图像图形学

相机与投影几何

计算机视觉

- 图像图形学（Graphics）
  - 我们怎么获得视觉信息
  - 无监督学习，更关注数学建模
- 感知、识别、推理
  - 我们能从视觉信息中获得什么
  - 监督学习，更关注数据驱动
- 生成
  - 我们怎么创造新的视觉信息
  - 自监督学习，模型与数据兼备

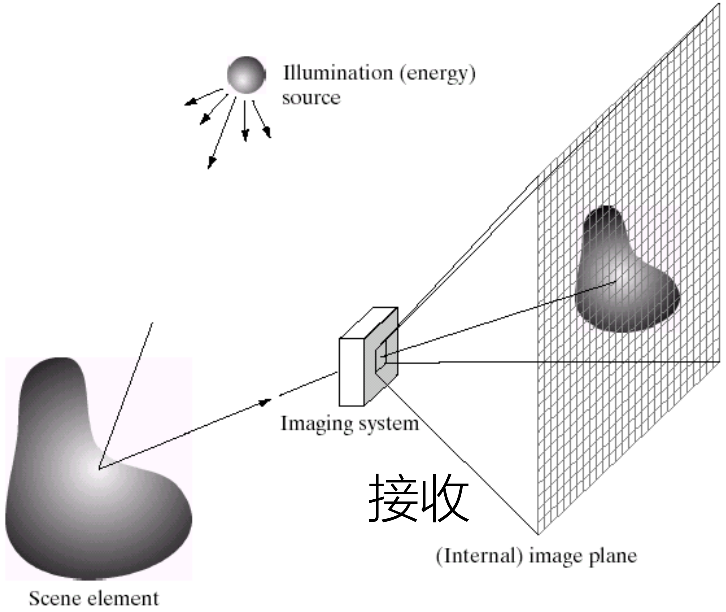
图像图形学

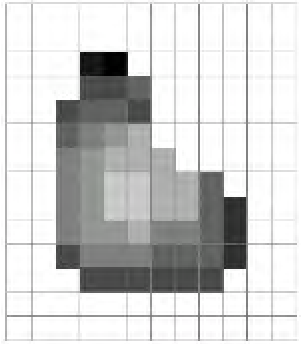
- 我们怎么获得视觉信息
- **无监督学习**，更关注数学建模
- 照相机模型与成像原理
- 图像处理基础
- 二维几何
- 三维几何
- **次要目的**：提升学生的拍照、修图水平
- **首要目的**：为感知理解提供基础

什么是图像?

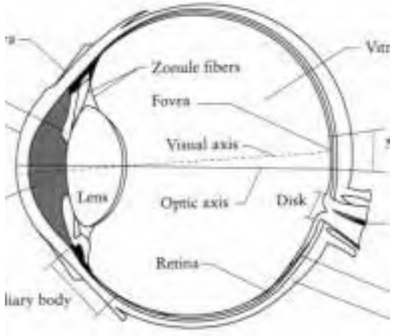
光照

反射





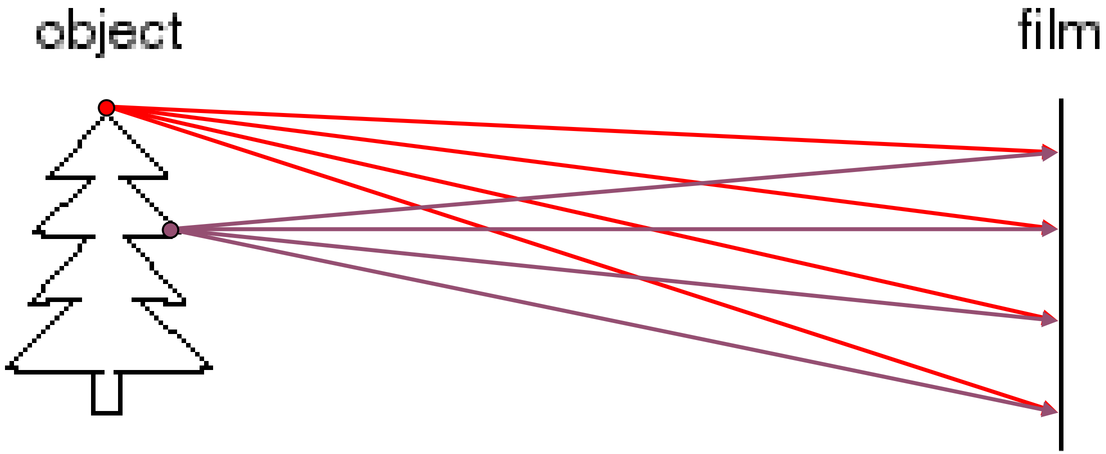
数字照相机



眼睛

Source: A. Efros

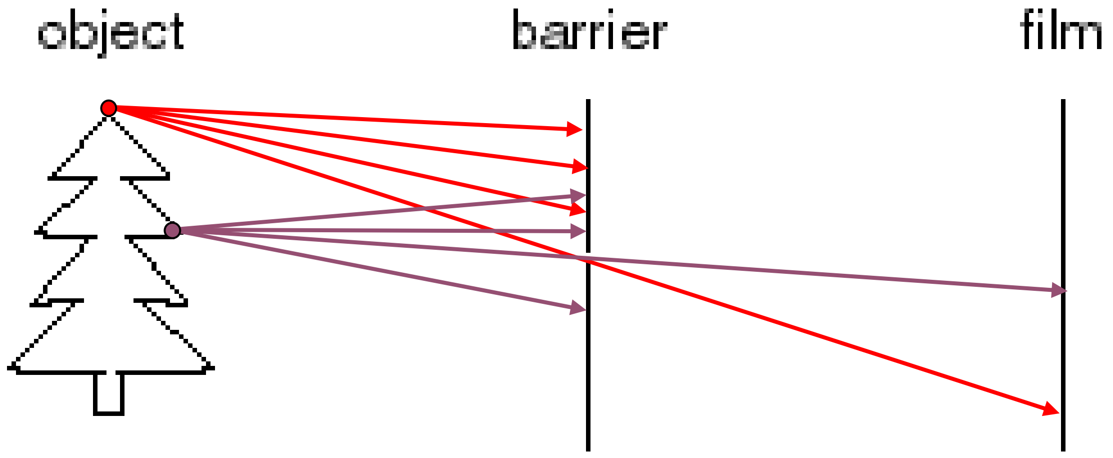
图像的形成



假如你来设计几个照相机…

- 把胶卷放直接在物体前面
- 这是一个好的设计吗？
- 我们会得到什么？

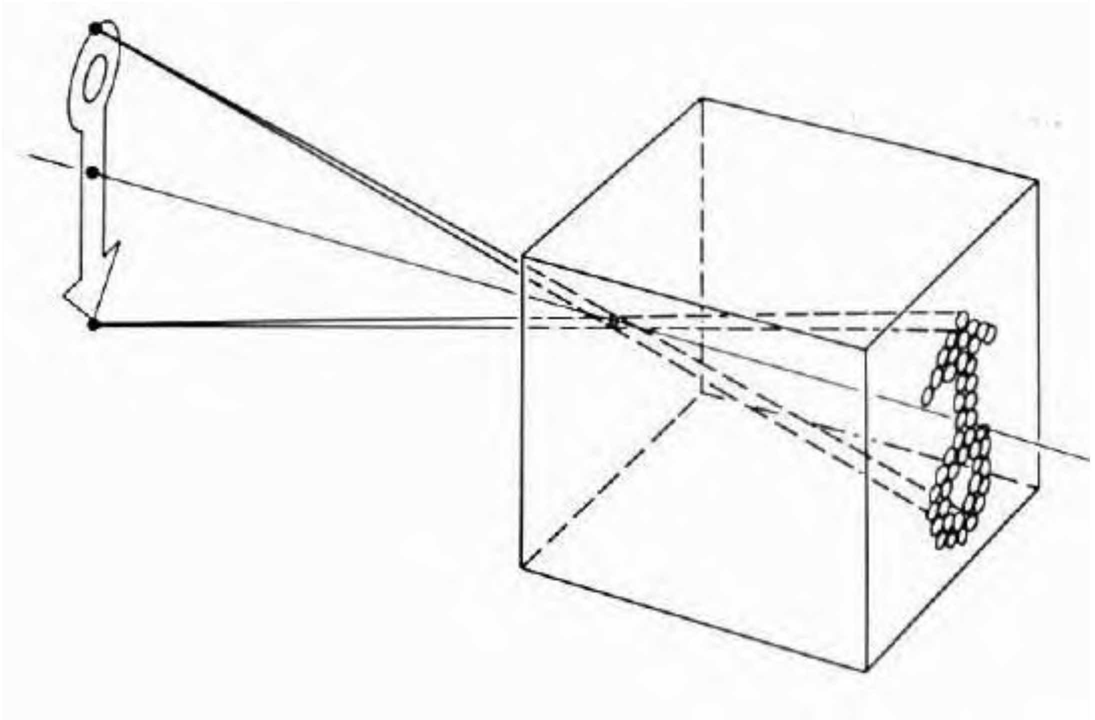
针孔照相机



加一块挡板挡住绝大多数光线

- 没有模糊了！
- 图像在胶卷上会怎么呈现呢?

Camera Obscura 暗箱



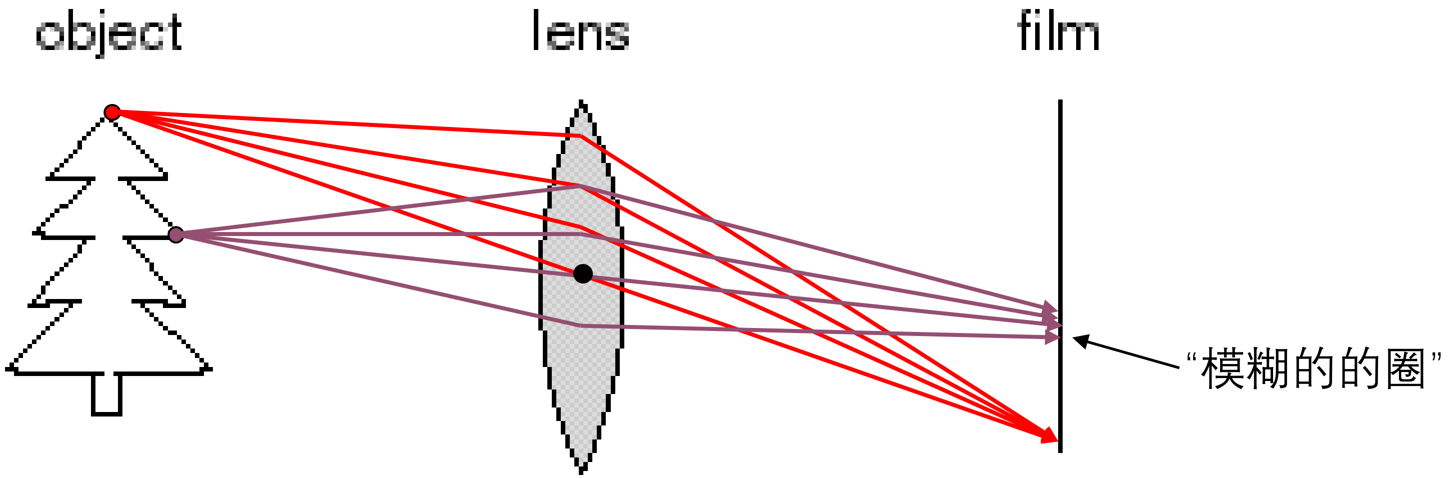
为什么相片还是会有模糊?

我们最常调节照相机的什么设定?

Slide by A. Efros

http://www.debevec.org/Pinhole/

镜头

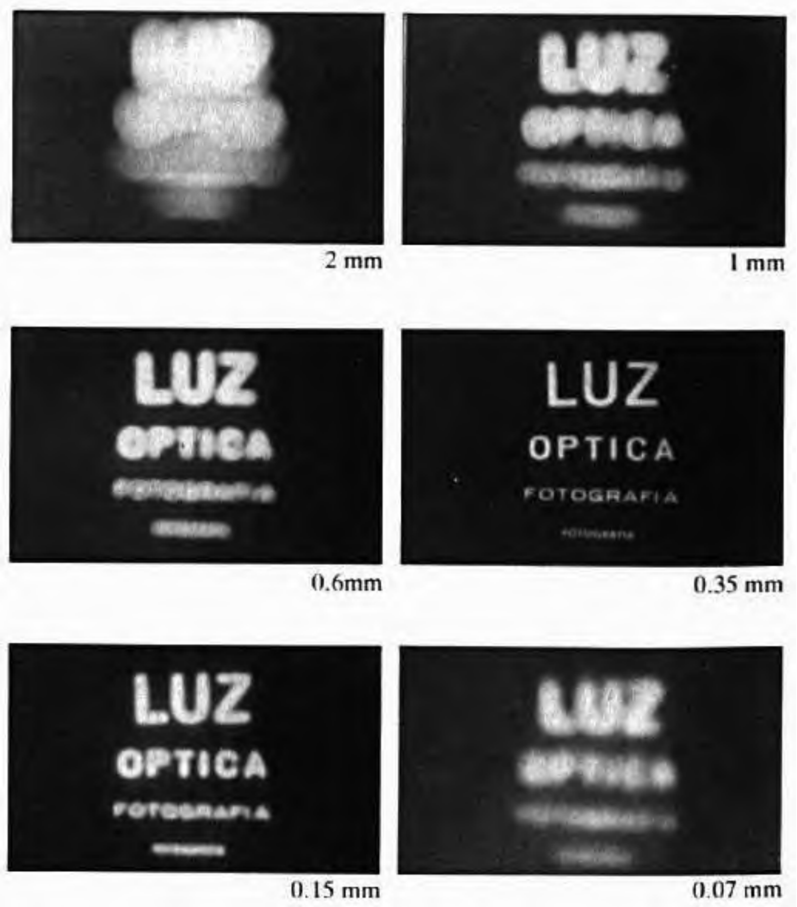


镜头通过折射光线来实现光的聚焦

- 会产生“焦点”和“焦距”
  - 其他的点会产生模糊的“圈”
- 改变镜头可以改变焦距

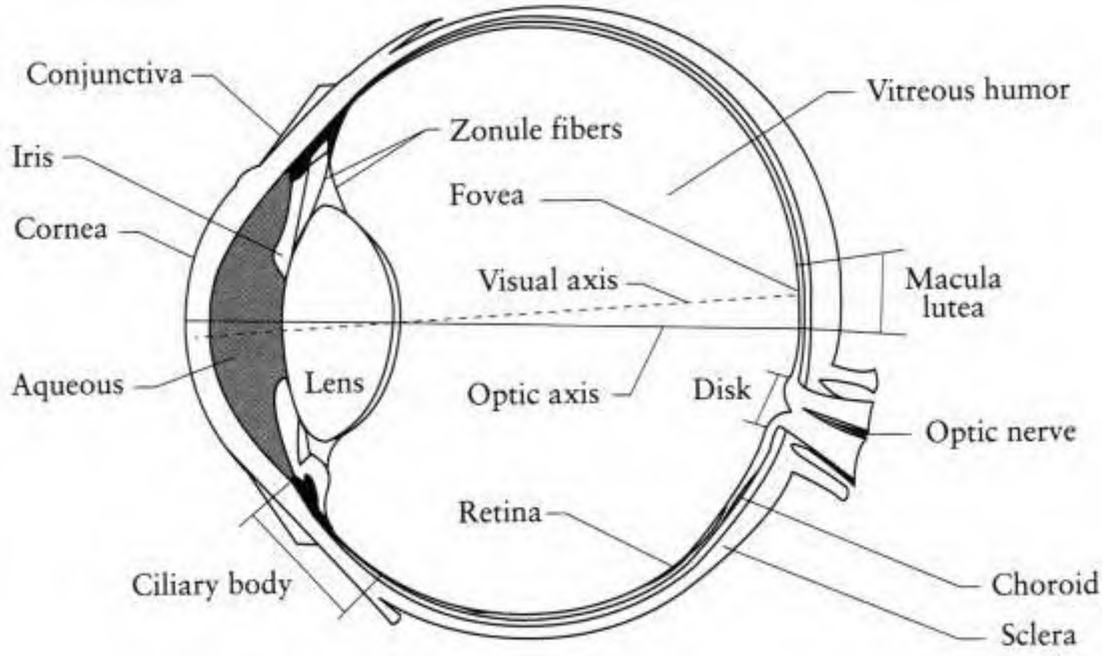
焦距

不同焦距可以调节
清晰度



Fredo Durand

眼睛



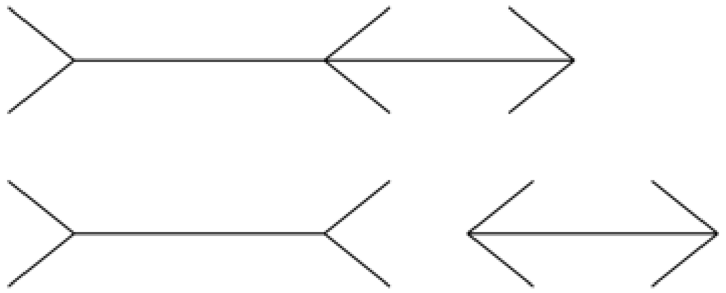
人眼可以近似认为是一个照相机

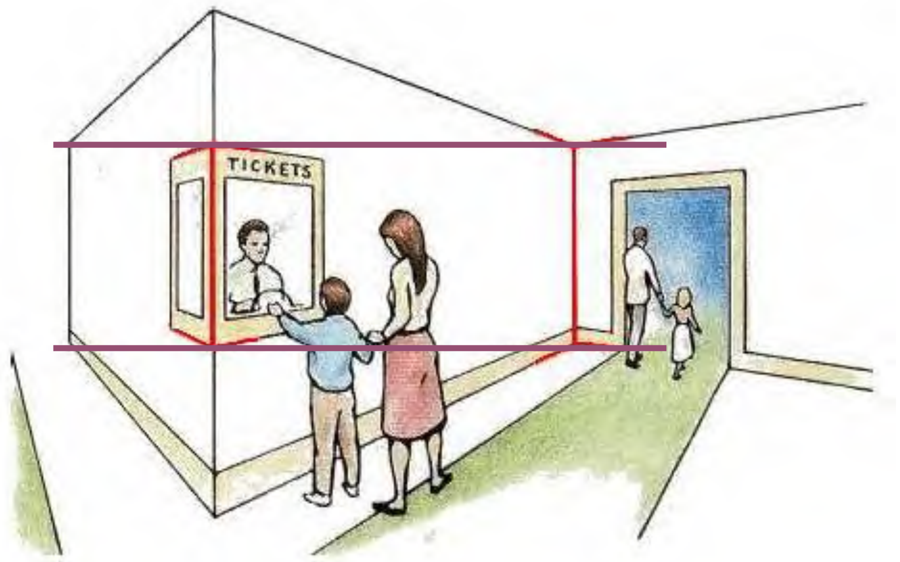
- Iris **虹膜** – 控制瞳孔大小
- Pupil **瞳孔** – 调节“光圈”
- 胶卷在哪里?
  - **retina 视网膜**

Projection 投影：What You See (with Eyes) is What You Get?

Projection 投影：换个角度看

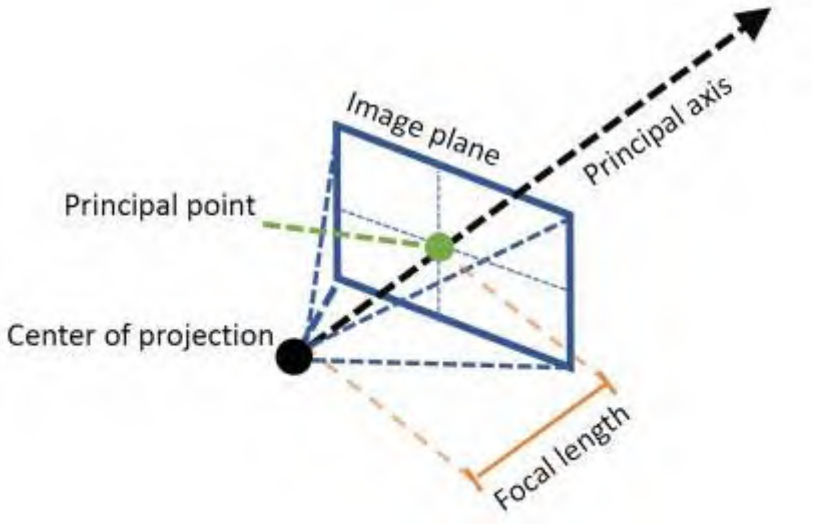
Müller-Lyer Illusion



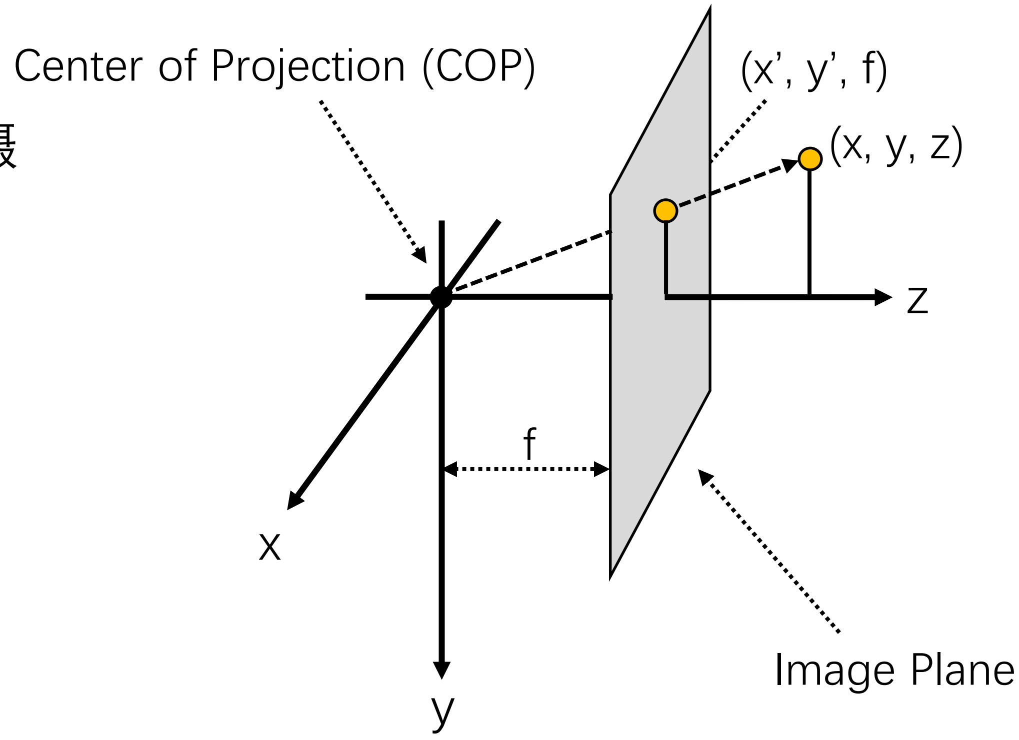


https://en.wikipedia.org/wiki/Müller-Lyer_illusion

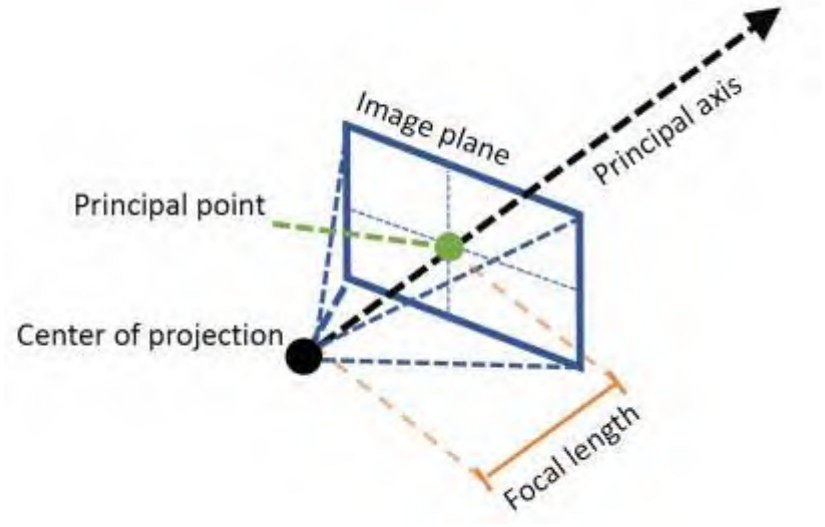
Modeling projection: 投影模型



- 坐标系统
  - 使用针孔相机做近似，将三维世界的信息投影到感光材料上，从而形成二维图像
  - 用小孔当作坐标原点 (Center of Projection, or COP)
  - 图像平面 (或 Projection Plane 投影平面) 针孔摄像机的感光材料或图像平面
  - 为了方便计算图像平面与z轴垂直
  - 针孔相机应朝向z轴的反方向，为了方便表示我们使用投影（透视投影）而不是相机模型来举例



