



中山大學 軟件工程學院
SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

Lecture 01：计算机图形学介绍

SSE315：计算机图形学
Computer Graphics

陈壮彬

软件工程学院

chenzhb36@mail.sysu.edu.cn

Today's topics

□ 计算机图形学介绍

□ 绘制一个简单的 3D 图像

□ 课程内容介绍

□ 课程安排及评分

Today's topics

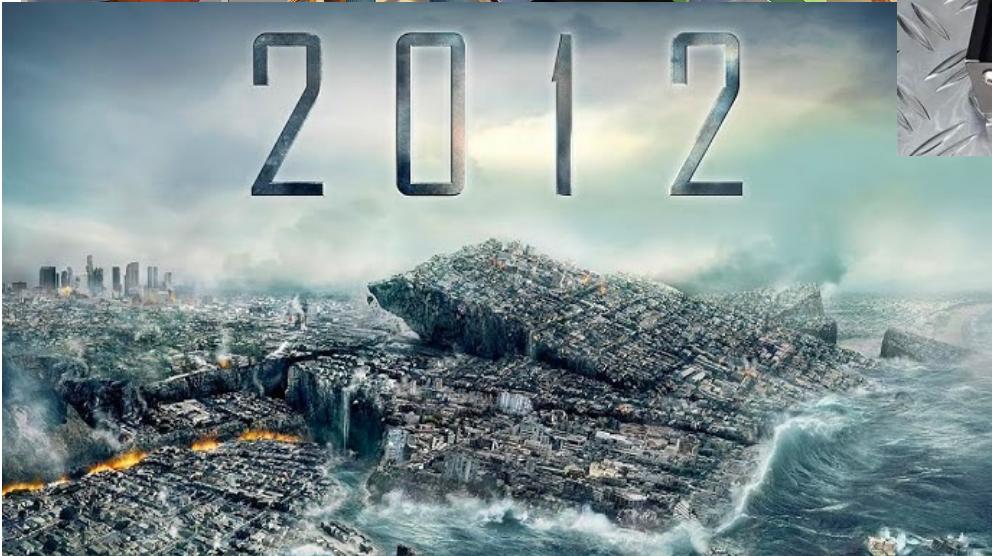
□ 计算机图形学介绍

□ 绘制一个简单的 3D 图像

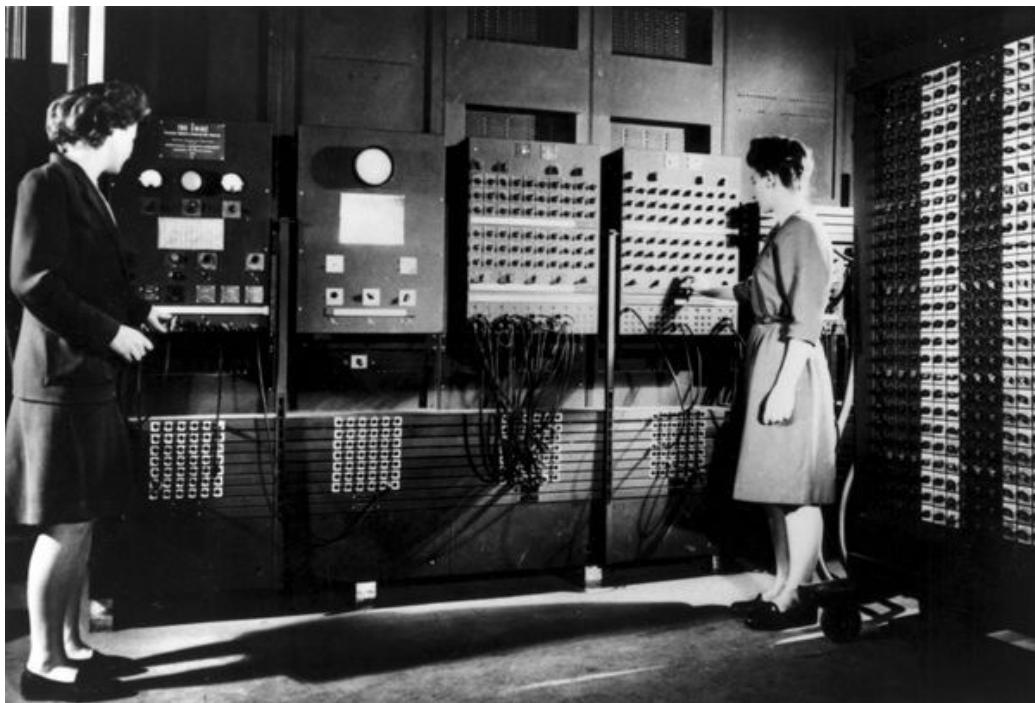
□ 课程内容介绍

□ 课程安排及评分

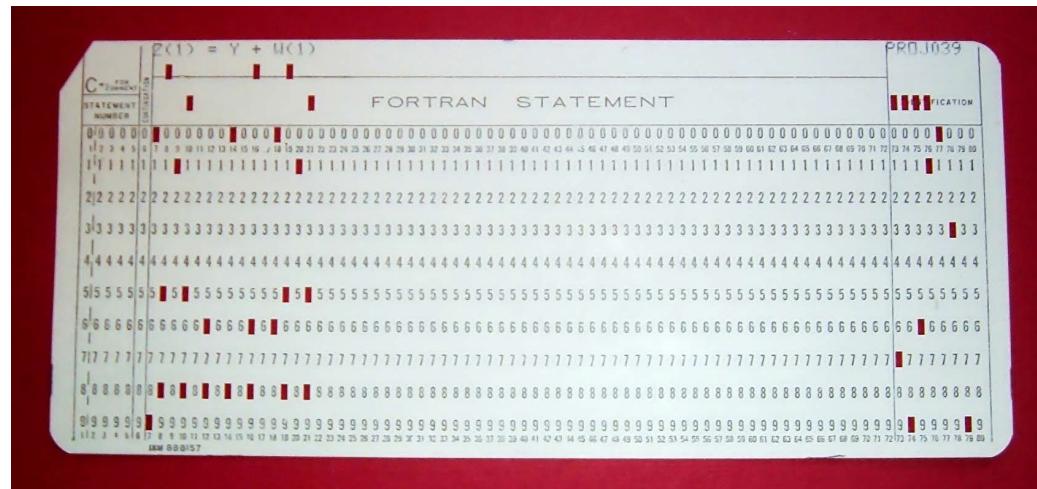
计算机图形学的第一印象



**什么是计算机图形学？
为什么我们需要它？**

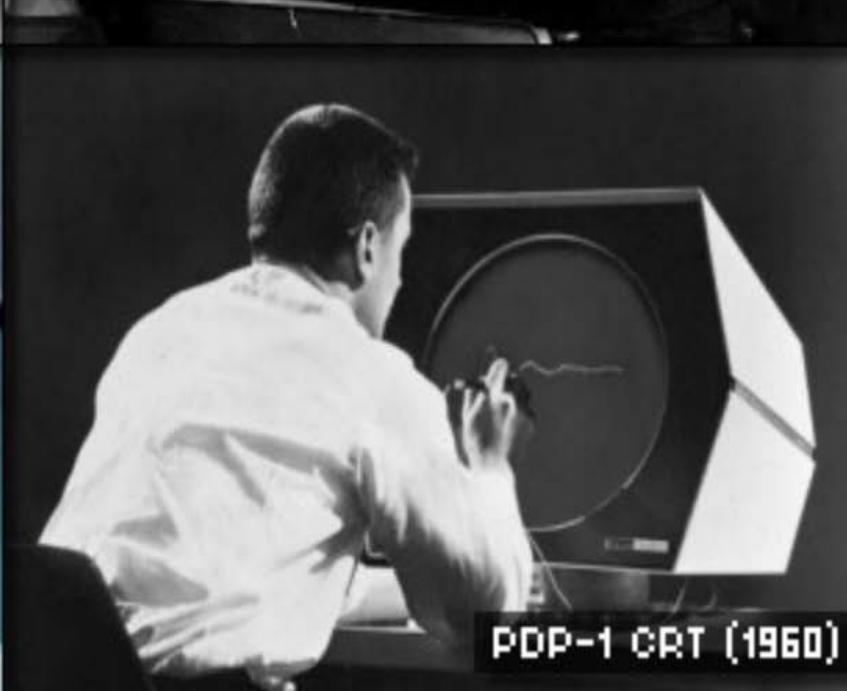


Early computer, 1945
(ENIAC 埃尼阿克)



Punch card (~120 bytes)

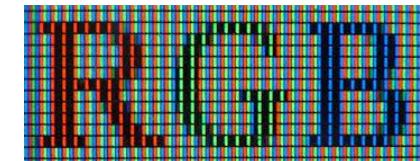
更好的计算机交互方式！



Sketchpad (Ivan Sutherland, 1963)



MACINTOSH (1984)



像素

2020: 8k monitor
7680x4320 (~95MB)



2020 virtual reality headset:
2x2160x2160 @ 90Hz => 2.3GB/s

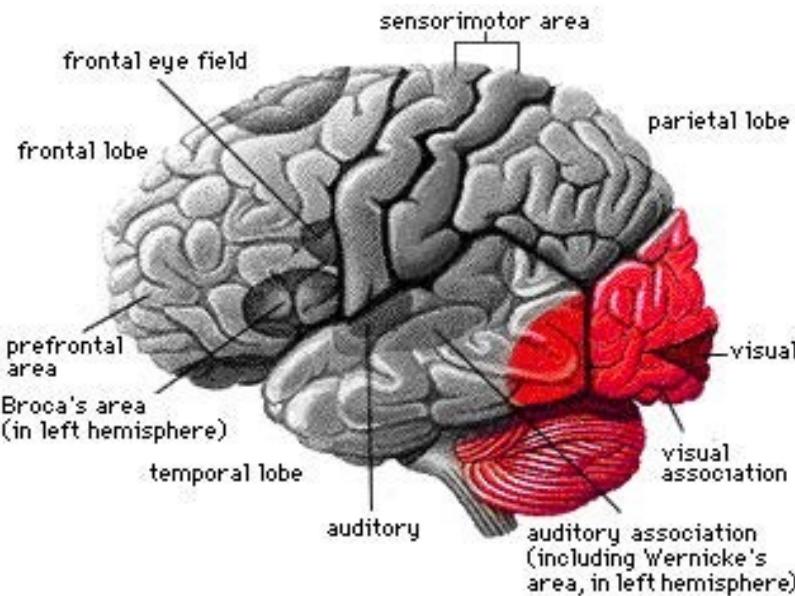
苹果 Vision Pro
有 2300 万像素！

苹果 Vision Pro



【何同学】苹果 Vision Pro 深度体验

Why virtual information?



大脑中有近 30% 的部分专门用于处理视觉信息！

眼睛是我们头部接收信息最快、最高效的"接口"

一图胜千言

什么是计算机图形学？

com•put•er graph•ics /kəm'pyoodər 'grafiks/ *n.*

The use of computers to synthesize visual information.

合成



Computation



Virtual information



Digital information

什么是计算机图形学？

com•put•er graph•ics

The use of computers to synthesize

Why only visual?

visual information.



Computation



Digital information

Virtual information



**自诞生以来，计算机图形学已
经发展了很多年... 不再仅仅是
点亮像素那么简单！**

将数字信息转化成感官信息



sound



touch

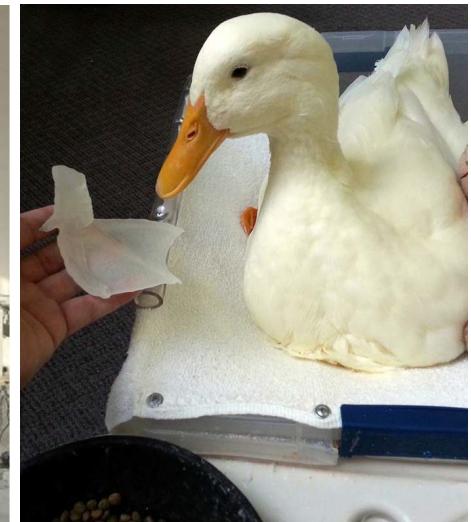
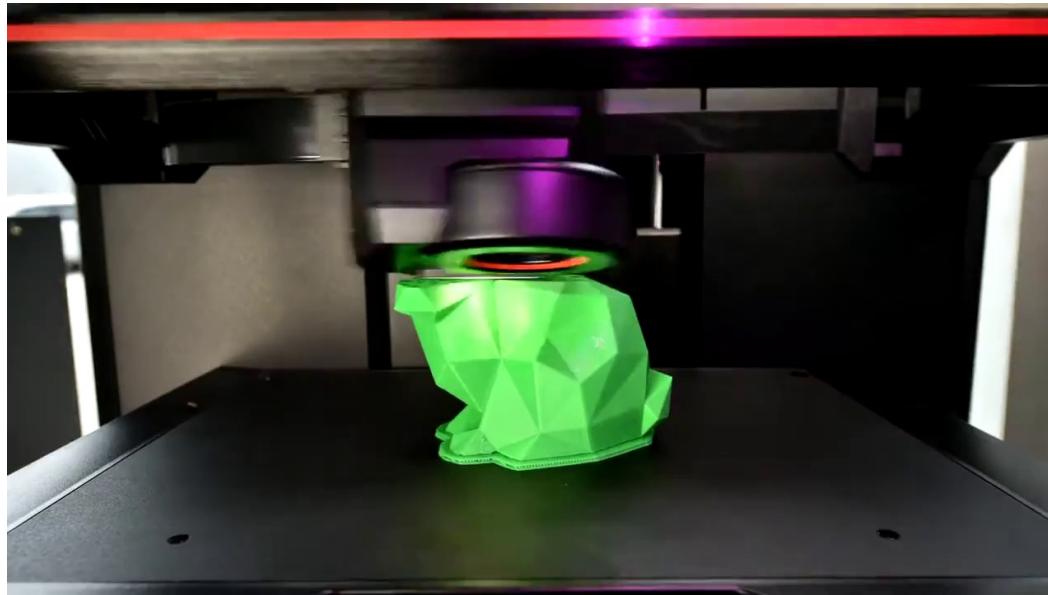
com•put•er graph•ics /kəm'pyoodər 'grafiks/ *n.*
The use of computers to synthesize and manipulate
sensory information.

