

Computer Graphics

-Introduction

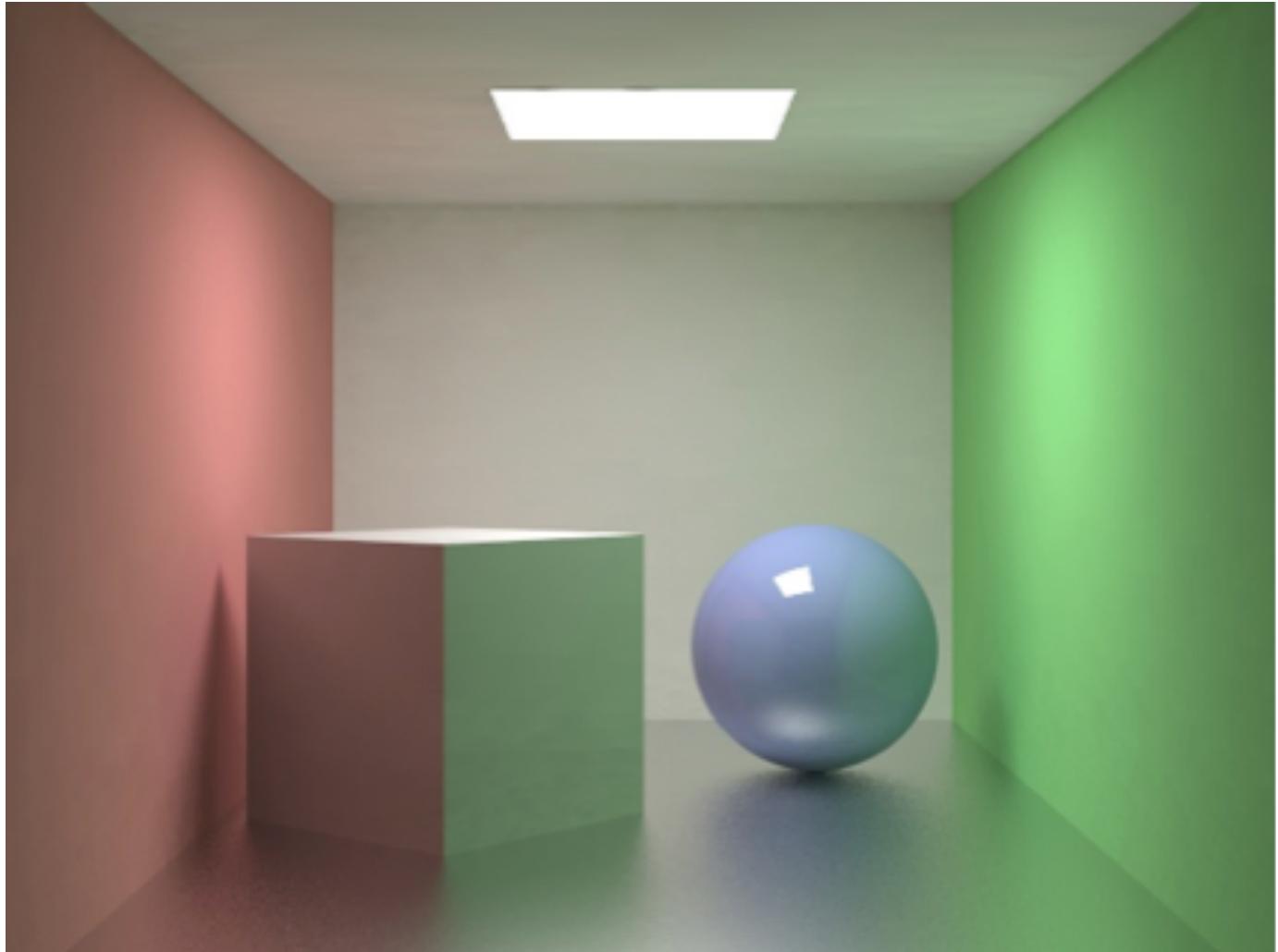
Junjie Cao @ DLUT

Spring 2019

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

Context

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



What is computer graphics?

