Computer Graphics -Introduction

Junjie Cao @ DLUT

Spring 2018

http://jjcao.github.io/ComputerGraphics/

Computer Graphics

- One of the "core" computer science disciplines:
  - Algorithms and Theory
  - Artificial Intelligence
  - Computer Architecture
  - **Computer Graphics**
  - Computer Security
  - Computer Systems
  - Computer Vision
  - Databases
  - Machine Learning
  - Networks
  - Software Engineering

Context

- History
- Applications
- What is CG
- Administrative Stuff
- Topics
- Trends

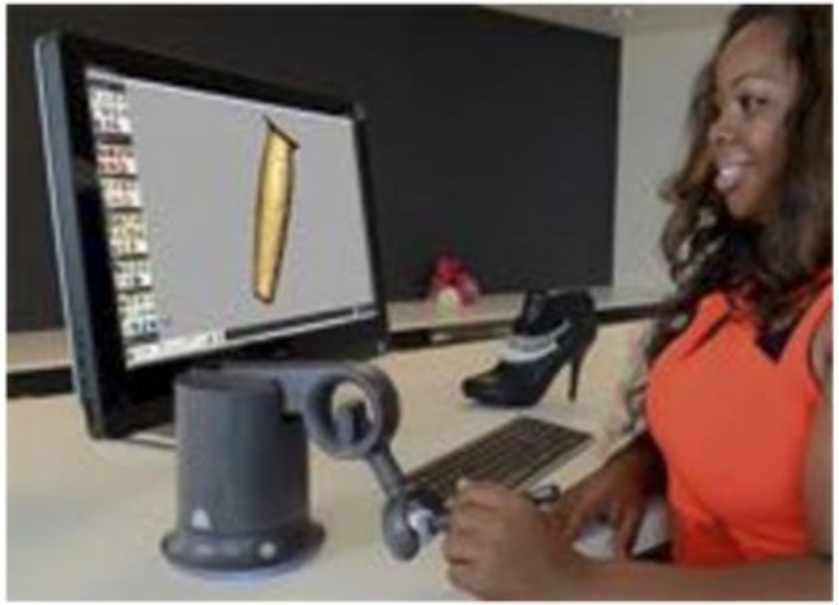
Computer Graphics vs. Vision

What is computer graphics?

- The use of computers to synthesize and manipulate **visual information**.

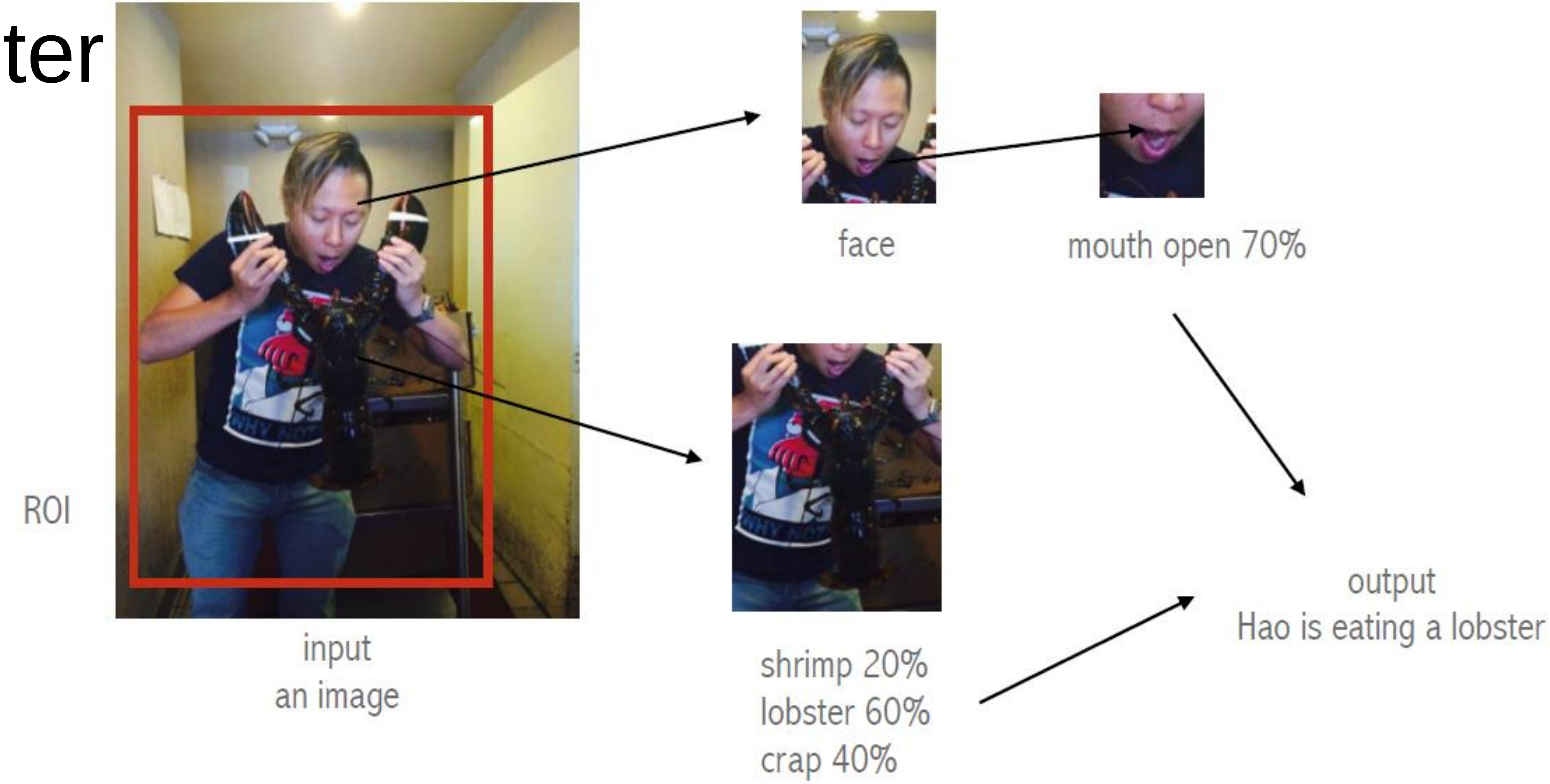
- The use of computers to synthesize and manipulate **sensory information**.

(sound)



(touch)

Computer Vision



ROI

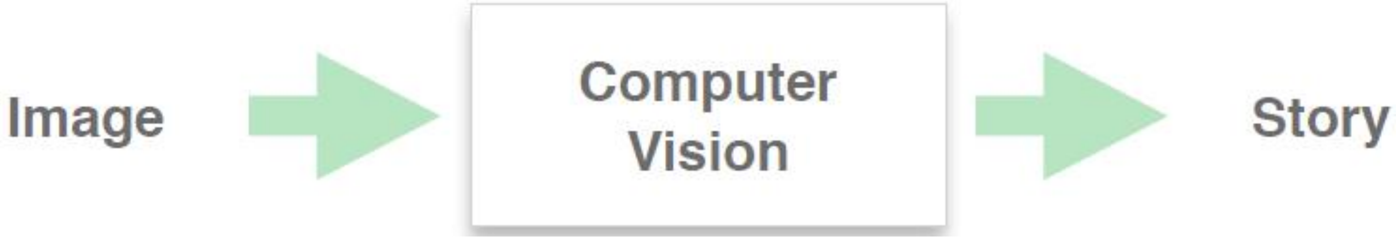
input
an image

face

mouth open 70%

shrimp 20%
lobster 60%
crap 40%

output
Hao is eating a lobster



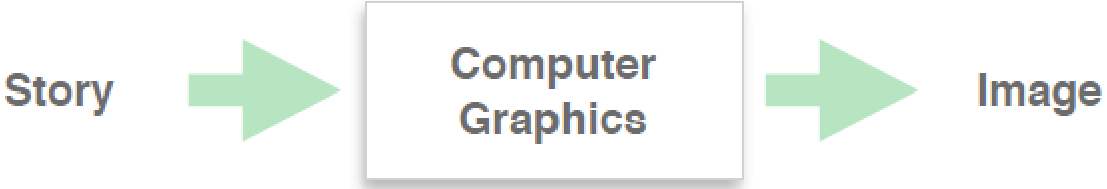
Image

Computer Vision

Story

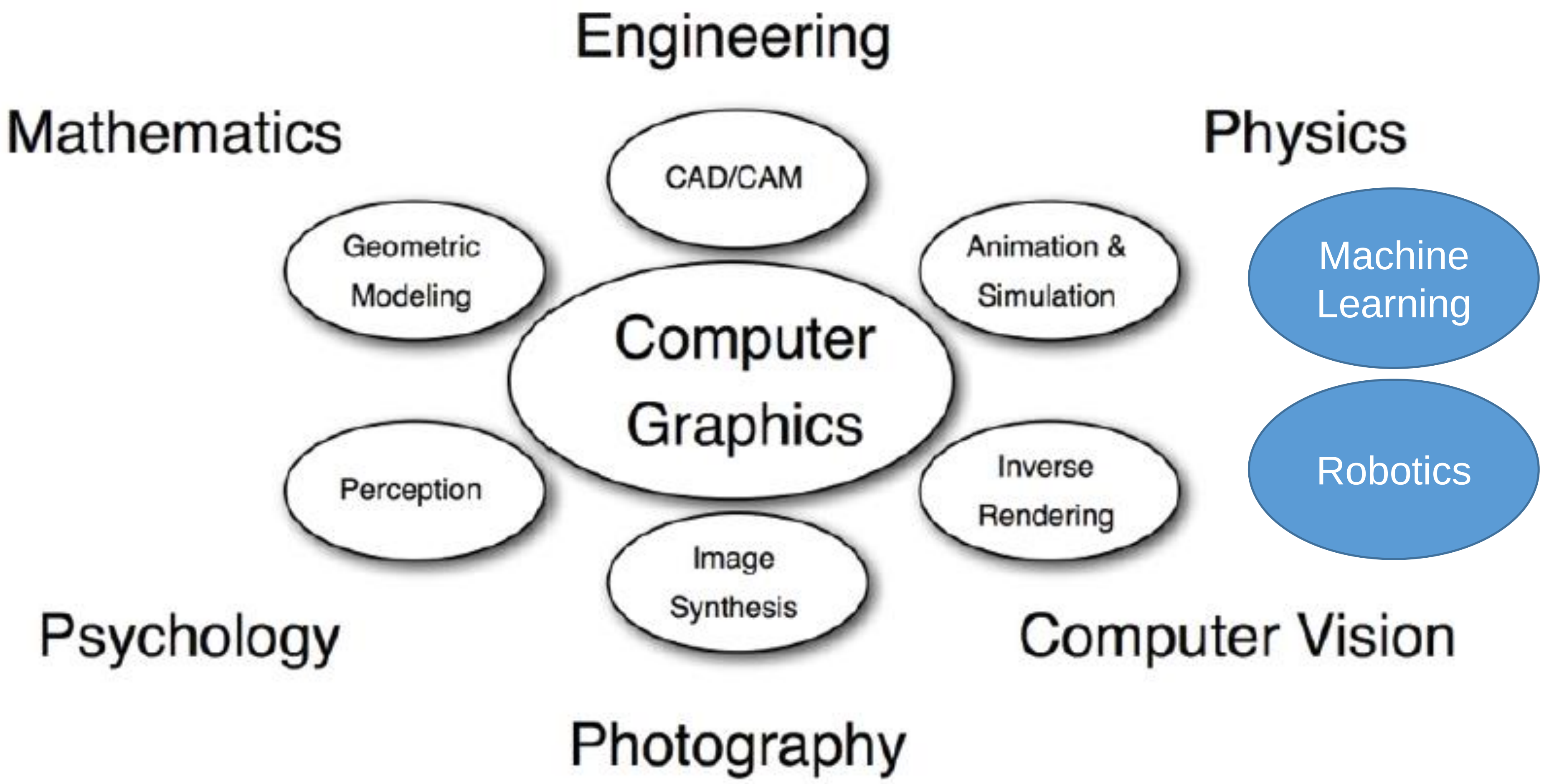
Computer Graphics

Action!

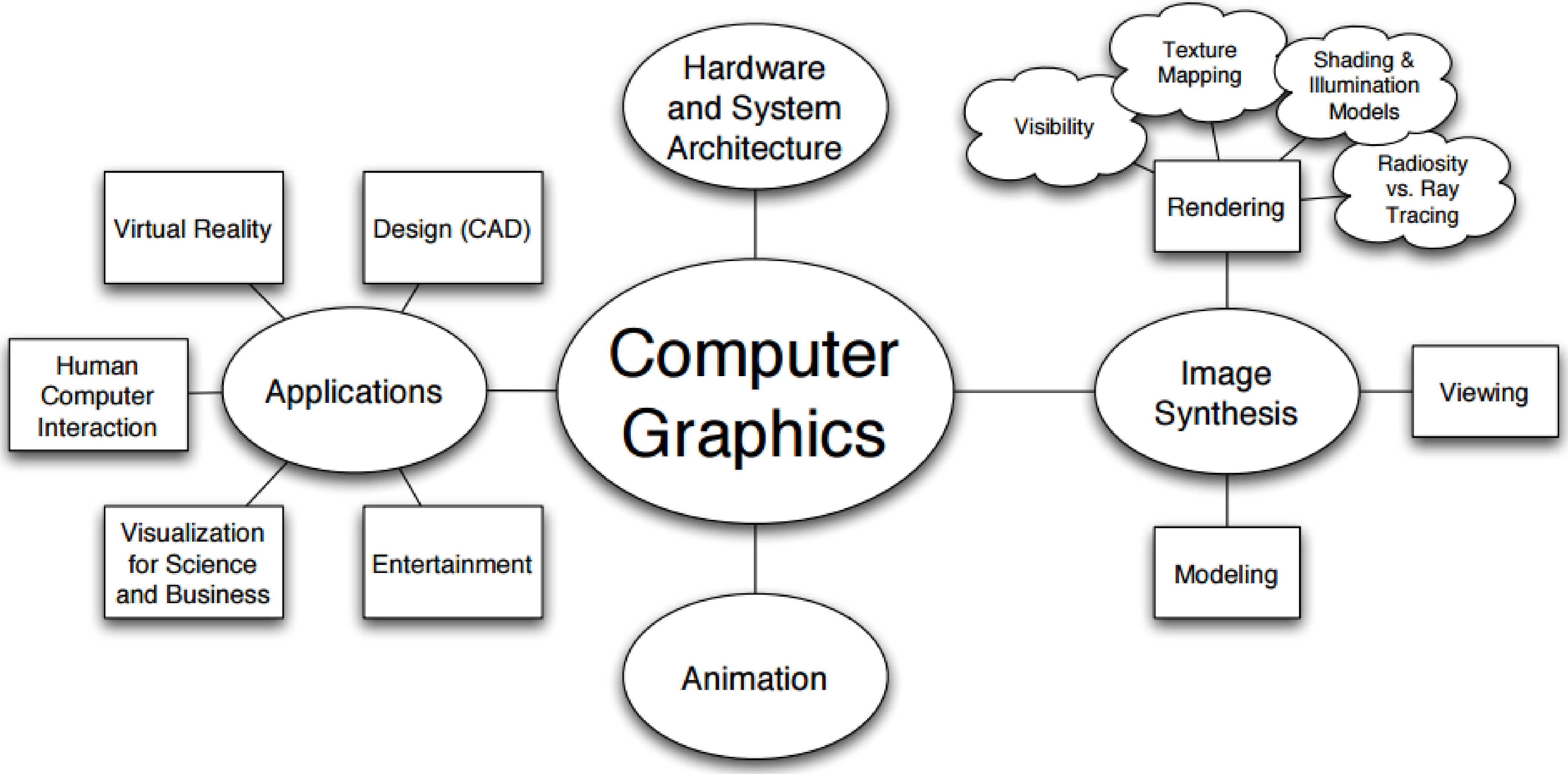


Story → Computer Graphics → Image

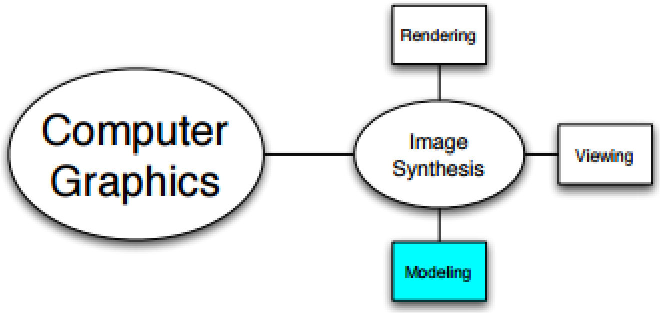
Related to many Disciplines



What Is Computer Graphics?