(...What about taste? Smell?!)

将数字信息转化成感官信息



将数字信息转化成物理实体



什么是计算机图形学? again...

com•put•er graph•ics /kəm'pyoodər 'grafiks/ *n.*
The use of computation to turn **digital information** into
sensory stimuli.

感官刺激

即便是这个定义也有点狭窄...

SIGGRAPH 2024 Technical Papers Trailer



SIGGRAPH 2025
Vancouver+ 10-14 August

The Premier Conference & Exhibition on
Computer Graphics & Interactive Techniques
Technical Papers Trailer

<https://www.bilibili.com/video/BV1q4KpzREGK/>

计算机图形学无处不在！

游戏 Games



王者荣耀

无主之地 3 (2019)



游戏 Games



黑神话：悟空

黑神话制作组游戏科学的十年蜕变

- 2012年：腾讯《斗战神》(前三章) 用户满意度达90%，最高在线人数达60万，创国内历史新高
- 2013年：因工作室一系列运营决策失误，《斗战神》用户游戏体验大幅下滑，在线玩家数暴跌
- 2014年：冯骥与杨奇带领公司部分骨干辞职，成立《游戏科学》
- 2015年：游戏科学为站稳脚跟，迎合市场风向研发手游《百将行》，赚到了公司第一桶金
- 2016年：公司第二部作品《战争艺术：赤潮》，一度成功登顶苹果手游销量榜首，解决了公司的生存问题(赚快钱)
- 2018年：公司决定转战单机游戏领域(当时没有一款真正的国产3A单机游戏)，《黑神话：悟空》立项，游戏编号B1，意味黑神话系列的第一部作品
- 2020年：为扩大项目团队，公司发布《黑神话：悟空》的首支游戏实机演示，上线后立即收到全网玩家的热捧(但当时公司基本只有演示视频...)
- 2024年：《黑神话：悟空》发售，3天销售突破1000万，全平台在线人数达300万，steam在线人数历史第二，单机游戏第一，打破多项记录
- 2025年，销量突破3000万套，获得游戏开发者大会(GDC)最佳视觉艺术奖等多项提名

电影 Movies



阿凡达 2009 vs. 阿凡达 2022



动画 Animation



疯狂动物城
(2016)

冰雪公主 2
(2019)