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



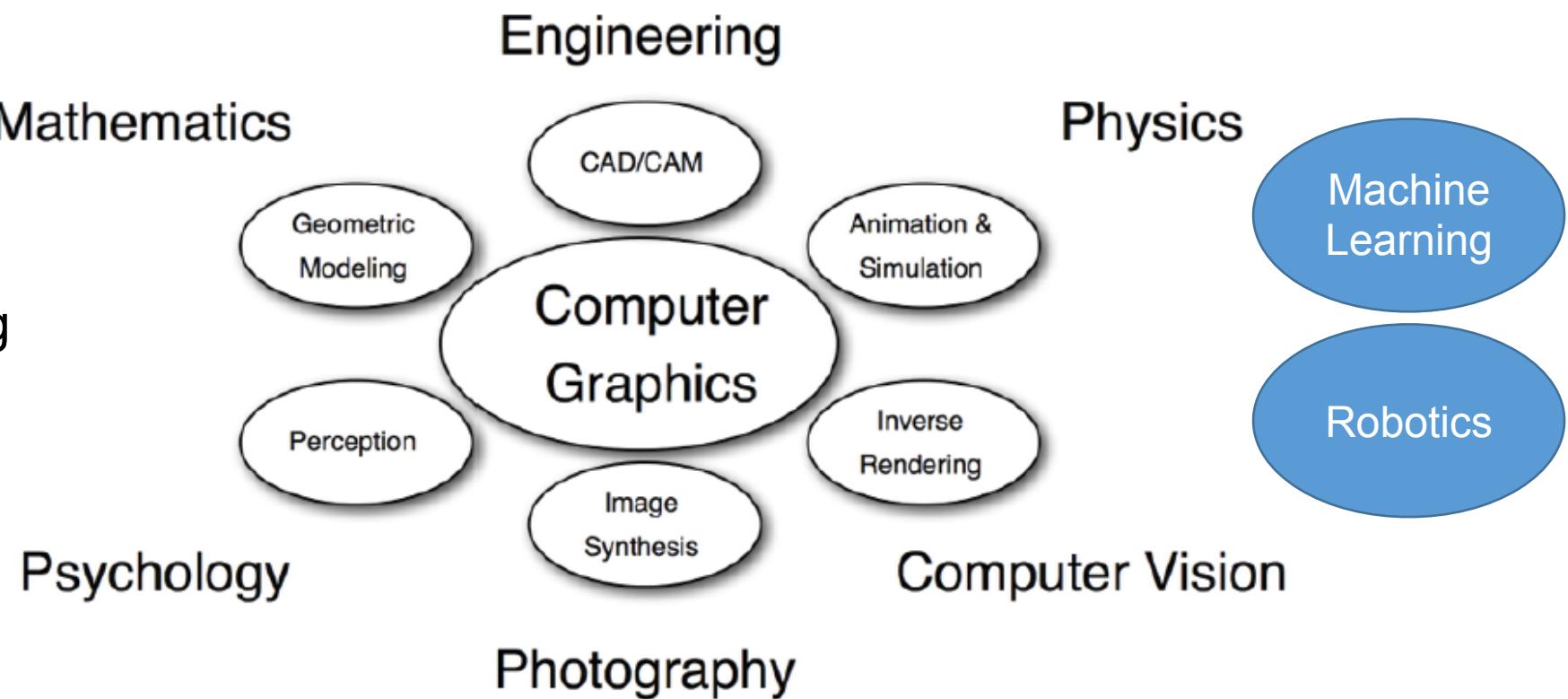
(sound)



(touch)