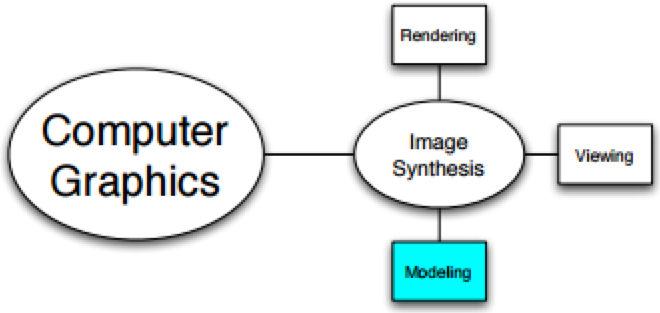

Computer Graphics
Hardware and System Architecture
Applications
Virtual Reality
Design (CAD)
Human Computer Interaction
Visualization for Science and Business
Entertainment
Animation
Image Synthesis
Rendering
Visibility
Texture Mapping
Shading & Illumination Models
Radiosity vs. Ray Tracing
Viewing
Modeling

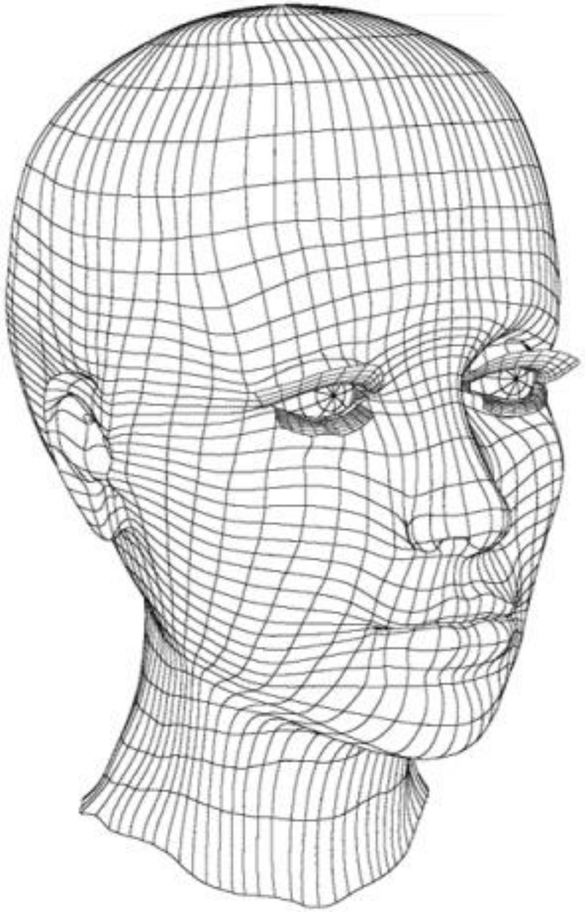
Modeling



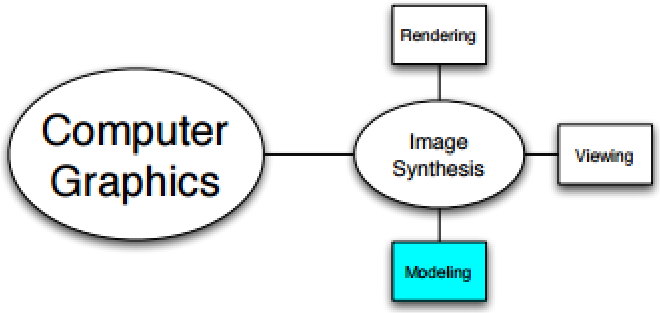
- How to represent real environments
    - Geometry: curves, surfaces, volumes
    - Photometry: light, color, reflectance
- How to build these representations
    - Interactive: sculpt it
    - Algorithmic: let it grow (fractals, extraction)
    - Scanning: via 3D sensing

Modeling: Interactive

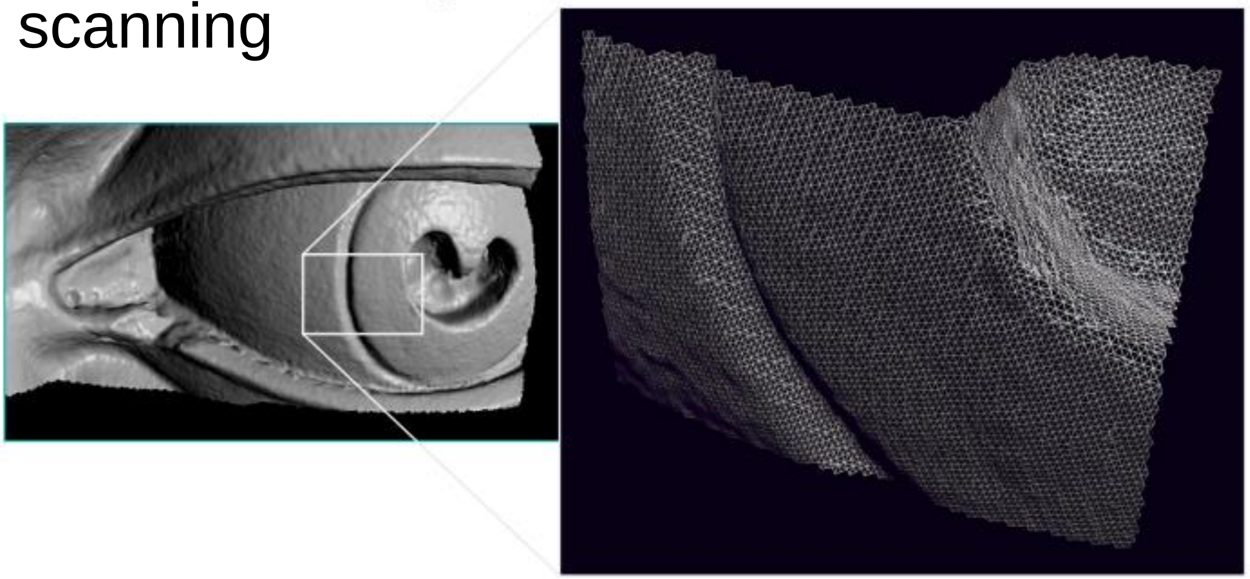




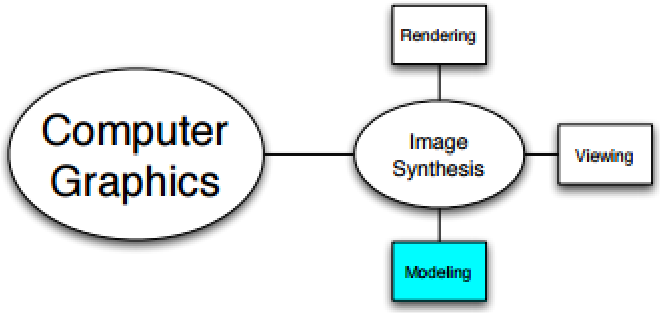
Modeling: Scanning

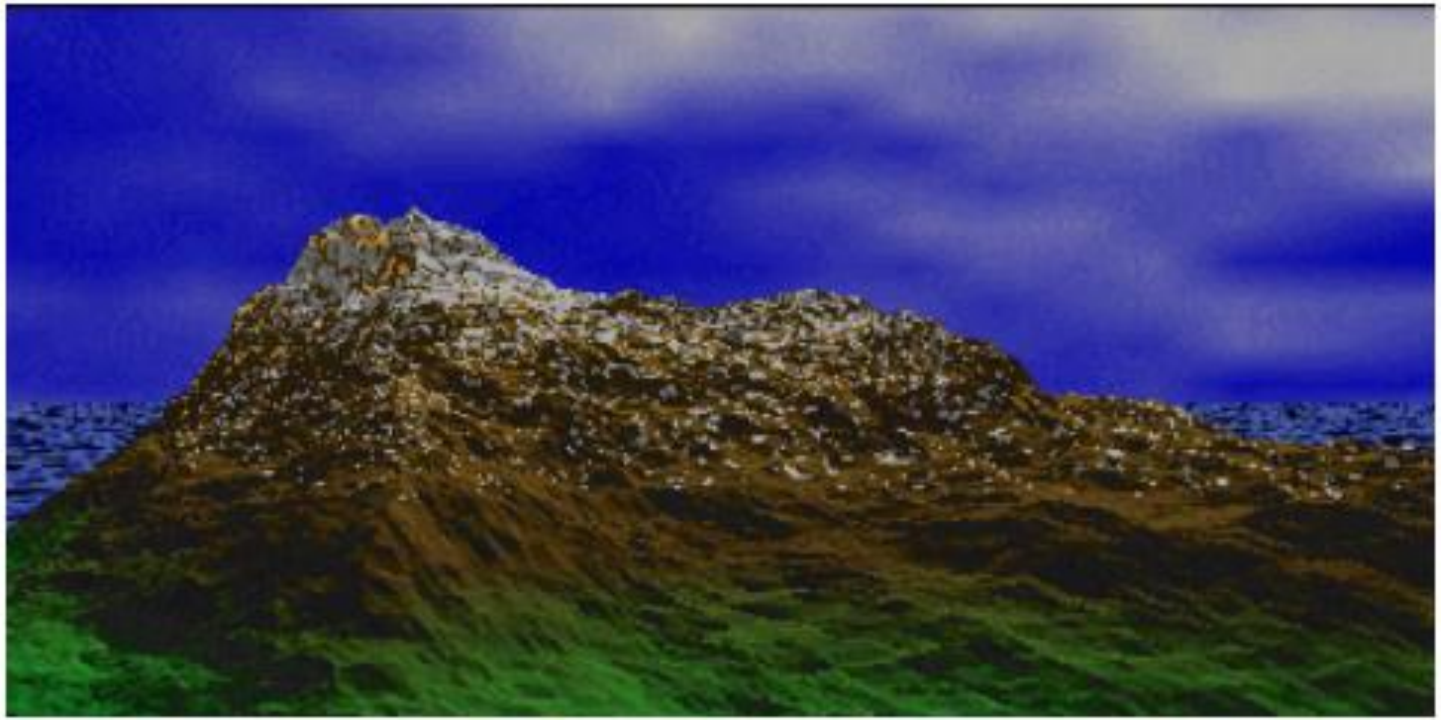


- David
  - 480 individually aimed scans
  - 2 billion polygons
  - 7,000 color images
  - 32 gigabytes
  - 30 nights of scanning
  - 22 people



Modeling: Algorithmic and Procedural

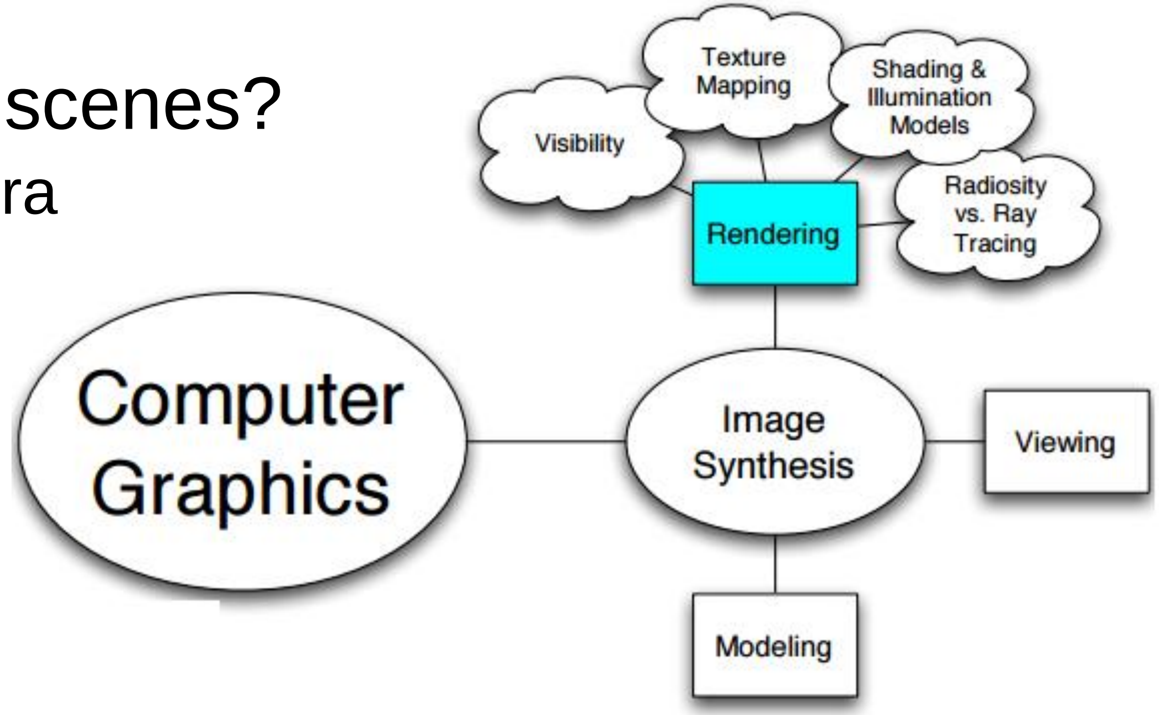




fractals

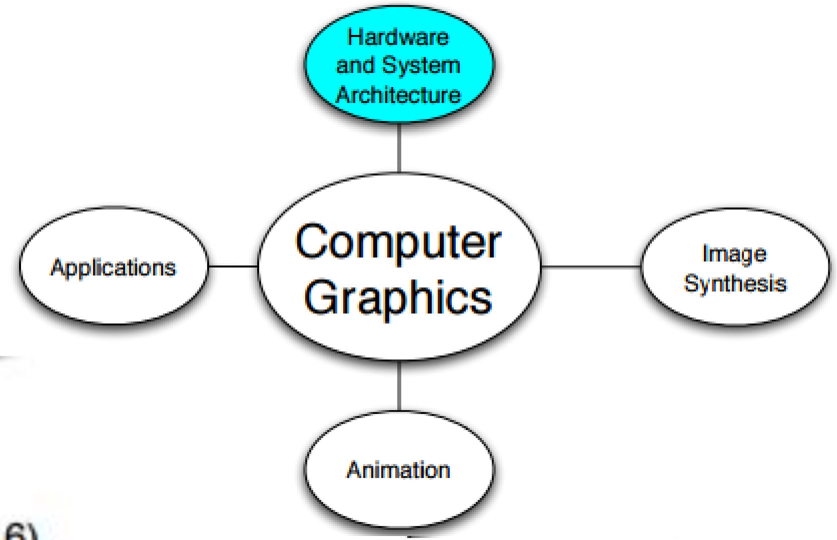
Rendering

- What is an image?
  - Distribution of light energy on 2D “film”
- How do we represent and store images?
  - Sampled array of “pixels”: p[x,y]
- How do we generate images from scenes?
  - Input: 3D description of scene, camera
  - Project to camera’s viewpoint
  - Illumination (position, direction, color, brightness)