Modeling projection: 投影模型

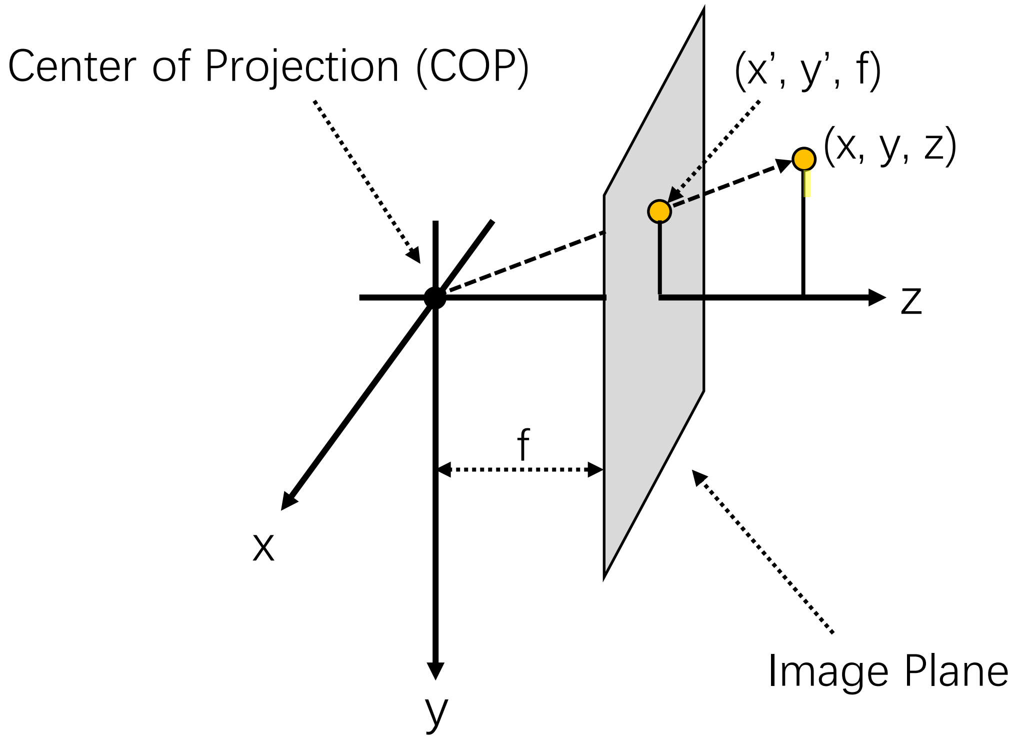


- 投影公式
  - 三维点投影到二维图像平面上： from (x,y,z) to COP
  - 相似三角形

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z}, f)$$

  - 在图像平面:

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$



Homogeneous coordinates : 齐次坐标

- 投影模型并非线性变换 (/z)，不方便用矩阵形式表示
  - 使用齐次坐标

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$ 场景齐次坐标

从齐次坐标转换到三维坐标

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Perspective Projection：透视投影

把投影在齐次坐标用矩阵相乘表示:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

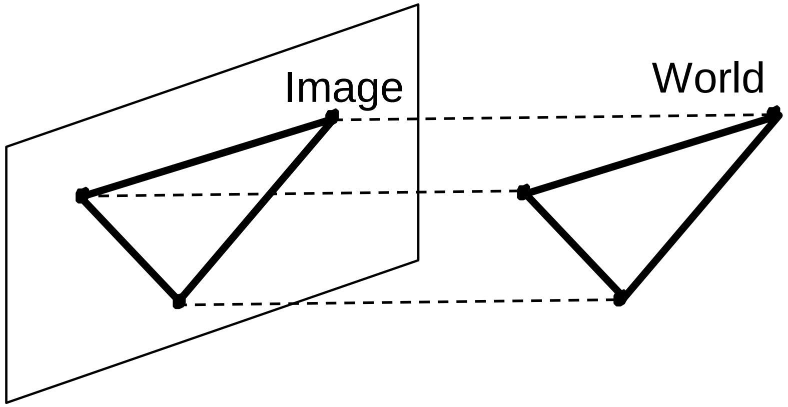
继而转化为平面坐标

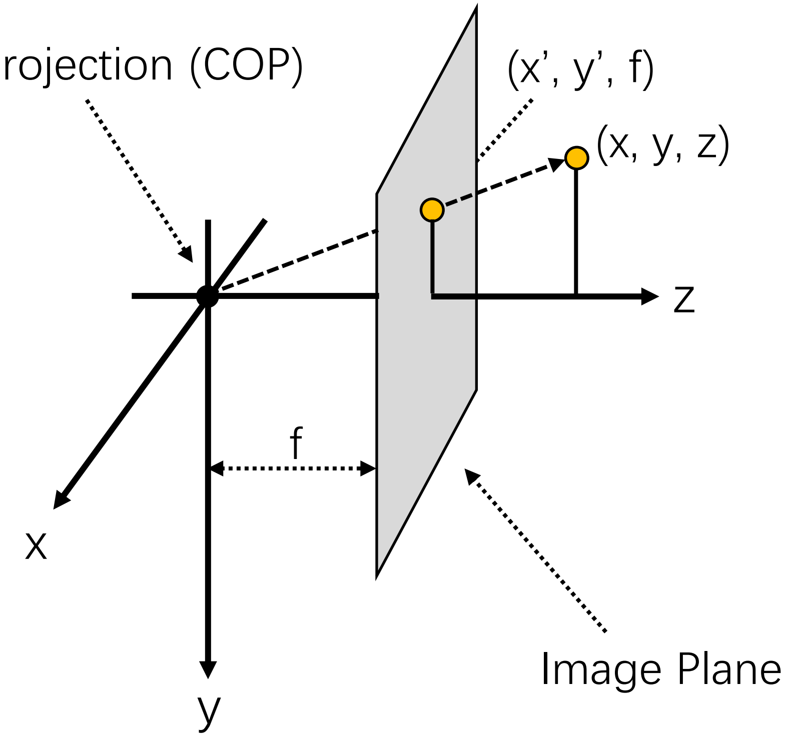
这就是 **透视投影**

- 左边的矩阵是 **投影矩阵**

Orthographic projection：正交投影

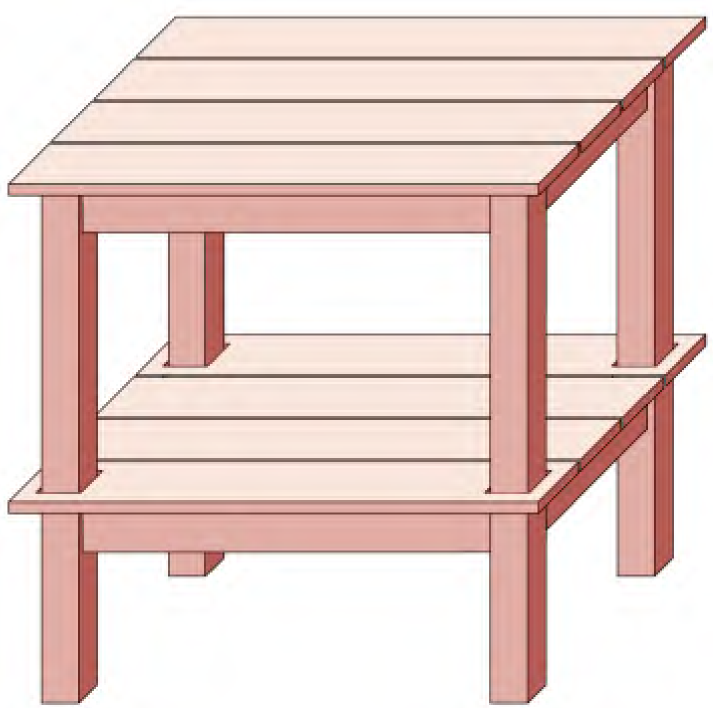
- 一种特殊的透视投影
  - 图像平面和原点距离无限远



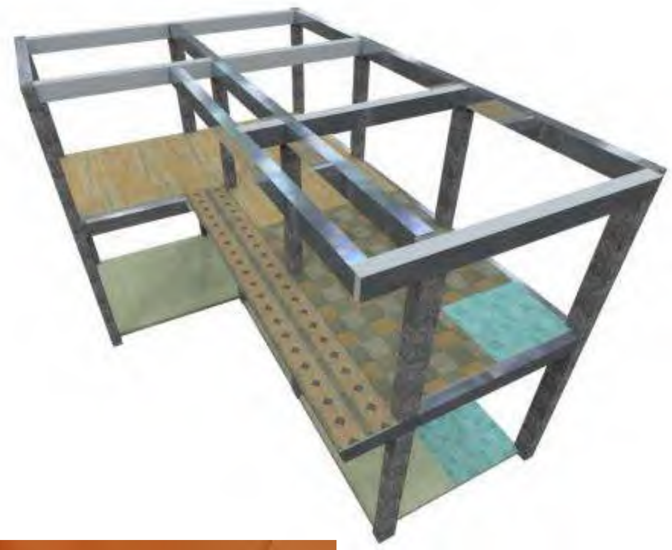


$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

正交投影



透视投影



变种

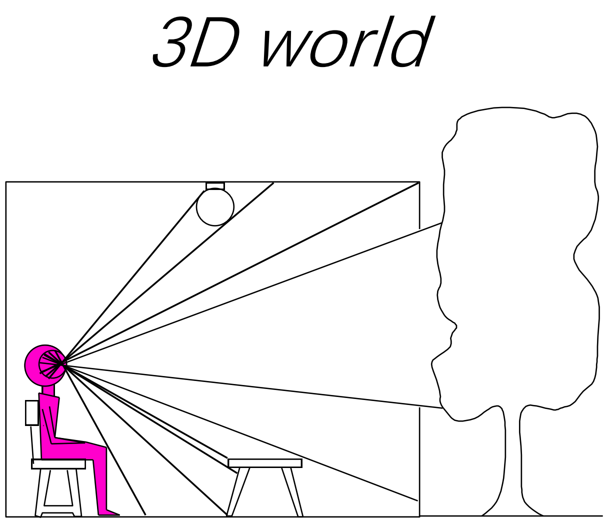
- Scaled orthographic：缩放正交投影，允许将场景中的物体在某个方向上进行缩放，而不会影响其在其他方向上的大小
  - 也叫“Weak Perspective 弱透视投影”

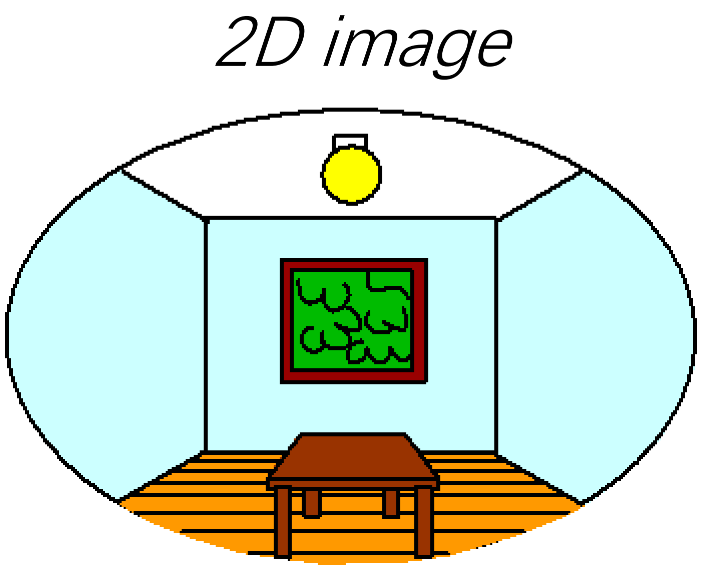
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1/d \end{bmatrix} \Rightarrow (dx, dy)$$

- Affine projection：仿射投影
  - 也叫“Paraperspective 平行透视”

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

投影与维度下降 (3D to 2D)





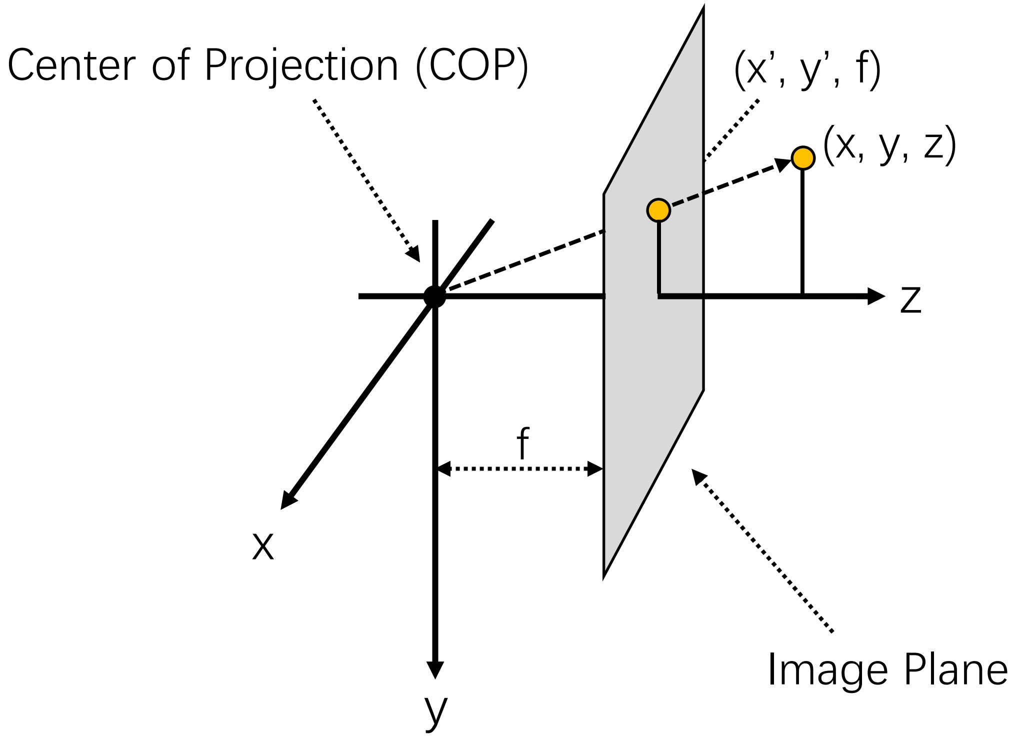
我们损失了什么?

- 视角（相机坐标）
- 距离（长度）

Slide by A. Efros

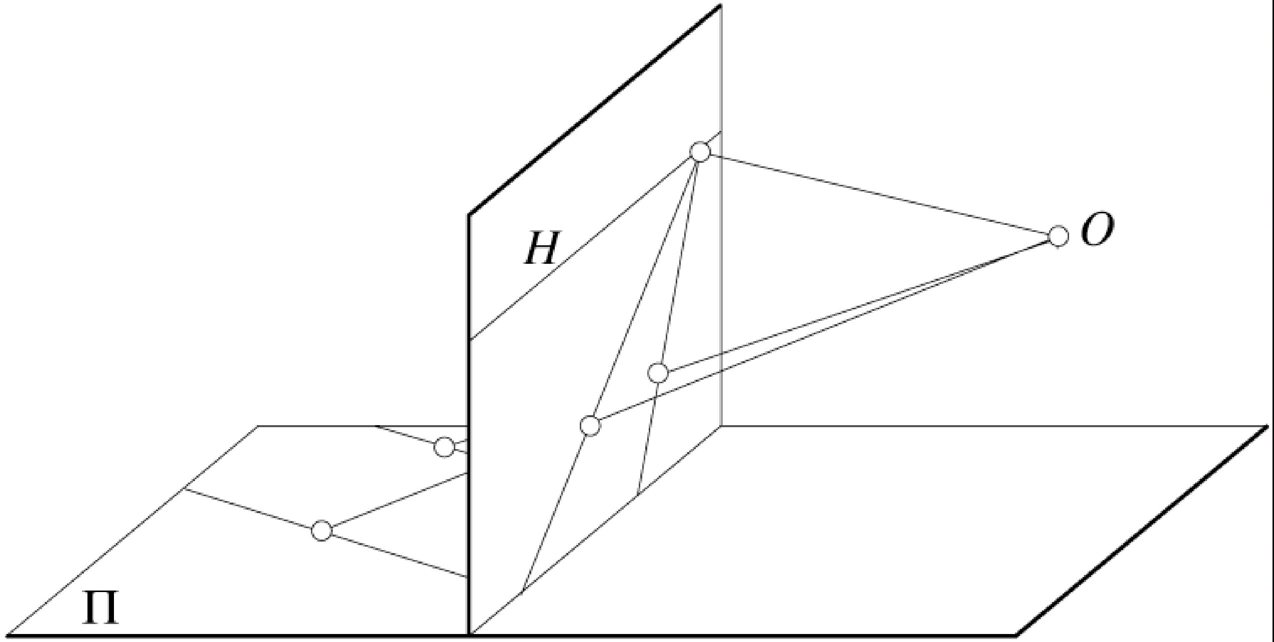
投影的性质

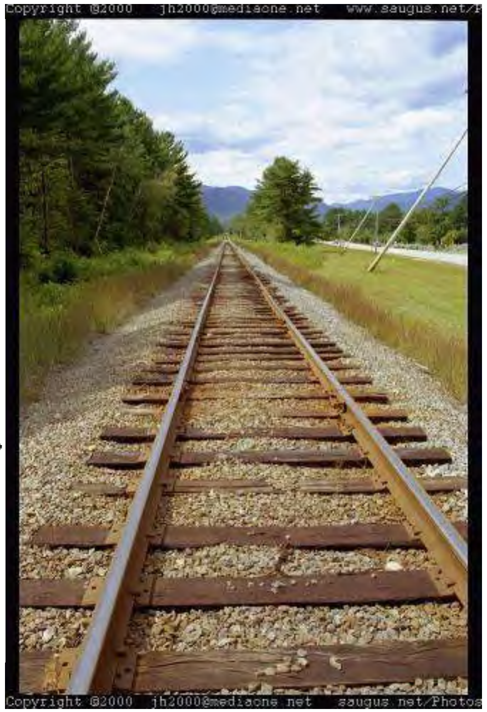
- Many-to-one: 原点出发同一条射线的点会投影到图像的同一个点
- 点 → 投影之后还是点
- 线 → 线
  - 过原点（或延长线）的线会变成点
- 平面 → 平面
  - 过原点（或延长面）的面会变成线



投影的性质

- 两条平行线也有相交的那一天：消失点 vanishing point
  - 图像里有多少个消失点?
  - 图像里是否还是有两条平行线没有相交的那一天呢?

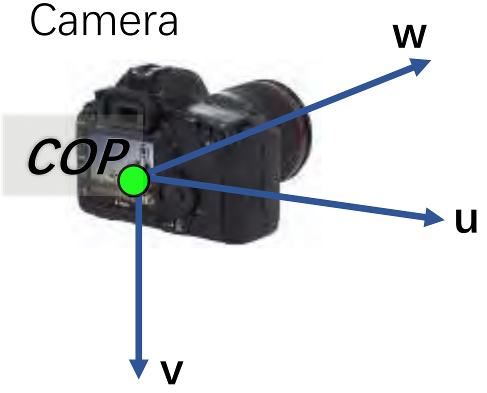




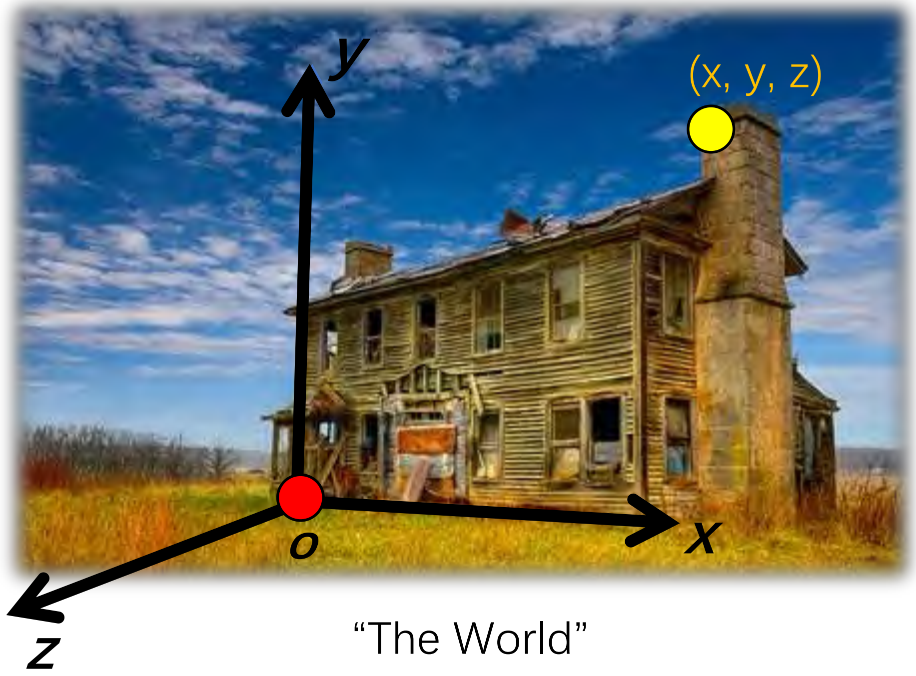
相机参数

- 我们怎么做相机的几何模型? 怎么把现实世界的点（x, y, z）投影到图像像素点上?

Camera



三个重要的坐标系统:
1. 世界坐标
2. 相机坐标
3. 图像坐标



“The World”

坐标系叠加

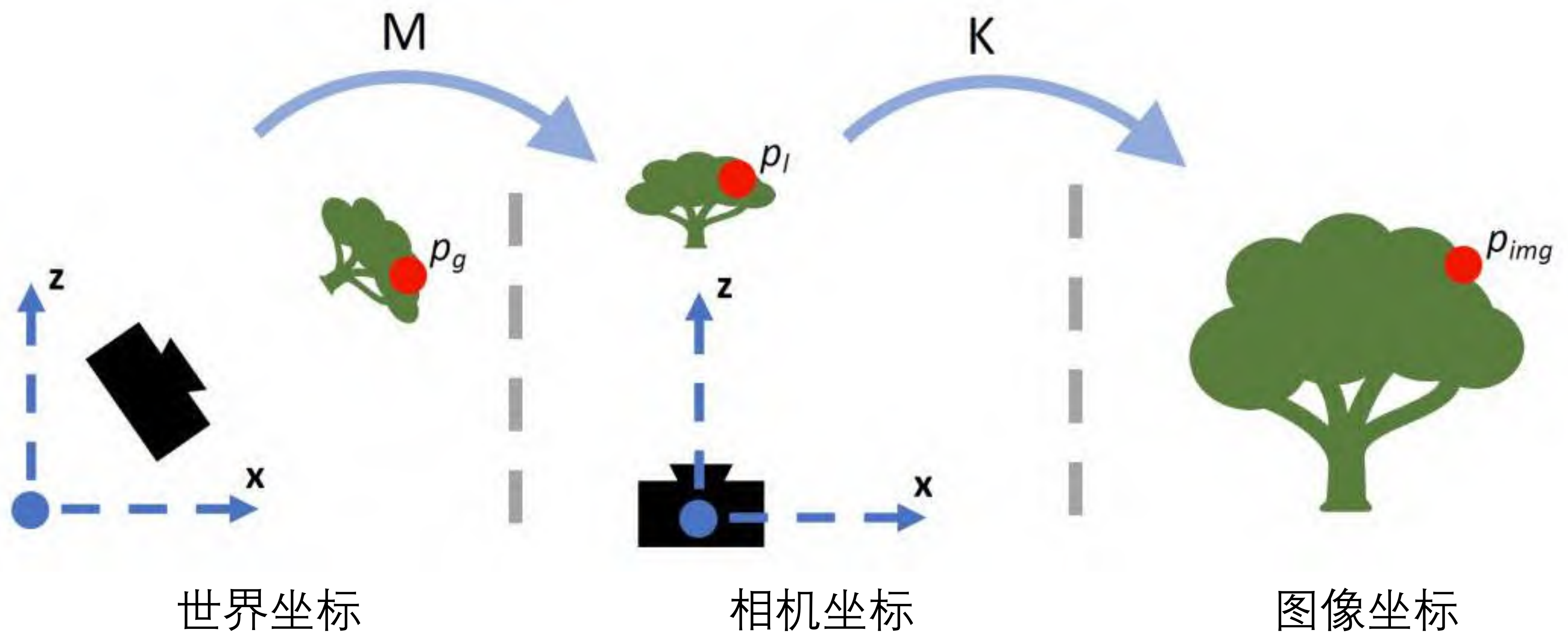
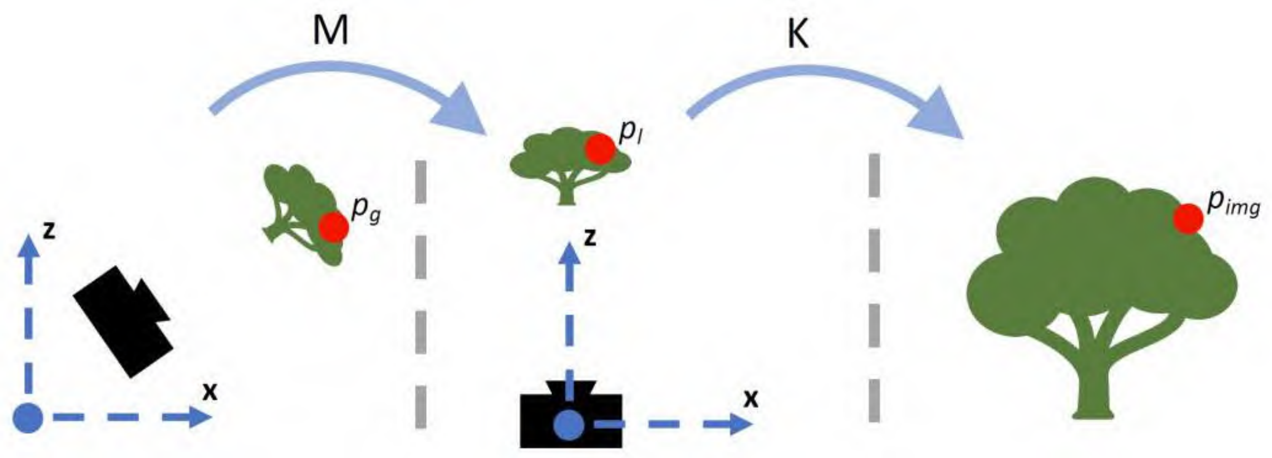


