



计算机图形学基础

胡事民

清华大学计算机科学与技术系



今天的课程内容

- 图形学中的一些重要概念
 - 颜色视觉
 - 图像和像素
 - 三角网格模型
 - 光照模型与明暗处理
 - 视点变换和视点方向



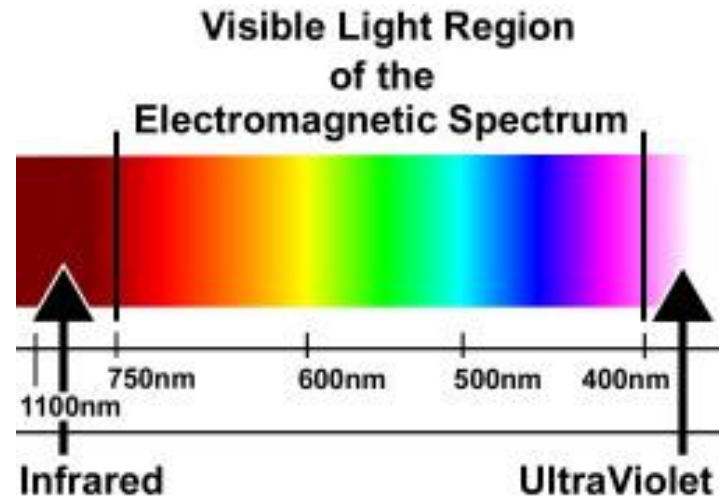
颜色视觉

- 什么是颜色?

- 颜色是人眼对不同波长的光的能量的感知:

- 不同波长的电磁波 (electromagnetic waves) 对应不同的颜色;

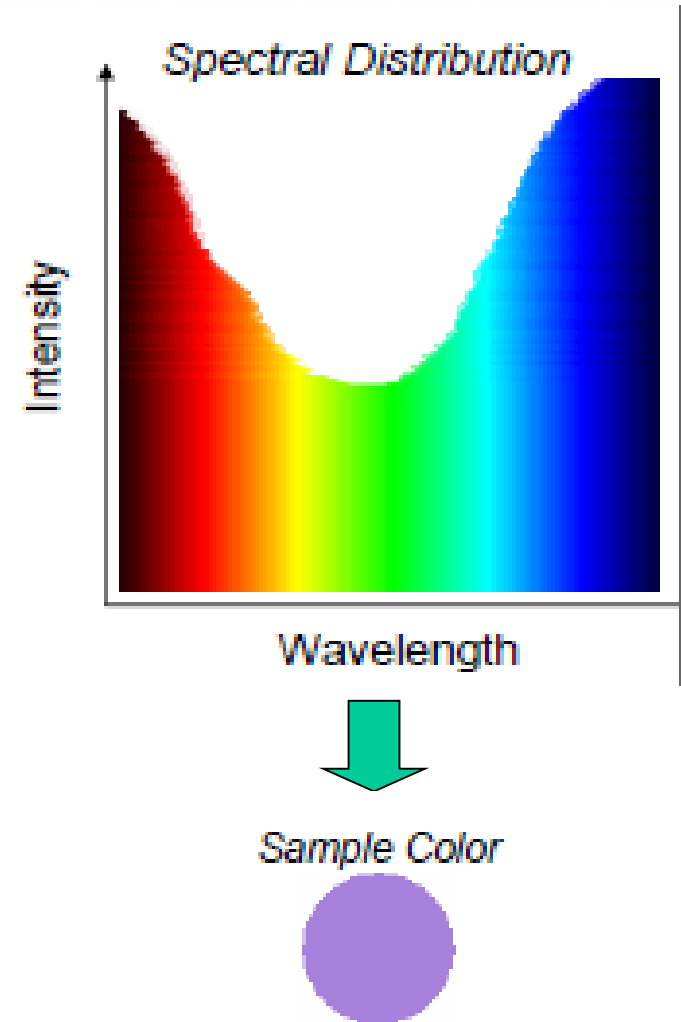
- 人眼能感知到的光称为可见光, 其波长范围在380nm 到 760 nm 之间





光的谱分布

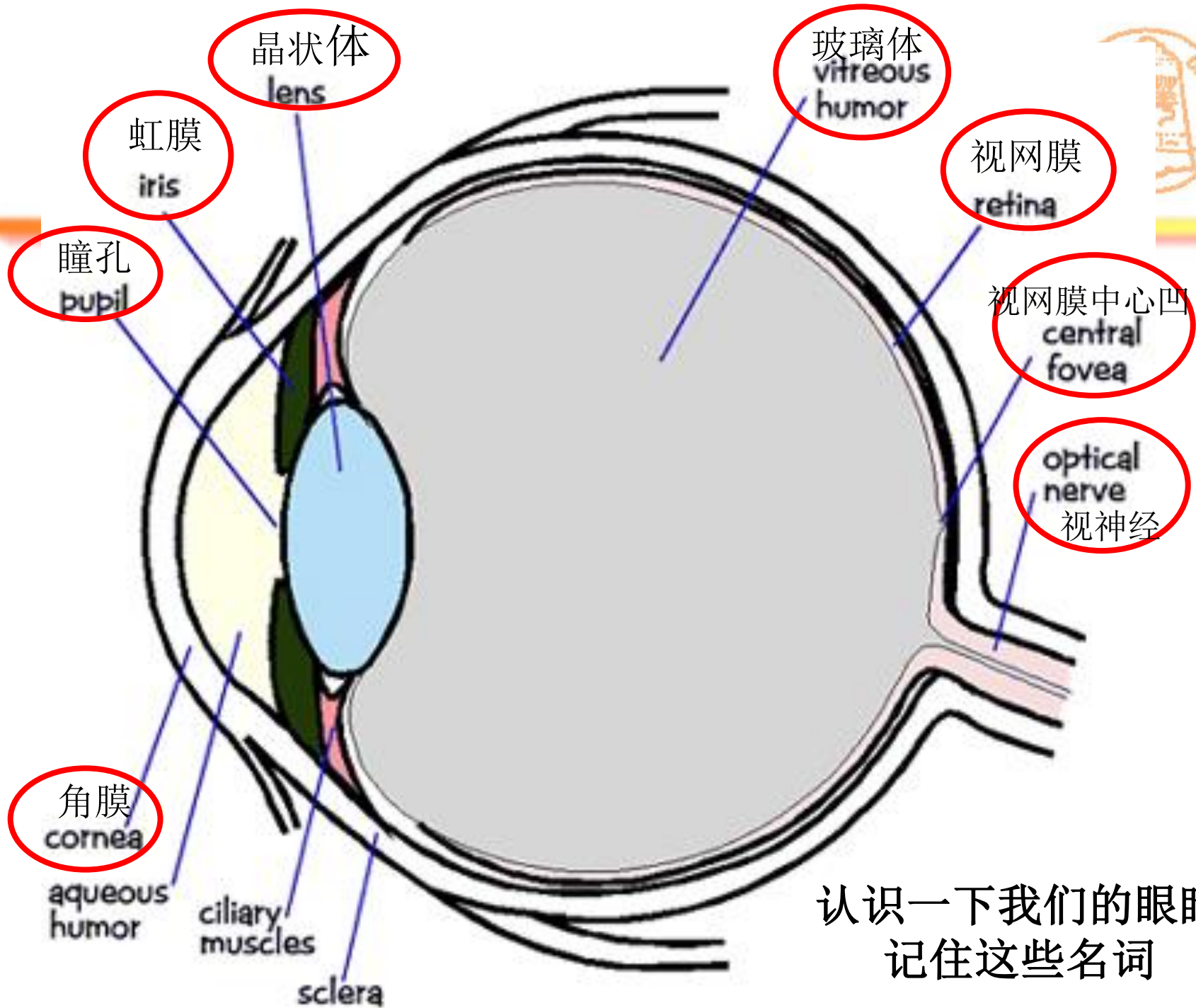
- “光”是由不同波长的电磁波按照某种能量分布混合叠加而成
 - 例如，“白光”是由所有可见波长的电磁波以相等的强度混合得到的
- 谱分布：光在各个可见波长分量的强度分布函数称为光的谱分布





光的谱分布

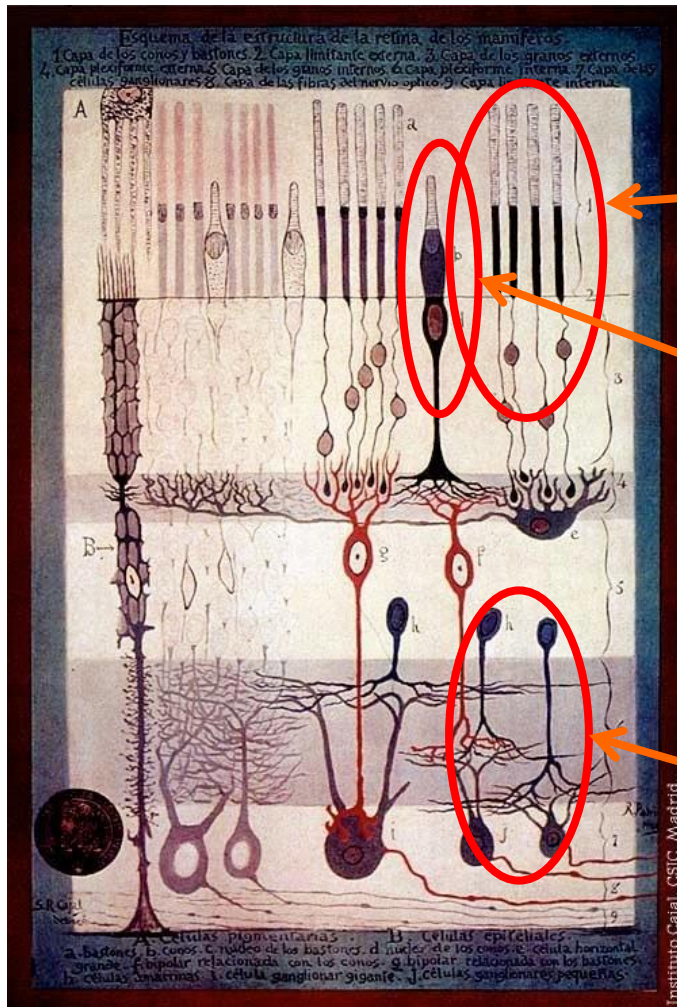
- 与光类似，颜色也可以使用谱分布函数来进行描述
- 然而，使用谱分布函数来表示颜色，不仅复杂，而且谱分布函数和颜色之间并不是1-1映射
- 实际上，不同的谱分布函数可能对应为同一种颜色，这种现象被称为“异谱同色”



认识一下我们的眼睛
记住这些名词



视网膜——我们的感知器!



杆状细胞(rod),

对亮度更敏感（因而在夜间的视力由杆状细胞主导）

锥状细胞(cone),

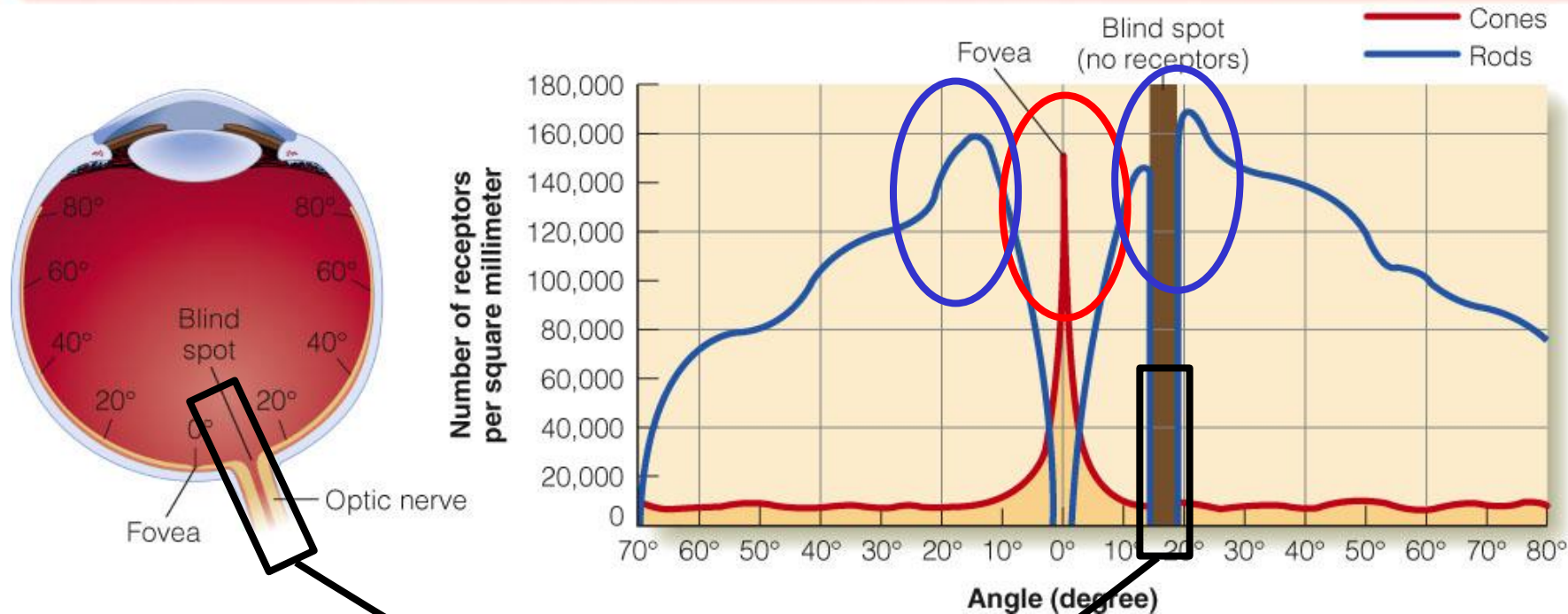
感知色彩，主要分为对红光敏感，对绿光敏感，对蓝光敏感的三类。

神经节细胞(ganglion),

神经节细胞负责将这些细胞受到的刺激传递给视神经。



杆状细胞与锥状细胞的密度



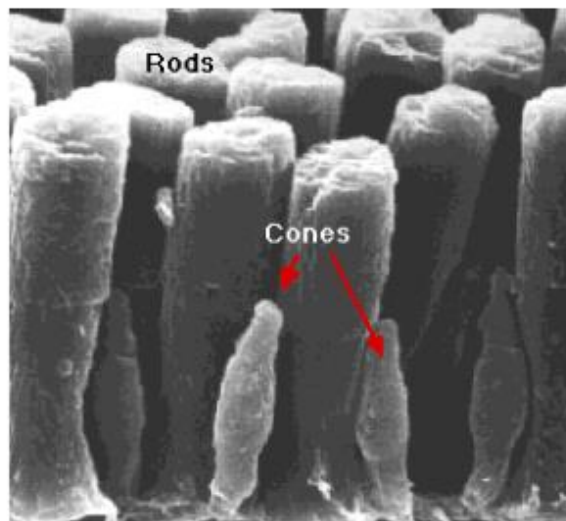
右图显示了各种感光细胞在视网膜上的分布。横坐标是相对于视网膜中心凹的角度，纵坐标则是相应的感光细胞数目。

