



第 4 章 感知 (2)

- ◆ 飞跃时间测距方式
- ◆ 三角测距
- ◆ 计算机视觉基本原理
- ◆ 景深测距：锐度比较、点扩展函数
- ◆ 双摄像机立体视觉
- ◆ 光流法
- ◆ 颜色跟踪



飞越时间测距

◇ 测距传感器:

◇ 对环境中物体（障碍物）距离进行测量的仪器

◇ 距离信息的作用:

◇ 定位和环境建模的关键要素



飞越时间测距

◇ 飞越时间测距典型代表:

- ◇ 超声波传感器: 超声波
- ◇ 激光测距传感器: 电磁波

◇ 声音或电磁波的传播距离由下式计算:



$$d = c \cdot t$$

- ◇ 其中
- ◇ d = 传播距离 (通常是往返距离)
- ◇ c = 波传播速度
- ◇ t = 飞越时间



飞越时间测距

- ◇ 需要指出：
 - ◇ 声音的传播速度 v : 0.3 m/ms
 - ◇ 电磁信号的传播速度 v : 0.3 m/ns,
 - ◇ 快一百万倍
 - ◇ 3 米
 - ◇ 超声系统飞越时间 10 ms
 - ◇ 激光测距仪仅 10 ns
 - ◇ 测量电磁信号的飞越时间不是容易的事
 - ◇ 激光测距传感器昂贵, 精密
- ◇ 飞越时间测距传感器的质量主要取决于：
 - ◇ 返回信号精确到达时间的不确定性
 - ◇ 飞越时间测量的不准确性 (尤其对激光测距仪)
 - ◇ 发射波束的开角 (主要指超声测距传感器)
 - ◇ 波束与目标的交互作用 (表面吸收、镜面反射)
 - ◇ 波速变化 (如, 超声波速受环境温度影响)
 - ◇ 移动机器人与目标的相对速度 (动态情况下)



超声传感器

- ◇ 发送压力（超声纵波）波包
- ◇ 引起反射的物体距离 d 可以根据声音传播速度 c 和飞越时间 t 进行计算

$$d = \frac{c \cdot t}{2}$$

- ◇ 空气中声音速度 c (340 m/s) 由下式给出:

$$c = \sqrt{\gamma \cdot R \cdot T}$$

其中,

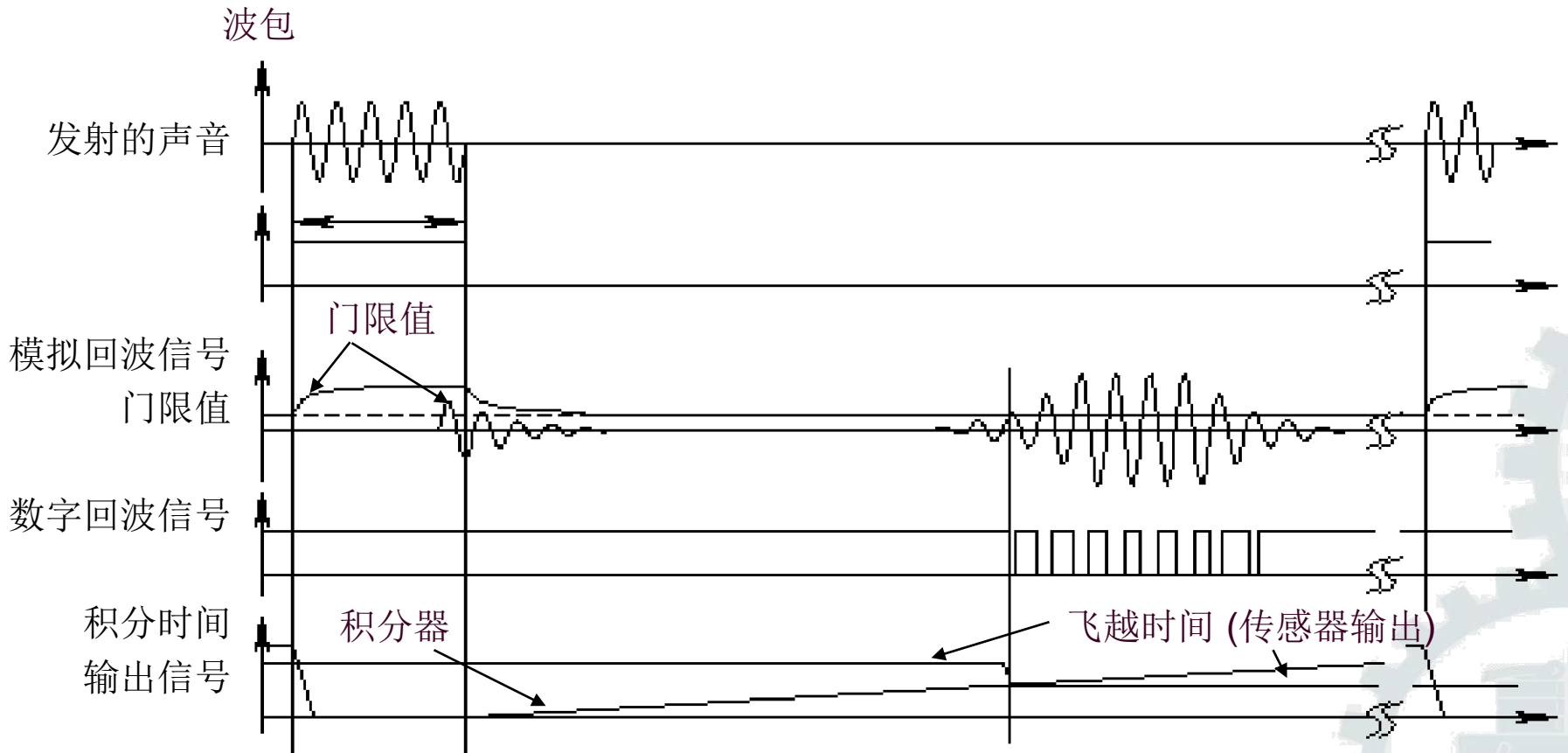
γ : 特定的热比率

R : 气体常数

T : 温度, 单位 K (Kelvin, 绝对温度)



超声传感器

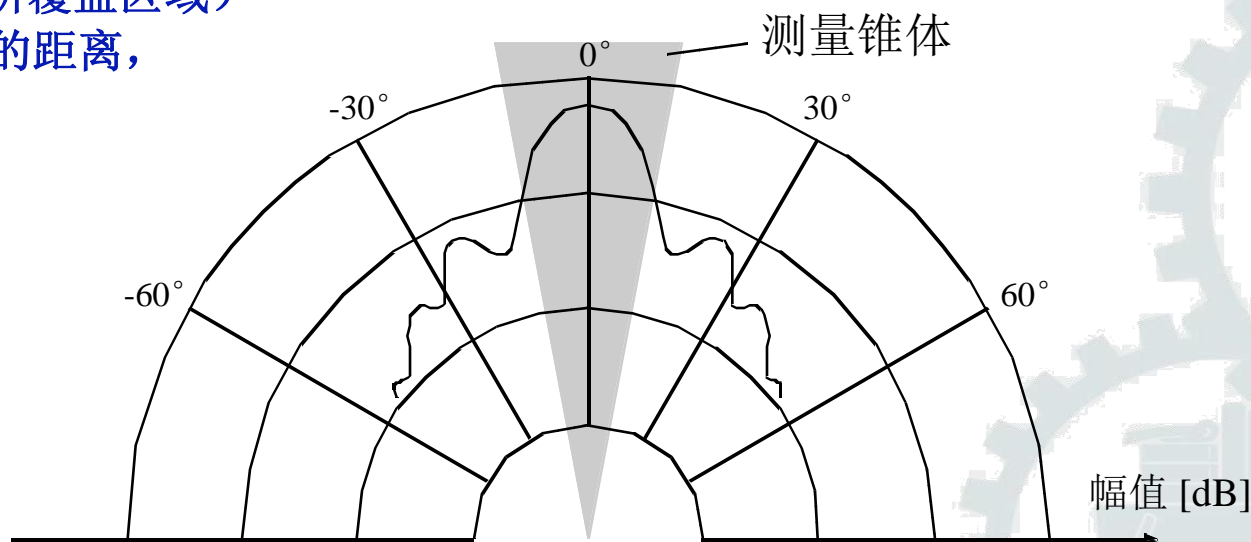


超声传感器的信号



超声传感器

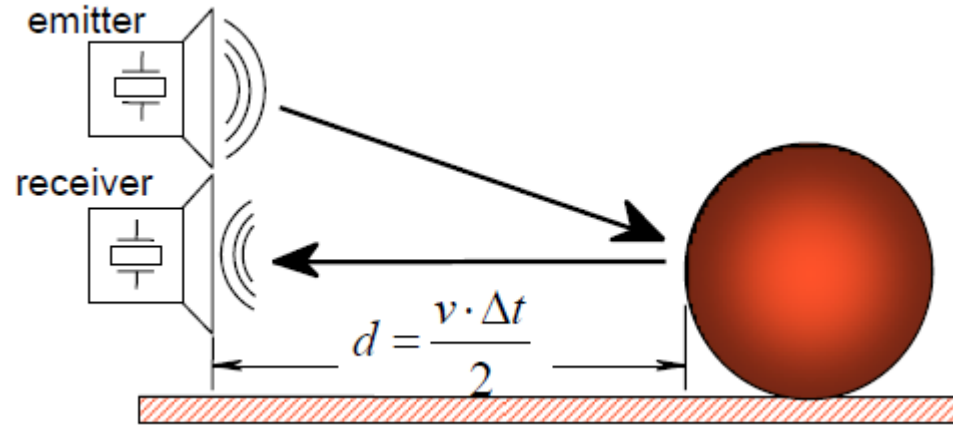
- ◇ 典型频率：40 - 180 kHz
- ◇ 声波发生：压电变换器
 - ◇ 收发器分开或一体
- ◇ 声音以锥形波方式传播
 - ◇ 开角（锥顶角）在 20 到 40 度之间
 - ◇ 恒定深度的区域（锥体所覆盖区域）
 - ◇ 对一段弧 (3D时是球面)的距离，而不是对点的距离



超声传感器波强度的典型分布



超声传感器

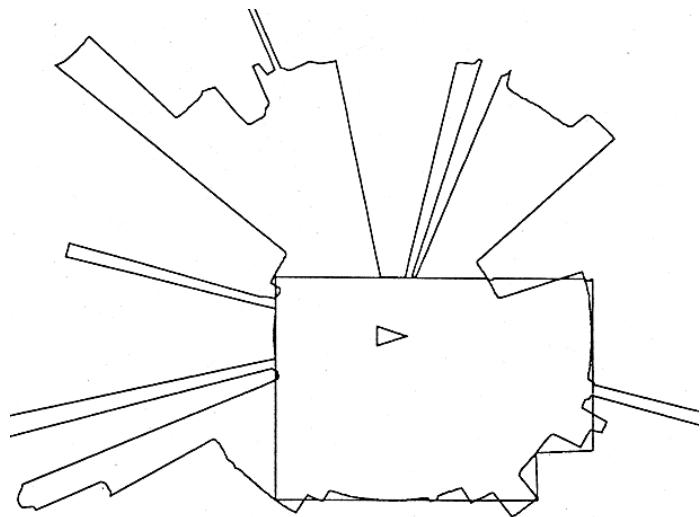
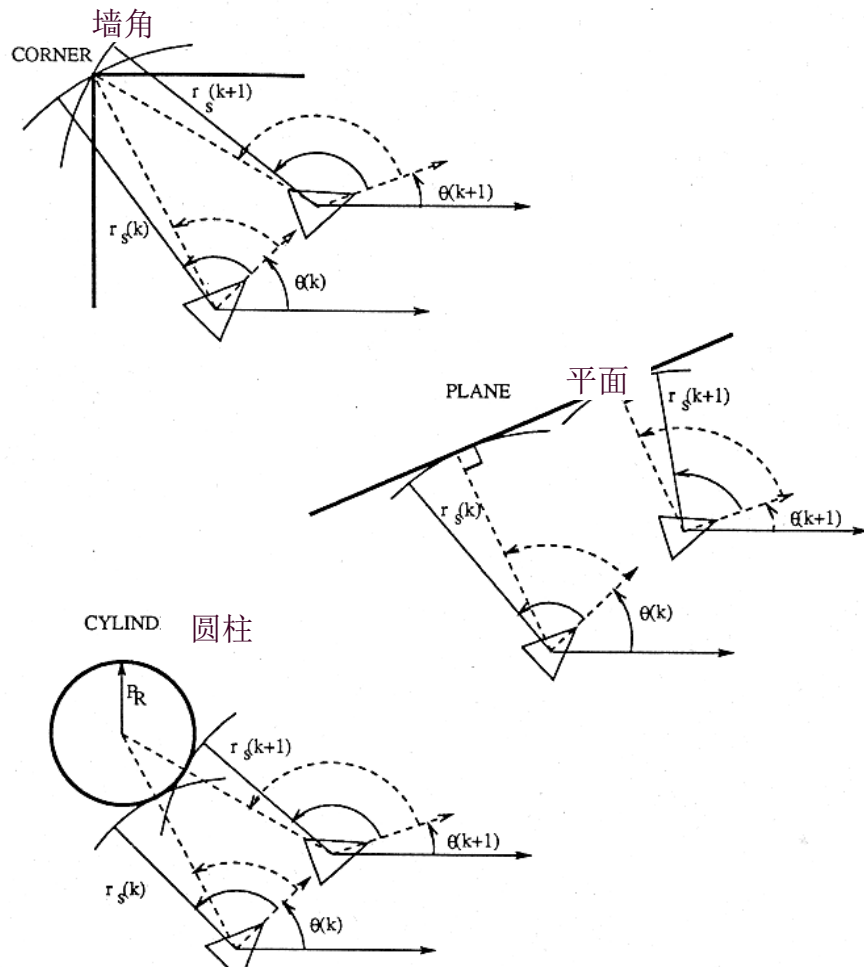




超声传感器

超声传感器的其他问题

- ◆ 非点测量，圆弧段测量
- ◆ 柔软表面，吸收大部分声音能量
- ◆ 表面与声音传播方向不是垂直的
 - ◆ 镜面反射



a) 360° 扫描

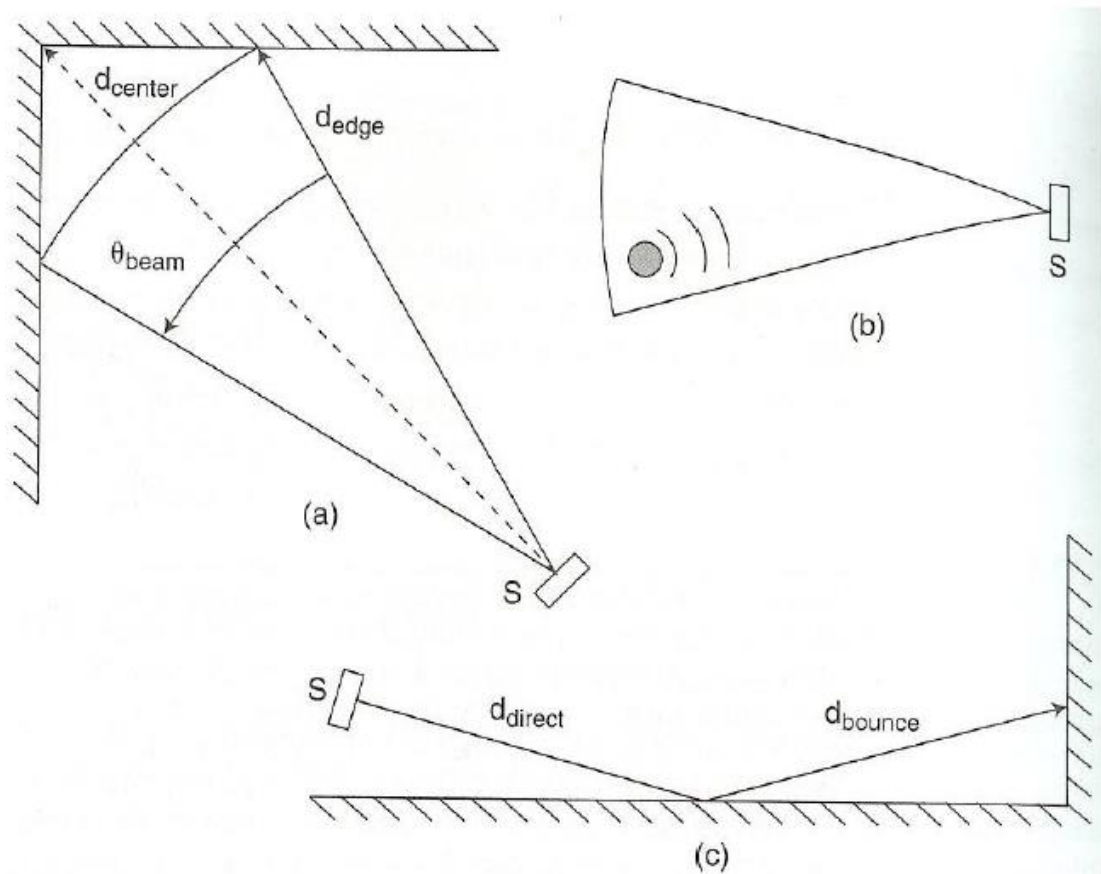
0.5 meters

b) 从不同几何基元得到结果



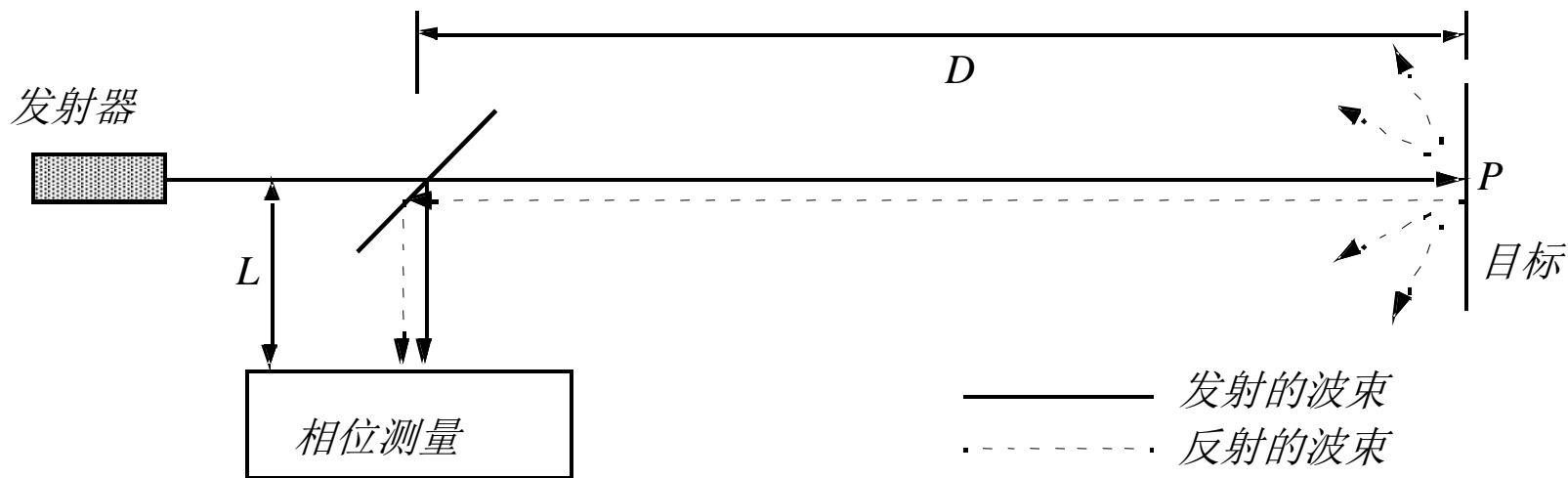
超声传感

- ◆ 返回距离比实际距离短
- ◆ 回波不足以产生读数
- ◆ 返回距离比实际距离长





激光测距传感器 (飞越时间, 电磁式) (1)