Figure credit: Peter Hedman

相机参数

把世界坐标系下的(x, y, z) 投影到图像上

- 把 (x, y, z) 转换到相机坐标系下
- 前提是需要什么?
    - 相机位置 (世界坐标系下)
    - 相机朝向 (世界坐标系下)
- 然后把点投影到图像坐标系上
    - 需要知道相机内参

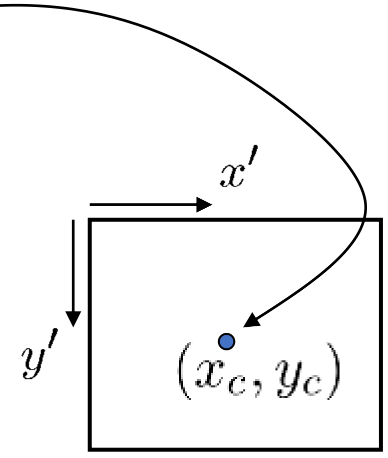


相机参数

相机有很多参数

- 相机光学中心与世界坐标系的变换T
- 图像平面旋转角度 R
- 焦距 f, 图像中心点 (c_x, c_y), 图像分辨率α
- 蓝色为“外参,” 红色为“内参”

投影公式



$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi}\mathbf{X}$$

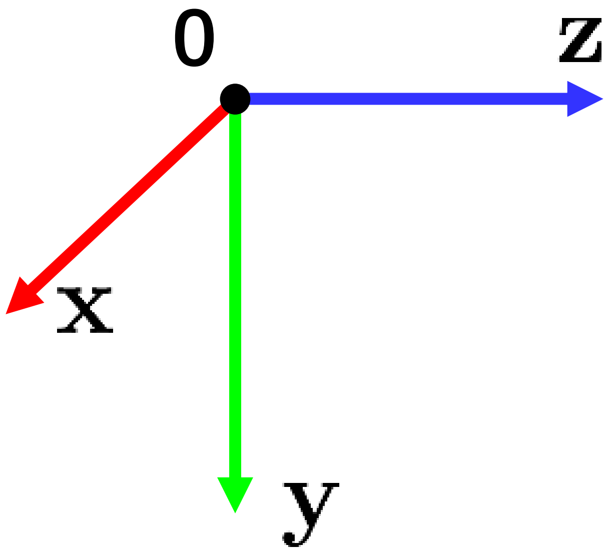
- 投影将是很多变换的叠加
- 以上参数可以把这些变换拆解

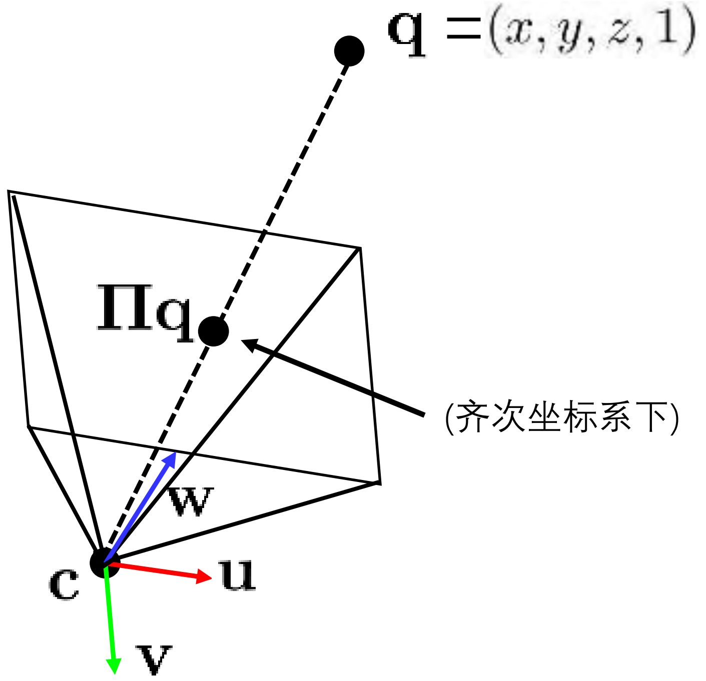
identity matrix

$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} f & s & c_x \\ 0 & \alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{内参}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{投影}} \underbrace{\begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 0 \end{bmatrix}}_{\text{旋转}} \underbrace{\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{T}_{3\times1} \\ \mathbf{0}_{1\times3} & 0 \end{bmatrix}}_{\text{坐标变换}}$$

- 请注意：这里只是简单模型，真实相机很复杂！

投影矩阵

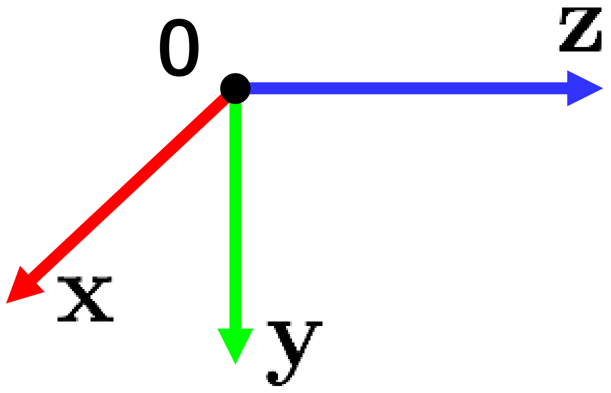




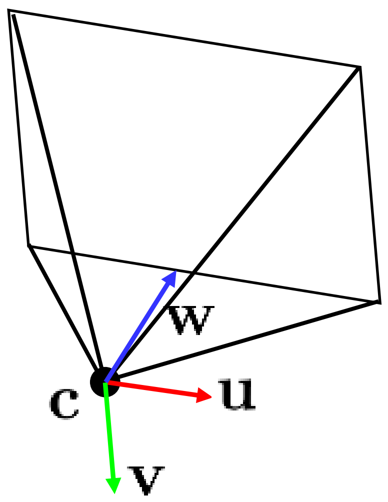
Π: 如何将世界坐标中的点（x, y, z）
投影到图像坐标（u, v）上

外参

- 投影中心为原点, x轴向右, y轴向下, z轴向前

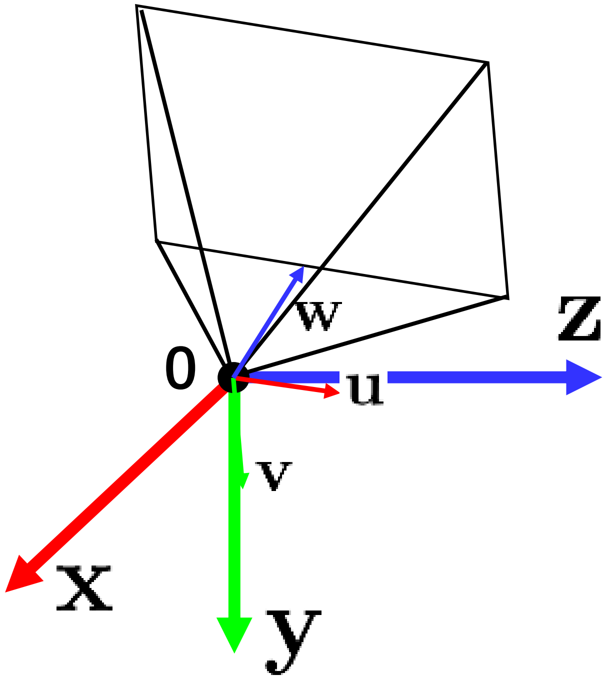


Step 1:世界坐标系的变换 -**c**



外参

- 投影中心为原点, x轴向右, y轴向下, z轴向前



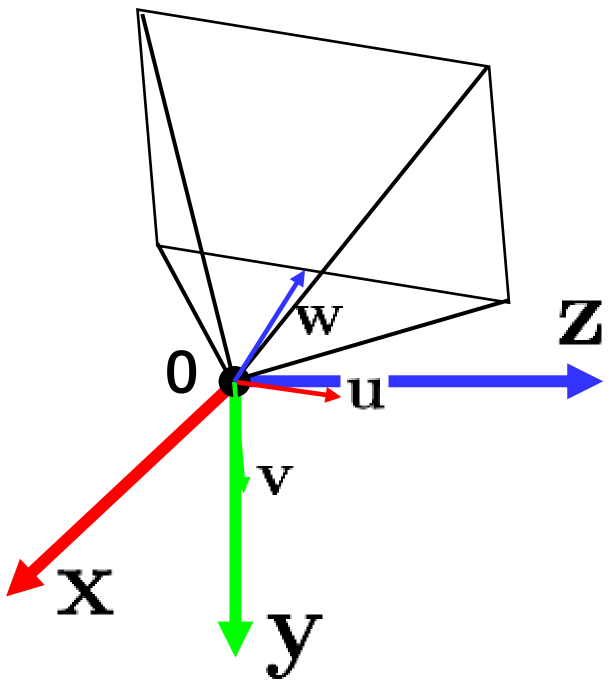
Step 1:世界坐标系的变换 -**c**

矩阵形式:

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{3\times3} & -\mathbf{c} \\ 0\ \ 0\ \ 0 & 1 \end{bmatrix}$$

外参

- 投影中心为原点, x轴向右, y轴向下, z轴向前



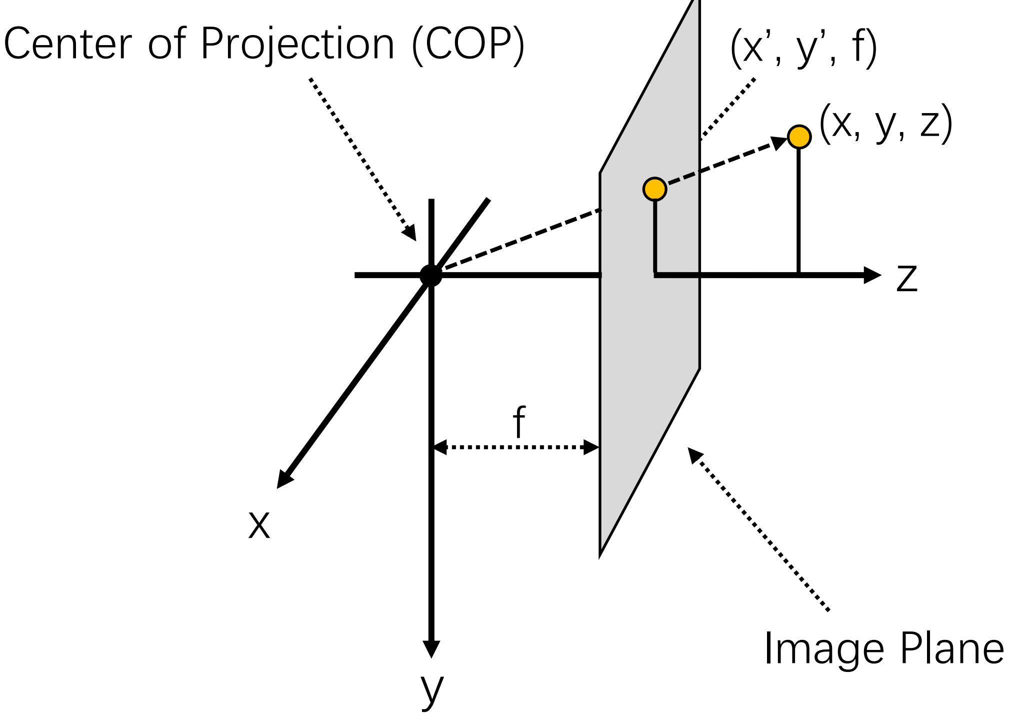
Step 1:世界坐标系的变换 -**c**
Step 2: 旋转 **R**

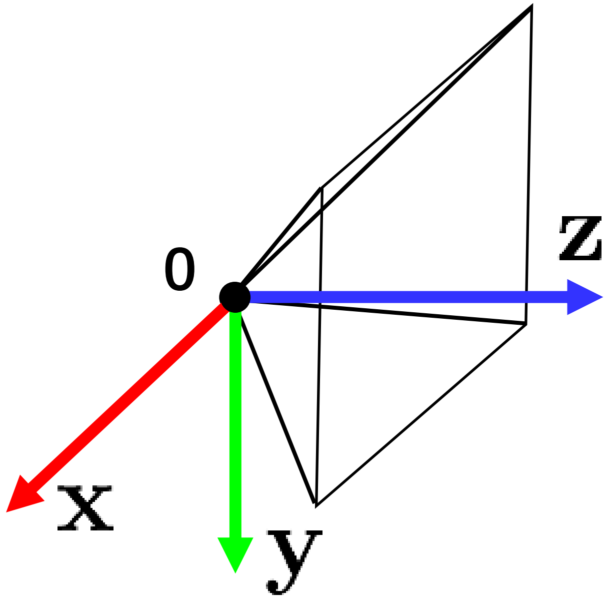
$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

3x3 rotation matrix

外参

- 投影中心为原点, x轴向右, y轴向下, z轴向前





Step 1:世界坐标系的变换 -**c**
Step 2: 旋转 **R**

$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

(with extra row/column of [0 0 0 1])

透视投影

$$\underbrace{\begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(将三维射线上的点投影到二维平面)

(内参)

一般来说， $\mathbf{K} = \begin{bmatrix} f & s & c_x \\ 0 & \alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}$ (上三角矩阵)

α : **纵横比** (一般是1除非不是方形图像)

s : **倾斜** (一般是0 除非图像平面与相机光轴之间的倾斜)

(c_x, c_y) : **主点** (一般是(w/2,h/2) 除非出现偏移)

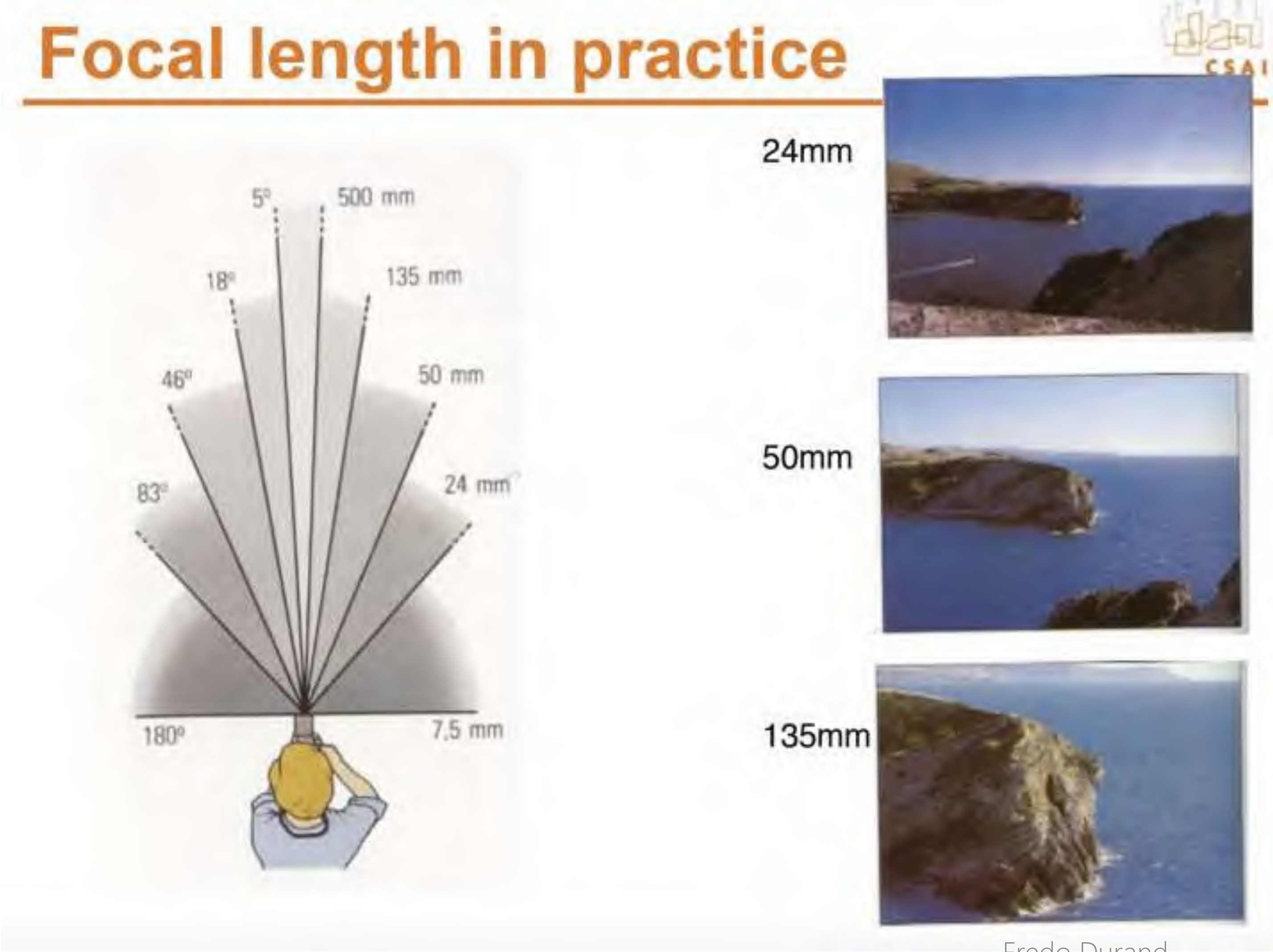
一般相机的内参模型

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

- **2D 仿射变换**： 放缩 f(焦距) + 变换(c_x, c_y) (主点)

焦距

不同焦距有不同的视野



Fredo Durand

焦距

- 可以当作放大

24mm

50mm

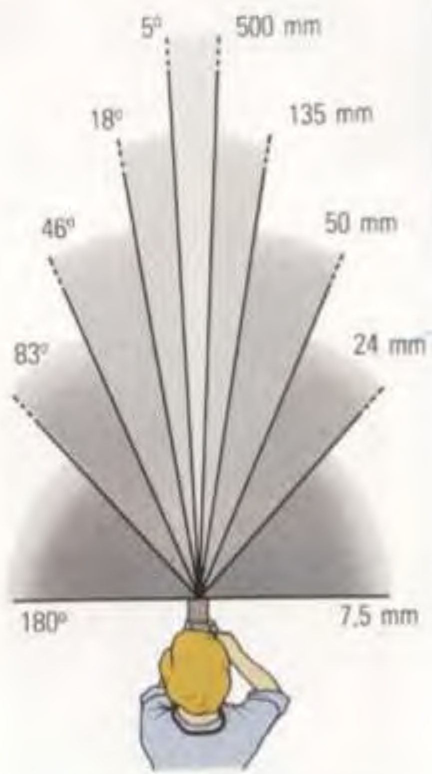
200mm

800mm

- 也可以认为是*视野的变换*

焦距 Focal length = cropping

不同焦距可以看清
不同距离的景色



24mm

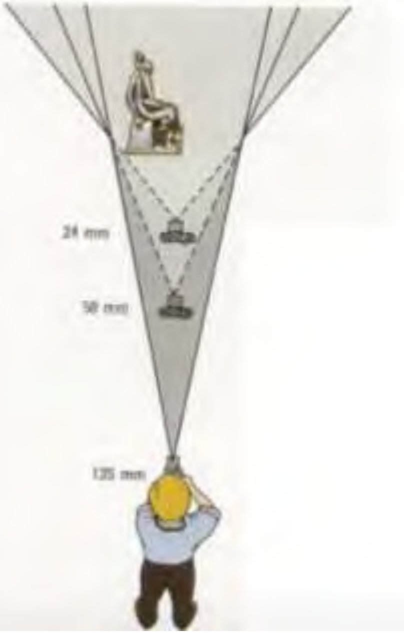
50mm

135mm

Fredo Durand

Focal length vs. viewpoint

- **Telephoto makes it easier to select background (a small change in viewpoint is a big change in background.**

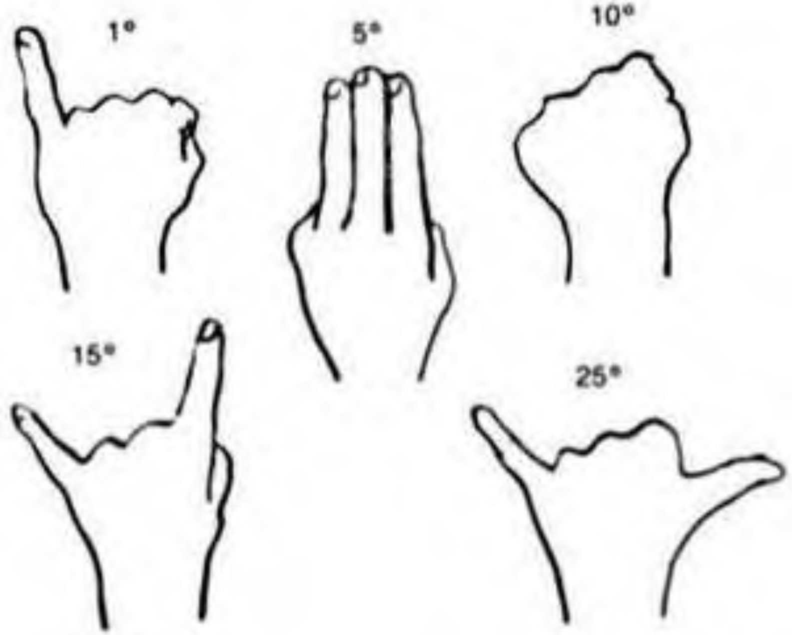


Grand-angulaire 24 mm

Normal 50 mm

Longue focale 135 mm

视场角与剪裁因子 CheatSheet



APS-C Crop Body Measurement Table

Lens	After 1.62 Multiplier	APS-C Sensor (1.62 lens multiplier) Canon 60D, 7D, 70D, T3i, T4i	Hand Positions
18mm	29.16mm	**Three hands wide** at full arms length.	
28mm	45.36mm	**Slightly less than two hands wide** at full arms length.	
35mm	56.7mm	**One hand + width of one fist** at full arms length.	
50mm	81mm	**One hand wide + width of thumb** at full arms length.	
55mm	89.1mm	**Slightly less than one hand wide** at full arms length.	
85mm	137.7mm	**Inside edge of thumb to tip of forefinger wide** with hand in "L" shape, thumb up.	

固定前景下不同焦距拍摄效果

不同焦距拍摄效果不同

广角（短焦）　　标准镜头　　长焦镜头

知道为什么
专业摄影师
都用大长焦
了吧！

http://petapixel.com/2013/01/11/how-focal-length-affects-your-subjects-apparent-weight-as-seen-with-a-cat/

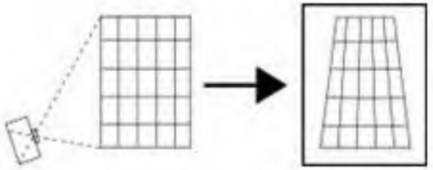
透视失真

- 拍摄建筑物时经常出现这种情况：

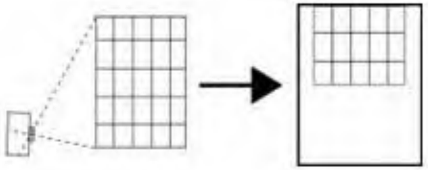
Source: F. Durand

透视失真

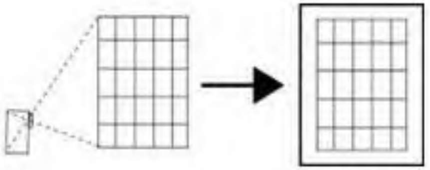
- 怎么办呢?