其中的红色曲线是锥细胞在视网膜上的分布，我们可以看出锥细胞主要集中在视网膜中心凹处。因而人眼在视野中心处辨色力最强。

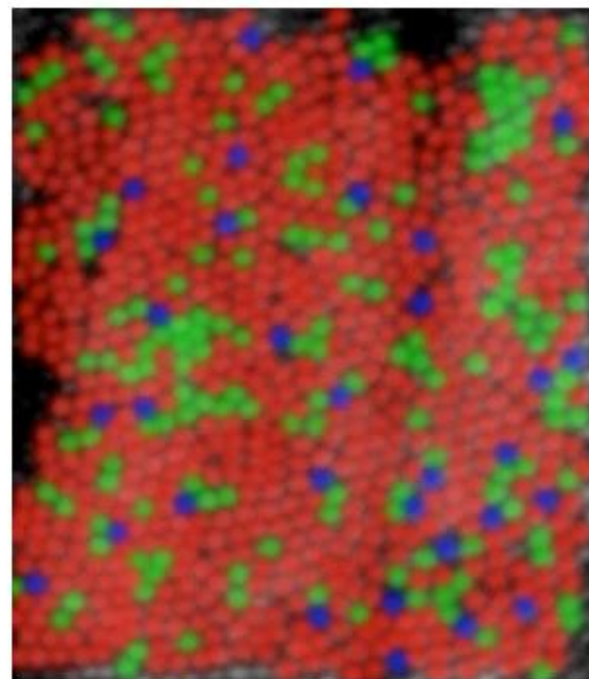


为何采用RGB 颜色空间？

- 左图是视网膜组织的显微镜成像图。人眼中大约有1.2亿个杆状细胞与五六百万个锥状细胞。右图中细胞显示的颜色表示它哪种颜色的光刺激最敏感。



120 million rods
5-6 million cones



R 63% - G 31% - B 6%



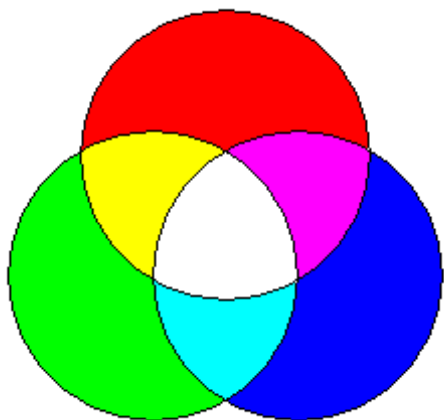
RGB 颜色空间

- 在所有用于表示色彩的颜色空间中，**RGB** (**红绿蓝**) 颜色空间在计算机图形学中的使用最为广泛：
 - 一种颜色通过三通道向量 (r, g, b) 来表示；
 - 部分常用操作可以转化为对 RGB 三通道分别独立处理；
 - 在计算机中，通常将 r, g, b 三个分量分别规整化为 $[0, 1]$ 内的浮点数；或 $[0, 255]$ 内的 8bit 无符号整数

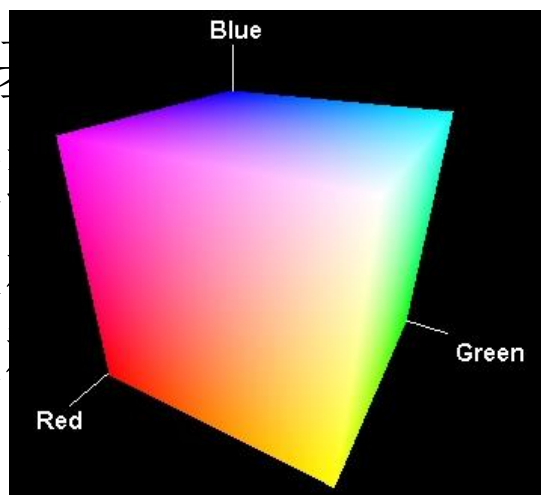


RGB 颜色空间

- 具体来说，颜色表示为三个基本色：**红色 (R)**，**绿色 (G)**，**蓝色 (B)** 的线性组合：
– $C = rR + gG + bB$



绿蓝作为
: 的三刺激
膜中有三
绿、蓝三



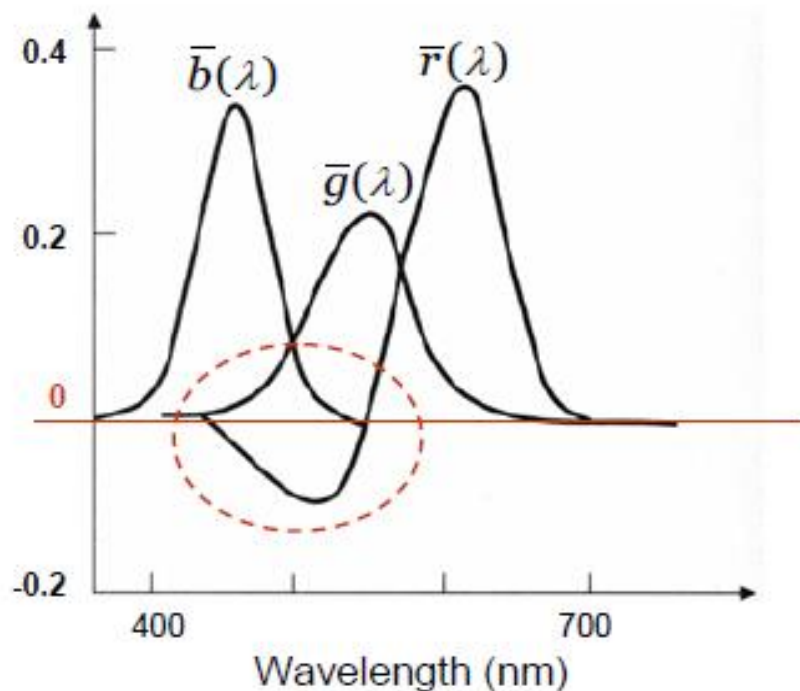
细胞，

RGB 颜色空间的图示



RGB 颜色空间

- 然而，一部分色彩无法表示成 R, G, B 光波的正线性组合，这是 RGB 颜色空间的一个缺点
- 右图为所有可见光对应的 RGB 分量图：





其他颜色空间

- 其他常用的颜色空间还包括：
 - CMY
 - HSV
 - CIE XYZ



CMY 颜色空间

- CMY: 采用不同于 RGB 的另外一组基本颜色:
 - Cyan (青), Magenta (品红), Yellow (黄): 分别是 R, G, B 的补色 (complements)

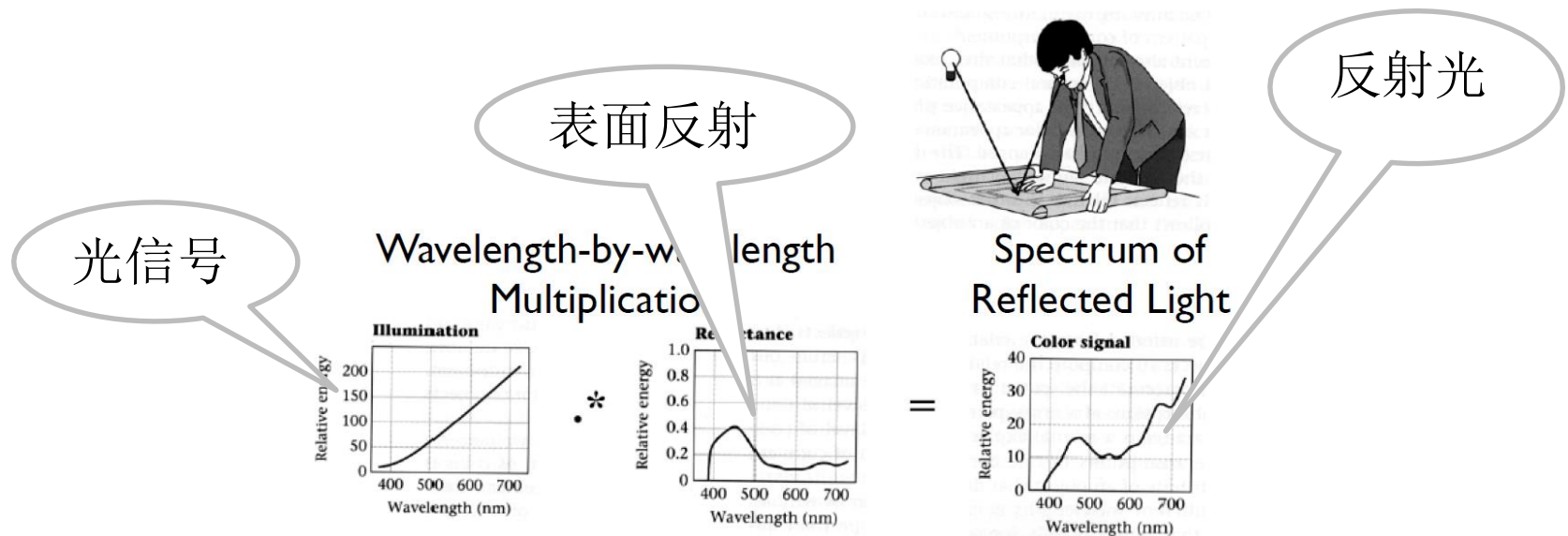
$$\begin{array}{c} \boxed{\text{C}} \\ \boxed{\text{M}} \\ \boxed{\text{Y}} \end{array} = \begin{array}{c} \boxed{1} \\ \boxed{1} \\ \boxed{1} \end{array} - \begin{array}{c} \boxed{\text{R}} \\ \boxed{\text{G}} \\ \boxed{\text{B}} \end{array}$$

- CMY 被称为“减色系统”，而 RGB 被称为“加色系统”：
 - RGB — (0, 0, 0)为黑, (1, 1, 1)为白
 - CMY — (1, 1, 1)为黑, (0, 0, 0)为白,印刷业只能用CMY模型



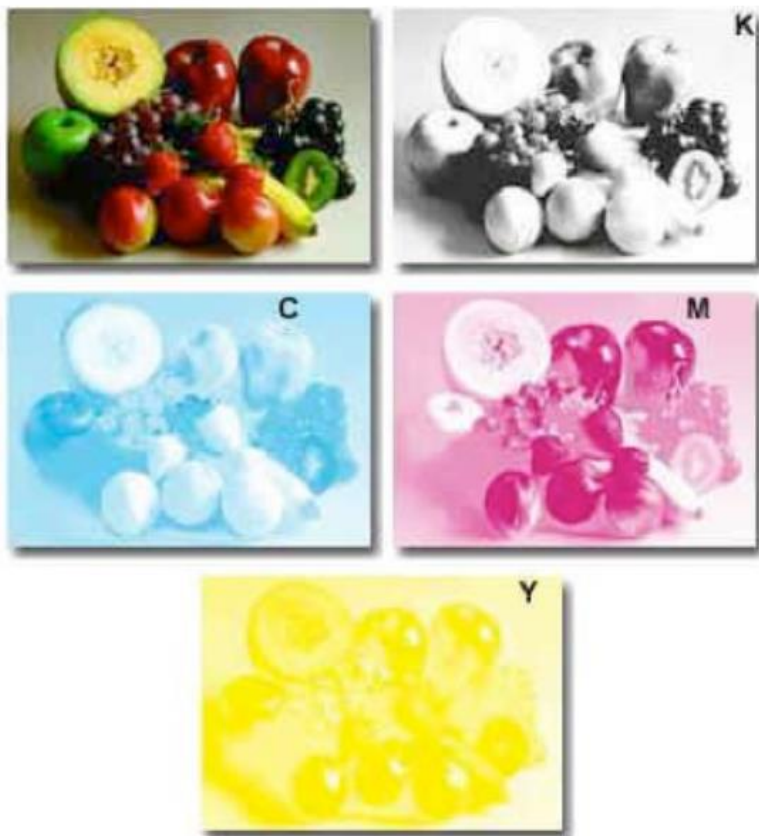
RGB（加色）-CMY（减色）

- 加性混合主要应用于主动发光的物体，如液晶显示器、电视机等。
- 减性混合主要应用于被动发光的物体，其颜色由物体表面的反射参数决定。





- 打印机的CMYK颜色系统（减性混合）

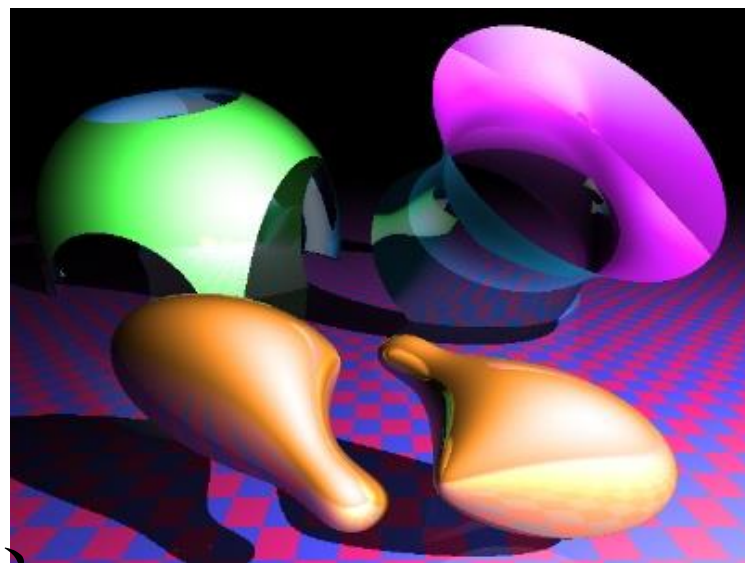


油墨本身并不能发光，打印出物体的颜色是通过反射光的形式表现出来的。



HSV 颜色空间

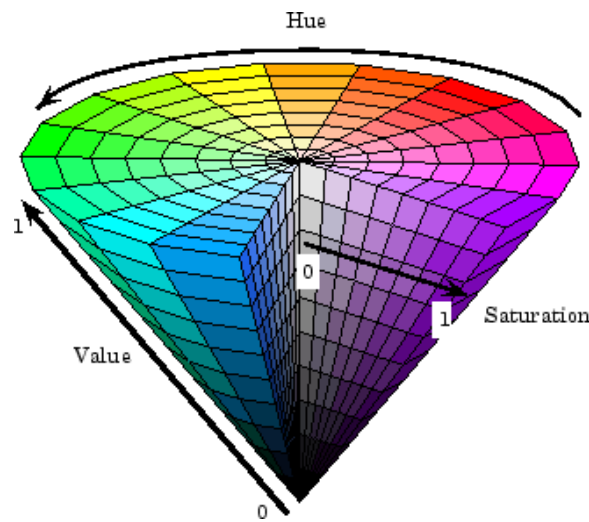
- 在8bit的 RGB 颜色空间中，可以包含多达 $256*256*256 = 16,777,216$ 种可能的颜色
- RGB 颜色空间不够符合直觉，无法做出更鲜艳/更亮之类的调整，难以描述和定位如此大量的颜色
- HSV 系统则提供了一个直观的方法来对颜色进行准确的选择
- HSV 颜色空间应用于：艺术领域，以及图像处理、分形图像...