- ◇ 发射波束与接收波束同轴
- ◇ 发射器用直射激光束照射目标
- ◇ 接收器检测波束往返时间
- ◇ 带有镜子的机械装置实现扫描
 - ◇ 实现 2 维或 3 维测量



激光测距传感器 (飞越时间, 电磁式) (2)

飞越时间测量

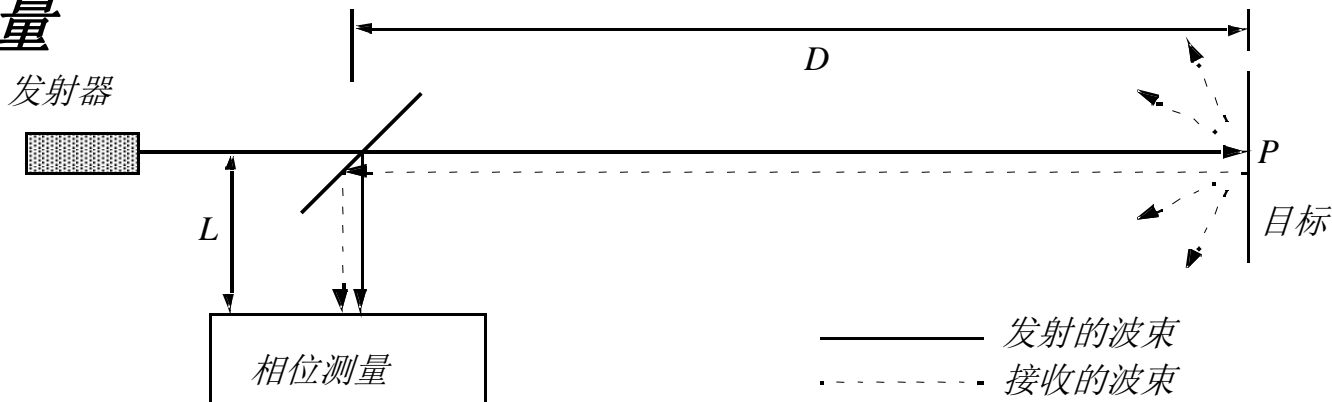
◇ 脉冲式激光

- ◇ 直接测量波束往返花费的时间
- ◇ 能分辨皮秒 (ps)
- ◇ 测量调频连续波与接收的反射波之间的差频
- ◇ 测量反射波相移, 产生距离估计
 - ◇ 技术上比前两种方法简单



激光测距传感器 (飞越时间, 电磁式) (3)

◇ 相移测量



$$\lambda = c/f \quad D' = L + 2D = L + \frac{\theta}{2\pi} \lambda$$

其中,

c : 光速; f : 调制频率; D' : 射出光束覆盖的距离

◇ 对于 $f = 5 \text{ Mhz}$ (如 A.T&T. 传感器产品), $\lambda = 60 \text{ m}$



激光测距传感器 (飞越时间, 电磁式) (4)

- ◇ 从光束分离器到目标的距离 D :

$$D = \frac{\lambda}{4\pi} \theta$$

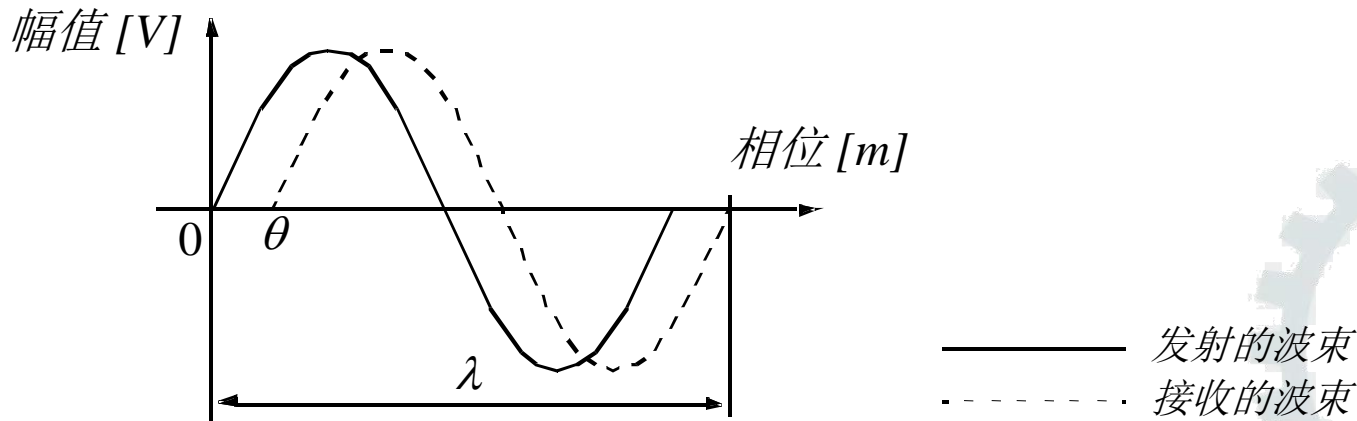
(2.33)

- ◇ 其中

- ◇ θ : 测量得到的发射和接收光束的相位差

- ◇ 理论上, 模糊的距离估计

- ◇ 如 $\lambda = 60 \text{ m}$, 距离 5 m 的目标 = 距离 65 m 的目标





激光测距传感器 (飞越时间, 电磁式) (5)



SICK

- ◆ 距离 (相关估计) 的不确定性反比于所接收信号幅值的平方。
- ◆ 因此, 深色的、远距离的物体, 不如亮色的、近距离的物体容易估计距离...

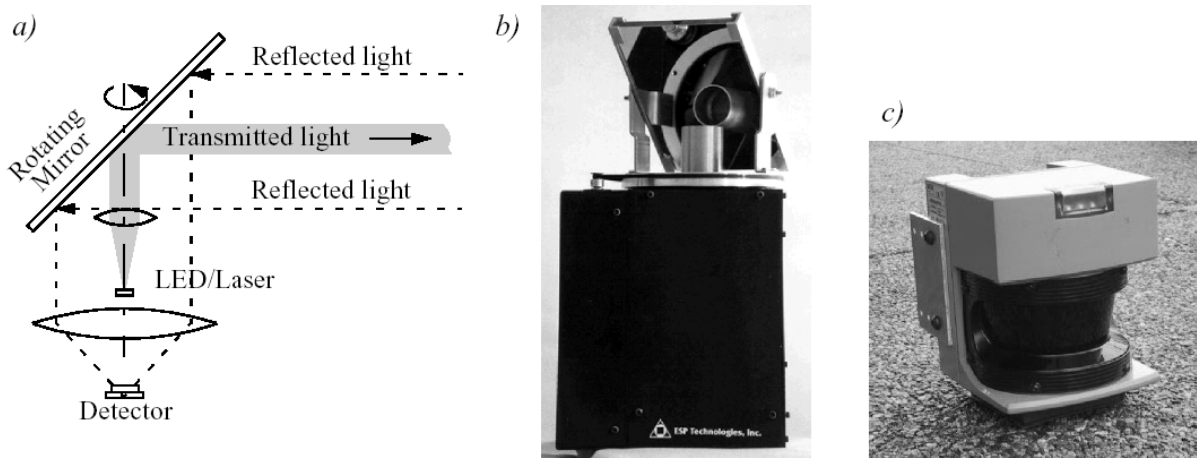


Figure 4.11
 (a) Schematic drawing of laser range sensor with rotating mirror; (b) Scanning range sensor from EPS Technologies Inc.; (c) Industrial 180 degree laser range sensor from Sick Inc., Germany

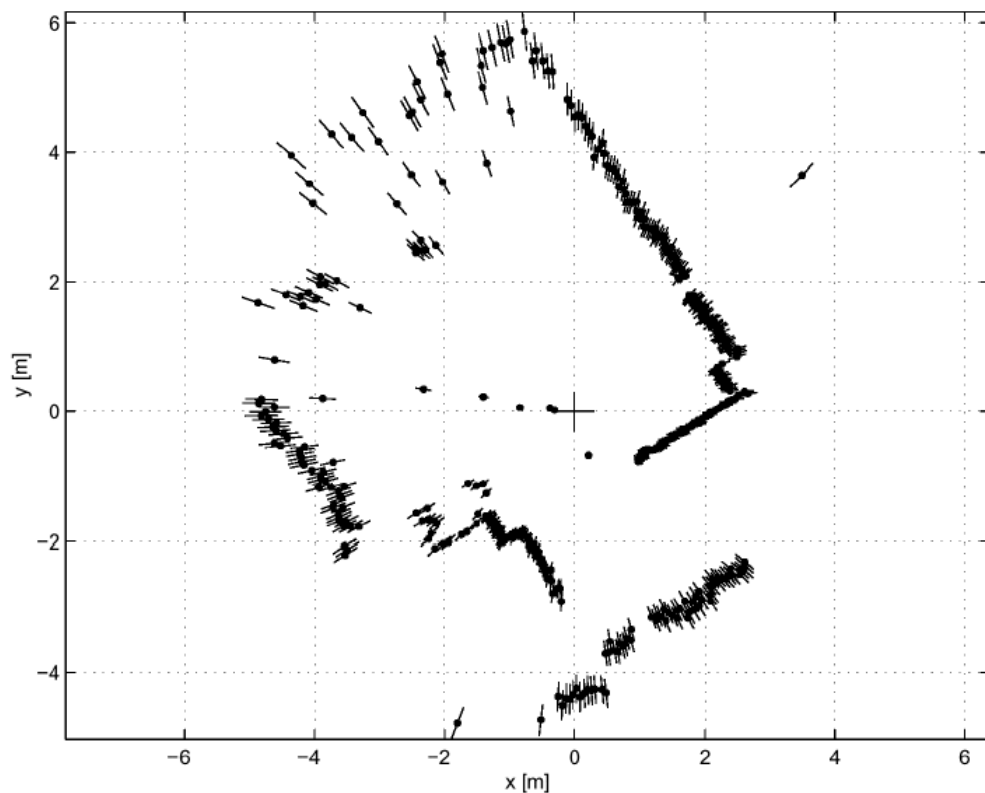
图 4.11

- (a) 带有转动镜的激光距离传感器的示意图; (b)EPS技术公司的扫描距离传感器; (c)德国Sick公司的工业180°激光距离传感器



激光测距传感器 (飞越时间, 电磁式)

- ◆ 带有旋转镜的 2D 激光距离传感器的典型距离场图像。通过测量点的线段长度说明了不确定性。



C Carnegie Mellon University

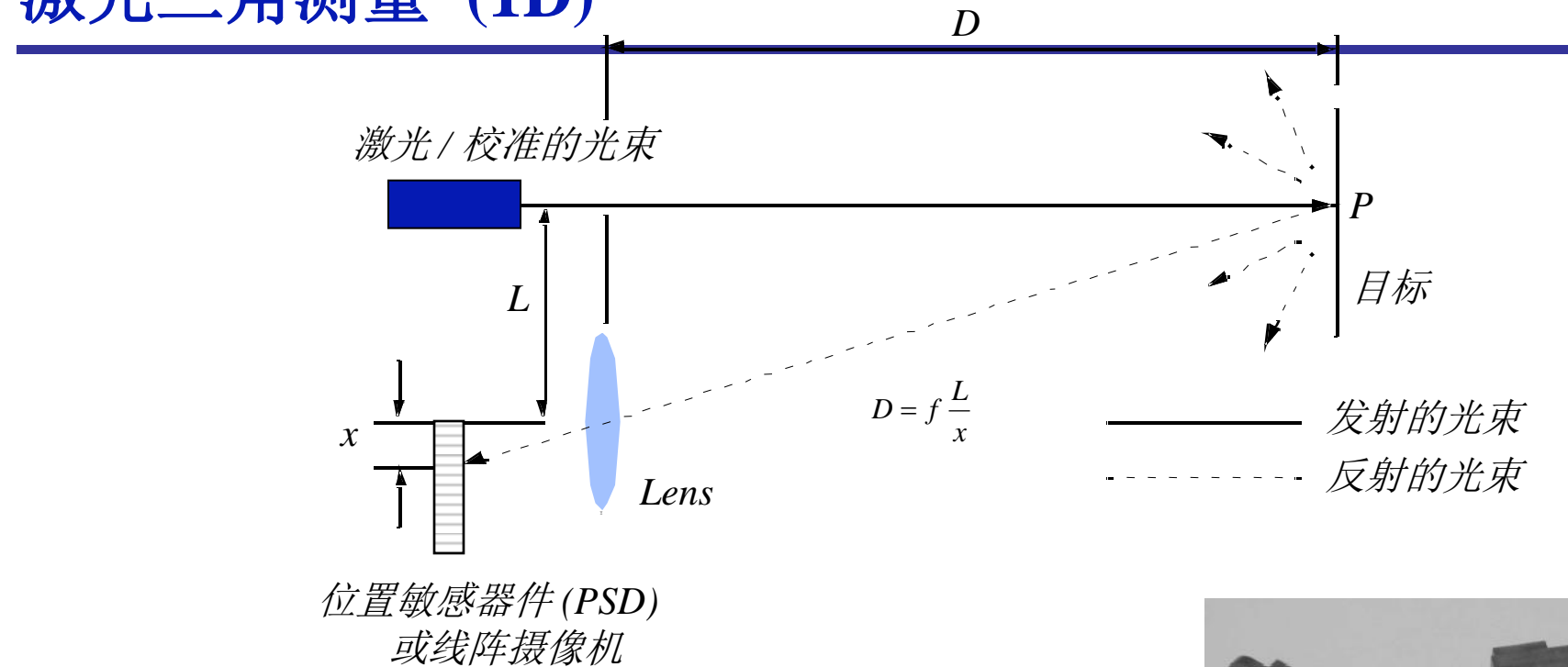


基于三角测量法测距

- ◇ 利用几何性质，确定物体距离
- ◇ 如，将已知的光模式（如点、线）投影到环境
 - ◇ 其（已知的光模式）反射光，被感光线阵或面阵（摄像机）传感设备捕获。
 - ◇ 简单的三角测量就可以确定距离。
- ◇ 如，被捕获物体的尺寸精确已知
 - ◇ 不需要光投影就可以进行三角测量

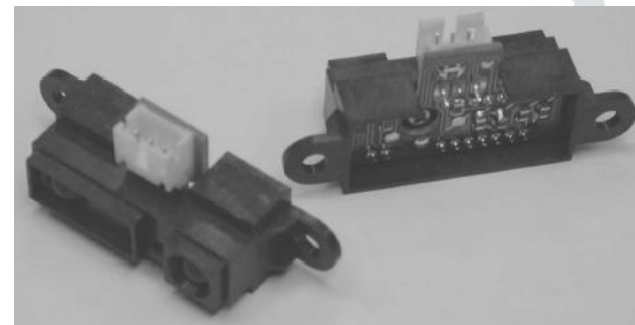


激光三角测量 (1D)



激光三角测量的原理

$$D = f \frac{L}{x}$$



◆ 距离与 $1/x$ 成正比。



毫米波雷达

波段	频率	传播特性	主要用途
超长波	3-30KHz	空间波为主	海岸潜艇通信；远距离通信；超远距离导航
长波	30-300KHz	地波为主	越洋通信；中距离通信；地下岩层通信；远距离导航
中波	0.3-3MHz	地波与天波	船用通信；业余无线电通信；移动通信；中距离导航
短波	3-3MHz	地波与天波	远距离短波通信；国际定点通信
米波	30-300MHz	空间波	电离层散射；流星余迹通信；人造电离层通信；对空间飞行通信；移动通信
分米波	0.3-3GHz	空间波	小容量微波中继通信；对流层散射通信；中容量微波通信
厘米波	3-30GHz	空间波	大容量微波中继通信；数字通信；卫星通信；国际海事卫星通信
毫米波	30-300GHz	空间波	无人驾驶距离检测；直升机高压线识别；大气层通信，波导通信



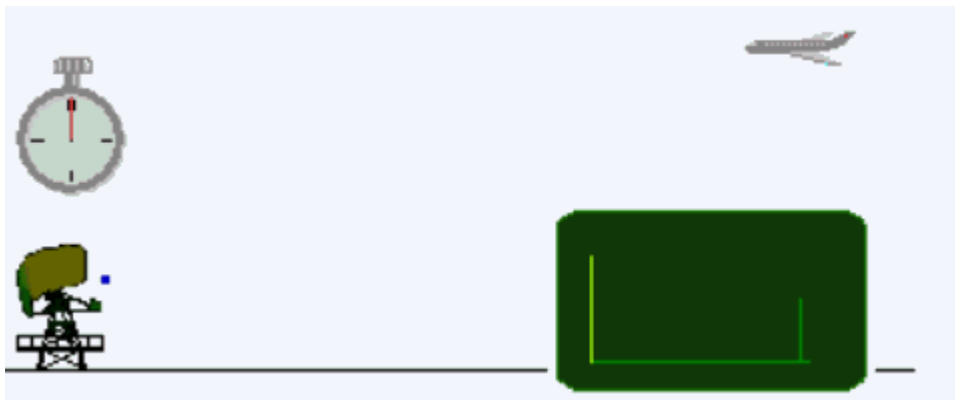
毫米波雷达

◇ 毫米波

- ◇ 工作频率30-100GHz，波长1-10mm之间的电磁波
- ◇ 毫米波波长介于微波和厘米波之间，兼有微波雷达和光电雷达的一些优点

◇ 毫米波雷达

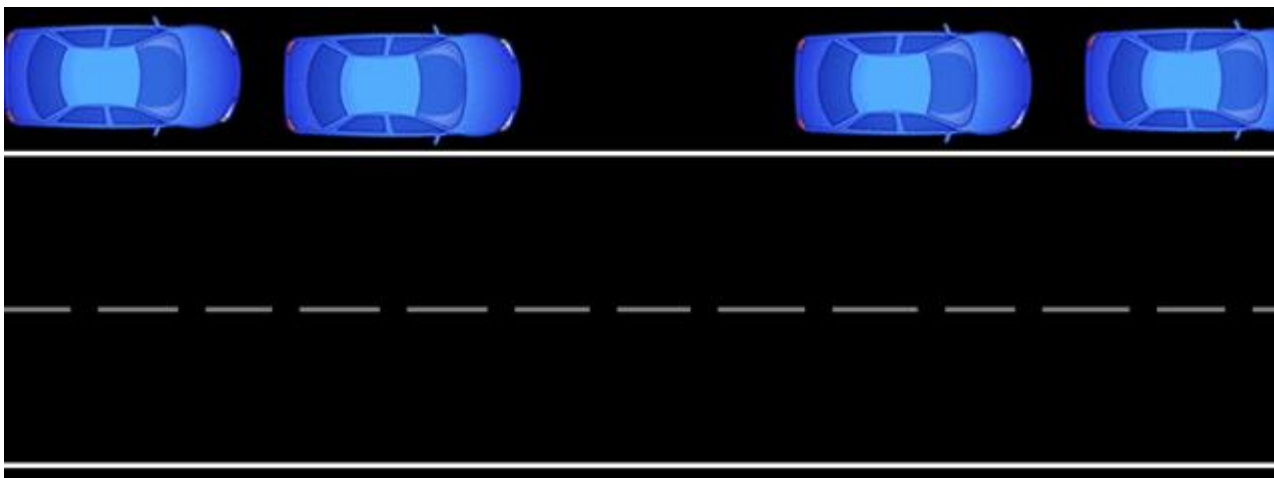
- ◇ 工作频段在毫米波频段的雷达，发出电磁波，接收回波，根据收发时间差测得距离
- ◇ 测距范围较大，可达200多米
- ◇ 可对目标进行有无检测、测距、测速以及方位测量





