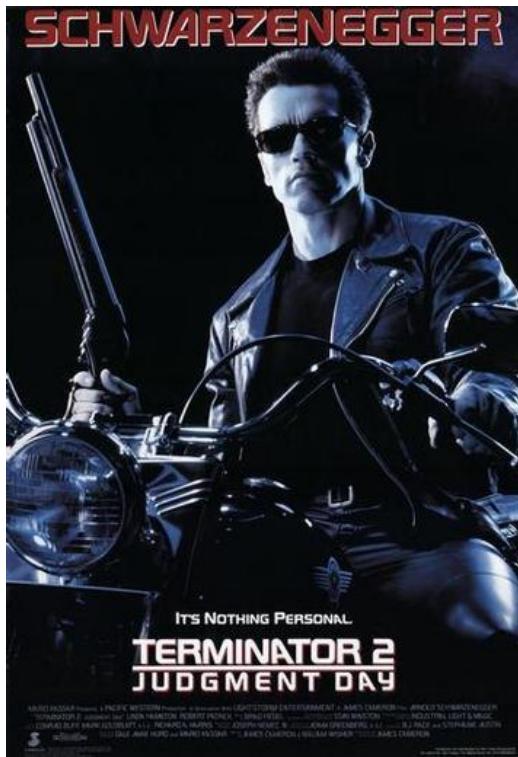
BY GAVIN FRASER MACKENZIE

PUBLISHED SEP 07, 2021



<https://gamerant.com/best-game-graphics-at-release/>

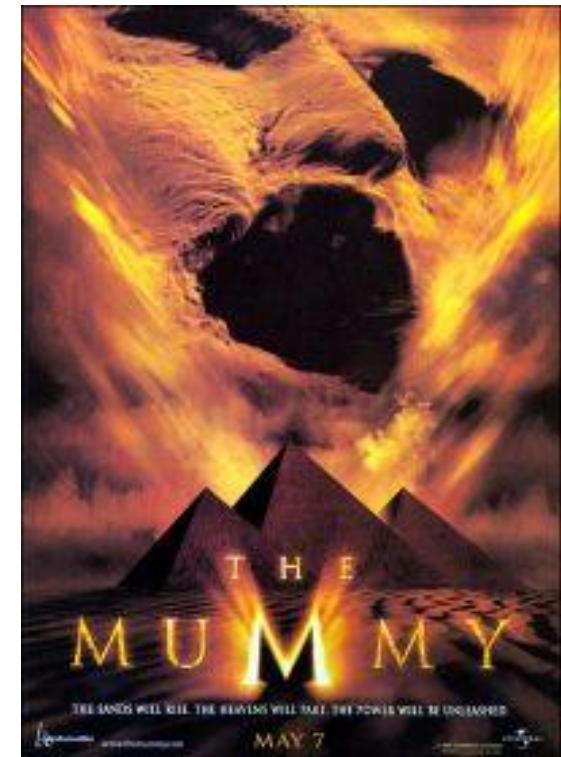
# Special effects – ILM



1991



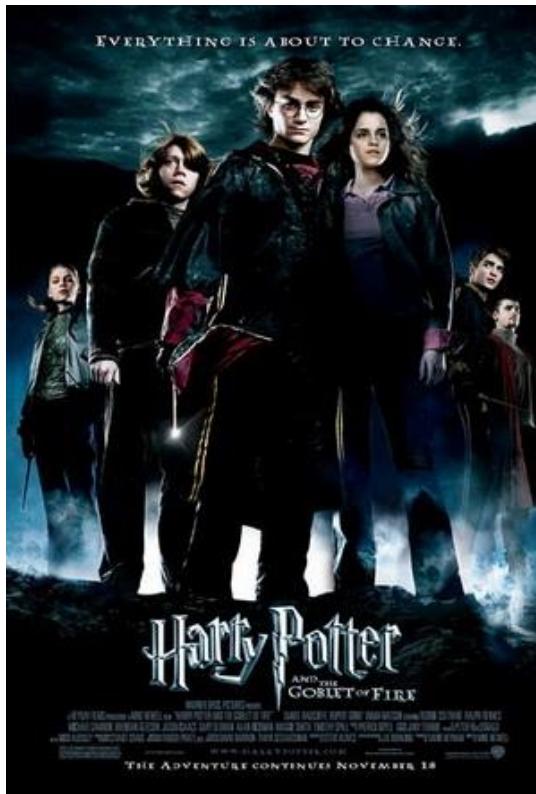
1994



1999

[Wikipedia]

# Special effects – ILM



2005



2009

[Wikipedia]



2013

# Special effects – ILM



2015



2016

[Wikipedia]



2017

# Special effects – ILM



2018



2019



2019

[Wikipedia]

# 14 groundbreaking movies that took special effects to new levels

Lucien Formichella Jan 11, 2020, 6:30 PM



<https://www.insider.com/most-groundbreaking-cgi-movies-ever-created-2020-1>

# 35 greatest CGI movie moments of all time

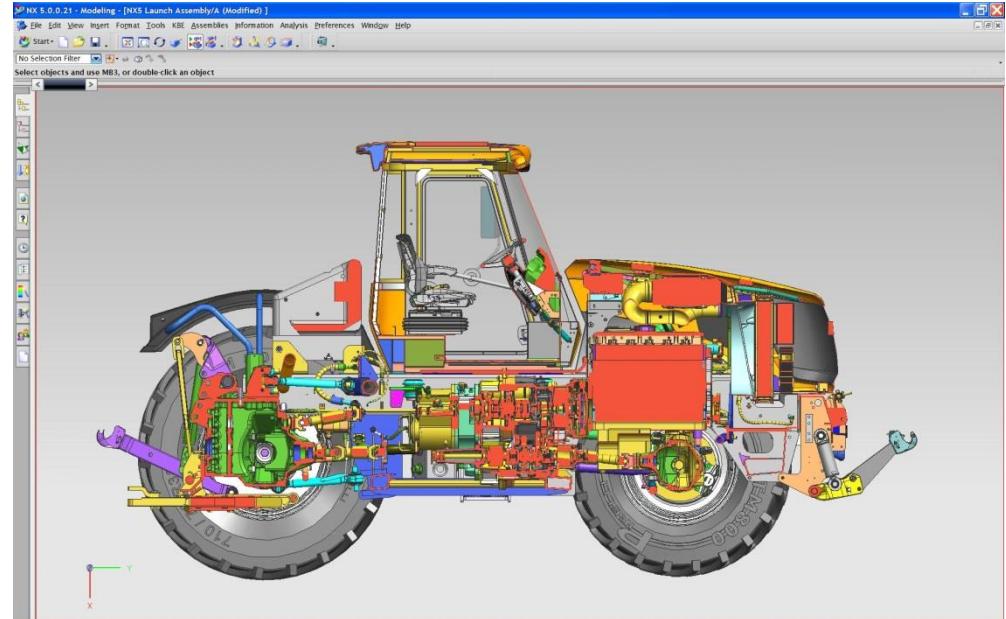
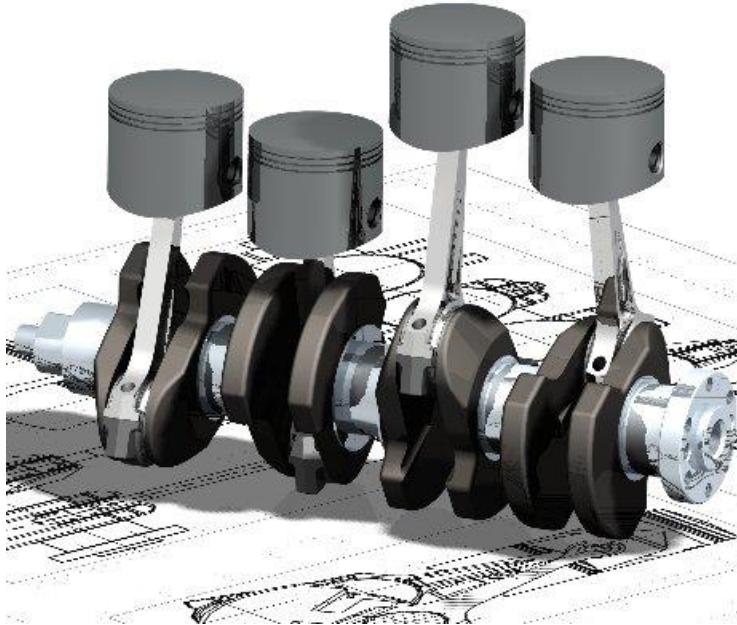
By Creative Bloq Staff published August 01, 2019

Our pick of the best CGI movie moments in live action films.



<https://www.creativebloq.com/3d-tips/cgi-movie-moments-1234014>

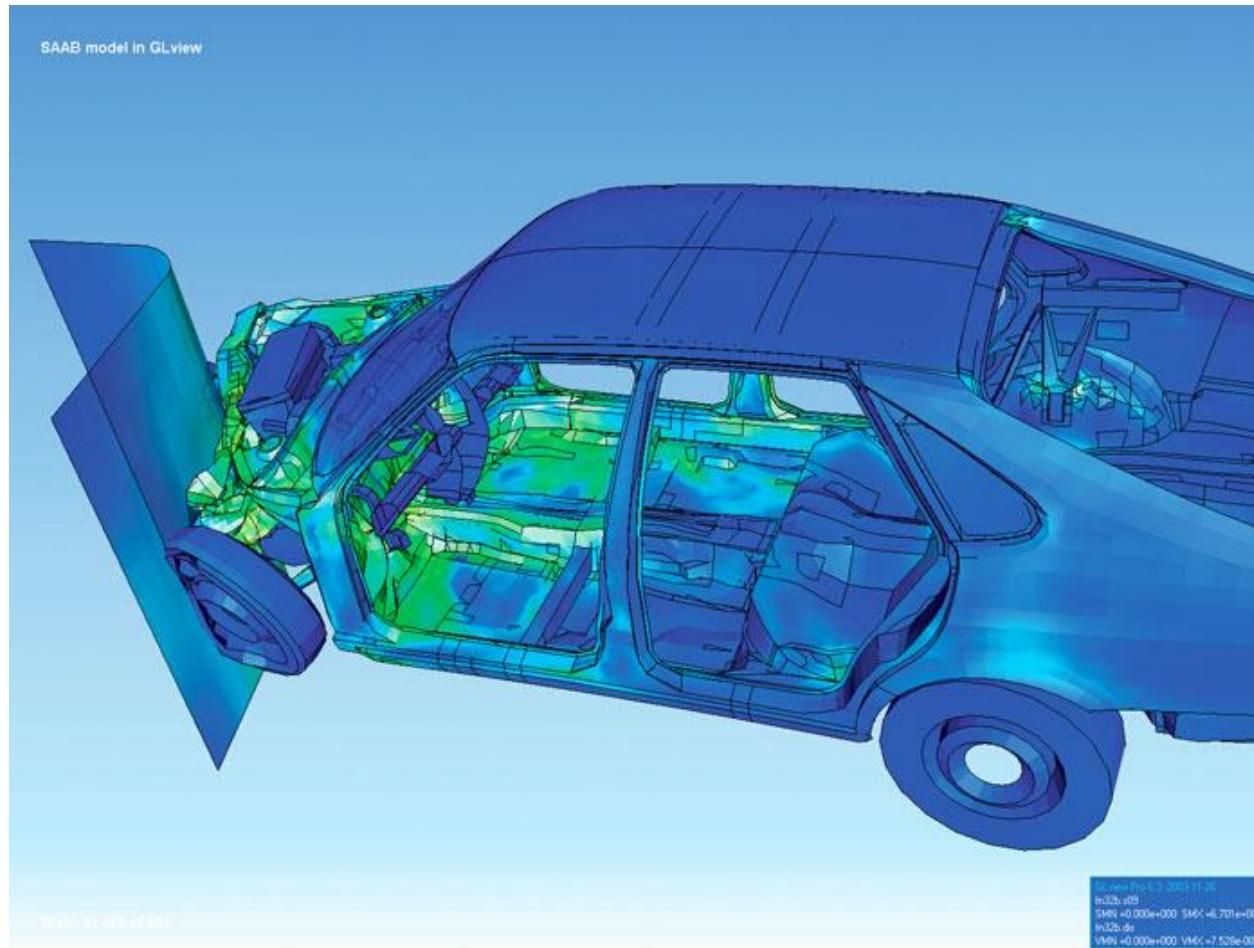
# Computer-Aided Design



CAD mockup

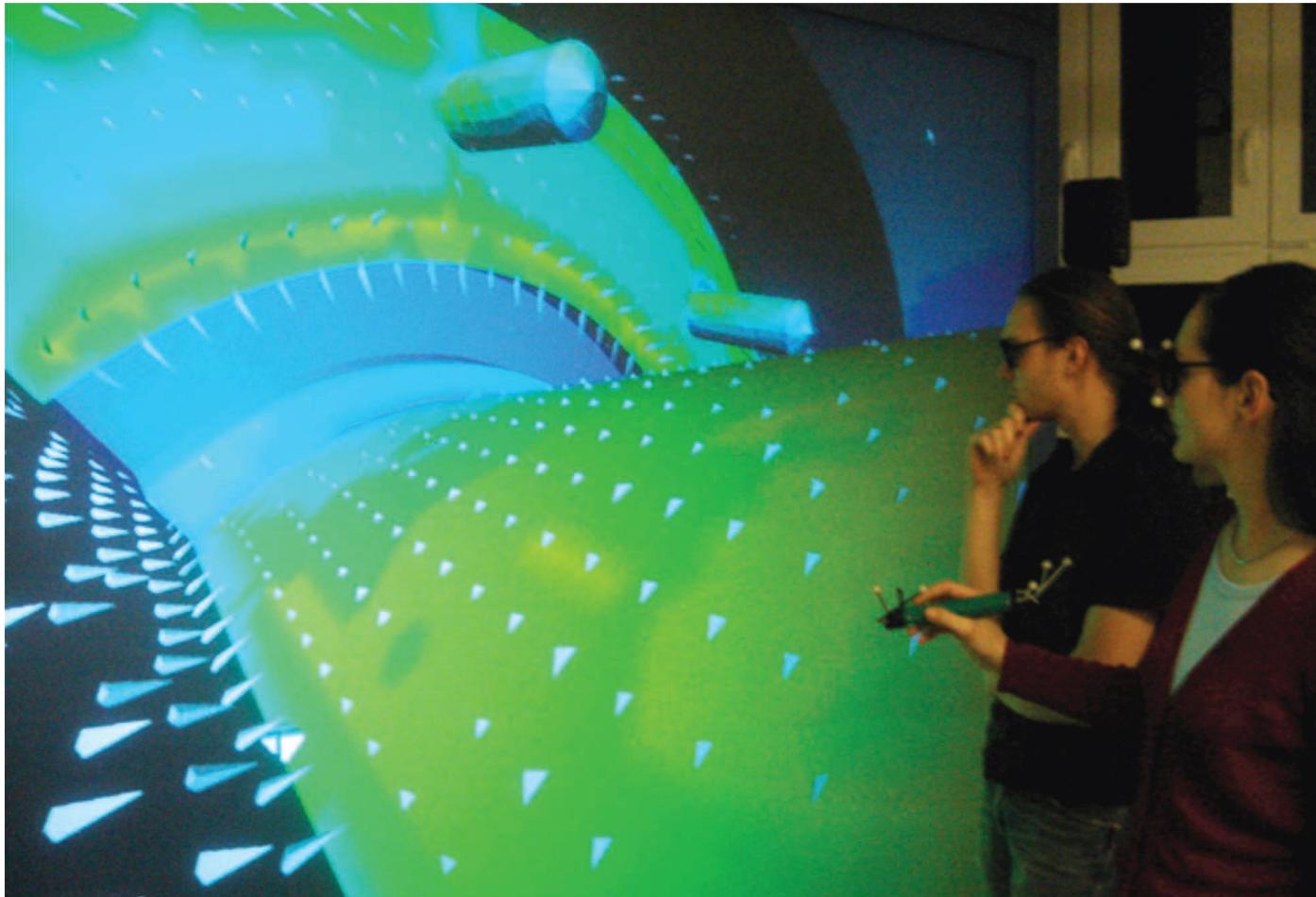
[Wikipedia]