Realistic lighting environments

Hardware


Hardware and System Architecture

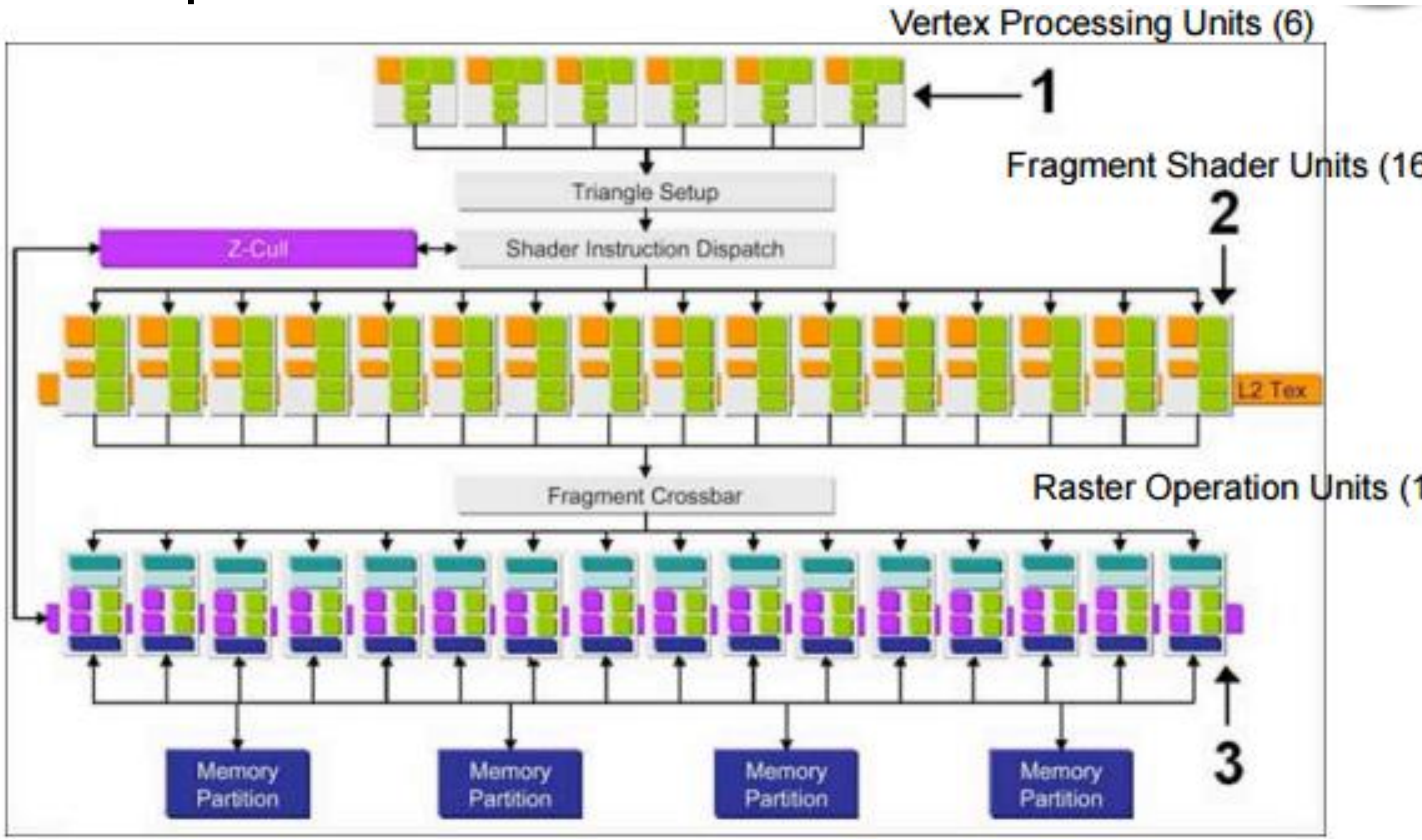
Applications

Computer Graphics

Image Synthesis

Animation

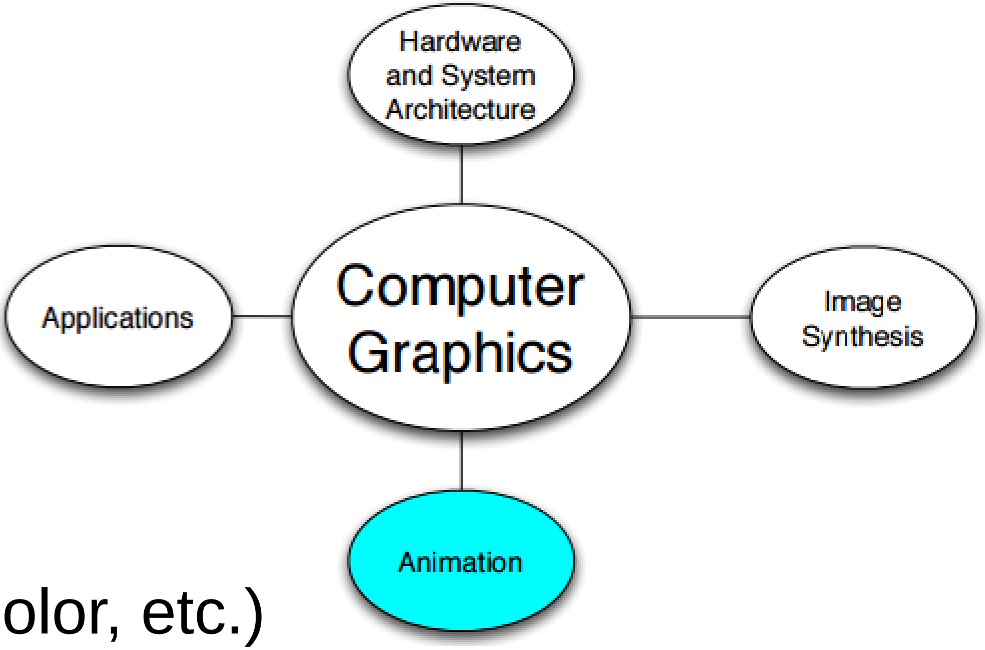
- Example: NVIDIA GeForce 6800



Game => High performance computing => Deep learning

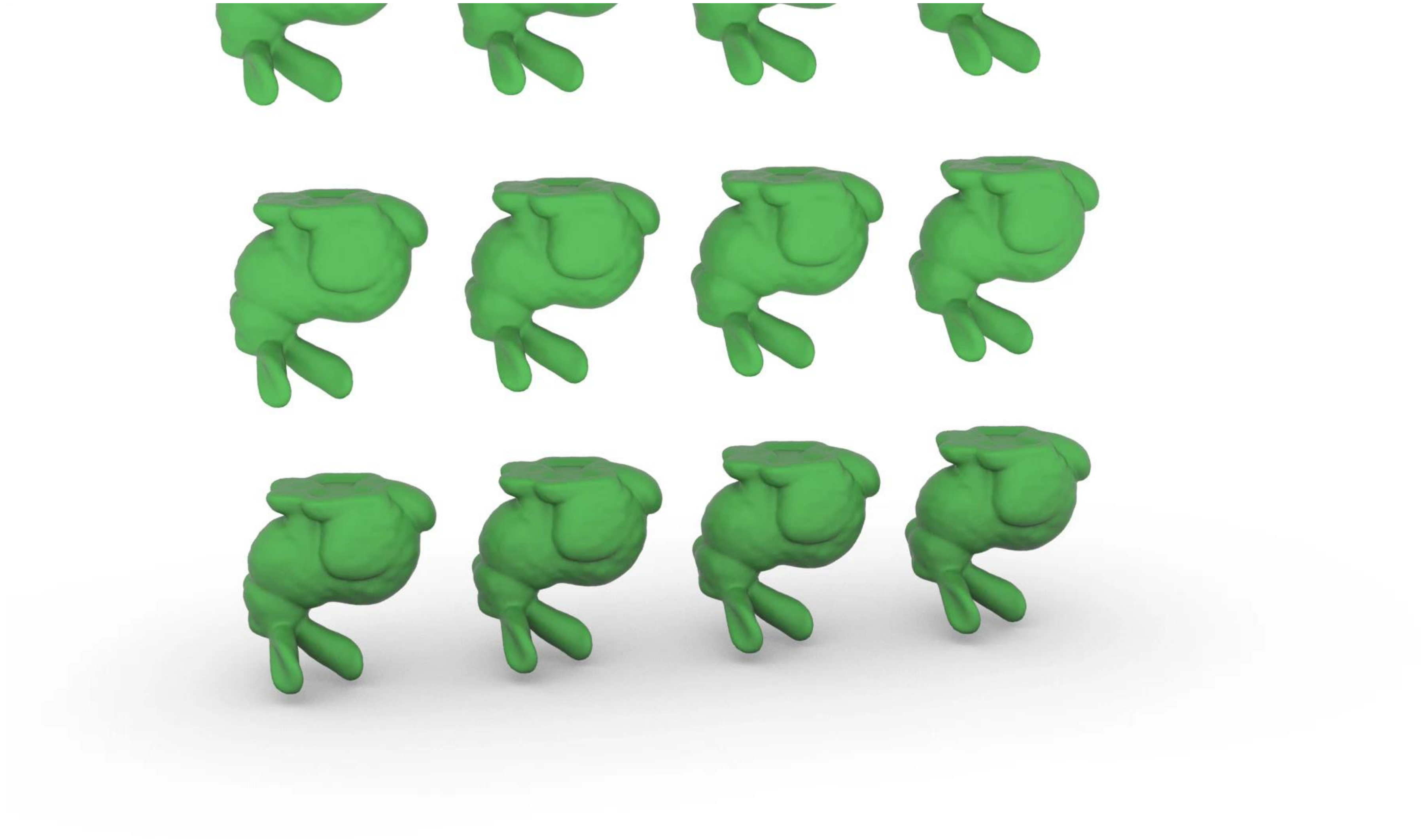
A watched flower never blooms, but an untended willow grows.

Animation



- Model how things move
- Temporal change of
  - Objects (position, orientation, size, shape, color, etc.)
  - Camera (position, direction, angle, focus, etc.)
  - Illumination (position, direction, color, brightness)
- Represent motion
  - Sequence of stills
  - Parameter curves

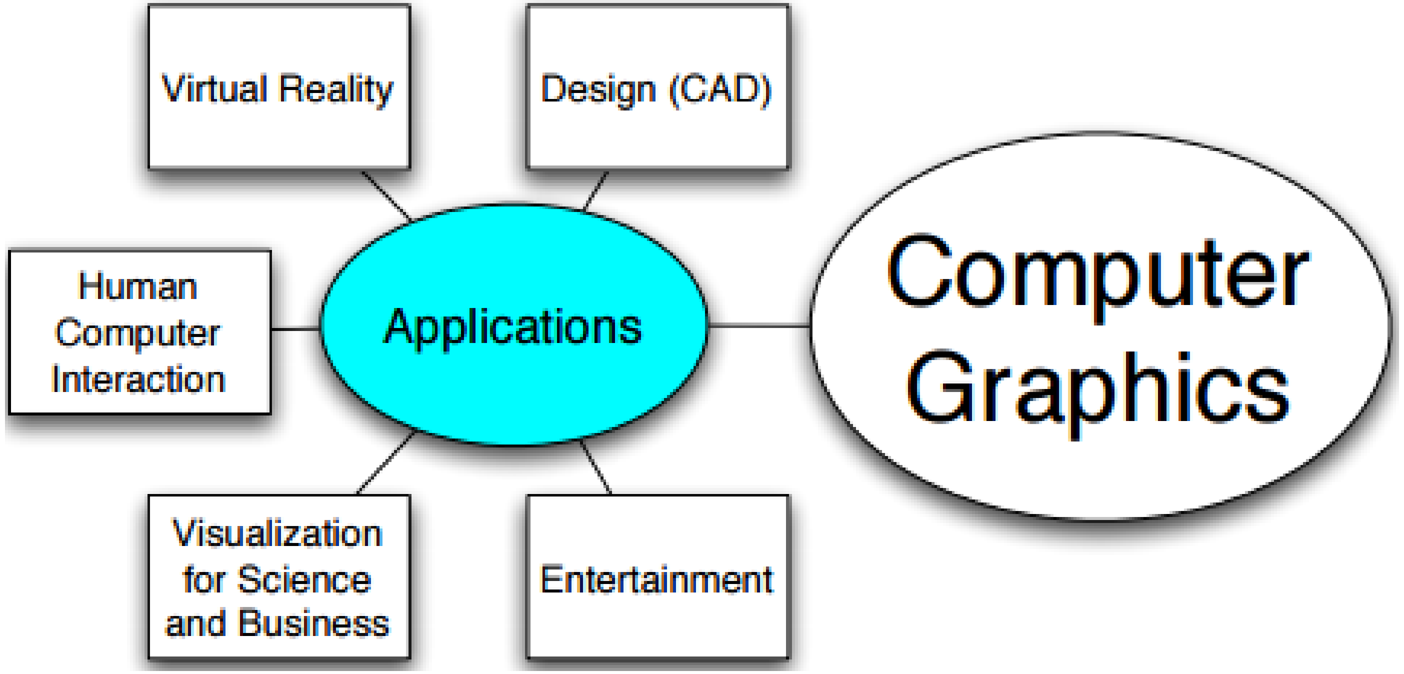
Physically-based simulation of motion



Physics + Computational Geometry + Animation + Ray Tracing

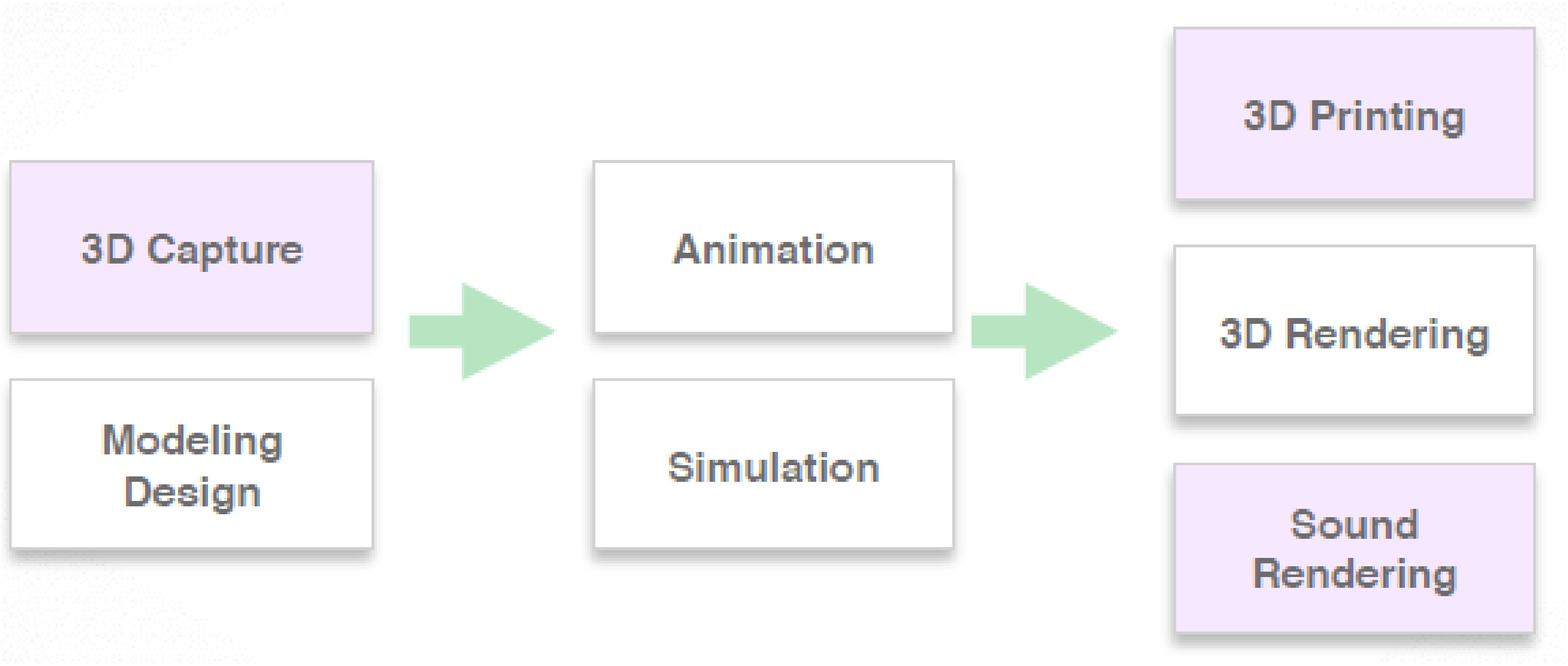
Barbic, James, SIGGRAPH 2010

Uses Of Graphics



- Special effects
- Feature animation
- Computer Games
- Virtual environments
- Visualization (science, business, cartography, ...)
- Design
- Interaction