毫米波雷达

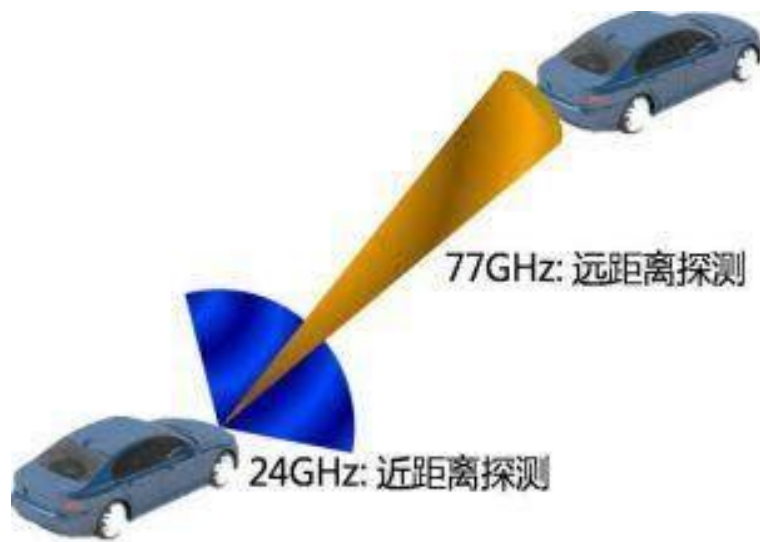
- ◆ 毫米波雷达分为远距离雷达（LRR）和近距离雷达（SRR），由于毫米波在大气中衰减弱，所以可以探测感知到更远的距离，其中远距离雷达可以实现超过200m的感知与探测。
- ◆ 毫米波雷达的多项优势，使其目前在汽车防撞传感器中占比较大，根据IHS的数据，毫米波/微波雷达+摄像头在汽车防撞传感器中占比达到了70%。





毫米波雷达

- ◆ 目前市场主流使用的车载毫米波雷达按照其频率的不同，主要可分为两种：24GHz毫米波雷达和77GHz毫米波雷达。
- ◆ 通常24GHz雷达检测范围为中短距离，用作实现BSD（BlindSpotDetection，盲点探测系统），而77GHz长程雷达用作实现ACC（AdaptiveCruiseControl，自适应巡航系统）。





毫米波雷达

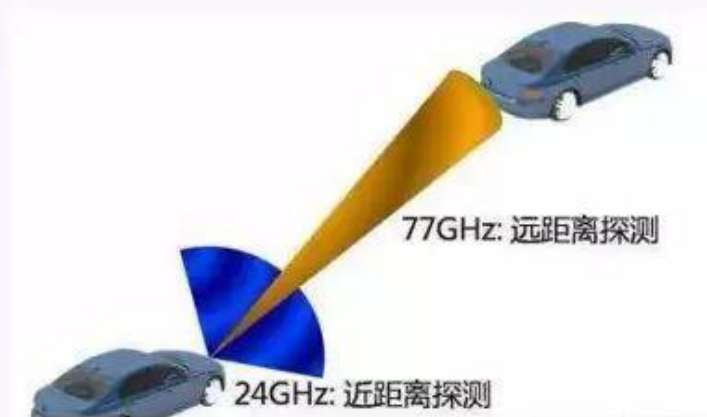
毫米波雷达主要应用在汽车领域，包括24GHz和77GHz毫米波雷达。在国内市场，24GHz与77GHz毫米波雷达主要存在4点区别。

1. 77GHz毫米波雷达的体积更小。

由于24GHz雷达的频率更低波长更长，因此雷达所需要的天线就更长，24GHz毫米波雷达比77GHz毫米波雷达的体积更大。

2. 77GHz毫米波雷达所需要的工艺更高。

77GHz毫米波雷达最大的制造难度体现在其工艺上，77GHz毫米波雷达由于体积小，其线路板的面积很小，因此射频线路的设计难度非常高，成片的成品率也比较低。





毫米波雷达

3. 77GHz毫米波雷达的检测精度更好。

相比于24GHz雷达，77GHz雷达的波长更小，虽然绕射能力比24GHz雷达要弱，但是其检测精度更高。

4. 24GHz毫米波雷达的射频芯片对相对77GHz雷达射频芯片更易获取。

各大厂商经过多年对24GHz毫米波雷达的研发，市场上24GHz毫米波雷达的产品体系已经相对成熟，供应链已经相对稳定。

在全球，77GHz毫米波雷达将在行业普遍产业化。而限于芯片技术等原因，我国77GHz毫米波雷达的大规模应用将稍微推后。





毫米波雷达

目前来说，毫米波雷达的技术主要由博世、大陆、电装、奥托立夫、法雷奥、德尔福、海拉、采埃孚天合等厂商主导，其中法雷奥、海拉和博世占据我国 24 GHz 雷达市场 60% 以上的出货量，德国大陆、博世和德尔福占据我国 77 GHz 雷达市场 80% 以上的出货量。

主要厂商	主要产品	雷达频率	探测距离
博世 (Bosch) 德国	中距离雷达MRR	76-77GHz	前向: 不大于160m 后向: 不大于80m
	长距离雷达LRR4	76-77GHz	不大于250m
大陆 (Continental) 德国	短距离雷达SRR320	24-25GHz	
	长距离雷达ARS410	76-77GHz	不大于170m
	长距离雷达ARS430	76-77GHz	不大于250m
德尔福 (Delphi) 美国	中距离雷达ESR2.5	76-77GHz	不大于174m
奥托立夫 (Autoliv) 瑞典	短距离雷达	24-25GHz	
电装 (Denso) 日本	长距离雷达	76-77GHz	
采埃孚天合 (ZF-TRW) 德国	中距离雷达AC100	24-25GHz	不大于 150m
海拉 (Hella) 德国	24 GHz雷达	24.15 GHz	0.75m至70m



毫米波雷达

我国的毫米波雷达，相比于国外企业，仍属于起步阶段，但其技术也在不断加速升级，24 GHz 毫米波集成电路已实现规模量产。芜湖森思泰克、杭州智波、华域汽车等实现了 24 GHz 毫米波雷达的批量生产，沈阳承泰科技、湖南纳雷科技、南京隼眼电子、北京行易道、华域汽车、厦门意行半导体等均完成了 77 GHz 雷达的样机测试。

主要厂商	雷达频率	主要优势	相对劣势
湖南纳雷科技	24GHz	中段距离雷达已经产品化	产品关注点较分散
杭州智波	24GHz 77GHz	亚太投资，会提供一定的技术支持	研发经验相对薄弱
芜湖森思泰克	24GHz 77GHz	研发实力在国内属前沿，24GHz与77GHz雷达进度较快	产品线定位分散
沈阳承泰科技	77GHz	77GHz雷达已在内部测试阶段	缺乏量产概念
南京隼眼电子	77GHz	依托东南大学国家毫米波雷达重点实验室	投入较为不足
北京行易道	77GHz	77GHz毫米波雷达研究经验丰富	缺乏汽车工业相关经验
华域汽车	24GHz	24GHz雷达研发已有多年经验，上市公司资源齐全	缺乏创新及技术积累
厦门意行半导体	24GHz 77GHz	是目前国内少数提供24GHz汽车主动安全雷达射频前端MMIC解决方案的企业	技术实力待提高

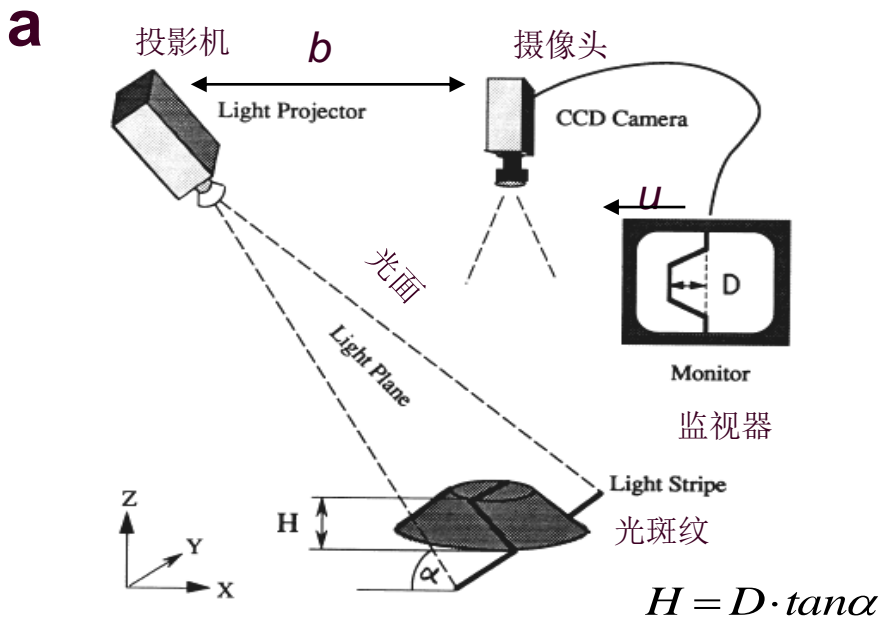
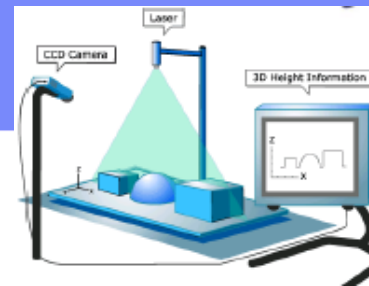


毫米波雷达

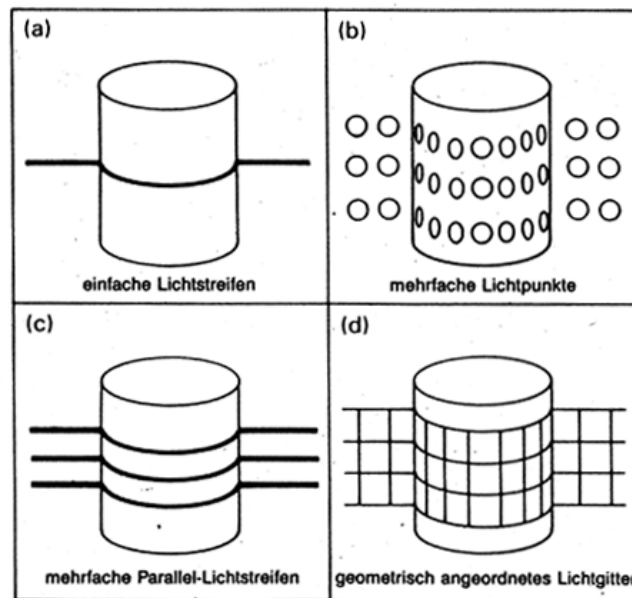
	毫米波雷达	超声波	激光雷达	红外传感器
最大作用距离 (m)	200	15	150	35
径向运动	好	好	好	差
切向运动	差	差	差	好
角度测量能力	较好	好	很好	不能
环境限制因素	全天候、不易受环境影响	风、沙尘等	雨天	温度
成本	中	低	高	中低
穿透性	好	较差	较差	差
优点	不受天气情况和夜间的影响，探测距离远	价格低，数据处理简单，体积小	测距精度高，方向性强，响应时间快，不受地面杂波干扰	成本低，夜间不受影响
缺点	成本较高，目标识别难度较大，可与摄像头互补使用	易受天气和温度影响，最大测量距离一般只有几米	成本很高，不能全天候工作。遇浓雾、雨、雪天气无法工作	会受天气影响，只能探测到近距离地物体



结构光 (视觉, 2 或 3D)



b



- ◆ 将已知的结构光（模式）投射到环境，避免了标准的校正问题。
- ◆ 利用转动的镜子，投射光的纹理(上图b)或发射已校正的光（可能是激光）。
- ◆ 采用摄像机进行光感知
- ◆ 根据简单的几何关系，可以确定到一个照亮的点的距离。



结构光 (视觉, 2 或 3D)

◆ Microsoft Kinect

◆ 以色列公司PrimeSense在2010年开

◆ 主要构成部件

- ◆ IR projector
- ◆ IR camera
- ◆ VGA camera
- ◆ Microphone array
- ◆ Motorized tilt



RGB
Camera

IR
Camera

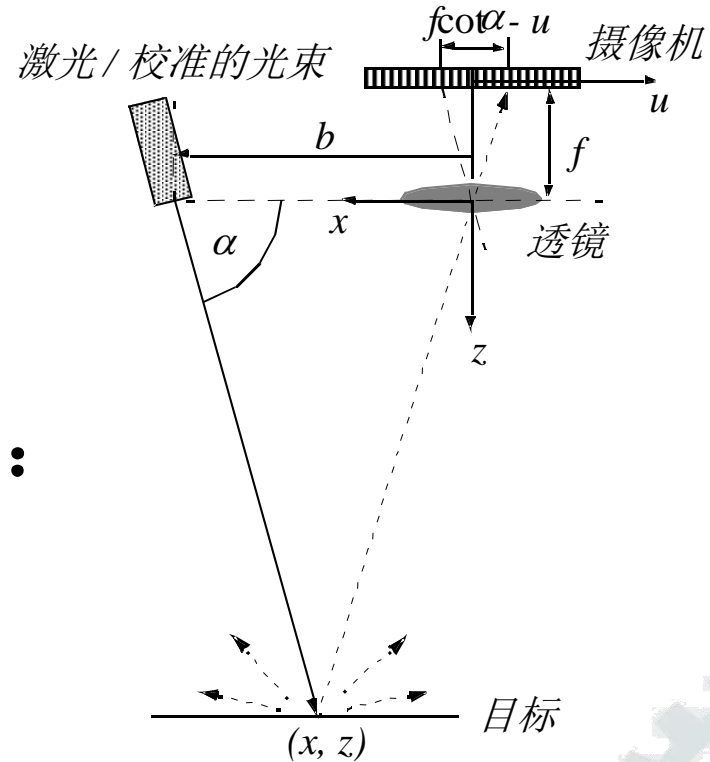
IR Laser
Projector





结构光 (视觉, 2 或 3D)

◆ 对原理的空间示意图



◆ 由图，简单的几何特征表明：

$$x = \frac{b \cdot u}{f \cot \alpha - u} ; \quad z = \frac{b \cdot f}{f \cot \alpha - u}$$

发射的光束 ——
反射的光束 - - - - -



结构光 (视觉, 2 或 3D)

- ◇ 距离分辨率被定义为三角测量增益 G_p :

$$\frac{\partial u}{\partial z} = G_p = \frac{b \cdot f}{z}$$

- ◇ α 的影响:

$$\frac{\partial \alpha}{\partial z} = G_\alpha = \frac{b \sin \alpha^2}{z}$$

- ◇ 基线长度 b :

- ◇ b 越小, 传感器可以越紧凑。

- ◇ b 越大, 距离分辨率越高。

注意: 如果 b 大, 被照亮点不在接收器视场中的机会增加。

- ◇ 焦距 f :

- ◇ 改变焦距 f 长度, 则

- ◇ 或者提供大视场 (焦距短)

- ◇ 或者改善距离分辨率 (焦距长)

- ◇ 然而, 长焦距意味着传感器的体积大



飞行时间摄像机

- ◇ 相差测量
- ◇ 直接飞行时间测量

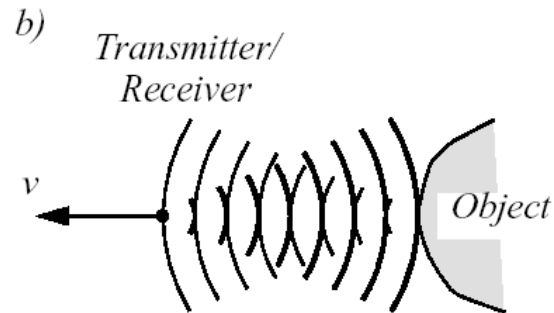
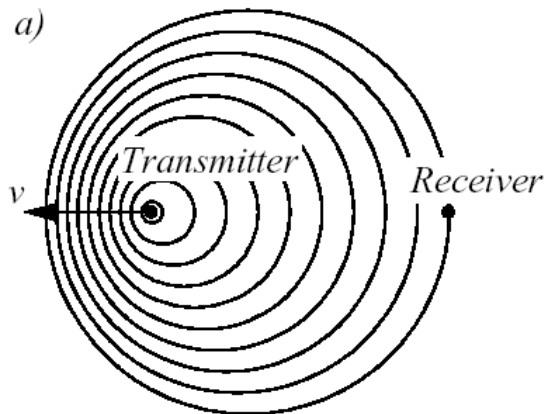


Swiss Ranger 3000
(produced by MESA)





基于多普勒效应的感知 (雷达或声音)



a) 两个物体之间的多普勒效应

b) 运动和静止物体之间的多普勒效应

$$f_r = f_t (1 + v/c) \text{ 如果发射机运动}$$

$$f_r = f_t \frac{1}{1 + v/c} \text{ 如接收机运动}$$

$$\Delta f = f_t - f_r = \frac{2f_t v \cos \theta}{c} \quad \text{多普勒频移}$$

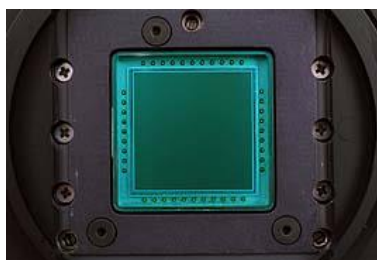
$$v = \frac{\Delta f \cdot c}{2f_t \cos \theta} \quad \text{相对速度}$$

- 声波: 如, 工业过程控制、安全、寻鱼、测量地速
- 电磁波: 如, 振动测量、雷达系统、目标跟踪



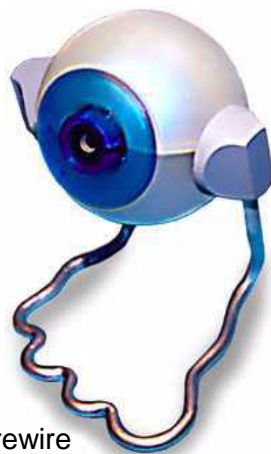
基于视觉的传感器：硬件

◆ CCD (电荷耦合器件。几百万 5 到 25 微米大小的光敏放电电容器)



2048 x 2048 CCD array

阵列



Orangemicro iBOT Firewire

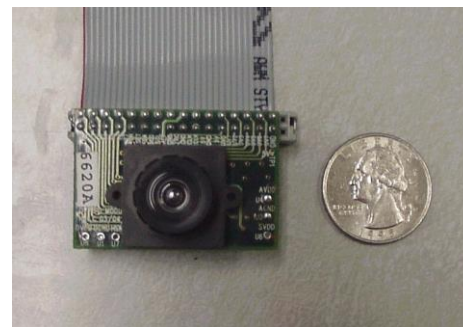


Sony DFW-X700



Canon IXUS 300

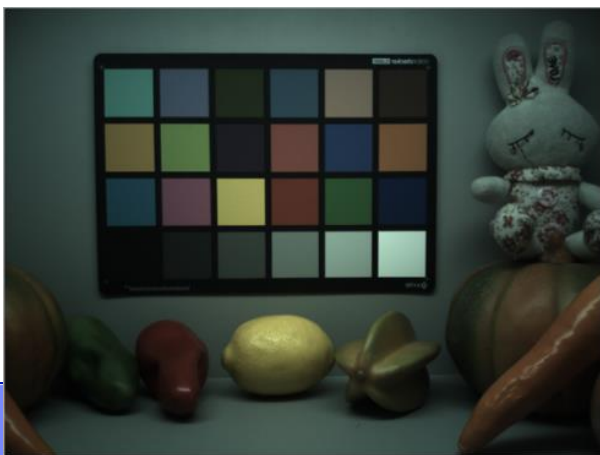
◆ CMOS (互补金属氧化半导体技术)