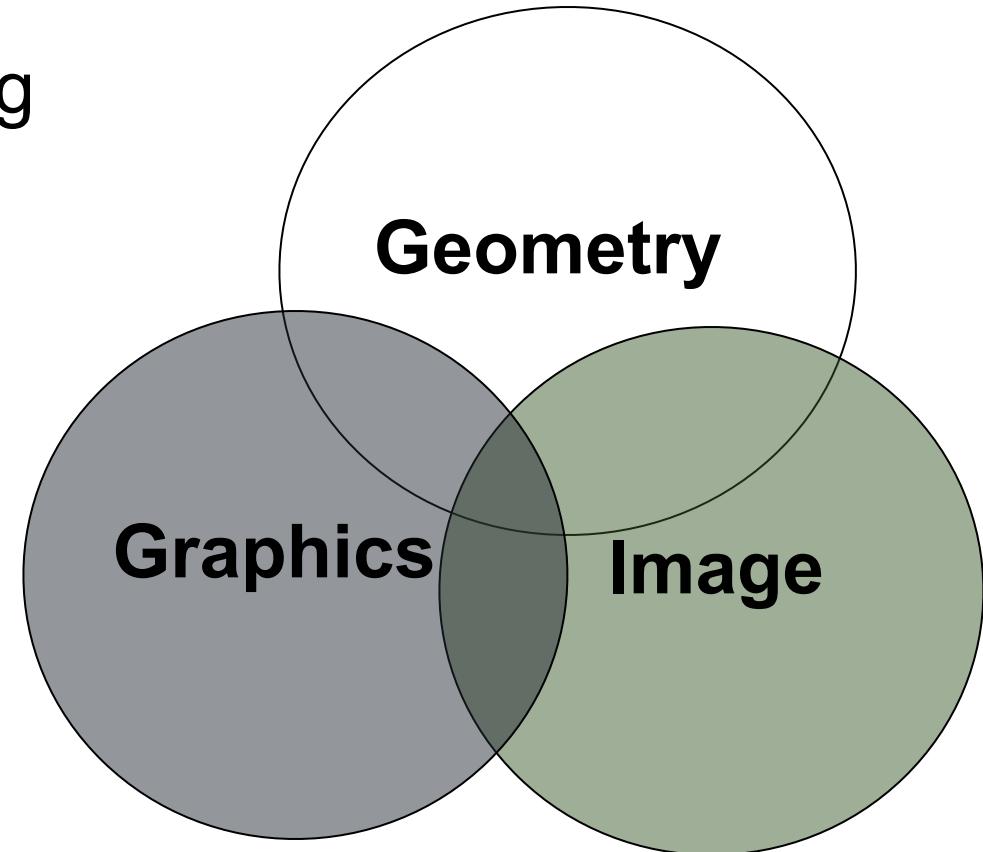
Related to many Disciplines

- 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



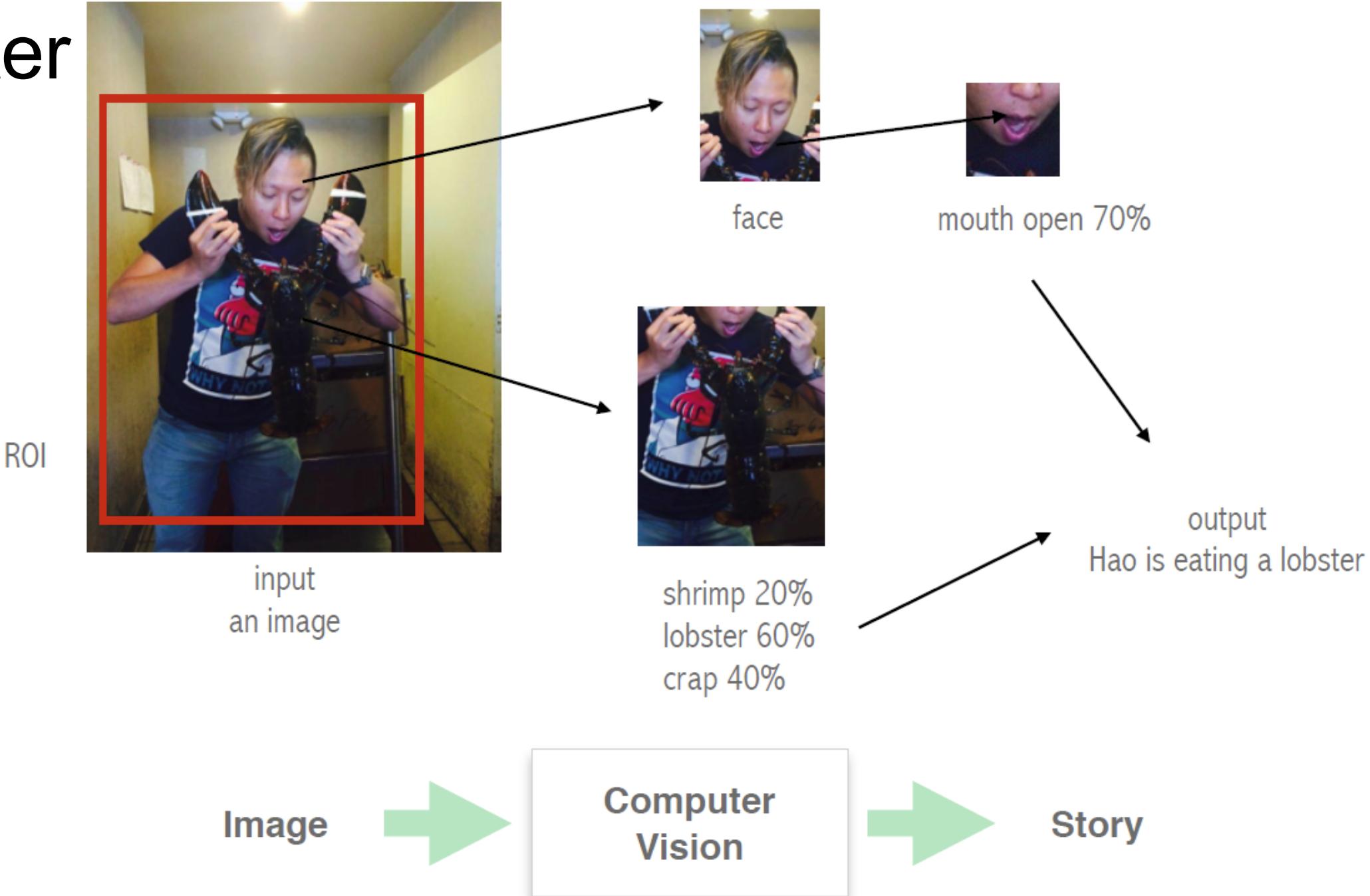
几何、图形、图像密不可分

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



Computer Graphics vs. Vision

Computer Vision



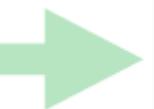
Computer Graphics



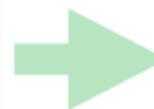
Action!