# CAD – Simulation



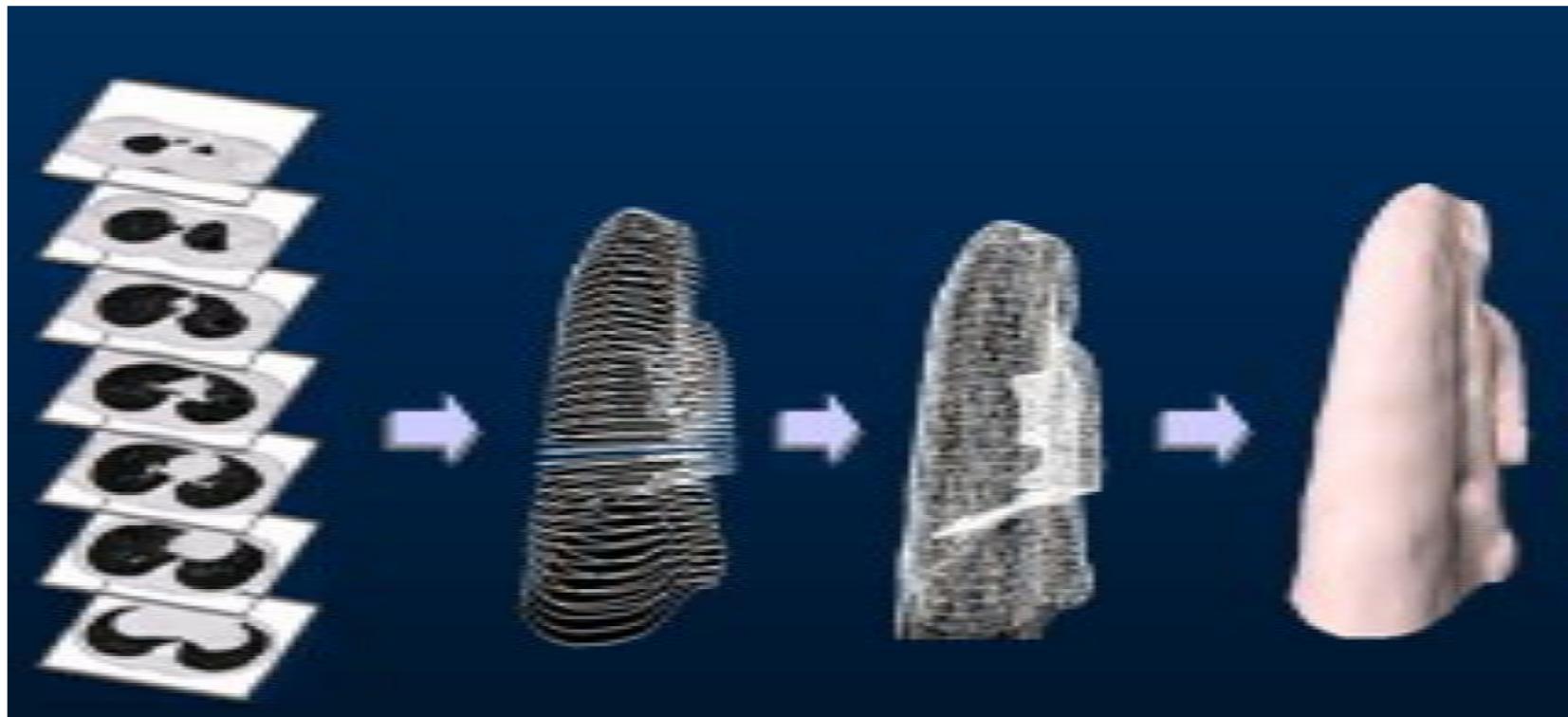
[Wikipedia]

# VR / AR Visualization



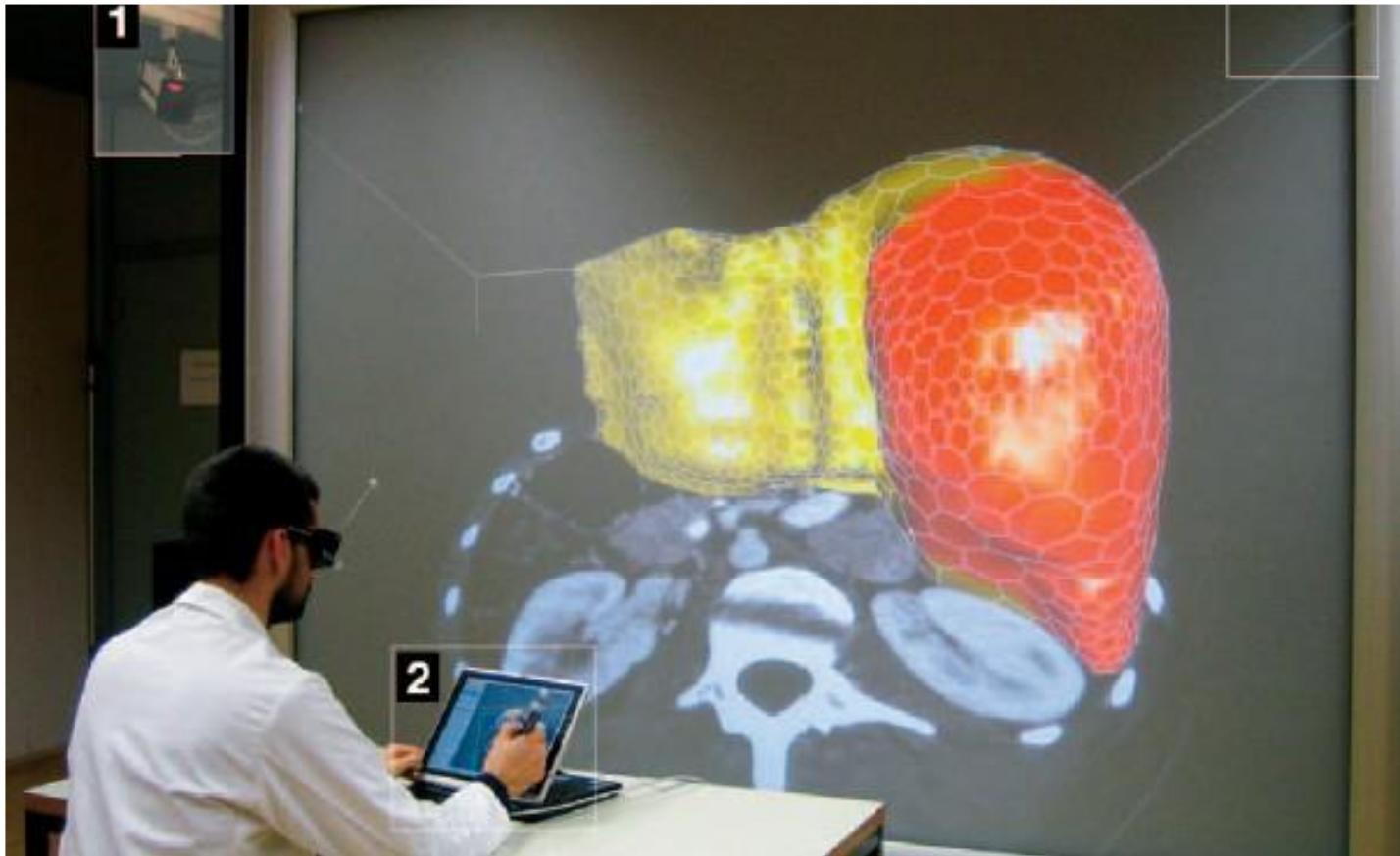
[Weidlich et al, 2008]