基于视觉的传感器

- ◇ CCD与CMOS控制电路的不同
- ◇ 光圈与快门
- ◇ 白平衡
 - ◇ 灰度世界理论
 - ◇ 全反射理论





视觉假象





视觉假象



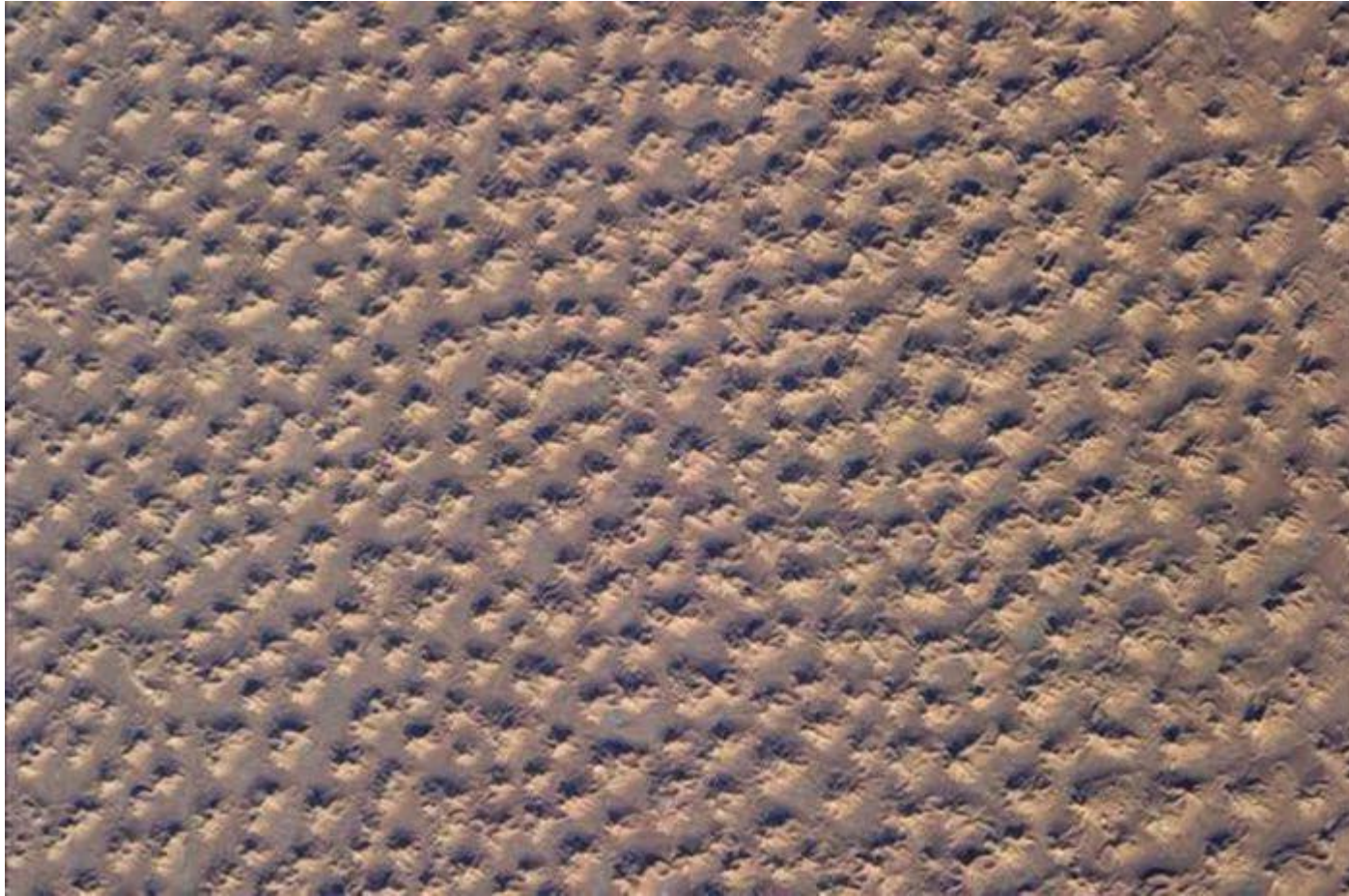


视觉假象



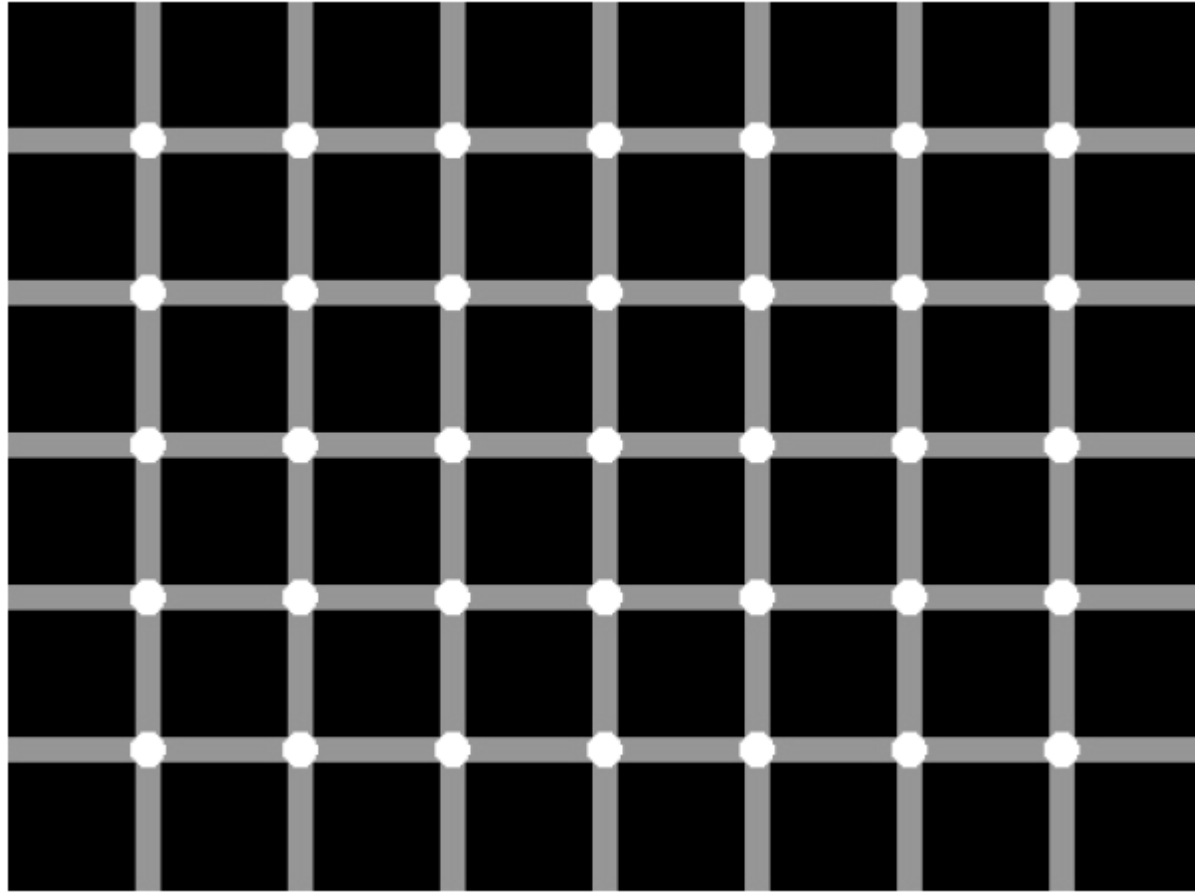


视觉假象



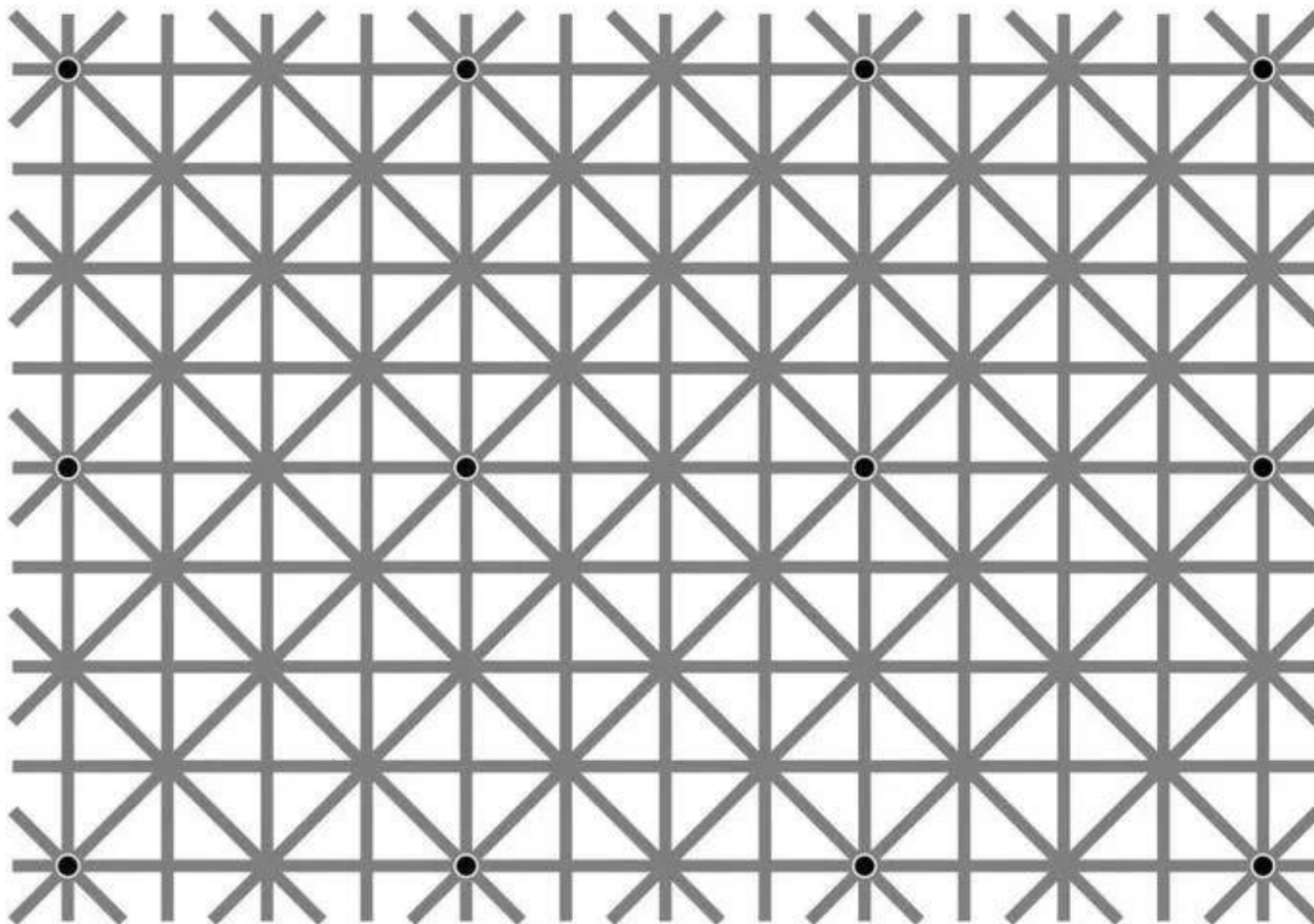


视觉假象



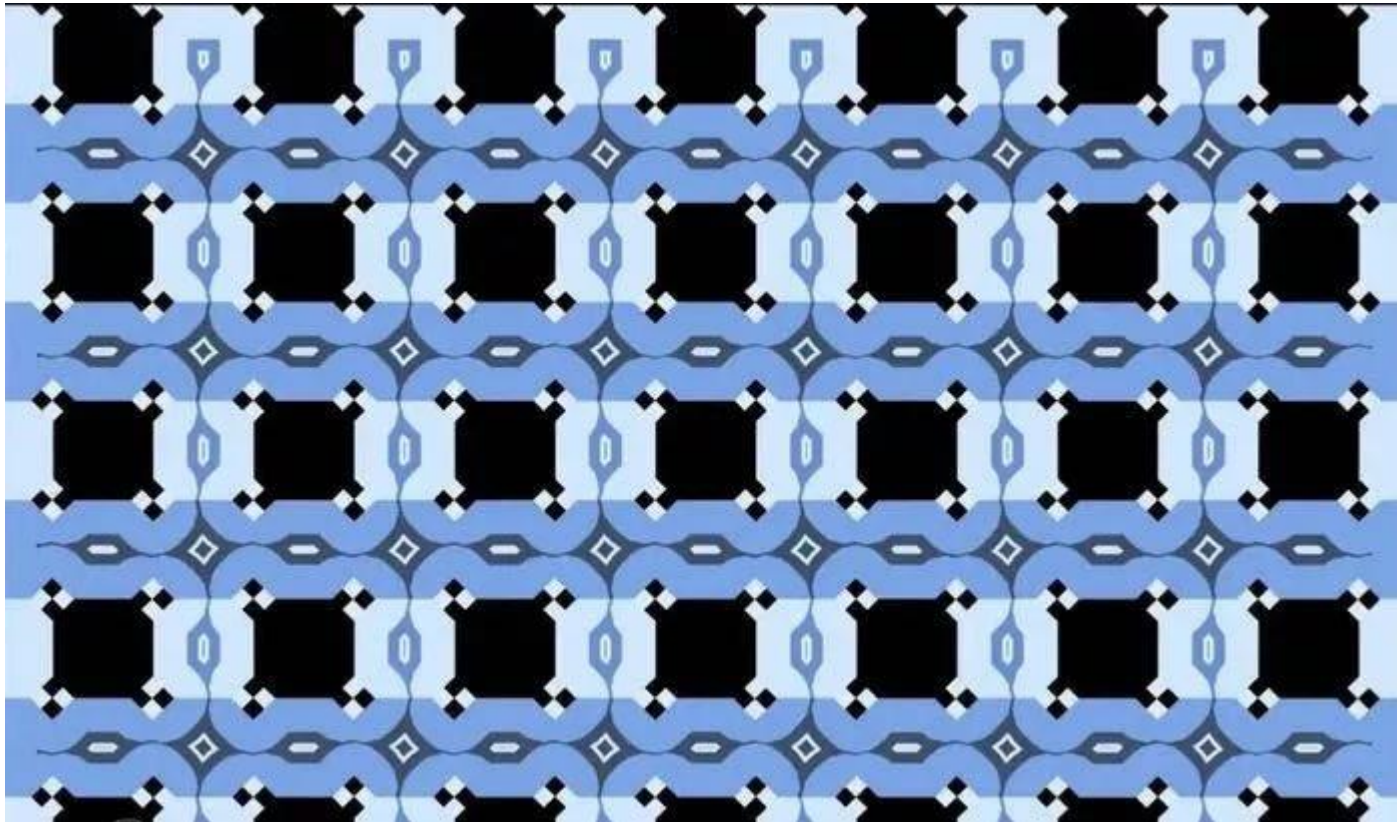


视觉假象



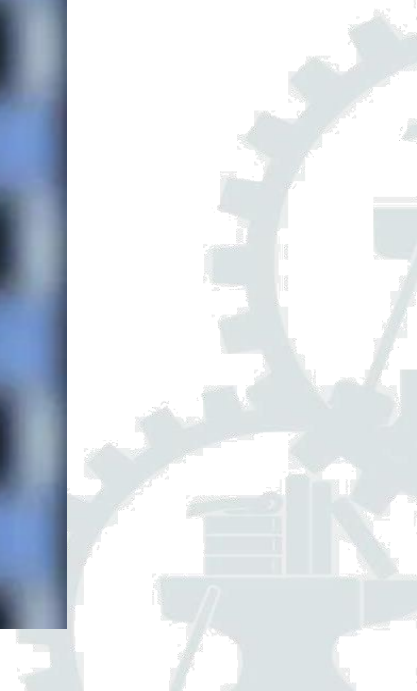
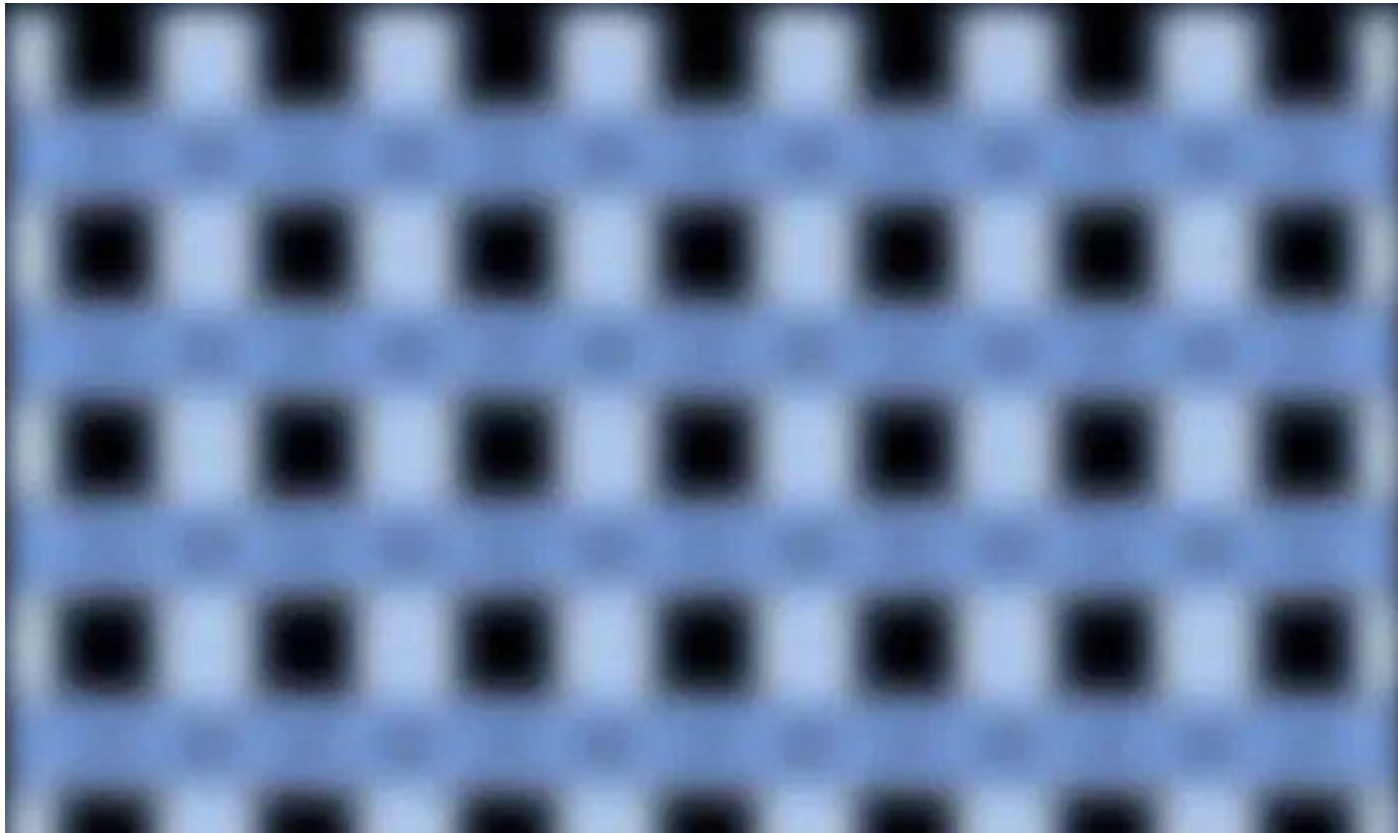