3D Computer Graphics Pipeline



3D Capture

Modeling Design

Animation

Simulation

3D Printing

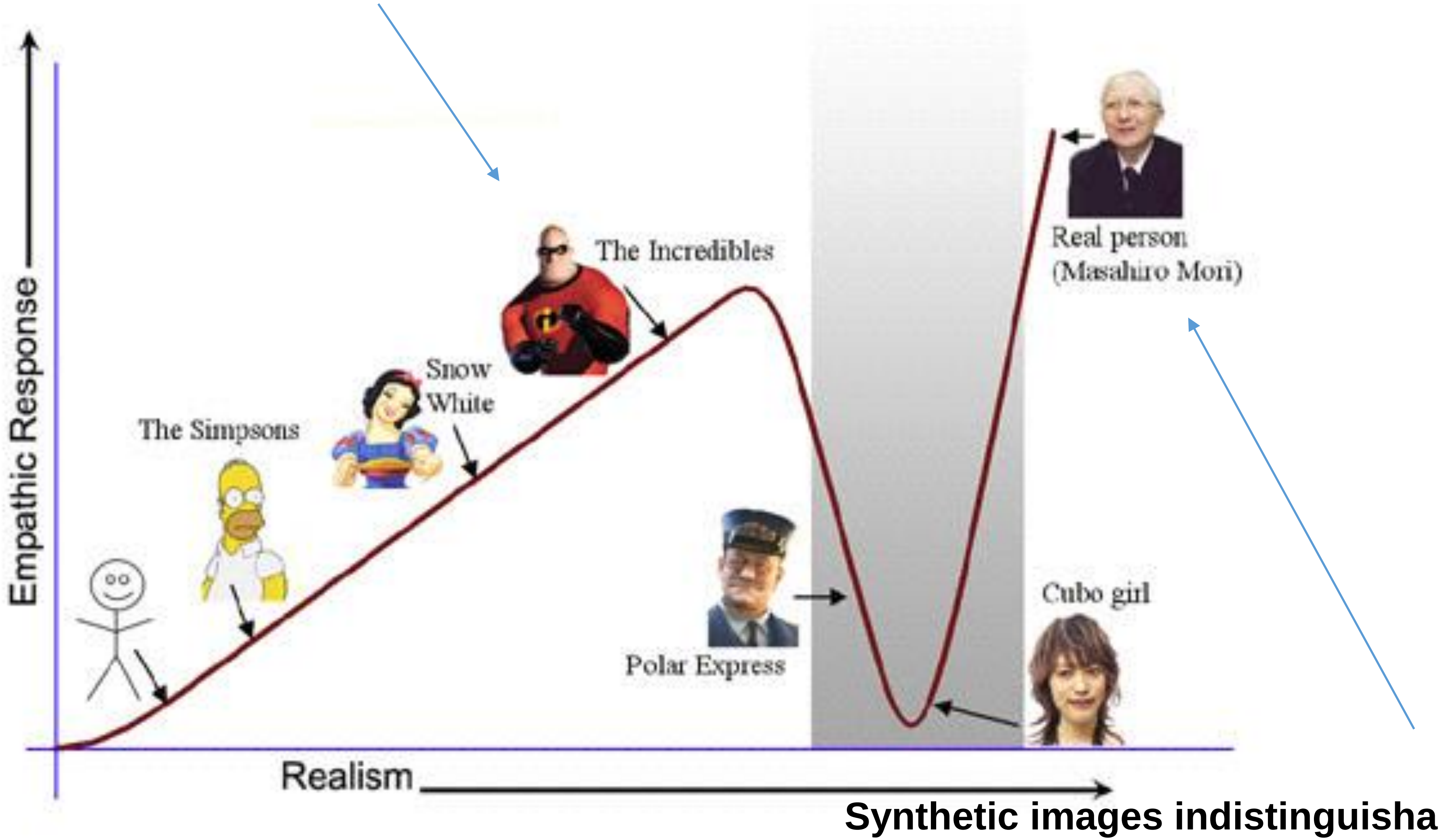
3D Rendering

Sound Rendering

Emerging Fields

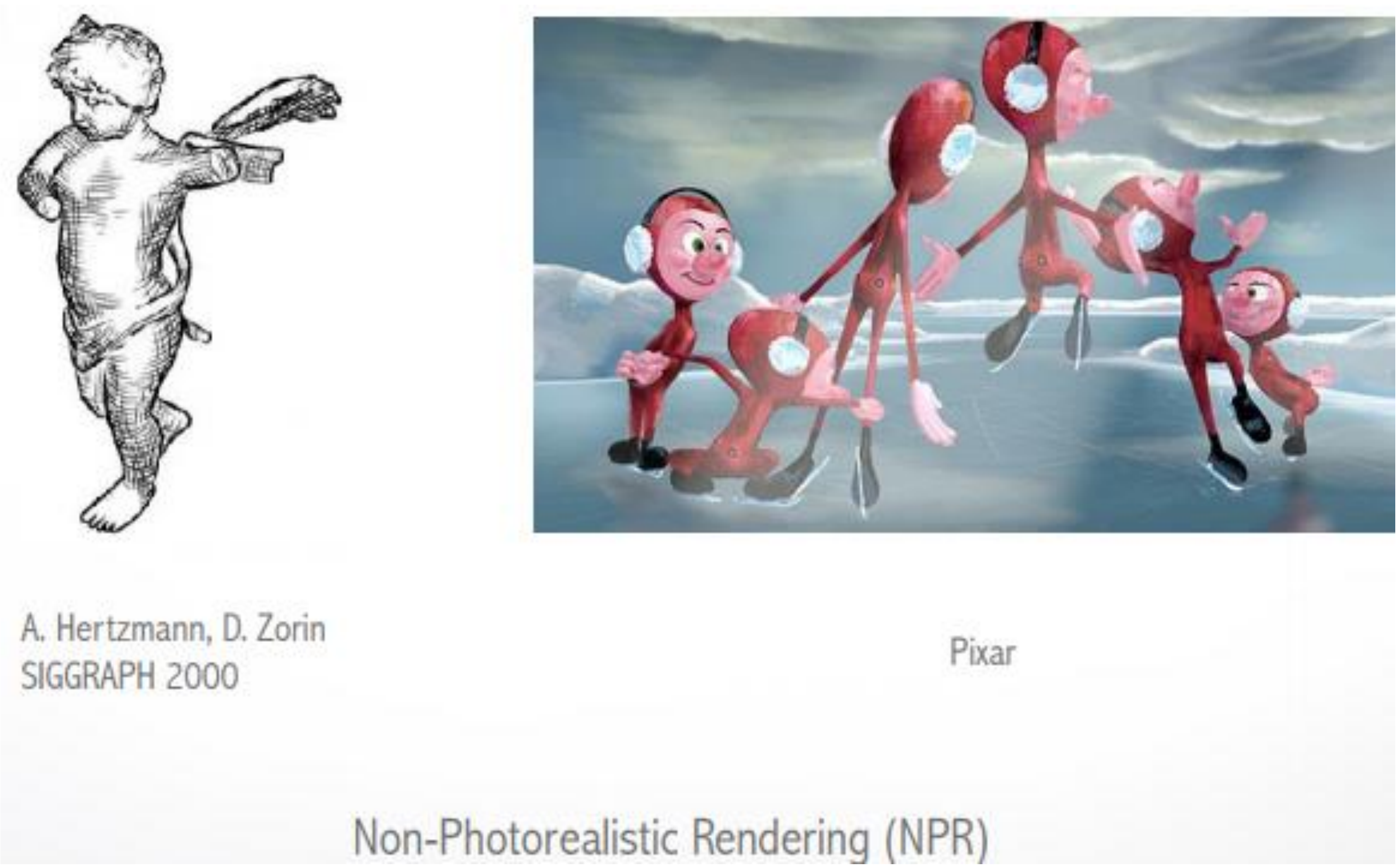
Goals in Computer Graphics

Creating a new reality (not necessarily scientific) Practical, aesthetically pleasing, in real time



Synthetic images indistinguishable from reality
Practical, scientifically sounds, in real time

Computing Illustrations



A. Hertzmann, D. Zorin
SIGGRAPH 2000

Pixar

Non-Photorealistic Rendering (NPR)

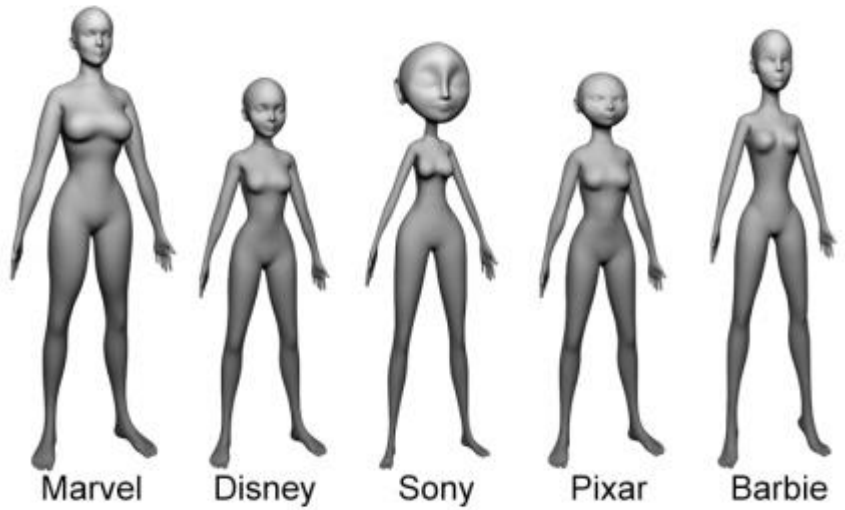


Figure 2: Style templates created from character reference.

Appealing female avatars from 3D body scans:
Perceptual effects of stylization, 2016

SIGGRAPH & SIGGRAPH Asia

ACMSIGGRAPH

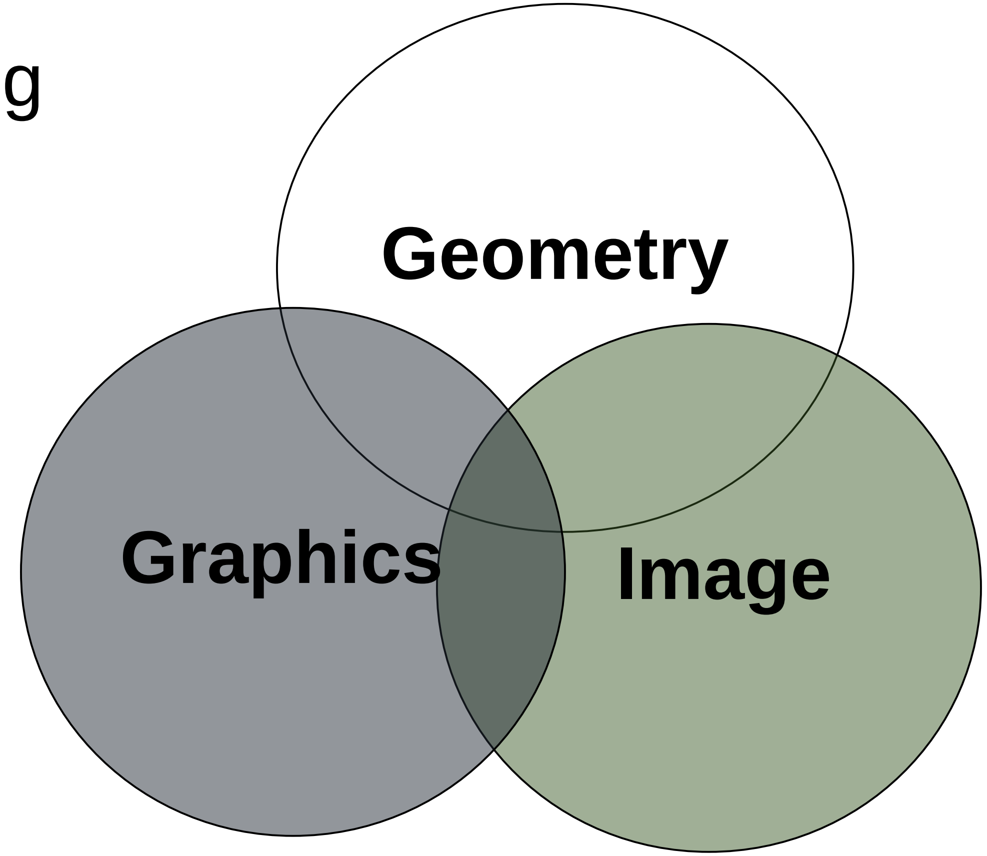
- Main computer graphics event
- Twice a year
- up to 30K attendees
- Academia, industry, artists

SIGGRAPH & SIGGRAPH Asia

- SIGGRAPH 2017 Technical Papers Preview Trailer

几何、图形、图像密不可分

- PDE method for Image processing
- Image interpolation
- Geometry Image
- Mesh filtering
- Segmentations
- Compression
- ……



Administrative Stuff

Course Information On-Line

- http://jjcao.github.io/ComputerGraphics/
  - Schedule (slides, readings)
  - Assignments (details, due dates)
  - Software (libraries, tutorial, links)

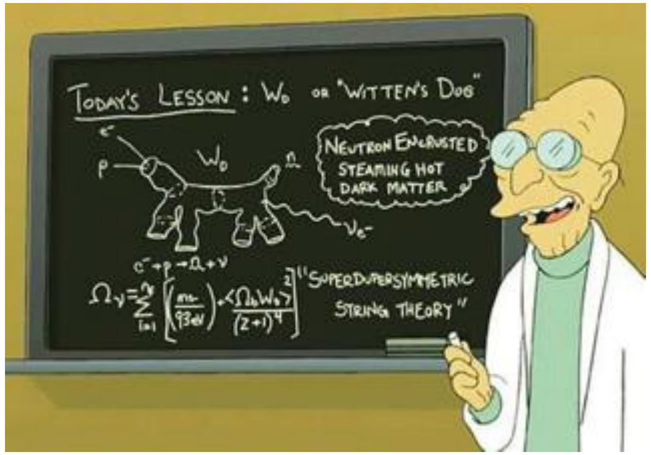
- https://piazza.com/
  - Submit assignments
  - Forum, Q/A

The team

- **Instructor**
  - **Junjie Cao, jjcao@dlut.edu.cn,** http://jjcao.github.io
- **Assistants**

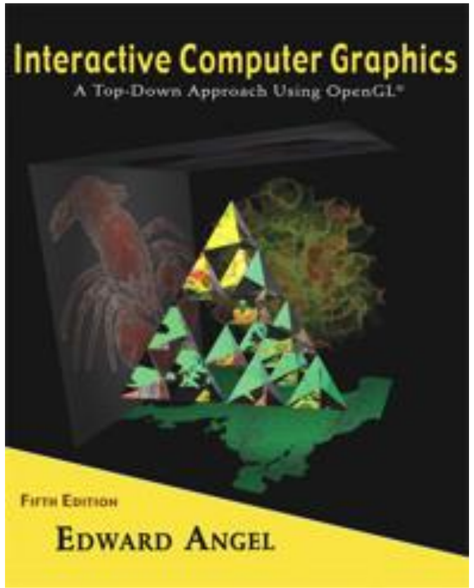
Prerequisites/What Is It I Expect?