设计 Design



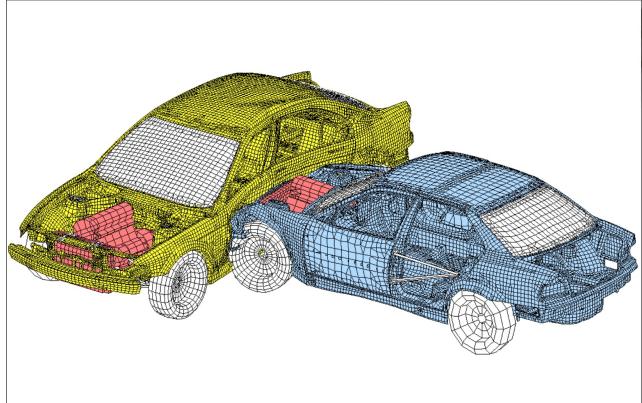
汽车建模



汽车模型



汽车碰撞
实验

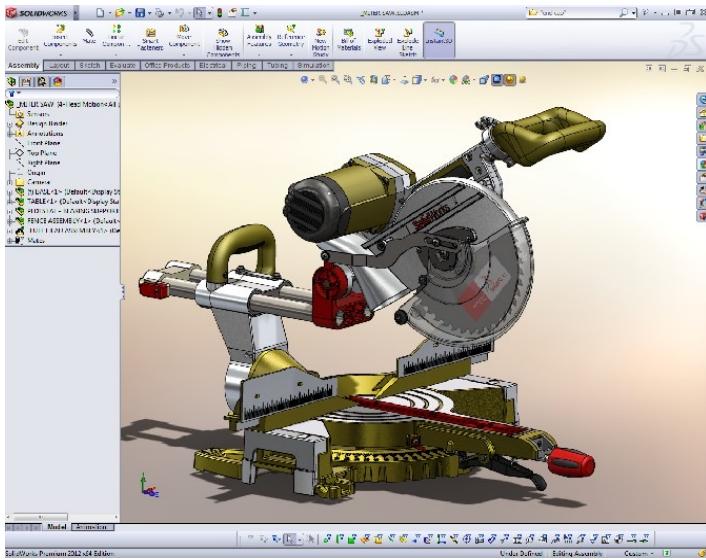


设计 Design



宜家 Ikea - 75% 的展示图片都是渲染出来的

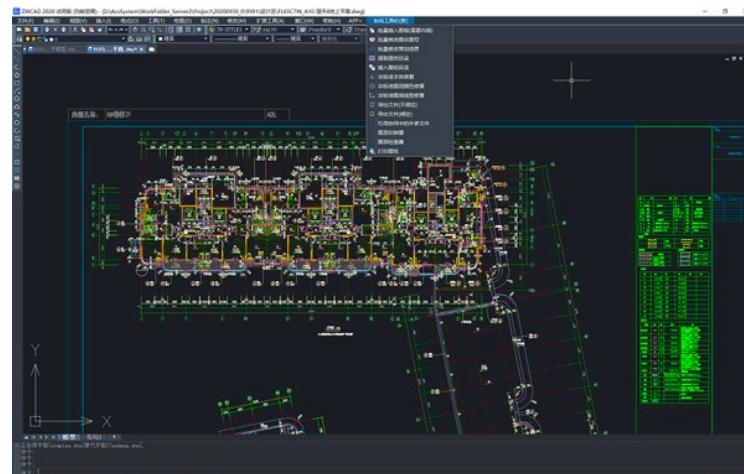
计算机辅助设计 CAD



SolidWorks



AutoCAD



中望 CAD

仿真 Simulation

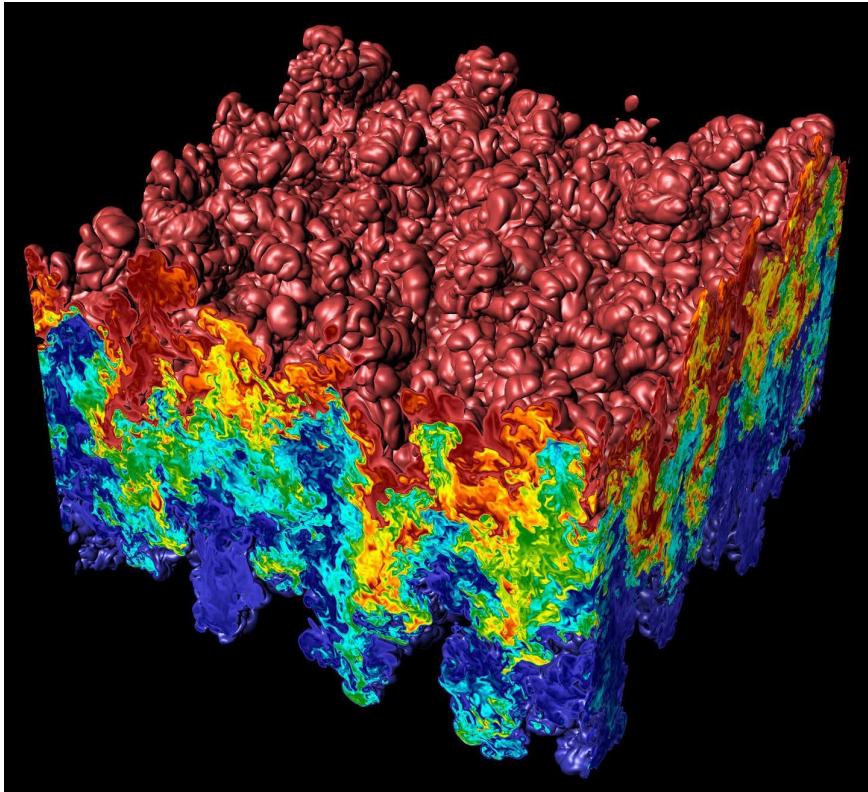


沙尘暴

《星际穿越》中的黑洞



科学/数学研究可视化

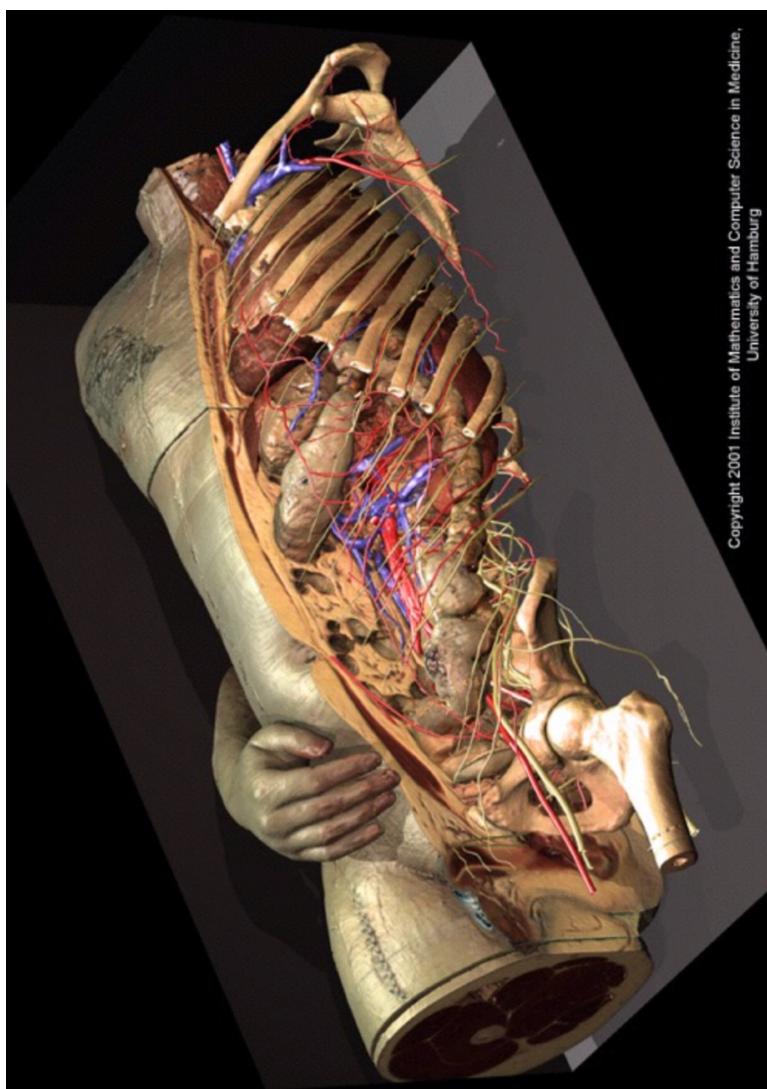


瑞利-泰勒
不稳定性

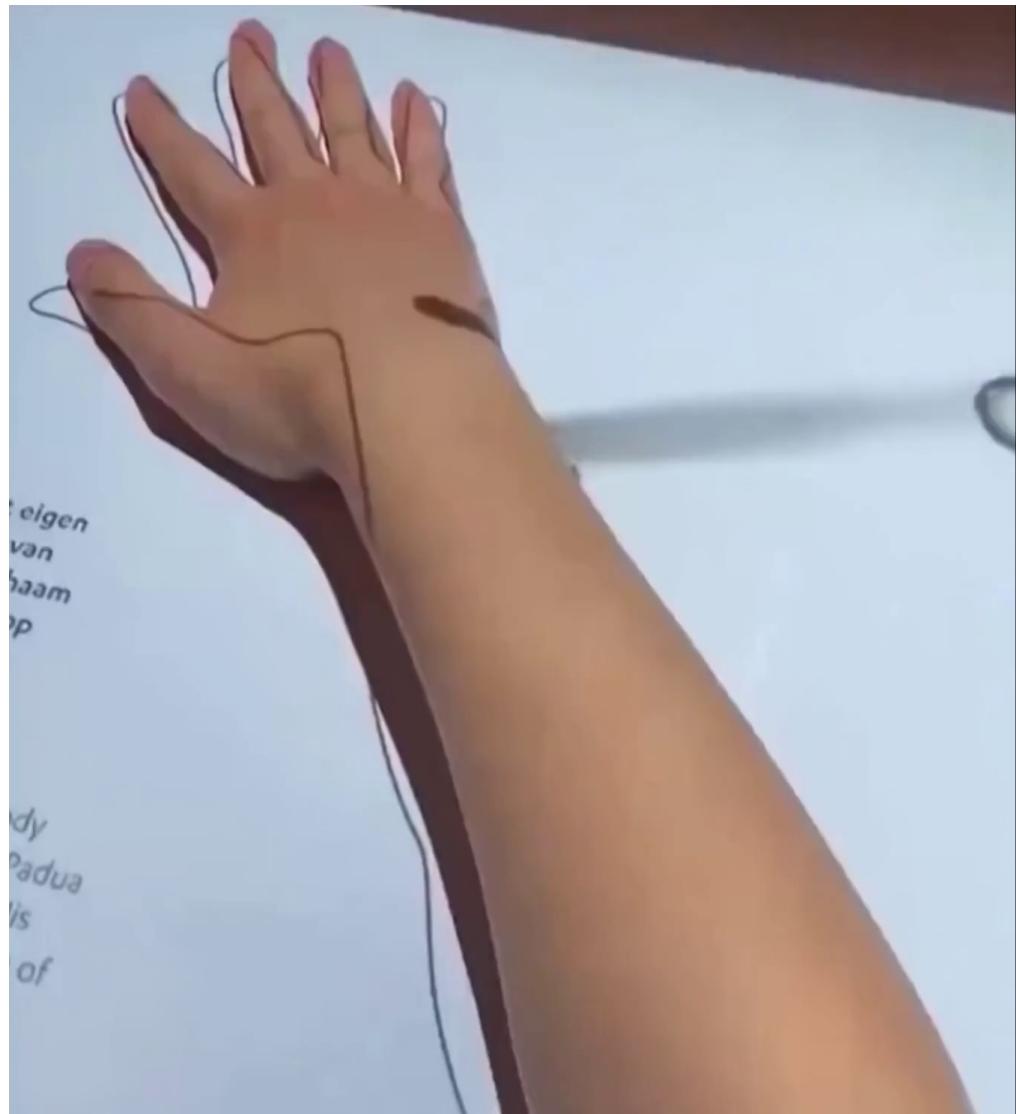


NASA 宇宙观测

医学/解剖学可视化



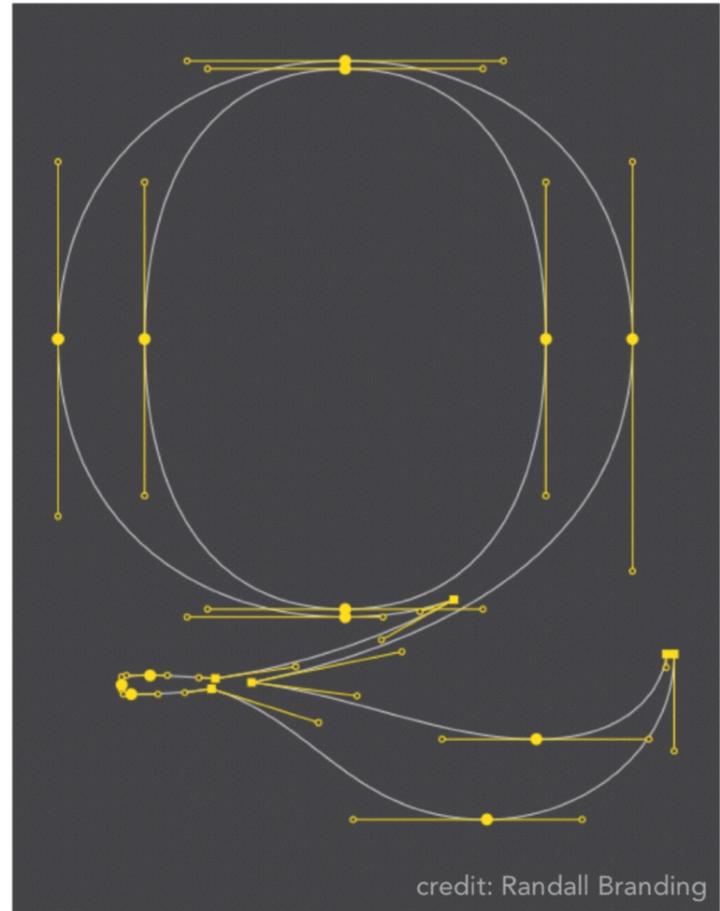
Copyright 2001 Institute of Mathematics and Computer Science in Medicine,
University of Hamburg



字体排版 Typography

The Quick Brown
Fox Jumps Over
The Lazy Dog

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 01234567890



计算机图形学基石

□ 所有这些图形学的应用均需要**复杂的理论和系统**

□ 理论 Theory

- **Basic representations** (如何对形状/运动进行数字编码)
- **Sampling & aliasing** (如何获取并重现信号?)
- **Numerical methods** (如何以数值的方式处理信号?)
- **Radiometry & light transport** (光是如何表现的?)
- **Perception** (这些与人类有什么关系? 人是如何感知的?)
- ...

□ 系统 Systems

- 并行、异构的处理
- 专门的图形编程语言
- ...

简单的实战：画一个立方体

实战：建模并画一个立方体

□ 目标：生成立方体的真实图像的第一步...

□ 关键问题

- 建模 (Modeling) : 我们如何描述一个立方体?
- 渲染 (Rendering) : 我们如何可视化这个模型?



建模一个立方体

□假设我们的立方体

- Centered at the origin (0, 0, 0)
- Has dimensions 2*2*2 and length 2
- Edges are aligned with x/y/z axes

□问题1：What are the coordinates of the cube vertices?

- | | |
|-----------------|------------------|
| A: (1, 1, 1) | E: (1, 1, -1) |
| B: (-1, 1, 1) | F: (-1, 1, -1) |
| C: (1, -1, 1) | G: (1, -1, -1) |
| D: (-1, -1, 1) | H: (-1, -1, -1) |

□问题2：What about the edges?

AB, CD, EF, GH,
AC, BD, EG, FH,
AE, CG, BF, DH

画一个立方体

□现在我们已有了如下对立方体的描述

VERTICES

A: (1, 1, 1)	E: (1, 1, -1)
B: (-1, 1, 1)	F: (-1, 1, -1)
C: (1, -1, 1)	G: (1, -1, -1)
D: (-1, -1, 1)	H: (-1, -1, -1)

EDGES

AB, CD, EF, GH,
AC, BD, EG, FH,
AE, CG, BF, DH

□我们如何把这个 3D 的立方体画在 2D 的平面上呢？

□基本策略

这一步怎么做？

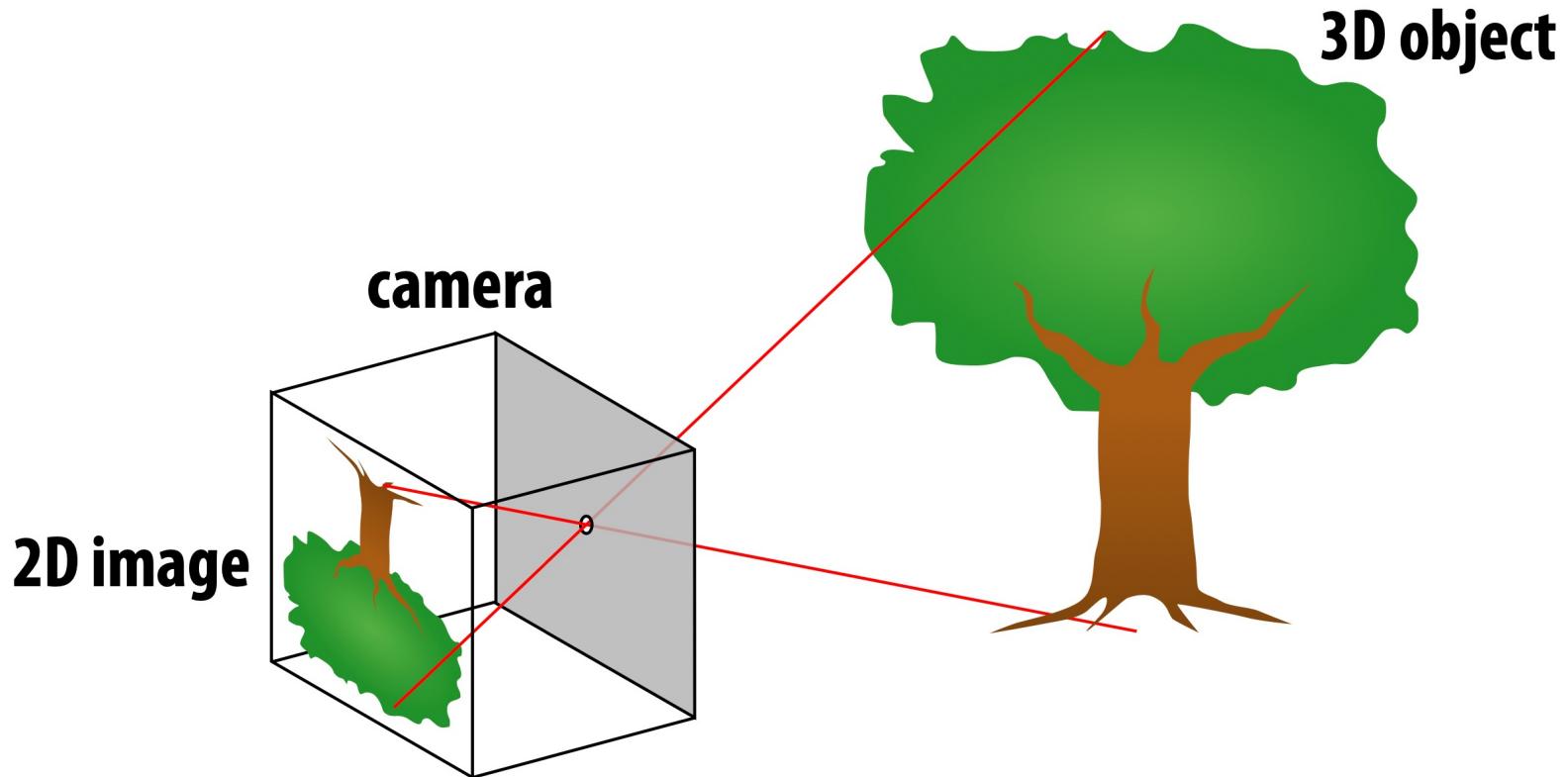
1. Map 3D vertices to 2D points in the image
2. Connect 2D points with straight lines

透视投影 Perspective projection

□ 越远的物体看起来越小 (perspective)

□ 为什么会这样?

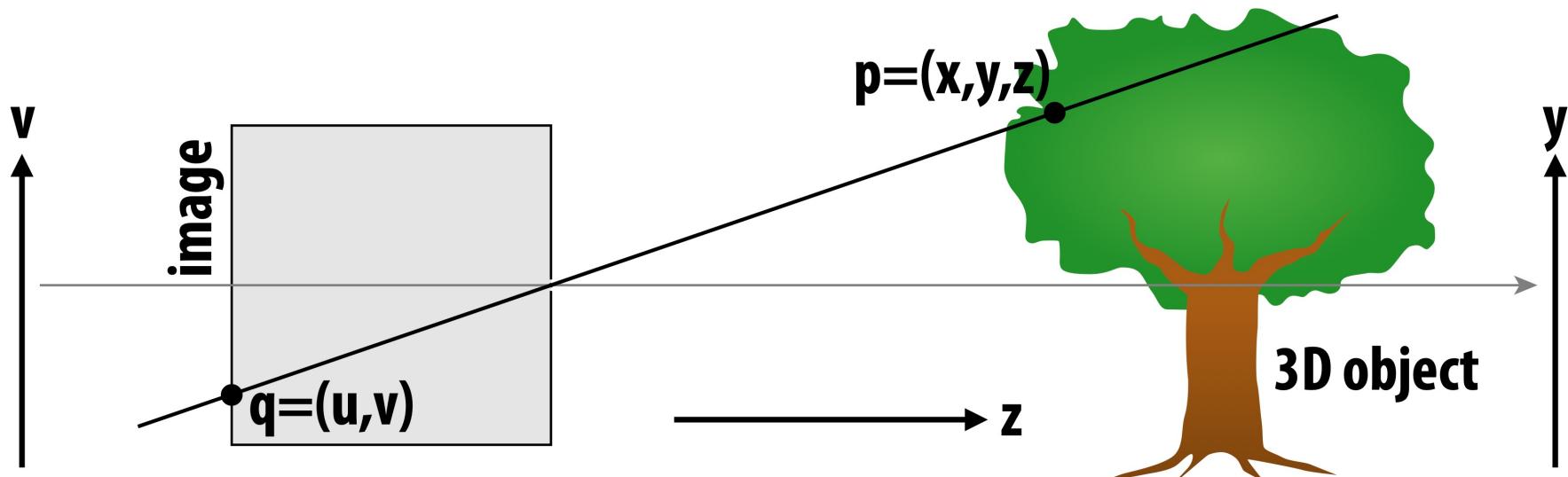
□ 考虑一个简单的相机“针孔”模型



Perspective projection: side view

□ 点 $p=(x, y, z)$ 将出现在图像上具体哪个点?