# Medical Data Visualization

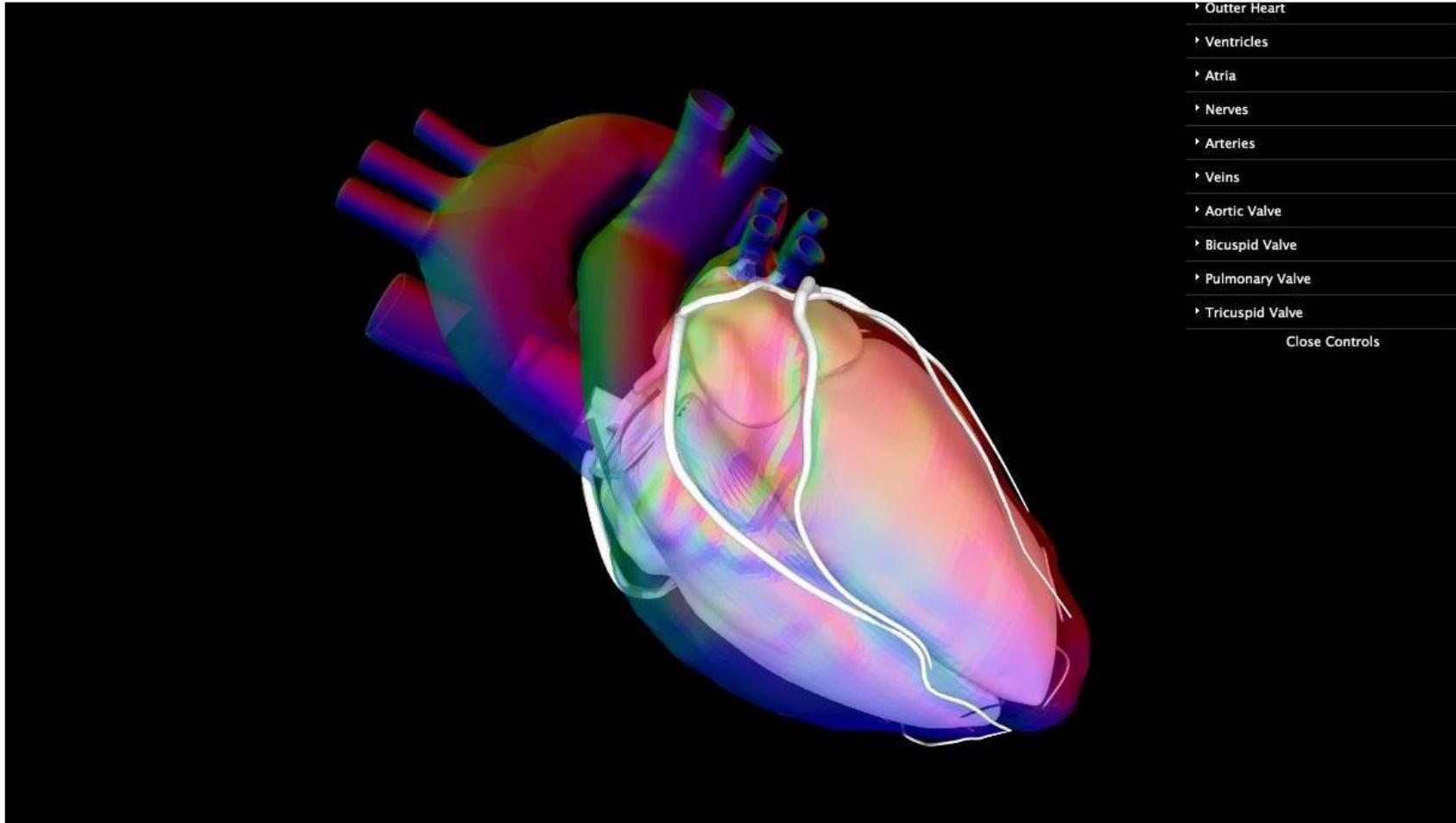


From CT data to lung models  
U.Aveiro, 2004

# Medical Data Visualization



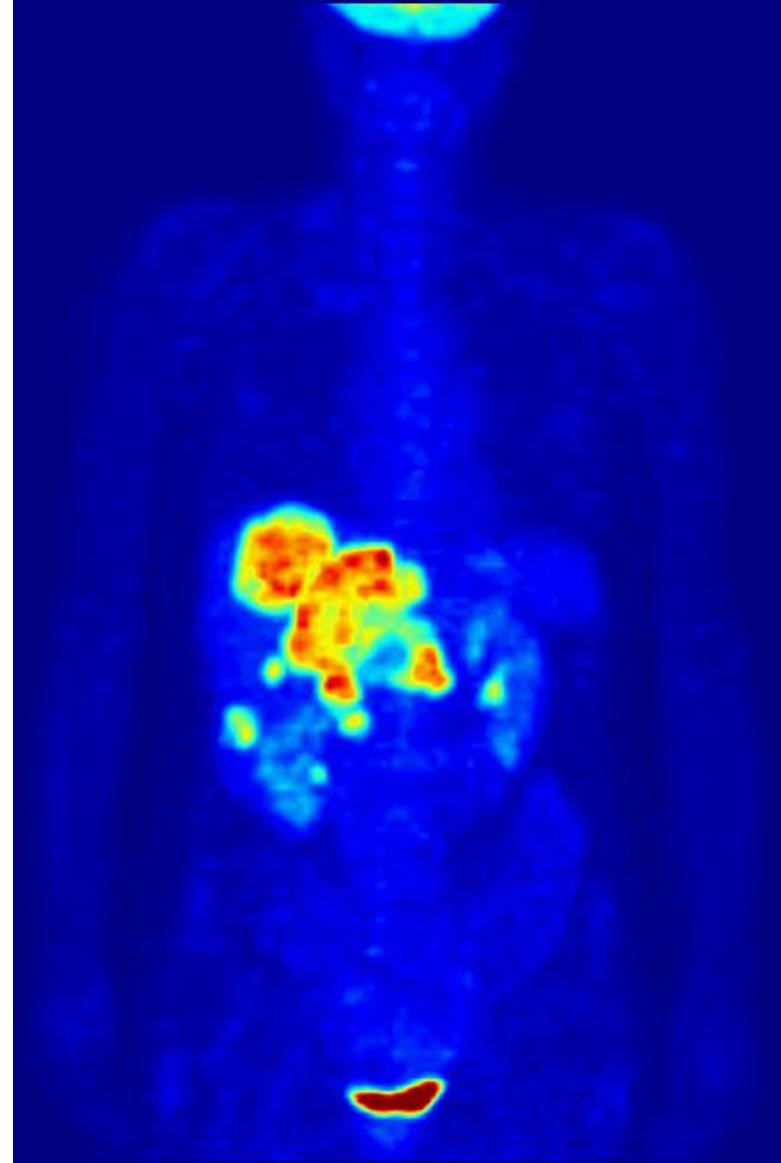
# Medical Data Visualization



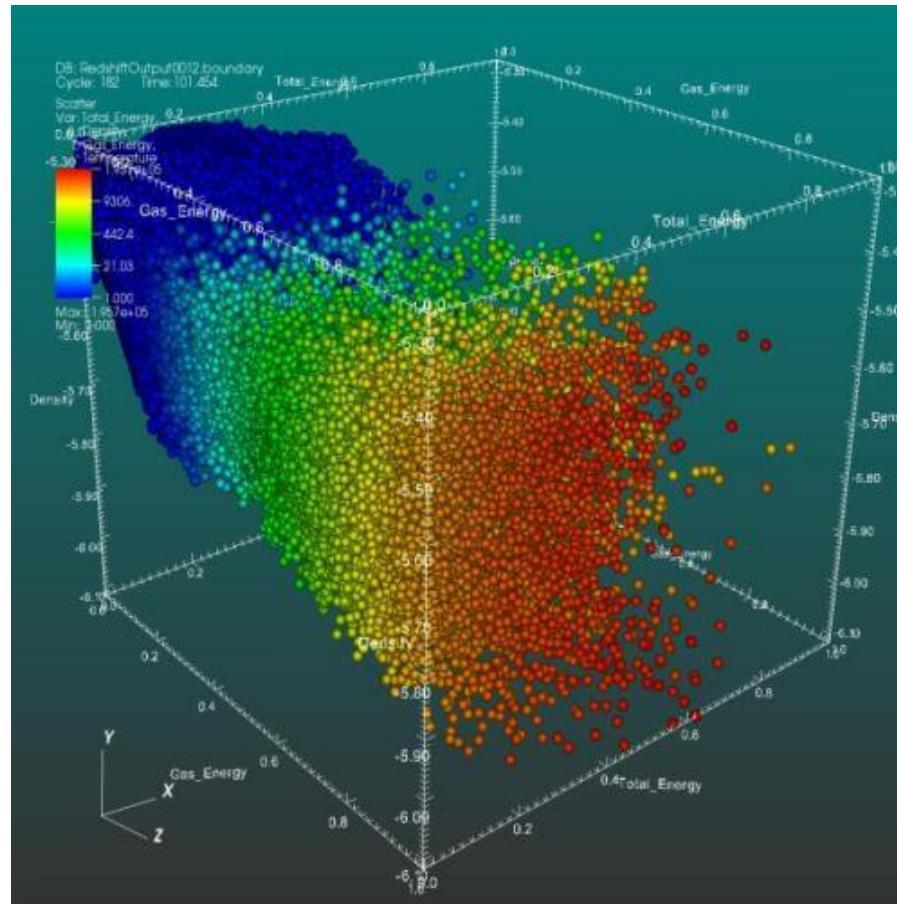
# Medical Data Visualization

- PET scan for tumor diagnosis

[Wikipedia]



# Data Visualization



[Wikipedia]

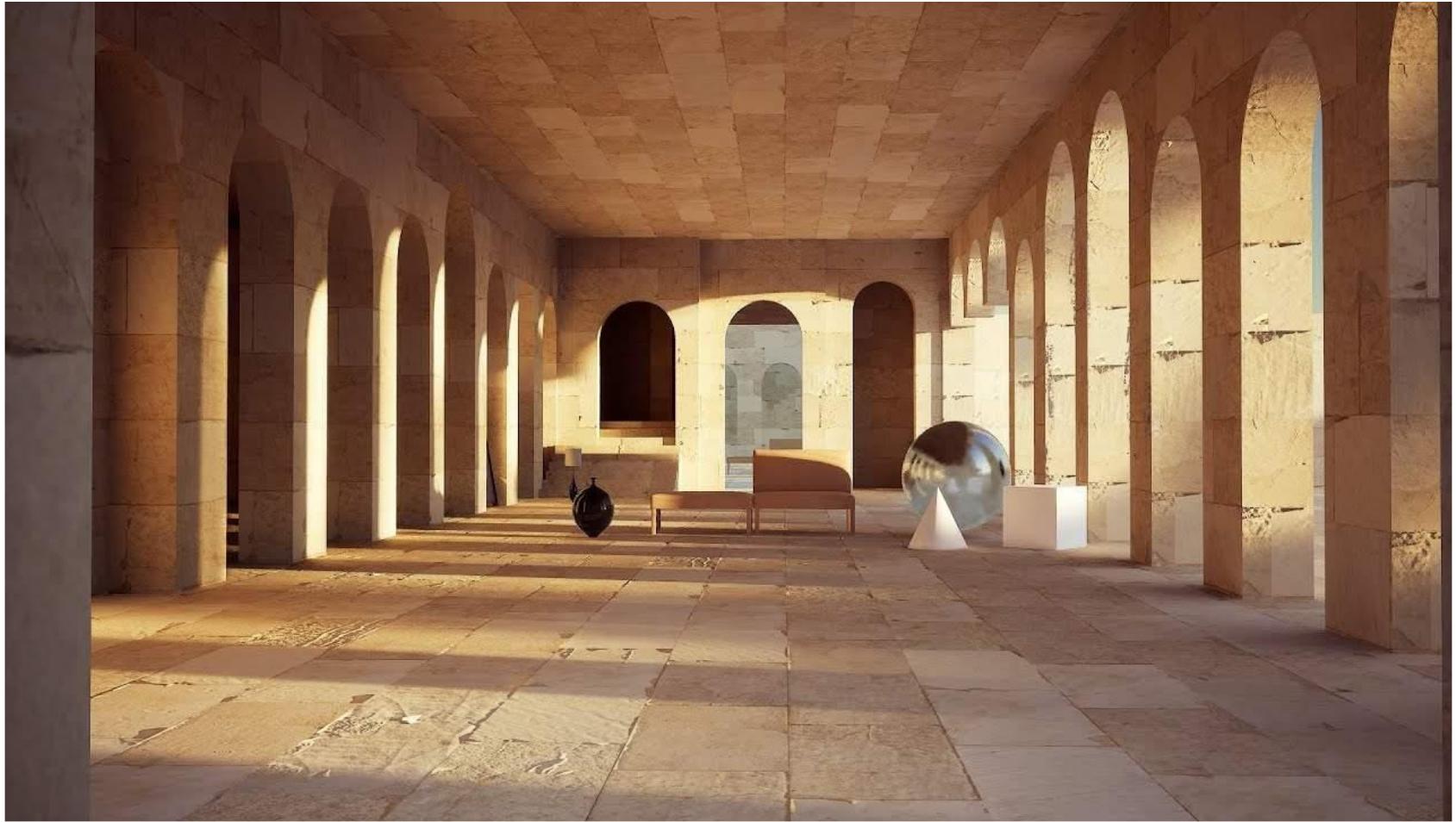
# Visualizing Population Density



# Realistic Image Synthesis – dezeen



# Global Illumination



immersivelearning.news



# CG – Some YouTube videos

- SIGGRAPH 2021
  - Technical Papers Trailer
  - VR Theater
  - Computer Animation Festival Electronic Theater



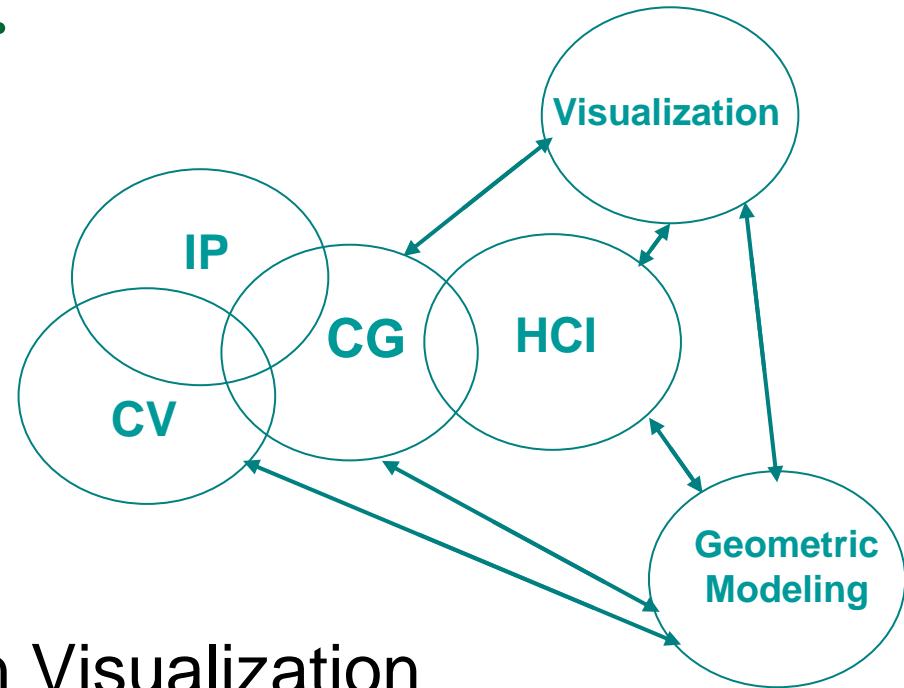
# **CG IS NOT ALONE**

# Computer Graphics vs ...

		Output	
		Model	Image
Input	Model	Geometric Modeling	Computer Graphics
	Image	Computer Vision	Image Processing

# CG is not alone...

- Core areas:
  - CG, IP, CV and HCI
- Satellite areas:
  - Geometric Modeling
  - Data and Information Visualization
- What is common?
  - CG, IP : image file formats, color models, ...
  - CG, CV : 3D model representations, ...
  - IP, CV : noise removal, filters, ...



# Example – Medical Imaging

- Processing pipeline
  - Noise removal
  - Segmentation
  - Generating 2D / 3D models
  - Data visualization
  - User interaction
  - ...



[[www.mevislab.de](http://www.mevislab.de)]

# EVOLUTION

# Lara Croft



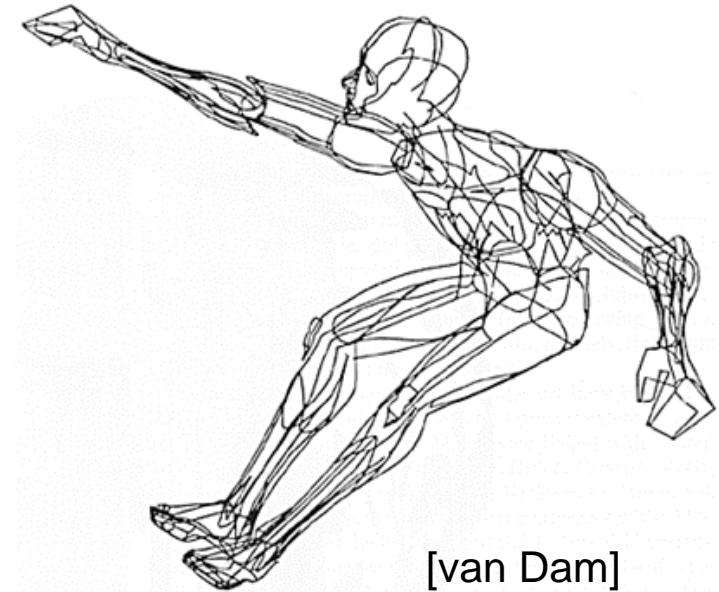
# Computer Graphics: 1950 – 1960

## ■ Earliest days of computing

- Pen **plotters**
- Simple **calligraphic displays**

## ■ Issues

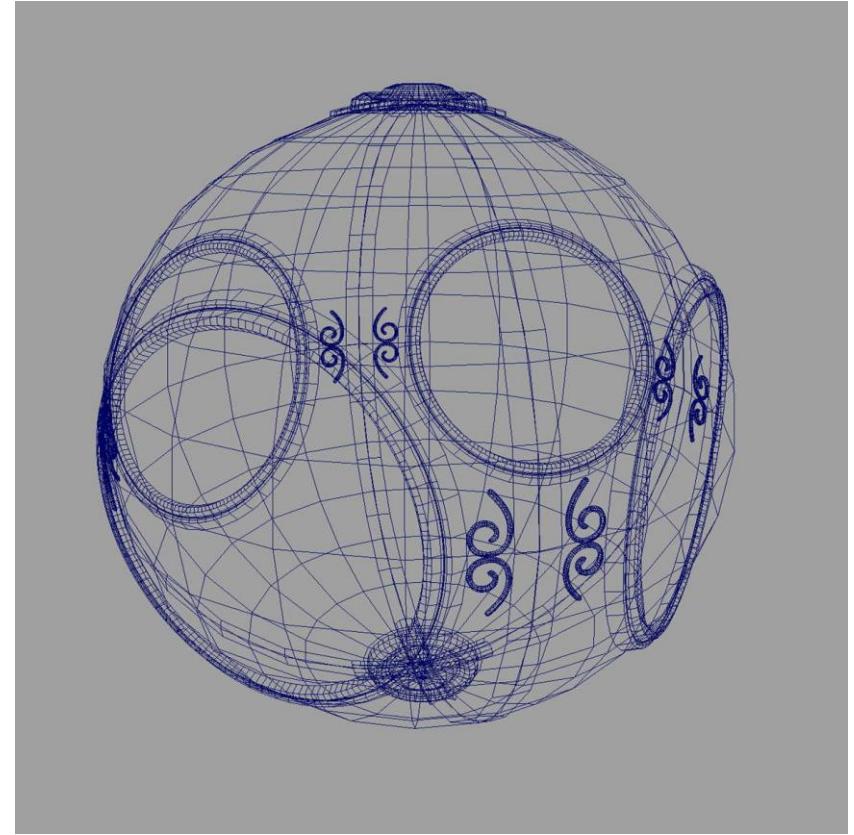
- Cost of display refresh
- Slow, unreliable, expensive computers



[van Dam]

# Computer Graphics: 1960 – 1970

- **Wireframe graphics**
  - Draw only lines !