向上倾斜摄像头
建筑物上下比例失衡

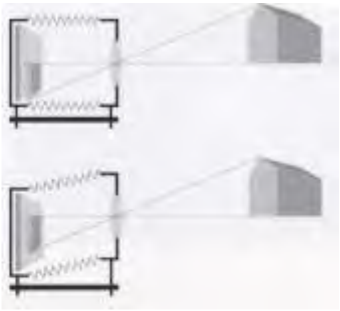


保持照相机水平，则建筑
物上面部分拍不到



把透镜向上移就可以
解决这个问题

- 解决方案: view camera (lens shifted w.r.t. film)



http://en.wikipedia.org/wiki/Perspective_correction_lens

Source: F. Durand

透视失真

- 效果:

Source: F. Durand

透视失真

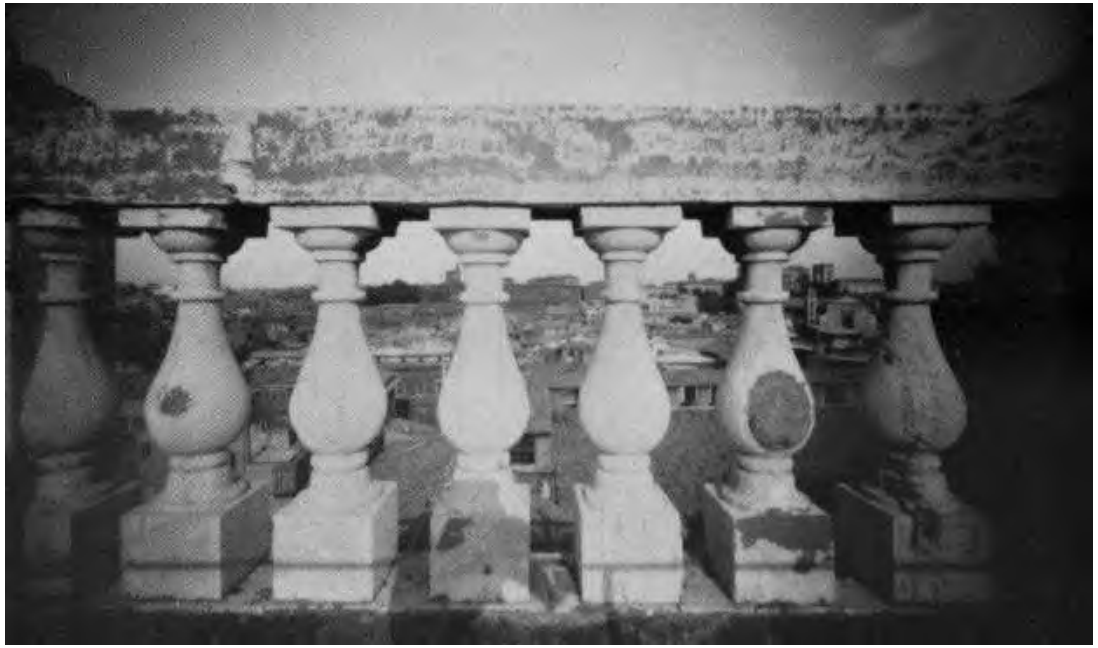
- 水平失真如何解决?

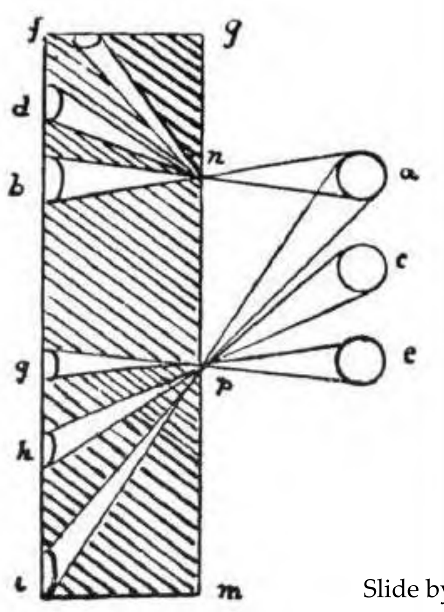
Image source: F. Durand

透视失真: 人像

透视失真

- 外面的柱子看起来更大
- Problem pointed out by Da Vinci





Slide by F. Durand

https://aaronhertzmann.com/2022/02/28/how-does-perspective-work.html

Distortion-Free Wide-Angle Portraits on Camera Phones

(a) A wide-angle photo with distortions on subjects' faces.

(b) Distortion-free photo by our method.

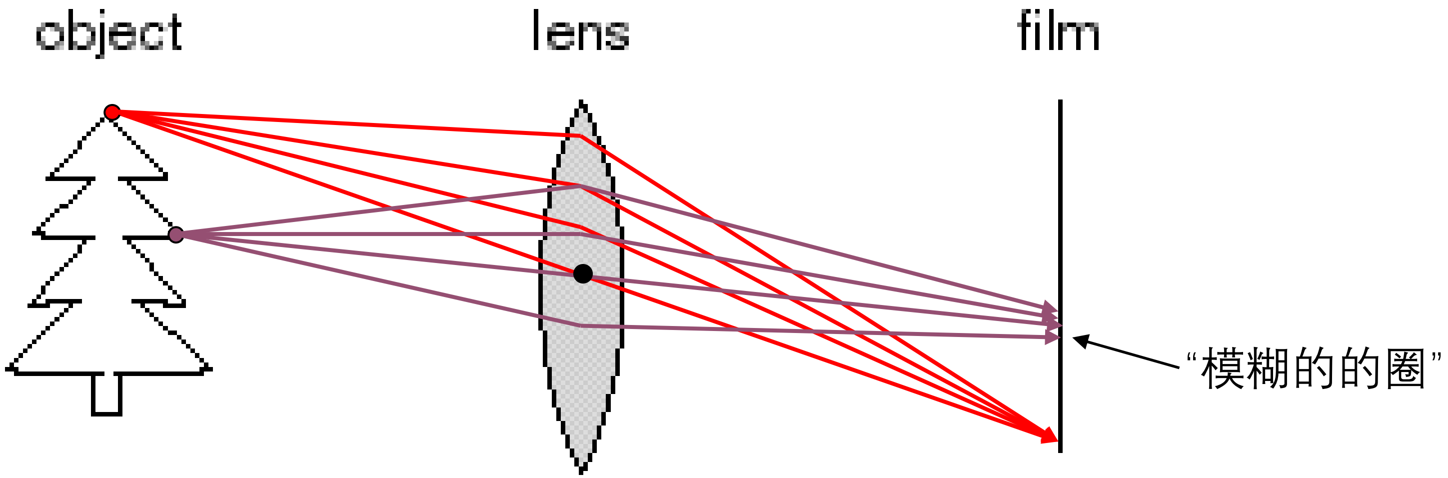
YiChang Shih, Wei-Sheng Lai, and Chia-Kai Liang, Distortion-Free Wide-Angle Portraits on Camera Phones, SIGGRAPH 2019
https://people.csail.mit.edu/yichangshih/wide_angle_portrait/

怎么利用透视失真?

https://baijiahao.baidu.com/s?id=1614554646776556897&wfr=spider&for=pc

回顾——相机与投影几何

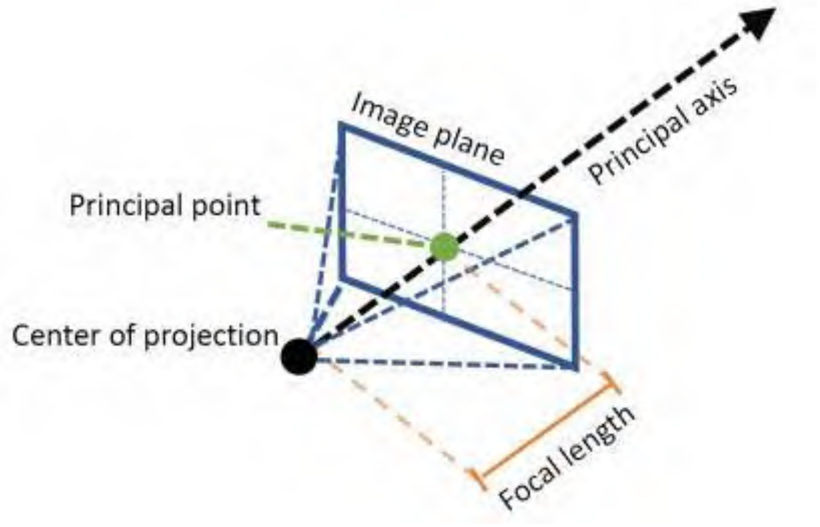
单孔相机与镜头



镜头通过折射光线来实现光的聚焦

- 会产生“焦点”和“焦距”
  - 其他的点会产生模糊的“圈”
- 改变镜头可以改变焦距

投影模型：三维到二维的线性变换

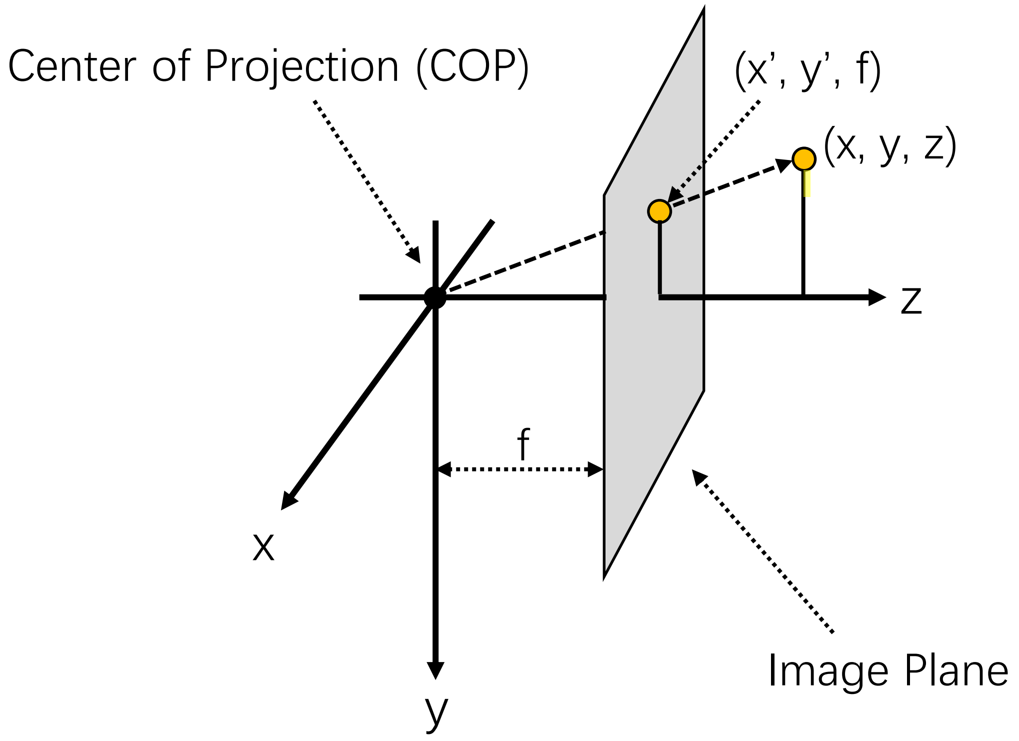


- 投影公式
  - 三维点投影到二维图像平面上： from (x,y,z) to COP
  - 相似三角形

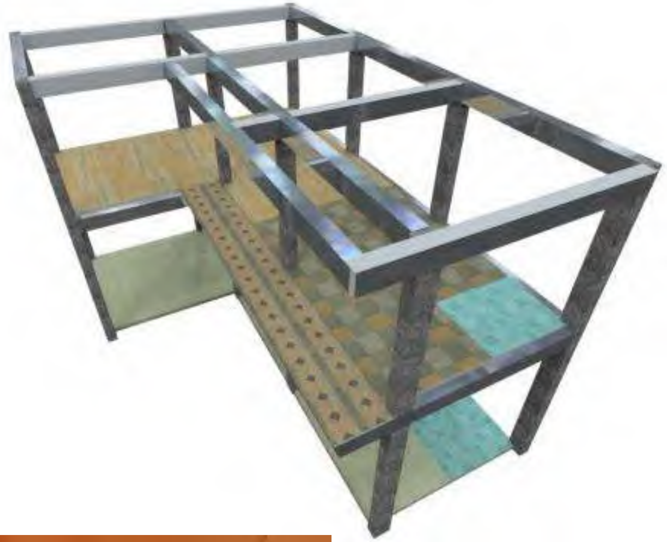
$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z}, f)$$

  - 在图像平面:

$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$



透视投影：近大远小



相机投影：坐标系叠加

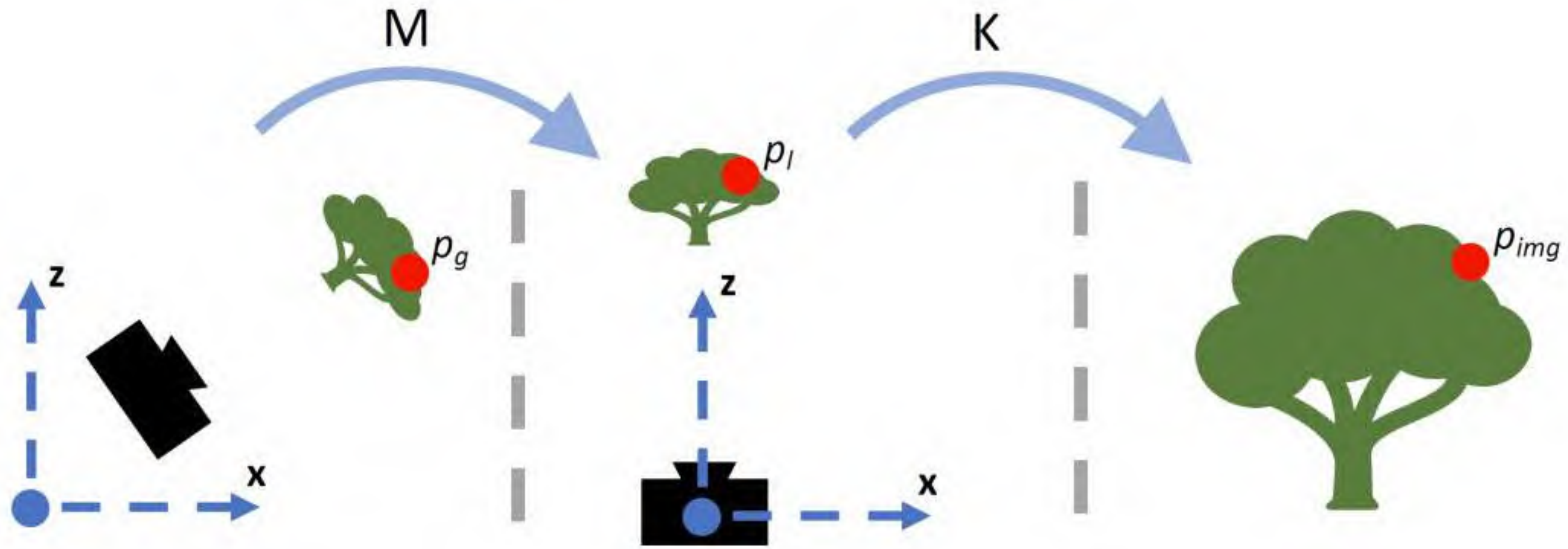
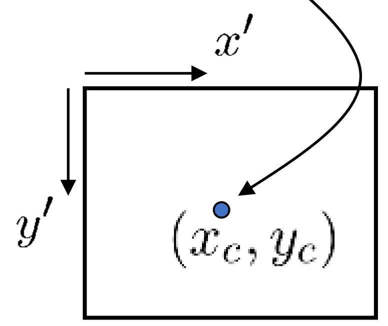


Figure credit: Peter Hedman

相机参数与投影矩阵的关系

相机有很多参数

- 相机光学中心与世界坐标系的变换T
- 图像平面旋转角度 R
- 焦距 f, 图像中心点 (c_x, c_y), 图像分辨率α
- 蓝色为"外参," 红色为"内参"



投影公式

$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$

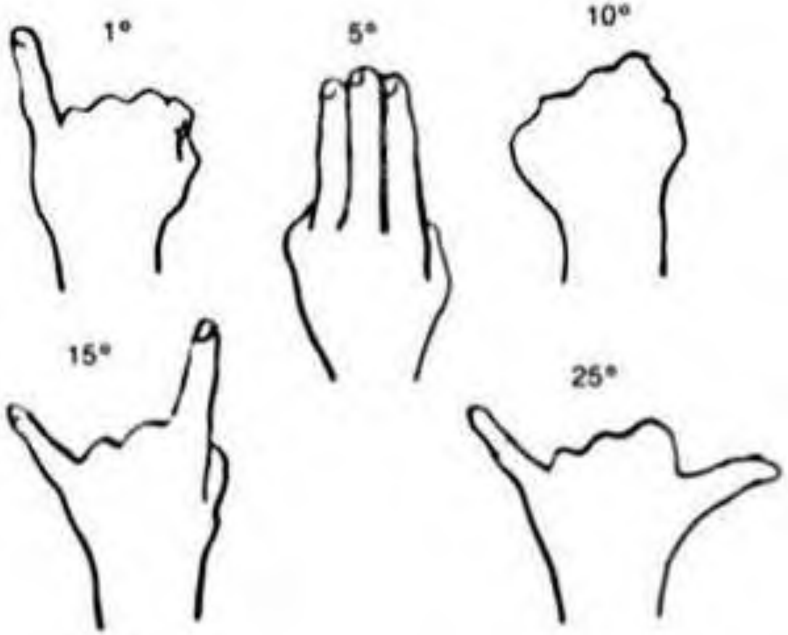
- 投影将是很多变换的叠加
- 以上参数可以把这些变换拆解

identity matrix

$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} f & s & c_x \\ 0 & \alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{内参}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{投影}} \underbrace{\begin{bmatrix} \mathbf{R}_{3\times 3} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 0 \end{bmatrix}}_{\text{旋转}} \underbrace{\begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{T}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 0 \end{bmatrix}}_{\text{坐标变换}}$$

- 请注意：这里只是简单模型，真实相机很复杂！

视场角与剪裁因子 CheatSheet



APS-C Crop Body Measurement Table

Lens	After 1.62 Multiplier	APS-C Sensor (1.62 lens multiplier) Canon 60D, 7D, 70D, T3i, T4i	Hand Positions
18mm	29.16mm	**Three hands wide** at full arms length.	
28mm	45.36mm	**Slightly less than two hands wide** at full arms length.	
35mm	56.7mm	**One hand + width of one fist** at full arms length.	
50mm	81mm	**One hand wide + width of thumb** at full arms length.	
55mm	89.1mm	**Slightly less than one hand wide** at full arms length.	
85mm	137.7mm	**Inside edge of thumb to tip of forefinger wide** with hand in "L" shape, thumb up.	

不同焦距拍摄效果不同

广角（短焦）

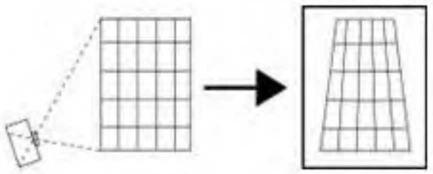
标准镜头

长焦镜头

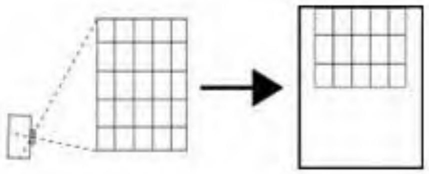
知道为什么
专业摄影师
都用大长焦
了吧！

透视失真

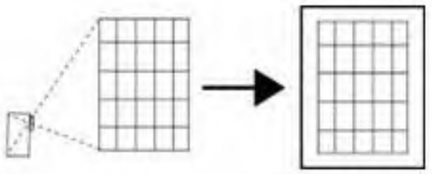
- 怎么办呢?



向上倾斜摄像头
建筑物上下比例失衡

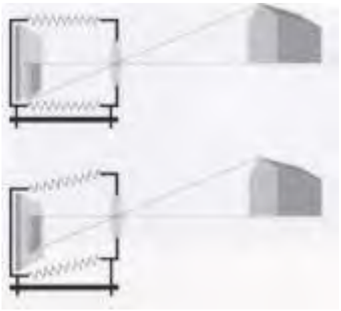


保持照相机水平，则建筑
物上面部分拍不到



把透镜向上移就可以
解决这个问题

- 解决方案: view camera (lens shifted w.r.t. film)



http://en.wikipedia.org/wiki/Perspective_correction_lens

Source: F. Durand

怎么利用透视失真?

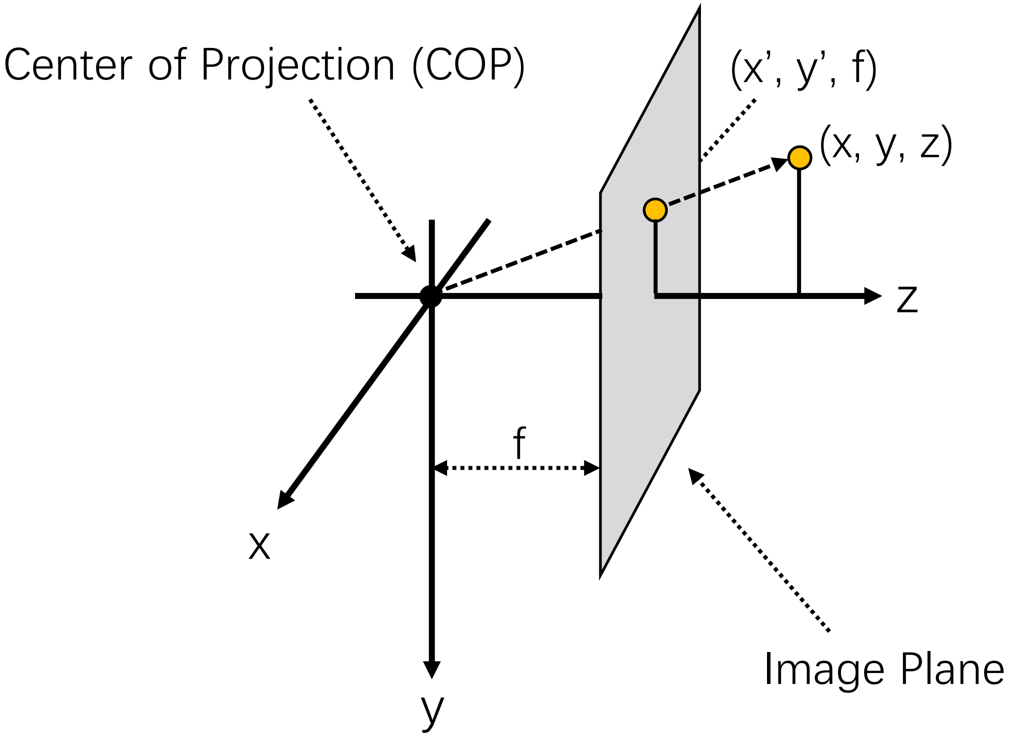
https://baijiahao.baidu.com/s?id=1614554646776556897&wfr=spider&for=pc

视觉错觉

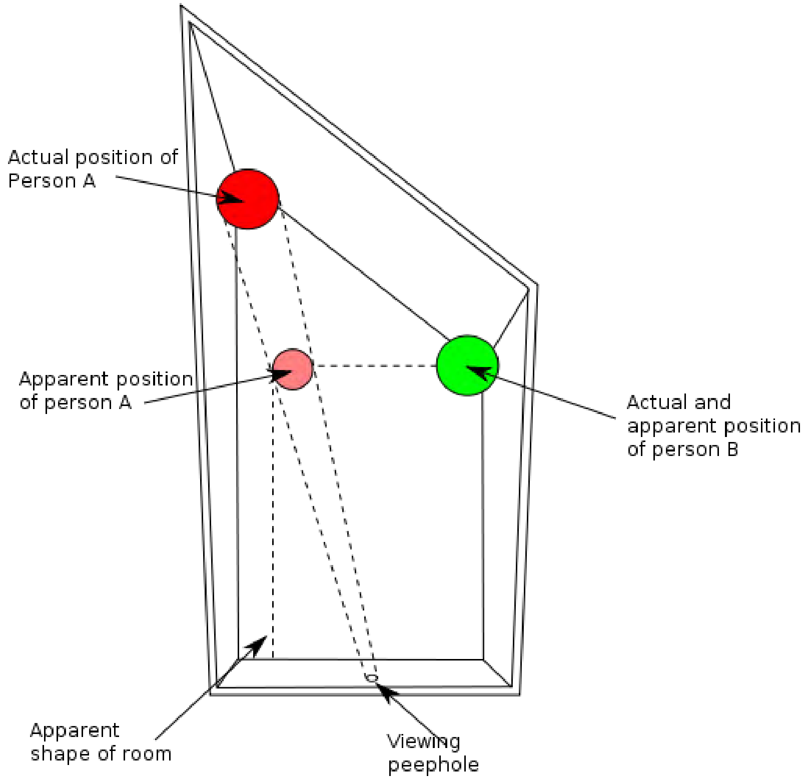
错觉原理


MAKE POVERTY HISTORY
CoolOpticalIllusions.com


CoolOpticalIllusions.com


Center of Projection (COP)
(x’, y’, f)
(x, y, z)
z
f
x
y
Image Plane

Ames Room：猜猜怎么做到的?