□ 我们将该点记为 $q=(u, v)$

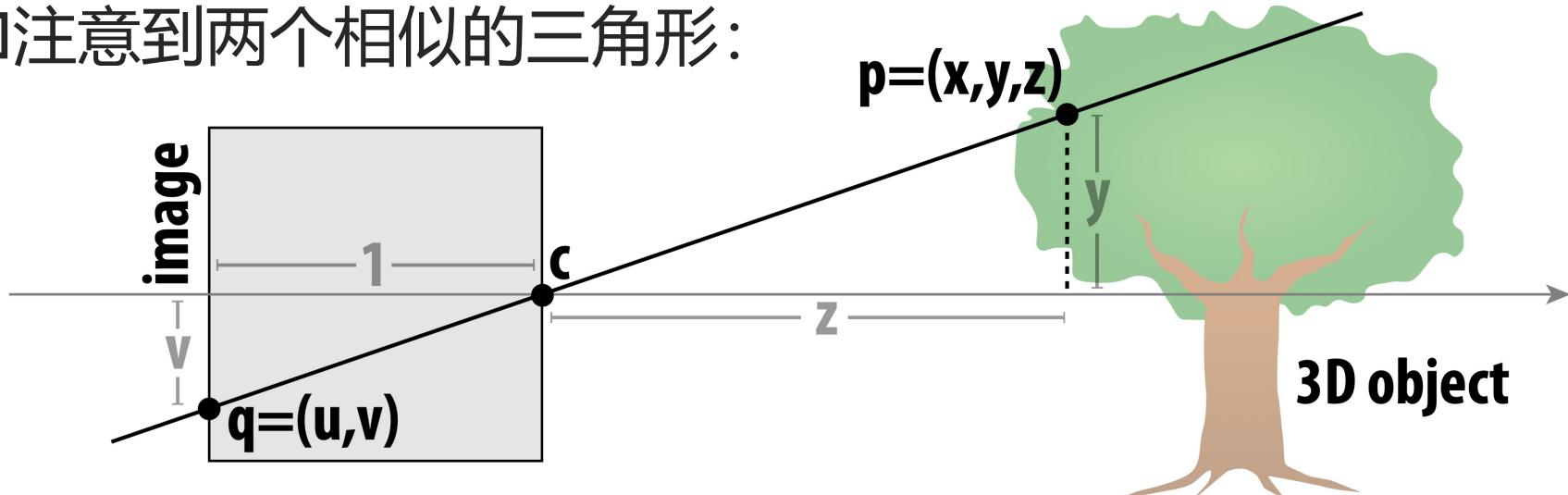


Perspective projection: side view

□ 点 $p=(x, y, z)$ 将出现在图像上具体哪个点?

□ 我们将该点记为 $q=(u, v)$

□ 注意到两个相似的三角形:



□ 假设相机为单位尺寸, 原点在针孔 c

□ 可得 $v/1 = y/z$, 垂直坐标就是斜率 y/z

□ 类似的, 横坐标为 $u = x/z$

因此, 我们得到
 $u = x/z$
 $v = y/z$

把立方体画出来！

口重复以下简单的算法 12 次

- 每次针对立方体的一条边

- 假设相机的位置在 $c = (2, 3, 5)$

- 将边两端的 (X, Y, Z) 转换成 (u, v)

1. 将点 (X, Y, Z) 减去相机 c 得到 (x, y, z)