HSV颜色空间

- HSV：圆锥形的颜色空间
 - **Hue** (色调) 也叫色相，它描述了色彩的本征属性，即我们常说的一个色彩是红、橙、黄、绿、青、蓝、紫，等等
 - **Saturation** (饱和度) 也叫纯度：饱和度越低，色彩越白
 - **Value of brightness** (亮度)：亮度越低，色彩越黑



HSV更好地描述了人眼对颜色的感知距离，
大部分软件采用基于HSV空间的调色板



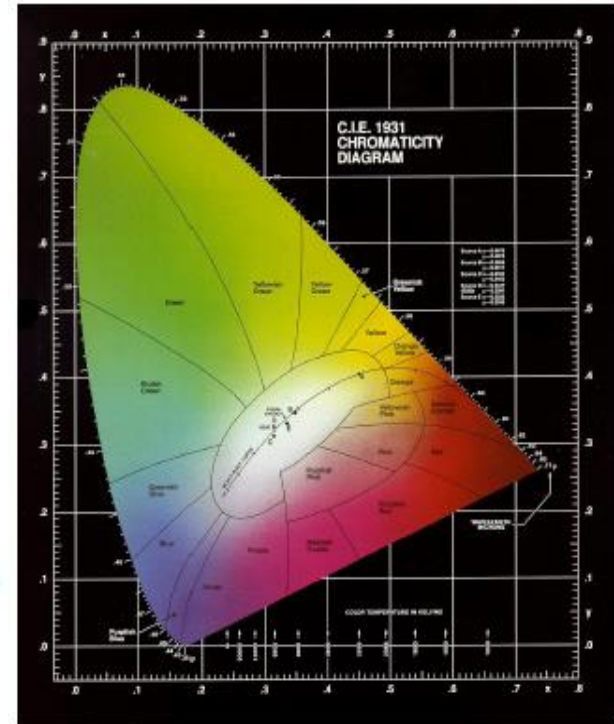
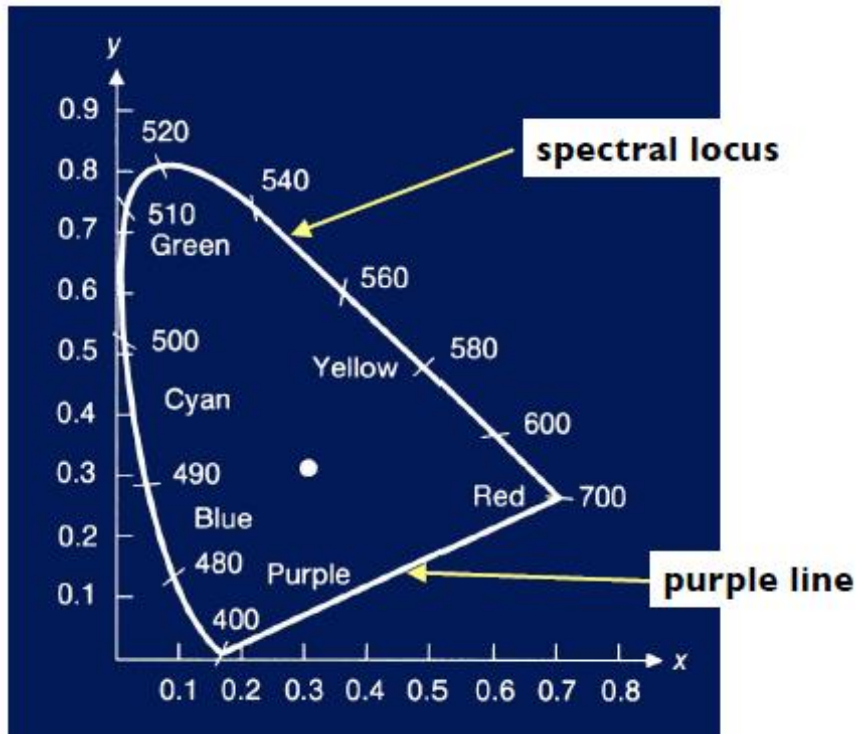
CIE XYZ 颜色空间

- CIE XYZ 颜色空间
 - 由国际发光照明委员会于1931年提出
- CIE XYZ 色彩空间可以表示所有可感知的颜色 (RGB 空间不能)
 - 颜色基 XYZ 是 颜色基 RGB 的线性变换
 - 更多地应用于颜色科学的研究
 - 与人类的感知系统密切相关



CIE XYZ 色度图

- 可以将 CIE XYZ 颜色空间可视化如下色度图，其中： $x = X/(X+Y+Z)$ ； $y = Y/(X+Y+Z)$

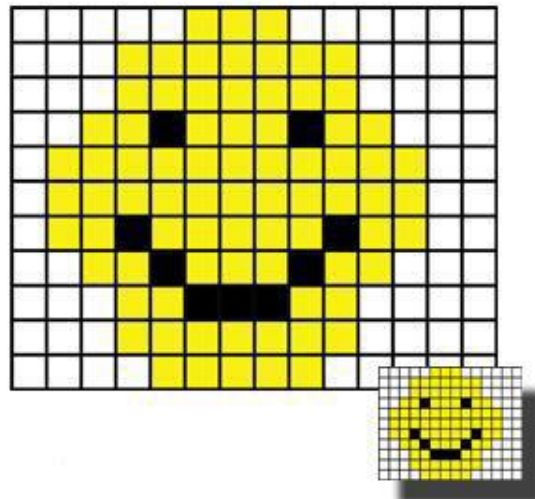




图像和像素

- 图像

- 图像可以看成是一个二维离散函数： $f(x, y)$
- 函数 f 的定义域是以矩阵形式排列着的格子，这些格子被称为 **像素 (pixel)**
- 函数 f 的取值为各个像素的色彩：对于彩色图像，可以是 RGB 或者 RGBA（其中A表示透明度）；对于灰度图像， f 为单值函数






People behind Pixel



SIGGRAPH2010

Siggraph 2010: The people behind the pixel, Part 1...
cinemawithoutborders.com

THE 38TH INTERNATIONAL CONFERENCE AND EXHIBITION ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES

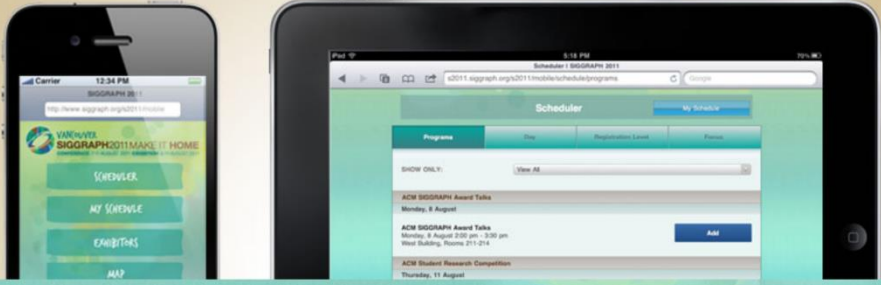
 **VANCOUVER SIGGRAPH2011 MAKE IT HOME**
CONFERENCE 7-11 AUGUST 2011 EXHIBITION 9-11 AUGUST 2011 VANCOUVER CONVENTION CENTRE

[Mobile Site](#)
[Create Account](#) | [Login](#)

[For Submitters](#) [For Volunteers](#) [For Attendees](#) [For the Media](#) [For Exhibitors & Advertisers](#)

Mobile Scheduler

AVAILABLE NOW: PROGRAMS, EXHIBITORS, AND MAPS OPTIMIZED FOR YOUR MOBILE DEVICE!



The image shows a smartphone and a tablet displaying the Mobile Scheduler app. The smartphone screen shows a menu with options: SCHEDULE, MY SCHEDULE, EXHIBITORS, and MAP. The tablet screen shows a detailed view of the scheduler with a table of events.

Program	Date	Registration Link	Phone
ACM SIGGRAPH Award Talks	Monday, 8 August		
ACM SIGGRAPH Award Talks	Monday, 8 August 8:00 pm - 9:00 pm	West Building, Rooms 211-214	Add
ACM Student Research Competition	Thursday, 11 August		



三角网格模型

- 图形学的基本目标是什么？
 - 从虚拟的三维场景及特定的相机参数中，生成二维图像
 - 三维场景又以怎样的数据结构来表示？
 - 简单球体、长方体可直接用其几何方程描述；对于复杂模型，则需要使用**参数曲线和曲面**或者更一般的**网格模型**来进行描述
 - 网格模型之中又以**三角网格**最为常用



三角网格的定义

- 三角网格是由一系列欧氏空间中的三维顶点以及连接这些顶点的若干三角面片组成，具体包括：
 - 顶点集合 $V = (v_1, v_2, \dots, v_n)$
 - 面片集合 $F = (f_1, f_2, \dots, f_m)$
 - 其中 F 中的每个面片 f_i 都是由 V 中的顶点构成的空间三角形：
$$f_1 = (v_{a1}, v_{b1}, v_{c1}), f_2 = (v_{a2}, v_{b2}, v_{c2}), \dots$$



几个例子

- 模型“牛”上显示了三角网格的结构
- 模型“龙”和“人头”是利用三角网格（smooth shading模式）进行绘制的结果
- Shading的概念
后面会进行介绍





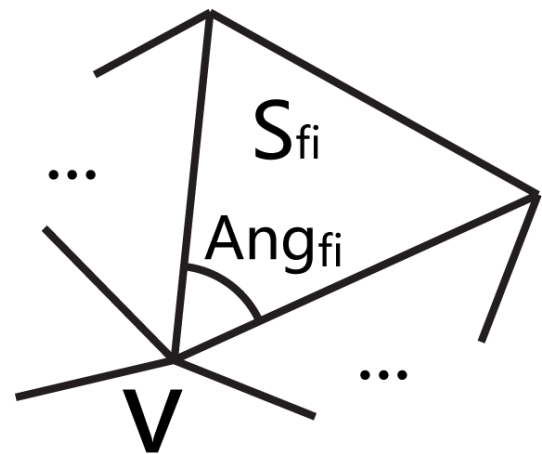
法向量

- 三角面片的法向量 (normal):
 - 三角面片的法向量是垂直于该三角面片所在平面的非零向量;
 - 对于每个三角面片而言, 其法向量都有两种可能的朝向;
 - 法向量的朝向决定了三角面片的正面与反面;
 - 对于连续可定向的三角网格整体而言, 相邻的三角面片需要具备一致的法向量朝向



法向量

- 三角网格顶点的法向量可以通过其周围所有三角面片的法向量不同方式的加权平均计算：
 - 假设 v 是 k 个三角面片 f_1, f_2, \dots, f_k 所共用的顶点；
 - 按算数平均计算：
 - $N_v = (N_{f1} + \dots + N_{fk}) / k$
 - 按面积加权平均计算：
 - $N_v = (S_{f1}N_{f1} + \dots + S_{fk}N_{fk}) / (S_{f1} + \dots + S_{fk})$
 - 按角度加权平均计算：
 - $N_v = (Ang_{f1}N_{f1} + \dots + Ang_{fk}N_{fk}) / (Ang_{f1} + \dots + Ang_{fk})$





三角网格的简单绘制

- 基于颜色的绘制
 - 给三角网格模型的每个顶点指定颜色属性
 - 模型表面每个点的颜色通过其所在三角面片顶点的颜色插值得到
- 基于光照的绘制
 - 给网格模型每个面片指定正反面的材质属性
 - 指定一个虚拟的光照环境
 - 如何计算材质在不同光照条件下呈现出来的颜色就涉及到了**光照模型**的问题



光照模型

- 光照模型 (lighting model 或 illumination model) 用于计算光的强度：
 - 局部光照明 (Local Lighting)
 - 关注物体直接受到光源影响所产生的光照效果
 - 全局光照明 (Global Lighting)
 - 除了关注直接光源照射的影响
 - 还关注光线在场景中其他物体表面发生折射、反射等作用后，间接照射到物体表面上产生的影响



光照模型的历史

- 1967年, Wylie 等人第一次在显示物体时加入了光照明效果, 认为光的强度与物体到光源的距离成反比关系
- 1970年, Bouknight 提出了第一个光反射模型:
 - Lambert 漫反射光 + 环境光
 - 发表于 [Communication of ACM](#)



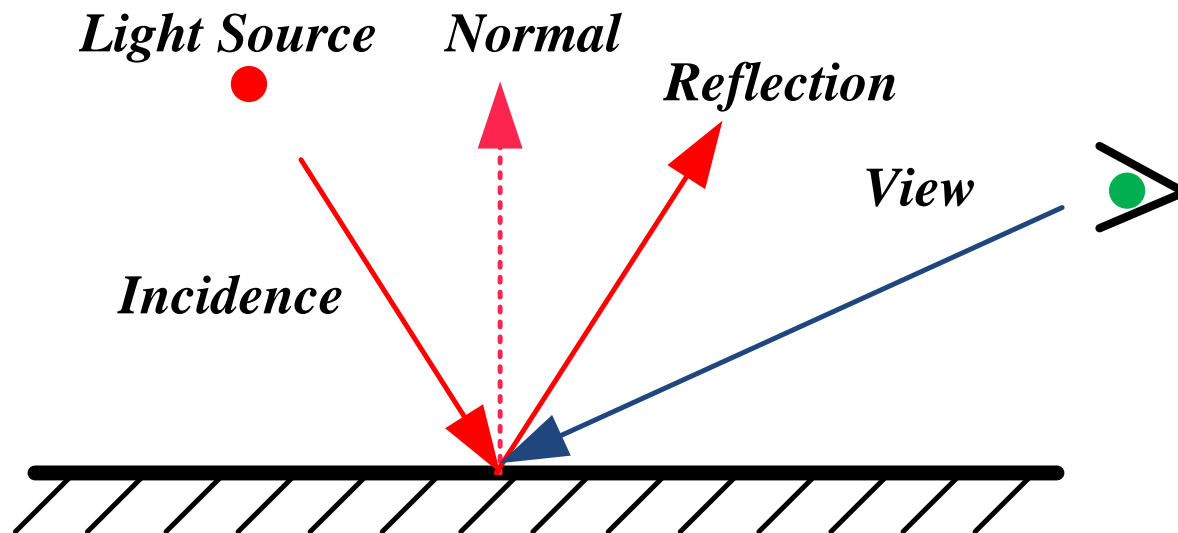
光照模型的历史

- 1971年, Gouraud 提出了漫反射模型加插值的思想:
 - Lambert 漫反射光 + Barycentric 插值
 - 发表于 [IEEE transactions on Computers](#)
- 1975年, Phong 提出了图形学中第一个有影响也是最有影响的光照模型: Phong 模型
 - 漫反射 (diffuse light) + 环境光 (ambient light) + 高光 (specular light)
 - 发表于 [Communication of ACM](#)



光的传播

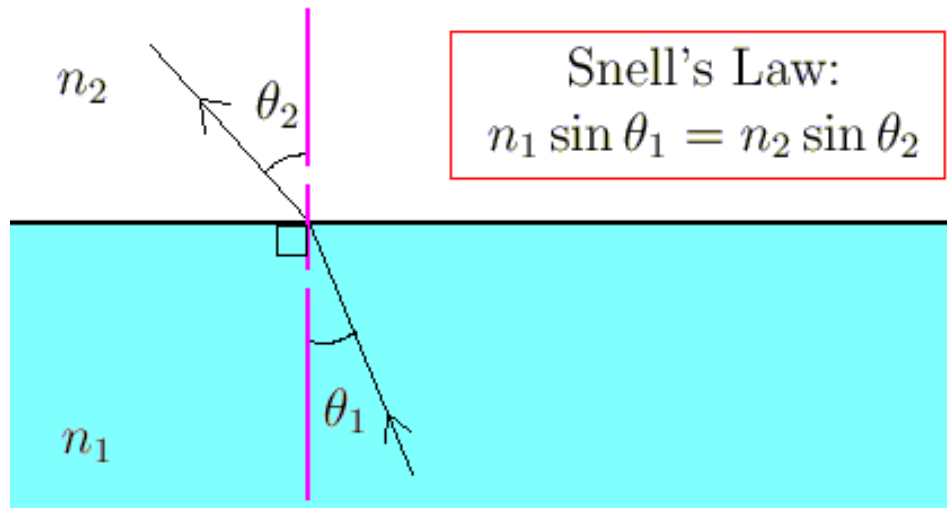
- 光的传播遵循反射定律：
 - 入射角等于反射角；
 - 入射光线、反射光线、以及反射面的法向量位于同一平面内





光的传播

- 折射定律 (也叫 Snell 定律):
 - 入射角和折射角的正弦值之比是一个仅仅取决于介质的常数;
 - 这个常数称为相对折射系数





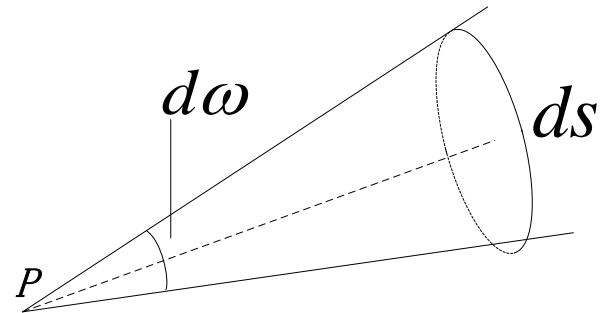
光的度量方法

- 立体角 (Solid Angle):