Story



Computer
Graphics

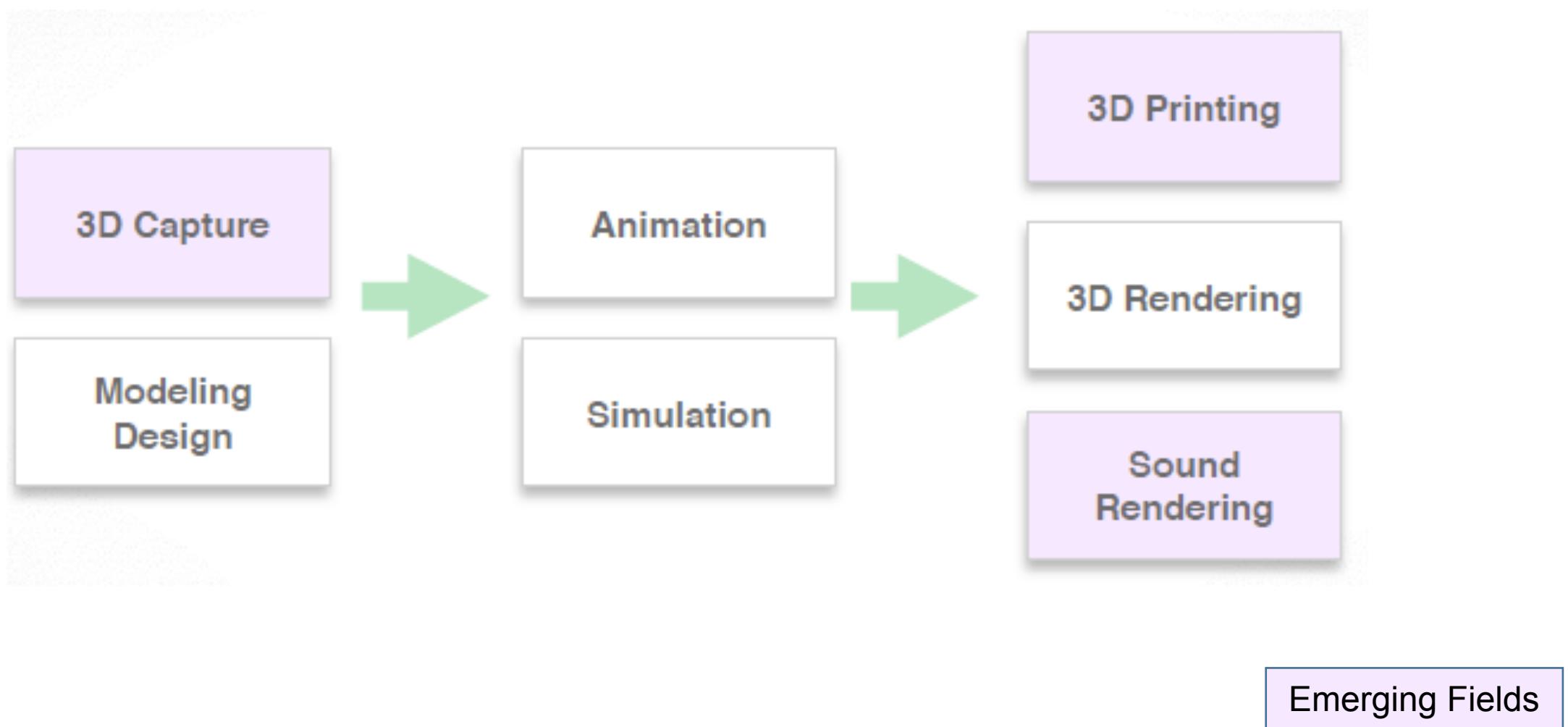


Image

What is Graphics? -- Generating images!

- Create them
 - Modeling
 - Animation
 - Rendering
- Manipulate them

3D Computer Graphics Pipeline



SIGGRAPH & SIGGRAPH Asia



ACMSIGGRAPH

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



SIGGRAPH & SIGGRAPH Asia

- SIGGRAPH 2018 Technical Papers Preview Trailer

Computer Graphics

- What you need to show other people your dreams.
- Mathematics made visible.

Course Overview

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

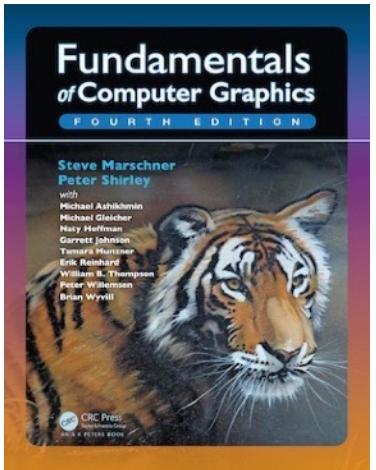
The team

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

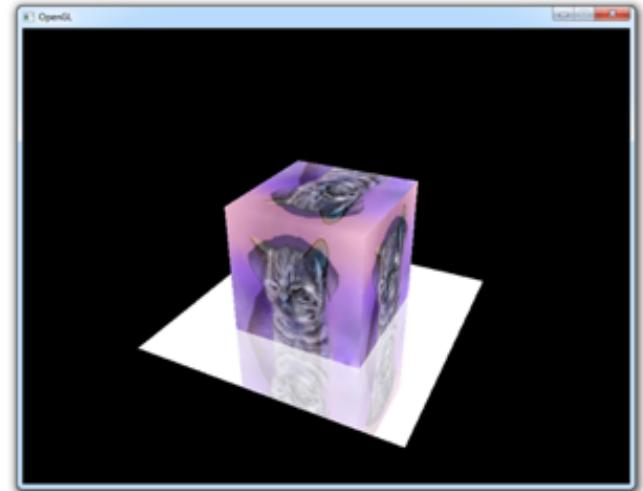
Course Topics

- **Images, image processing, color science**
- **Modeling in 2D and 3D**
- **Rendering 3D scenes** (using ray tracing and using the GPU)
- **Geometric transformations**
- **The graphics pipeline**
- **Animation**

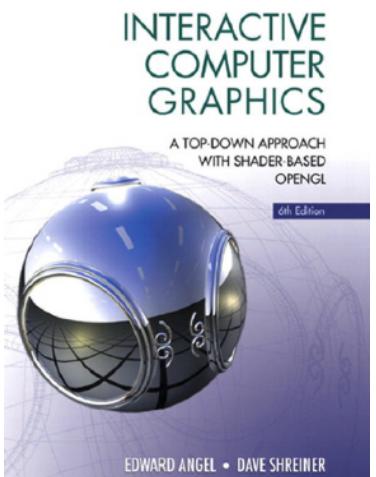
Textbooks



Marschner & Shirley
Fundamentals of Computer Graphics
4th edition, 2016



<https://open.gl>

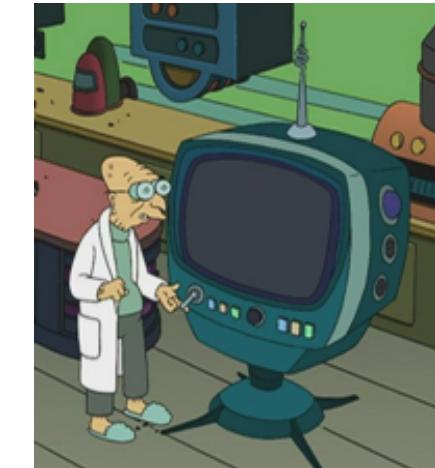


Edward Angel
Interactive Computer Graphics: A top-down approach with OpenGL
6th Edition, 2012

Prerequisites/What Is It I Expect?