2. 将 (x, y) 除以 z 得到 (u, v) , 用分数表示

- 将 (u_1, v_1) 和 (u_2, v_2) 连接起来

VERTICES

A: $(1, 1, 1)$

E: $(1, 1, -1)$

B: $(-1, 1, 1)$

F: $(-1, 1, -1)$

C: $(1, -1, 1)$

G: $(1, -1, -1)$

D: $(-1, -1, 1)$

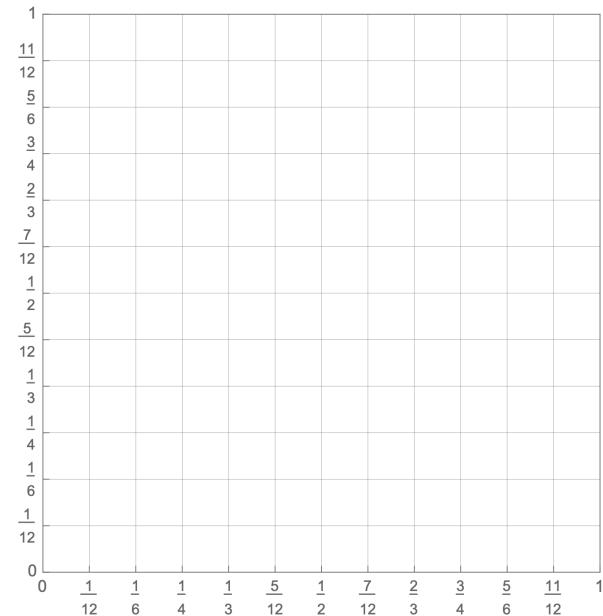
H: $(-1, -1, -1)$

EDGES

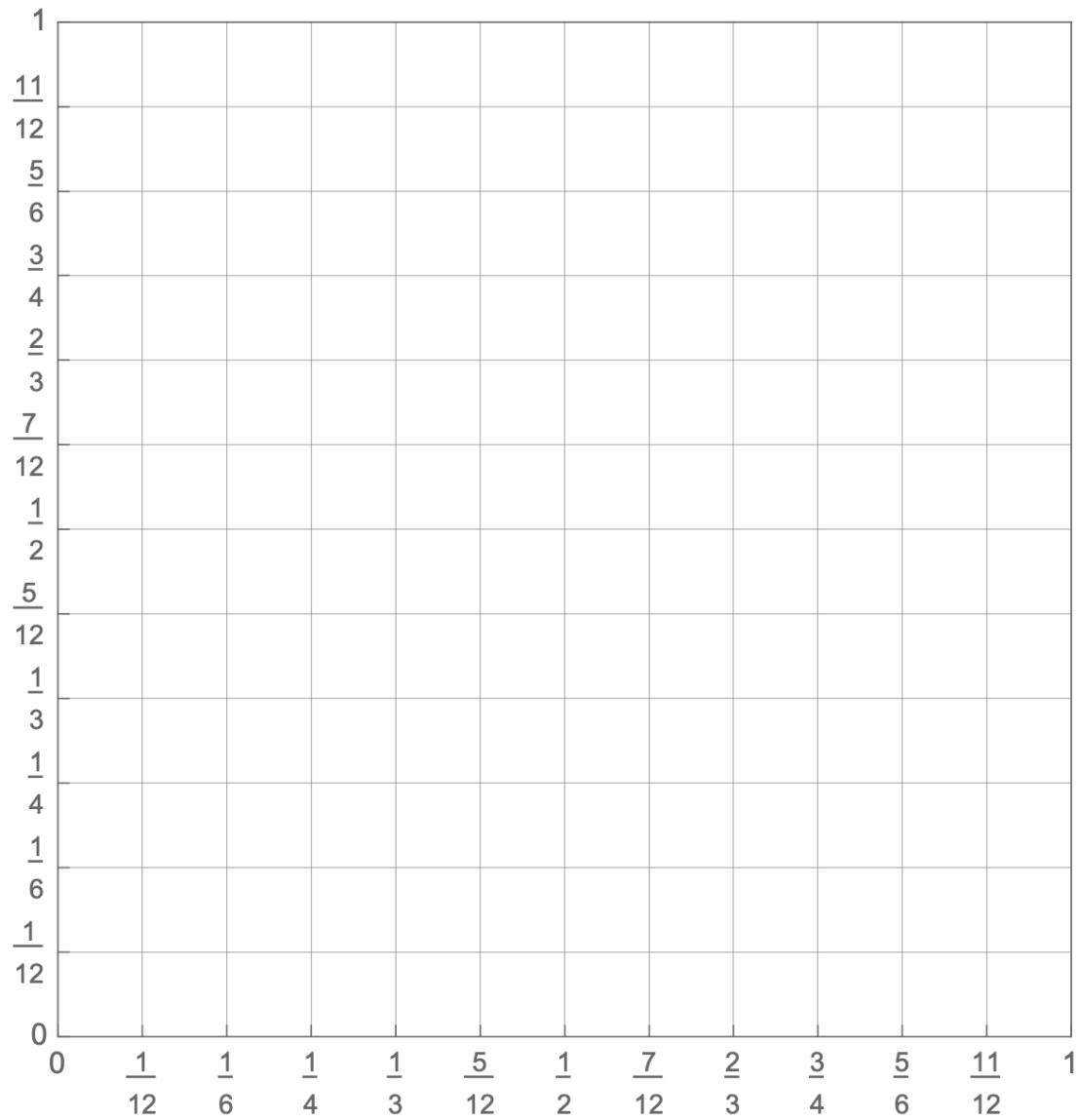
AB, CD, EF, GH,

AC, BD, EG, FH,

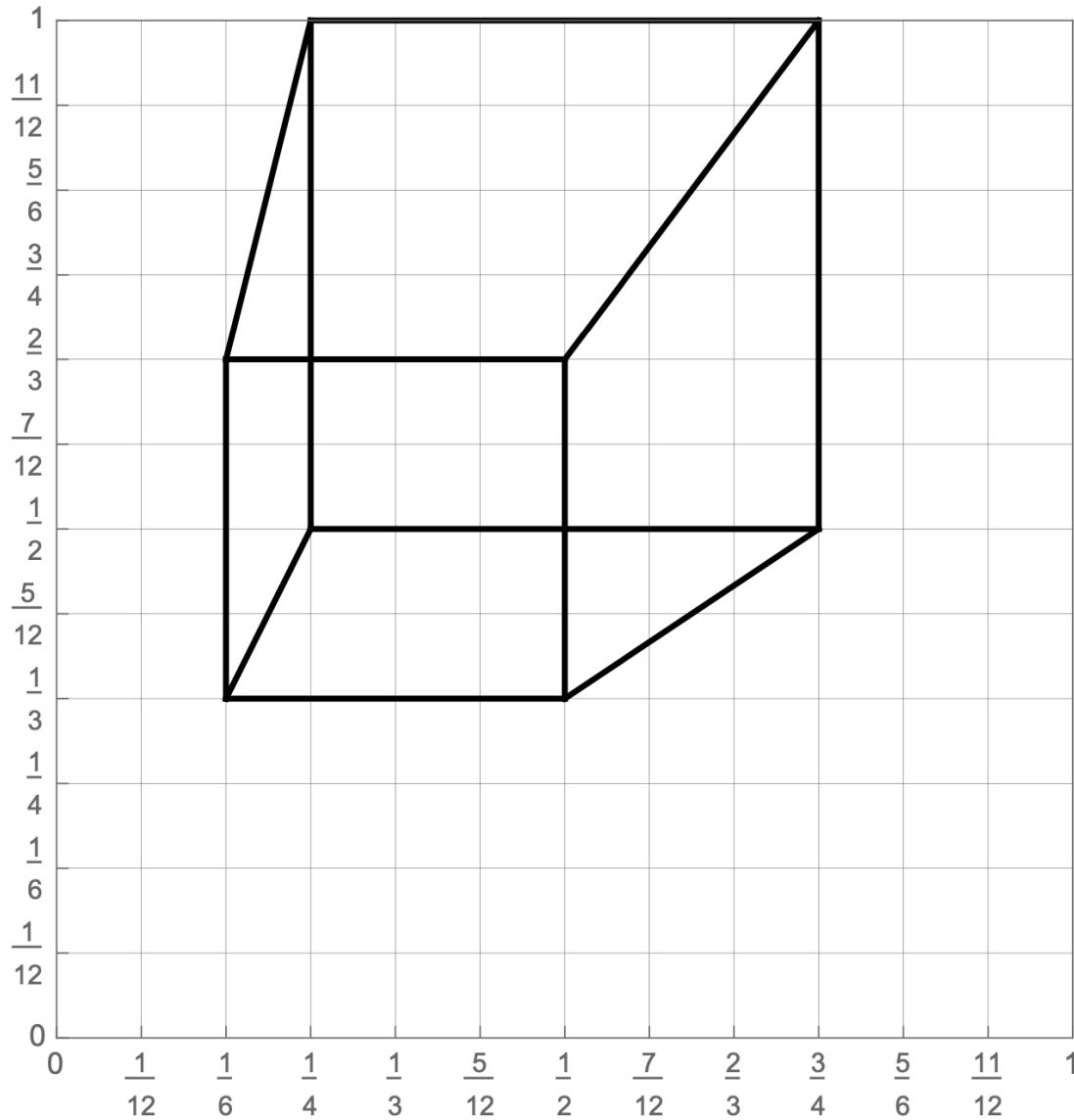
AE, CG, BF, DH



输出立方体



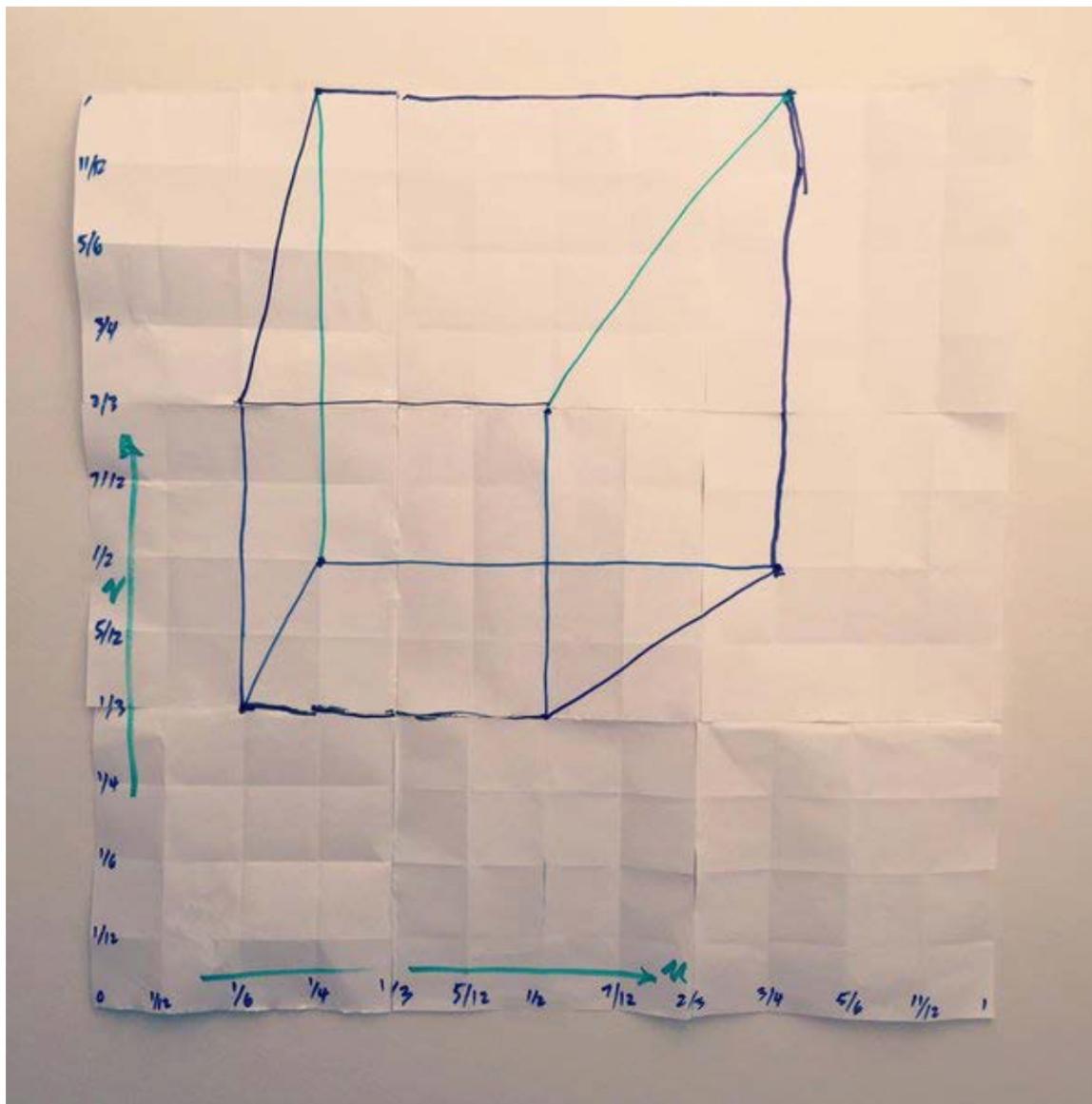
输出立方体



2D coordinates:

- A: $\frac{1}{4}, \frac{1}{2}$
- B: $\frac{3}{4}, \frac{1}{2}$
- C: $\frac{1}{4}, 1$
- D: $\frac{3}{4}, 1$
- E: $\frac{1}{6}, \frac{1}{3}$
- F: $\frac{1}{2}, \frac{1}{3}$
- G: $\frac{1}{6}, \frac{2}{3}$
- H: $\frac{1}{2}, \frac{2}{3}$

手绘结果



成功！我们以纯算法的方式（计算机程序），将数字信息转化为视觉信息。



Computation



Digital information

Virtual information



等等...