[Angel]

# Computer Graphics: 1960 – 1970

## ■ Ivan Sutherland's Sketchpad

- PhD thesis at MIT (1963)
- Man-machine interaction
- Processing loop
  - Display something
  - Wait for user input
  - Generate new display

Demo at YouTube

[<http://history-computer.com>]

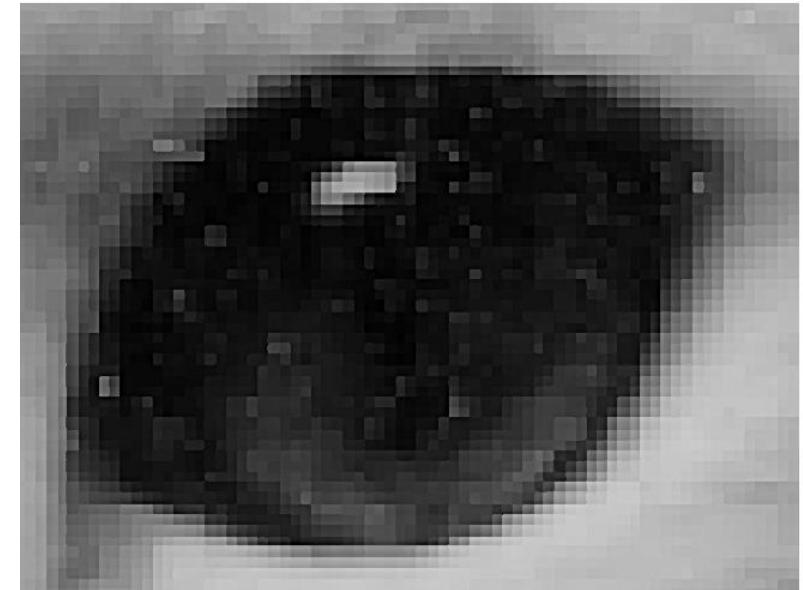
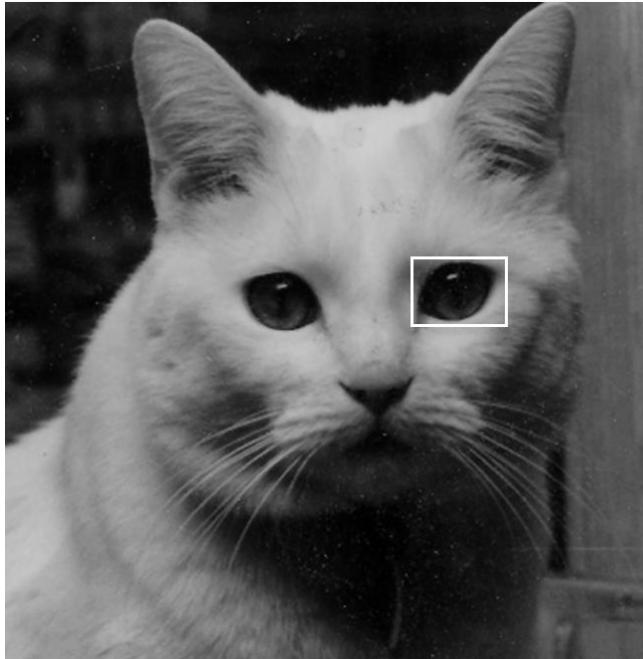


# Computer Graphics: 1970 – 1980

- Raster graphics
  - Allows drawing polygons
- First graphics standards
- Workstations and PCs
- WIMP GUI + WYSI**A**WYG
  - Desktop metaphor
  - Selection and direct manipulation

# Raster graphics

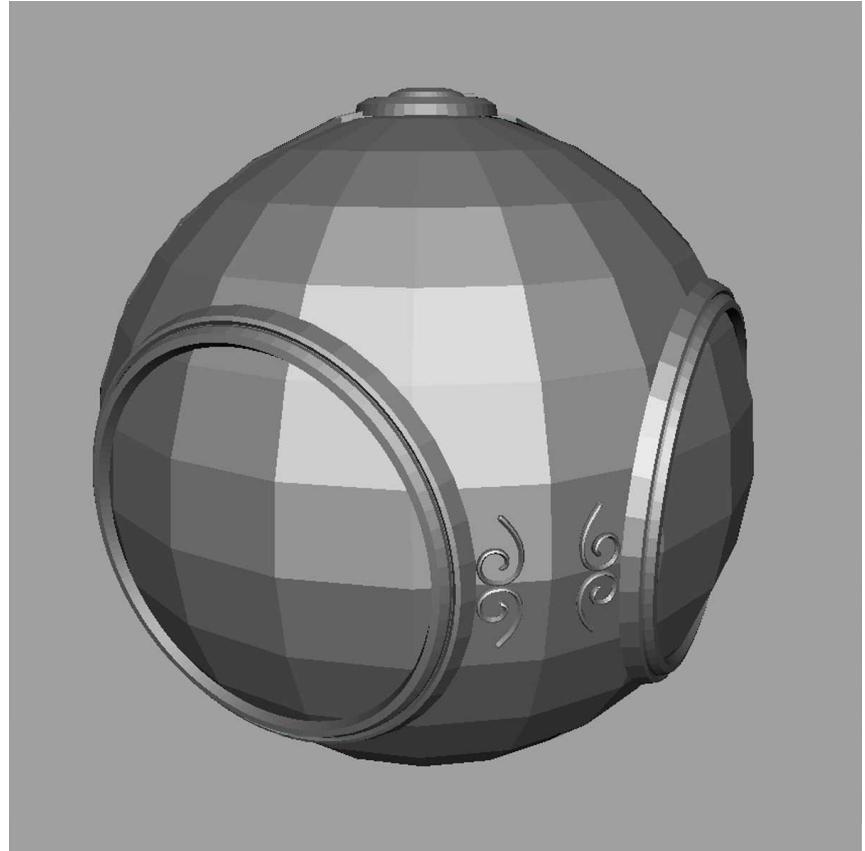
- Image produced as an array (the **raster**) of picture elements (**pixels**) in the **frame buffer**



[Angel]

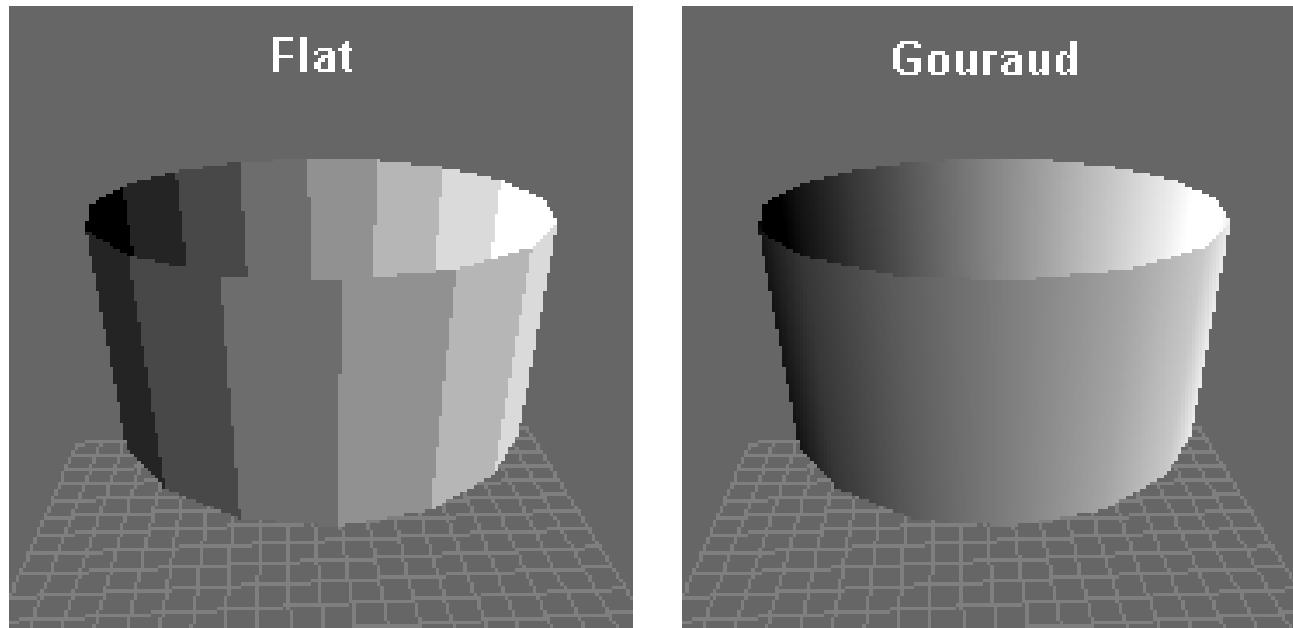
# Raster graphics

- Drawing polygons
- Illumination models
- Shading methods



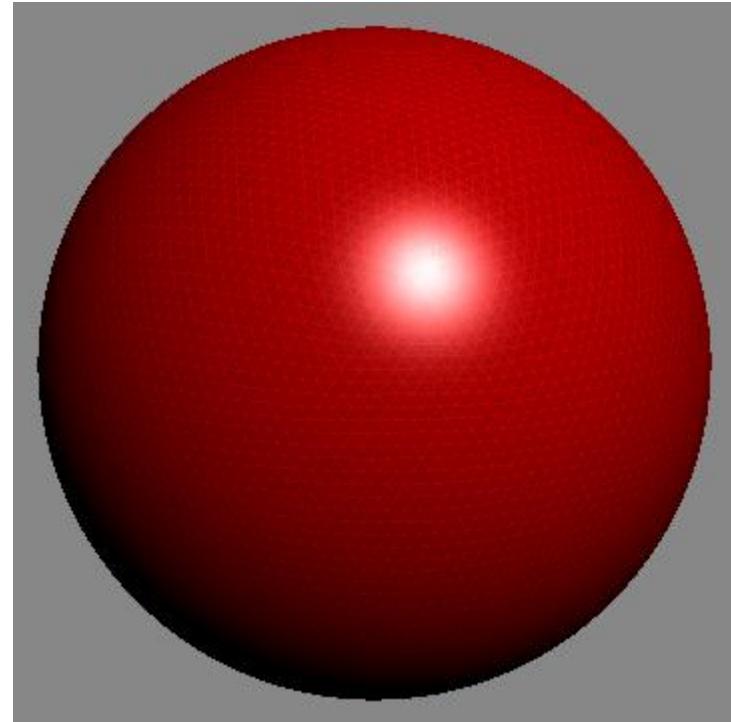
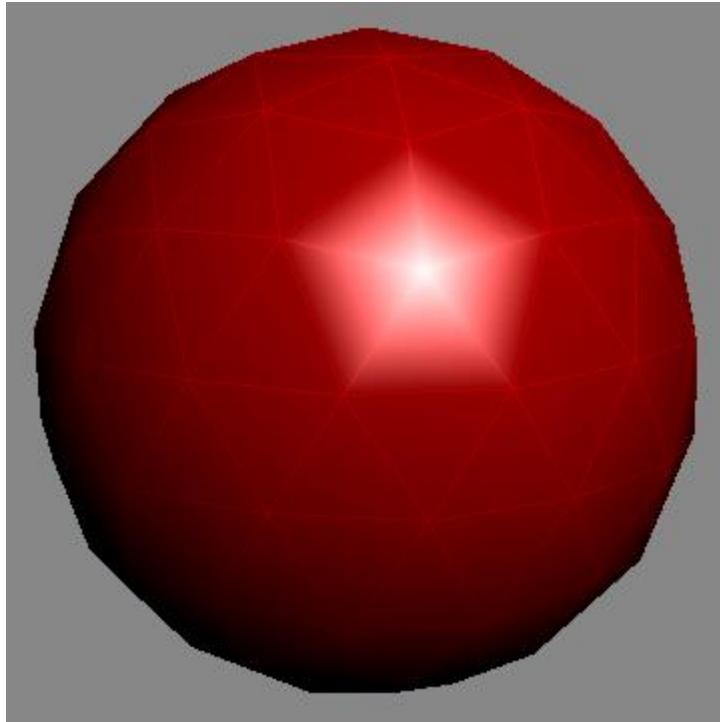
[Angel]

# Gouraud shading – 1971



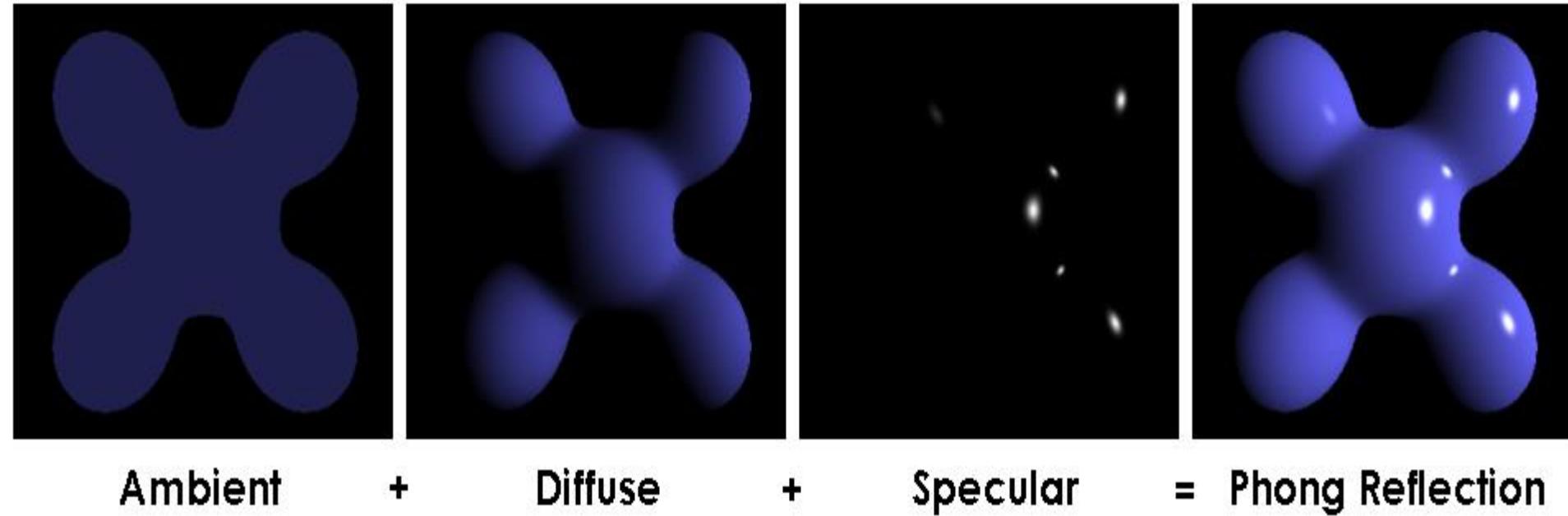
[Wikipedia]

# Gouraud shading



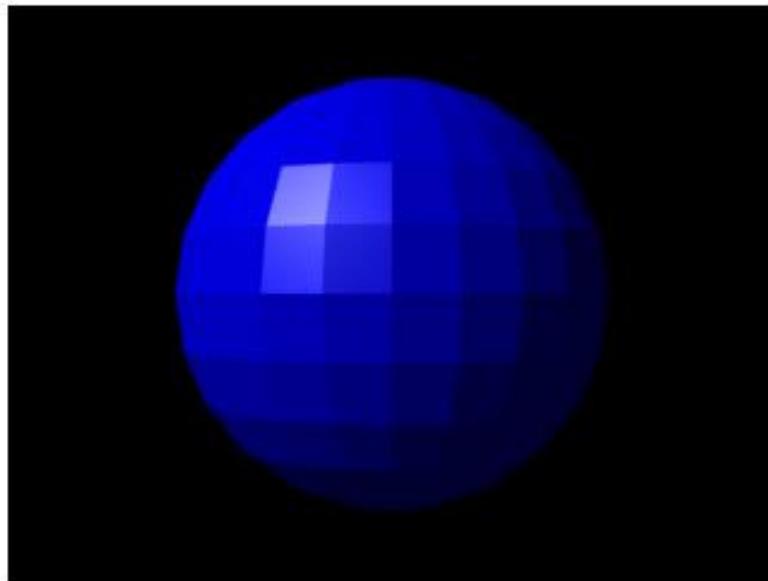
[Wikipedia]

# Phong reflection model – 1973

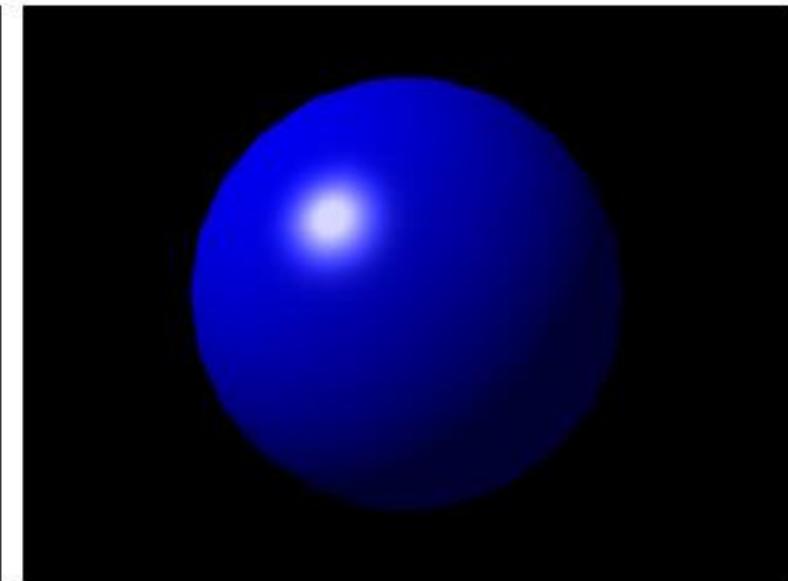


[Wikipedia]