- **Programming**

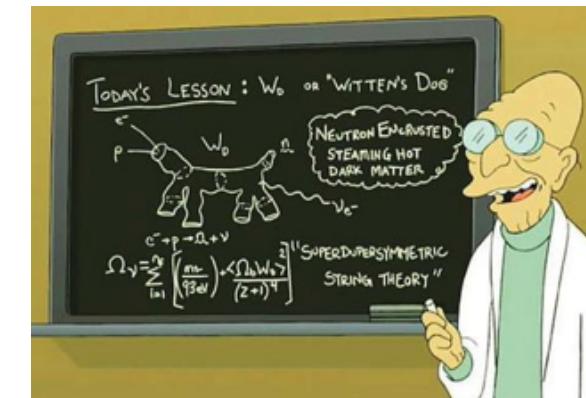
- ability to read, write, and debug
- small c++ programs (dozens of classes)
- understanding of basic data structures
- serious software design not required



- **Mathematics**

- vector geometry (vectors in 2D and 3D, dot/cross products, etc.)
- linear algebra (mainly linear transformations in 2 to 4D)
- basic calculus (calculating derivatives, understanding integration)
- probability (continuous probability for Monte Carlo)
- graphics is a good place to pick up some, but not all, of this

- Keeping up with the text(s) is very important



You will:

- explore fundamental ideas
- learn math essential to graphics
- implement key algorithms
- learn the basics of OpenGL
- write cool programs
- learn a bit about WebGL and doing graphics in the browser

- **You will not:**

- write very big programs

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

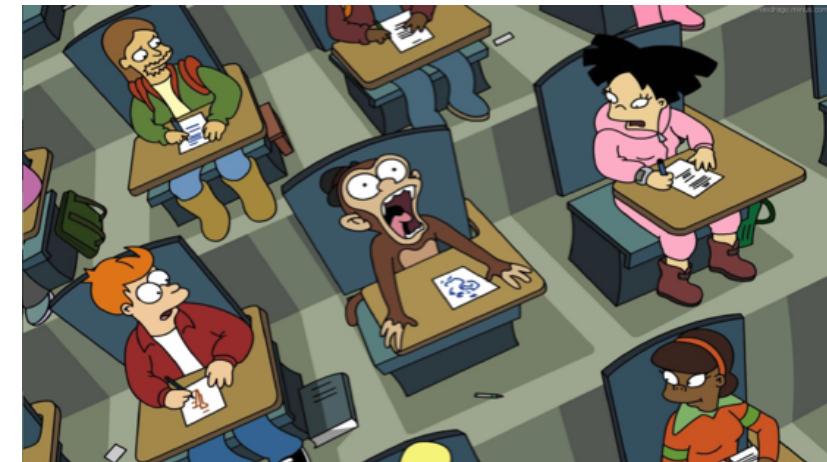
Text description;
Code segment for the function
Image illustration;

RF2

Text description;
Code segment for the function
Image illustration;

OF1

Text description;
Code segment for the function
Image illustration;



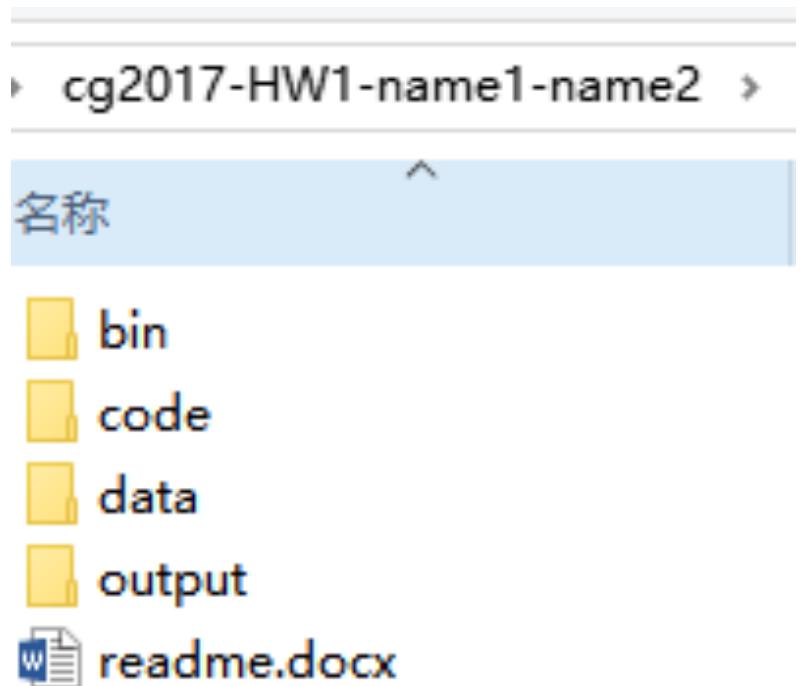
Code in electronic:

- 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



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:*****

2.1.3. 示意图

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：

```
blue_valence[0.3]
green_valence[0.5]
red_valence[0.8]
purple, else
```

2.2.2. Code

```
mesh_update_coloring()
    // Color visualization according to the valence of each vertex which is defined
    // by the function valence_coloring(mesh, valence).
    ...
    Mesh_color red = mesh_color(1.0, 0.0, 0.0);
    Mesh_color blue = mesh_color(0.0, 0.0, 1.0);
    Mesh_color green = mesh_color(0.0, 1.0, 0.0);
    Mesh_color purple = mesh_color(0.5, 0.0, 0.5);

    if (mesh_get_property(mesh, "valence") != NULL)
    {
        for (int i = 0; i < mesh_get_size(mesh); i++)
        {
            if (mesh_get_property(mesh, "valence")[i] >= 0.8)
                mesh_set_color(mesh, i, red);
            else if (mesh_get_property(mesh, "valence")[i] >= 0.5)
                mesh_set_color(mesh, i, green);
            else
                mesh_set_color(mesh, i, blue);
        }
    }
}
```

对每个顶点按顶点价值赋予颜色

2.2.3. 示意图

·1. 概述(Introduction)