我们要怎么在计算机中画出一条线呢？

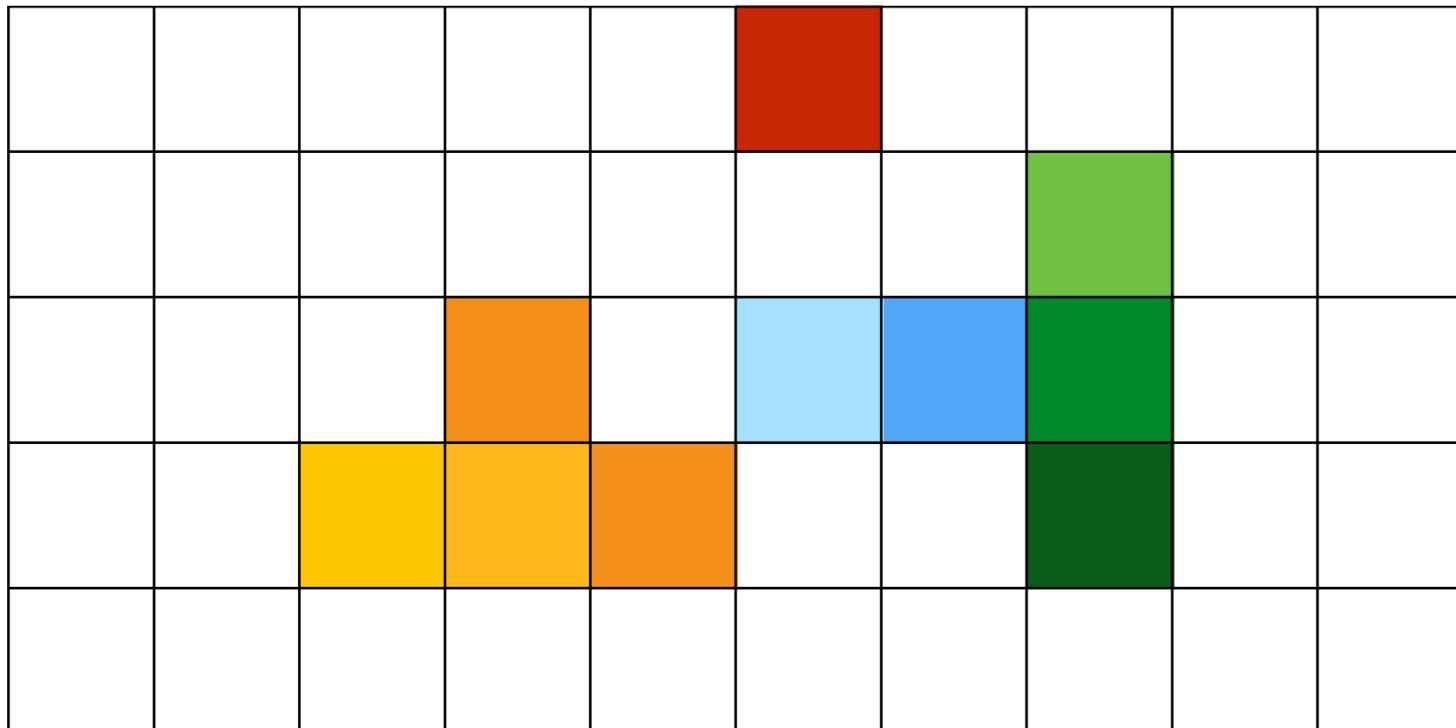
现代显示器上像素的特写照片



屏幕显示

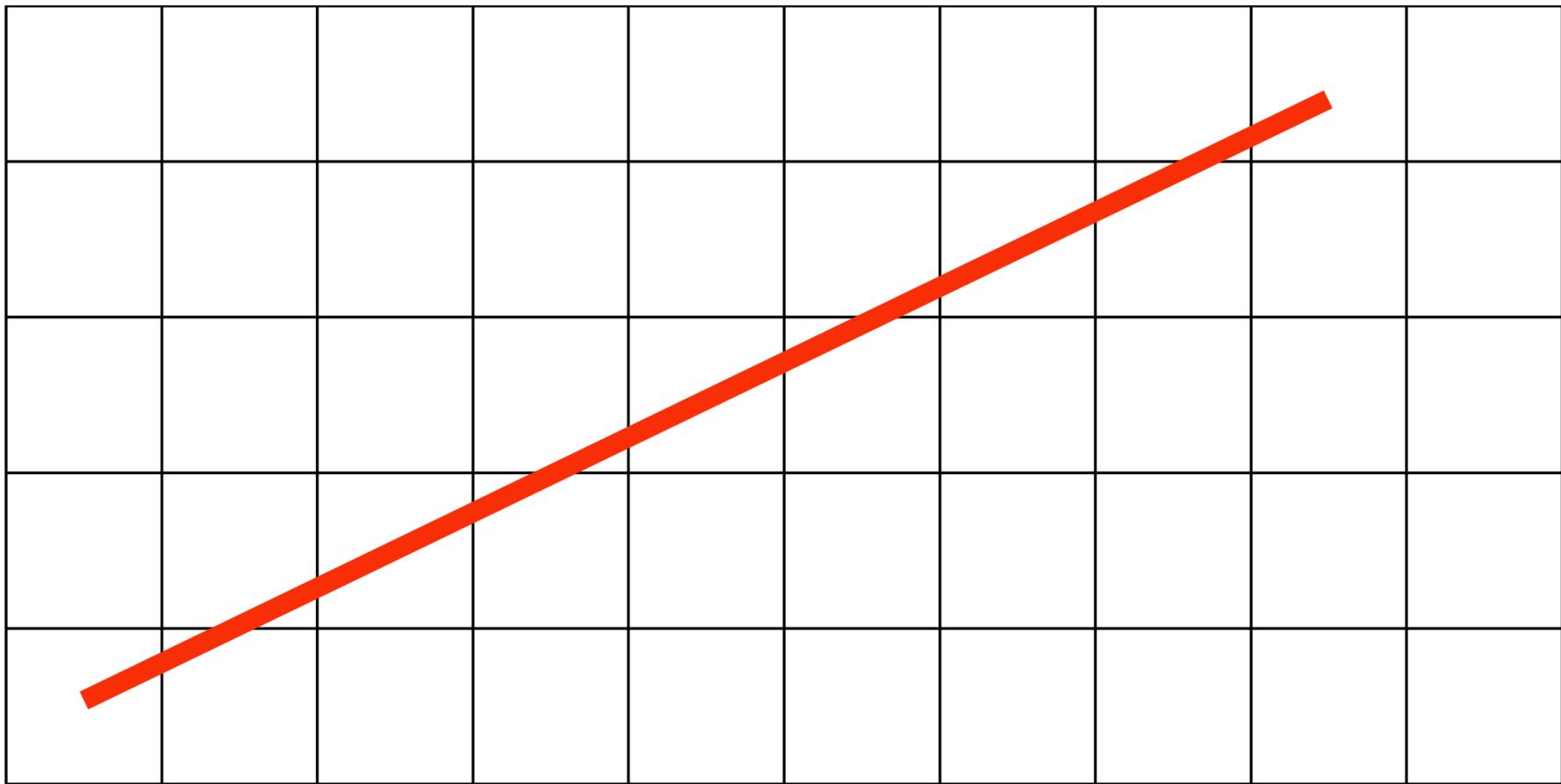
□ 屏幕常用的抽象形式

- 图像被表示为一个二维的“像素”（图片元素）网格
- 每个像素可以呈现一个独特的颜色值



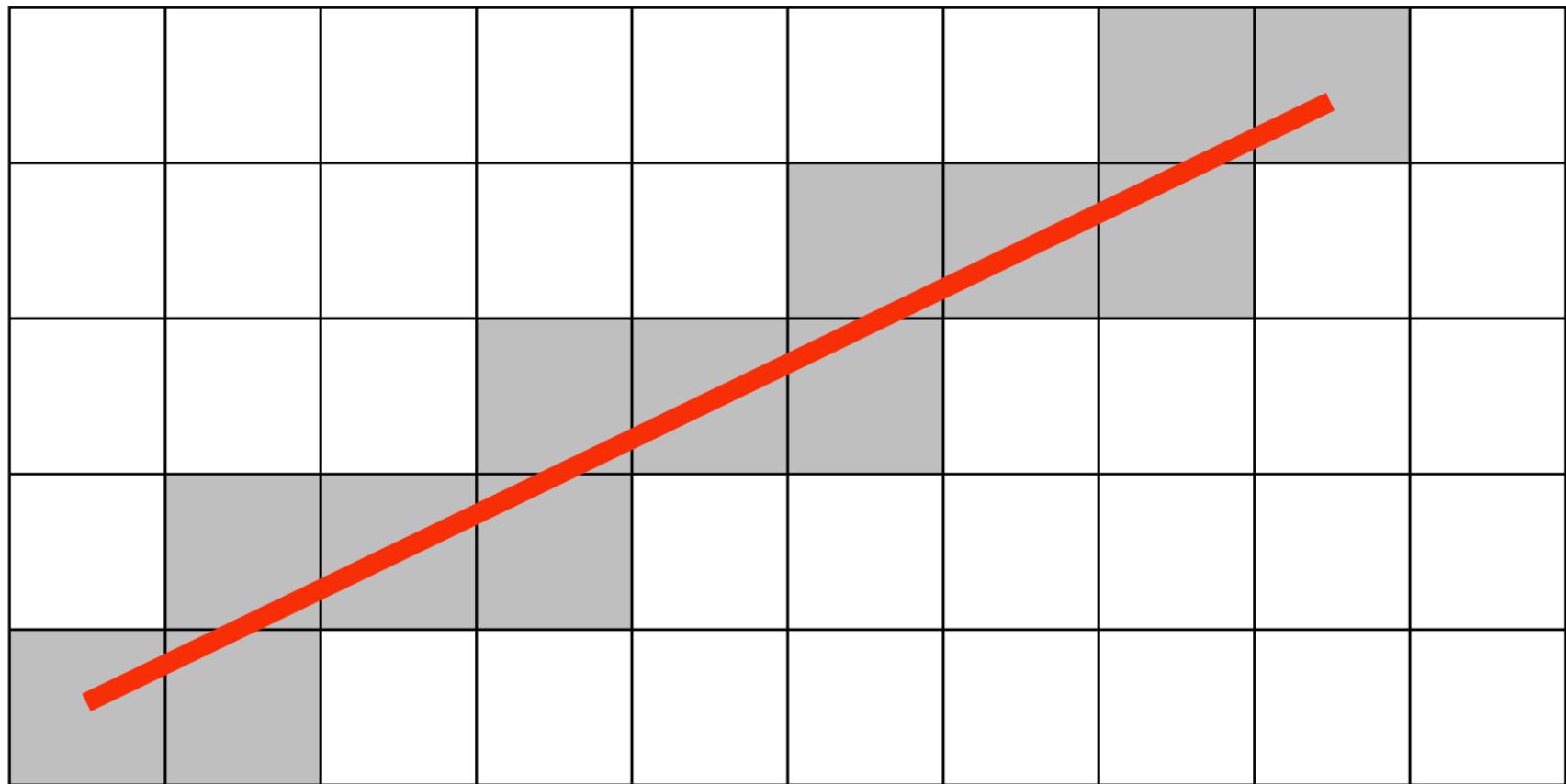
光栅化

口光栅化 (Rasterization): 将连续对象转换为光栅网格 (像素网格) 上的离散表示的过程



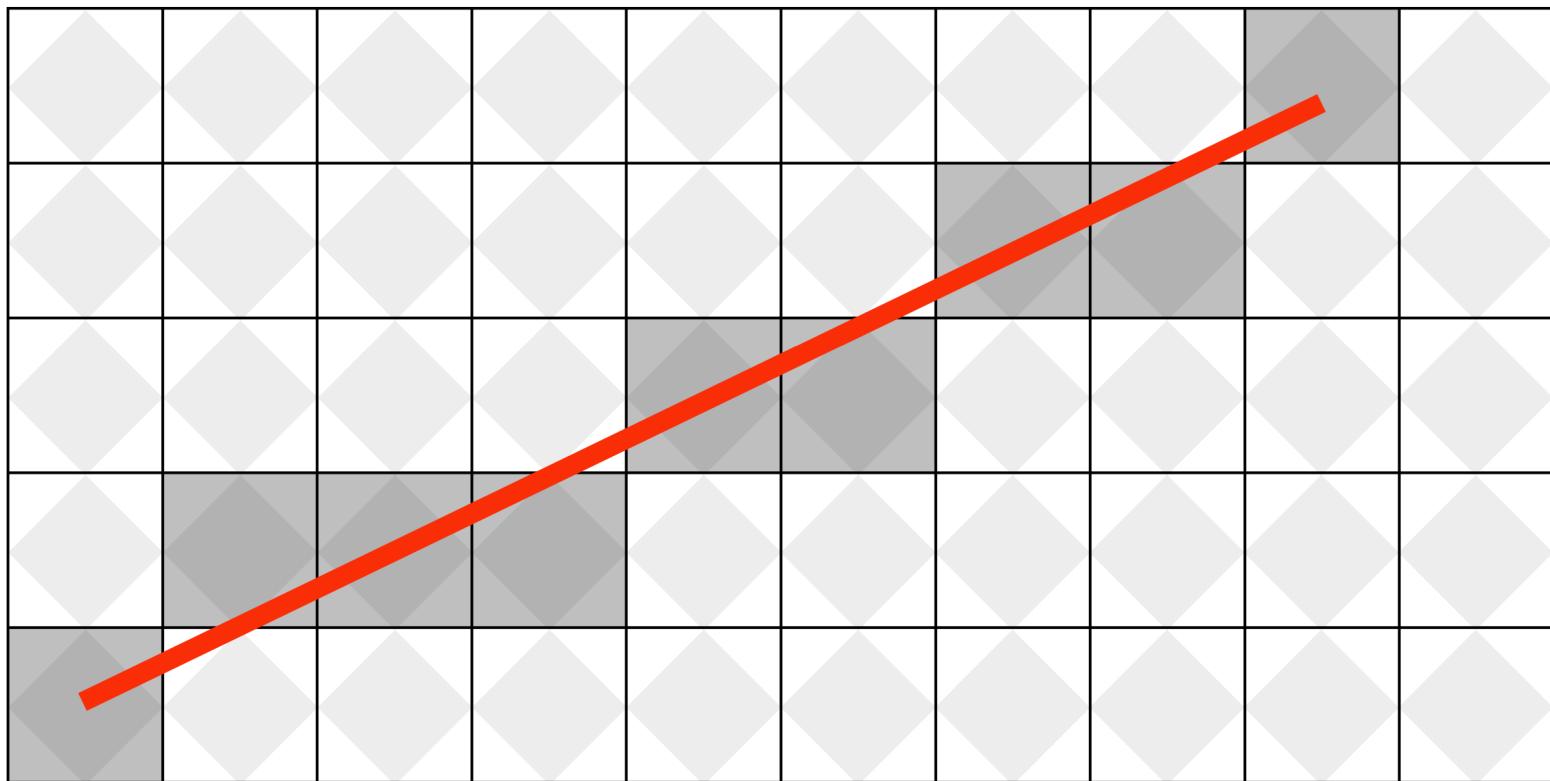
我们应该填充哪些像素来绘制直线？

点亮所有被线条穿过的像素？



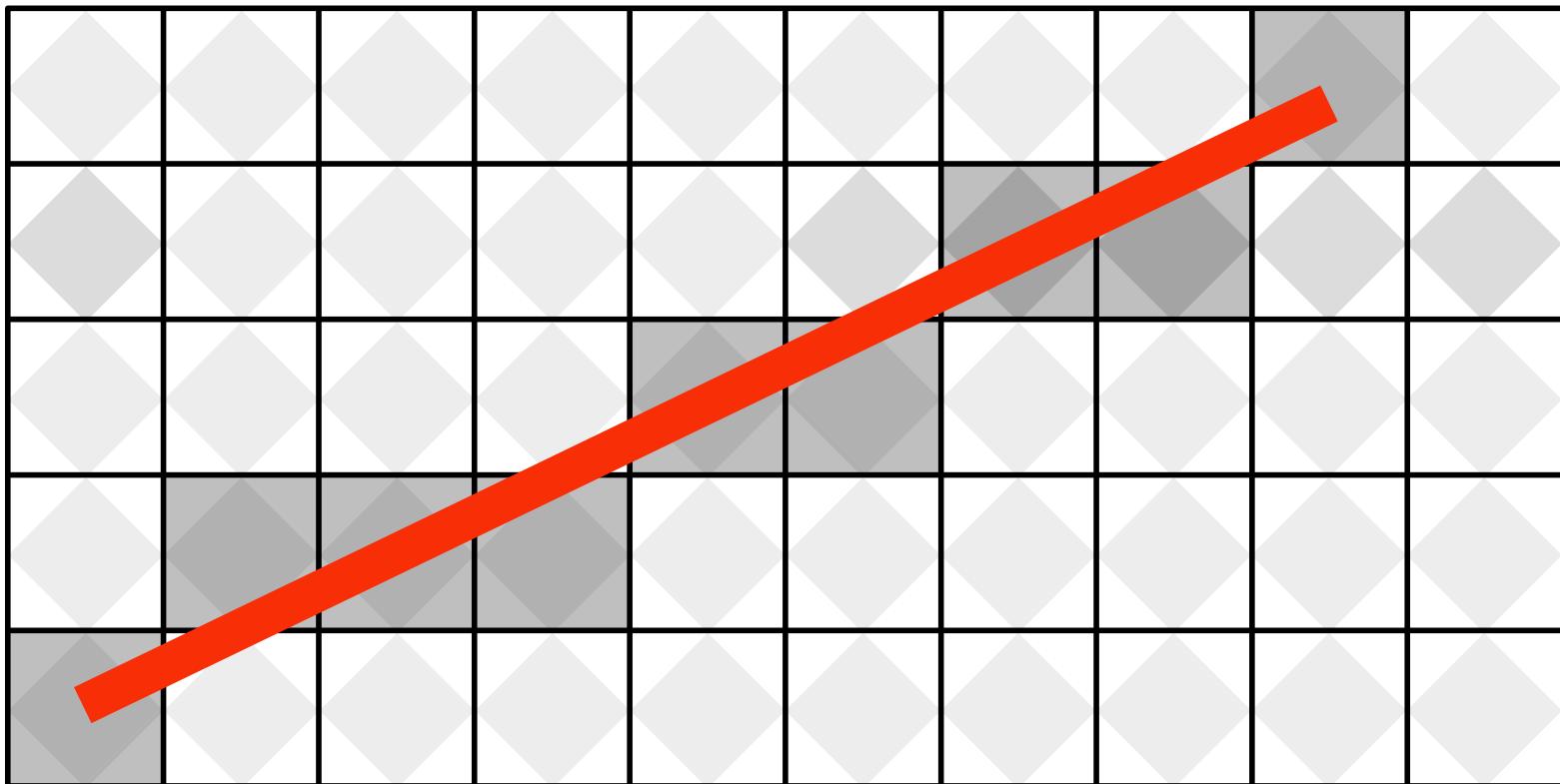
我们应该填充哪些像素来绘制直线？

钻石规则 Diamond rule (现代 GPU 使用):
如果线穿过响应的“钻石”，就点亮像素



我们应该填充哪些像素来绘制直线？

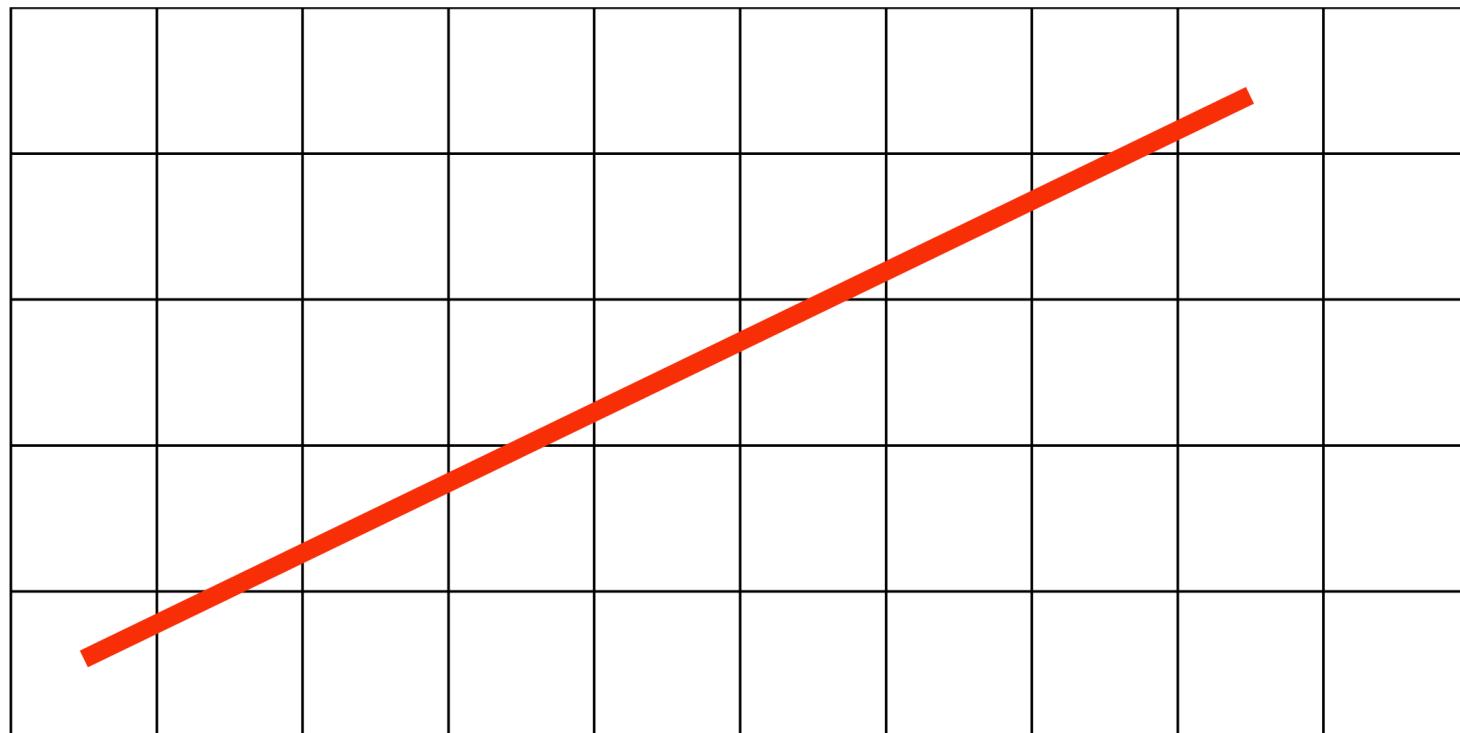
是否有更好的答案?
(考虑一条具有“厚度”的线条)



如何找到满足光栅化规则的像素？

□逐个检查图像中的每个像素，看看它是否符合条件...