图像图形学

单视图与立体视图

图像图形学

- 我们怎么获得视觉信息
- **无监督学习**，更关注数学建模
- 照相机模型与成像原理
- 图像处理基础
- 二维几何
- 三维几何
- **次要目的**：提升学生的拍照、修图水平
- **首要目的**：为感知理解提供基础

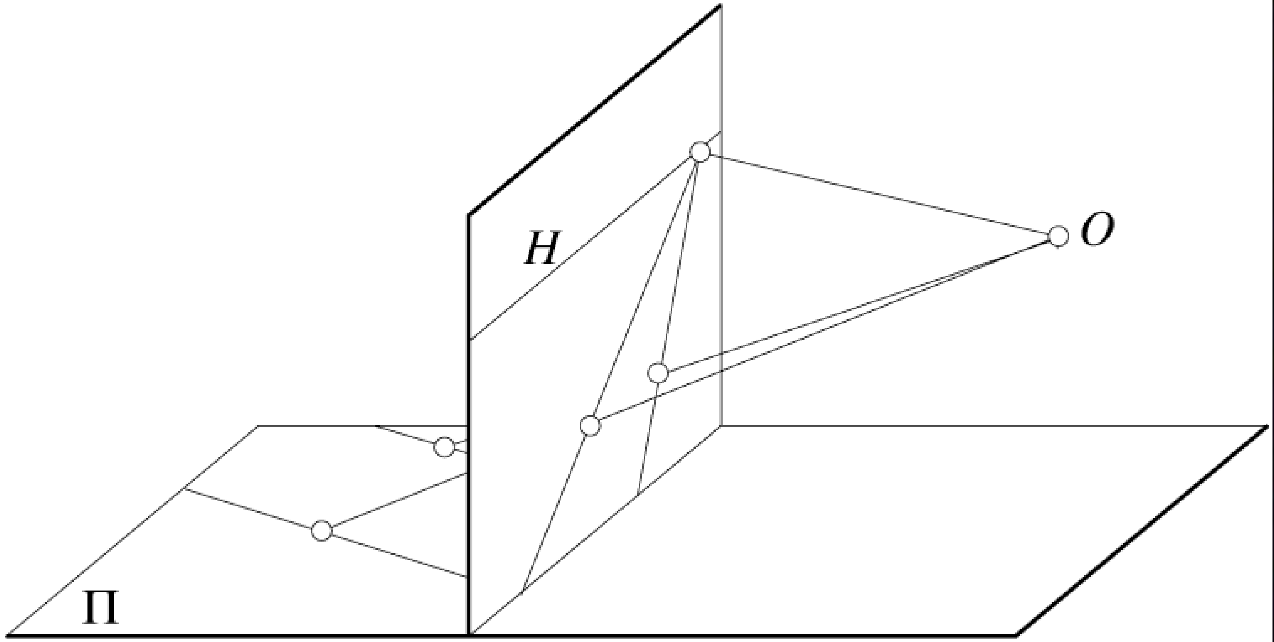
透视几何的应用

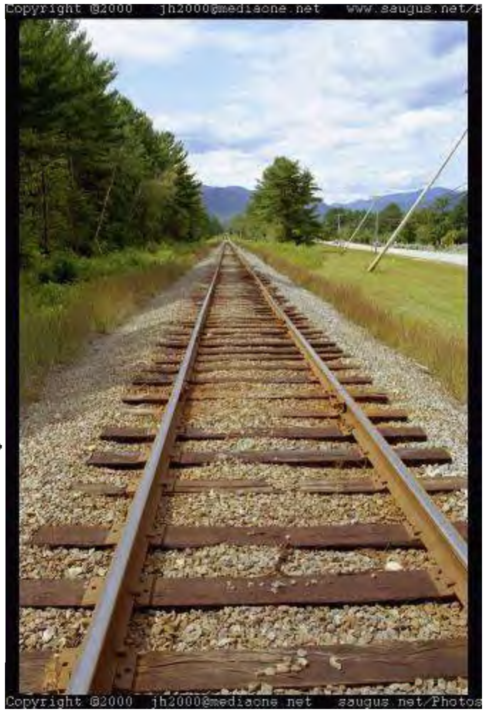
Vermeer's *Music Lesson*

Reconstructions by Criminisi et al.

投影的性质

- 两条平行线也有相交的那一天：消失点 vanishing point





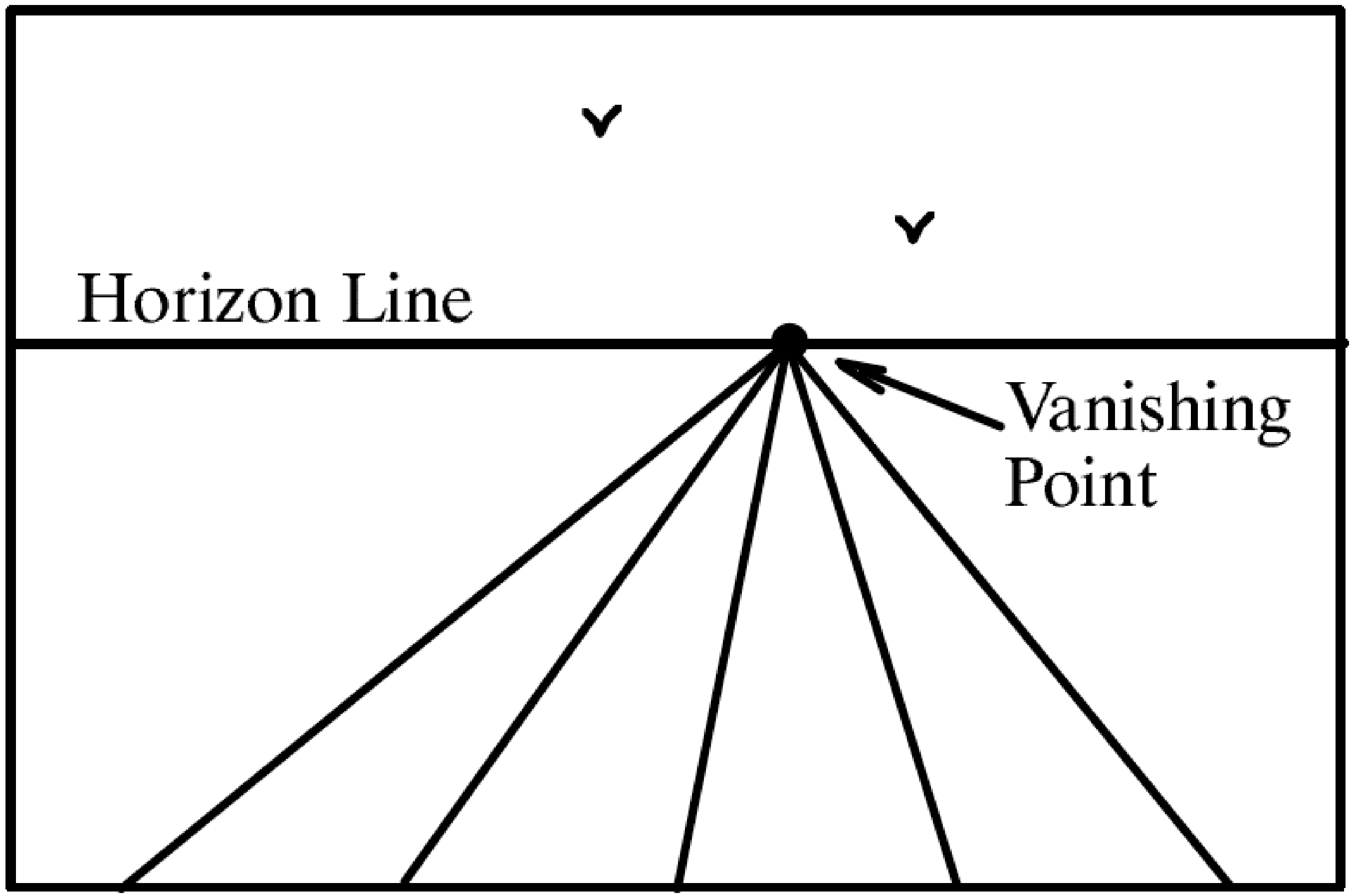


Figure 23.4
A perspective view of a set of parallel lines in the plane. All of the lines converge to a single vanishing point.

消失点: 无穷远的投影

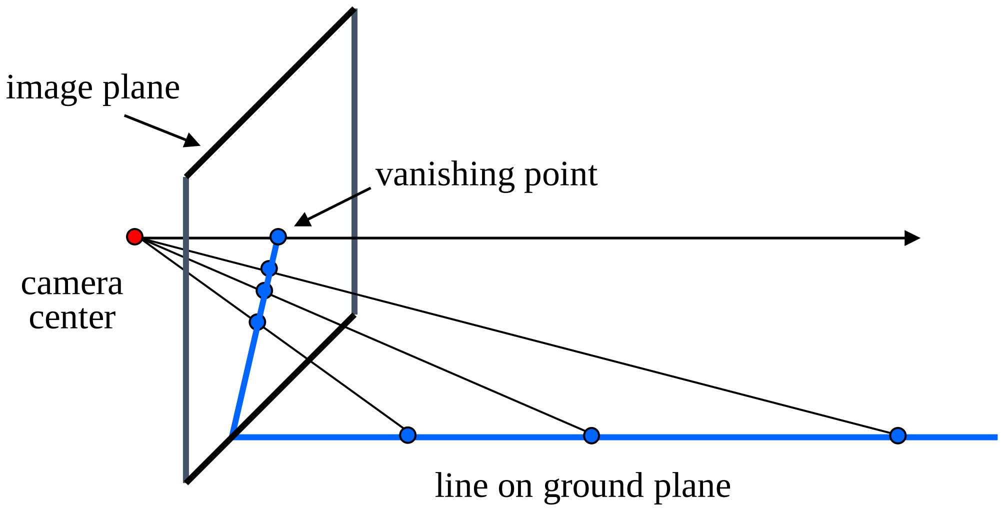
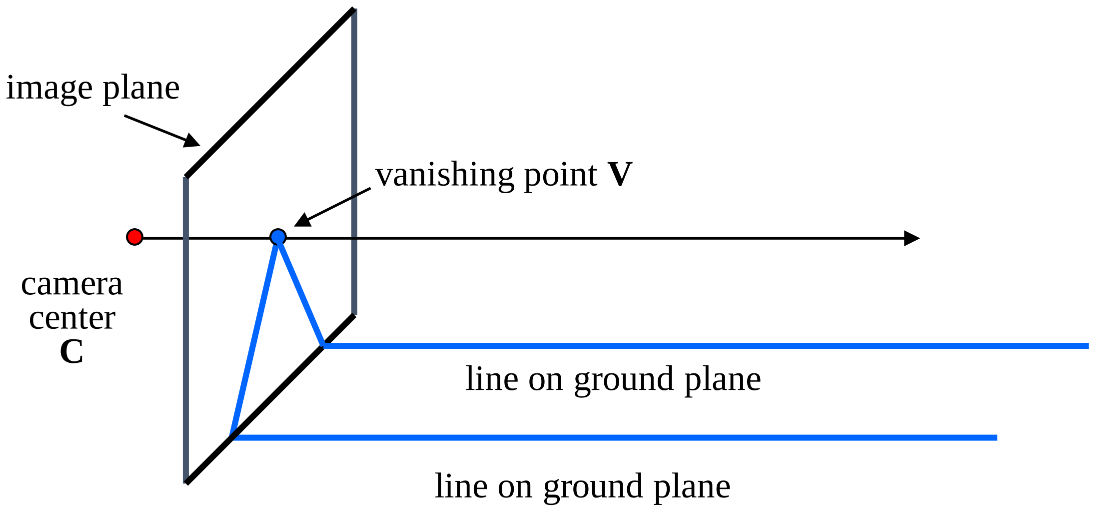


image plane
vanishing point
camera
center
line on ground plane

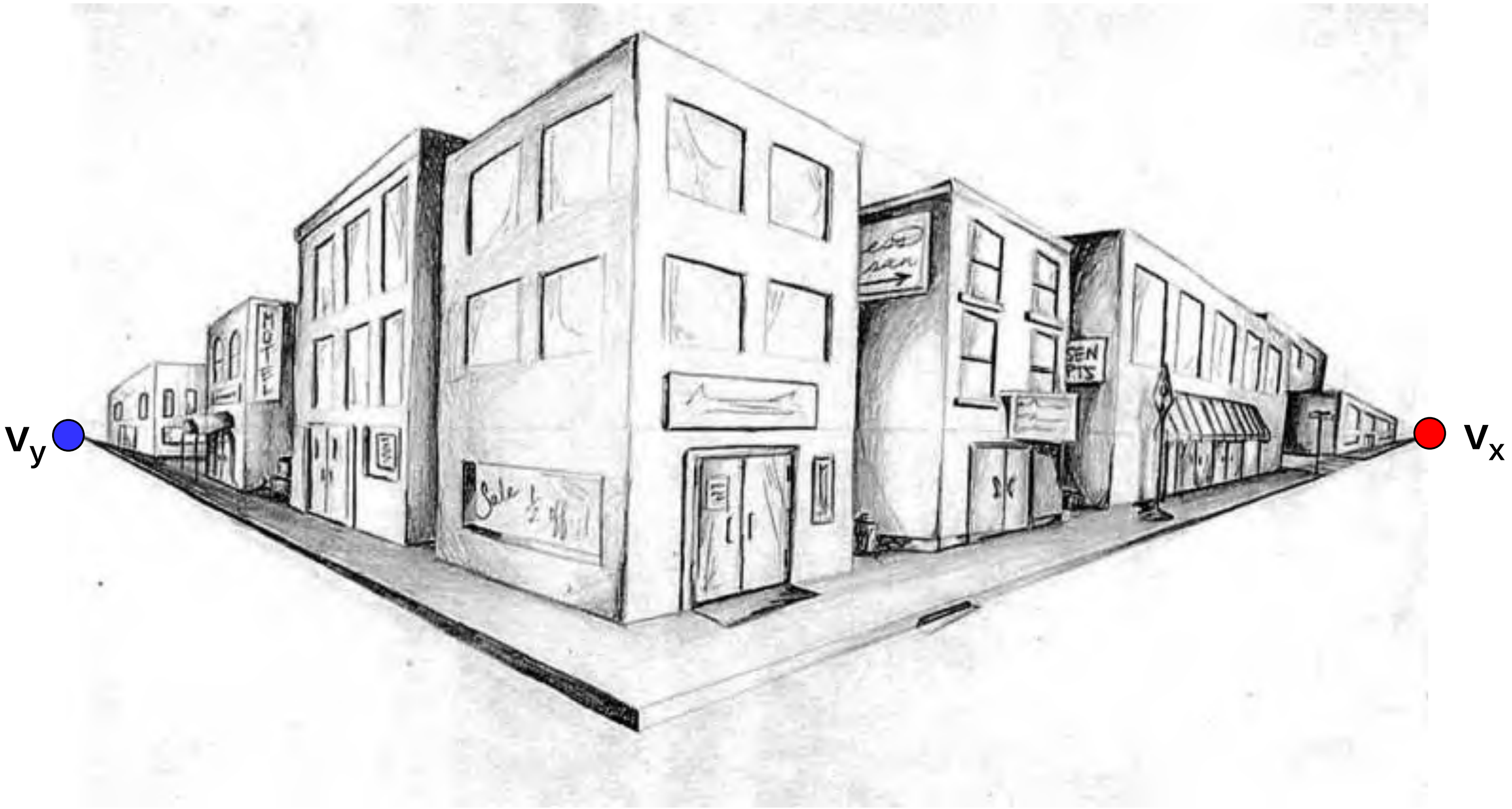
消失点: 地平面与平行线



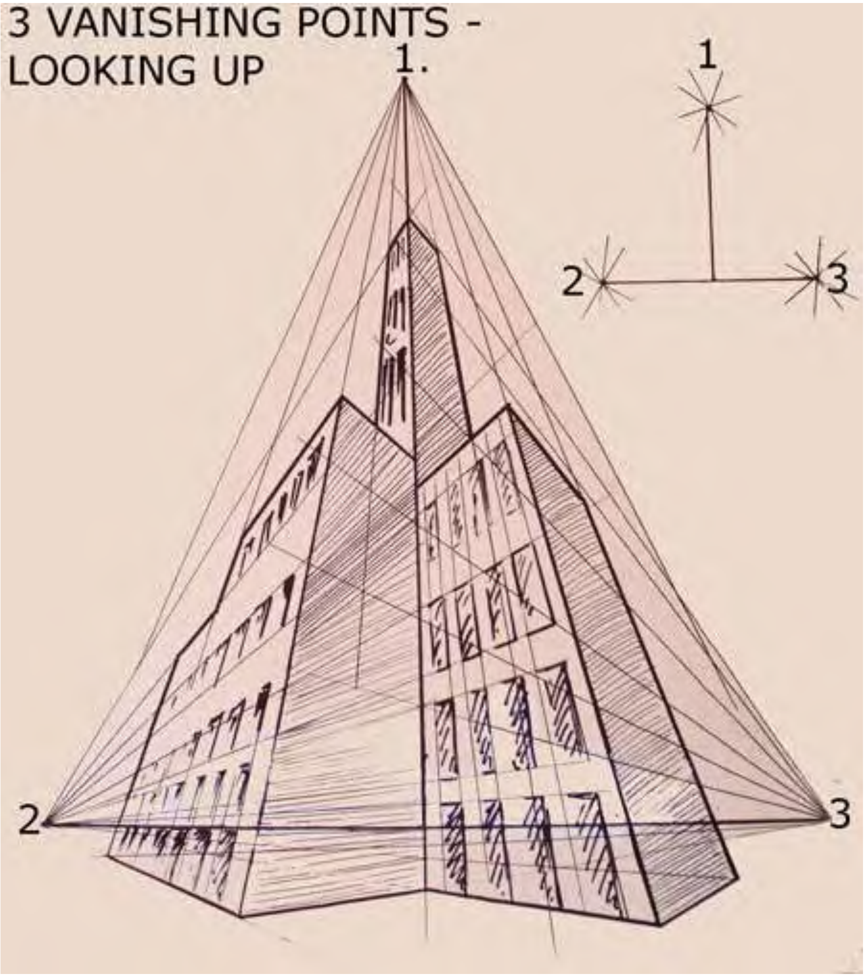
- 性质
  - 地平面中的一组平行线有同一个消失点
  - C 到 v 的射线也与这些线平行
  - 一张图像可能有多于一个消失点
    - 图像上的所有点都有可能是消失点

一个消失点

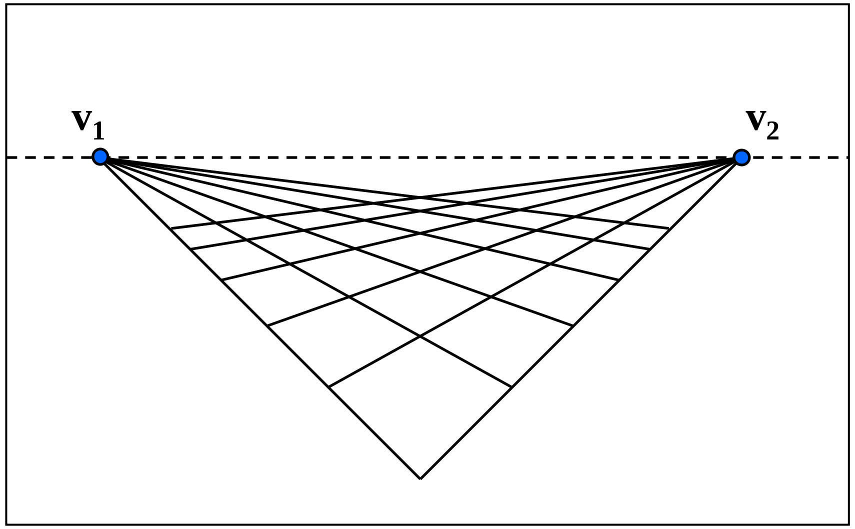
几个消失点?



三个消失点

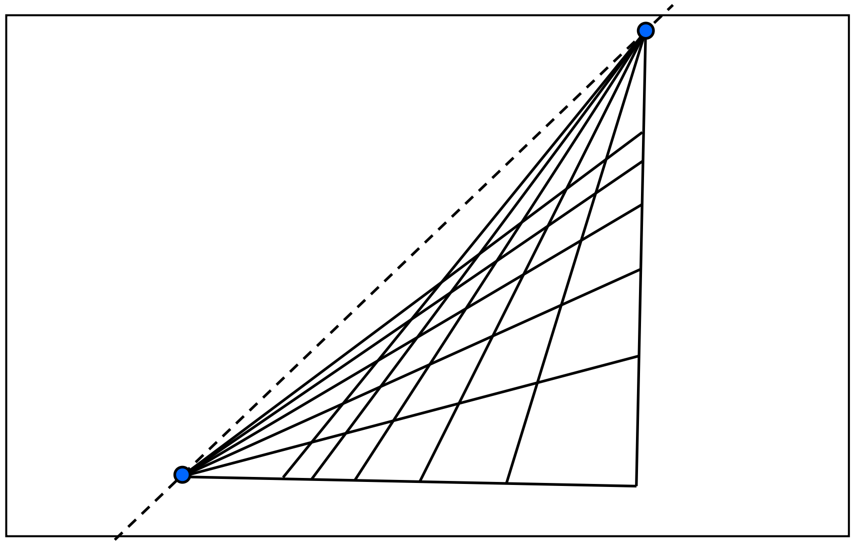


消失线



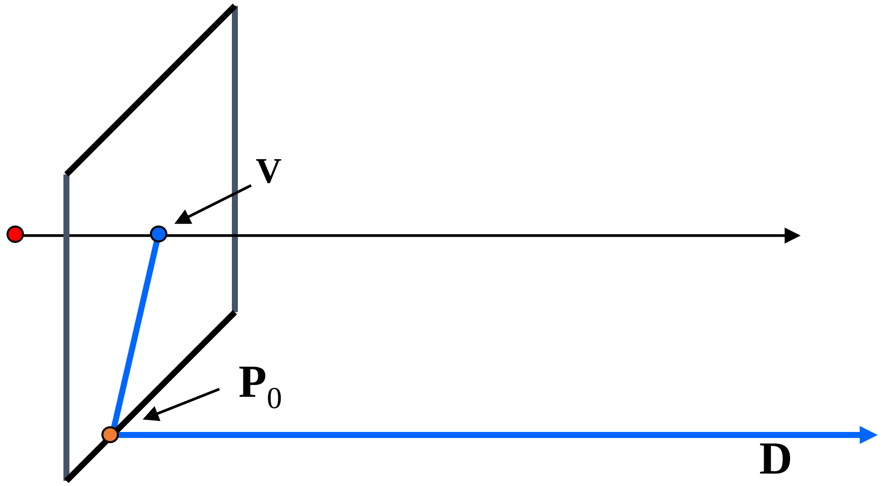
- 消失点的集合
  - 平行线的集合定义一个消失点
  - 所有这些消失点构成一个*水平线*
    - 或者叫*消失线*
  - 不同的平面会对应不同的消失线

消失线



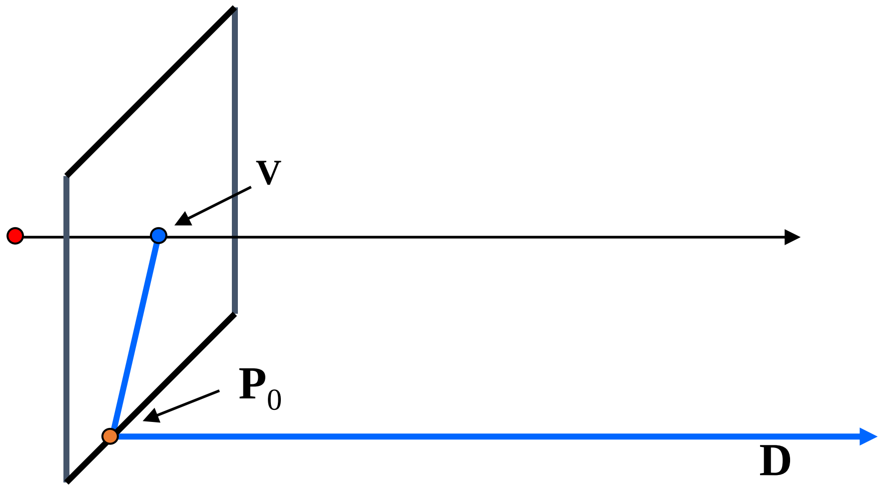
- 消失点的集合
    - 平行线的集合定义一个消失点
    - 所有这些消失点构成一个*水平线*
        - 或者叫*消失线*
    - 不同的平面会对应不同的消失线

计算消失点



$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

计算消失点

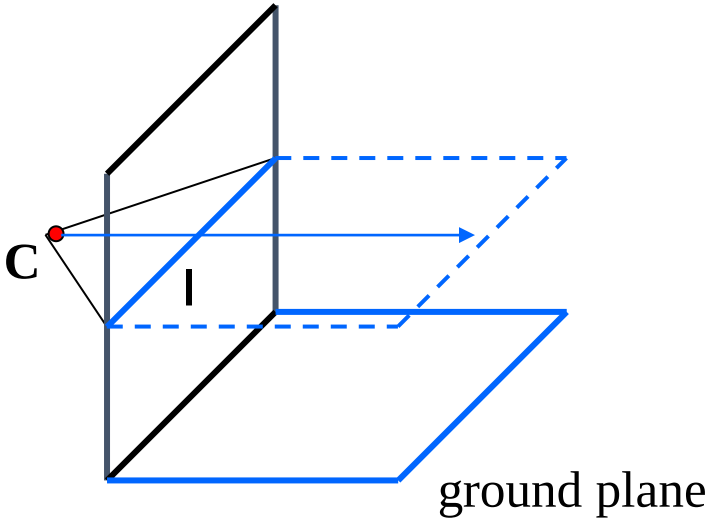


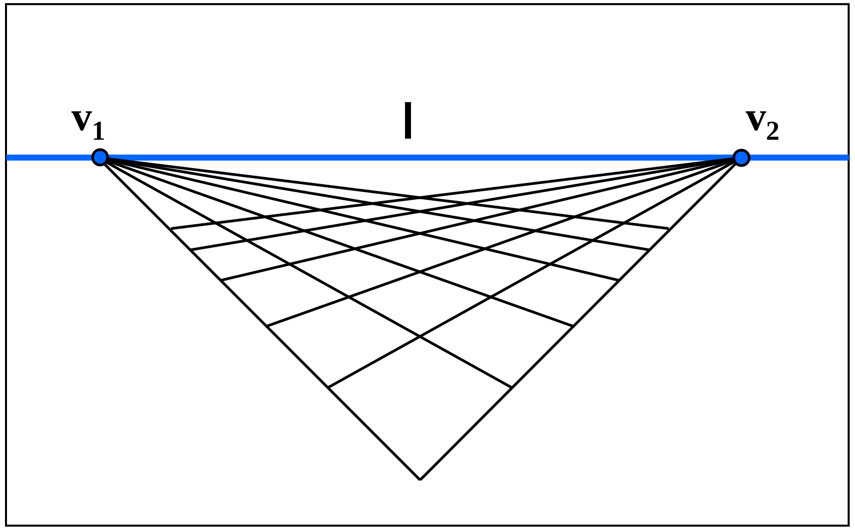
$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X / t + D_X \\ P_Y / t + D_Y \\ P_Z / t + D_Z \\ 1/t \end{bmatrix}$$