·1.1. 如何使用本程序

1. 读入并显示网址: DecimateGlobe.bunny.off
2. 计算顶点价(右侧)
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 布特 `void mesh_coloring(mesh, valence*)` 类型的 `Valence` 属性。这个属性存储了 vertex 的 `valence`，也就是 1-8 之间的顶点数。为了获得 `Valence`，我们设计两条基于迭代器的循环。外层从遍历 `mesh` 上的每个 vertex，内部再遍历一个该顶点的所有 vertex，从而得到顶点价。值得注意的是，我们可以利用 `mesh_it` 生成的 `auto` 自动生成迭代器类型，而不需要详细了解这三步 Mesh `valence` 泛型 Mesh: `valence`。

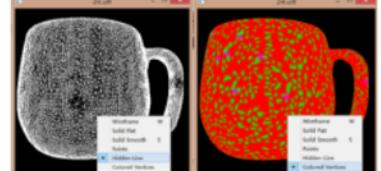
·2.1.2. Code

```
void mesh_coloring()
{
    // Compute valence of each vertex and store it.
    // We implement this function
    mesh->set_property("valence", "valence");
    for (auto iter = mesh->vertices_begin(); iter != mesh->vertices_end(); ++iter)
    {
        mesh->property("valence", *iter) = 0;
        for (auto it = mesh->neighbors(*iter); it != mesh->neighbors(*iter).end(); ++it)
            mesh->property("valence", *iter) += 1;
    }
}
```

计算顶点价

·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?



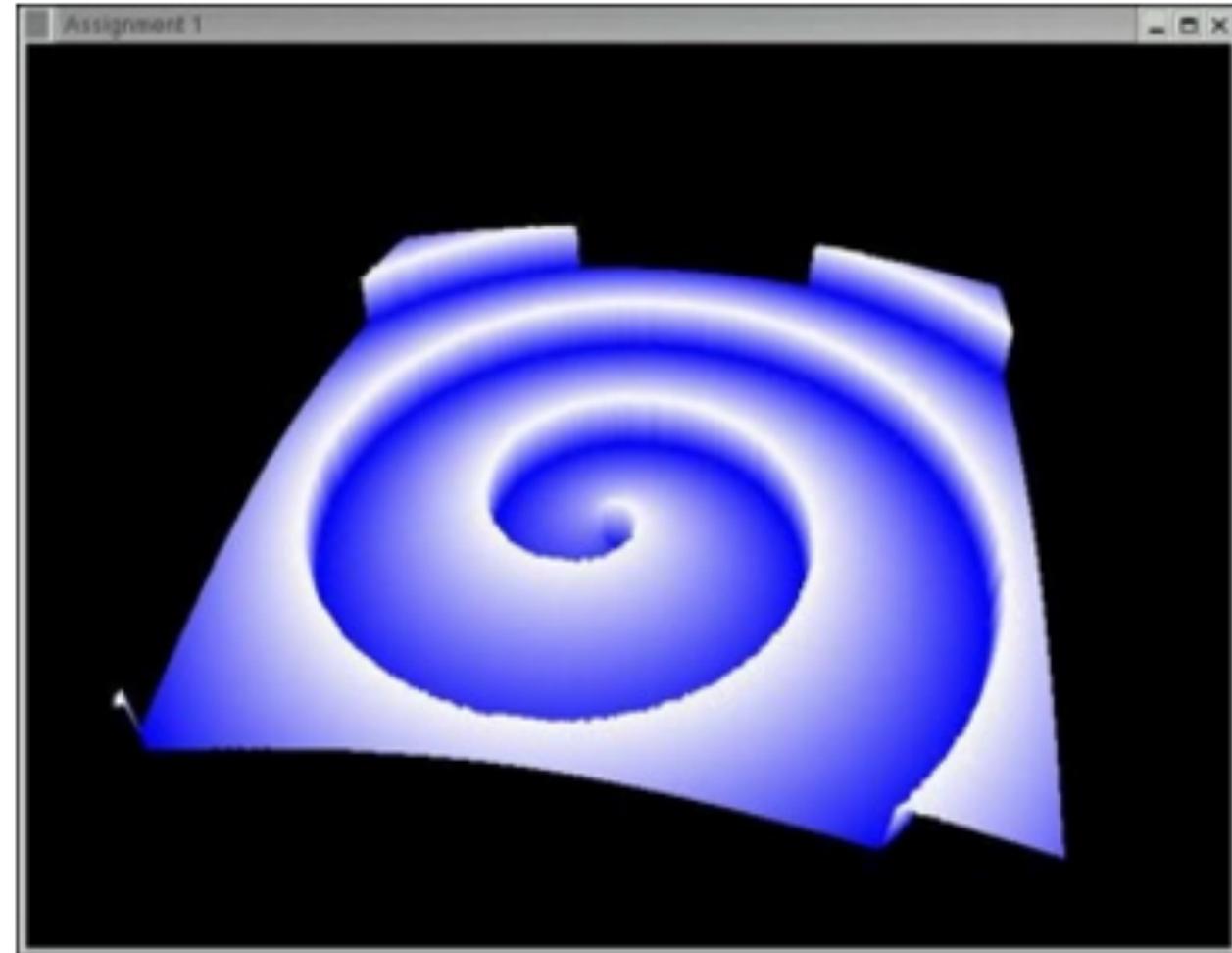
Assignment Policies

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

Assignment 1 – Height field

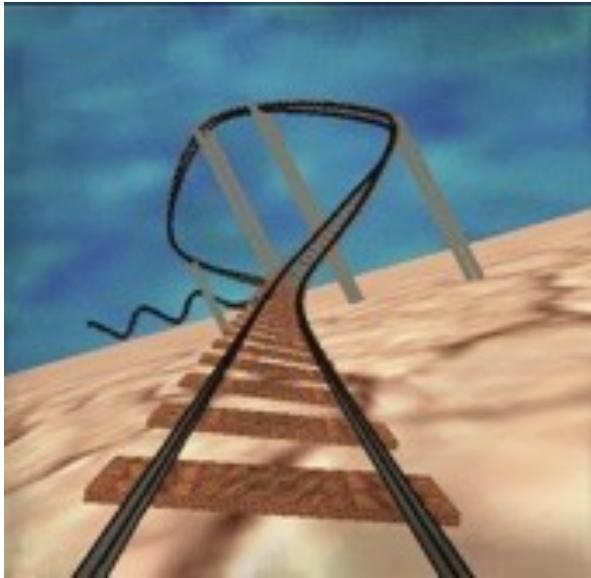


input (source image)