- 图片的像素数量 $O(n^2)$ vs. 最多 $O(n)$ 个像素被点亮
- 是否有更好的算法？例如，计算量与图像中被点亮的像素数量成正比



增量线光栅化

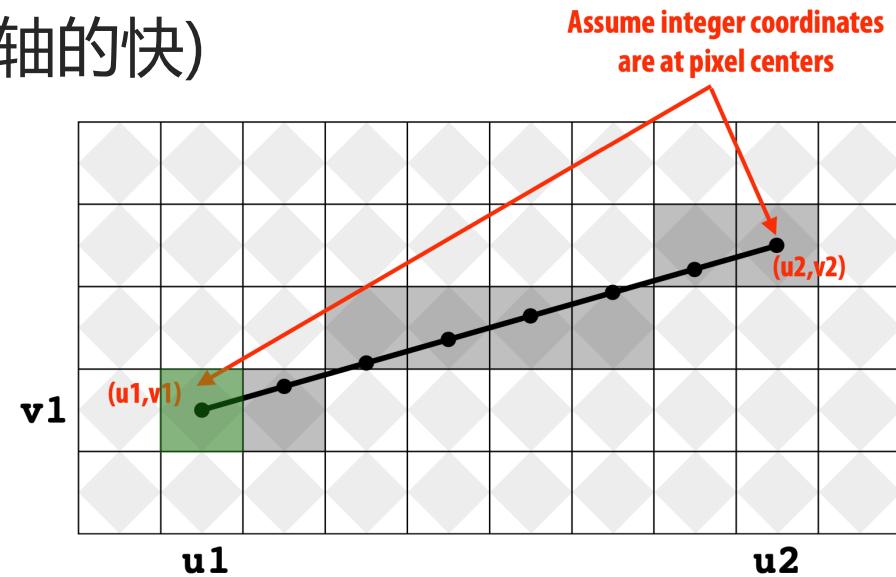
□ 我们用两个点表示一条线: $(u_1, v_1), (u_2, v_2)$

□ 线的**斜率** $s = (v_2 - v_1) / (u_2 - u_1)$

□ 考虑一个具体的例子

- $u_1 < u_2, v_1 < v_2$ (线条指向右上方)
- $0 < s < 1$ (x 轴上的变化比 y 轴的快)

```
v = v1;  
for(u=u1; u<=u2; u++)  
{  
    v += s;  
    draw(u, round(v))  
}
```



易于实现，但不是当代软件/硬件画线的唯一方式

我们现在有了第一个完整的图形算法！

Digital information

VERTICES

A: (1, 1, 1)
B: (-1, 1, 1)
C: (1,-1, 1)
D: (-1,-1, 1)
E: (1, 1,-1)
F: (-1, 1,-1)
G: (1,-1,-1)
H: (-1,-1,-1)

EDGES

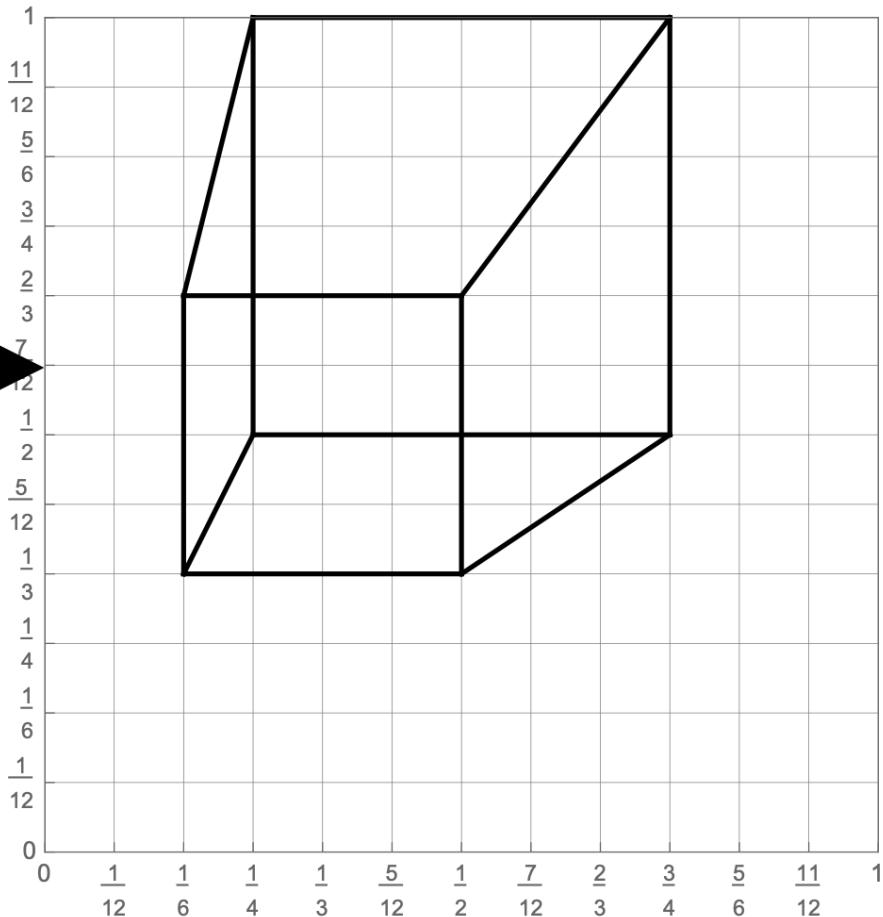
AB, CD, EF, GH,
AC, BD, EG, FH,
AE, CG, BF, DH

CAMERA

C = (2,3,5)

computation

Visual information



This is fundamentally what computer graphics is all about...

到目前为止，我们只画了一个简单的立方体线条

为了画出更真实的图片，需要对我们的世界进行更丰富的建模：

几何 (Geometry)
材料 (Materials)
灯光 (Lights)
相机 (Cameras)
运动 (Motion)

...

随着课程进行，我们将探索更多

Today's topics

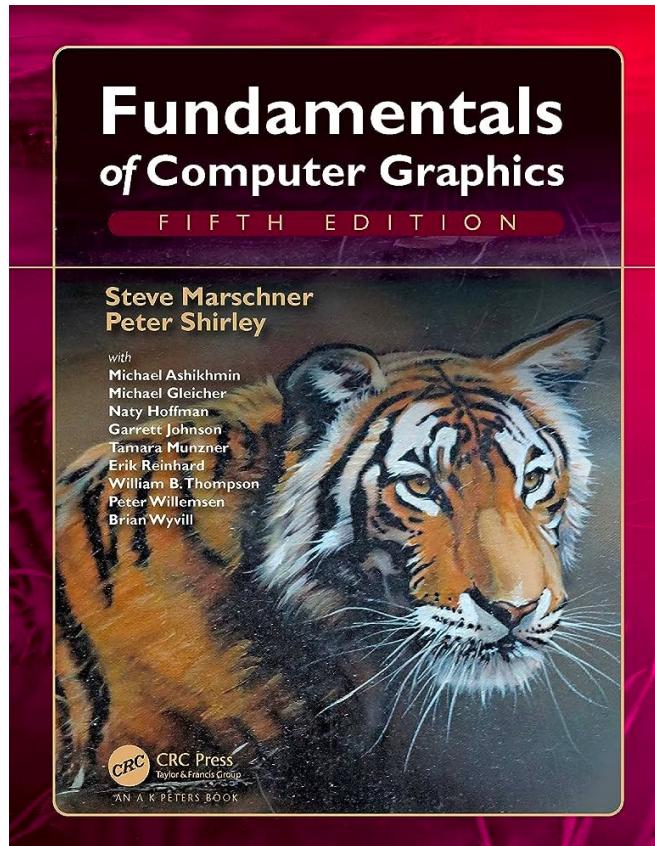
□ 计算机图形学介绍

□ 绘制一个简单的 3D 图像

□ 课程内容介绍

□ 课程安排及评分

课程教材及参考书目



课程主页

SSE315计算机图形学 (Computer Graphics)

中山大学软件工程学院本科三年级专业选修课

课程信息

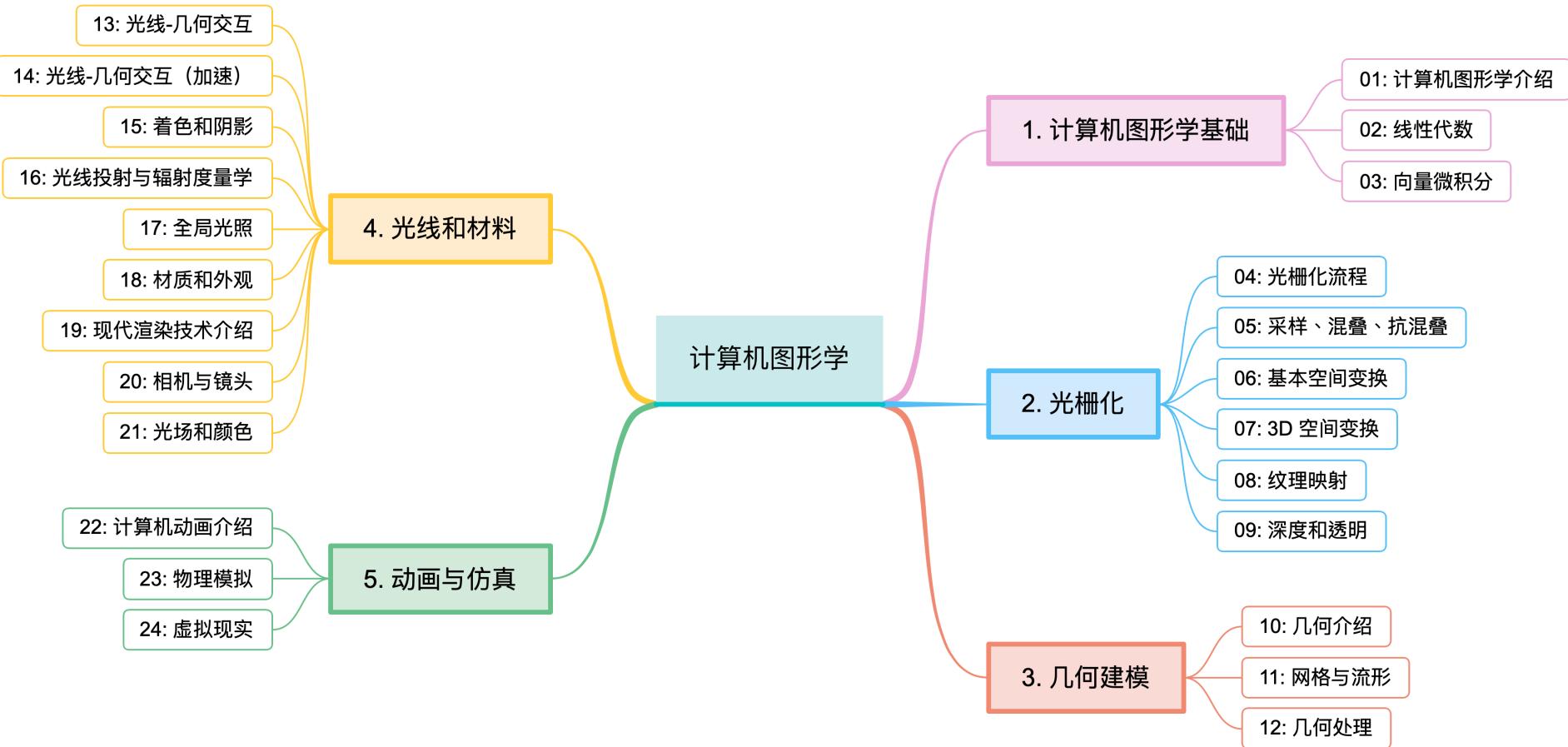
- 任课教师：陈壮彬
- 办公室：综合实验楼310-9
- 时间：1-17周，周一9、10节；1-9周，周三9、10节
- 地点：珠海校区-教学大楼-珠海C301