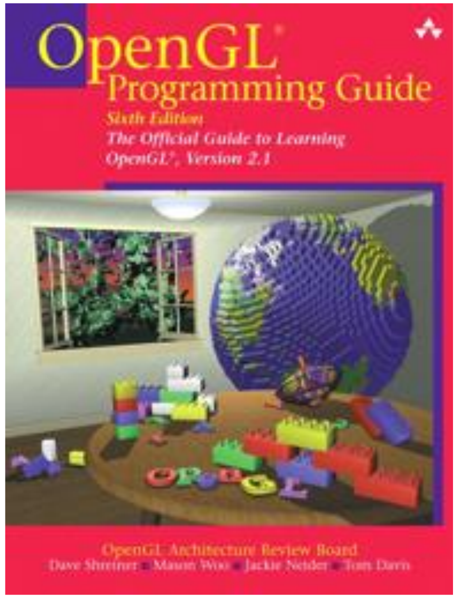
- Coding
  - C/C++
  - Preferably some previous OpenGL exposure
  - Data structures, algorithms
- Math
  - Linear Algebra
  - Differential Equations
- Keeping up with the text(s) is very important


TODAY'S LESSON : W₀ OR "WITTEN'S DOG"
NEUTRON ENCRUSTED STEAMING HOT DARK MATTER
W₀
"SUPERDUPERSYMMETRIC STRING THEORY"

Textbooks

- **Interactive Computer Graphics (“Angel”)**
  - A top-down approach with OpenGL, 6th Edition, Edward Angel, Addison-Wesley
- **OpenGL Programming Guide (“Red Book”)**


Interactive Computer Graphics
A Top-Down Approach Using OpenGL®
Fifth Edition
Edward Angel


OpenGL®
Programming Guide
Sixth Edition
The Official Guide to Learning
OpenGL®, Version 2.1
OpenGL Architecture Review Board
Dave Shreiner · Mason Woo · Jackie Neider · Tom Davis

Grading

- Classroom Test + Exercises 30%

- Assignments 70%: Document + Compilable code + Executable files **(Submit after deadline: -10%)**

  Assignment 1: 20 %

  Assignment 2: 25 %

  Assignment 3: 25 %

- Two students a team

Document in A4 & electronic: functions (required + optional)

RF1
Text description;
Code segment for the function
Image illustration;

RF2
Text description;
Code segment for the function
Image illustration;

OF1
Text description;
Code segment for the function
Image illustration;

Code in electronic:

- I can open *.sln and build it successfully and without modify setting and anything outside the folder.

- Compress whole folder into a zip
- Run packing.bat before compression

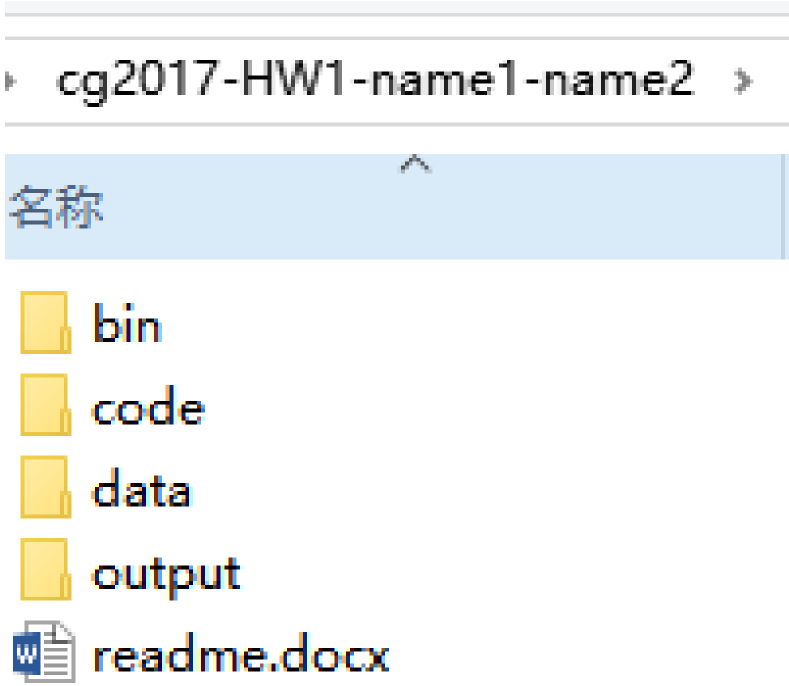
- Good function name and proper comments

Exe in electronic:

- A folder with exe, dll, and input data.

- Compress whole folder into a zip.

Example



cg2017-HW1-name1-name2

名称

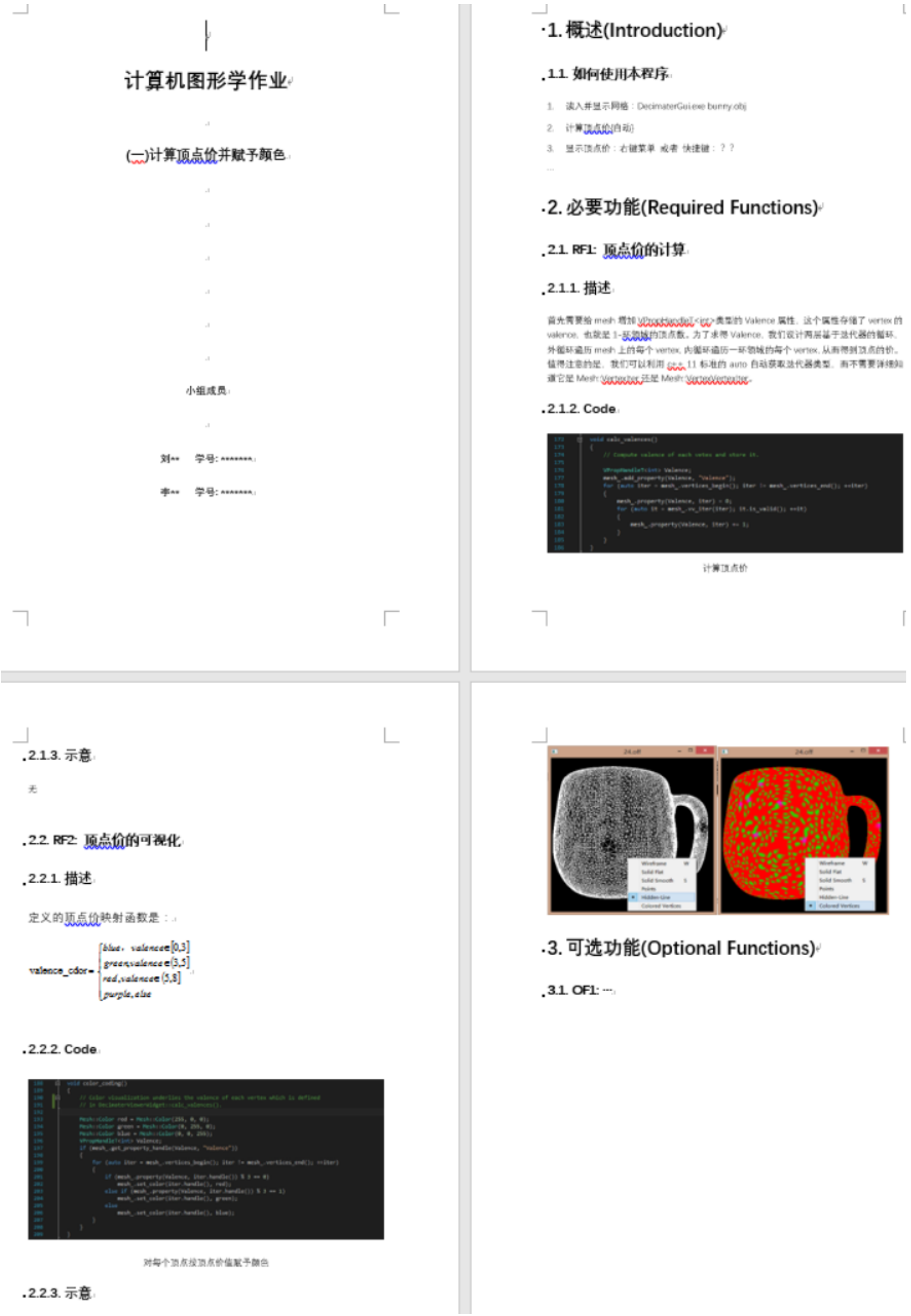
bin

code

data

output

readme.docx



计算机图形学作业

(一)计算顶点价并赋予颜色

小组成员

刘**　学号: *******

李**　学号: *******

1. 概述(Introduction)

1.1. 如何使用本程序

1. 读入并显示网格：DecimaterGui.exe bunny.obj
2. 计算顶点价(自动)
3. 显示顶点价：右键菜单 或者 快捷键：？？

…

2. 必要功能(Required Functions)

2.1. RF1: 顶点价的计算

2.1.1. 描述

首先需要给 mesh 增加 VPropHandleT<int>类型的 Valence 属性。这个属性存储了 vertex 的 valence，也就是 1-环领域的顶点数。为了求得 Valence，我们设计两层基于迭代器的循环。外循环遍历 mesh 上的每个 vertex，内循环遍历一环领域的每个 vertex，从而得到顶点的价。值得注意的是，我们可以利用 c++ 11 标准的 auto 自动获取迭代器类型，而不需要详细知道它是 Mesh::VertexIter 还是 Mesh::VertexVertexIter。

2.1.2. Code

```
void calc_valences()
{
    // Compute valence of each vetex and store it.

    VPropHandleT<int> Valence;
    mesh_.add_property(Valence, "Valence");
    for (auto iter = mesh_.vertices_begin(); iter != mesh_.vertices_end(); ++iter)
    {
        mesh_.property(Valence, iter) = 0;
        for (auto it = mesh_.vv_iter(iter); it.is_valid(); ++it)
        {
            mesh_.property(Valence, iter) += 1;
        }
    }
}
```

计算顶点价

2.1.3. 示意

无

2.2. RF2: 顶点价的可视化

2.2.1. 描述

定义的顶点价映射函数是：

$$\text{valence\_cdor}=\begin{cases} blue,\ valence\in[0,3] \\ green,valence\in[3,5] \\ red,valence\in(5,8] \\ purple,else \end{cases}$$

2.2.2. Code

```
void color_coding()
{
    // Color visualization underlies the valence of each vertex which is defined
    // in DecimaterViewerWidget::calc_valences().

    Mesh::Color red = Mesh::Color(255, 0, 0);
    Mesh::Color green = Mesh::Color(0, 255, 0);
    Mesh::Color blue = Mesh::Color(0, 0, 255);
    VPropHandleT<int> Valence;
    if (mesh_.get_property_handle(Valence, "Valence"))
    {
        for (auto iter = mesh_.vertices_begin(); iter != mesh_.vertices_end(); ++iter)
        {
            if (mesh_.property(Valence, iter.handle()) % 3 == 0)
                mesh_.set_color(iter.handle(), red);
            else if (mesh_.property(Valence, iter.handle()) % 3 == 1)
                mesh_.set_color(iter.handle(), green);
            else
                mesh_.set_color(iter.handle(), blue);
        }
    }
}
```

对每个顶点按顶点价值赋予颜色

2.2.3. 示意

3. 可选功能(Optional Functions)

3.1. OF1: ---

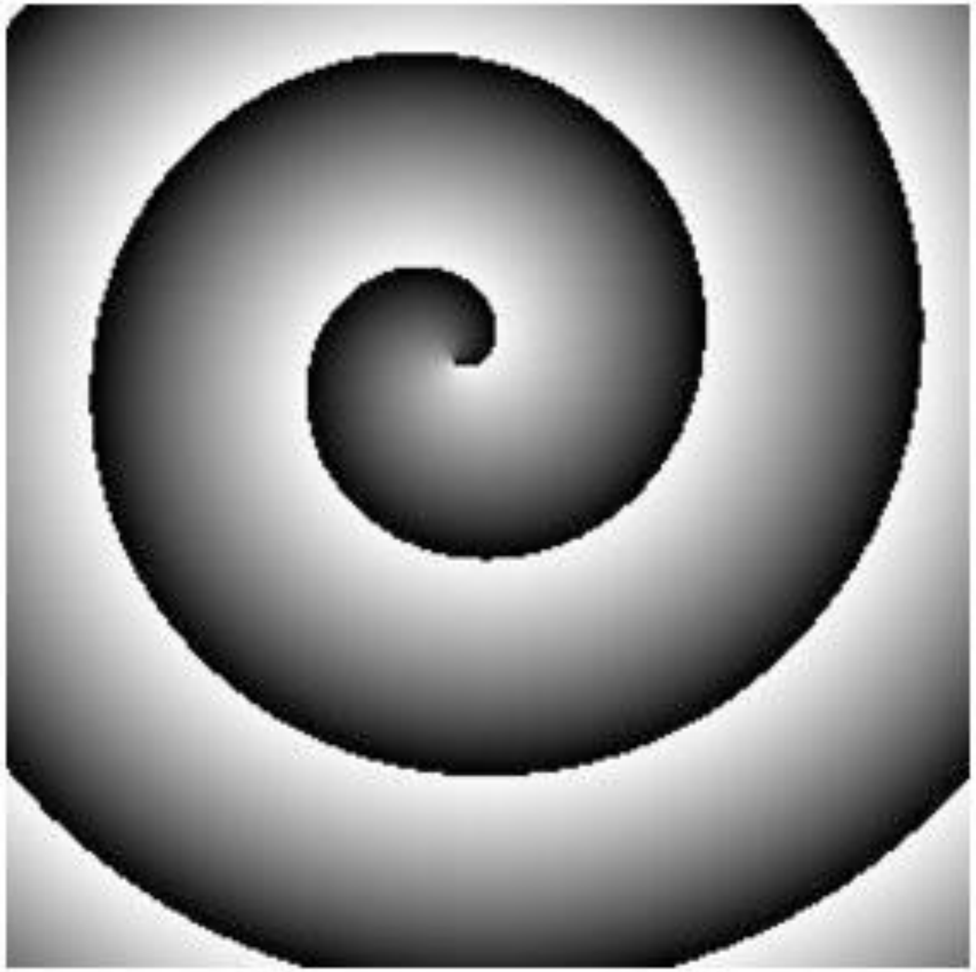
Academic Integrity

- Do not copy any parts of the assignments from anyone
- Do not look at other student’s code
- Collaboration only for the project
- Don’t cheat, okay?