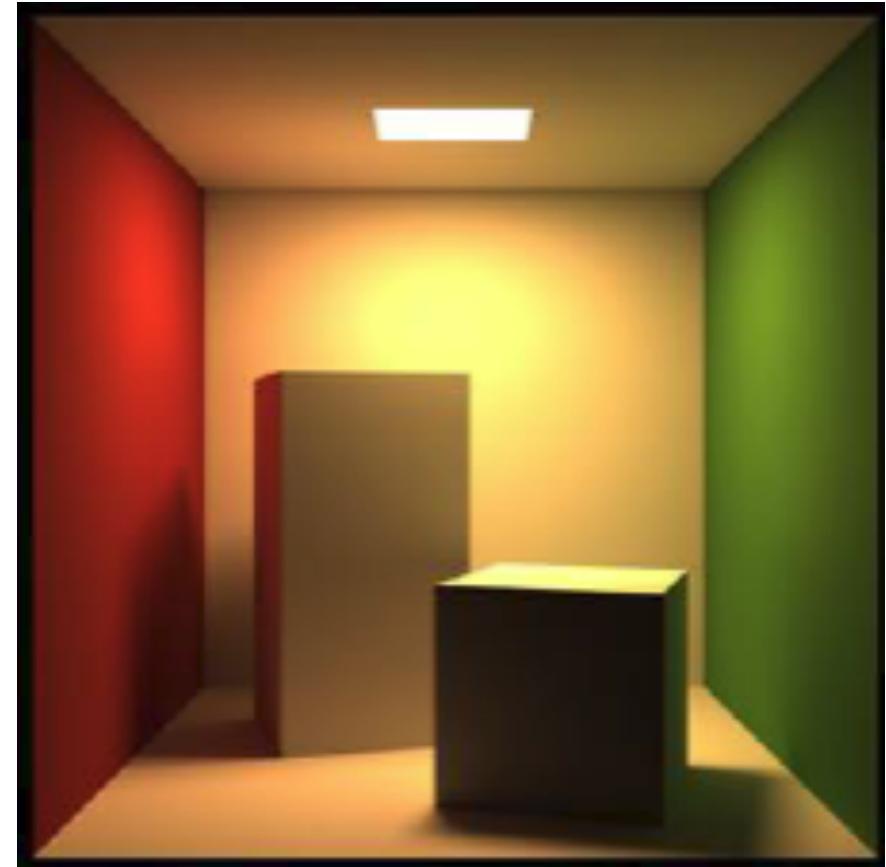
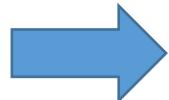
output (height field)