# Phong shading – 1973



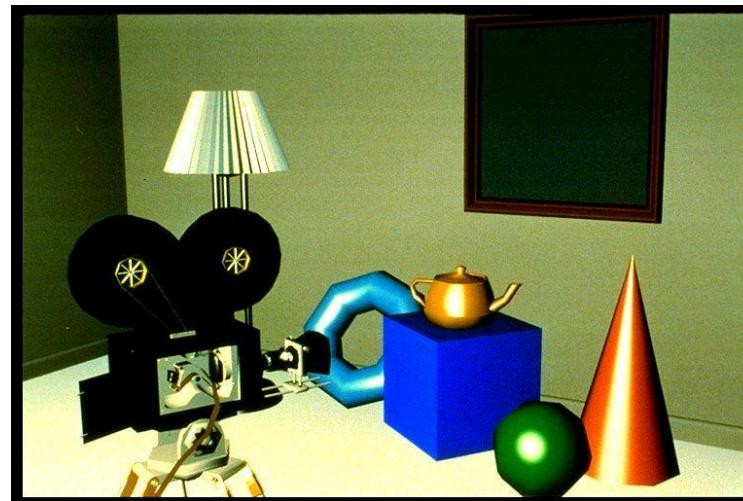
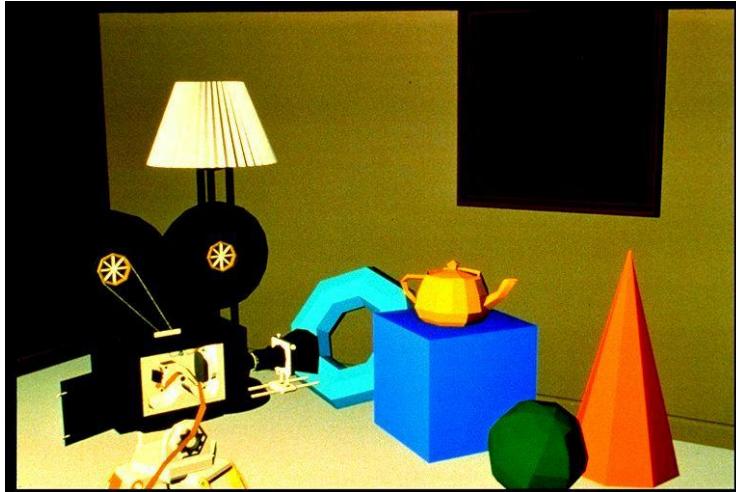
FLAT SHADING



PHONG SHADING

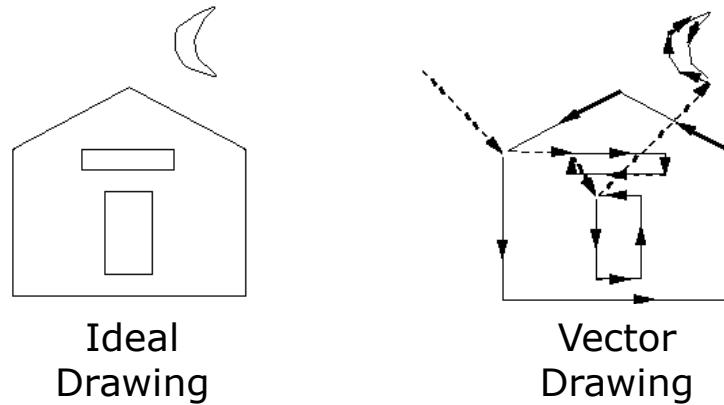
[Wikipedia]

# Can you spot the differences ?



# Vector graphics vs Raster graphics

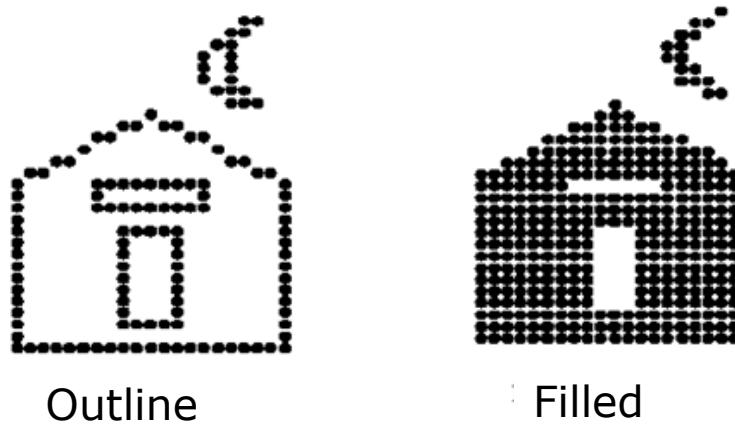
- **Vector graphics** is driven by display commands
  - `move(x,y); line(x,y); ...`
  - Survives as **SVG** – Scalable Vector Graphics



[van Dam]

# Vector graphics vs Raster graphics

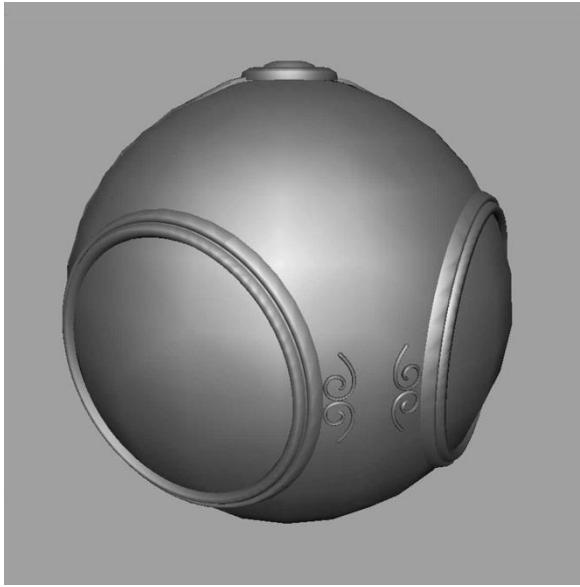
- **Raster graphics** is used in TV displays and laser printers
  - Lowest level of representation
  - No semantics
  - BUT **aliasing errors**



[van Dam]

# Computer Graphics: 1980 – 1990

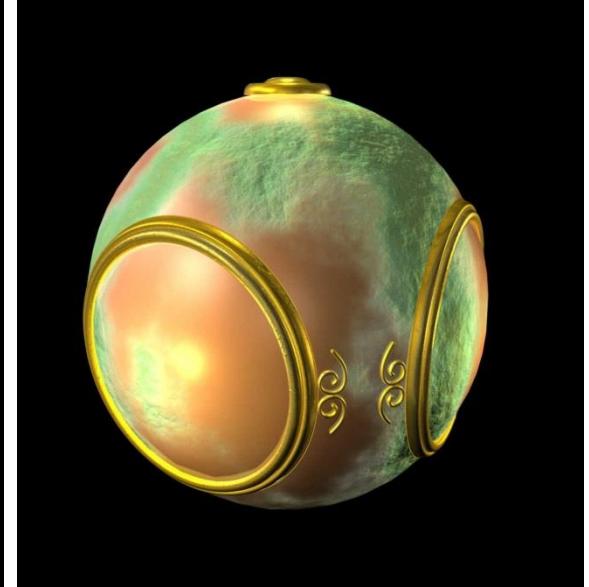
## ■ The quest for **realism**



Smooth shading



Environment mapping

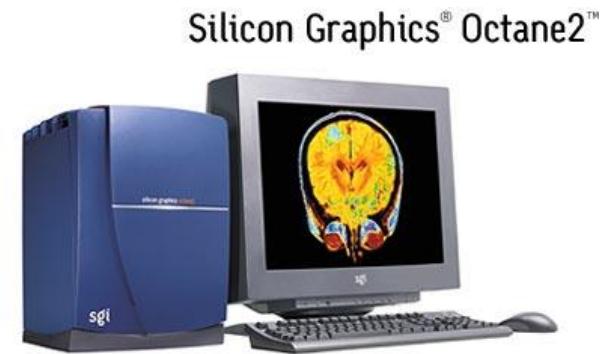


Bump mapping

[Angel]

# Computer Graphics: 1980 – 1990

- Special purpose **hardware**
  - Graphics workstations
- Industry-based **standards**
  - PHIGS
  - RenderMan
- Human-Computer Interaction



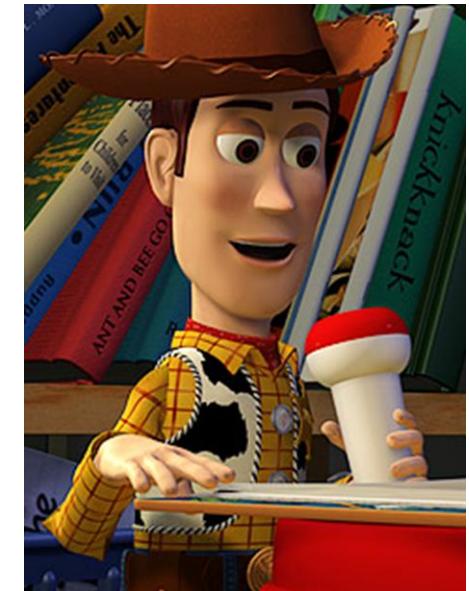
Silicon Graphics® Octane2™

Graphics workstations such as these have been replaced with commodity hardware (CPU + GPU),

[van Dam]

# Computer Graphics: 1990 – 2000

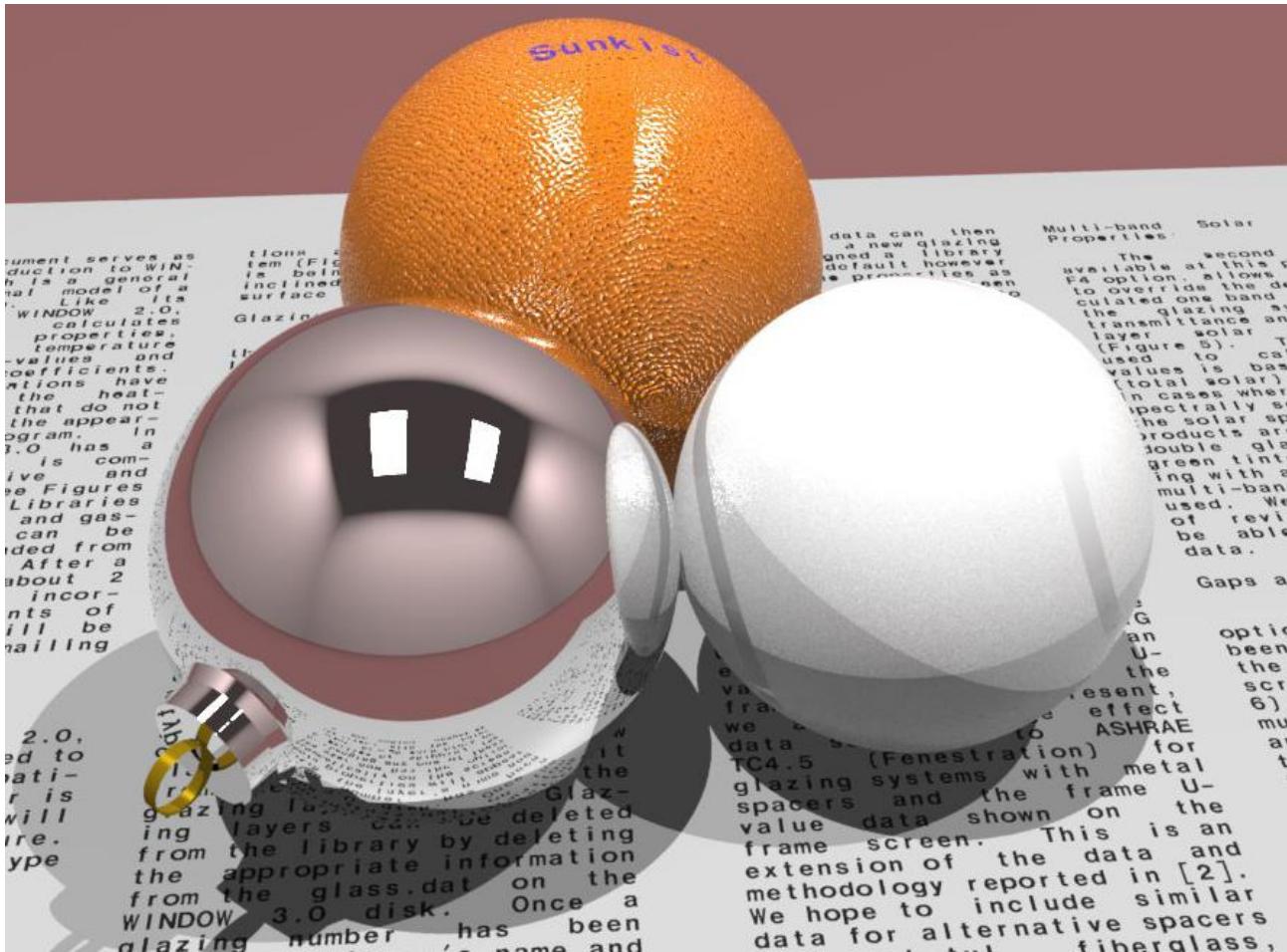
- OpenGL API
- First successful computer-generated feature-length animation film: Toy Story
- New hardware capabilities