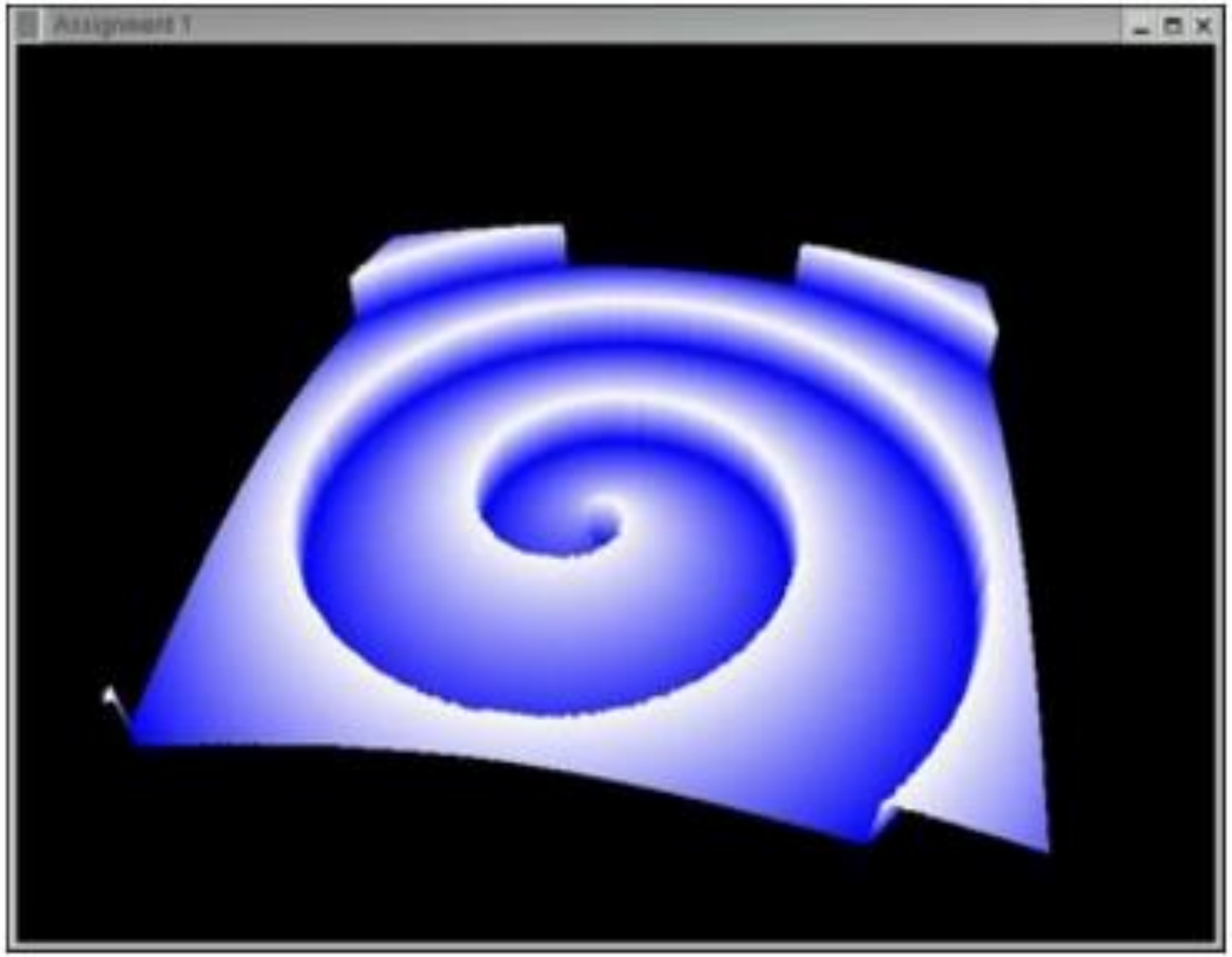
Assignment Policies

- Programming Assignments
  - Hand in via Piazza
  - Functionality and features
  - Style and documentation
  - Artistic impression
- Academic integrity policy applied rigorously

Assignment 1 – Height field

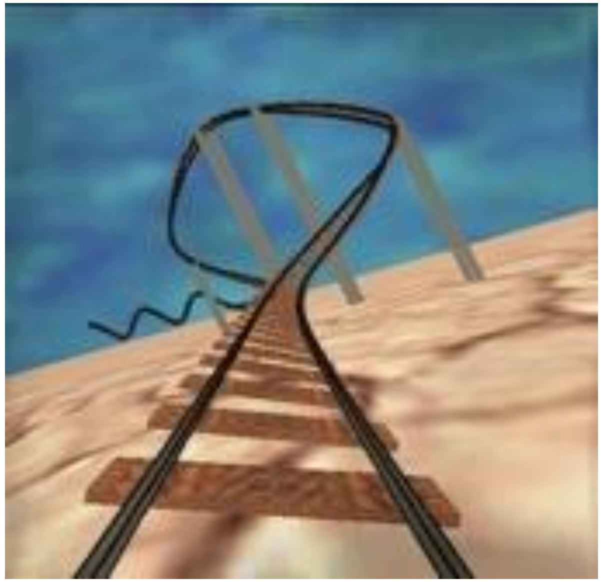


input (source image)

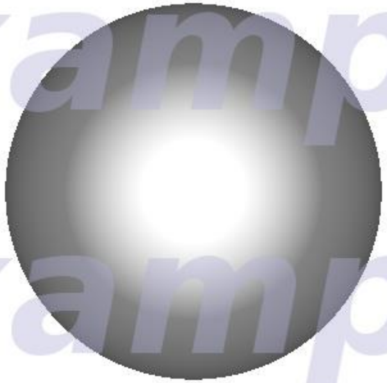


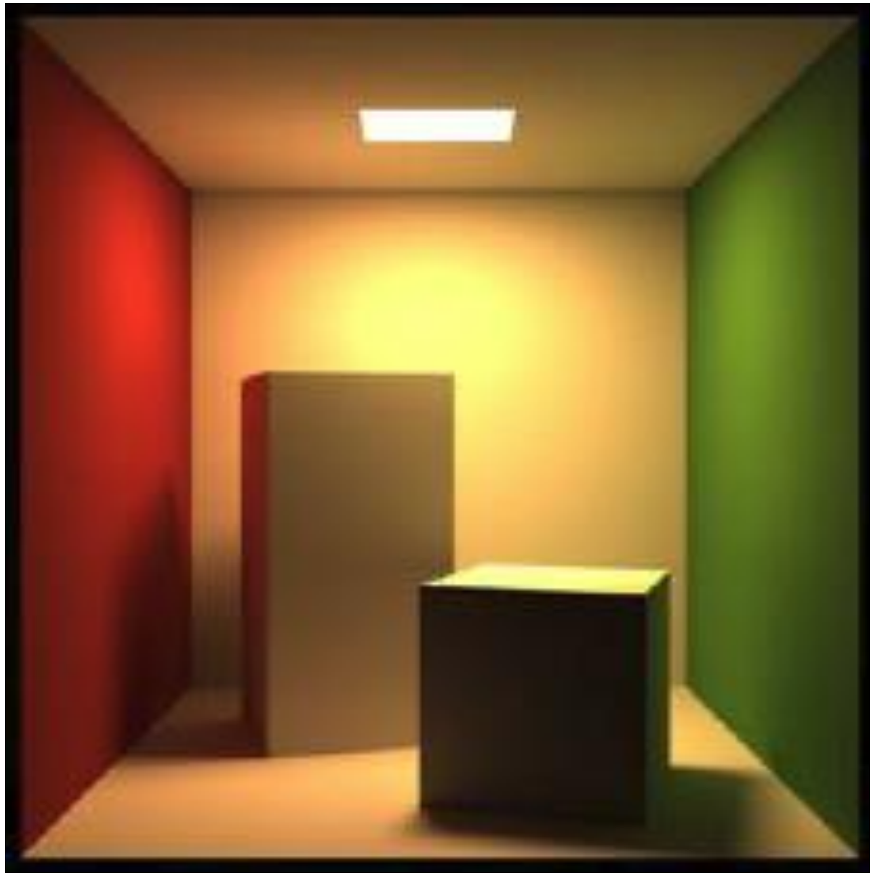
output (height field)

Assignment 2 – **Simulating a Roller Coaster**



Assignment 3 – Ray tracing





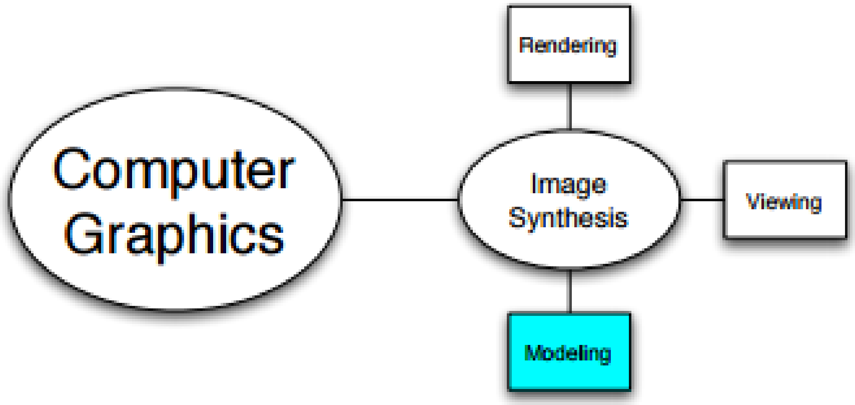
Other

- 留一个联系人，确定上机时间。

Introduction

- What is Computer Graphics?
  - Applications
  - History
  - Relations with other Disciplines
- Administrative Stuff
- **Course Overview**
- Research Trends

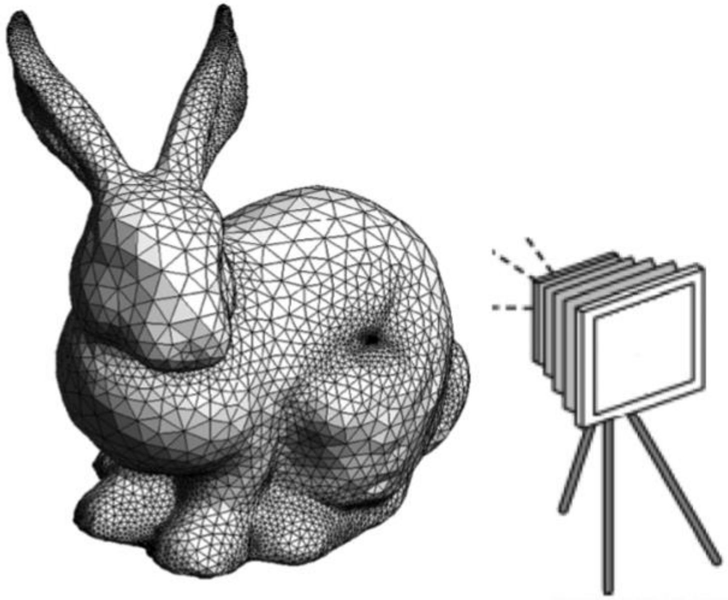
Topics / Course Overview

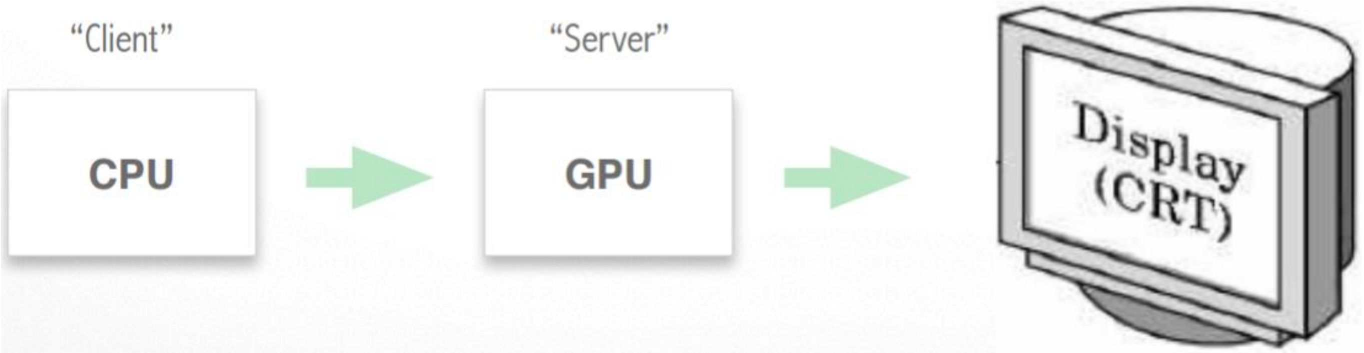


- **Theory / Computer Graphics Disciplines**
  - Image Processing: how to edit images
  - Modeling: how to represent objects
  - Rendering: how to create images of objects
  - Animation: how to control and represent motion
- **Practice: OpenGL graphics library**

OpenGL Basics

- Primitives and attributes
  - Text & fonts
- Color
- Viewing
- Control functions
  - Clients & servers
  - Event driven programming
- [Angel, Ch. 2]





“Client”

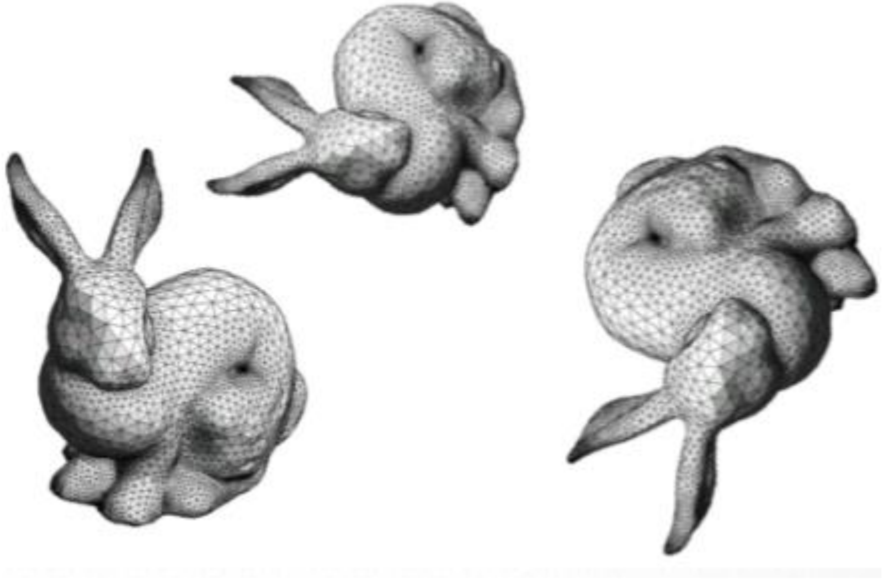
CPU

“Server”

GPU

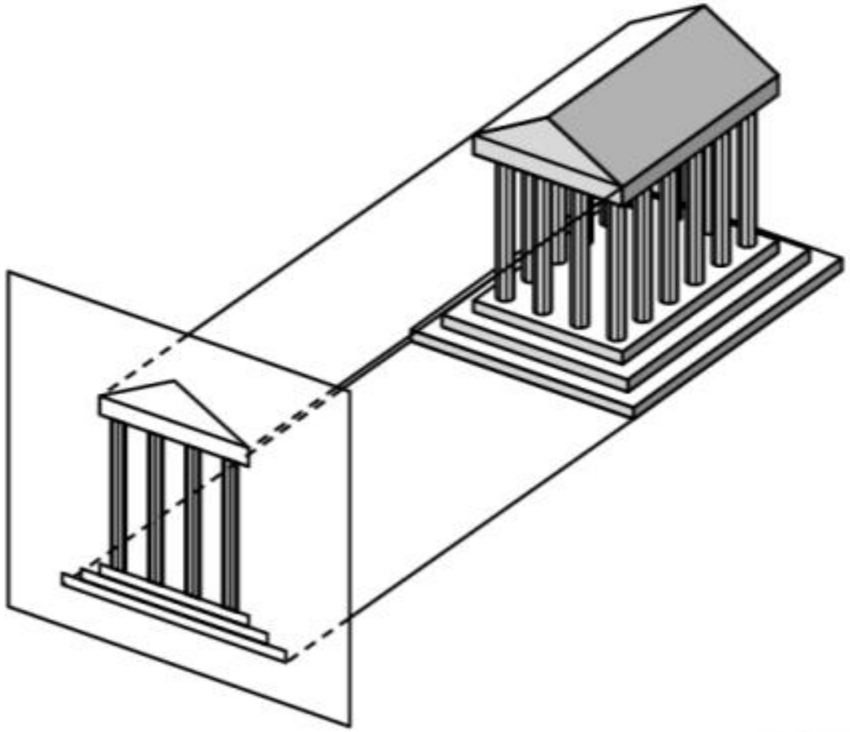
Objects & Transformations

- Linear algebra review
- Coordinate systems and frames
- Rotation, translation, scaling
- Homogenous coordinates
- OpenGL transformations
- [Angel, Ch. 3]