- 性质 $\mathbf{v} = \mathbf{\Pi P}_\infty$
  - P_∞ 是在无穷远的点, v 是它的投影
  - 仅仅与射线方向相关
  - 平行线 P_0 + tD, P_1 + tD 投影相交在P_∞

计算消失线

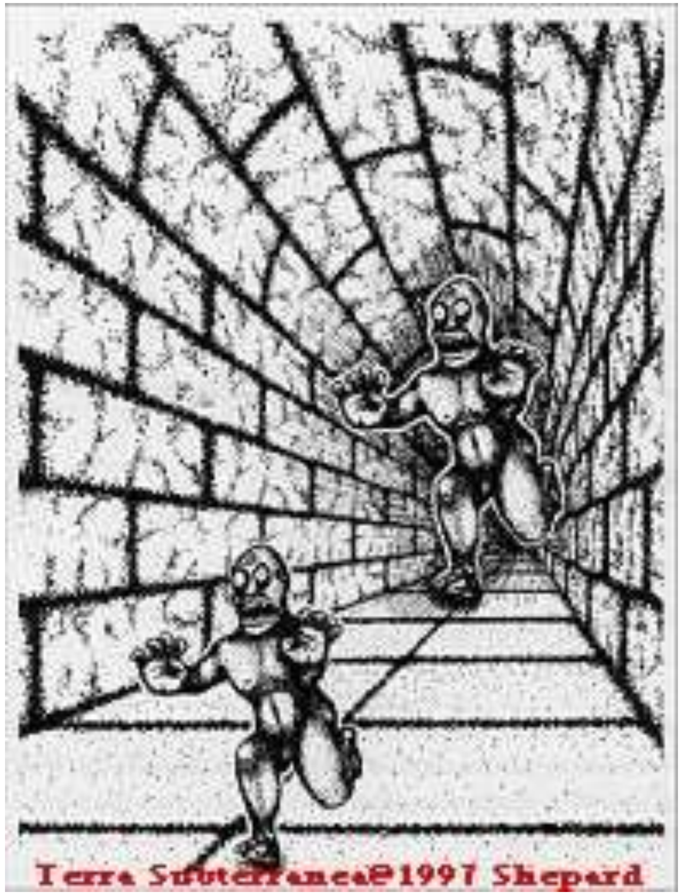


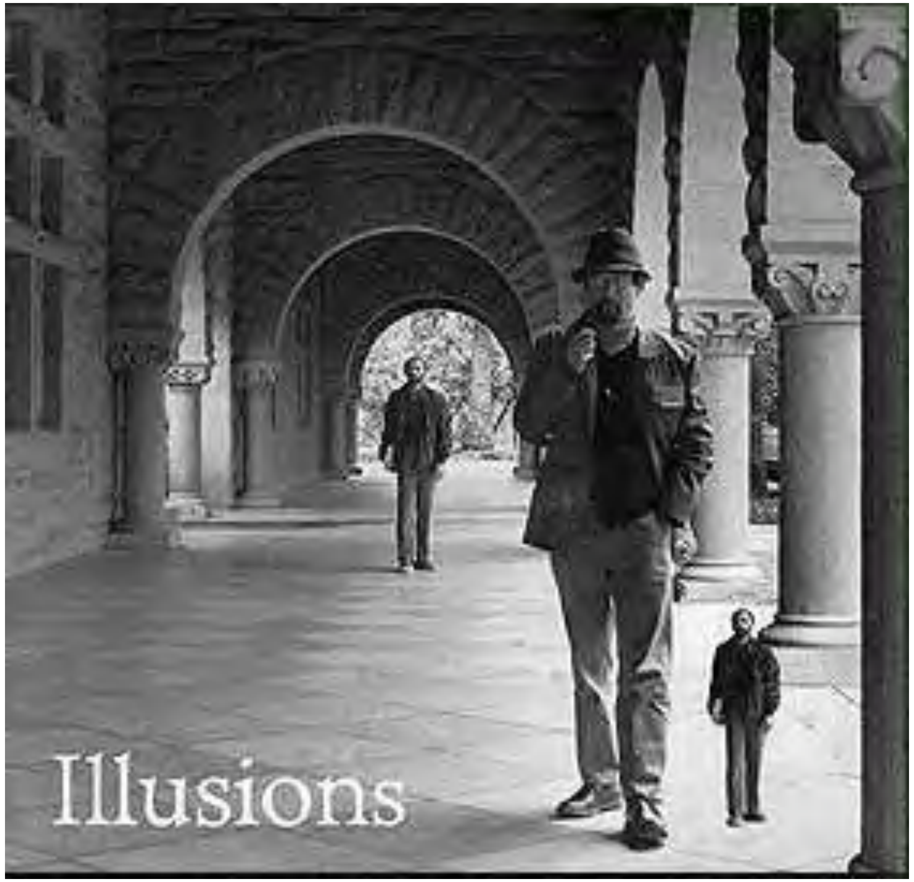


- 性质
  - l 是图像平面与过C的水平面的相交线
  - 可以通过地平面的两组平行线计算l
  - 所有与C 等高的点都会投影到l
    - 高于C 的点会投影在l的上方
  - 为计算物体高度提供了方法!

跳伞者在相机之上还是之下?

错觉又来了

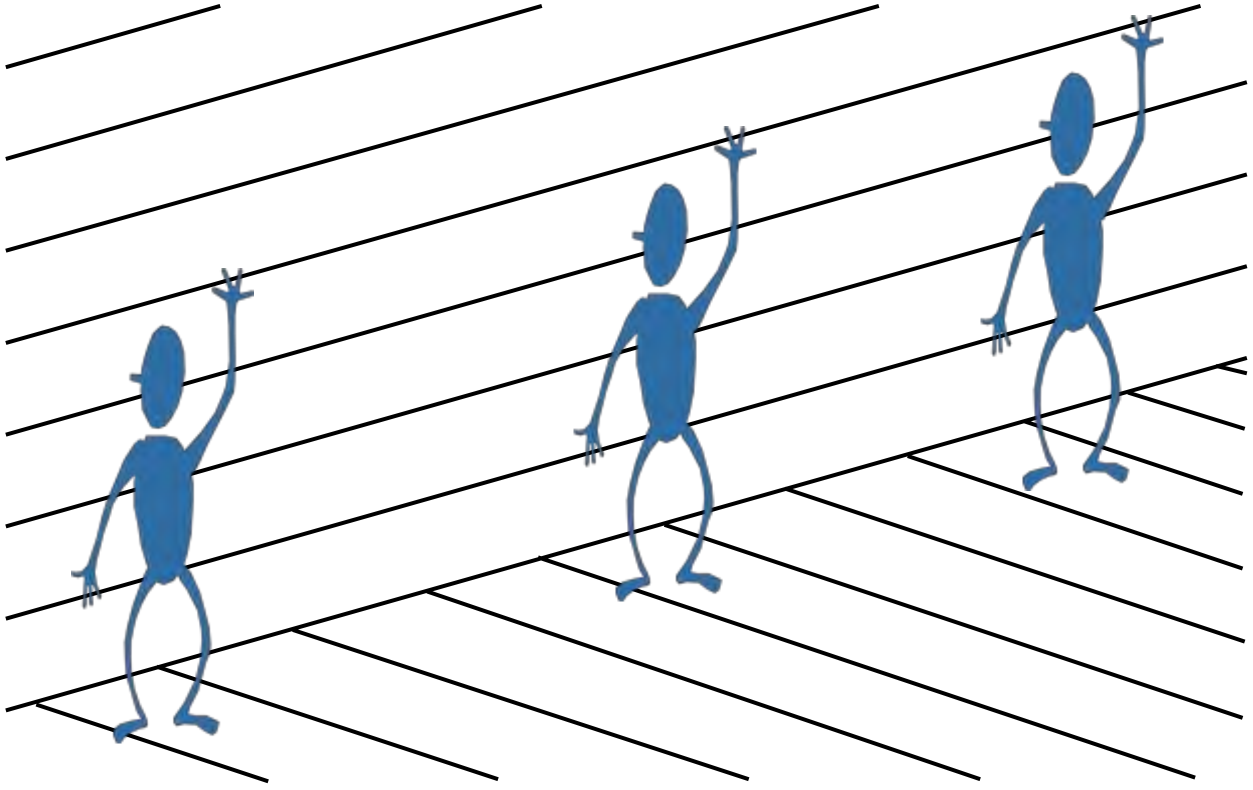




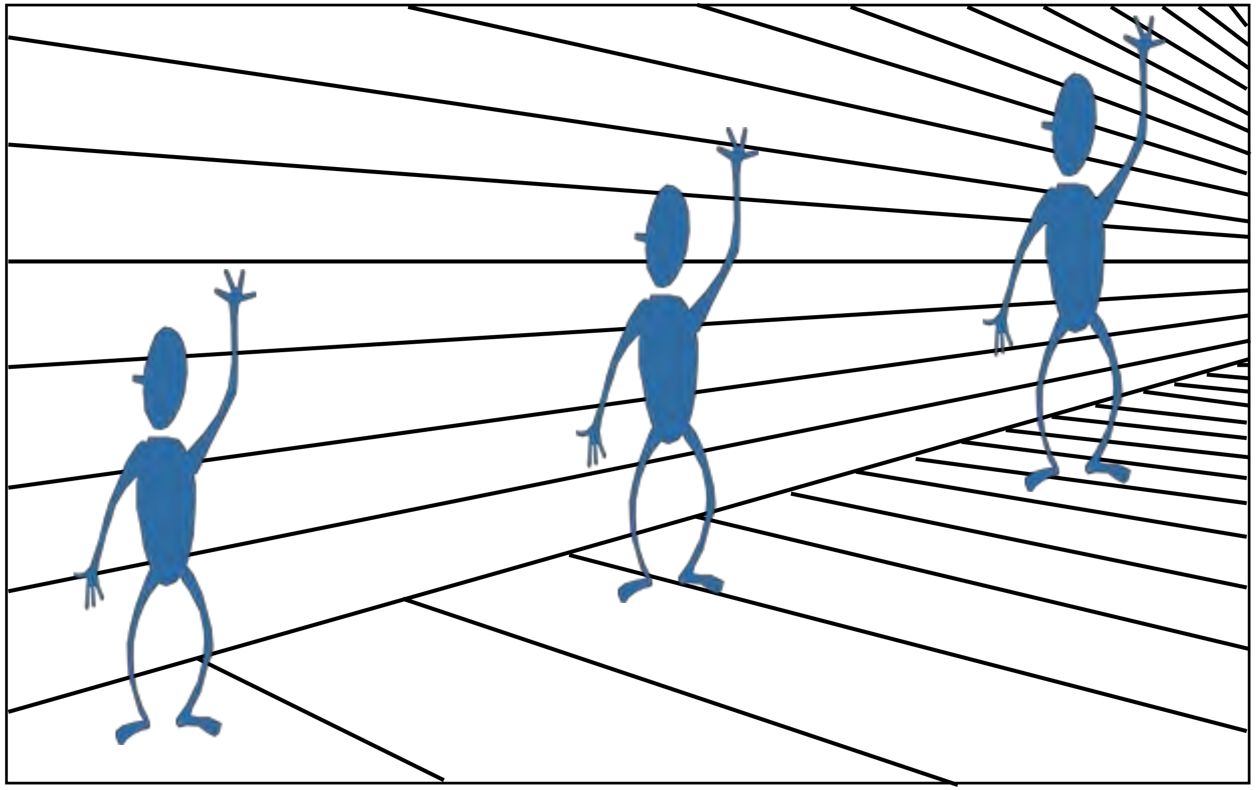
错觉又来了

错觉又来了

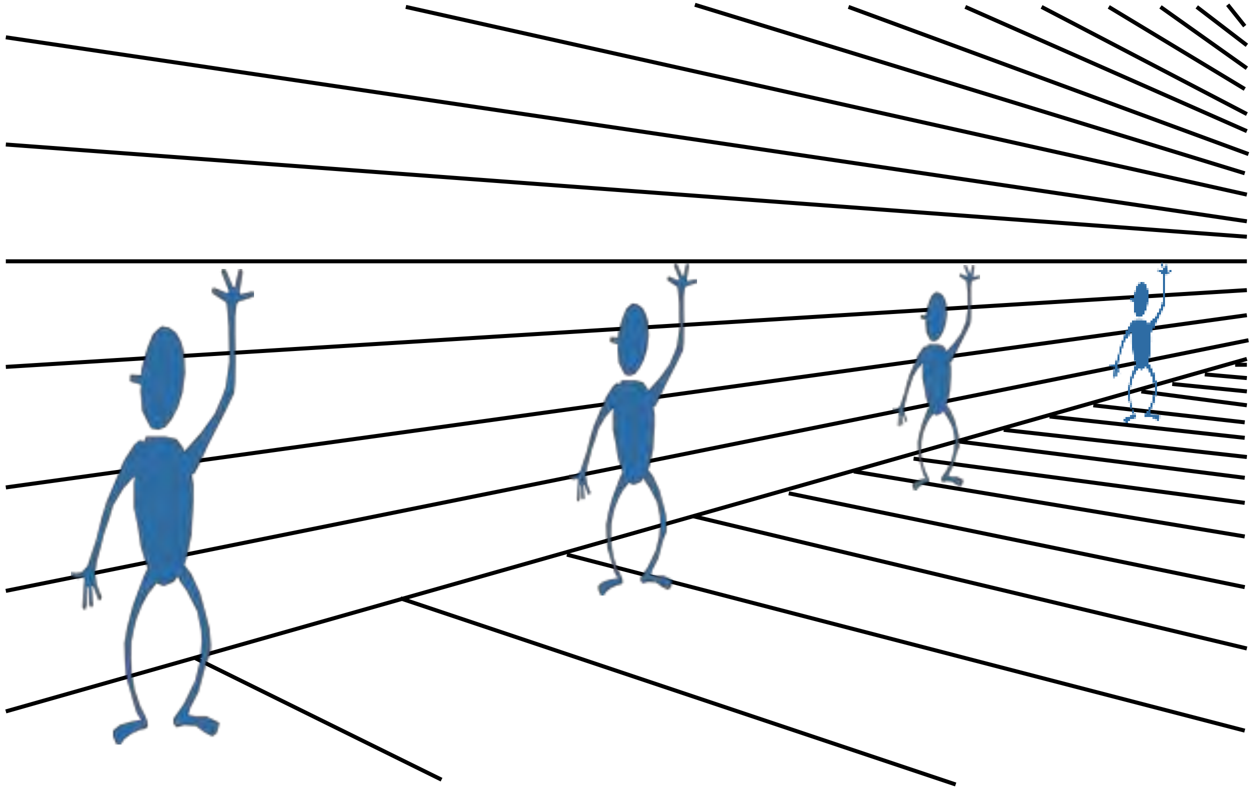
Perspective Cues: 透视线索



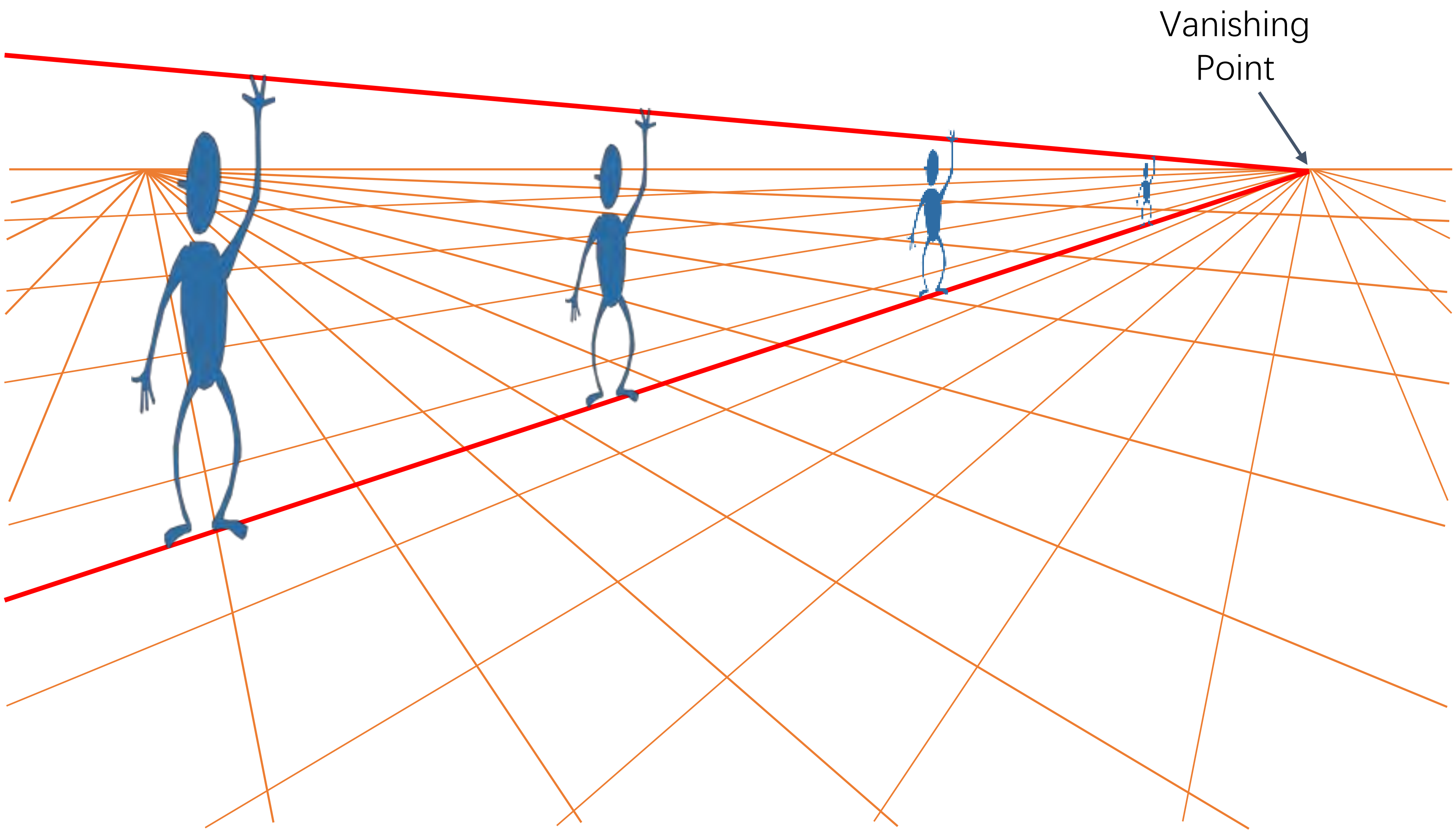
Perspective Cues: 透视线索



Perspective Cues: 透视线索

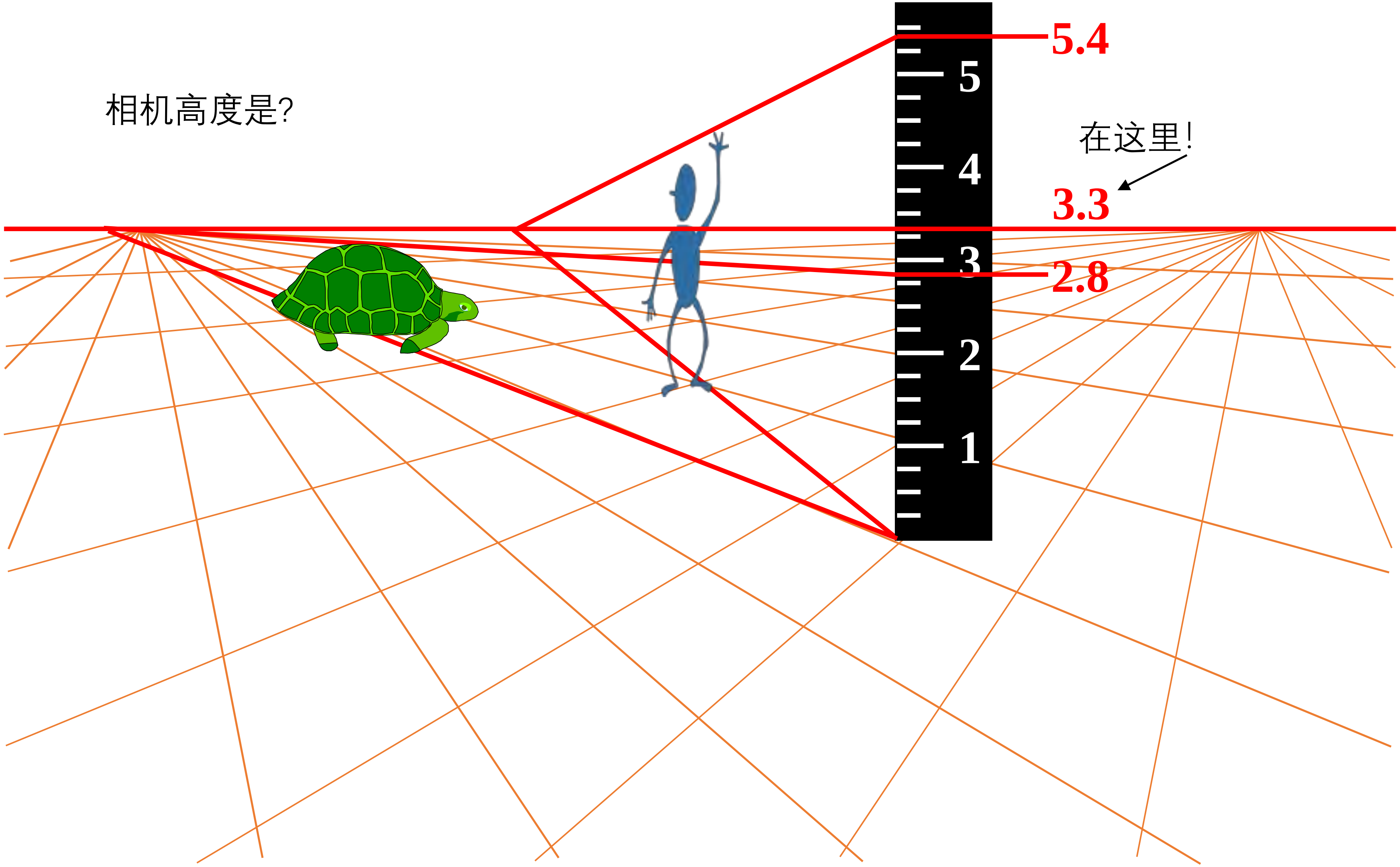


对比高度



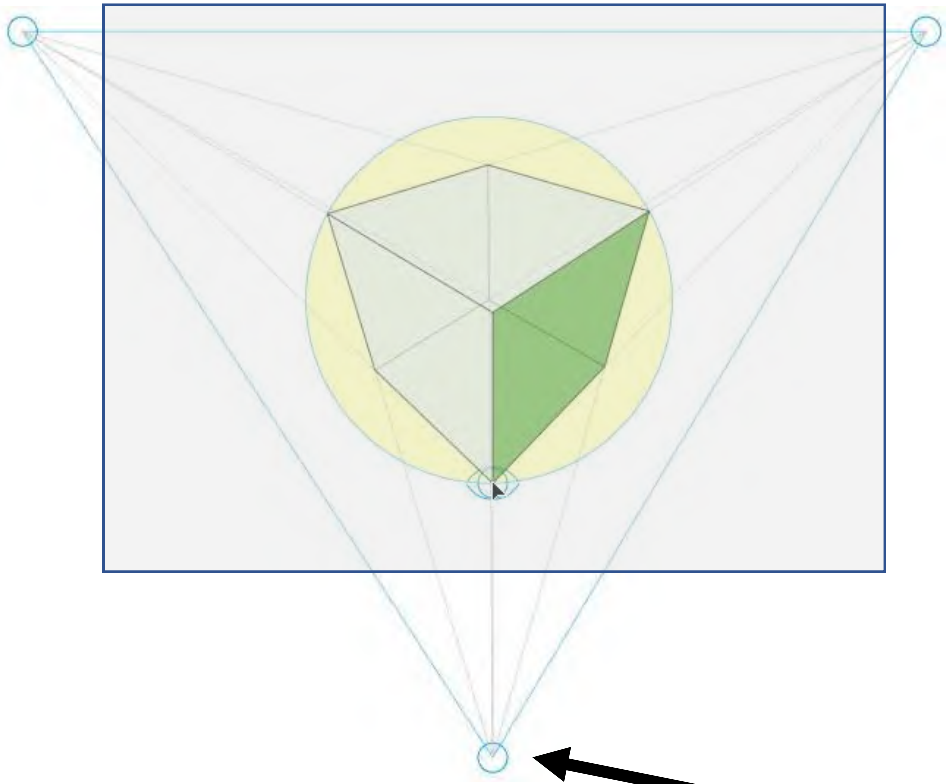
计算高度

相机高度是?