# Computer Graphics: 2000 – ...

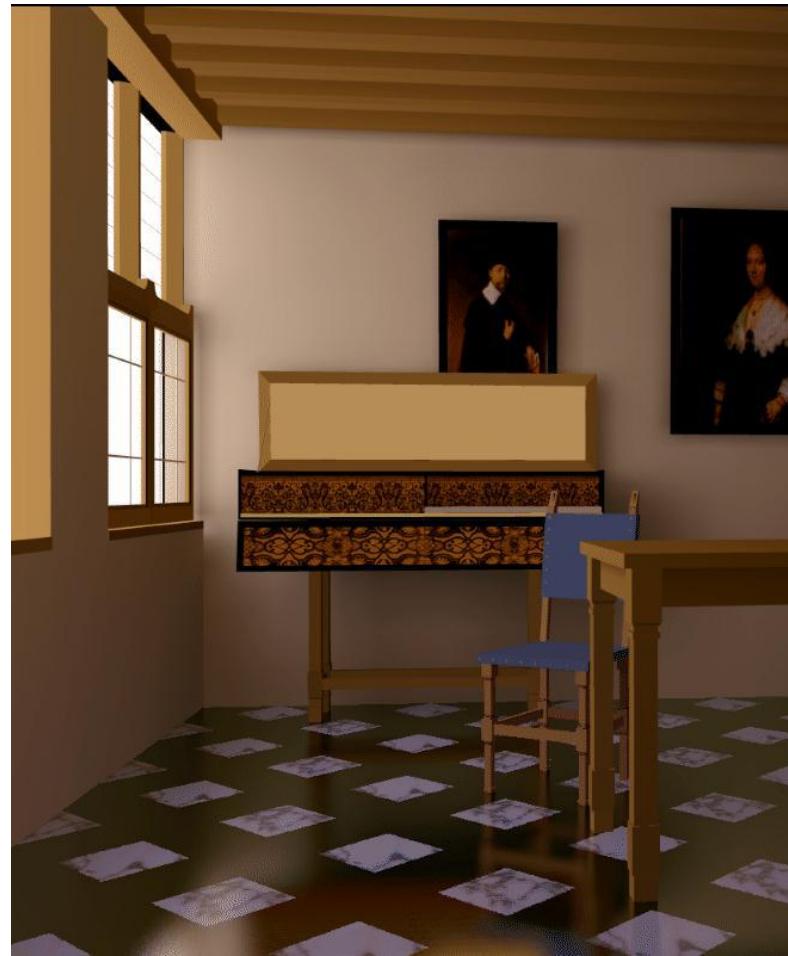
- Photorealism
- Graphics cards for PCs dominate the market
  - Nvidia
  - AMD (ATI)
- Game boxes / players determine the market
- CG is routine in the film industry

# Ray-Tracing example



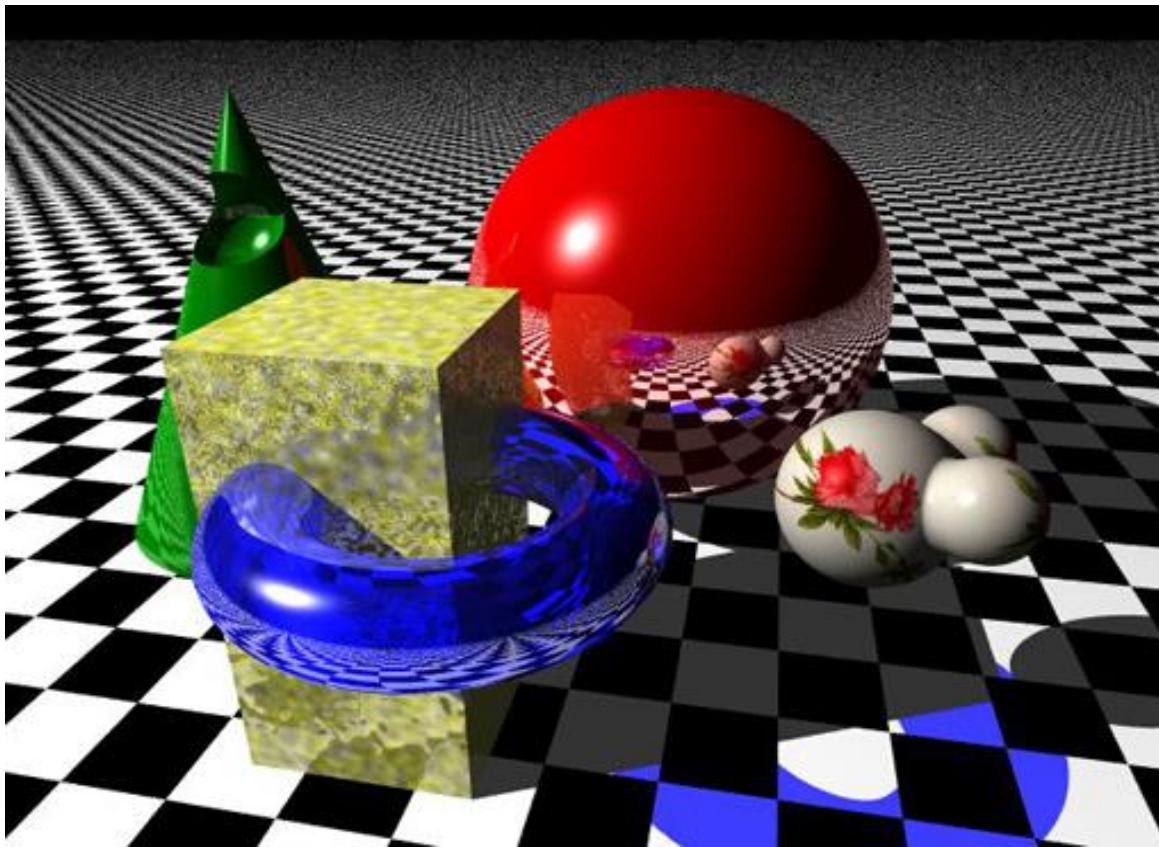
<http://radsite.lbl.gov/radiance/book/img/plate10.jpg>

# *“Vermeer’s Studio”*



Wallace & Cohen, 1987: Radiosity and Ray-Tracing

# Another Ray-Tracing example



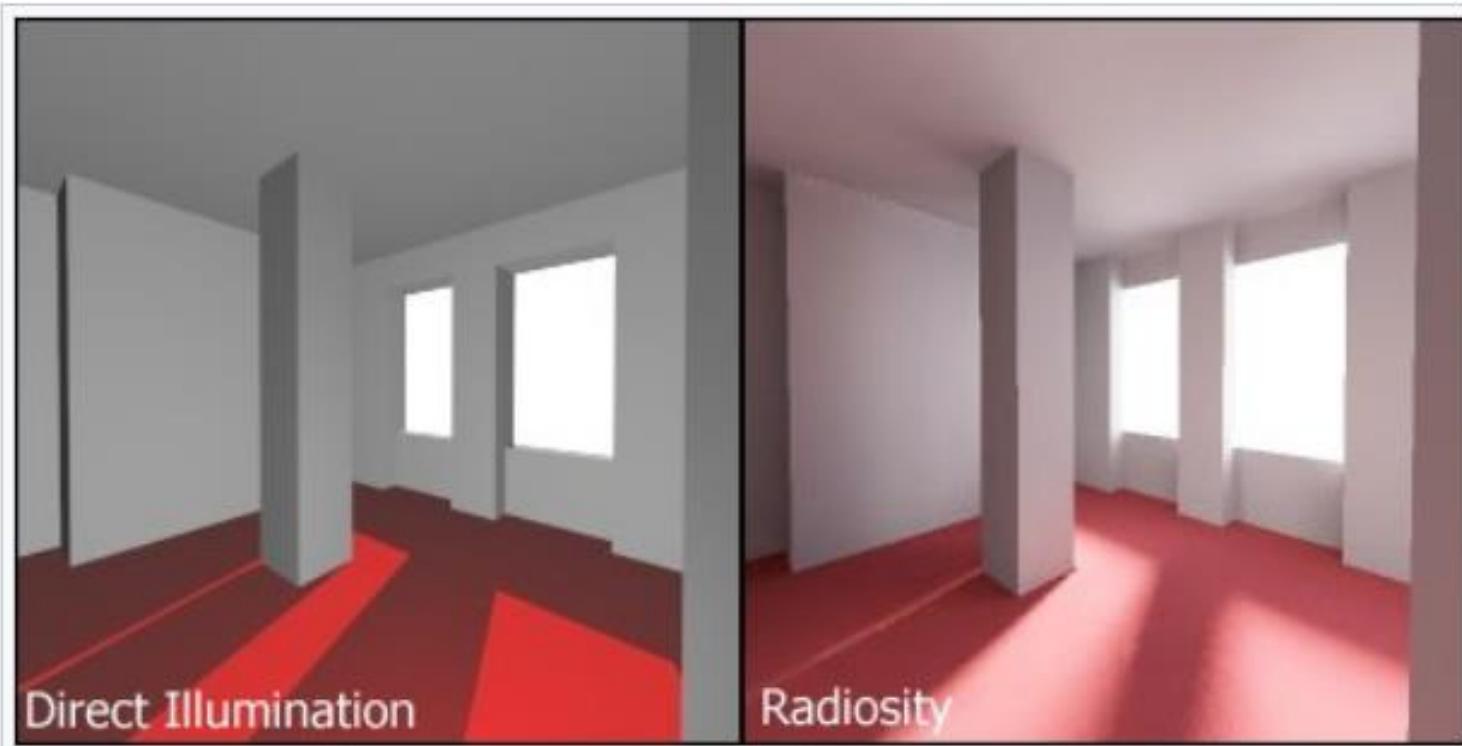
<http://www.tjhsst.edu/~dhyatt/superap/samplex.jpg>

# Ray-Tracing + Radiosity



[Wikipedia]

# Radiosity



Difference between standard direct illumination without shadow umbra, and radiosity with shadow umbra

[Wikipedia]

# Radiosity



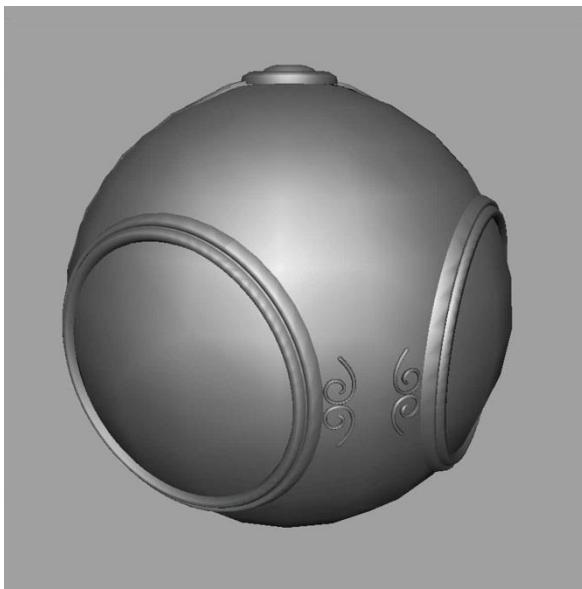
Without radiosity



With radiosity

[Burdea]

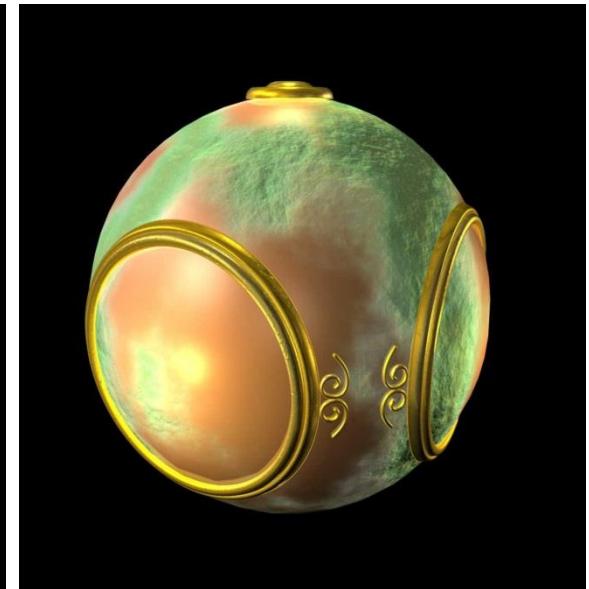
# Texture mapping



Smooth shading



Environment mapping



Bump mapping

[Angel]

# Textures – Simulating Ray-Tracing



[<http://www.okino.com>]

- Increased realism !!
  - 11 light sources + 25 texture maps

# TRENDS & TECHNOLOGIES

# Current Panorama

- Augmented Reality used in Hollywood film making
- Virtual Reality WILL be massive !
  - Crash test dummies
  - New car design / development
- Pixar is here !
  - RenderMan is free !!

# Current Panorama

- Science and Maths enabling and profiting from advances
  - Modeling / Simulation / Animation
- Gamification
  - Unreal Engine / Unity / Cryengine

# Enabling technologies for modern CG

## ■ Graphics subsystems

- Offload processing from CPU to GPU, for doing graphics operations much quicker

## ■ Hardware constant “revolution”

- Moore's Law
- Multi-core 64-bit CPUs
- Advances in commodity GPUs every 6 months vs. some years for general purpose CPUs

# Enabling technologies for modern CG

- Software improvements
- Algorithms and data structures
  - Modeling of materials
  - Rendering of natural phenomena
  - “Acceleration” data structures for rendering
- Parallelization
  - GPUs

# Enabling technologies for modern CG

## ■ Input devices

- Mouse / tablet & stylus / multi-touch
- Force feedback / game controllers
- Scanners / digitizers / digital cameras
- Body as interaction device



Xbox Kinect



Leap Motion

[Andy van Dam]



Nimble UX

# Enabling technologies for modern CG

## ■ Many **form factors**

- Laptops / desktops
- Smartphones / tablets
- Smartwatches
- HMDs
- Augmented Reality
- Virtual Reality



Microsoft's first Surface



Android Phones



Tablets



Apple Watch



Android Wear



[Andy van Dam]