- 衡量物体相对于某一视点 P 的视角大小:

- 最大立体角为? 4π

$$d\omega = \frac{ds}{r^2}$$



- 光通量 (Luminous Flux):

- 光通量定义为: 单位时间内通过面元 dS 的光能量, 记为 dF

- 发光强度 (Radiance):

- 发光强度定义为: 单位立体角内的光通量, 记为 I



光线传播的能量方程

- 光的传播遵循能量守恒定律：

$$I_i = I_d + I_s + I_t + I_v$$

- I_i ：入射光的能量
- I_d ：漫反射 (diffuse reflection) 光的能量
- I_s ：镜面反射 (specular reflection) 光的能量
- I_t ：折射 (refraction) 光的能量
- I_v ：被介质和物体所吸收的能量



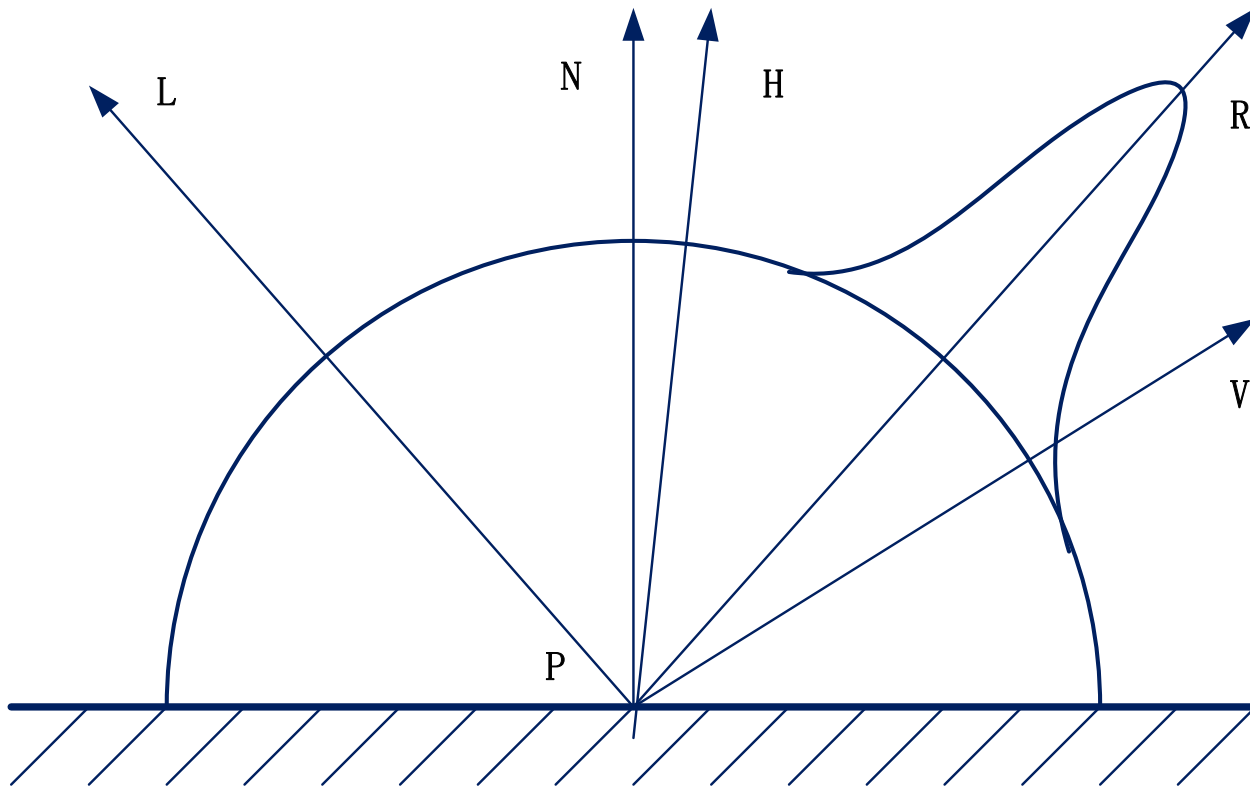
Phong 光照模型

- Phong 模型支持点光源和方向光源
- Phong 模型是局部光照模型，将局部光照明效果分解为三个部分：
 - 漫反射光效果
 - 镜面反射光效果
 - 环境光效果



Phong 光照模型

L 是入射光； R 是反射光； N 是物体表面的法向量； V 是视点方向； H 是 L 和 V 夹角的角平分线方向





Phong 光照模型

- 漫反射光效果

- 漫反射光的传播是各向同性的;
- 漫反射光的强度为:

$$I_d = I_i K_d * (L \cdot N)$$

- K_d 是漫反射系数
- K_d 具有三个分量 k_{dr}, k_{dg}, k_{db} 分别代表 R, G, B 三个通道的漫反射系数
- K_d 与模型自身的色彩紧密相关



Phong 光照模型

- 镜面反射光效果

- 对于光滑的平面，依据反射定律，反射光线往往集中在一个小的立体角内，这些反射光我们称之为镜面反射光；
- 镜面反射光的强度为：

$$I_s = I_i K_s * (R \cdot V)^n$$

- K_s 是镜面反射系数，与物体表面光滑程度相关；
- n 是反射指数； n 越大，则高光区域越集中



Phong 光照模型

- 环境光效果

- 环境光的强度为:

$$I_a = I_i K_a$$

- K_a 是物体对环境光的反射系数



Phong 光照模型

- 视角方向的发光强度为漫反射光分量、镜面反射光分量，以及环境光分量的发光强度之和：

$$I = I_i K_a + I_i K_s * (R \cdot V)^n + I_i K_d * (L \cdot N)$$

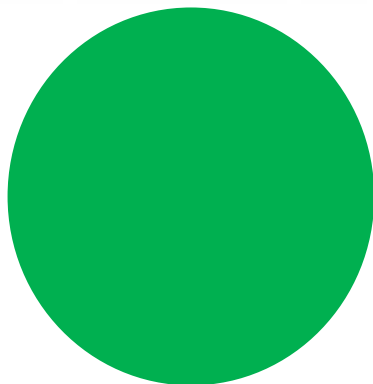


Phong 模型示例



漫反射光

+



环境光

+



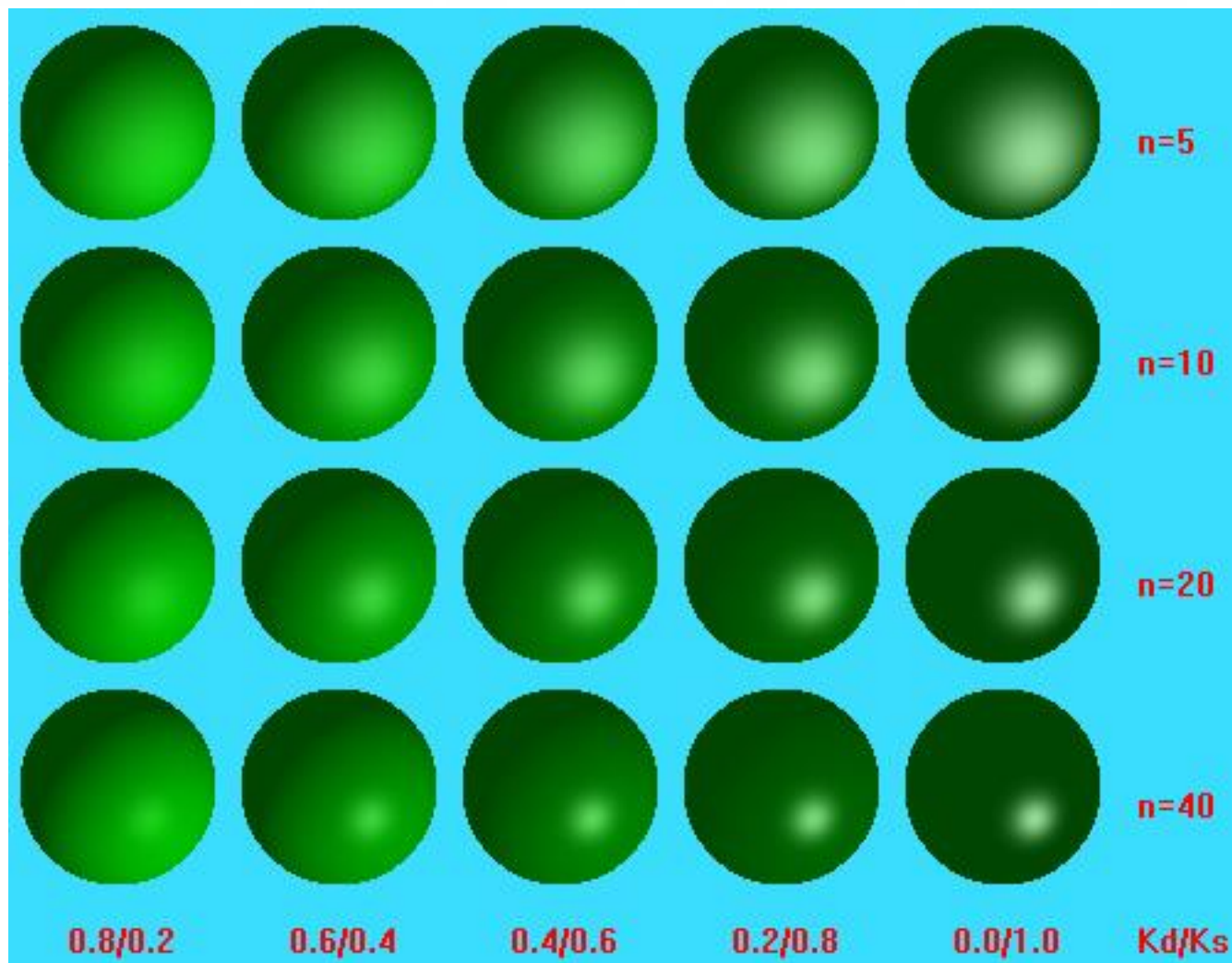
镜面反射光

=





Phong 模型示例





Wu Jianhua's Demo

Demo



明暗处理 (Shading)

- 考虑到物体表面的几何细节往往并不规则，为了减缓由模型离散化所导致的不光滑的色彩效果，通常的明暗处理除了使用光照模型外，还需要进行插值
- **Gouraud 明暗处理**和 **Phong 明暗处理**
 - Gouraud 明暗处理是对色彩进行插值
 - Phong 明暗处理则是对法向进行插值
- 为了概念的统一，我们将不做明暗处理的绘制模式称为flat shading



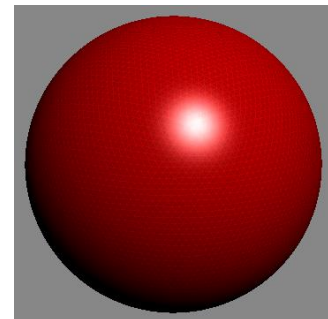
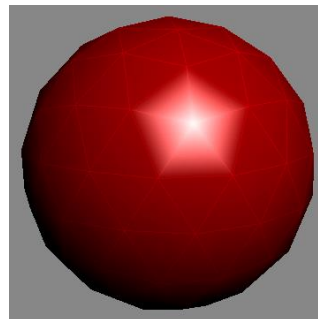
Gouraud 明暗处理

- 由 Gouraud 于 1971 年提出
 - 又被称为 Gouraud 插值
- 计算方法：
 - 首先计算所有模型顶点的颜色值；
 - 模型上的任何一点的颜色为：其所在三角面片三个顶点的颜色值重心插值 (Barycentric Interpolation) 的结果



Gouraud 明暗处理

- Gouraud明暗处理的问题：
 - 当三角面片在屏幕空间投影面积较大时，中心区域距离顶点较远，此时只通过顶点颜色的插值难以很好刻画内部区域应有的颜色
 - 这一问题在局部高光区域表现的尤其明显：
两个球体模型在相同的光照环境下以相同的速度旋转，当网格比较稀疏的时候，高光区域的过渡极其不平滑





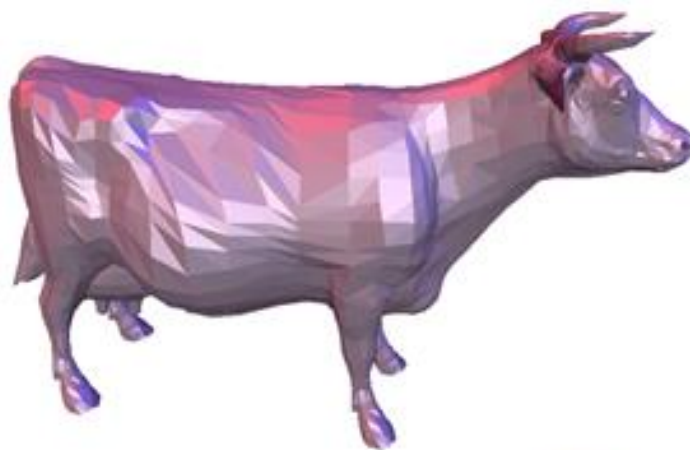
Phong 明暗处理

- 由 Bui Tuong Phong 于 1973 年在他的博士论文中提出，与 Gouraud 光照模型是不同的概念
 - 与 Gouraud 明暗处理不同，Phong 明暗处理不是对颜色插值，而是对点的法向量进行插值，得到连续的法向量场，再利用该法向量场逐点使用光照模型进行色彩的计算
 - 计算量大于 Gouraud 插值，但效果更加平滑
 - 现代图形硬件采用的明暗处理方法基本都是 Phong 明暗处理方法的变种



结果示例

Flat shading



Phong shading





视点变换和视点方向

- 图形学关注如何将由几何模型组成的三维场景绘制成高质量的彩色图像
- 变换在图形学中至关重要：通过变换，可以简洁高效地设置和编辑三维场景、光照位置和视点方向



为什么需要变换？

- 假设我们已经有了了一段可以绘制正方形 $[0,1]*[0,1]$ 的代码：
 - **`drawUnitSquare(0,0,1,1);`**
- 现在我们需要绘制一个平行于坐标轴，且左下和右上顶点坐标分别为 (lox, loy) 和 (hix, hiy) 的矩形，我们应该怎么做？



一种解法

- 一种方法是写一段新的代码：

```
drawRect(lox, loy, hix, hiy) {  
    glBegin(GL_QUADS);  
    glVertex2f(lox, loy);  
    glVertex2f(hix, loy);  
    glVertex2f(hix, hiy);  
    glVertex2f(lox, hiy);  
    glEnd();  
}
```

- 矩形简单可以这么做，可是对于复杂的例子（例如绘制一个茶壶，有大量的面片），怎么办？



利用变换的解法

```
drawRect(lox, loy, hix, hiy) {  
    glTranslate(lox, loy);  
    glScale(hix-lox, hiy-loy);  
    drawUnitSquare(0,0,1,1);  
}
```

- 这样的基于变换的解决方案在图形学中随时都需要用到；使用变换可以使代码更加快速、灵活、模块化



什么叫变换 (Transformation)?

- 变换是一个将空间中的点 x 映射成其他点 x' 的函数
- 广泛应用于: Morphing, Deformation, Viewing, Projection, Real-time shadows...