视觉假象



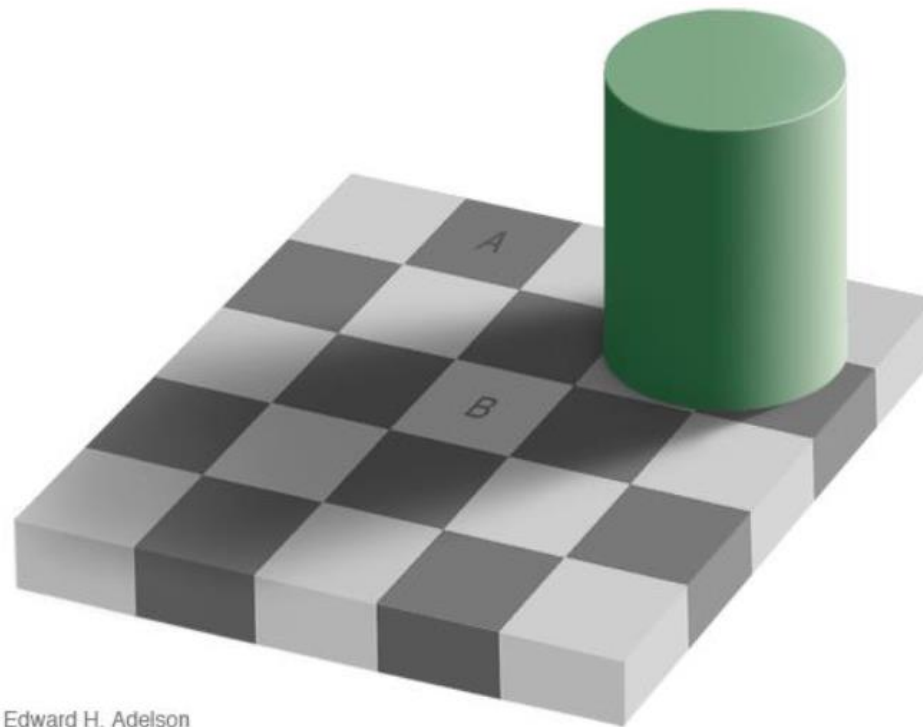


视觉假象





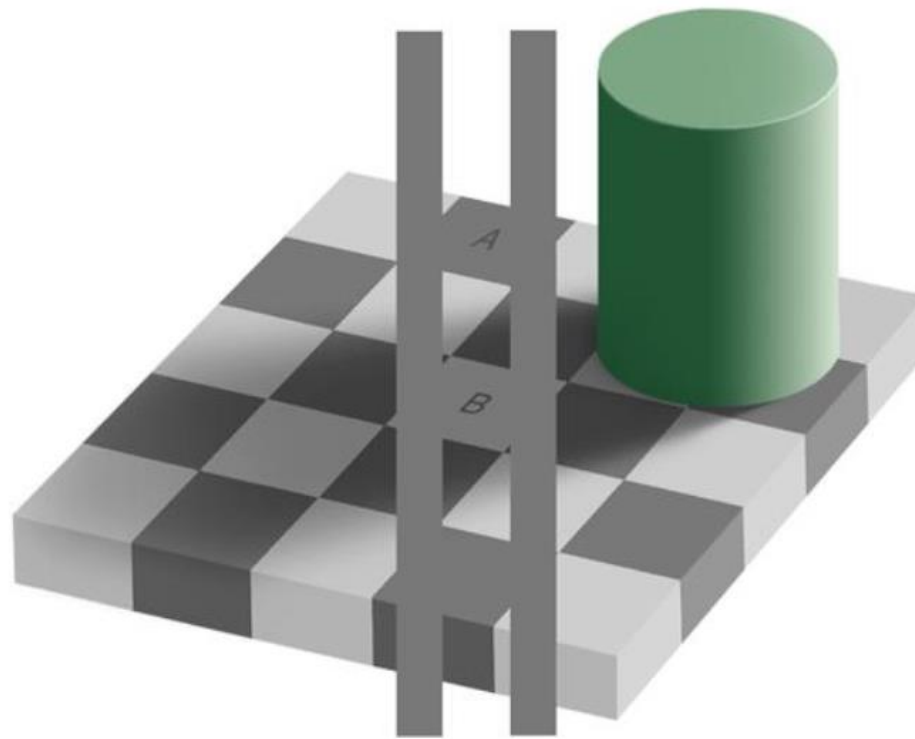
视觉假象



Edward H. Adelson

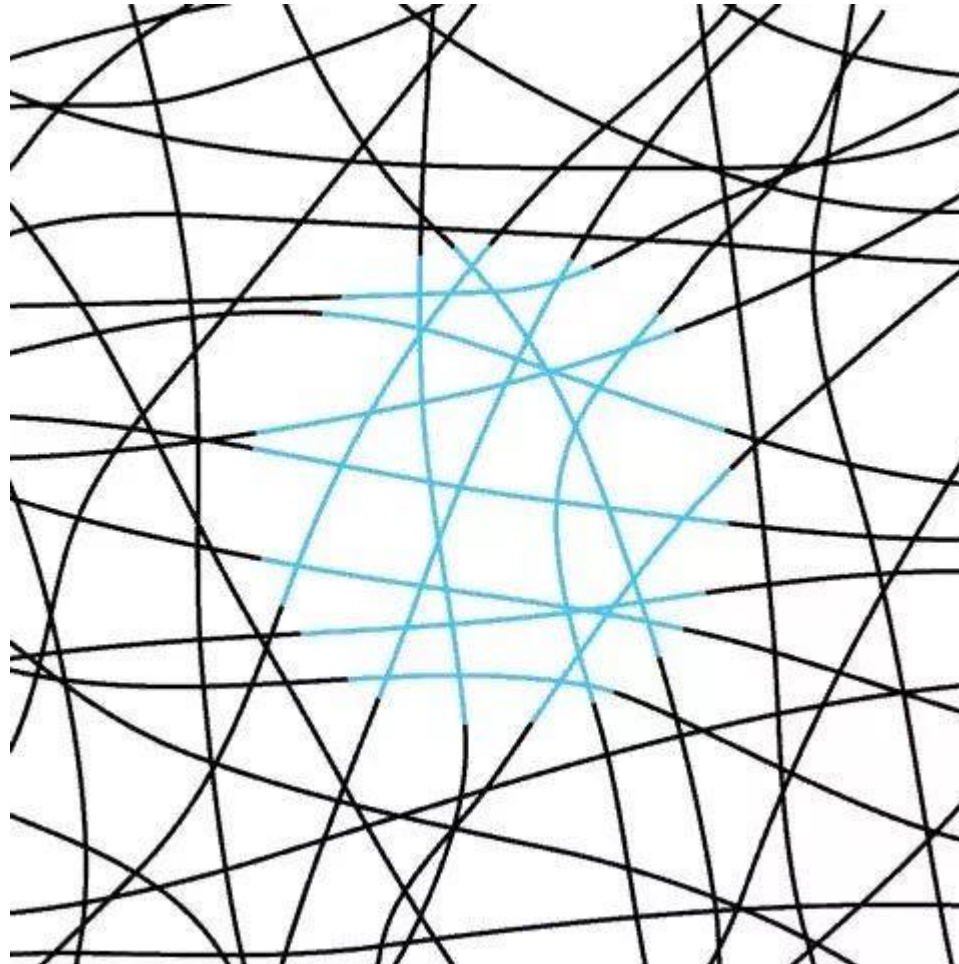


视觉假象





视觉假象



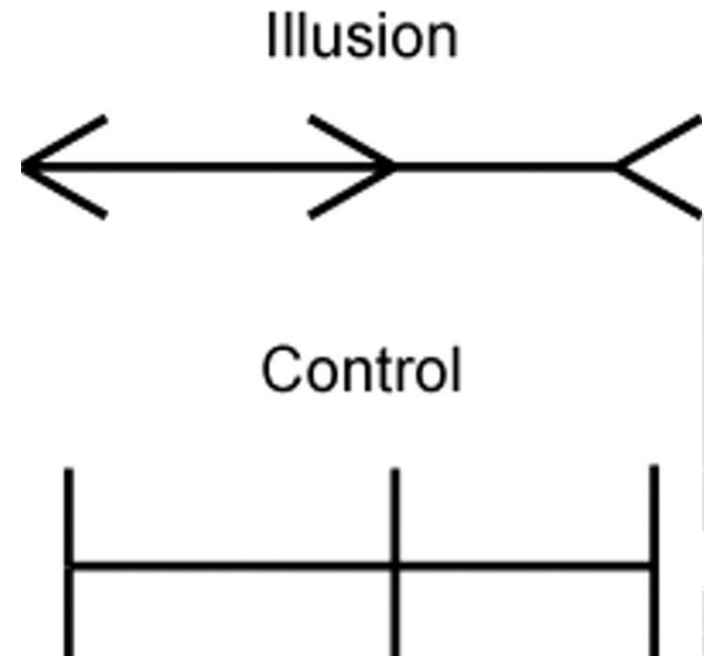


视觉假象

◆ Ames room illusion Illusion



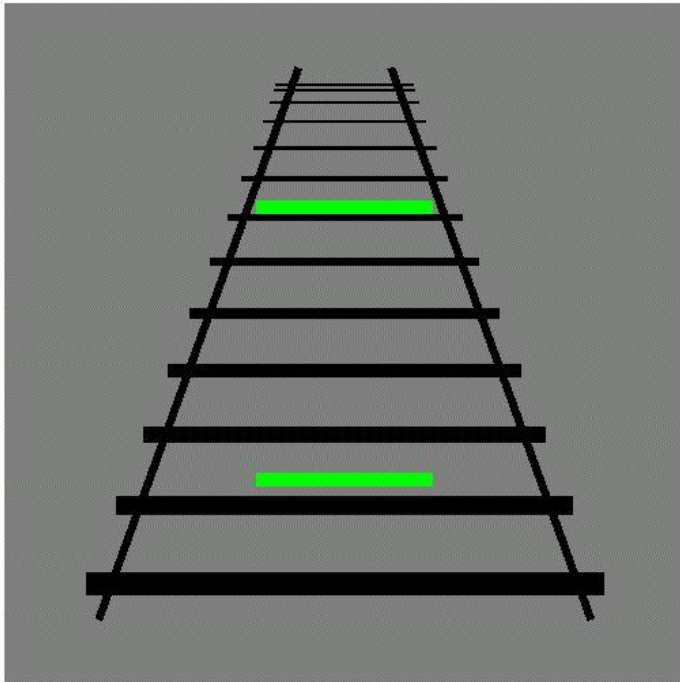
Muller-Lyer



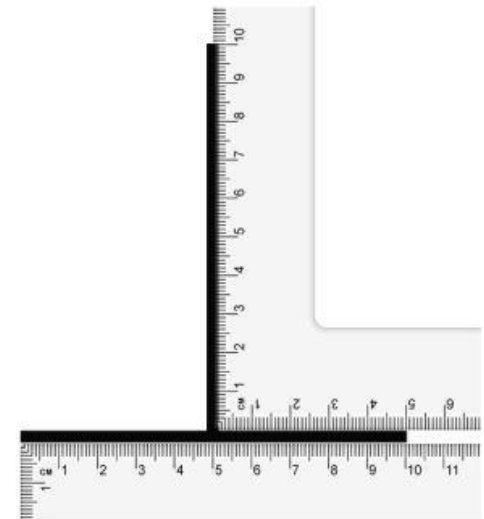
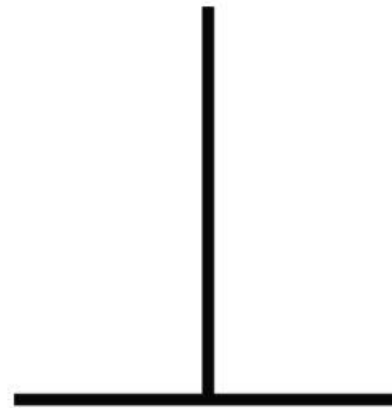


视觉假象

◇ Ponzon illusion vertical illusion



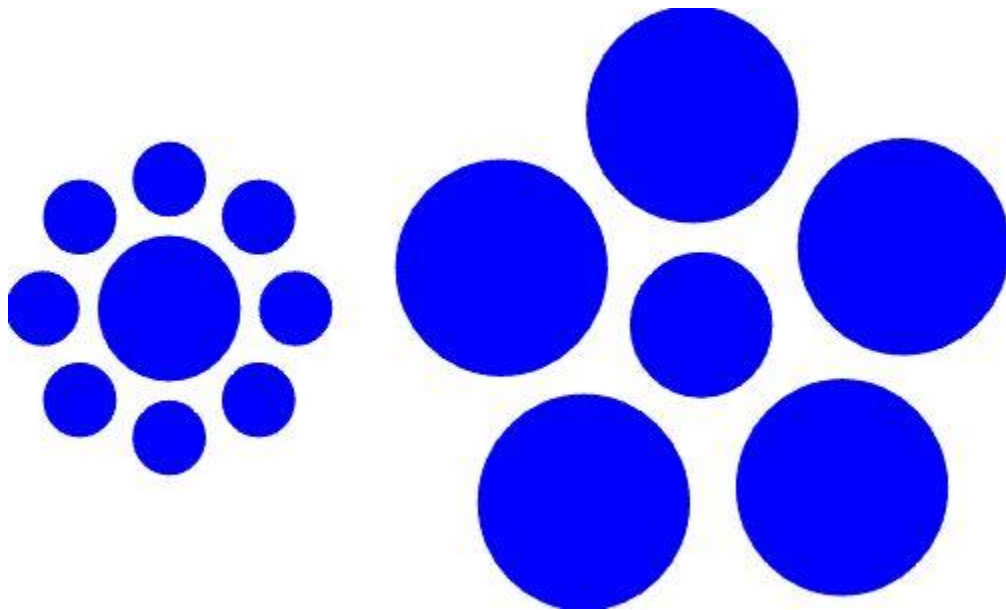
horizontal-





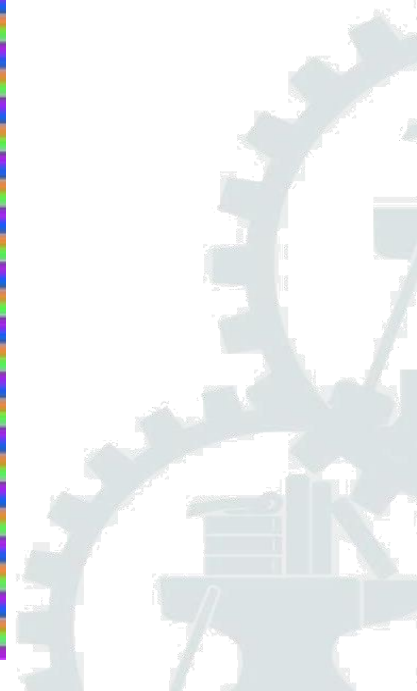
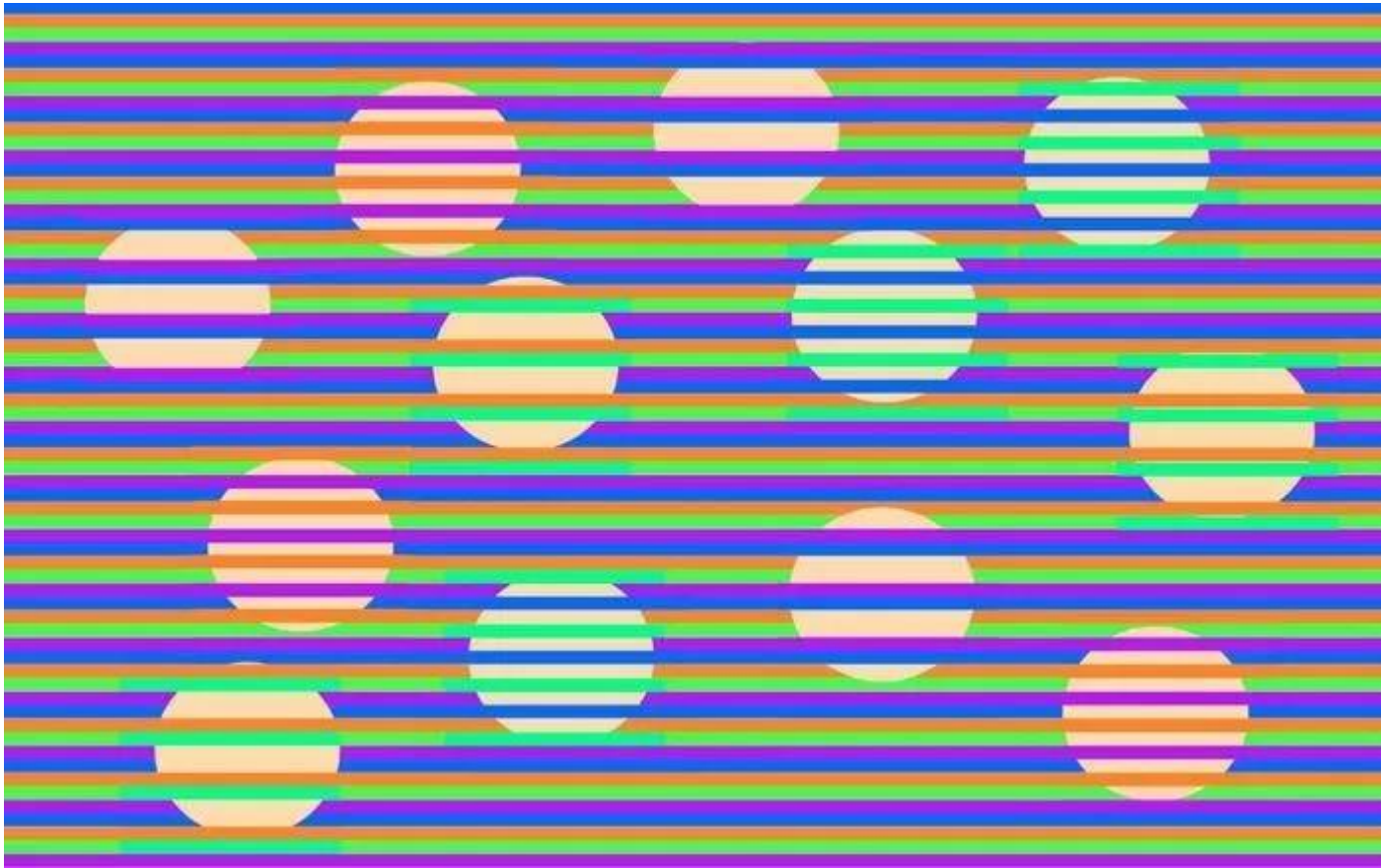
视觉假象