在这里!

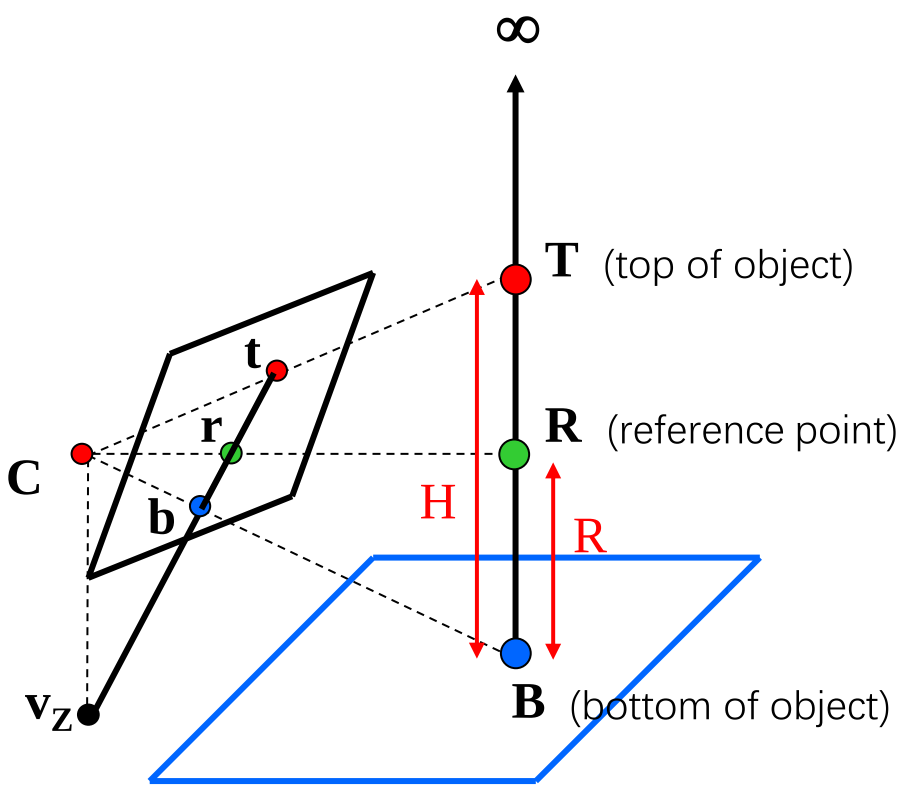
垂直消失点



找到垂直投影的
消失点

计算高度

- 线段长度比例投影以后不变



$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}=\frac{H}{R}$$

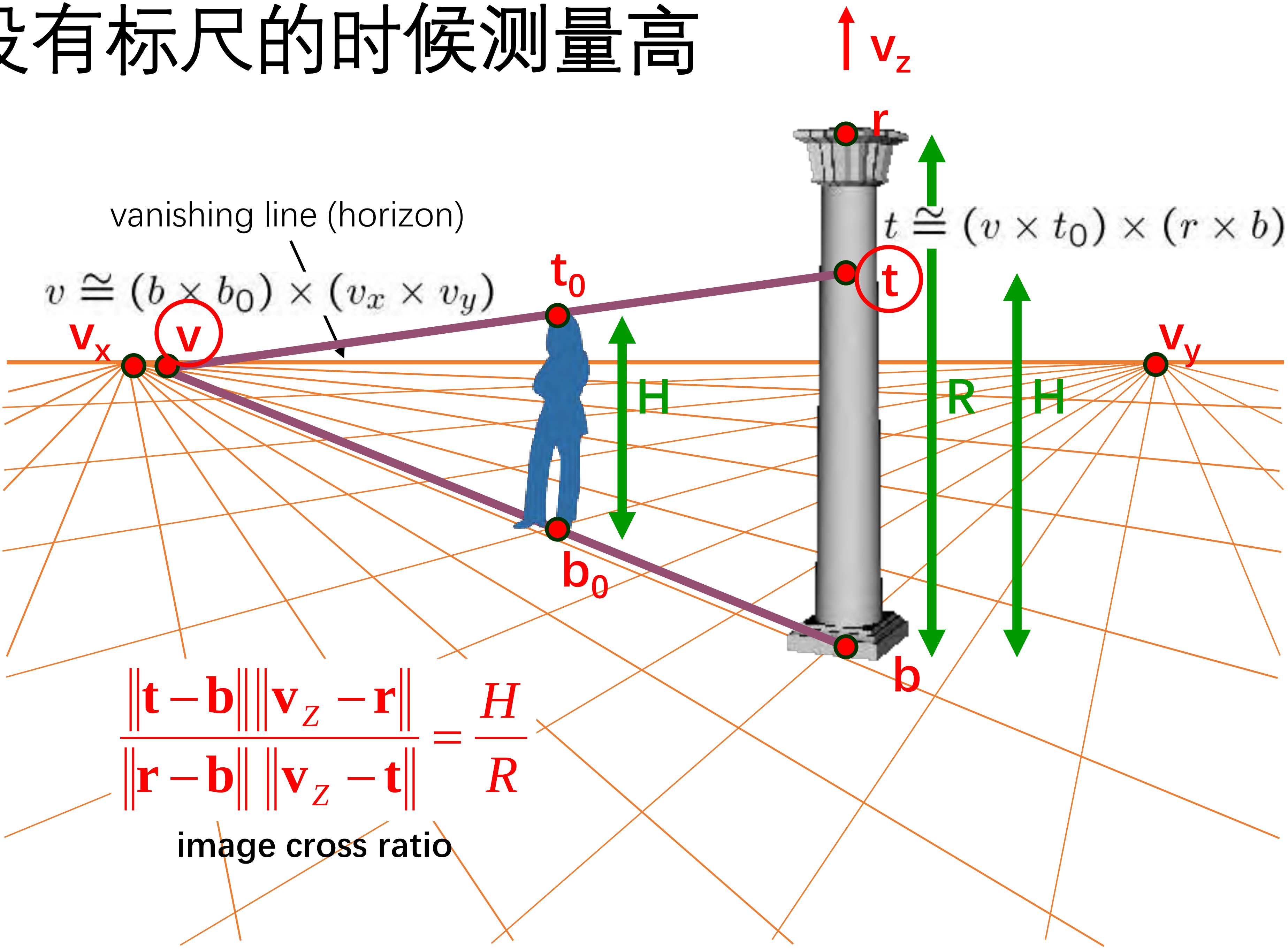
scene cross ratio

$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}=\frac{H}{R}$$

image cross ratio

scene points represented as $\mathbf{P}=\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$ image points as $\mathbf{p}=\begin{bmatrix}x\\y\\1\end{bmatrix}$

在没有标尺的时候测量高度



单张图像的三维重建

St. Jerome in his Study, H. Steenwick

Bringing Pictorial Space to Life: Computer Techniques for the Analysis of Paintings. Antonio Criminisi, Martin Kemp, Andrew Zisserman. 2002.

单张图像的三维重建

单张图像的三维重建

Flagellation, Piero della Francesca

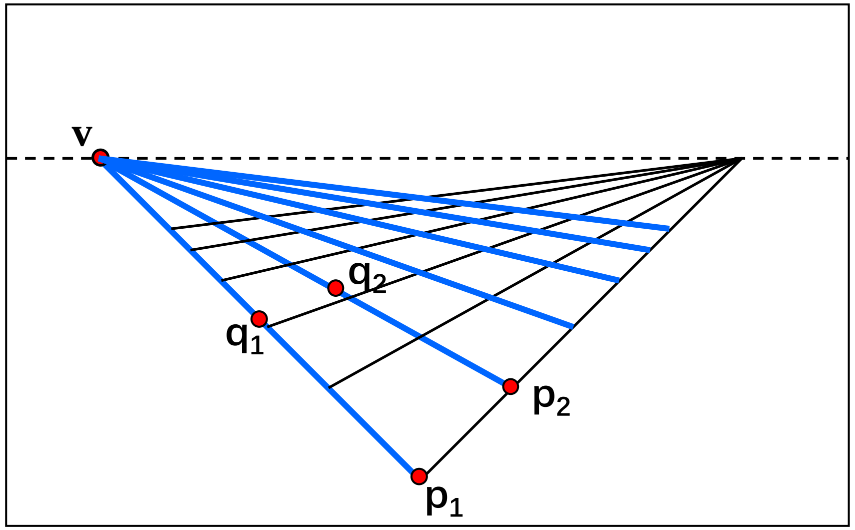
单张图像的三维重建

video by Antonio Criminisi

单张图像的三维重建

Flagellation. Piero della Francesca. c1453.

从线段计算消失点



Intersect p_1q_1 with p_2q_2

$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

怎么找两条平行的线段？怎么知道线段的长度？

相机校准

- 目标：估计相机参数
  - Version 1：解一下方程

$$\mathbf{x} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$

  - Version 2：分别计算相机参数
    - 内参 (焦距，主点，分辨率)
    - 外参 (转角，位置)

相机校准

$$\Pi = \begin{array}{c} \begin{bmatrix} \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} & \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} & \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} & \boxed{\begin{matrix} * \\ * \\ * \end{matrix}} \end{bmatrix} \\ \begin{matrix} \boldsymbol{\pi}_1 & \boldsymbol{\pi}_2 & \boldsymbol{\pi}_3 & \boldsymbol{\pi}_4 \end{matrix} \end{array} = \begin{bmatrix} \boldsymbol{\pi}_1 & \boldsymbol{\pi}_2 & \boldsymbol{\pi}_3 & \boldsymbol{\pi}_4 \end{bmatrix}$$

- $\boldsymbol{\pi}_1 = \mathbf{\Pi}\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T = \mathbf{v}_X$ (X vanishing point)
- similarly, $\boldsymbol{\pi}_2 = \mathbf{v}_Y$, $\boldsymbol{\pi}_3 = \mathbf{v}_Z$
- $\boldsymbol{\pi}_4 = \mathbf{\Pi}\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T = \text{projection of world origin}$

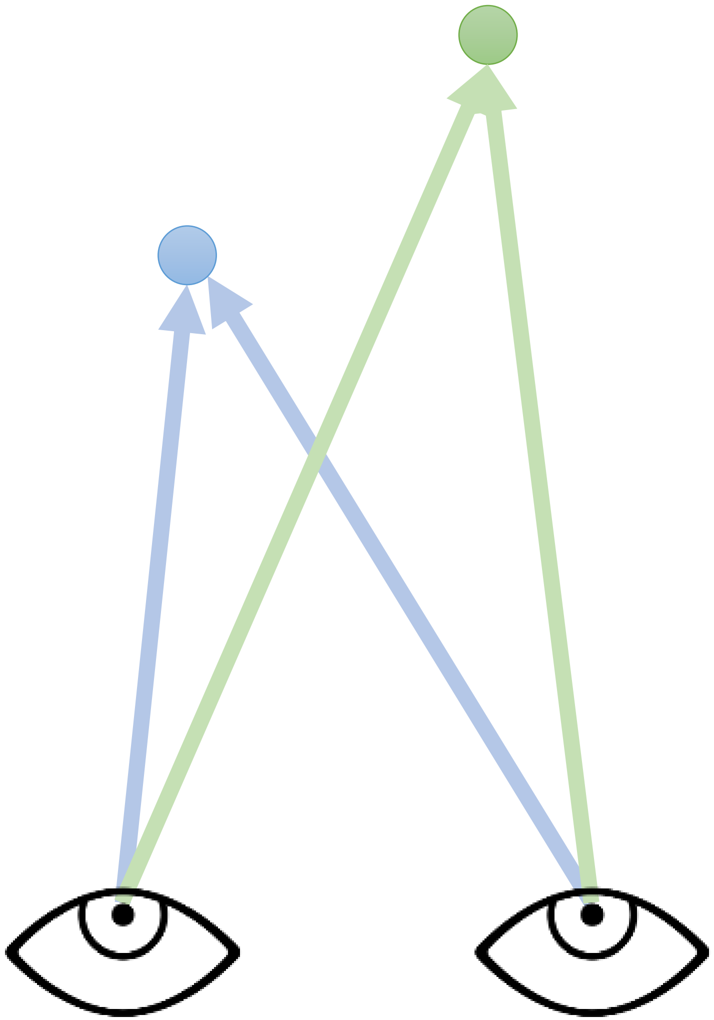
$$\mathbf{\Pi} = \begin{bmatrix} \mathbf{v}_X & \mathbf{v}_Y & \mathbf{v}_Z & \mathbf{o} \end{bmatrix}$$

AR 增强现实

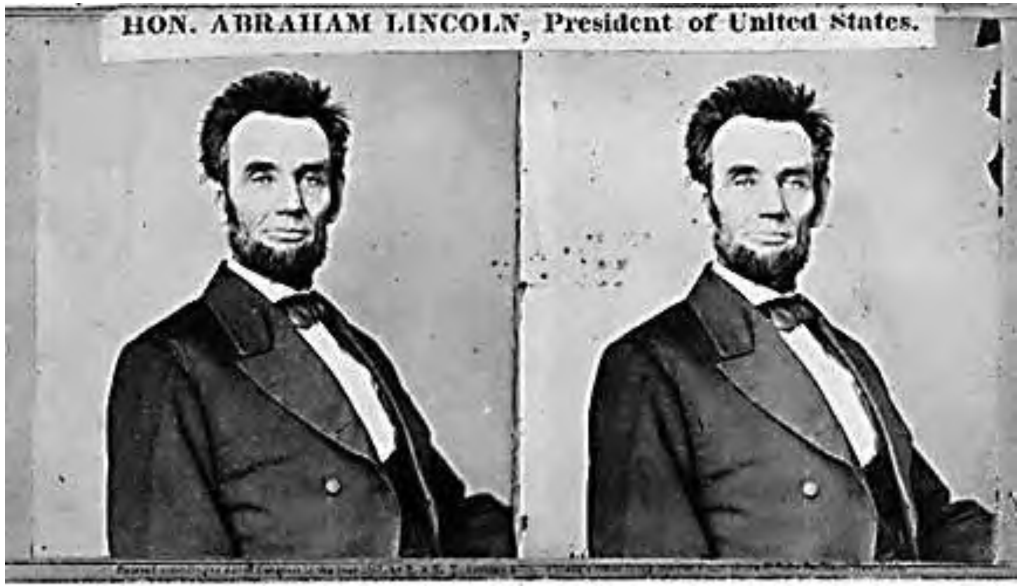
ArUco

双目视图：测距

- 在知道相机参数的情况下，若不借助额外工具，怎样测距？
- 使用双目照相机！



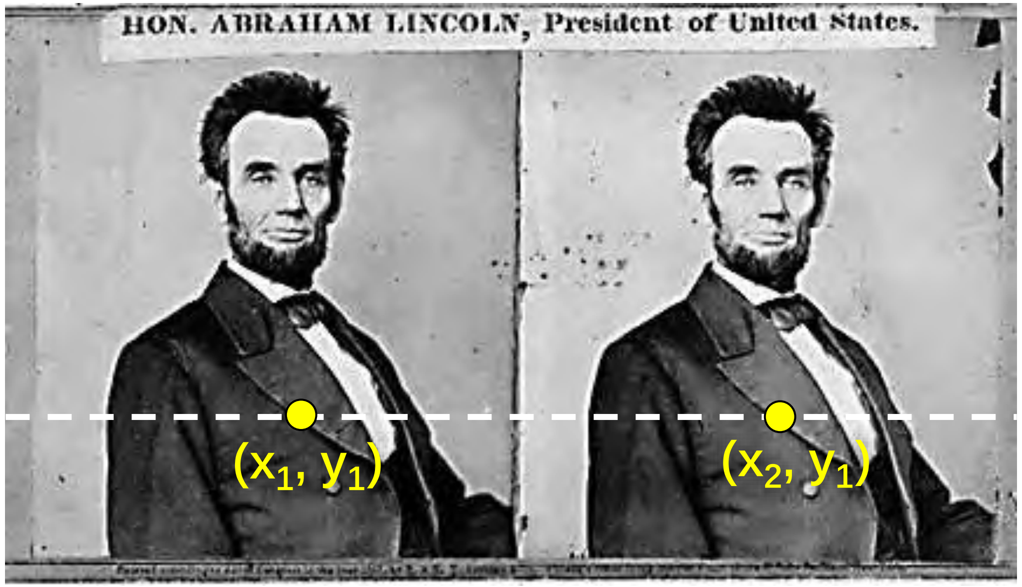
双目视图



- 给出两张不同视角拍摄的同一个物体
  - 我们怎么计算每个点的景深呢？
  - 基于同一个点在两张图中移动了多远！

Epipolar Geometry：极线几何

epipolar
lines



极线：在两个相机图像之间的几何关系。
每个像素在一个图像上都有对应的一条极线在另一个图像上。

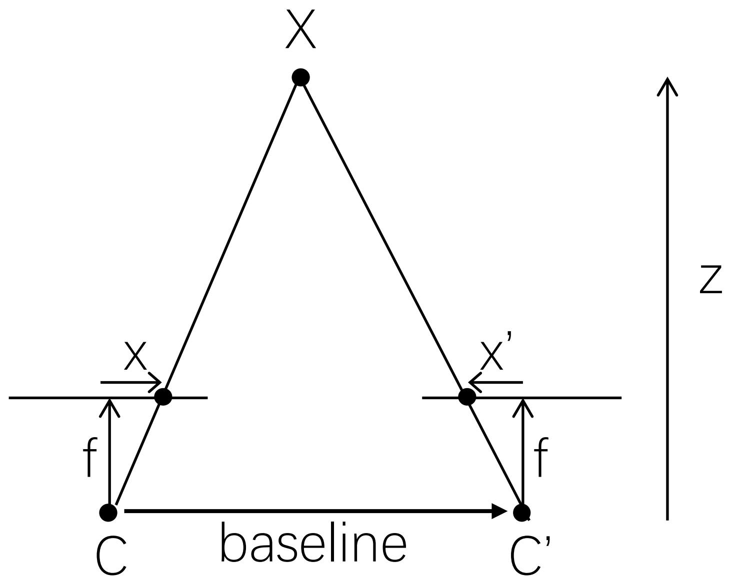
x_2 - x_1 = the *disparity* of pixel (x_1, y_1)

Disparity = inverse depth

http://stereo.nypl.org/view/41729

(Or, hold a finger in front of your face and wink each eye in succession.)

Depth from disparity：通过差异计算景深



$$disparity = x - x' = \frac{baseline * f}{z}$$

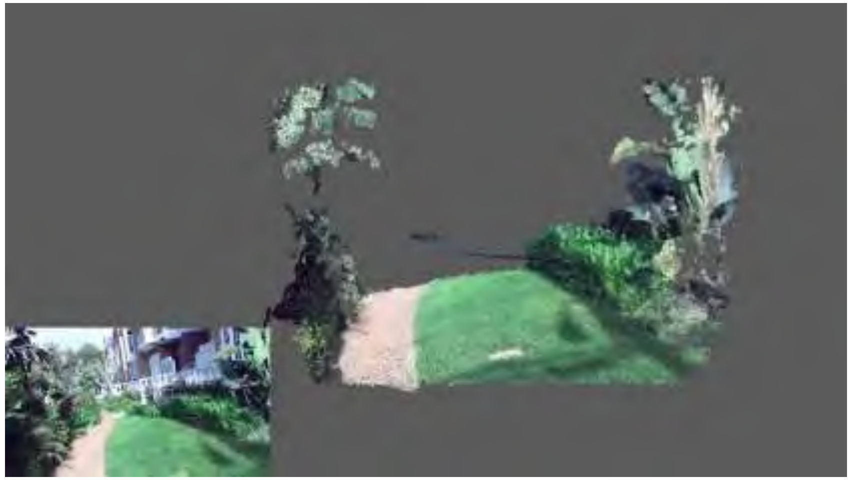
实时双目设备

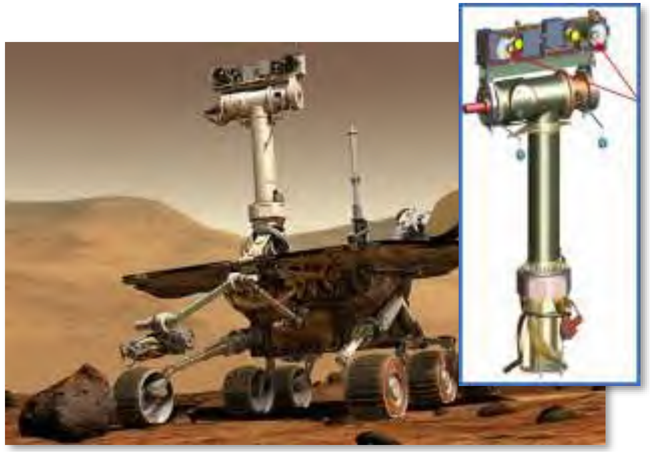
Nomad robot searches for meteorites in Antartica

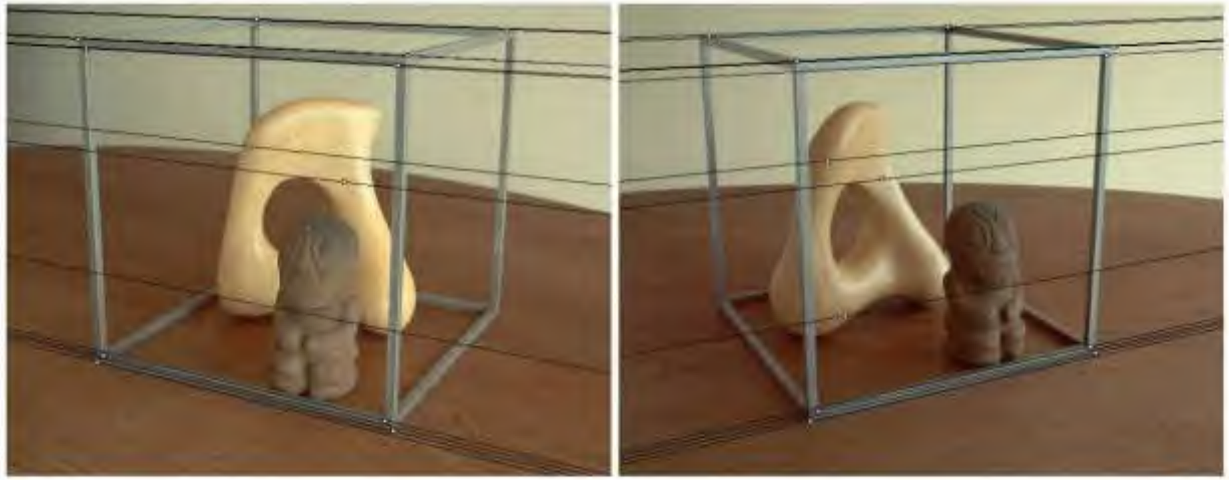
- 机器人巡航

目标：对世界进行三维建模

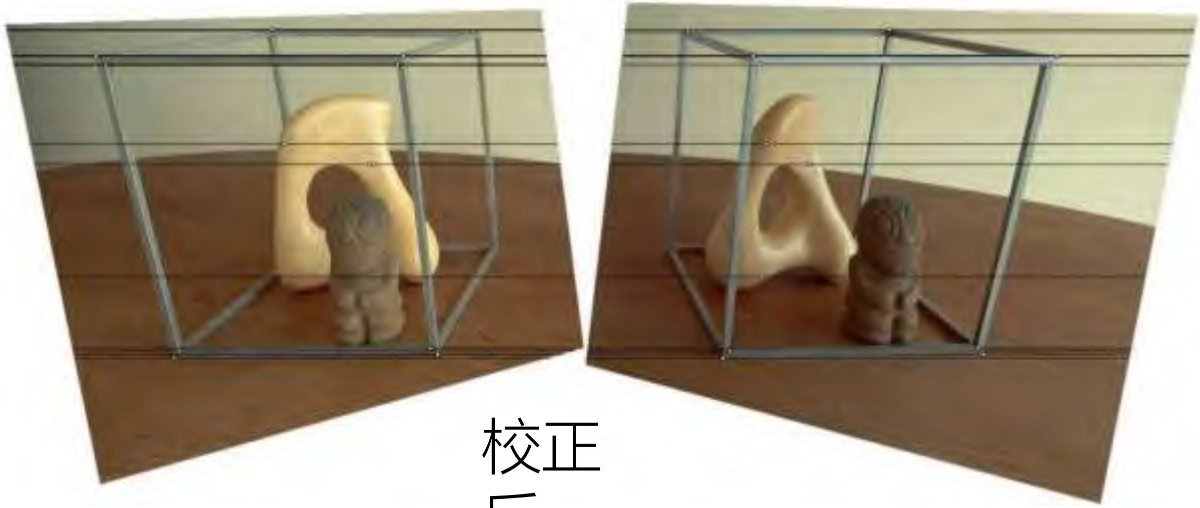
ZED 2i
Camera







原始镜头效果

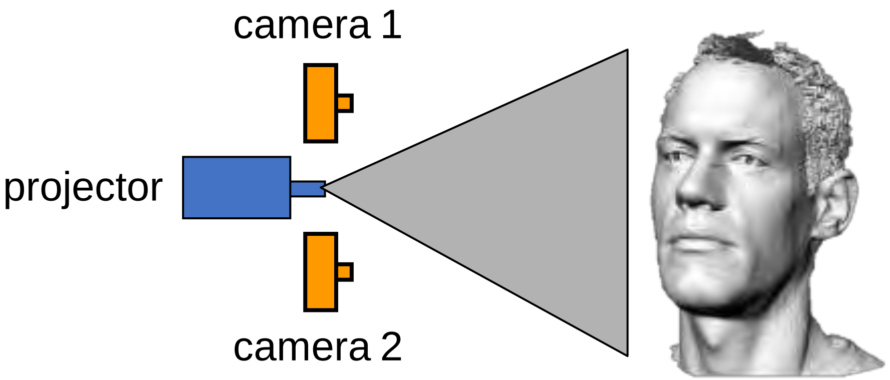


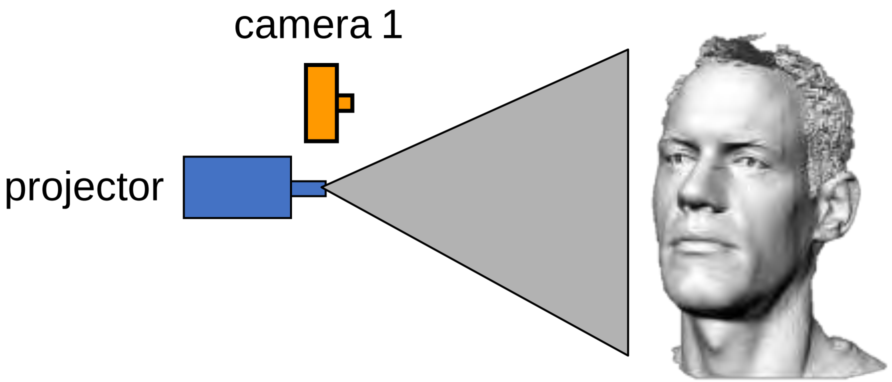
校正后

结构光

1.物体反应： 说明物体如何反应投射的光图案。光图案在物体表面上会发生形变，这个形变取决于物体表面的形状和距离。
2.相机捕捉： 提到一个或多个相机用于捕捉投射到物体上的光图案，记录光图案的形变和变化。
3.深度计算： 解释如何根据捕捉到的图像与投射图案的变化来计算物体表面上不同点的深度信息。这通常需要计算图案的位移或形变。