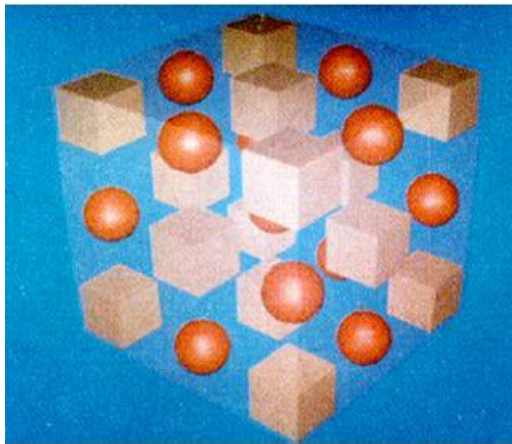


Fig 1. Undeformed Plastic

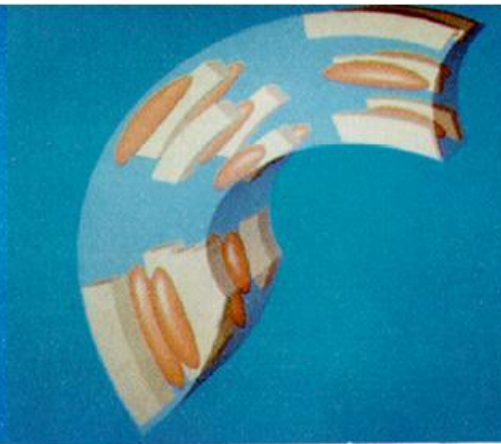
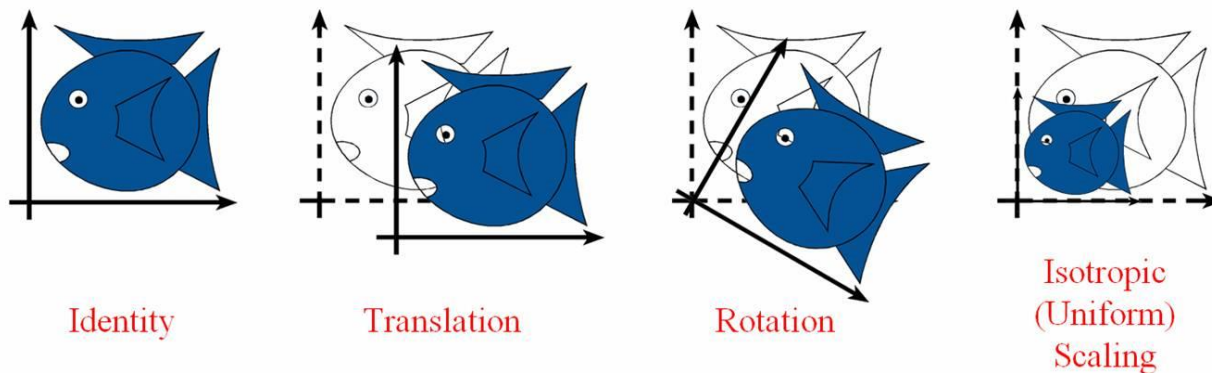


Fig 2. Deformed Plastic





简单变换



- 从左到右的变换分别是：
 - 不变 (**Identity**), 平移 (**Translation**), 旋转 (**Rotation**), 均衡缩放 (**Isotropic scaling**)
- 变换可以相互复合和嵌套：
 - 例如：先旋转，再缩放，最后再平移
- 简单变换都是可逆的



变换的分类

- 常见的变换有如下几类：
 - 刚体变换 (Rigid-body Transformation)
 - 相似变换 (Similarity Transformation)
 - 线性变换 (Linear Transformation)
 - 仿射变换 (Affine Transformation)
 - 投影变换 (Projective Transformation)

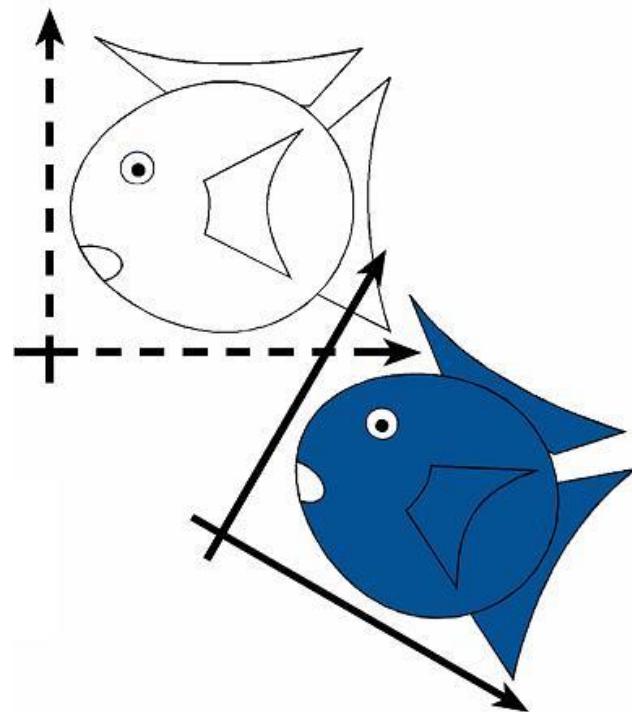


刚体变换

- 保持度量 (长度、角度、大小)。

- 刚体变换包括:

- 不变
- 平移
- 旋转
- 以及它们的复合

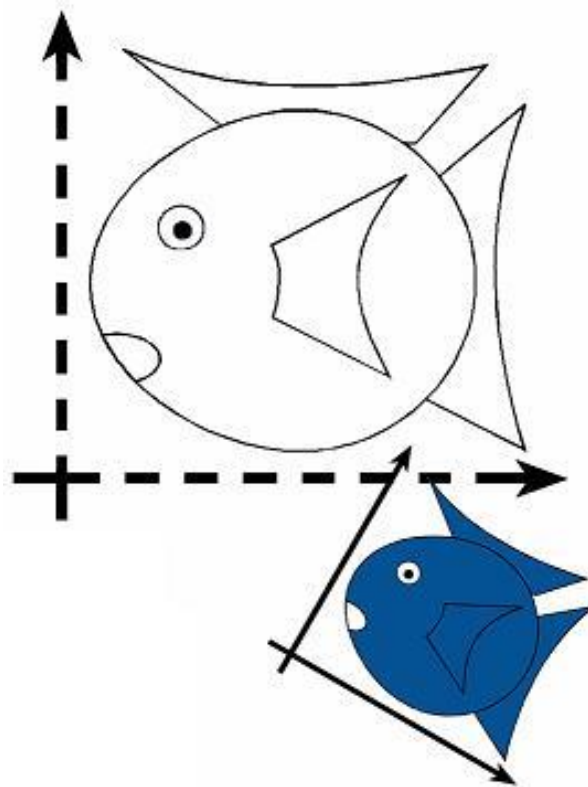


- 为何对称不是刚体变换?



相似变换

- 保持角度
- 相似变换包括：
 - 不变
 - 平移
 - 旋转
 - 均衡缩放
 - 以及它们的复合





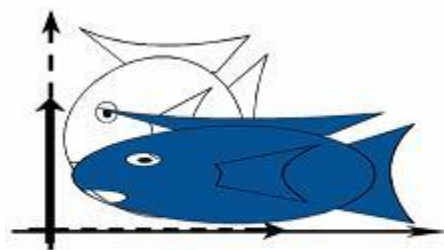
线性变换

- 线性变换满足如下方程:

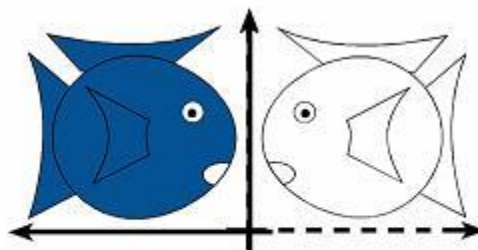
$$L(p+q) = L(p) + L(q) \quad aL(p) = L(ap)$$

- 线性变换包括:

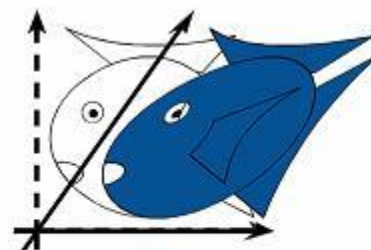
- 不变、旋转、缩放 (不一定要均衡缩放)
- 对称 (Reflection), 错切 (Shear)



Scaling



Reflection

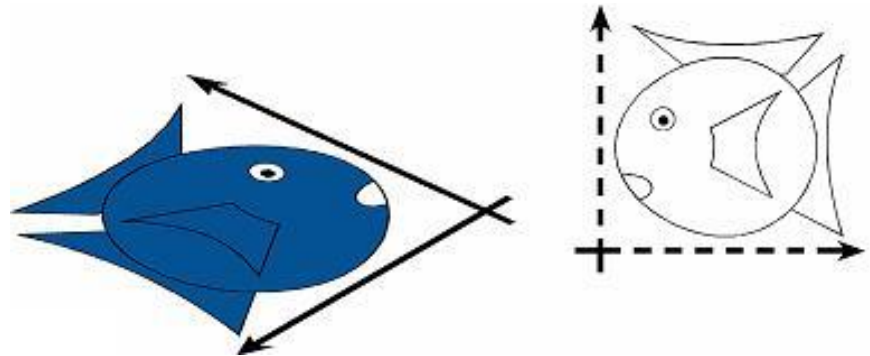


Shear

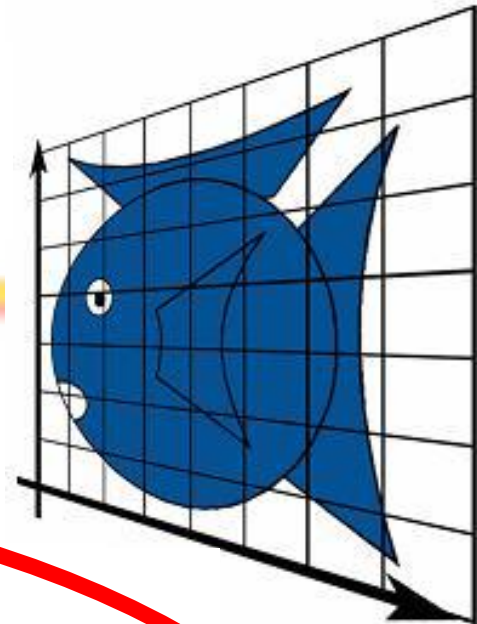


仿射变换

- 保持直线以及直线与直线平行关系
 - 两条在仿射变换之前平行的直线，在仿射变换之后依旧平行
- 仿射变换包括：
 - 线性变换
 - 相似变换
 - 以及它们的复合



投影变换



- 保持直线

投影变换

仿射变换

相似变换

线性变换

刚体变换

一般缩放

平移

单位变换
旋转

均衡缩放

对称
错切



变换的表示

- 考虑简单的二维仿射变换:

$$x' = ax + by + c \quad (1)$$

$$y' = dx + ey + f \quad (2)$$

- 可以将 (1), (2) 统一地写成矩阵形式:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix}$$

$$p' = M p + t$$



变换的表示

- 变换的表示涉及到了两个变量：**M** 和 **t**

$$p' = M p + t$$

- 然而，使用齐次坐标 (Homogenous Coordinates) 来表示 p 和 p' , 则上述公式成为:

$$p' = M p$$

- 什么是齐次坐标?



齐次坐标 (Homogeneous Coordinates)

- 齐次坐标的本质是使用 $d + 1$ 维数组来表示 d 维空间中的点和向量。

不使用齐次坐标表示

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix}$$

$$p' = M p + t$$

使用齐次坐标表示

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$p' = M p$$



齐次坐标

- 由于引入了新的维度，在三维空间下，我们使用 4×4 的作用矩阵，同时使用 (x, y, z, w) 来表示一个点或向量：

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$$

$$p' = \textcolor{red}{M} p$$



齐次坐标

- 在大部分情况下 $w = 1$ ，可以忽略。

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- 当齐次坐标被仿射矩阵作用时， w 不会改变；而如果被投影矩阵作用， w 会改变



齐次坐标的几何意义

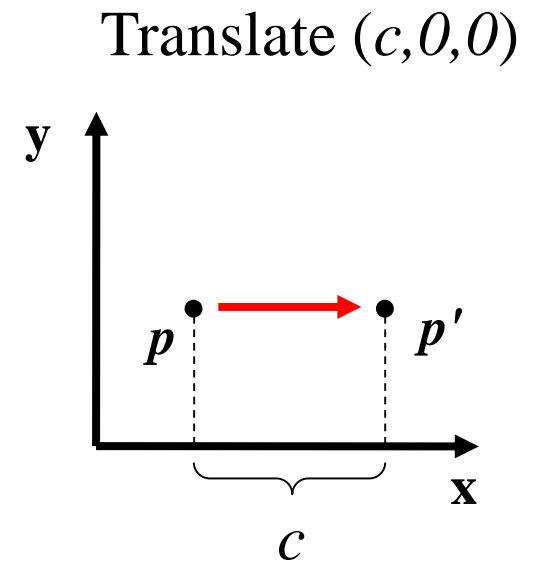
- 当 w 非零时，通过将所有四个坐标同时除以 w 以归一化
 - 例如： $(2x, 2y, 2z, 2)$ 等价于 $(x, y, z, 1)$
- 当 w 为零时，齐次坐标 $(x, y, z, 0)$ 可以理解为沿 (x, y, z) 方向无穷远的点



平移变换 (t_x, t_y, t_z)

- 可以看到，在齐次坐标的框架下，平移变换可以表示成为简单的矩阵乘法：

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

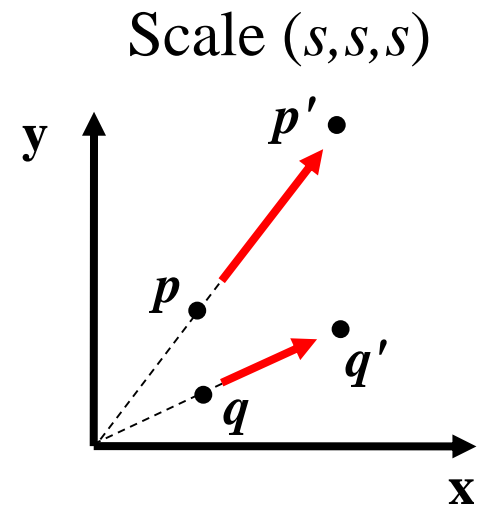




缩放变换 (s_x, s_y, s_z)

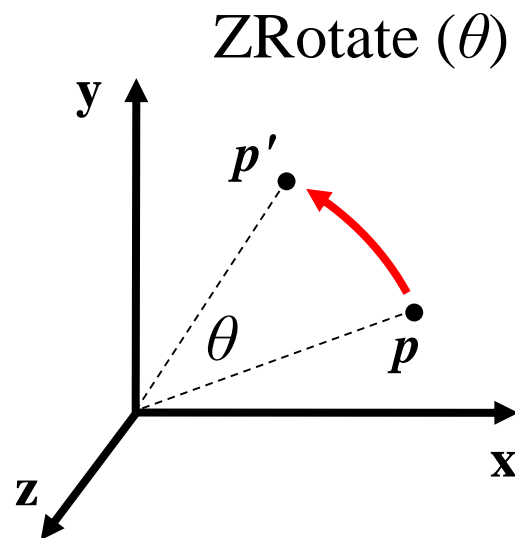
- 以原点为缩放中心的缩放变换可以表示成为:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



旋转变换

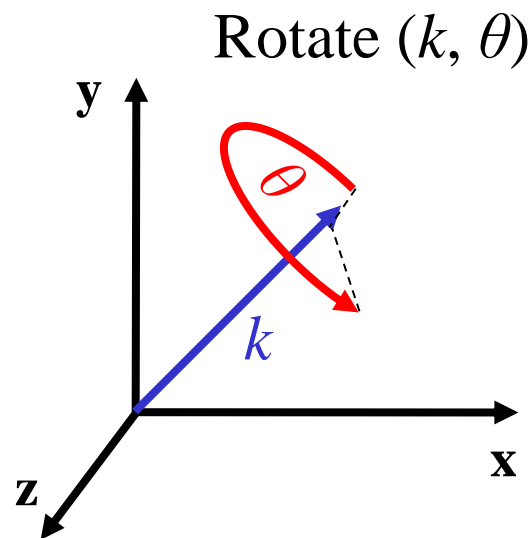
- 三维空间中物体的旋转具有三个独立的自由度
- 围绕 z 轴的旋转可以表示为:



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

旋转变换

- 围绕单位向量 (k_x, k_y, k_z) 的旋转变换可以写成如下的 Rodrigues 公式:



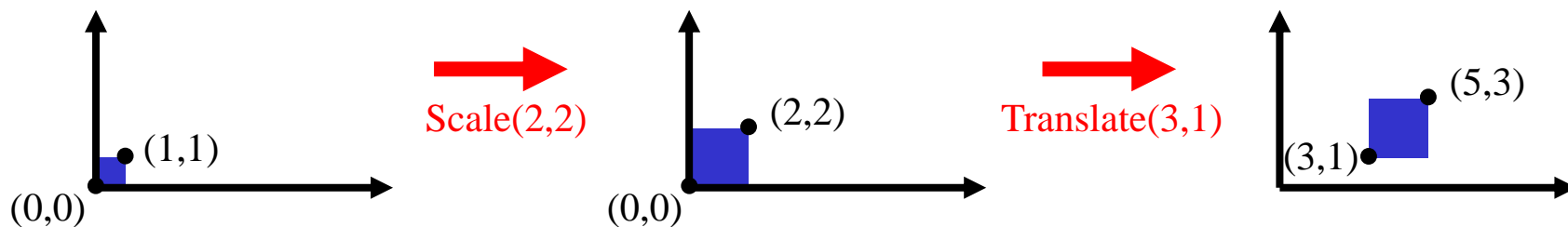
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} k_x k_x (1-c) + c & k_y k_x (1-c) - k_z s & k_x k_z (1-c) + k_y s & 0 \\ k_y k_x (1-c) + k_z s & k_y k_y (1-c) + c & k_y k_z (1-c) - k_x s & 0 \\ k_z k_x (1-c) - k_y s & k_y k_z (1-c) + k_x s & k_z k_z (1-c) + c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where $c = \cos\theta$ & $s = \sin\theta$



变换的复合 (combination)

一个缩放和平移进行复合的例子：



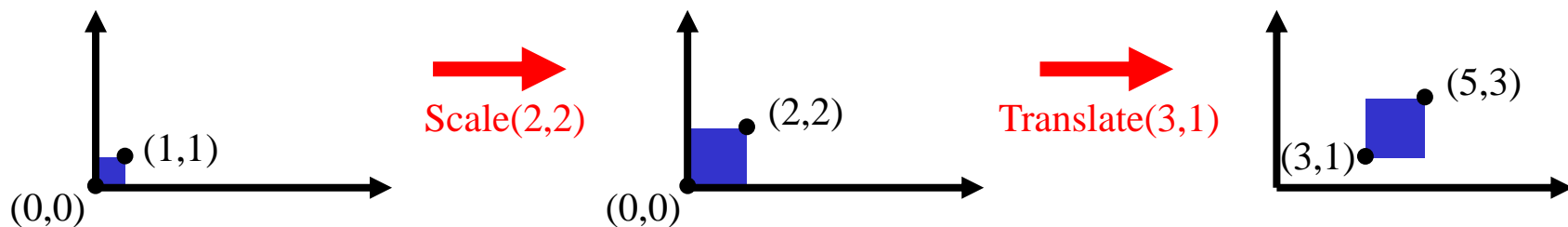
复合等价于矩阵乘法: $p' = T (S p) = TS p$

$$TS = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

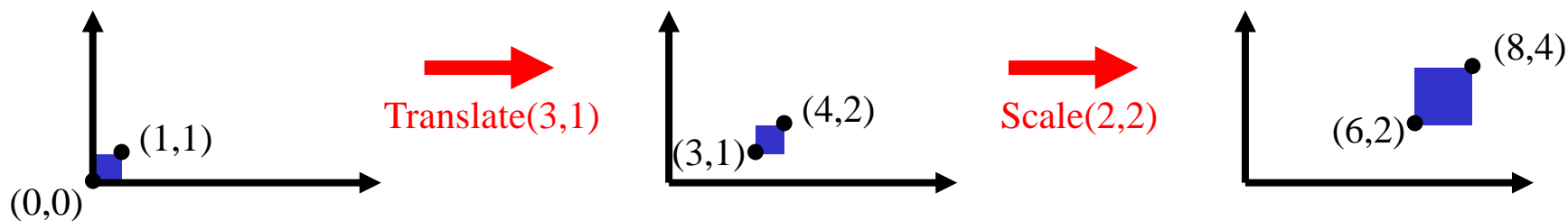


变换的复合不满足交换律

先缩放再平移: $p' = T(S p) = TS p$



先平移再缩放: $p' = S(T p) = ST p$





变换的复合不满足交换律

变换的复合不满足交换律，这是因为矩阵的乘法不满足交换律： **$TS \neq ST$**

先缩放再平移： $p' = T(S p) = TS p$

$$TS = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

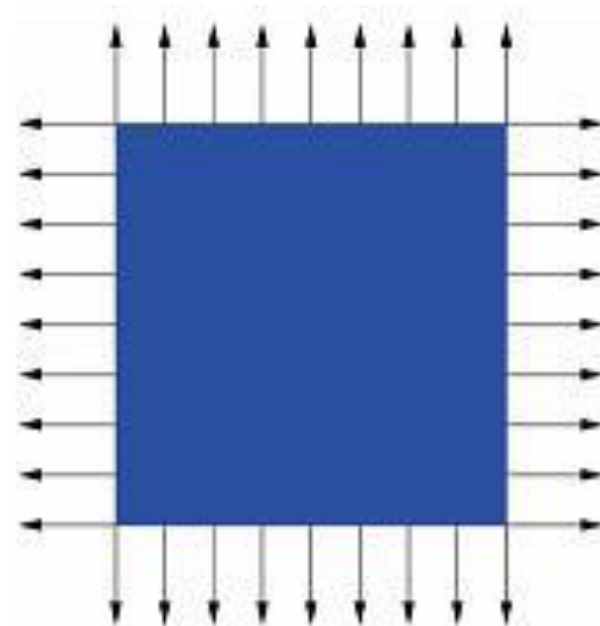
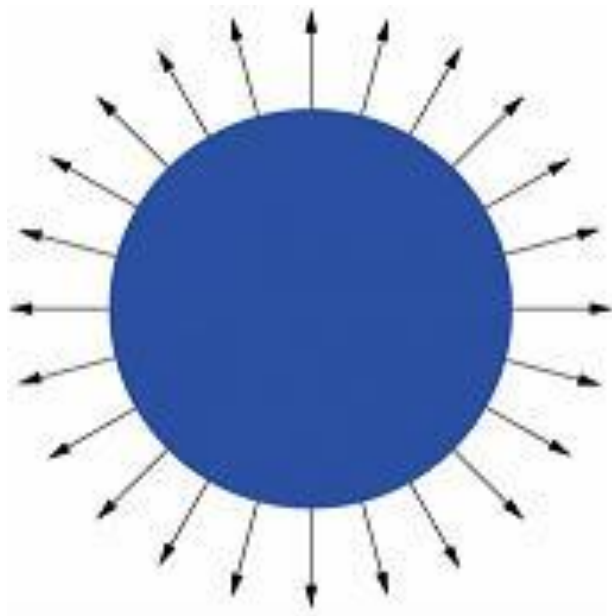
先平移再缩放： $p' = S(T p) = ST p$

$$ST = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 6 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$



法向量变换

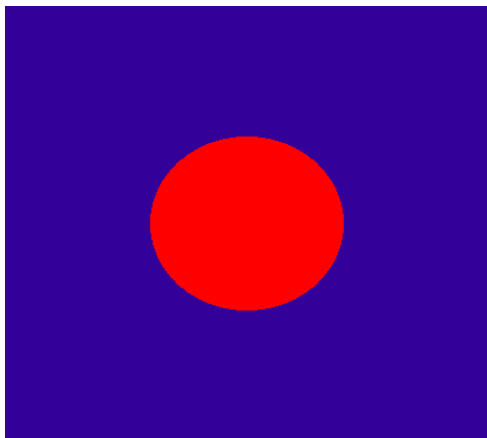
- 曲面的 (单位) 法向量是曲面与曲面正交的 (单位) 向量，它们是曲面最为重要的几何性质之一



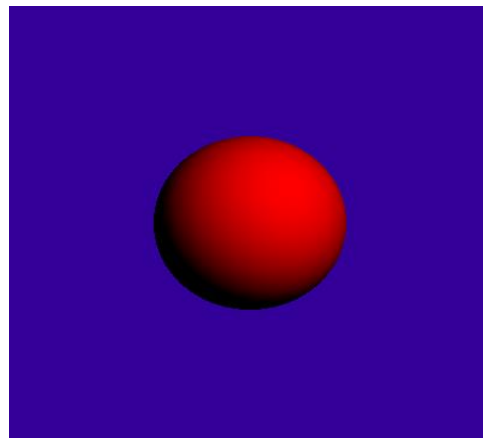


为什么法向量如此重要？

- 法向量是进行光照处理的必要输入，所有的光照模型都涉及到物体的法向量
- 只有知道了物体的法向量信息，才能绘制出具有三维立体感的图像



只有物体颜色

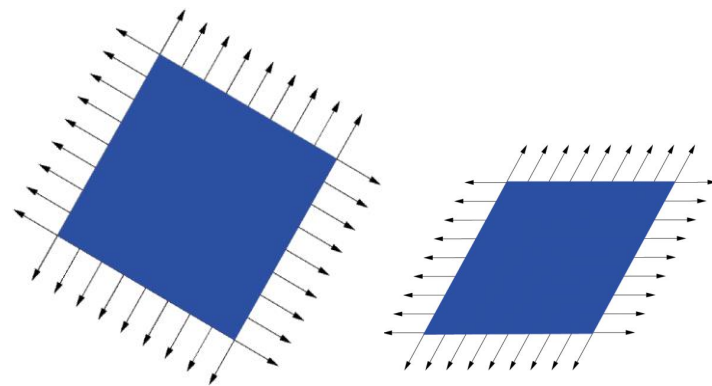
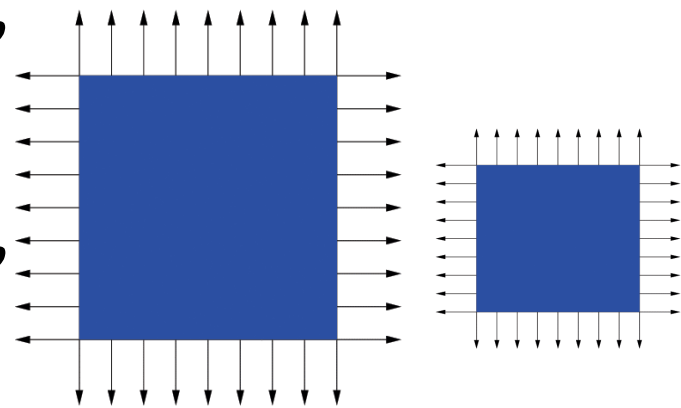


漫反射着色



像变换物体一样变换法向？

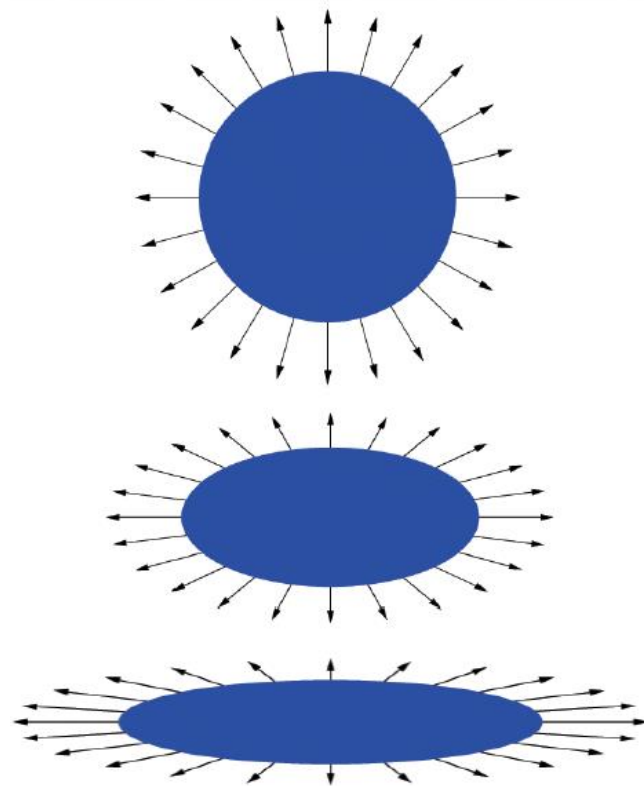
- 在我们对物体进行变换时，法向量可以类似变换吗？
- 我们发现，对于相似变换，法向量可以和物体使用**同样的变换方程**
- 然而对于右图中带有错切的仿射变换，**同样的方程则不适用**





像变换物体一样变换法向？

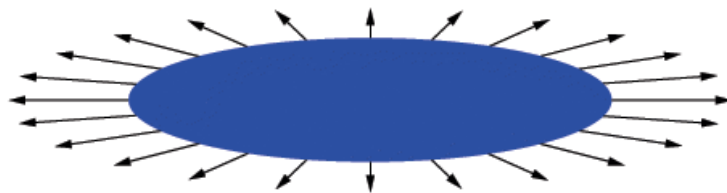
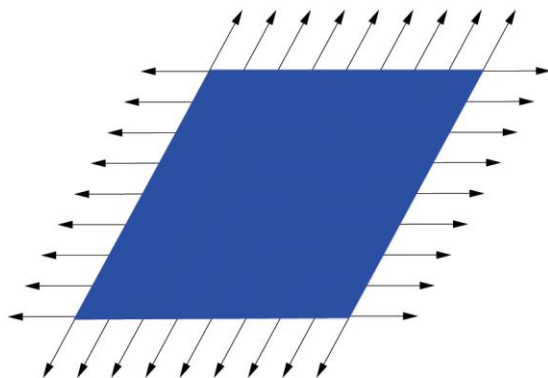
- 另一个例子（缩放）：
 - 对右图的球作非均衡的缩放，使用与物体同样的变换方程用于变换法向量会出现错误结果：



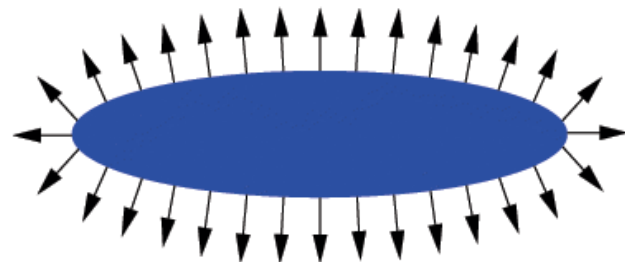
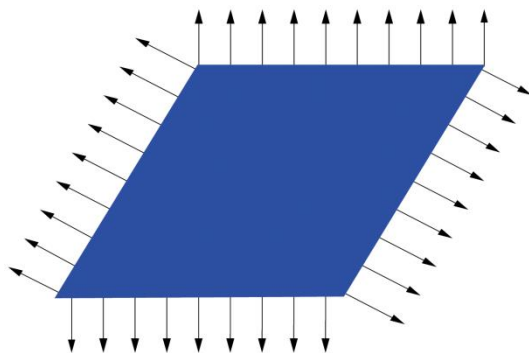


错切和缩放应有的法向变换

错误的法向变换



正确的法向变换

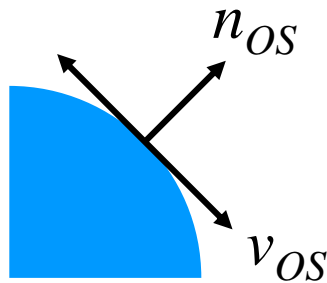


如何对法向量进行变换才能确保正确？

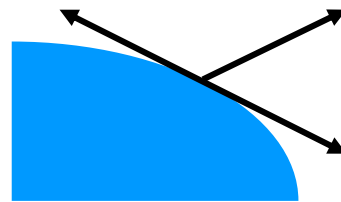


如何正确变换法向量?