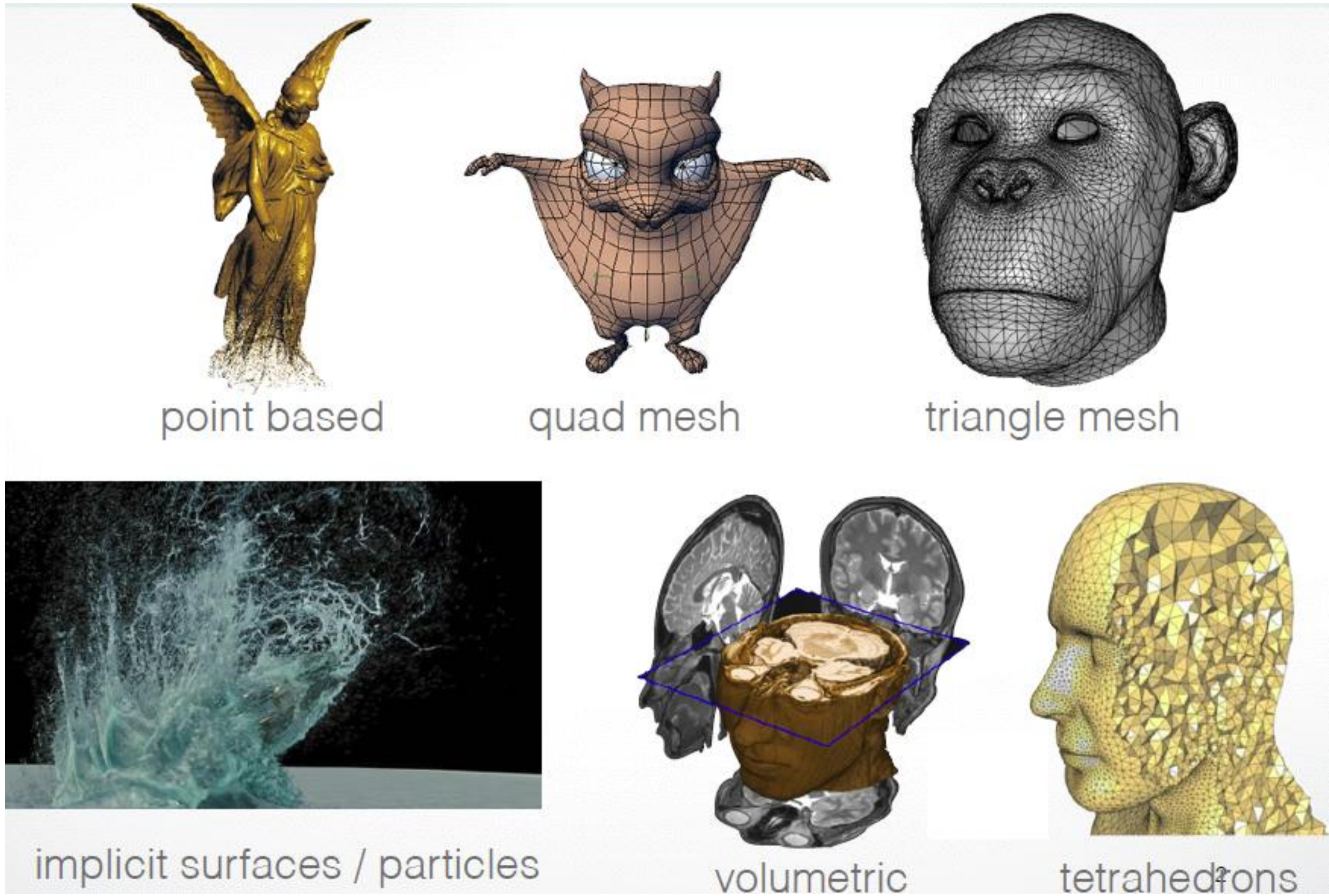
Viewing and Projection

- Orthographic projection
- Perspective projection
- Camera positioning
- Projection in OpenGL
- Hidden surface removal
- [Angel, Ch. 4]



Curves & Surfaces

- Recall 3D calculus
- Explicit representation: triangular mesh
- Implicit representation
- Parametric curves & surfaces
    - Hermite curves and surfaces
    - Bézier curves and surfaces
    - Splines
- Curves and surfaces in OpenGL
- [Angel, Ch. 10]



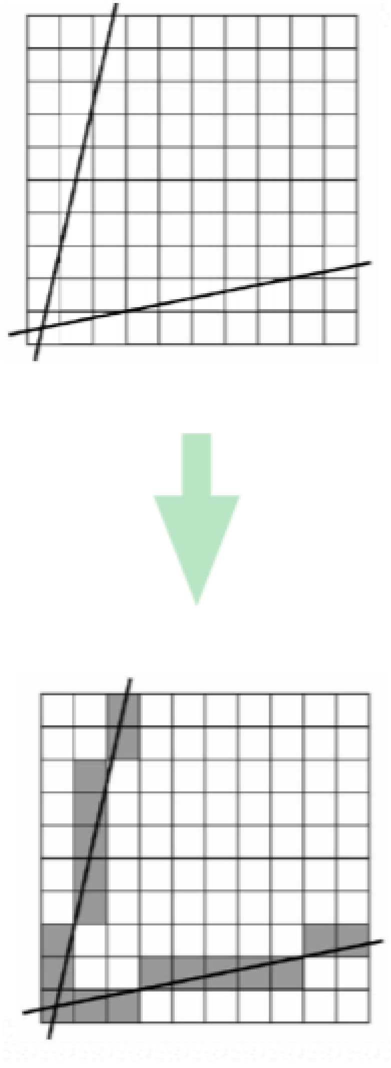
Light & Shading

- Light sources
- Ambient, diffuse, and specular reflection
- Normal vectors
- Material properties in OpenGL
- Radiosity
- [Angel, Ch. 5]

Tobian R. Metoc

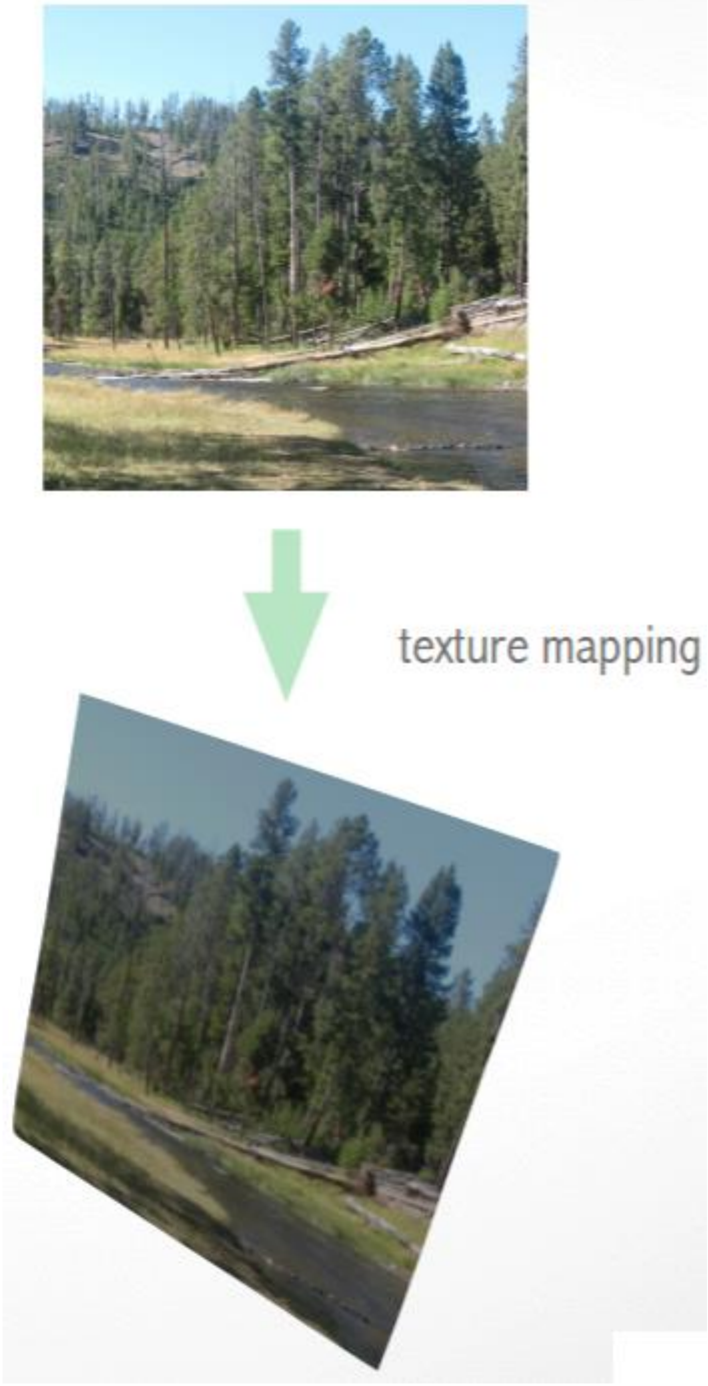
Rendering

- Clipping
- Bounding boxes
- Hidden-surface removal
- Line drawing
- Scan conversion
- Anti-aliasing
- [Angel, Ch. 6]



Textures and Pixels

- Texture mapping
- OpenGL texture primitives
- Bump maps
- Environment maps
- Opacity and blending
- Image filtering
- [Angel, Ch. 7]



Hierarchical Models

- Re-using objects
- Animations
- OpenGL routines
- Parameters and transformations
- [Angel, Ch. 8]

Advanced rendering - Ray Tracing

- Basic ray tracing [Angel, Ch. 11]
- Motion blur
- Soft shadows
- Local vs global illumination
- Interreflections
- Radiosity equation
- Solution methods

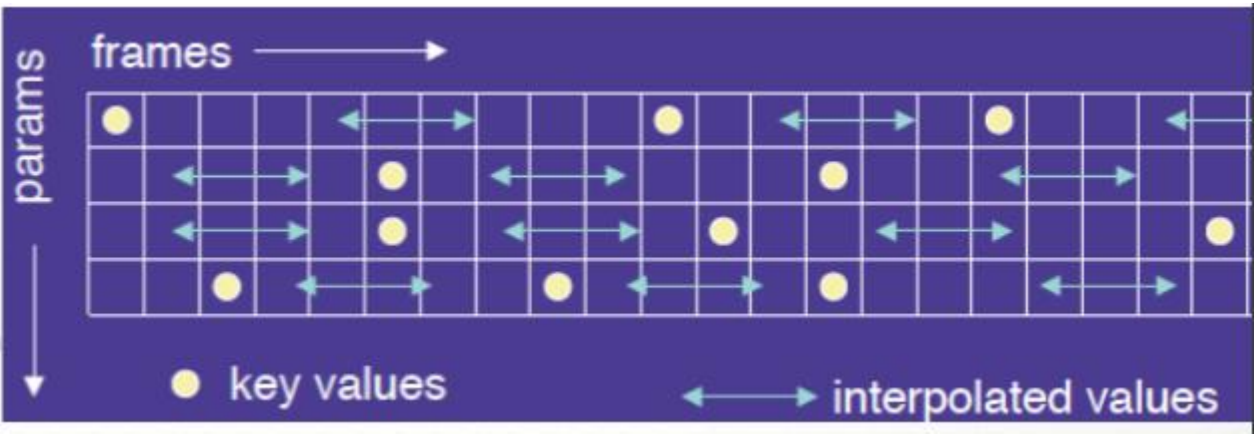
More Advanced Rendering - Radiosity

- Local vs global illumination
- Interreflections
- Radiosity equation
- Solution methods
- [Angel Ch. 13.4-5]

Animation

- Traditional Animation
- Keyframe Animation
- Computer Animation

ANIMATOR: Ham Luske—

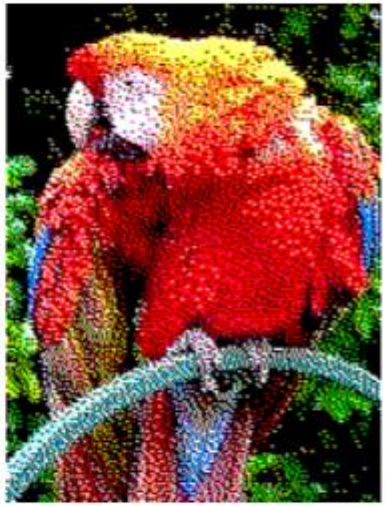


Physically Based Models

- Particle systems
- Spring forces
- Cloth
- Collisions
- Constraints
- Fractals
- [Angel, Ch. 9]

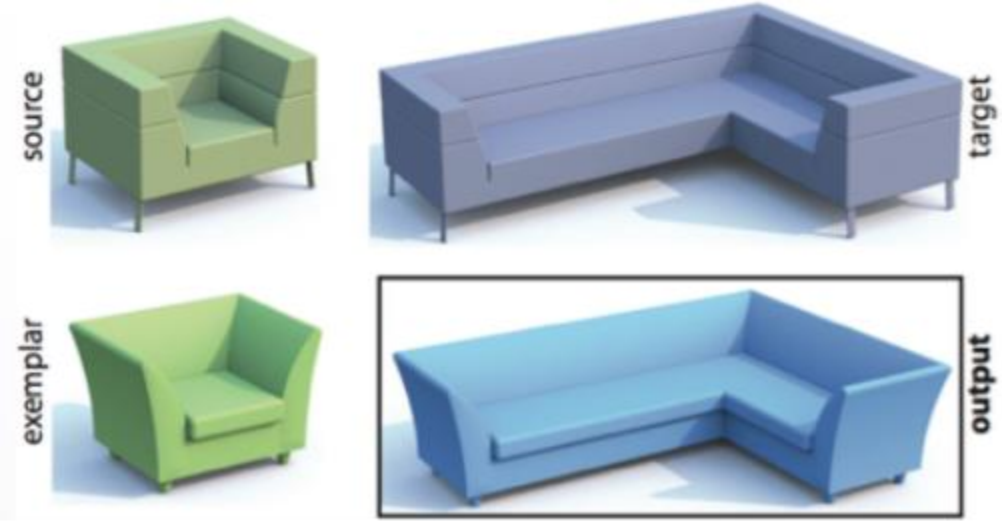
Image Processing

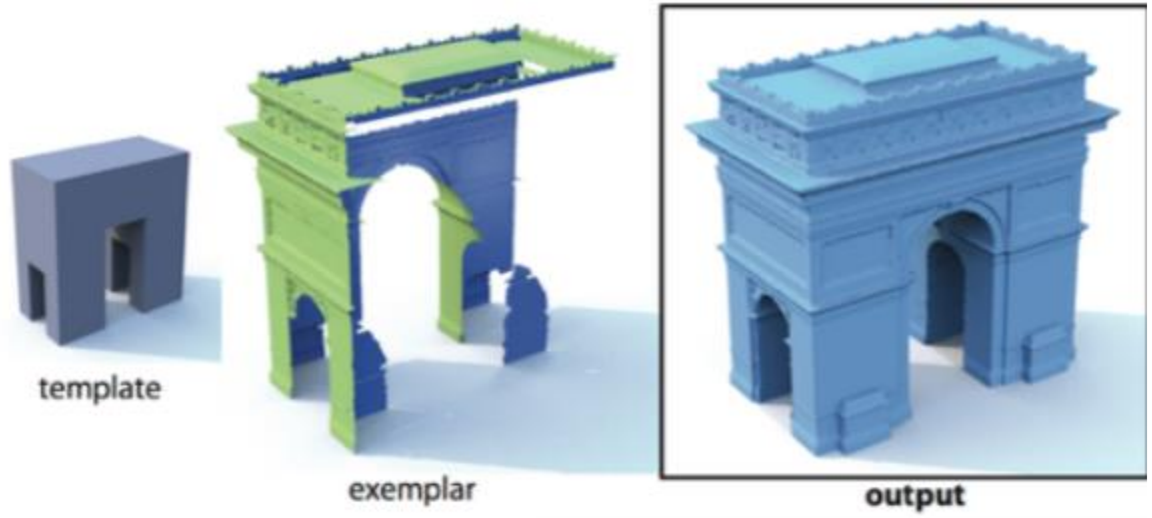
- Filters
- Dithering
- Blending
- Display Color Models



“Wildcard” Lectures

- Realtime 3D Reconstruction
- Geometry Processing
- Graphics & Machine Learning
- Data-Driven Modeling
- …