◇ Ebbinghaus Titchener illusion



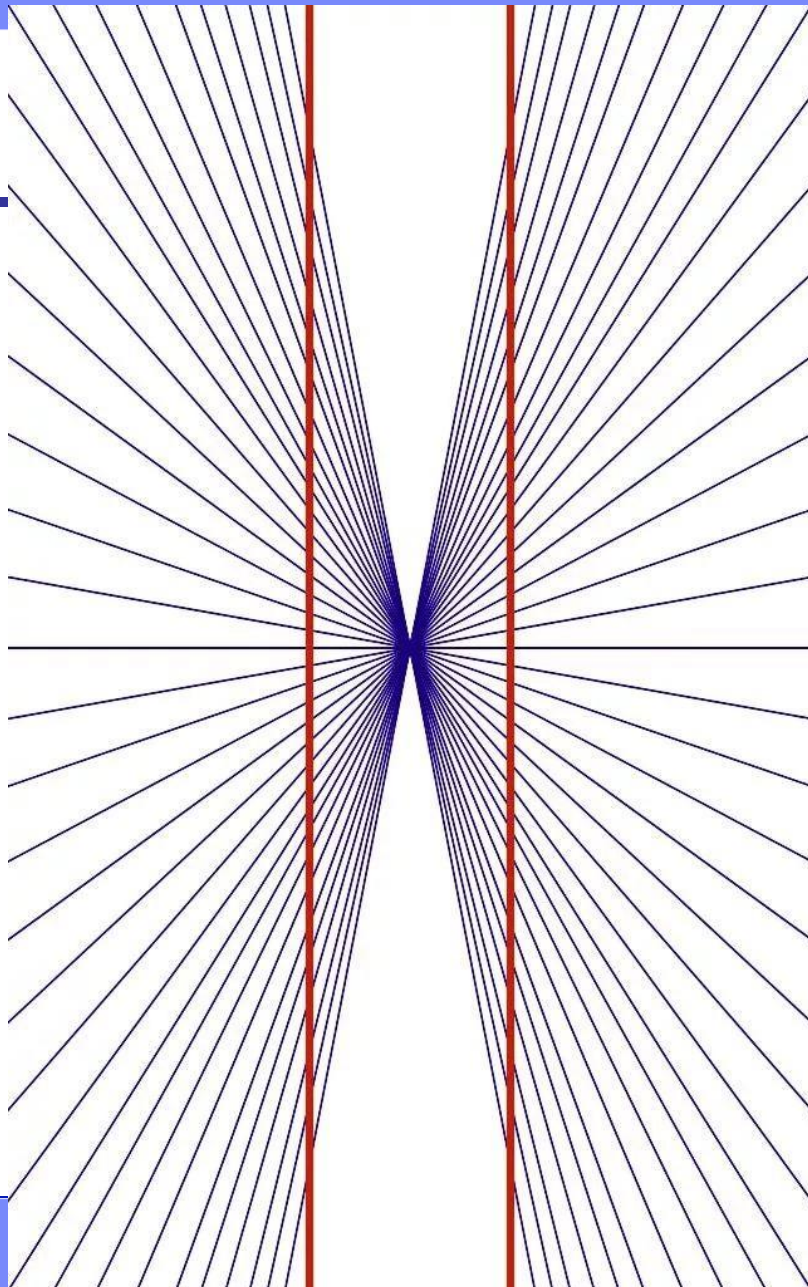


视觉假象



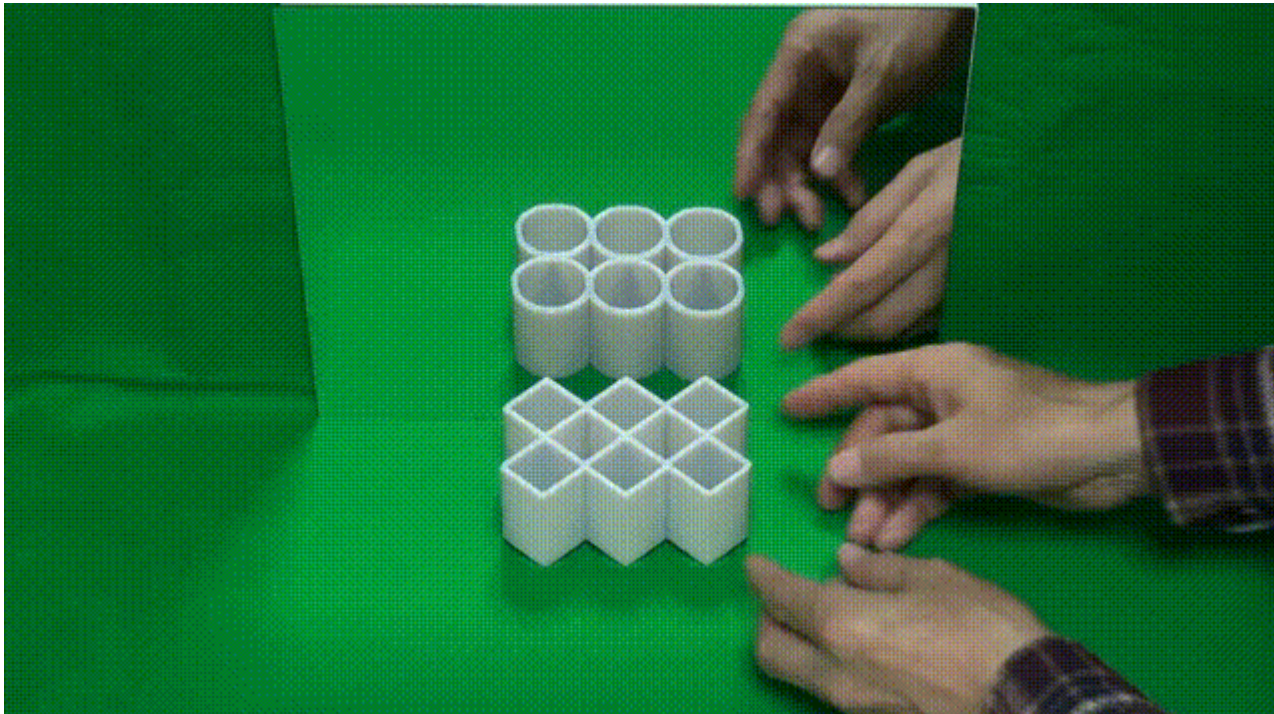


视觉假象



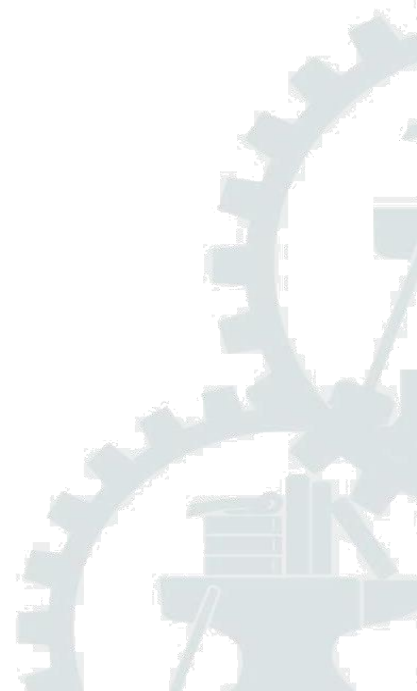


视觉假象



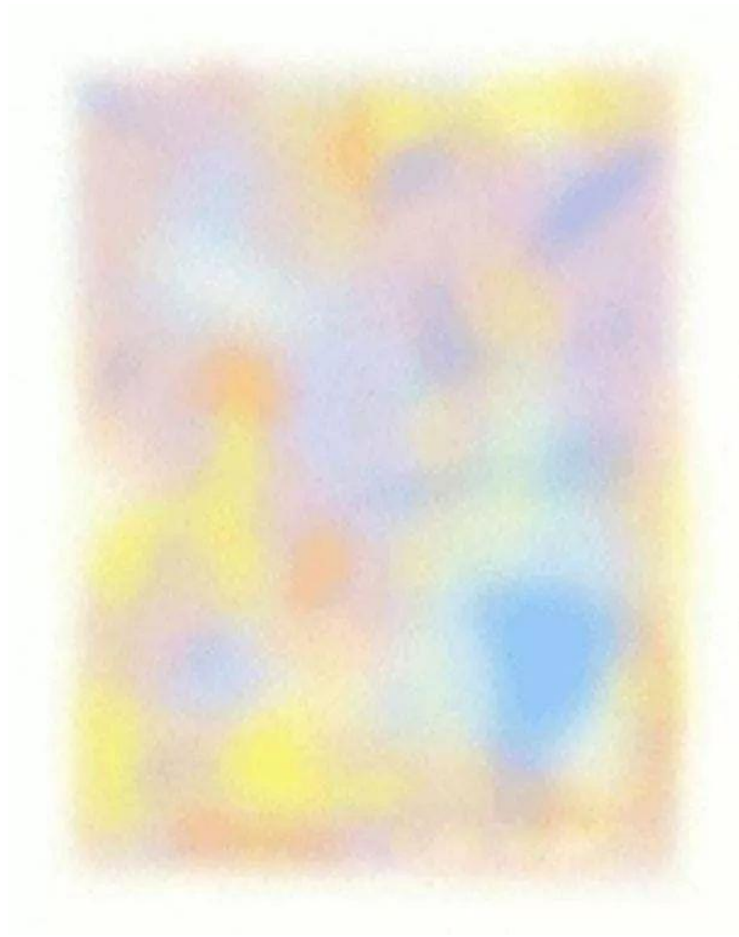


视觉假象





视觉假象





一般而言的视觉

- ◆ **视觉是我们最强的感知**。它为我们提供数量巨大的环境信息，使我们能够智能地与环境交互，且可以全然不直接与环境物理接触。无需惊奇，大量的力量都致力于为机器赋予视觉 (几乎从数字计算机技术起步时就开始了!)
- ◆ **视觉也是我们最复杂的感觉**。我们能够在照相纸上重建高分辨图像，但对于**大脑是如何处理从我们眼睛得到的信息的研究工作仍处于幼年期**。



基于视觉的传感器主要内容

- ◇ 基本成像原理（透视投影模型）

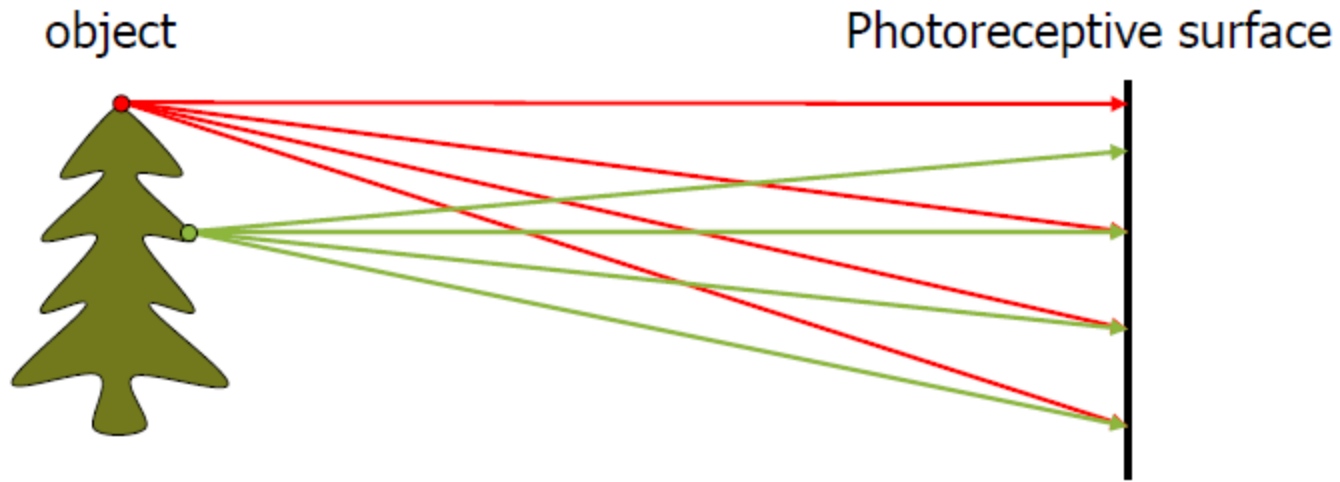
- ◇ 基本摄像机标定原理
 - ◇ 镜头畸变
 - ◇ 标定方法

- ◇ 视觉测距
 - ◇ 景深
 - ◇ 立体视觉

- ◇ 运动和光流
- ◇ 颜色跟踪传感器

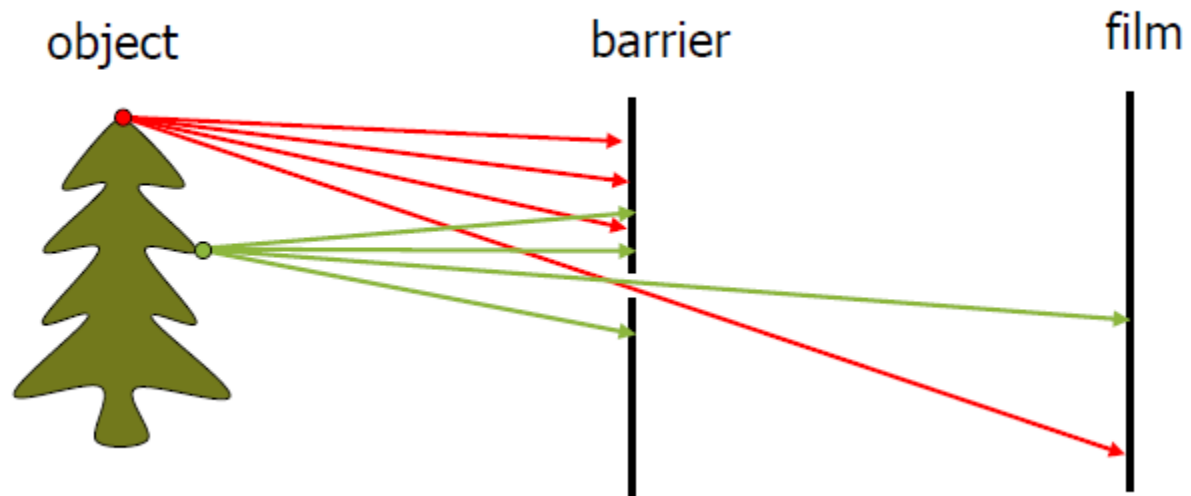


基本成像原理





基本成像原理





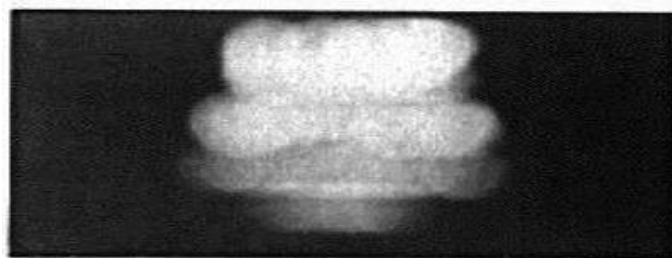
基本成像原理

◇ 自制针孔像机成像效果





基本成像原理



2 mm



1 mm



0.6mm



0.35 mm



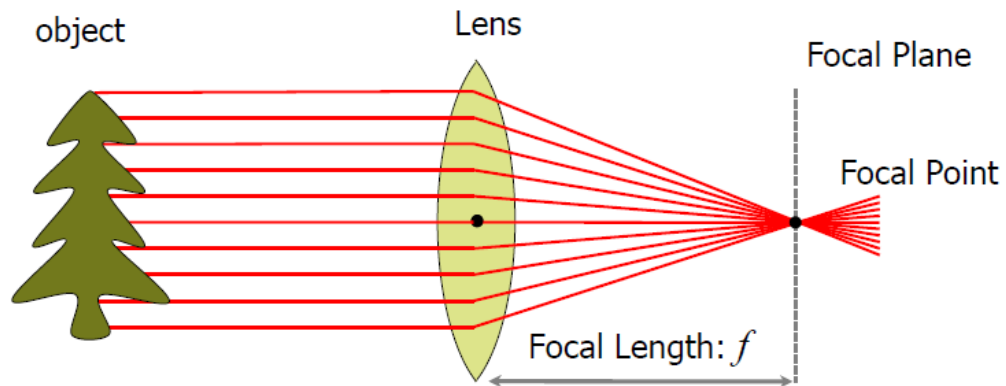
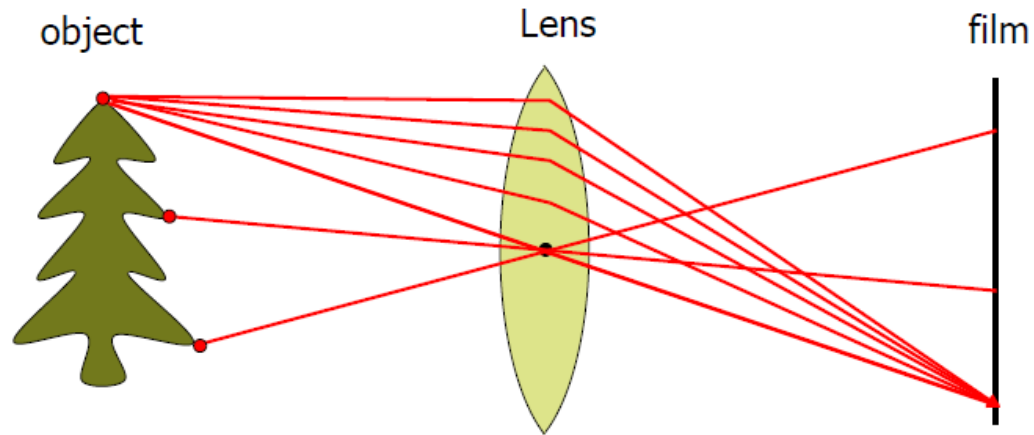
0.15 mm