Microsoft Hololens



Vive

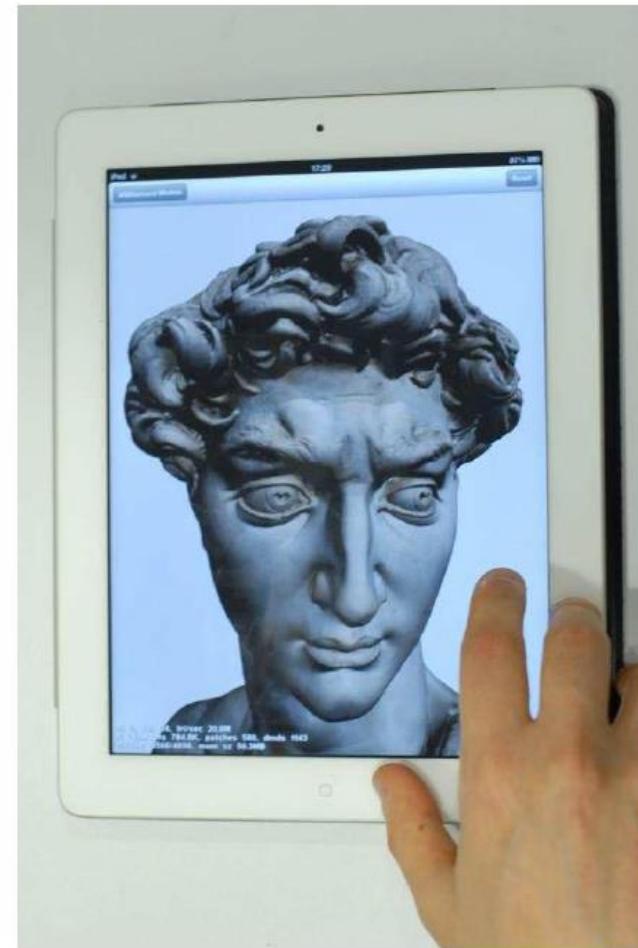


Oculus Rift



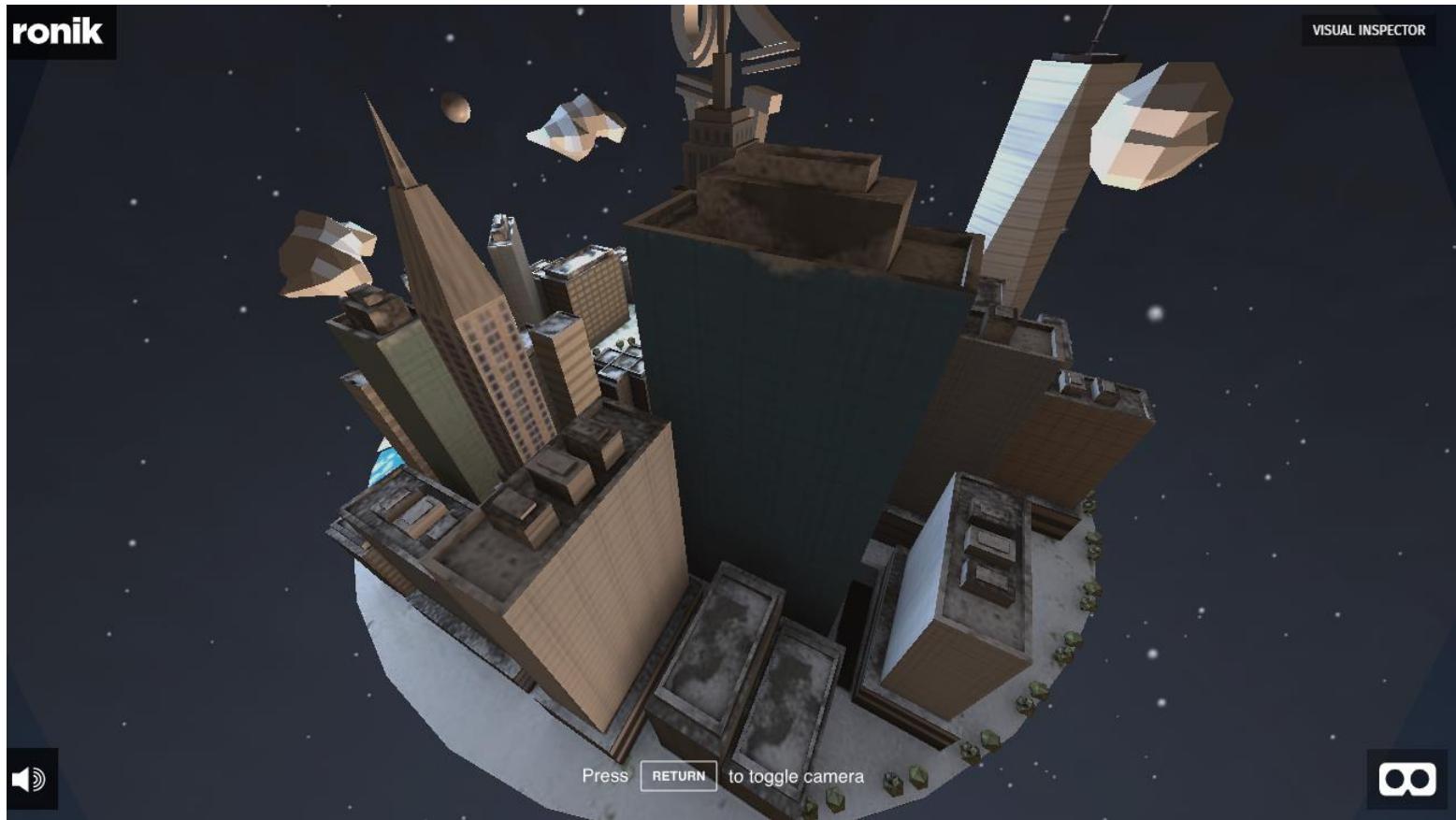
Google Cardboard

# Mobile Graphics



[SIGGRAPH Asia 2017 Course Notes]

# Web-based VR



[<https://aframe.io/examples/showcase/snowglobe/>]

# Web-based CG and VR

- PlayCanvas - The Web-first game engine
- WebXR
- Sketchfab

# **INTERACTIVE VS BATCH CG**

# CG Main Tasks

## ■ Modeling

- Construct individual models / objects
- Assemble them into a 2D or 3D scene

## ■ Animation

- Static vs. dynamic scenes
- Movement and / or deformation

## ■ Rendering

- Generate final images
- Where is the observer?
- How is he / she looking at the scene?

# Interactive Computer Graphics

- User controls **content**, **structure**, and **appearance** of objects and their displayed images, via rapid **visual feedback**
  - Also called **real-time** computer graphics or, in certain contexts, real-time **rendering**
- Remember **Sutherland's Sketchpad** (1963)
  - Monitor + light pen + function-key panels
  - Bimanual operation

# Interactive CG – Basic components

- **Input**
  - Mouse / stylus / multi-touch / in-air fingers / ...
- **Processing** and storing of the underlying models
- **Display / Output**
  - Screen / paper printer / 3D printer / video / ...

# Batch Computer Graphics

- Non-interactive, **off-line** rendering
- Final production-quality video and film
  - Animation / Special effects – FX
- Rendering a single frame of *The Good Dinosaur* (a 24 fps movie) averaged **48 hours** on a **30,000-core render farm!**
  - See statistics at [fxguide article](#)

# Batch Computer Graphics



Still from *The Good Dinosaur*

[Andy van Dam]



Pixar's Render Farm

# **MAIN TASKS**

# CG Main Tasks

## ■ Modeling

- Construct individual models / objects
- Assemble them into a 2D or 3D scene

## ■ Animation

- Static vs. dynamic scenes
- Movement and / or deformation

## ■ Rendering

- Generate final images
- Where is the observer?
- How is he / she looking at the scene?

# Modeling vs Rendering

## ■ Modeling

- Create models
- Apply materials to models
- Place models around scene
- Place lights in the scene
- Place the camera

[YouTube Demo](#)

## ■ Rendering

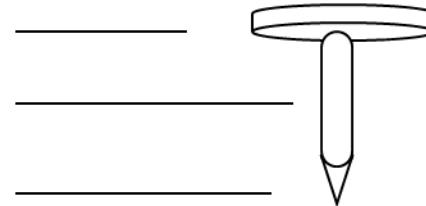
- Take picture with the camera

[van Dam]

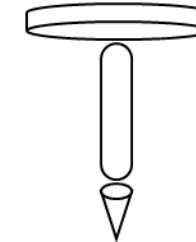
# Decomposition of a geometric model

- Hierarchy of geometrical components
- Reduction to primitives
  - Spheres, cubes, etc.
- Simple vs not-so-simple elements

Head  
Shaft  
Point



composition

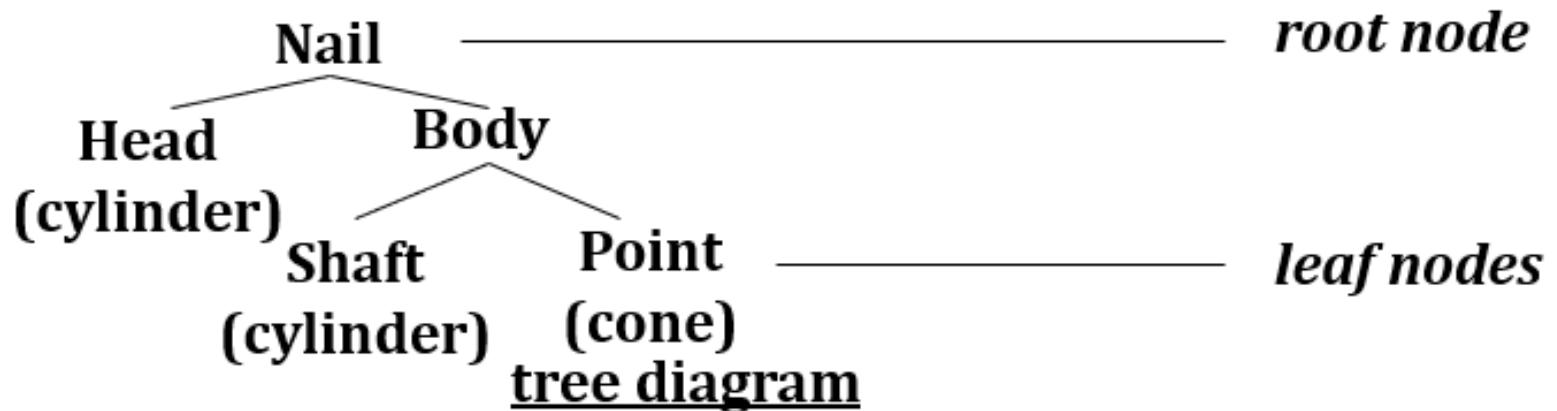


decomposition

[van Dam]

# Hierarchical representation

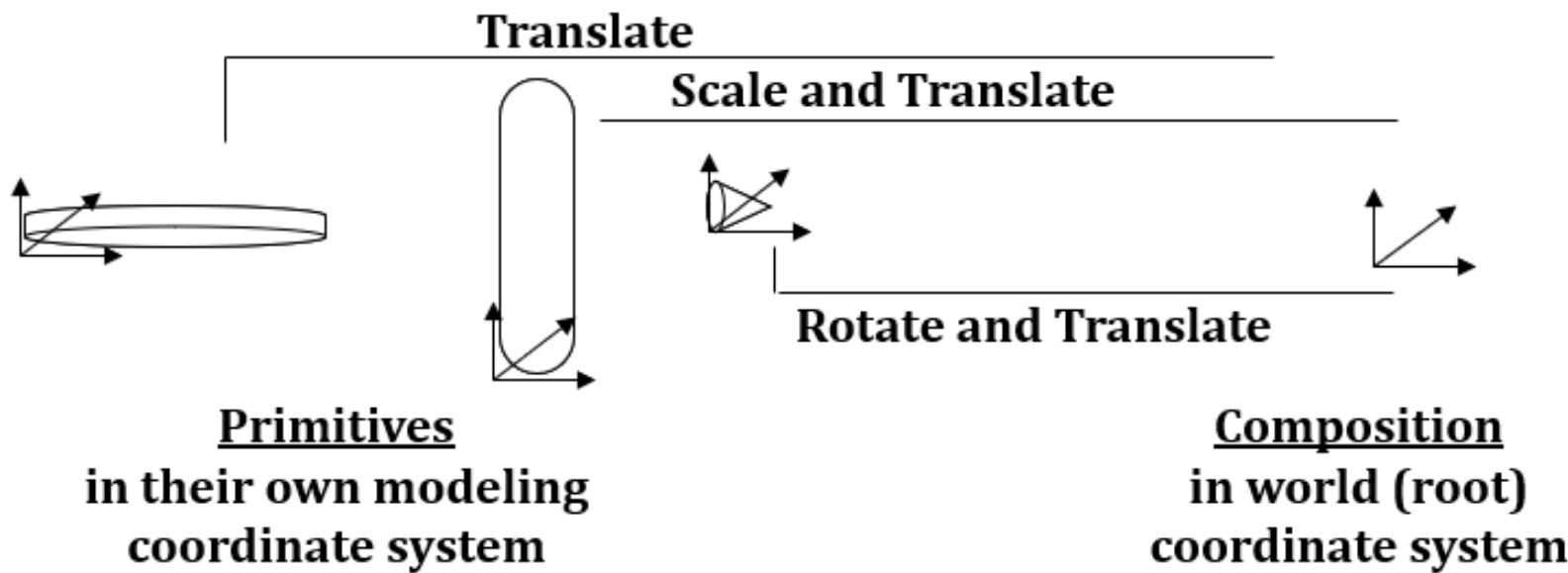
- Decomposition into collections of primitive shapes
  - Tree diagram
- **Scene-graph** : data structure to be rendered



[van Dam]

# Composition of a geometric model

- Assemble primitives to create final object
  - Apply affine transformations

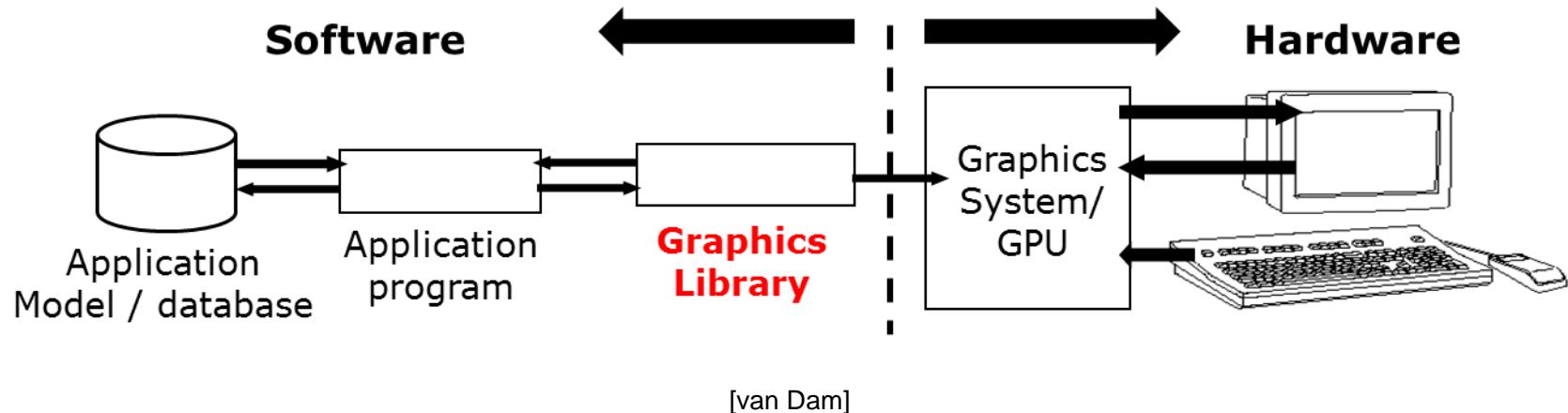


[van Dam]

# GRAPHICS APIs / LIBRARIES

# Interactive Computer Graphics

- Graphics library / package is **intermediary** between application and display hardware
- Application program **maps / renders** objects / models to images by calling on the **graphics library**
- User **interaction** allows image and / or model modification



# Graphics Libraries / APIs

- OpenGL, RenderMan, DirectX, Windows Presentation Foundation (WPF), HTML5 + **WebGL**, **three.js**, Vulkan,  
...
- **Primitives** : characters, points, lines, triangles, ...
- **Attributes** : color, line /polygon style, ...
- **Transformations** : rotation, scaling, ...
- **Light sources**
- **Viewing**
- ...



# API contents

- Functions for specifying / instantiating
  - Geometric primitives
  - Viewer / Camera
  - Light sources
  - Materials
  - ...
- Functions for simple user interaction
  - Input from devices: mouse, keyboard, etc.

# Geometric Primitives

## ■ Simple primitives

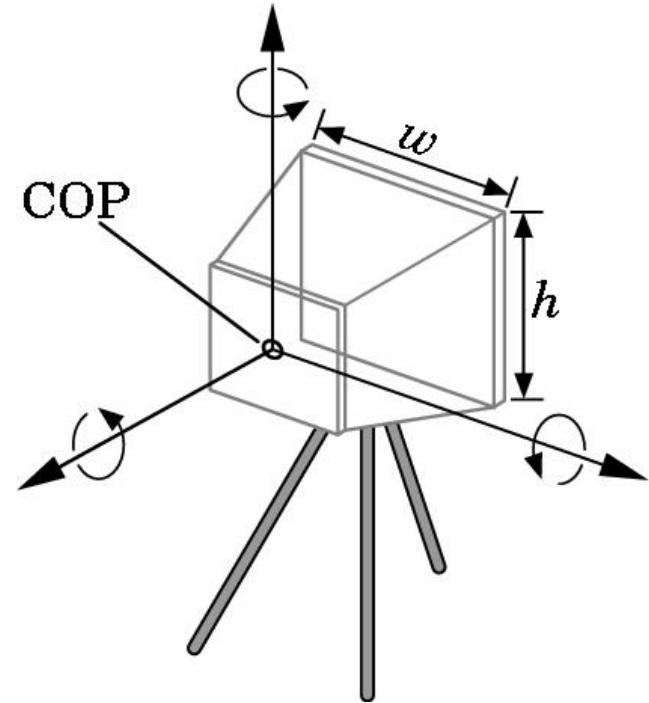
- Points
- Line segments
- Polygons

## ■ Geometric primitives

- Parametric curves / surfaces
- Cubes, spheres, cylinders, etc.