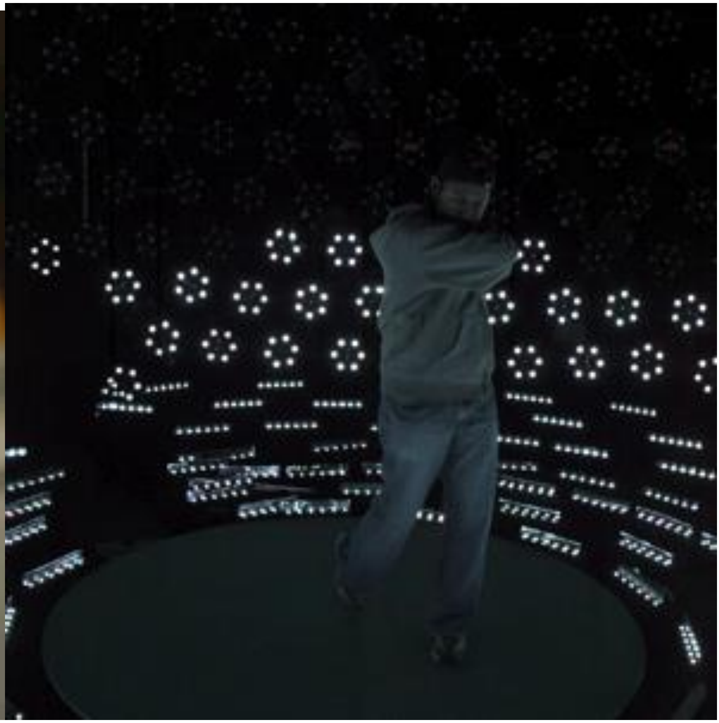


Trends

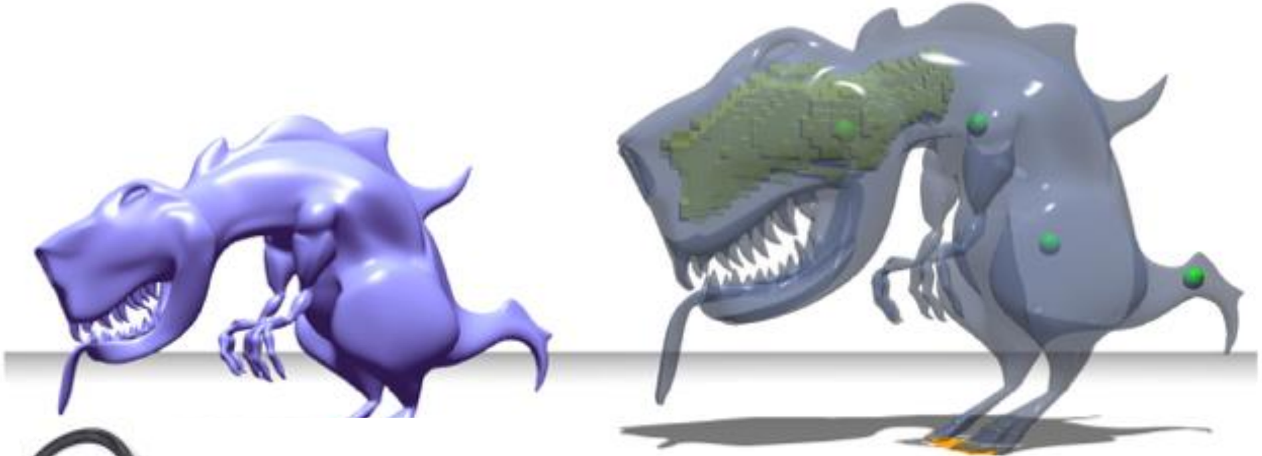
From Offline to Realtime

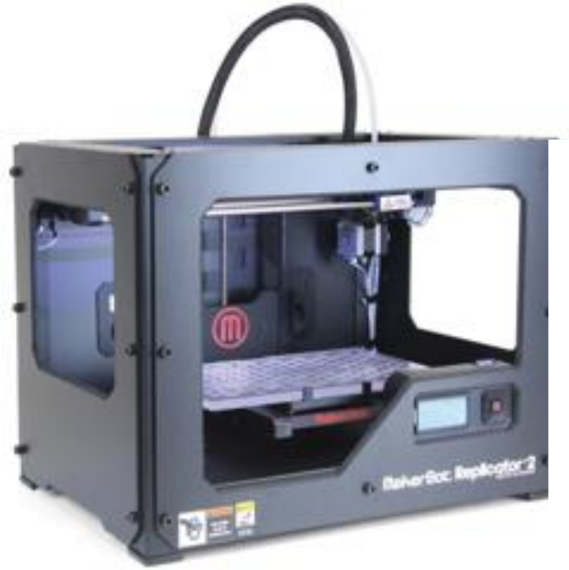
Unreal Engine Kite Demo (Epic Games 2015)

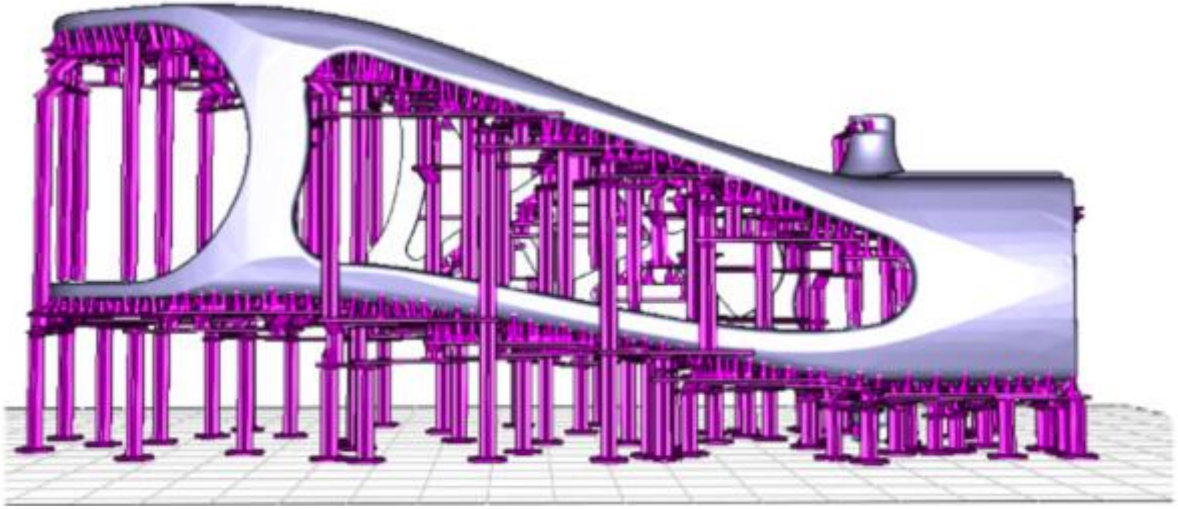
From Graphics to Vision



From Graphics to Fabrication







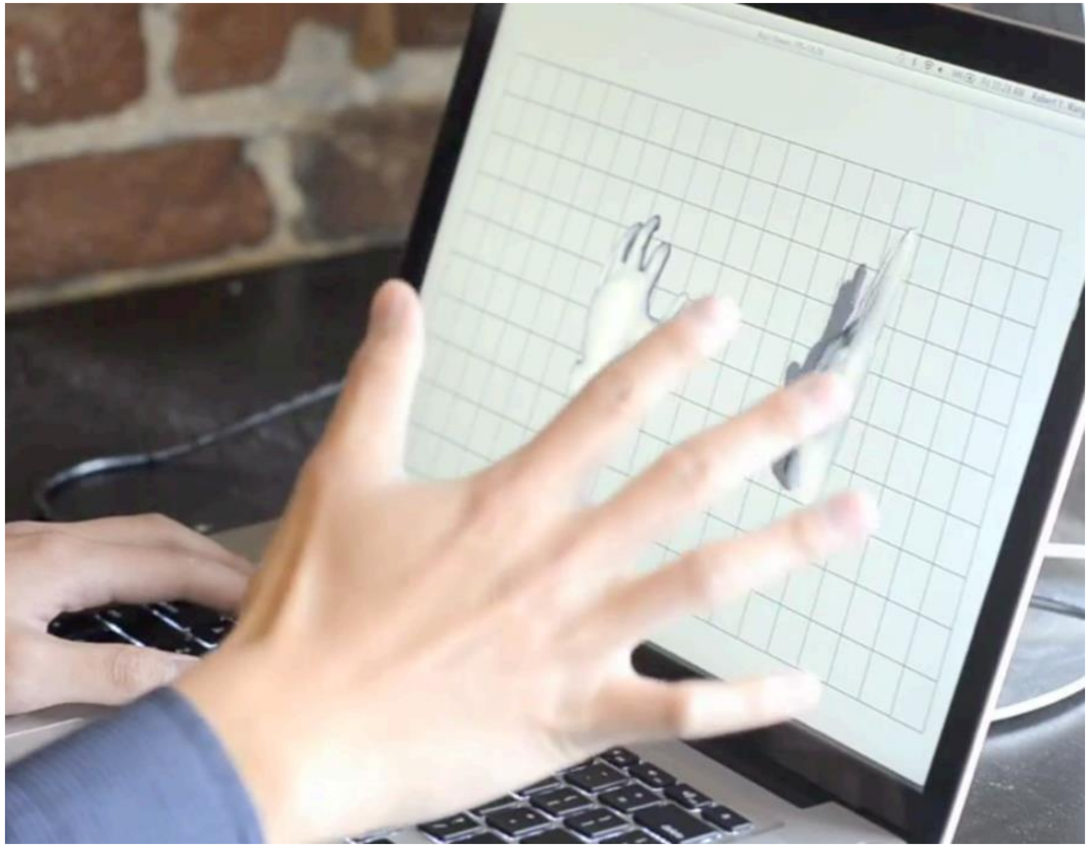
From Production to Consumers

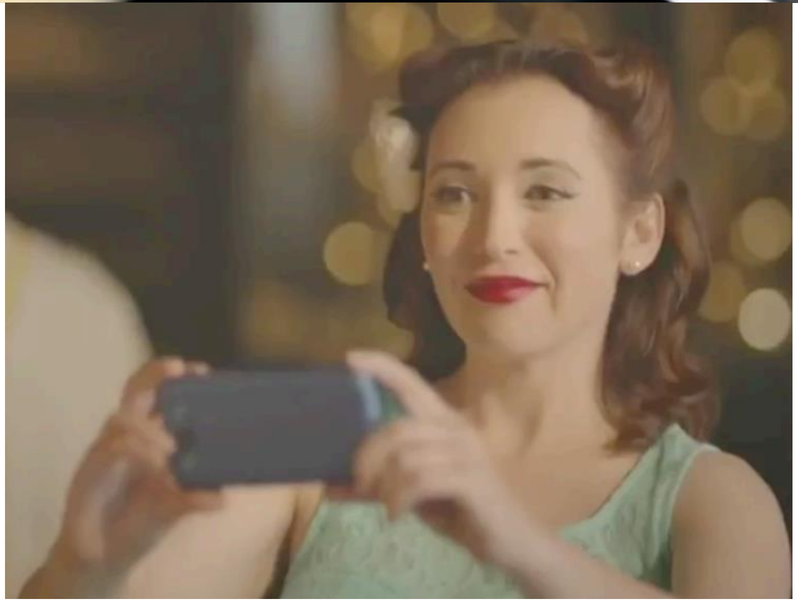
VFX

online shopping

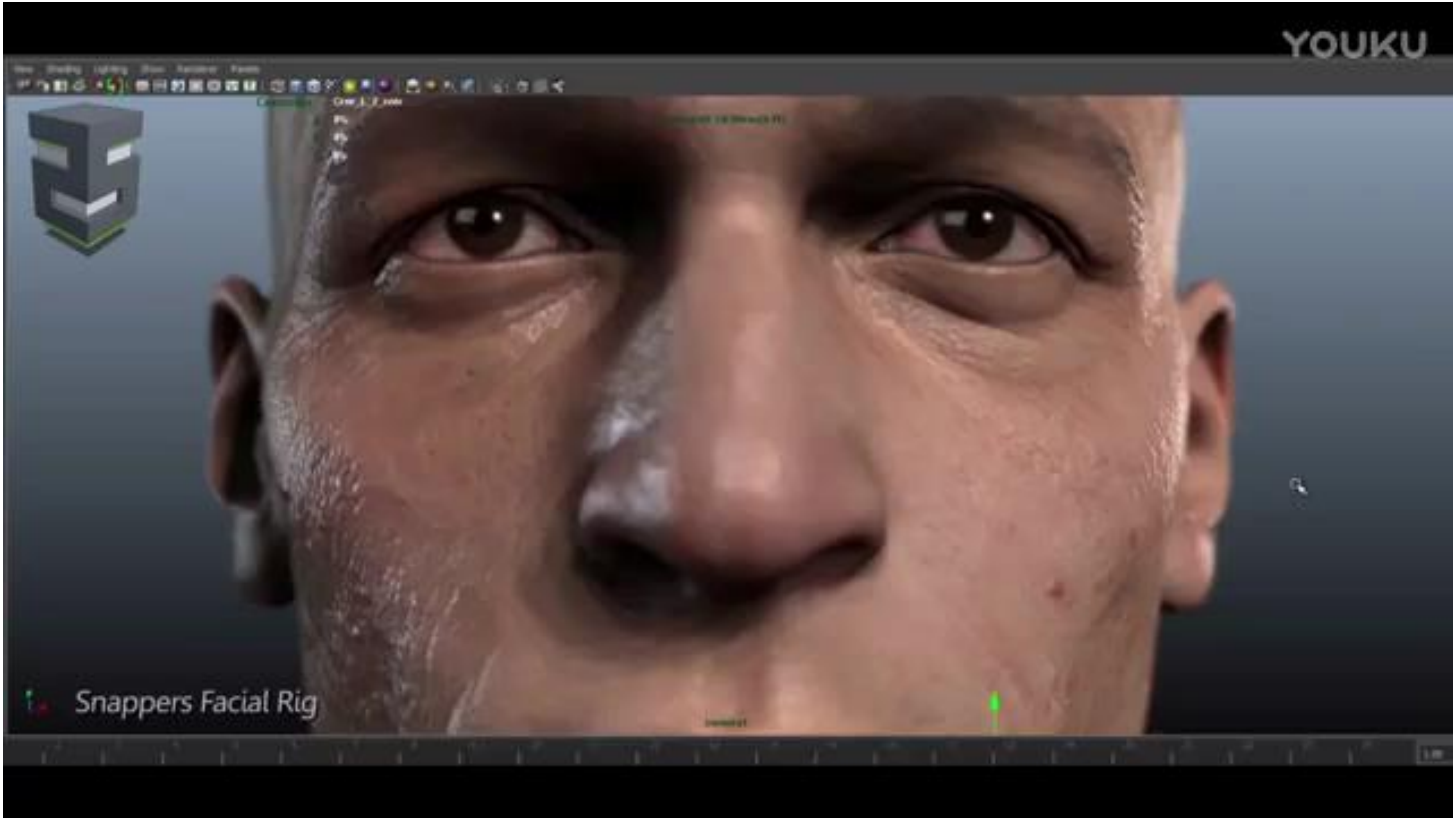
In Laptop, Tablet, Smartphone

For Everyone





Realtime Facial Animation



Snappers Facial Rig for Maya (also available for 3dsMax) by snappers mocaps

Acknowledgements

- **Lecture based on material from:**
  - CSCI 420: Computer Graphics FS 2015, by Hao Li, execllent slides and assignments: image 2 height fields, Simulating a Roller Coaster, ray tracing
  - Computer Graphics : 15-462/662 Fall 2016 - Carnegie Mellon University @ CMU
  - CS 148 Introduction to Computer Graphics and Imaging (Fall 2015) @ stanford
  - 6.837 Computer Graphics (fall 2011) @ MIT

Thanks