Li Zhang’s one-shot stereo





- 投射已知的、结构化的光图案到物体表面
  - 通过观察物体对光图案的扭曲或变形来推断物体的三维结构
  - Kinect， iPhone X (using IR)等设备测深度的手段

结构光

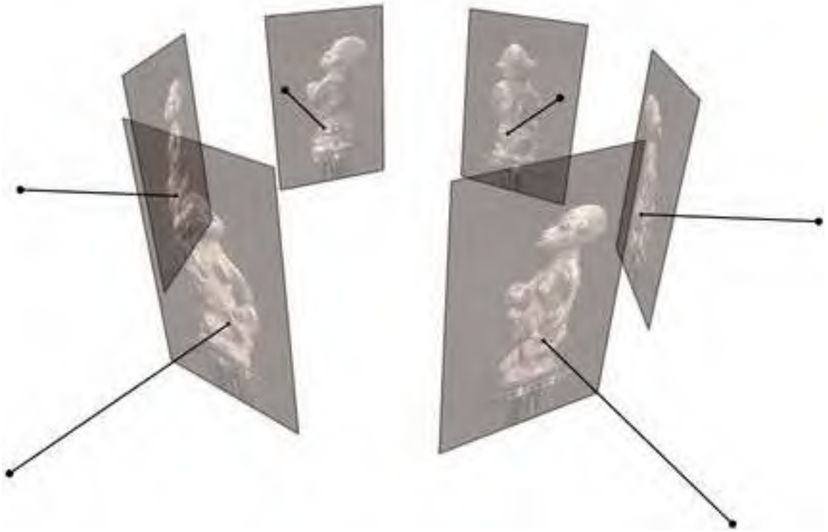
https://ios.gadgethacks.com/news/watch-iphone-xs-30k-ir-dots-scan-your-face-0180944/

多视角立体视图

Input: （已知相机参数下的）多视角图像

Output: 3D模型

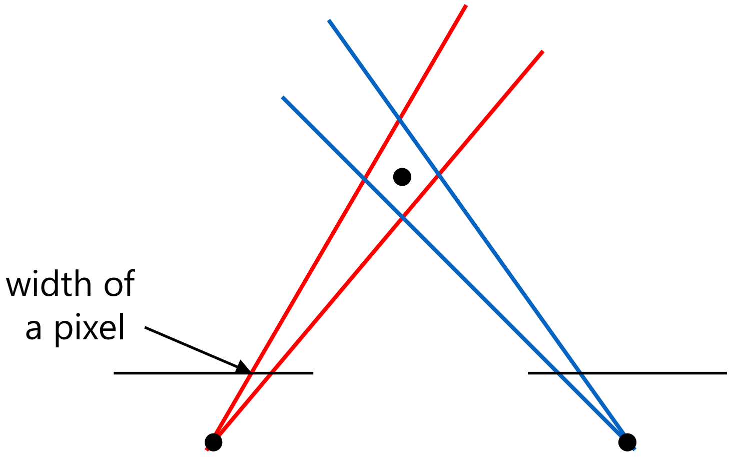
多视角相对双目更全面、更精准



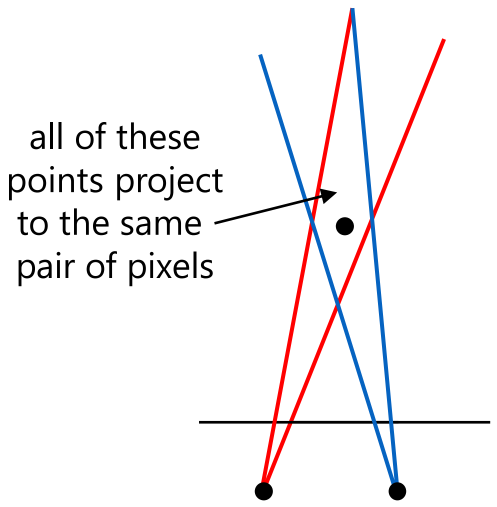
Binocular Stereo

Multi-view stereo

双目视图 vs 多视角视图



Large Baseline

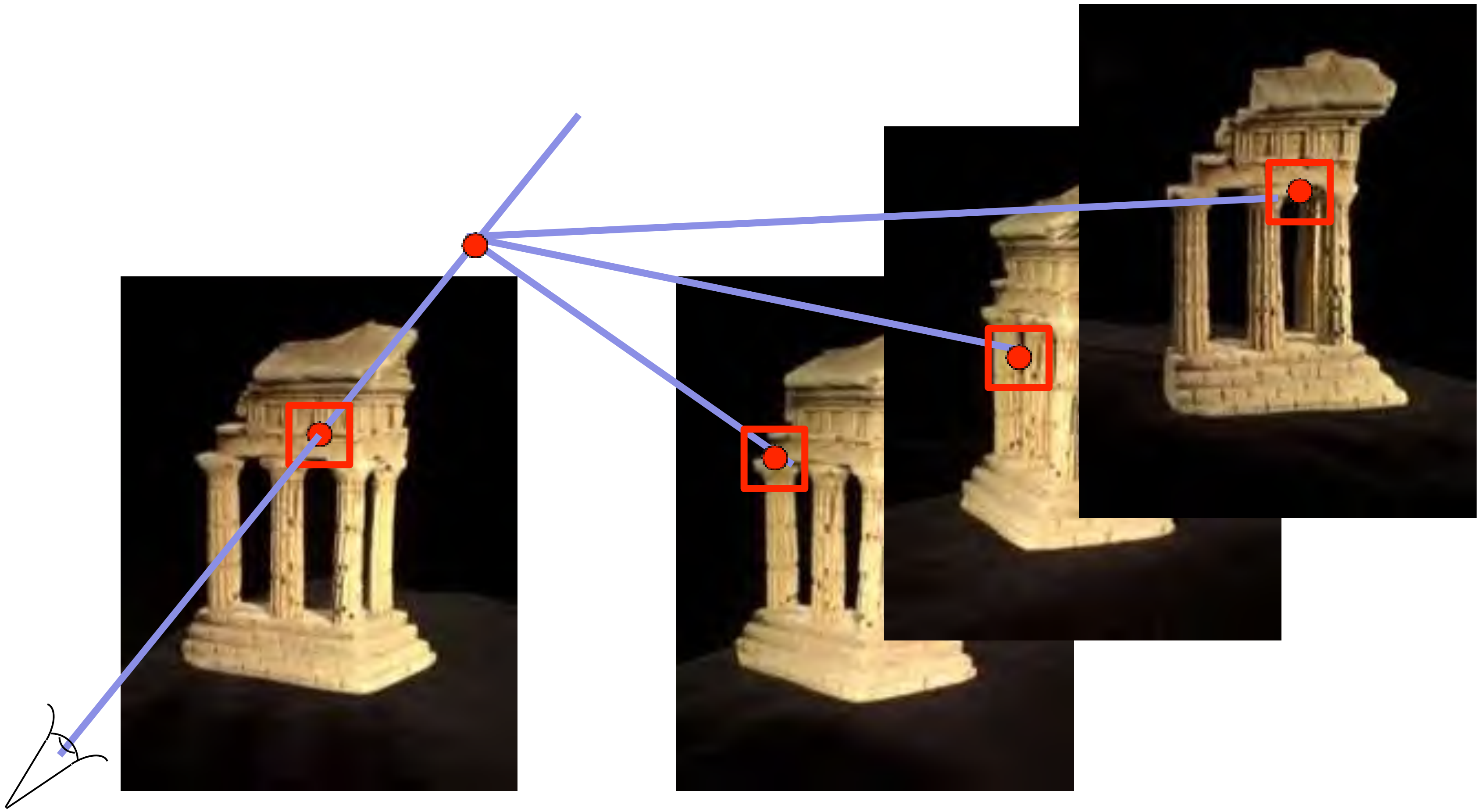


Small Baseline

哪种baseline比较好?

- Too small：相差小，精度较低
- Too large：难以找到匹配的点

多视角视图

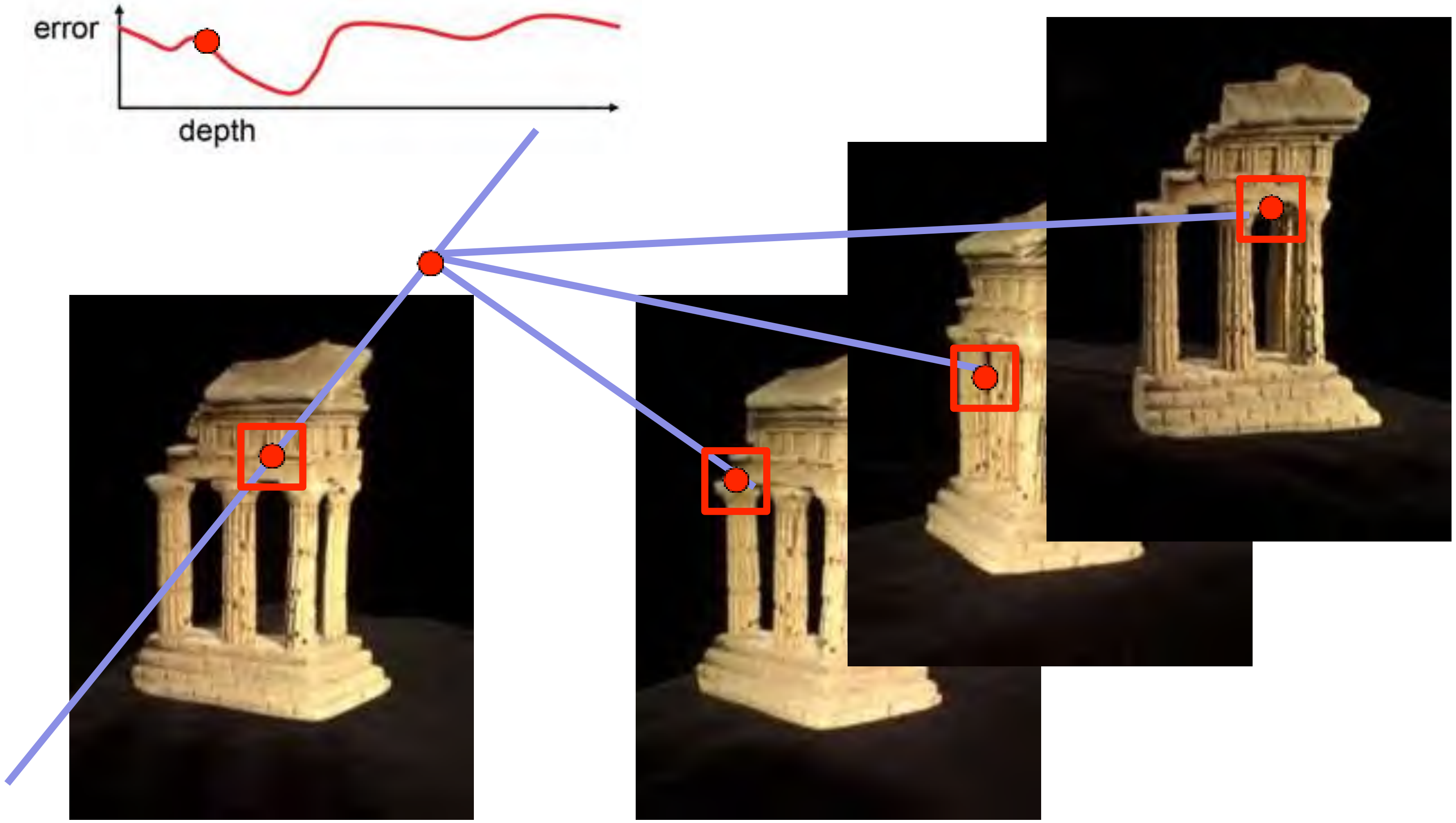


reference view

neighbor views

Source: Y. Furukawa

多视角视图

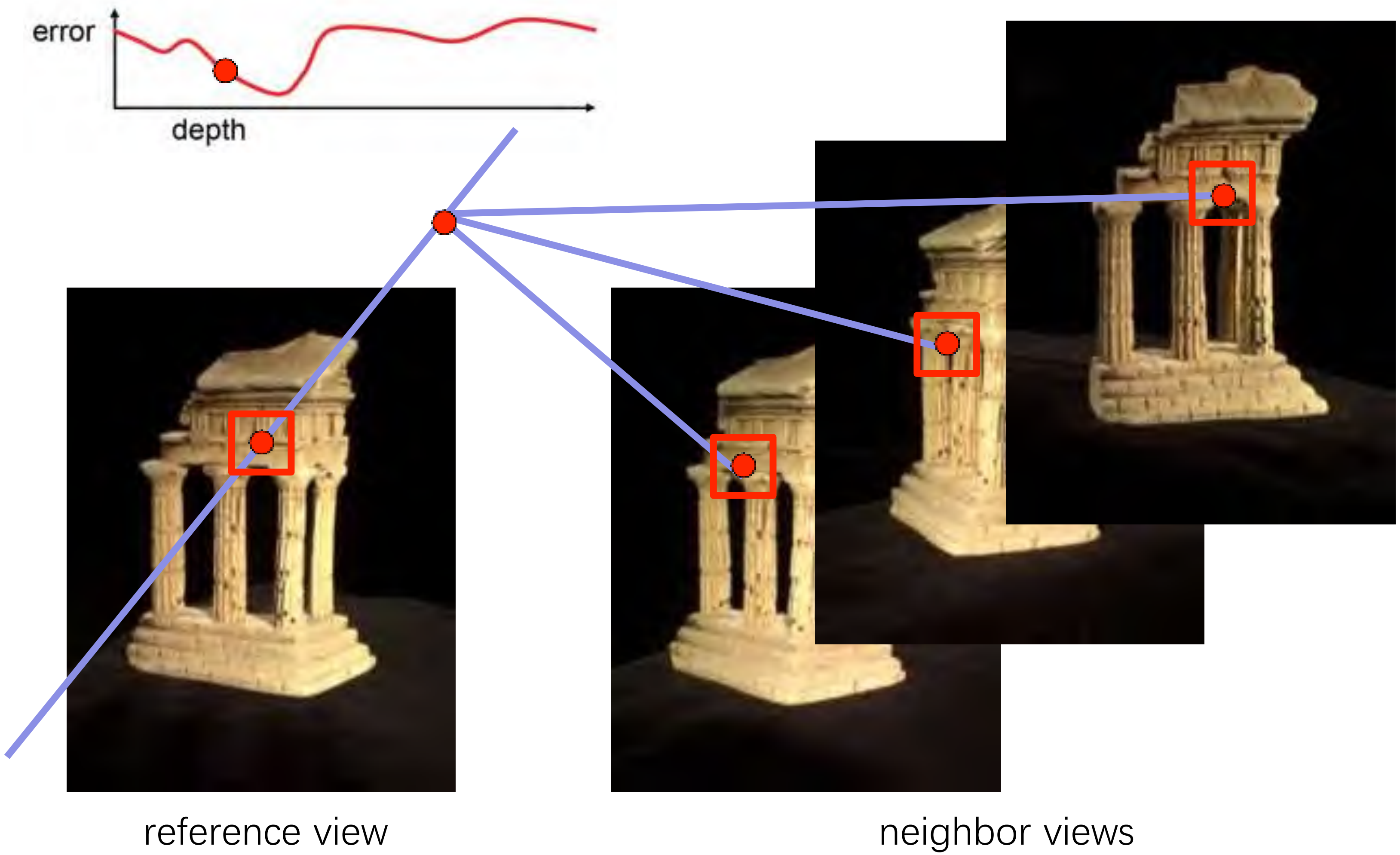


reference view

neighbor views

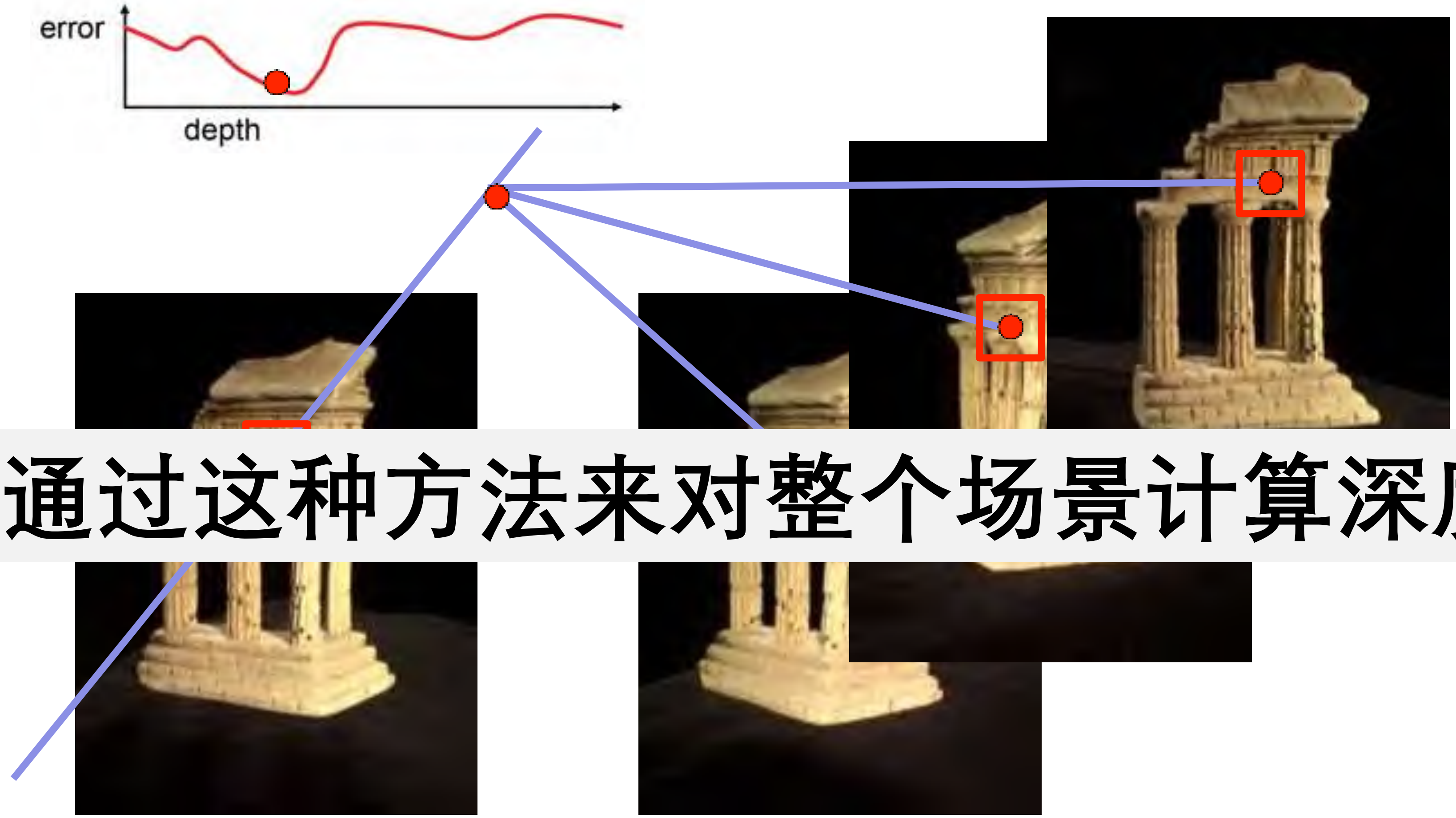
Source: Y. Furukawa

多视角视图



Source: Y. Furukawa

多视角视图



reference view　　neighbor views

通过这种方法来对整个场景计算深度

Source: Y. Furukawa


Google earth

Virtual Reality Video

Anderson, et al. *Jump: Virtual Reality Video*. SIGGRAPH Asia 2016.

Broxton, et al. *Immersive Light Field Video w a Layered Mesh Representation*. SIGGRAPH 2020.

拓展：形状

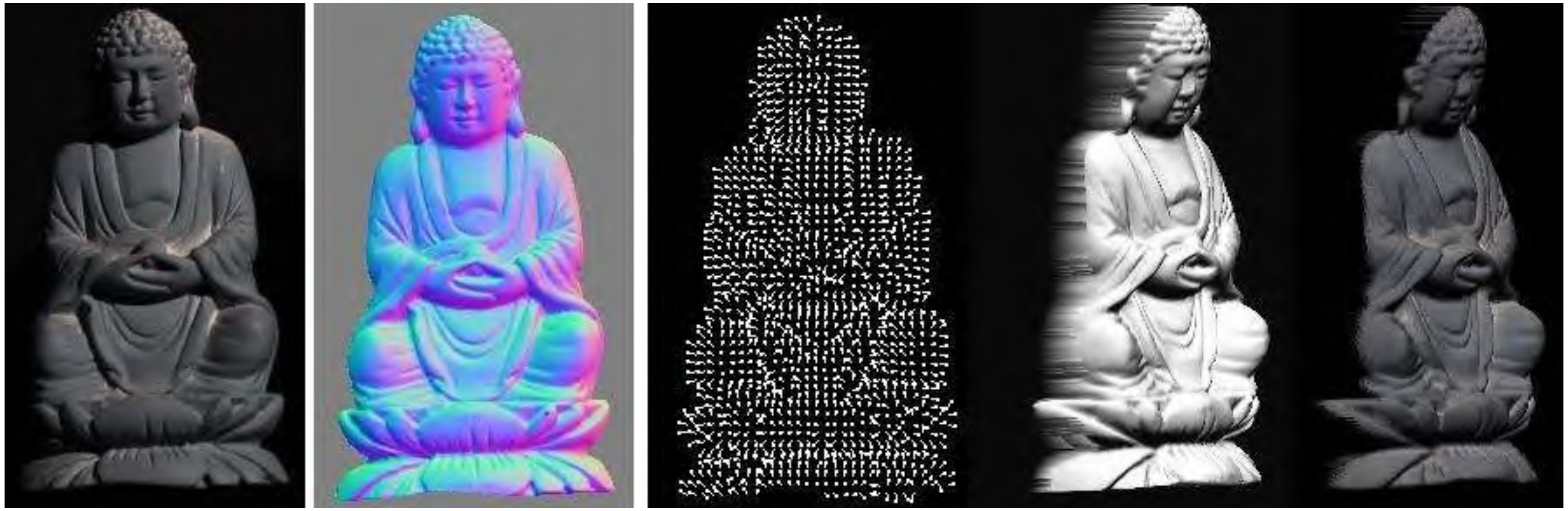
双目视图通过相机视差来计算深度

拓展：形状

利用多个光源和亮度来计算形状

Photometric Stereo：光度立体

形状表示方法



Input (1 of 12) | Normals (RGB colormap) | Normals (vectors) | Shaded 3D rendering | Textured 3D rendering