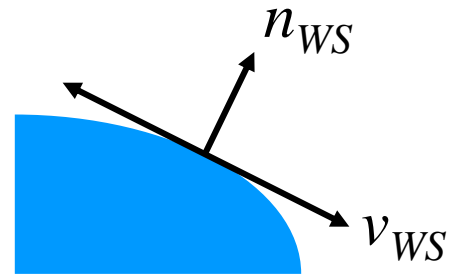
- 变换 **切平面 (*tangent plane*)**，再通过切平面计算法向量，而不是直接计算



原始表面



错误的变换



正确的变换

- 切平面上的任一向量 v_{OS} 变换后成为 v_{WS} :

$$v_{WS} = \mathbf{M} v_{OS}$$



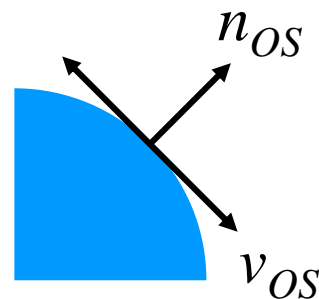
由切向量计算法向量

\mathbf{v}_{OS} 和 \mathbf{n}_{OS} 垂直: $\mathbf{n}_{OS}^T \mathbf{v}_{OS} = 0$

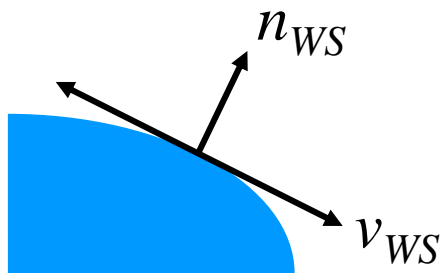
$$\mathbf{n}_{OS}^T (\mathbf{M}^{-1} \mathbf{M}) \mathbf{v}_{OS} = 0$$

$$(\mathbf{n}_{OS}^T \mathbf{M}^{-1}) (\mathbf{M} \mathbf{v}_{OS}) = 0$$

$$(\mathbf{n}_{OS}^T \mathbf{M}^{-1}) \mathbf{v}_{WS} = 0$$



\mathbf{v}_{WS} 和 \mathbf{n}_{WS} 垂直: $\mathbf{n}_{WS}^T = \mathbf{n}_{OS}^T (\mathbf{M}^{-1})$



$$\mathbf{n}_{WS} = (\mathbf{M}^{-1})^T \mathbf{n}_{OS}$$

法向量的变换矩阵是原变换矩阵的逆的转置



视点和投影模式

- 我们的眼睛能将三维场景感知为二维图像，大脑则会将二维图像再重构回三维
- 在计算机图形学中，我们利用投影模拟眼睛的功能和效果
- 投影的两个重要概念：
 - 视点变换：与相机（眼睛）位置及朝向相关
 - 投影模式：是将 3D 变换成为 2D 的变换模式，常用的包括正交投影和透视投影

正交投影 (Orthographic Projection)

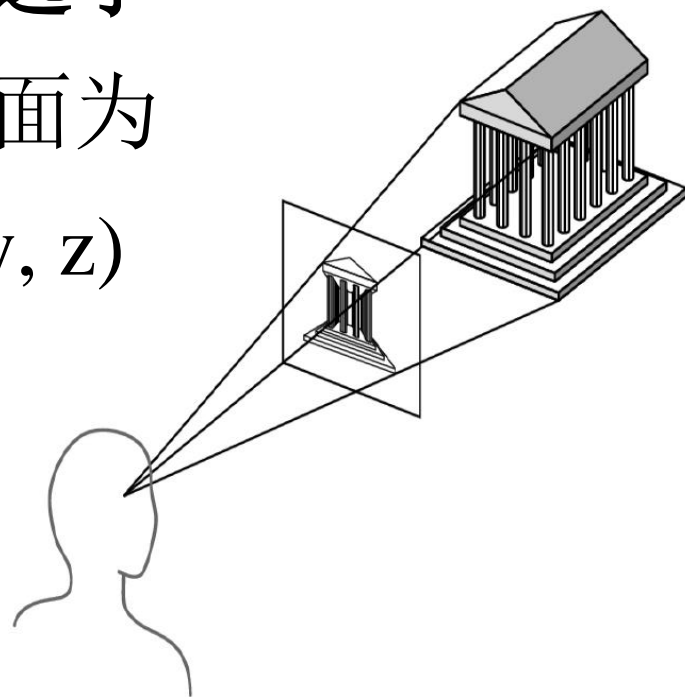


- 正交投影是视点在无穷远处的投影模式
- 缺乏立体透视效果
- 当 xy 平面是投影平面时，正交投影把 (x, y, z) 映射成为 $(x, y, 0)$



透视投影 (Perspective Projection)

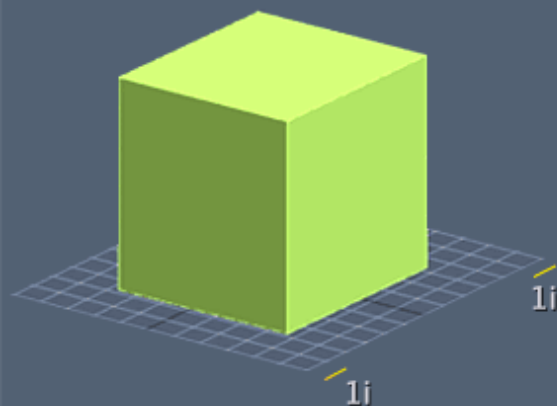
- 透视投影是视点在有限距离处的投影模式
- 具有立体透视效果 – 近大远小
- 当视点位于原点；投影平面为 $z = d$ 时，透视投影把 (x, y, z) 映射成为 $((d/z)x, (d/z)y, d)$





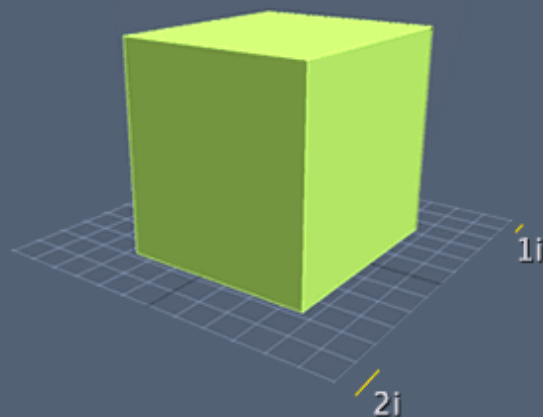
两种投影方式的对比

Orthographic



- Everything seems equal
- No Vanish-Point
- Parallel lines never touch

Perspective



- Closest things seems bigger
- Has Vanish-Point
- Parallel lines touch at infinity



透视投影矩阵

- 使用齐次坐标描述的透视投影变换可以写成矩阵形式：

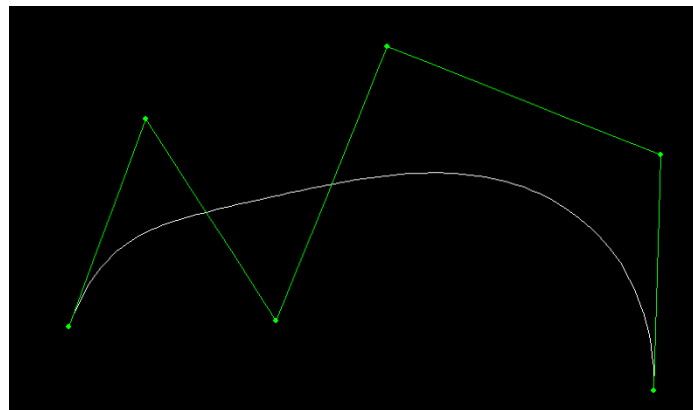
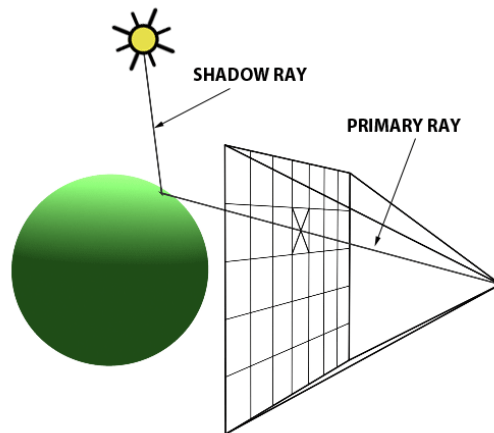
$$- T(x, y, z) = ((d/z)x, (d/z)y, d)$$

$$\begin{bmatrix} dx \\ dy \\ dz \\ z \end{bmatrix} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



课程考核

- 四个小作业
 - 光栅图形学
 - 光线投射
 - 样条曲线曲面
 - **Conditional GAN**
- 一个大作业，二选一
 - 真实感绘制（渲染）
 - 计图算法挑战赛

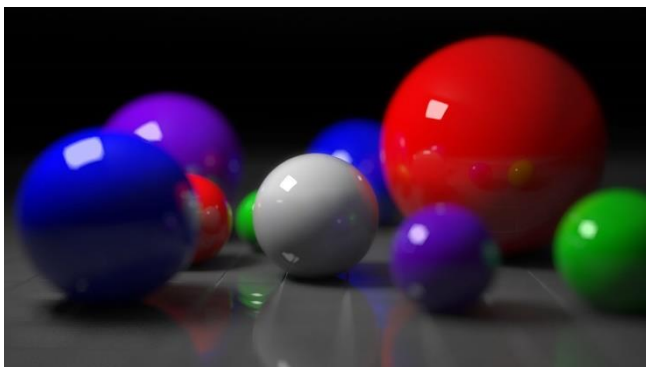




大作业-真实感绘制（渲染）

- 大作业

- 包含Bezier曲线的三维造型与光线跟踪绘制
- 报告+源码提交：





大作业-图形学算法挑战赛

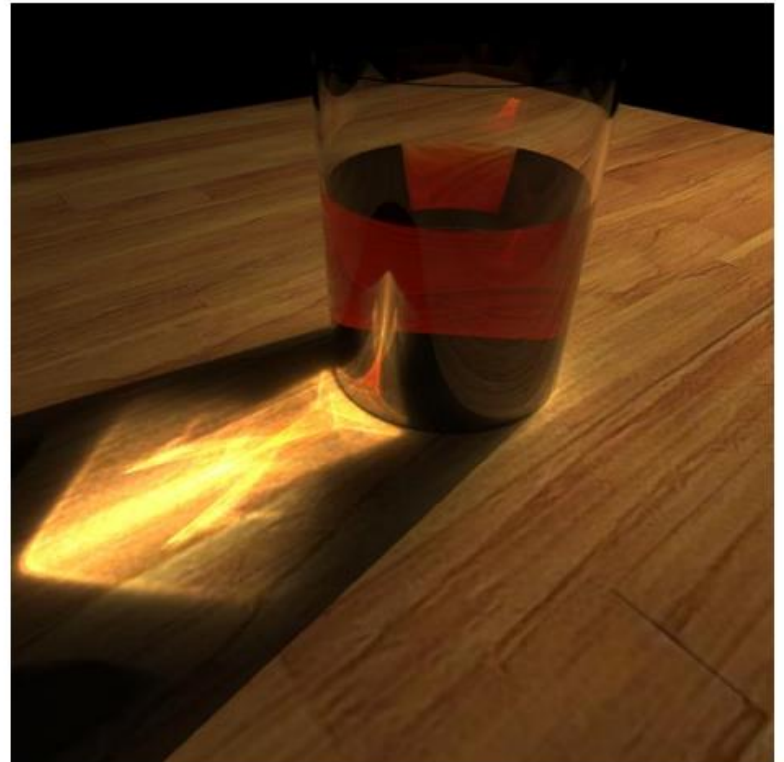
- 三人一组，在两个赛道中选择一个参加，详细要求后续会发布。
- 本赛道的目标是从语义分割图与风格参考图生成优美的风景图像。图像生成是GAN领域非常重要的方向，计图提供了一个包含27个GAN模型的库供赛题使用。
- 本赛道提供了十万余张高分辨率的风景图像供选手使用。
- 评价指标包括美学评价、专家打分、投票。





往年优秀作业

- 计20，项思陶 (USC)