# Camera specification

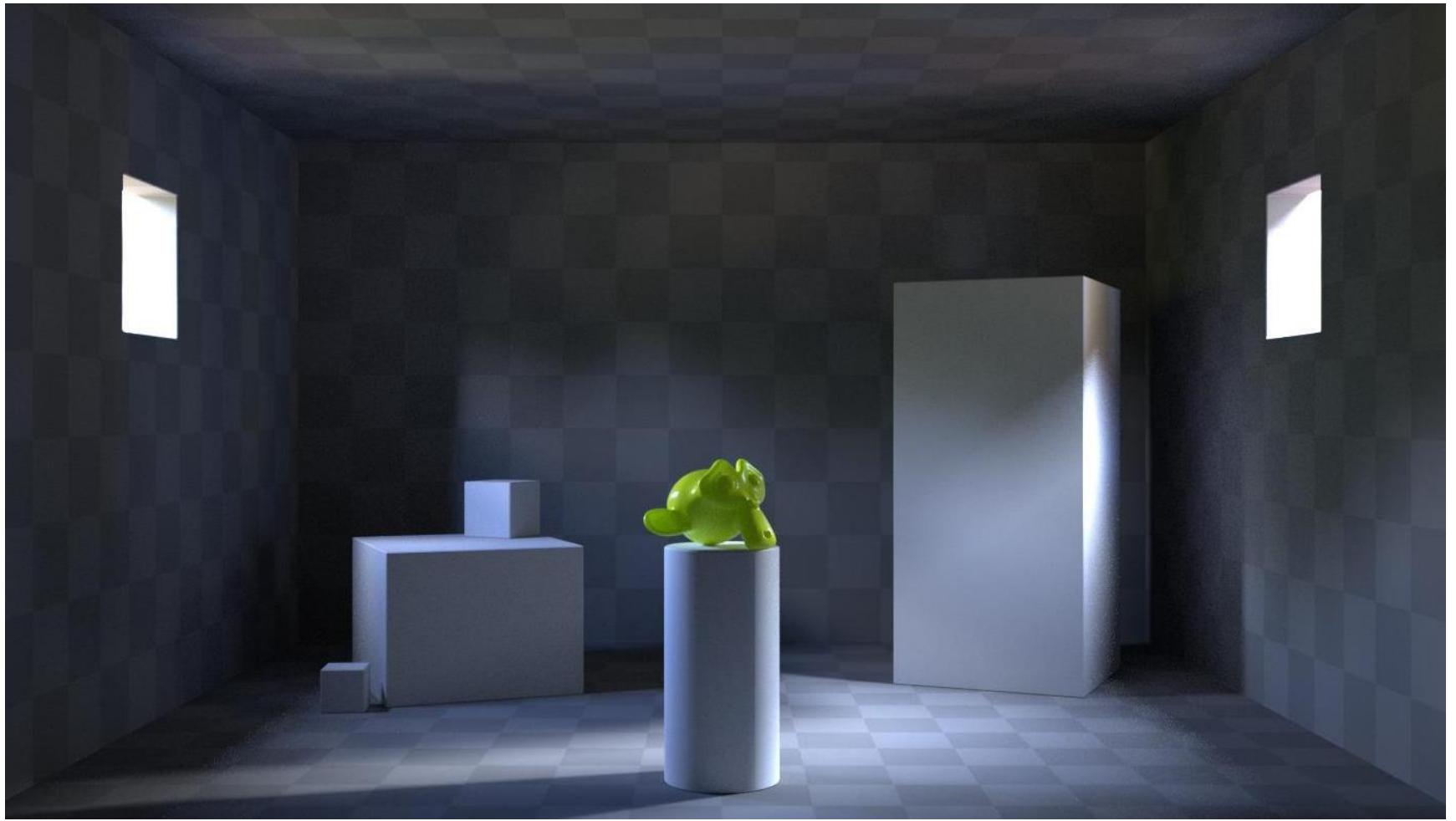
- Six degrees of freedom
  - Position of lens center
- Lens
- Film size
- Orientation of film plane



[Angel]

# Lights and materials

- Types of light sources
  - Point vs distributed light sources
  - Spotlights
  - Near and far sources
  - Color properties
- Material properties
  - Absorption: color properties
  - Scattering: diffuse and specular
  - Transparency



<https://blenderartists.org/t/what-if-i-want-a-low-light-in-door-scene/685160>

# OpenGL



- Multi-platform API for rendering 2D and 3D computer graphics
- Interaction with the GPU to achieve hardware-accelerated rendering
- Application areas
  - CAD
  - Virtual reality
  - Scientific and Information Visualization
  - ...

# OpenGL



## ■ OpenGL ES

- Subset for use in embedded systems and portable devices



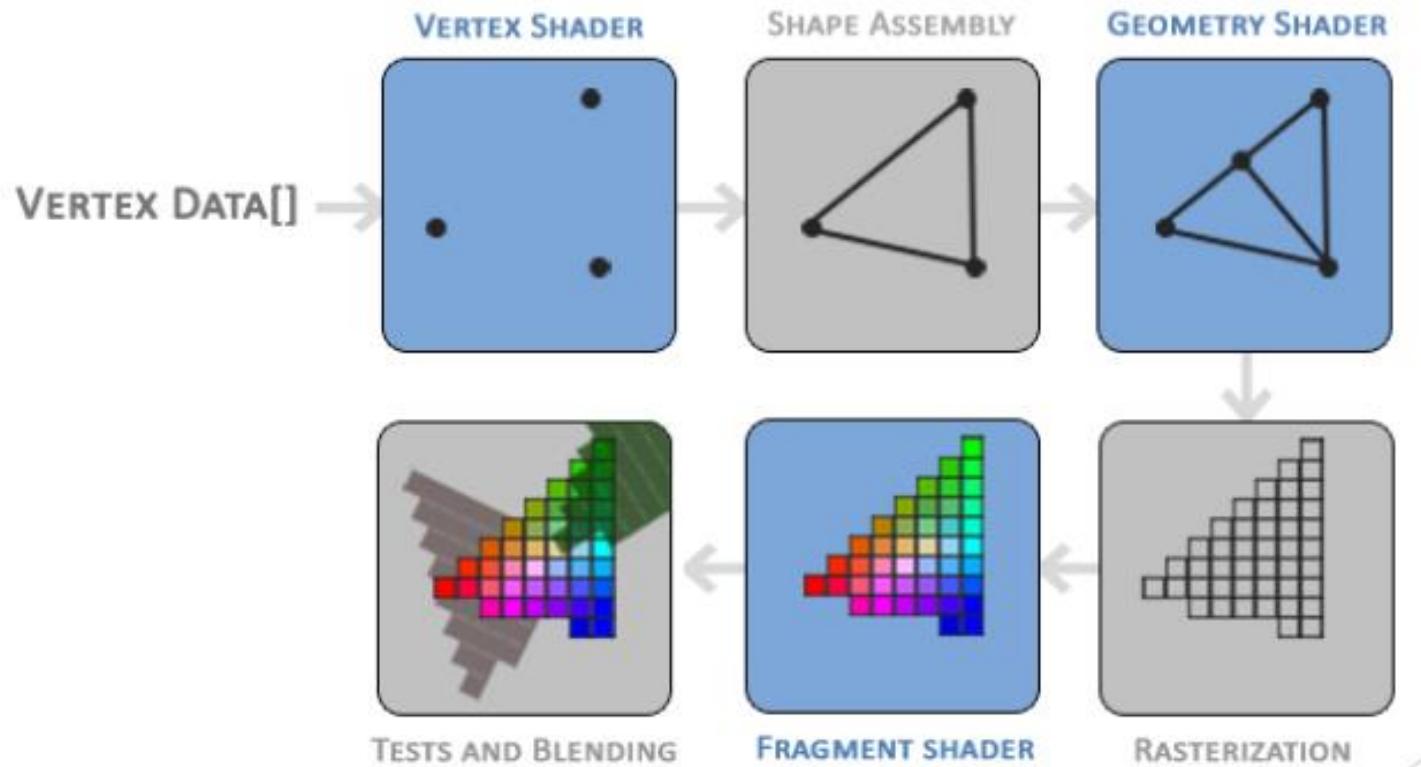
## ■ WebGL

- JavaScript API based on OpenGL ES 2.0
- Rendering interactive 2D and 3D graphics on any compatible browser, without the use of plug-ins

# GPUs

- Large collection of **highly parallel high-speed arithmetic units**; several thousand cores !
- GPUs run simple programs (“shaders”)
  - Take in **vertices** and **other data**
  - Output a **color value** for an **individual pixel**
- **GLSL**, (O)GL Shader Language, is a C-like language; controls arithmetic pipelines

# Shaders



[learnopengl.com](http://learnopengl.com)

# Three.js

- Create and display **3D CG** in **Web-browsers**
  - Cross-browser **Javascript API**
  - No need for browser plug-ins !
- No need to develop stand-alone apps !
- Uses **WebGL** !
- [threejs.org](http://threejs.org)

# Three.js

three.js r97

featured projects

[documentation](#)  
[examples](#)

[download](#)

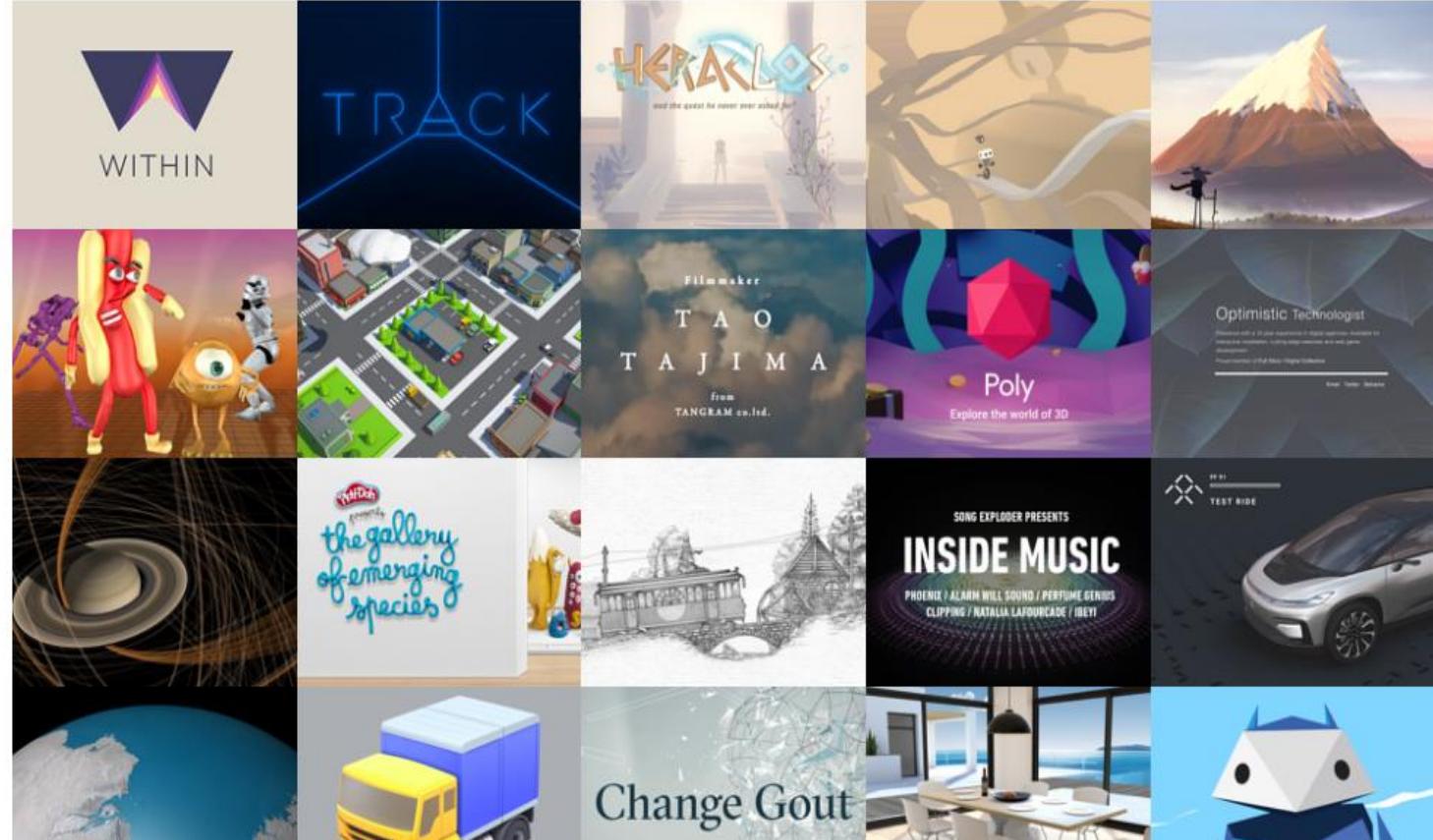
[source code](#)  
[questions](#)  
[forum](#)  
[irc](#)  
[slack](#)  
[google+](#)



Interactive  
3D Graphics  
Taught by Eric Haines



UDACITY



# RECAP

# In Summary

- Computer graphics involves both **real-time / interactive** and **batch / offline** applications
  - Both equally important, but different use cases
- **Photo-realism** has really advanced !!
  - But it still takes dozens of hours on fastest computers to mimic physical behavior

# In Summary

- Hardware evolution from vector to **raster graphics**
  - But we still have **SVG**
- **Geometry-based vs Image-based** graphics
  - Mathematical definition vs pixel manipulation

# In Summary

- Complex geometric models typically constructed **hierarchically**
  - Scene-graph data structure
- **Pixels** are discrete samples of continuous functions
  - Causes **artifacts** (“jaggies”/ “aliases”) to appear
  - Need fixing through “anti-aliasing”

# In Summary

- **Geometric models** allow representing objects or characters
  - Geometry / Detail / ...
  - Material / Appearance
  - Static vs Animated / Deformable models
- **Scene modeling**
  - Place models / light sources
  - Place the camera

# REFERENCES

# References

- S. Marschner, P. Shirley, “Fundamentals of Computer Graphics”, 4<sup>th</sup> ed, A K Peters, 2018
  - Chapter 1
  - <https://learning.oreilly.com/library/view/fundamentals-of-computer/9781482229417/>
- J. F. Hughes, A. van Dam, et al., “Computer Graphics: Principles and Practice”, 3<sup>rd</sup> ed, Addison-Wesley, 2013
  - Chapter 1
  - <https://learning.oreilly.com/library/view/hughes-computer-graphics-3-e/9780133373721/>

# ACKNOWLEDGMENTS

# Acknowledgments

- Some ideas and figures have been taken from slides of other CG courses.
- In particular, from the slides made available by Ed Angel, Andy van Dam, and Samuel Silva.
- Thanks!