0.07 mm



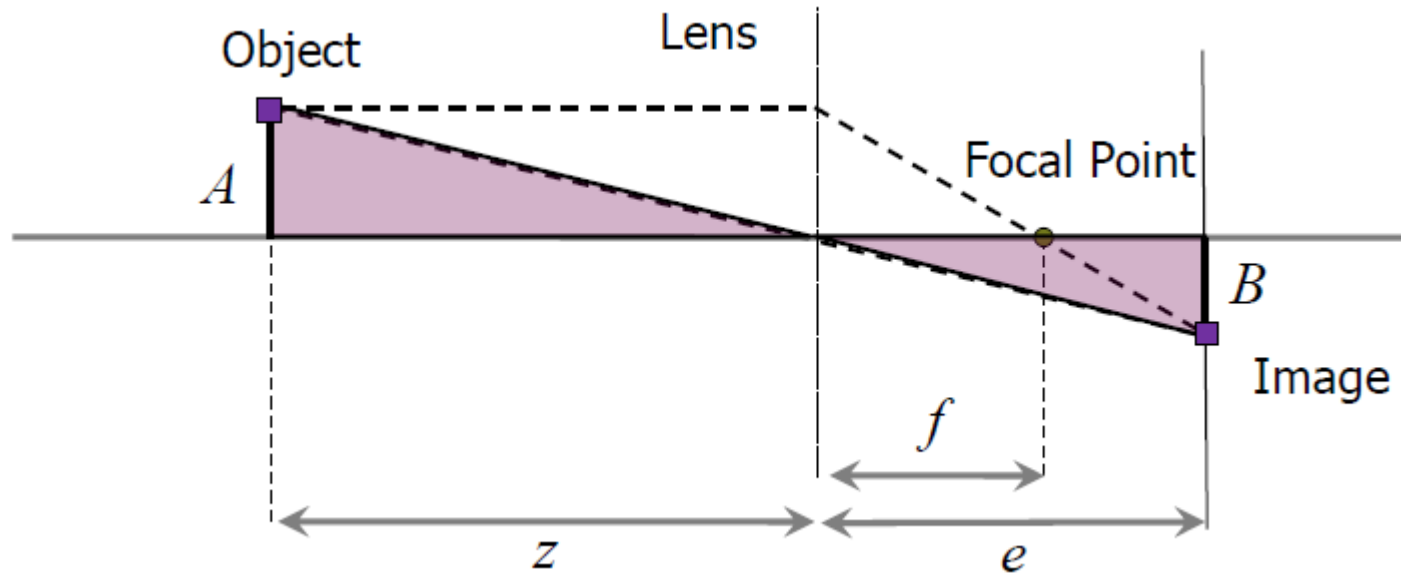
基本成像原理





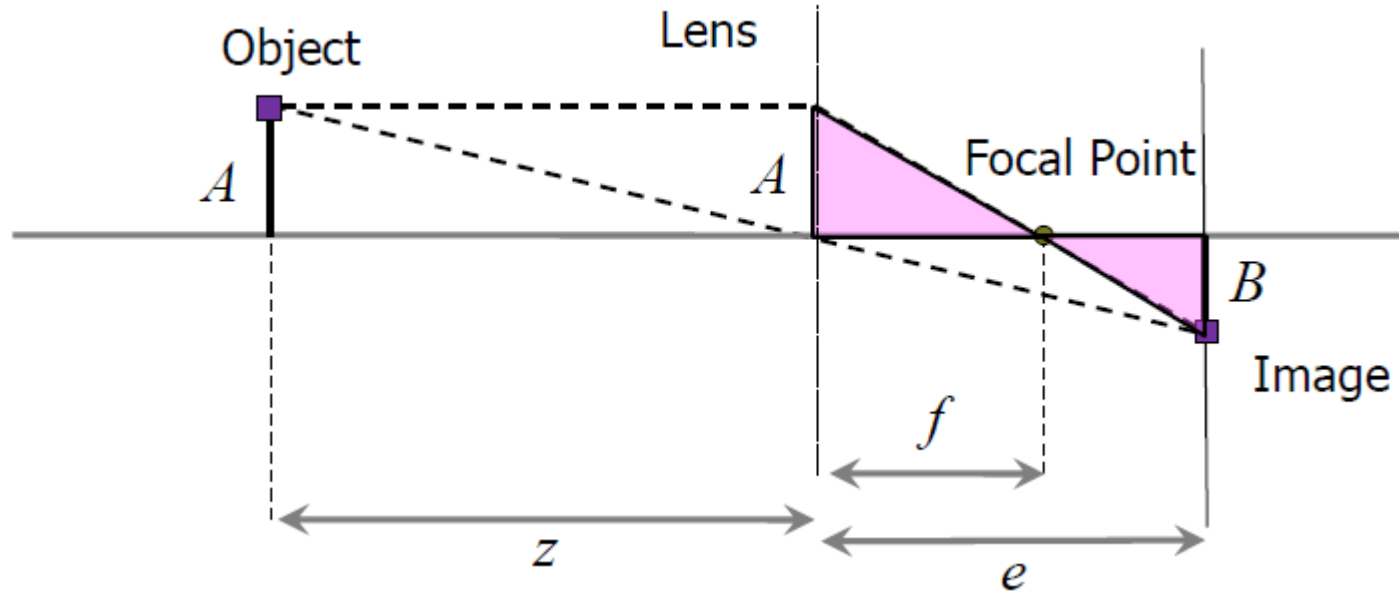
基本成像原理

$$\frac{B}{A} = \frac{e}{z}$$





基本成像原理

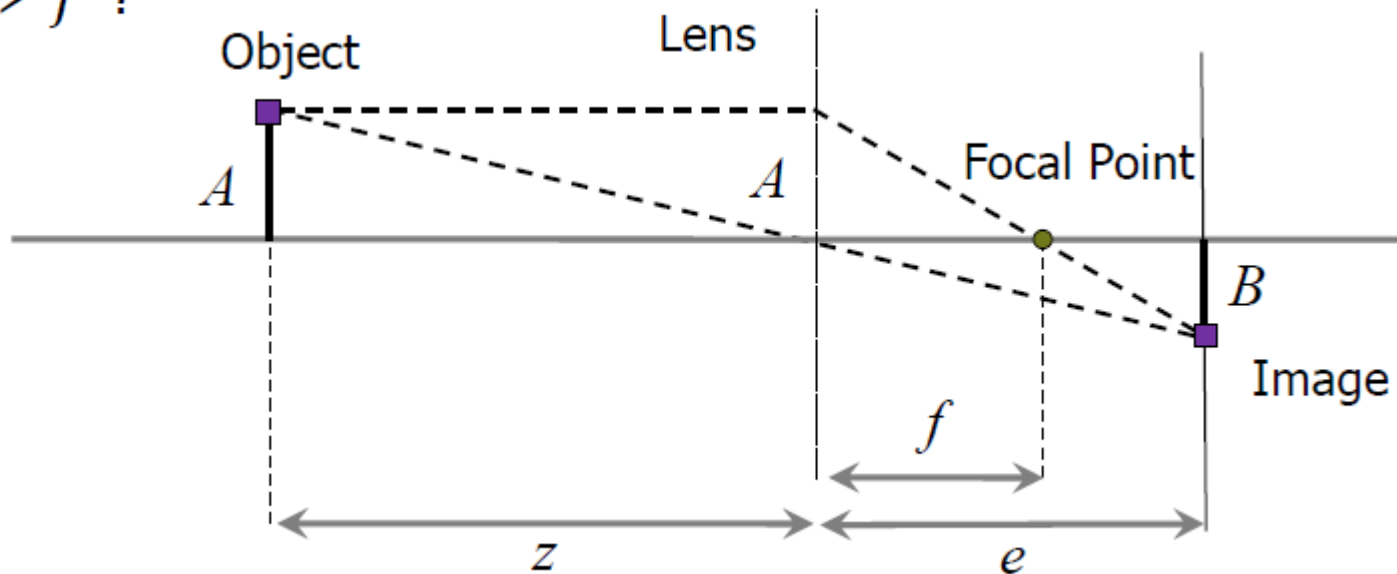


$$\left. \begin{aligned} \frac{B}{A} &= \frac{e}{z} \\ \frac{B}{A} &= \frac{e-f}{f} = \frac{e}{f} - 1 \end{aligned} \right\} \frac{e}{f} - 1 = \frac{e}{z}$$



基本成像原理

$z \gg f$?

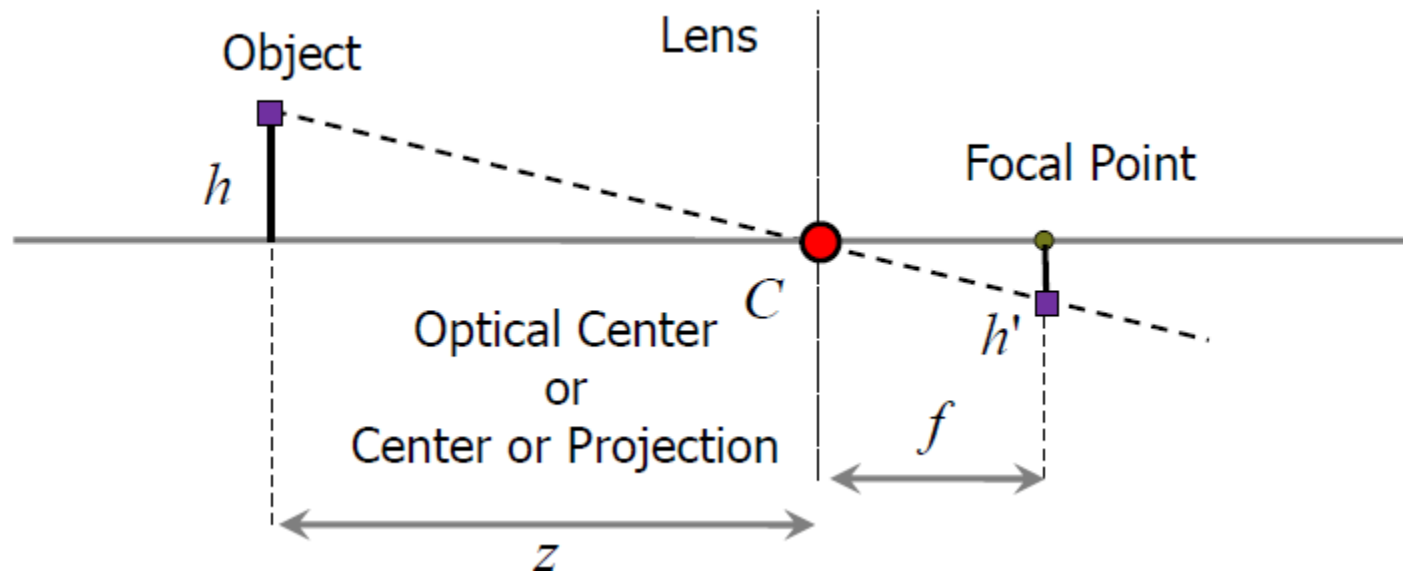


$$\frac{1}{f} = \underbrace{\frac{1}{z}}_{\cong 0} + \frac{1}{e} \Rightarrow \frac{1}{f} \approx \frac{1}{e} \Rightarrow f \approx e$$

需调整像平面的位置，
从而保证无穷远处的物
点能够在像平面上成像



基本成像原理

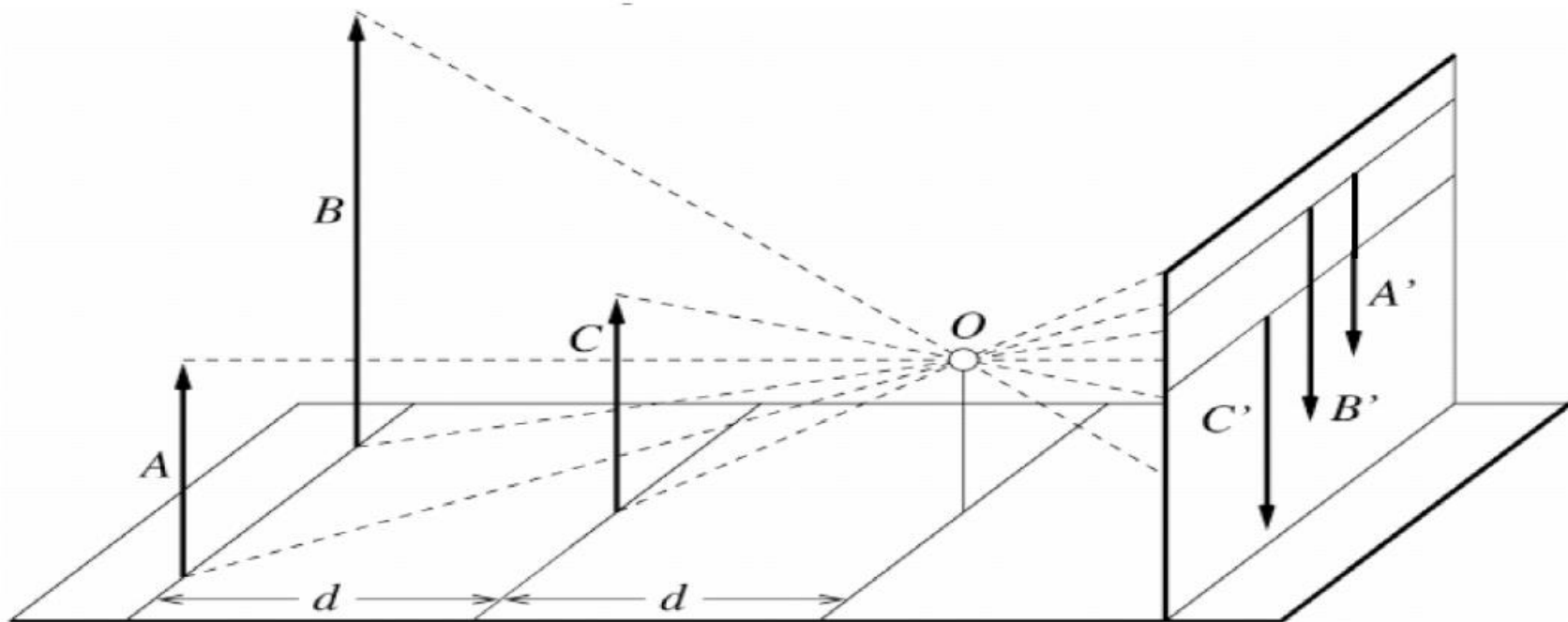


$$\frac{h'}{h} = \frac{f}{z} \Rightarrow h' = \frac{f}{z} h$$



基本成像原理

◇ 透视效果：远处的物体显得更小





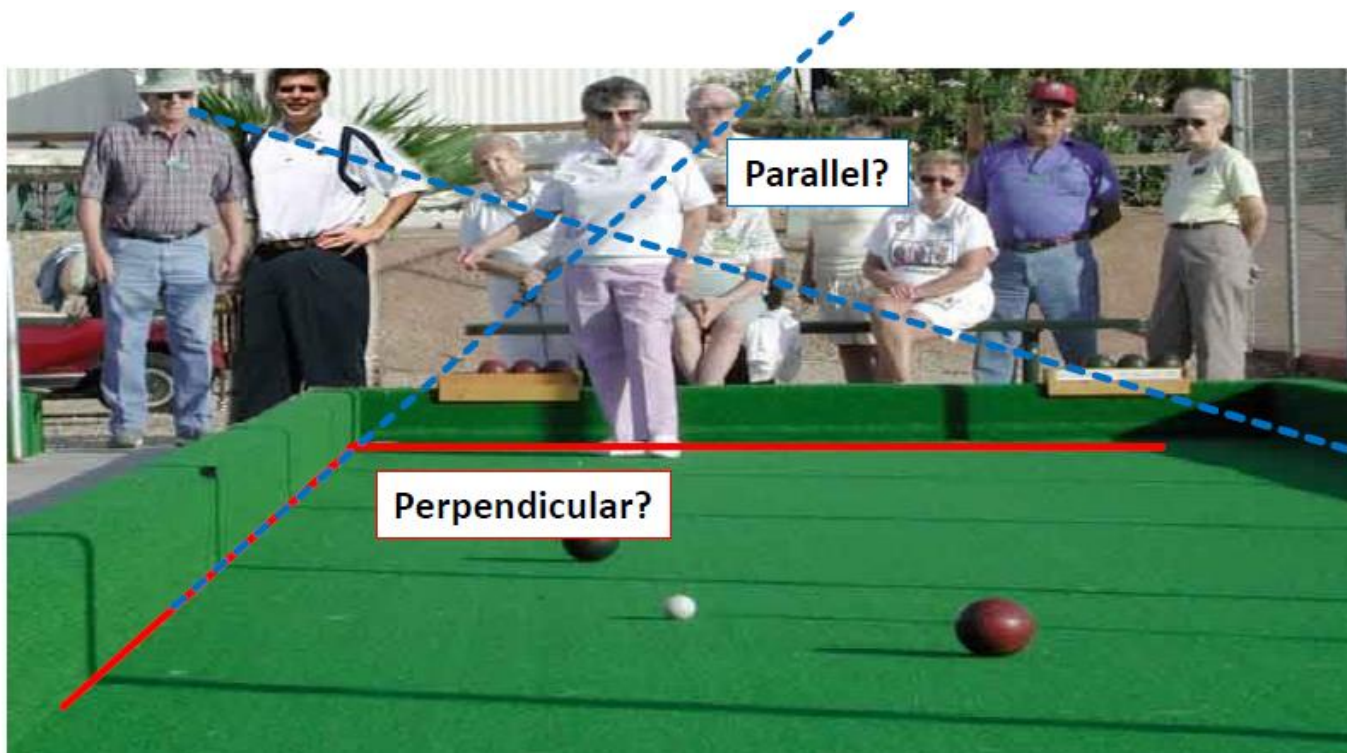
基本成像原理





基本成像原理

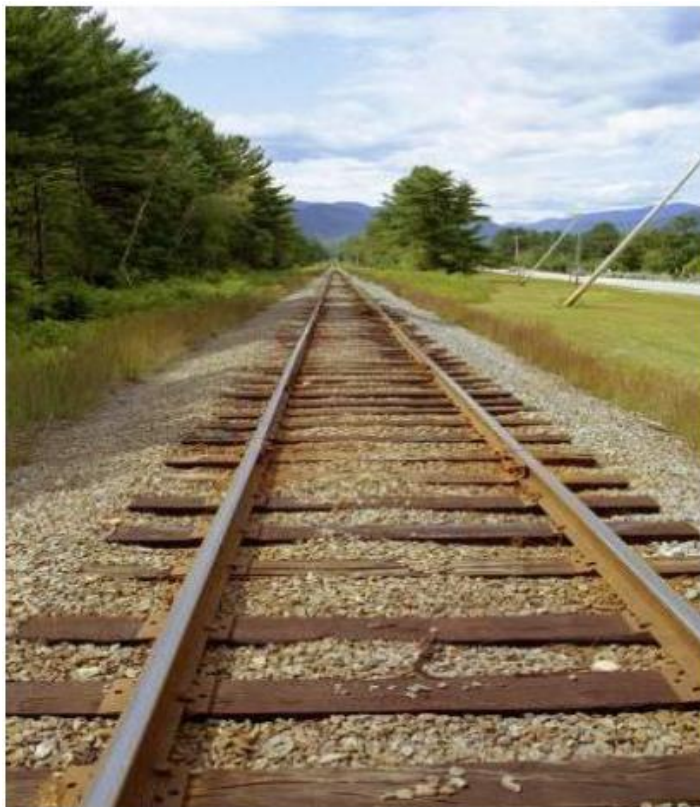
- ◇ 丢失的信息：长度、角度
- ◇ 保留的信息：直线仍然是直线





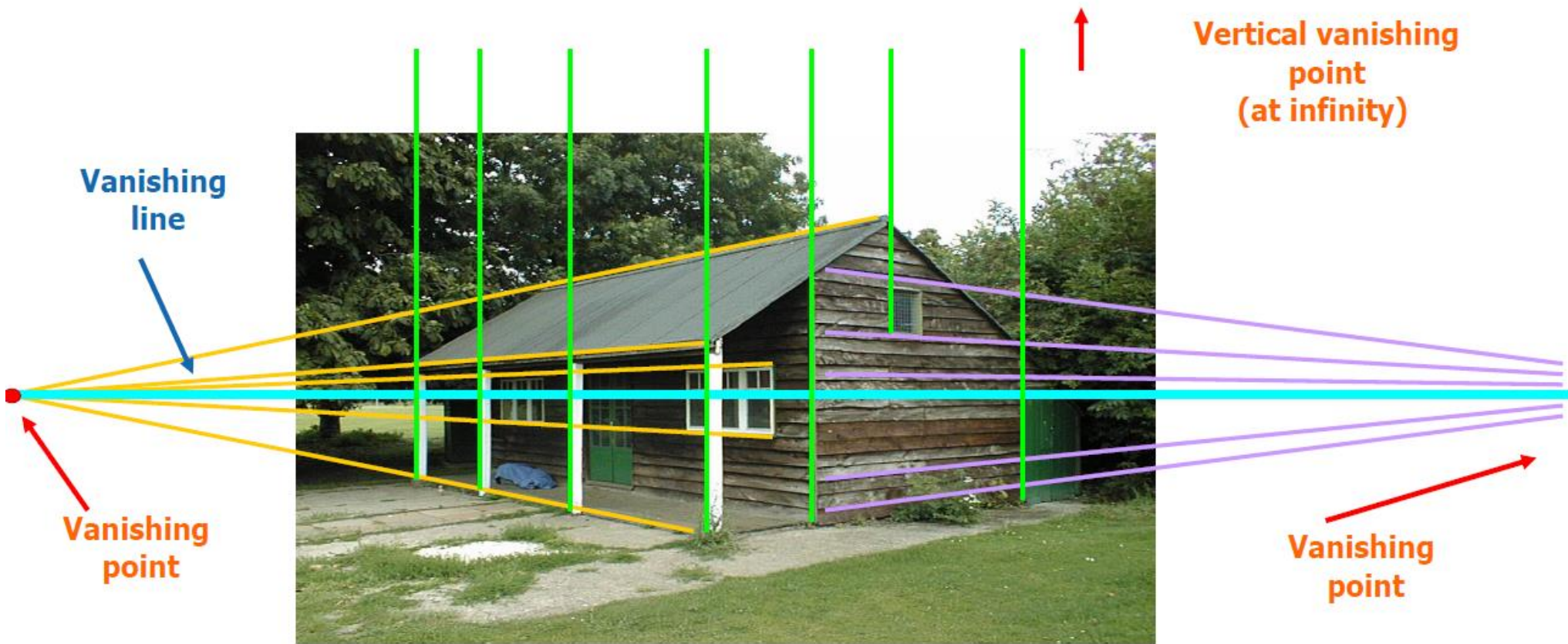
基本成像原理

◇ 消失点 (vanishing point)