往年优秀作业

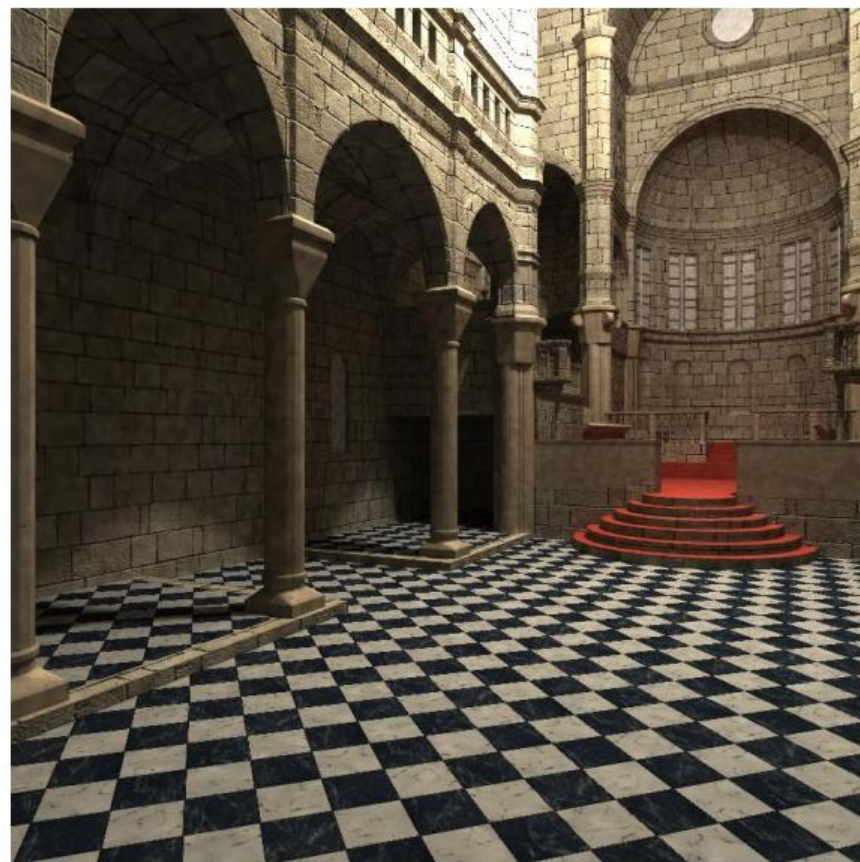
- 计30, 胡渊鸣 (MIT) 与冰雪奇缘





往年优秀作业

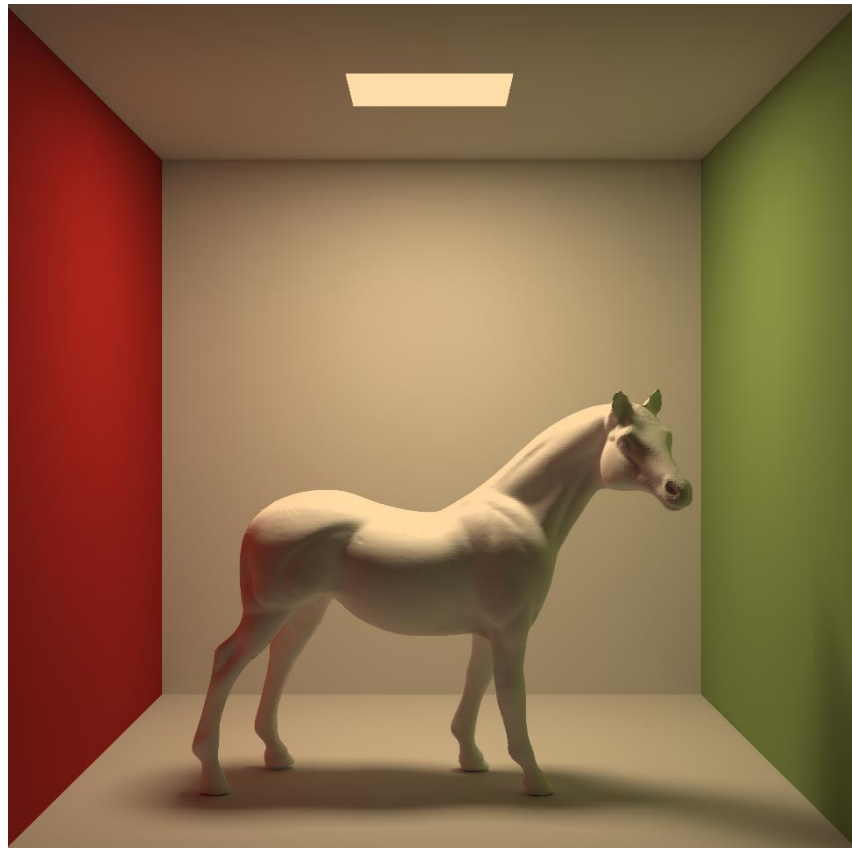
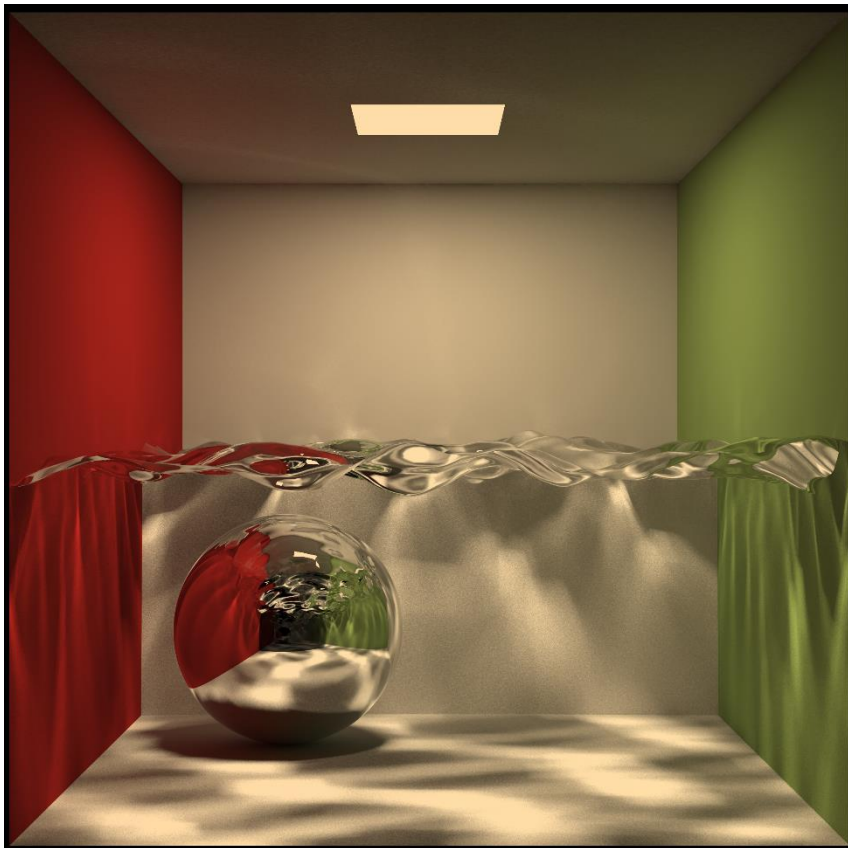
- 计33，古裔正





往年优秀作业

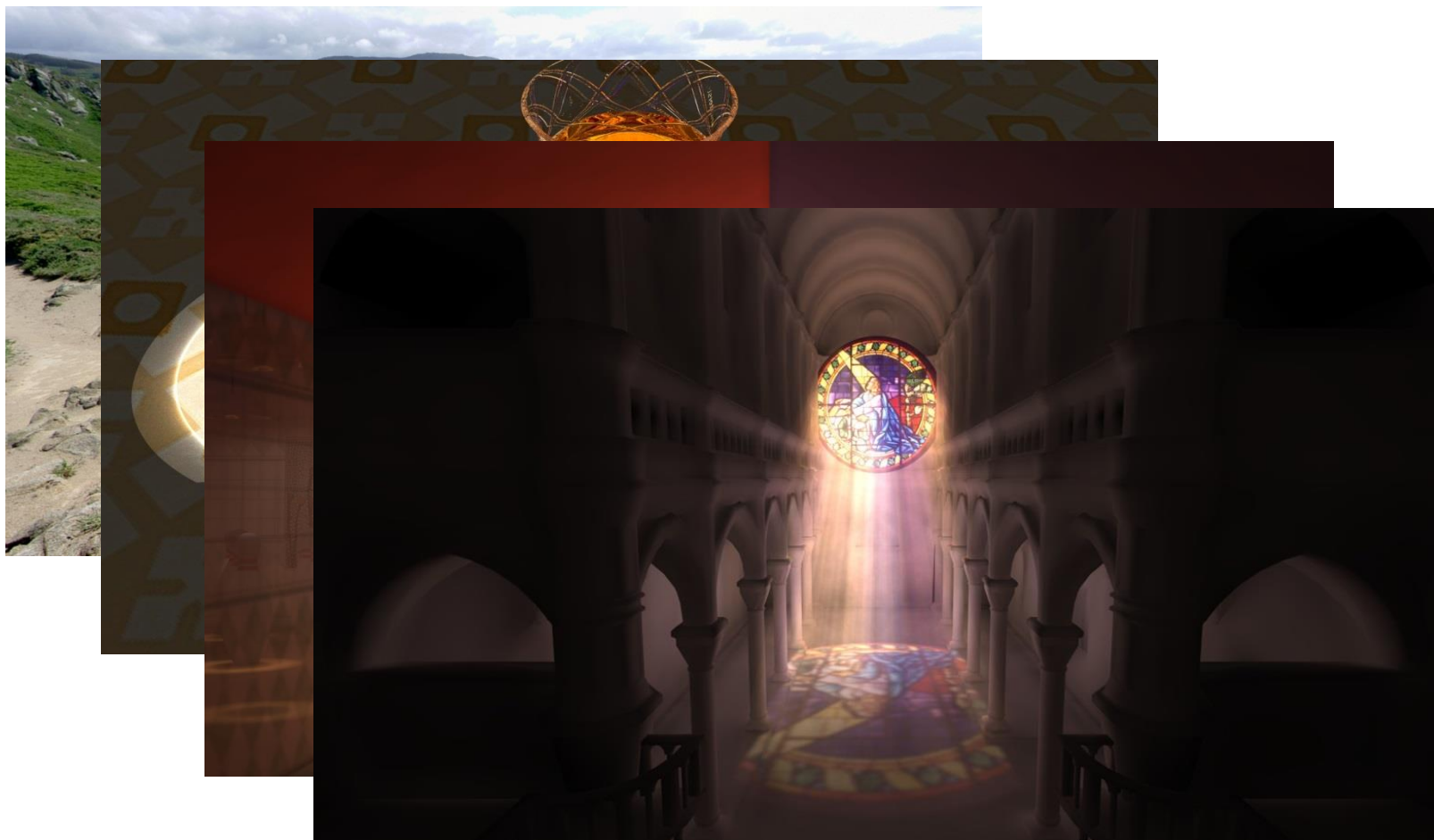
- 计30，王若松





2017年CS优秀作业

- 100分 （秦岳）





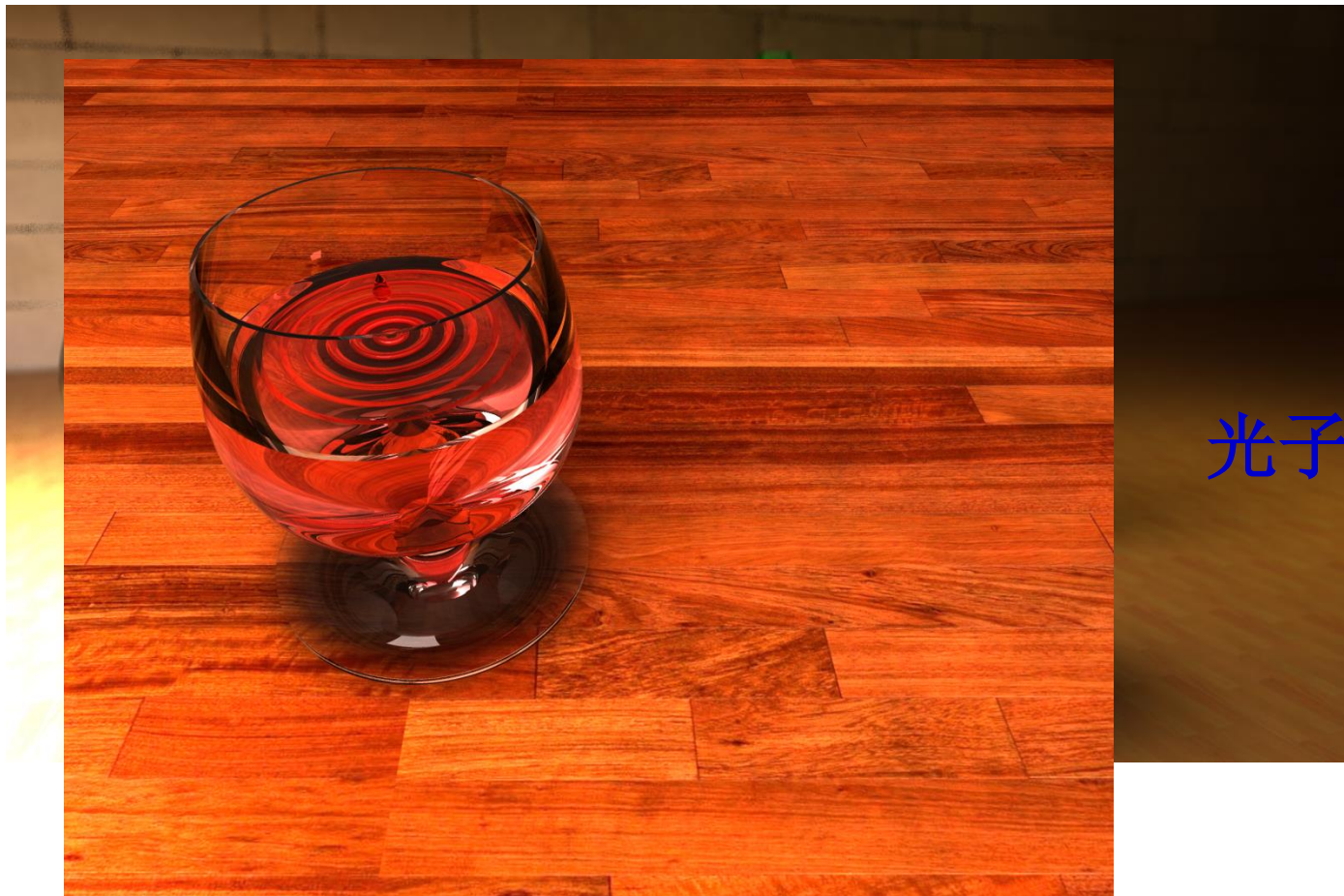
2017年CS优秀作业

- 97.5分





2017年Yao优秀作业



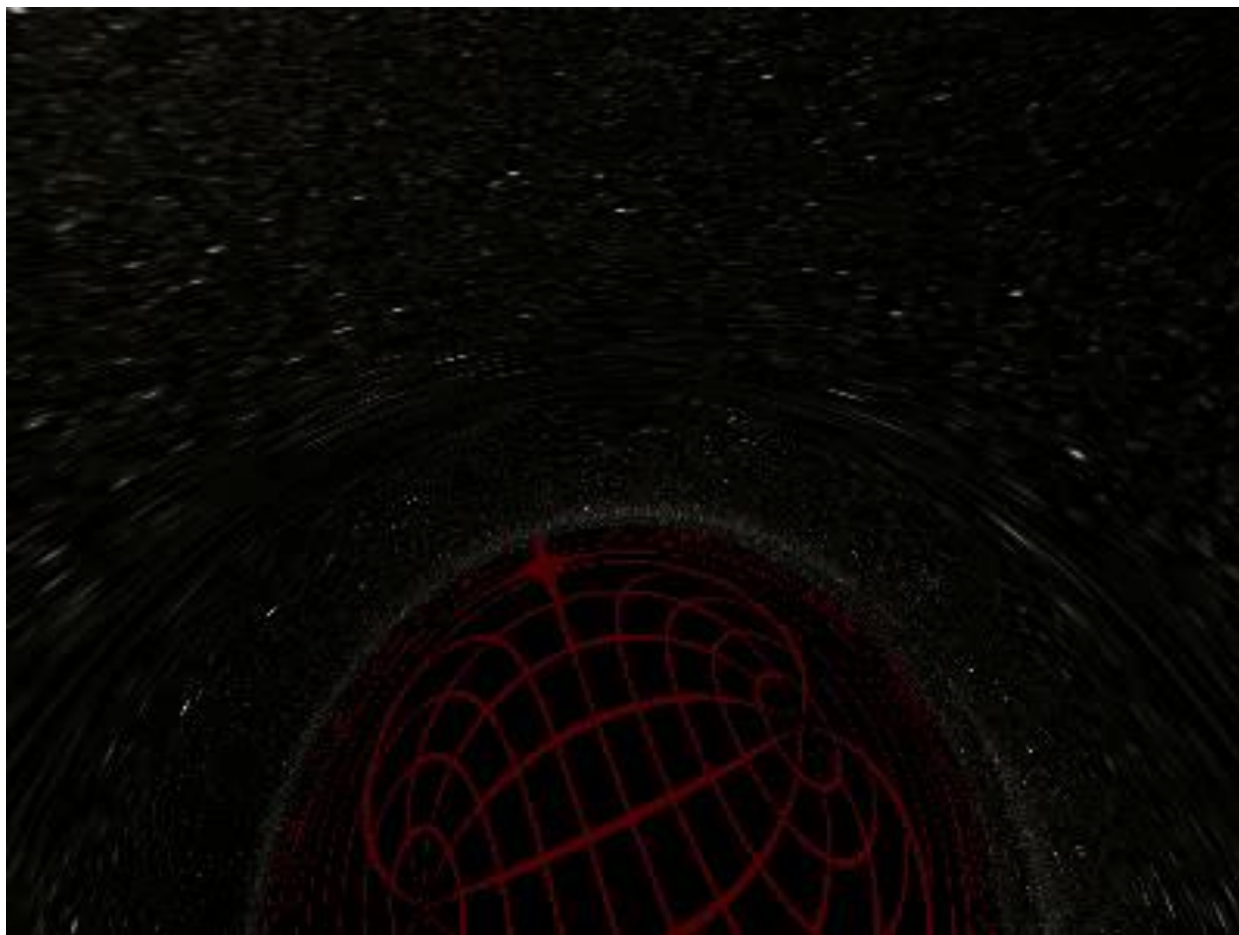
带景深

光子映射



2021年Yao优秀作业

- 刘方轩



弯曲时空下
的光线追踪
-黑洞



今日人物: Pat Hanrahan

- Pat Hanrahan, Stanford 教授
 - Google scholar 的H因子93
 - 论文300+篇, 引用51348次
 - Light field 5396, Volume rendering 2612
 - University of Wisconsin–Madison
PHD(1985), 纽约工学院、Pixar
 - 1989年, 受聘Princeton, 1995年受聘Stanford
 - 1993获SIGGRAPH成就奖, 2003年获Coons奖
 - 2004年获奥斯卡奖, 2020年图灵奖。
 - 美国工程院, 艺术与科学院 院士, ACM Fellow





谢谢！