Assignment 2 – Simulating a Roller Coaster



Assignment 3 – Ray tracing

Example
Example



Other

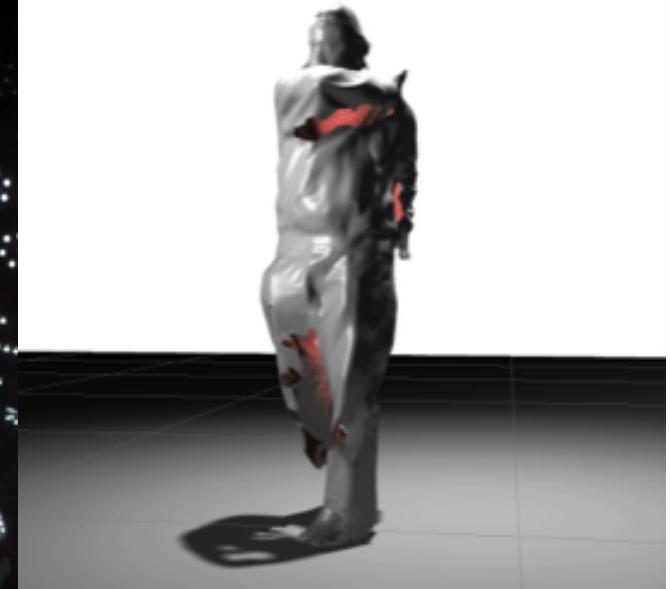
- 留一个联系人，确定上机时间。

Introduction

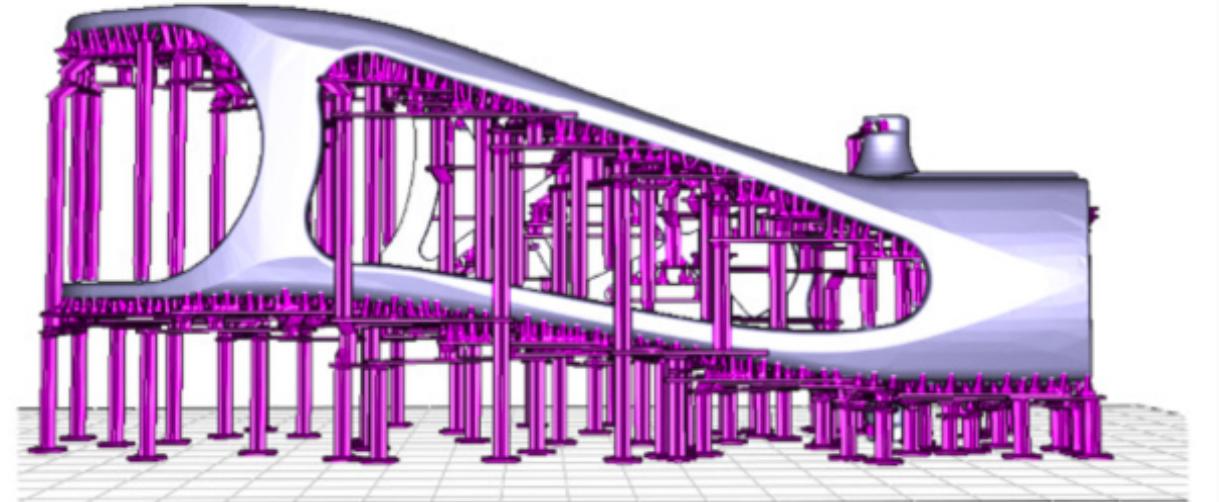
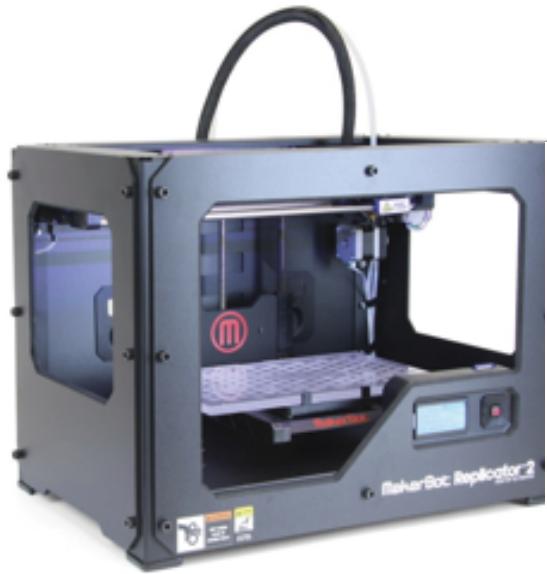
- What is Computer Graphics?
 - Applications
 - History
 - Relations with other Disciplines
- Course Overview
- **Research Trends**



From Graphics to Vision



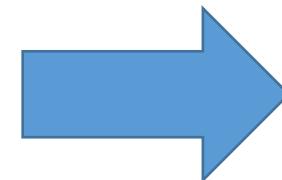
From Graphics to Fabrication



From Production to Consumers



VFX

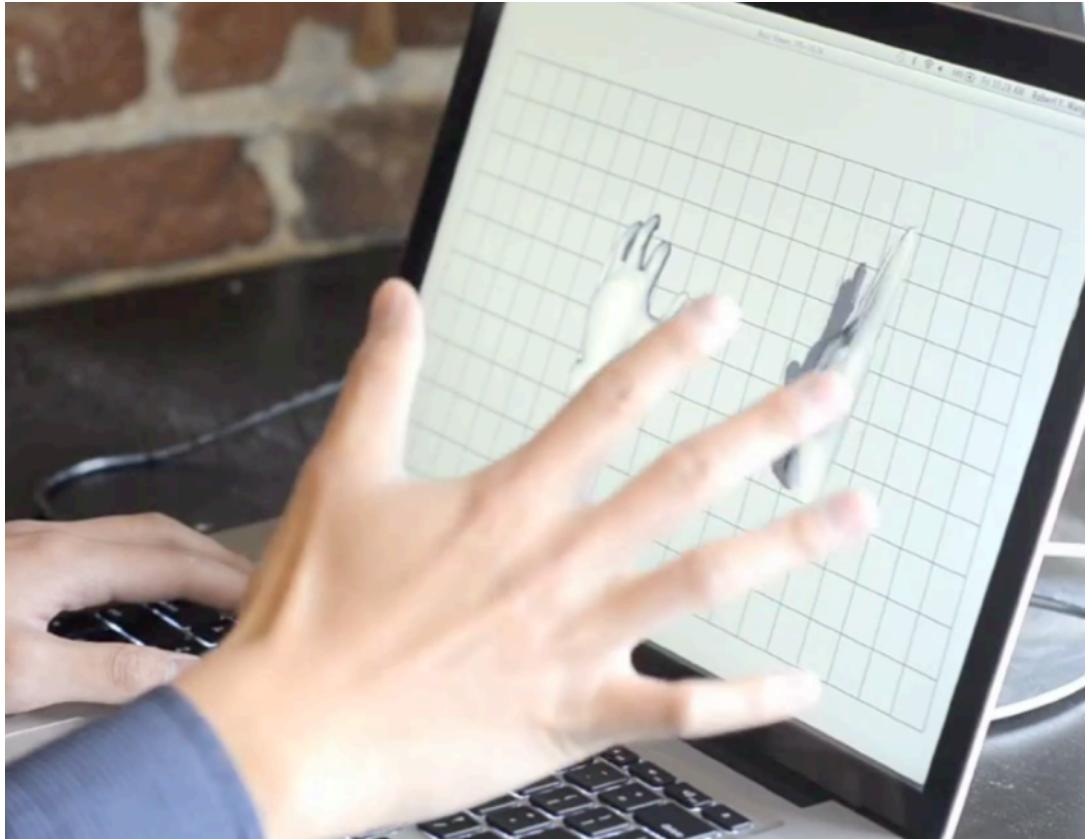


online shopping

In Laptop, Tablet, Smartphone



For Everyone



AI-Driven Computer Graphics

Thanks