课表

周次	主题	内容	下载
1	计算机图形学基础	计算机图形学介绍	课件

<https://zbchern.github.io/sse315.html>

课程课题



基础部分 Basics

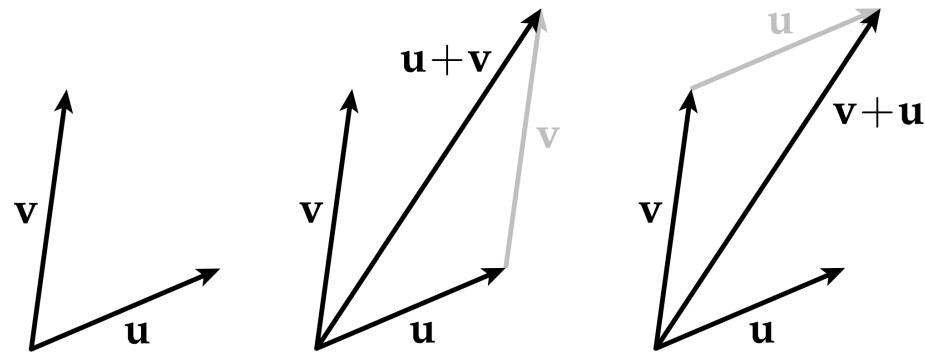
口计算机图形学的介绍以及线性代数和向量微积分

1. 计算机图形学基础

01: 计算机图形学介绍

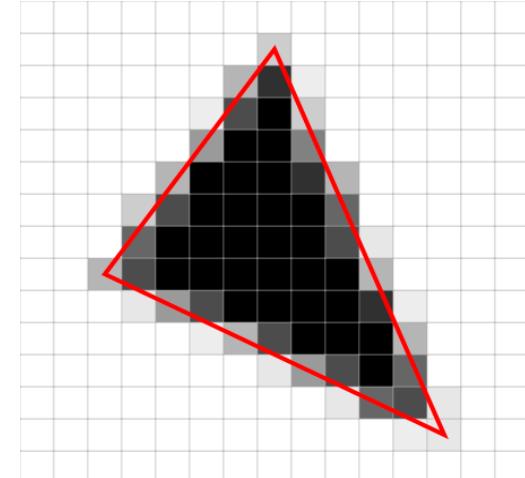
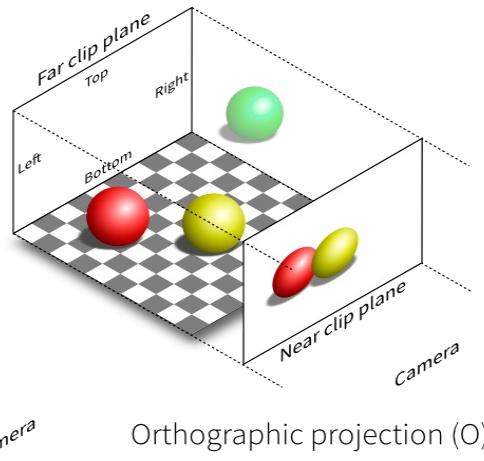
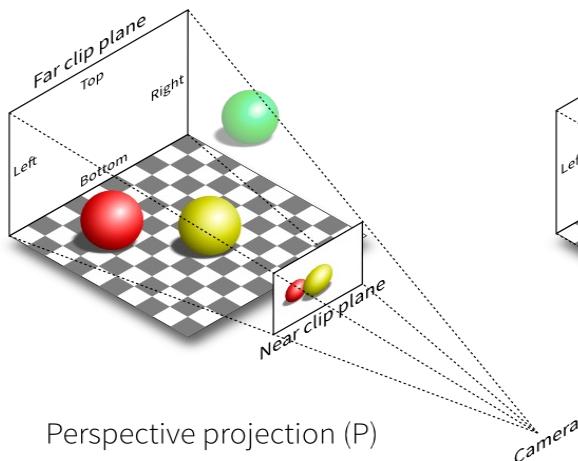
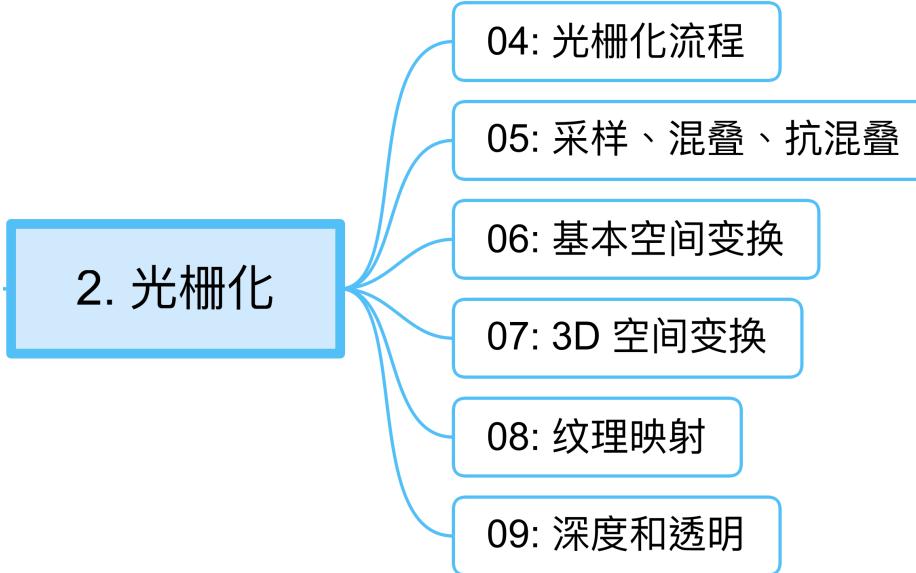
02: 线性代数

03: 向量微积分



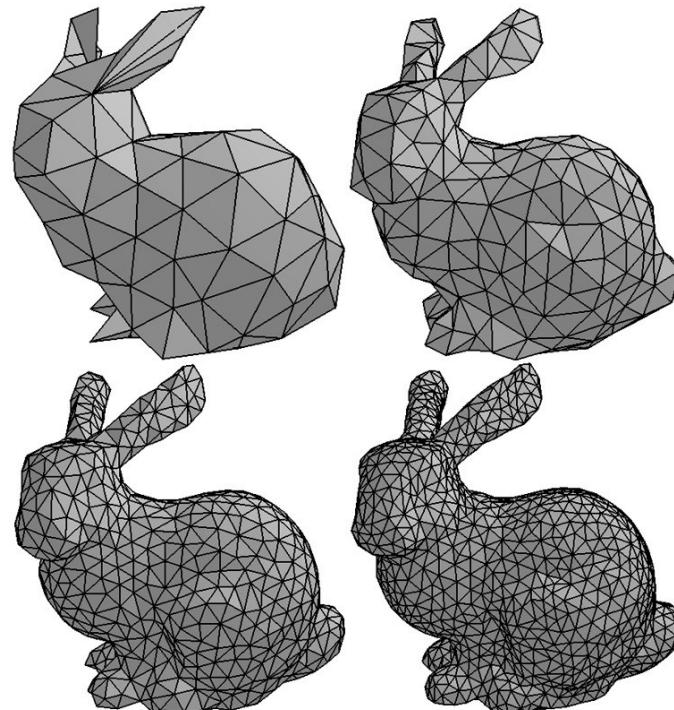
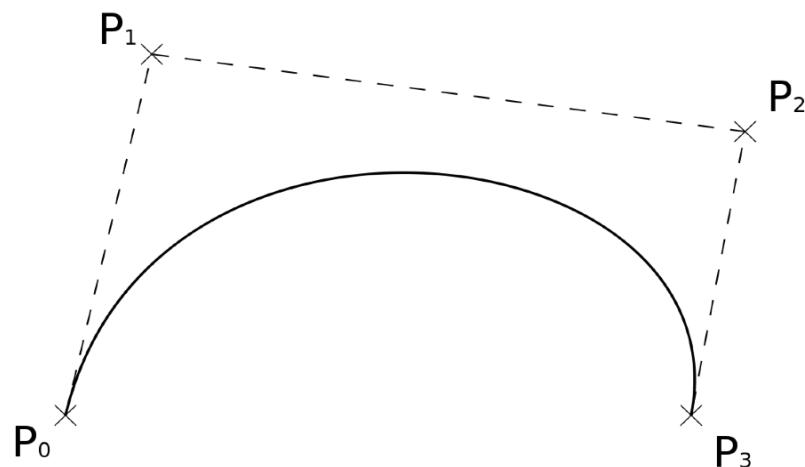
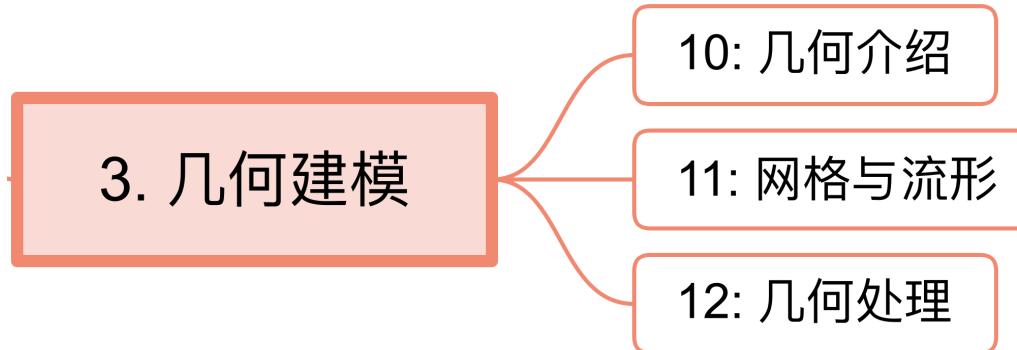
光栅化 Rasterization

将图形从其**几何表示**转换为**像素表示**的过程



几何建模 Geometric modeling

口介绍几何在计算机图形学中的表示



光照和材料 Lighting & Materials

口介绍计算机图形学中的光照和材质处理

13: 光线-几何交互

14: 光线-几何交互（加速）

15: 着色和阴影

16: 光线投射与辐射度量学

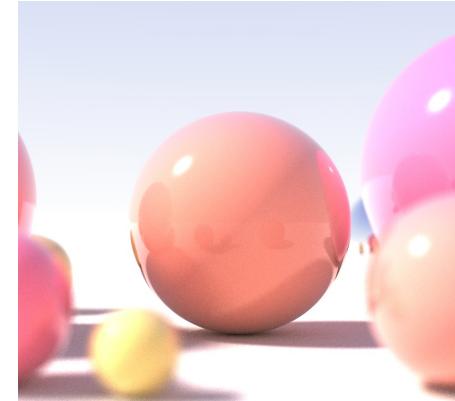
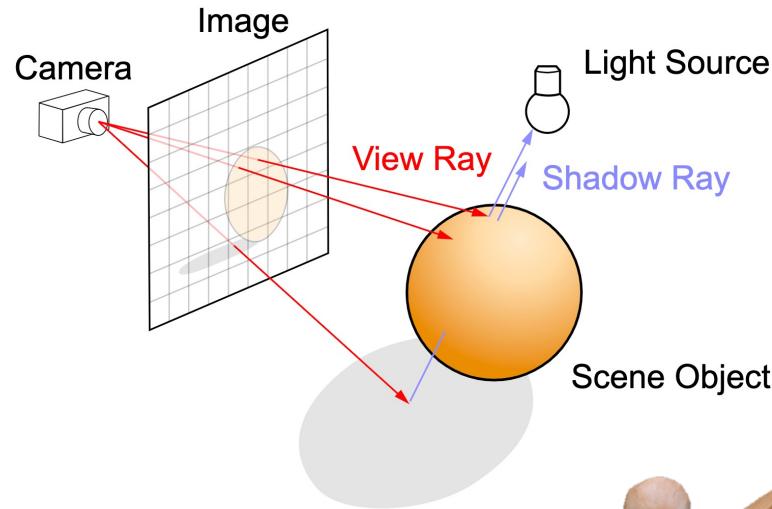
17: 全局光照

18: 材质和外观

19: 现代渲染技术介绍

20: 相机与镜头

21: 光场和颜色



4. 光线和材料



动画与模拟 Animation & Simulation

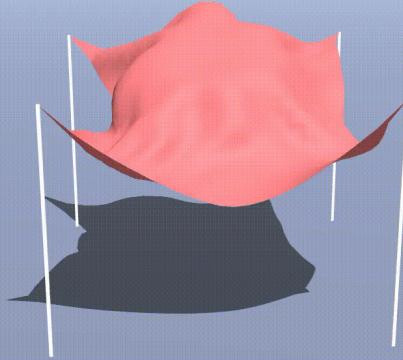
□介绍具有动态性和交互性的计算机图形学

22: 计算机动画介绍

23: 物理模拟

24: 虚拟现实

5. 动画与仿真



Today's topics

□ 计算机图形学介绍

□ 绘制一个简单的 3D 图像

□ 课程内容介绍

□ 课程安排及评分

Teaching staff

□任课教师

- 陈壮彬
- 办公室：公共实验楼 310-9
- 邮箱：[chenzb36@mail.sysu.edu.cn](mailto:chenzhb36@mail.sysu.edu.cn)

□助教

- 王爵鹏
- 办公室：公共实验楼 308
- 邮箱：wangjp39@mail2.sysu.edu.cn



课程安排

□ 上课 (52 学时理论课, 无实验课)

- 9-10 节 (19:00 pm - 20:40 pm) , C301
- 前九周: 周一和周三
- 后八周: 周一

□ 4 次左右的课后作业

- 理论题 and/or 编程题
- 独立完成, 严禁抄袭

□ 期末

- 课程论文

评分标准

成绩项	内容	占比	总成绩占比
平时成绩	考勤	10%	40%
	课后作业	90%	
期末考查	课程论文	100%	60%

课程微信群

Group: SSE315-2025-计算机
图形学





中山大學 软件工程学院
SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

谢谢

陈壮彬
软件工程学院
chenzhb36@mail.sysu.edu.cn