基本成像原理



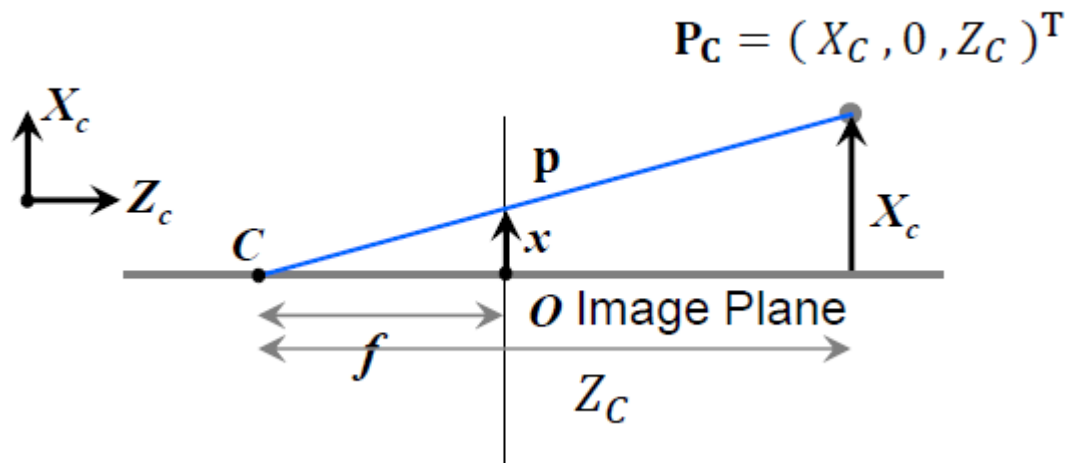


基本成像原理

◇ 物点 $\mathbf{P}_C = (X_c, 0, Z_c)^T$ 在像平面上的成像 $\mathbf{p} = (x, y)$

$$\frac{x}{f} = \frac{X_c}{Z_c} \Rightarrow x = \frac{fX_c}{Z_c}$$

$$\frac{y}{f} = \frac{Y_c}{Z_c} \Rightarrow y = \frac{fY_c}{Z_c}$$





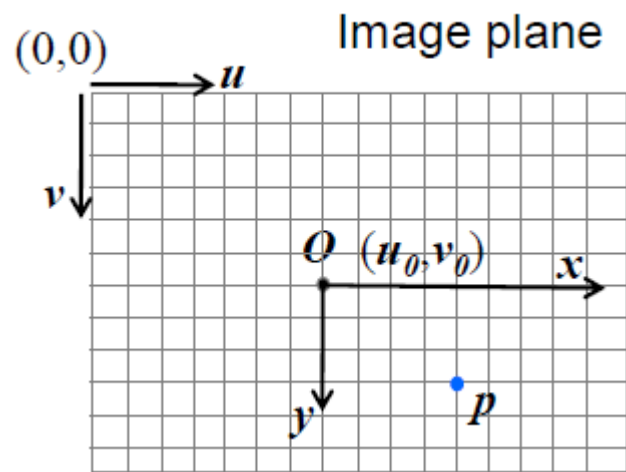
基本成像原理

◆ 将 p 点的坐标由图像平面内的坐标 (x, y) 转换为图像坐标 (u, v)

◆ 左上角为原点，考虑光学中心的位置 $u = u_0 + kx \Rightarrow u_0 + k\frac{fX_C}{Z_C}$

◆ 像素 $v = v_0 + ky \Rightarrow v_0 + k\frac{fY_C}{Z_C}$

$$O = (u_0, v_0)$$



$$p = \begin{pmatrix} u \\ v \end{pmatrix} \Rightarrow \tilde{p} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

转化为齐次坐标表示



$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & -f_u \cot \theta & u_0 \\ 0 & f_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ 1 \end{bmatrix}$$

K



基本成像原理

◇ 由物点到像素点的透视投影:

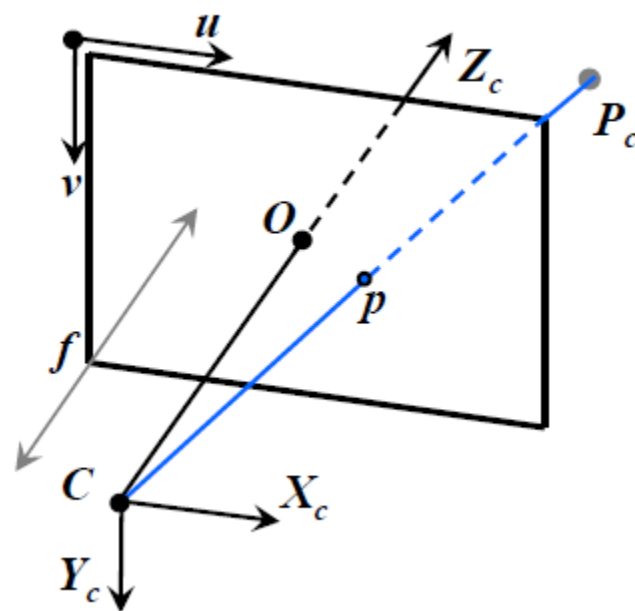
$$u = u_0 + kx \Rightarrow u_0 + k \frac{fX_c}{Z_c}$$

$$v = v_0 + ky \Rightarrow v_0 + k \frac{fY_c}{Z_c}$$

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} kf & 0 & u_0 \\ 0 & kf & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \alpha & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

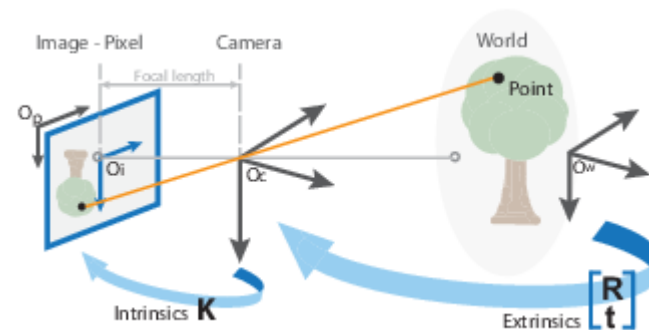
以像素衡量的焦距
内参矩阵





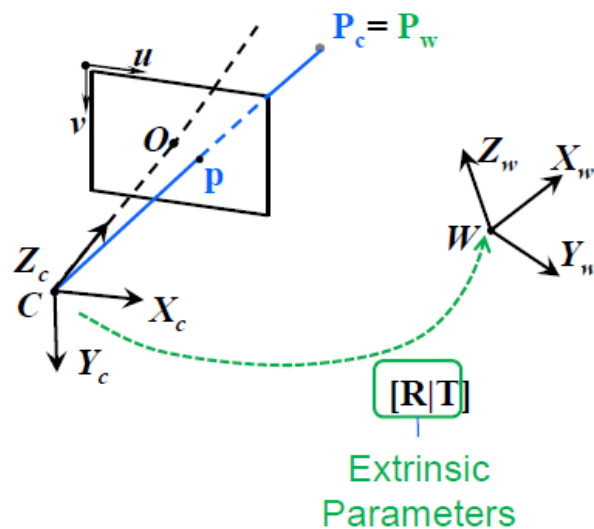
基本成像原理

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} R & | & T \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K [R|T] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$





镜头畸变

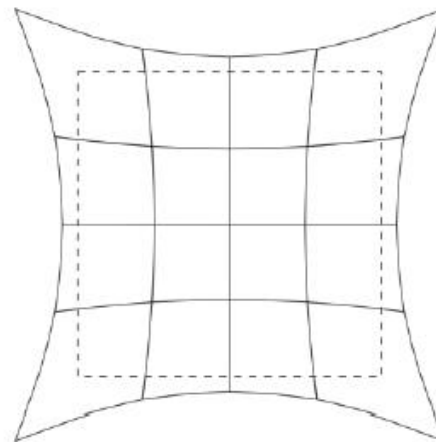
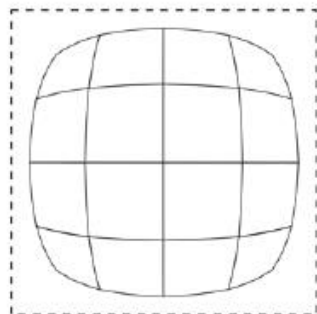
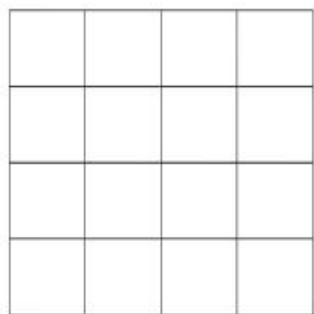
◇ 广角镜头拍摄的图像





镜头畸变

◇ 径向畸变



无畸变



桶形畸变



枕形畸变



镜头畸变

◇ k_1 为径向畸变参数，也属于摄像机内参

◇ 观测所得的含畸变坐标 (u_d, v_d) ，理想坐标 (u, v)

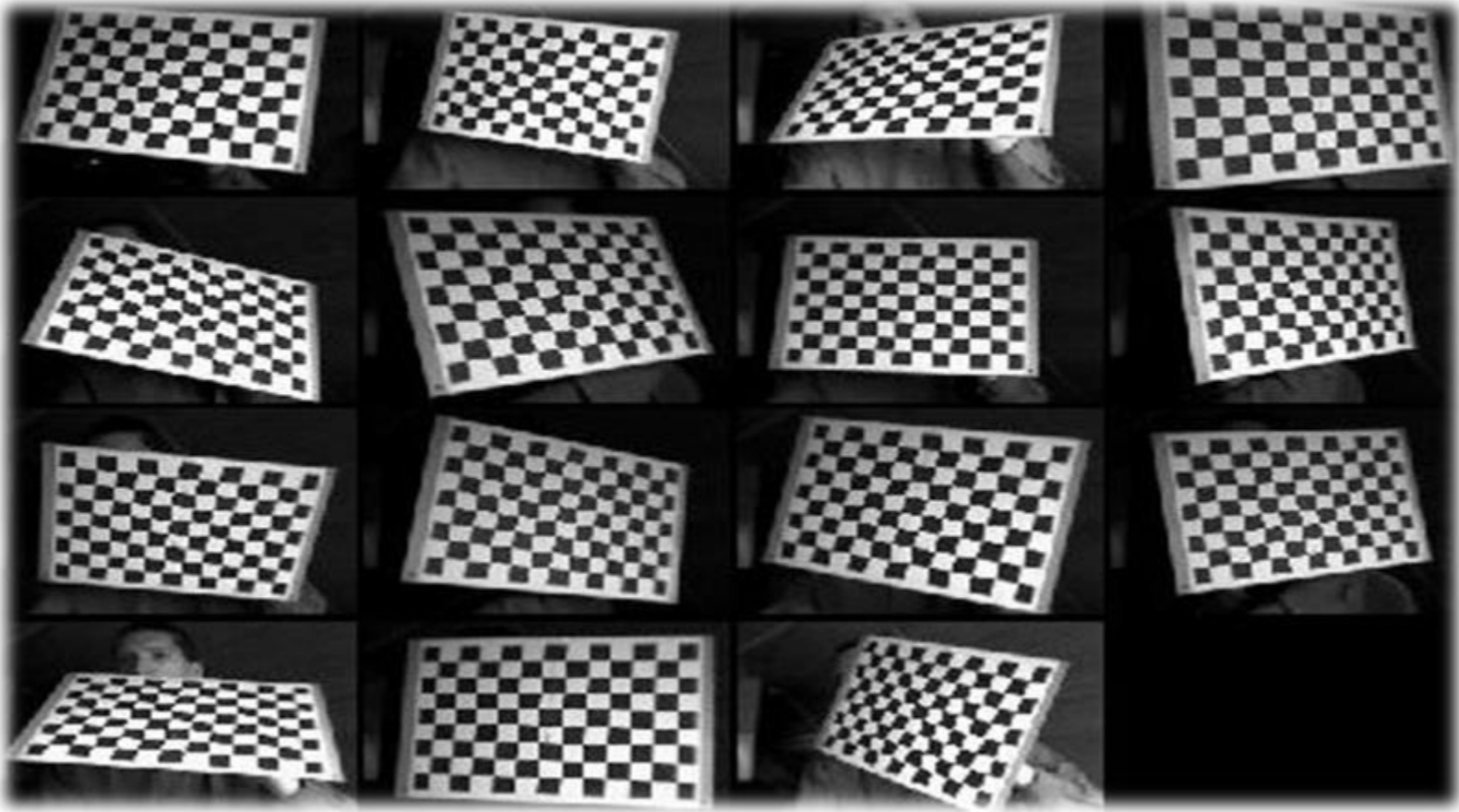
$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = (1 + k_1 r^2) \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

$$r^2 = (u - u_0)^2 + (v - v_0)^2$$

除径向畸变外，还包括切向畸变和法向畸变

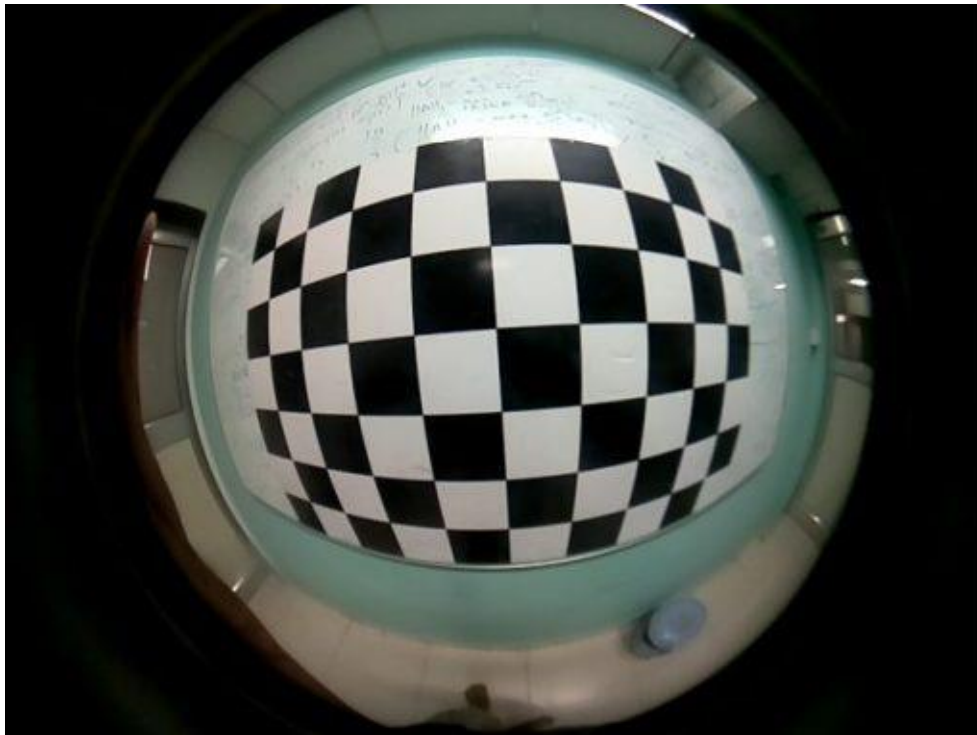


张正友标定法





张正友标定法





张正友标定法

1. 打印一张模板并贴在一个平面上
2. 从不同角度拍摄若干张模板图象
3. 检测出图象中的特征点
4. 求出摄像机的内参数和外参数
5. 求出畸变系数
6. 优化求精



相机标定实例

```
bool findChessboardCorners(InputArray image, Size patternSize, OutputArray corners,  
int flags = CALIB_CB_ADAPTIVE_THRESH + CALIB_CB_NORMALIZE_IMAGE )
```

```
void cornerSubPix(InputArray image, InputOutputArray corners, Size winSize, Size  
zeroZone, TermCriteria criteria)
```

```
double calibrateCamera(InputArrayOfArrays objectPoints, InputArrayOfArrays  
imagePoints, Size imageSize, InputOutputArray cameraMatrix, InputOutputArray  
distCoeffs, OutputArrayOfArrays rvecs, OutputArrayOfArrays tvecs)
```



相机标定实例

```
1 #include <opencv2/opencv.hpp>
2 #include <opencv2/calib3d/calib3d.hpp>
3 #include <opencv2/highgui/highgui.hpp>
4 #include <opencv2/imgproc/imgproc.hpp>
5 #include <stdio.h>
6 #include <iostream>
7
8 // Defining the dimensions of checkerboard
9 int CHECKERBOARD[2]{6,9};
10
11 int main()
12 {
13     // Creating vector to store vectors of 3D points for each checkerboard image
14     std::vector<std::vector<cv::Point3f> > objpoints;
15
16     // Creating vector to store vectors of 2D points for each checkerboard image
17     std::vector<std::vector<cv::Point2f> > imgpoints;
18
19     // Defining the world coordinates for 3D points
20     std::vector<cv::Point3f> objp;
21     for(int i{0}; i<CHECKERBOARD[1]; i++)
22     {
23         for(int j{0}; j<CHECKERBOARD[0]; j++)
24             objp.push_back(cv::Point3f(j,i,0));
25     }
26
```



相机标定实例

```
27
28 // Extracting path of individual image stored in a given directory
29 std::vector<cv::String> images;
30 // Path of the folder containing checkerboard images
31 std::string path = "./images/*.jpg";
32
33 cv::glob(path, images);
34
35 cv::Mat frame, gray;
36 // vector to store the pixel coordinates of detected checker board corners
37 std::vector<cv::Point2f> corner_pts;
38 bool success;
39
40 // Looping over all the images in the directory
41 for(int i{0}; i<images.size(); i++)
42 {
43     frame = cv::imread(images[i]);
44     cv::cvtColor(frame,gray,cv::COLOR_BGR2GRAY);
45
46     // Finding checker board corners
47     // If desired number of corners are found in the image then success = true
48     success = cv::findChessboardCorners(gray, cv::Size(CHECKERBOARD[0],
49     CHECKERBOARD[1]), corner_pts, CV_CALIB_CB_ADAPTIVE_THRESH |
49     CV_CALIB_CB_FAST_CHECK | CV_CALIB_CB_NORMALIZE_IMAGE);
```



相机标定实例

```
50     /*
51     * If desired number of corner are detected,
52     * we refine the pixel coordinates and display
53     * them on the images of checker board
54     */
55     if(success)
56     {
57         cv::TermCriteria criteria(CV_TERMCRIT_EPS | CV_TERMCRIT_ITER, 30, 0.001);
58
59         // refining pixel coordinates for given 2d points.
60         cv::cornerSubPix(gray,corner_pts,cv::Size(11,11),
61 cv::Size(-1,-1),criteria);
62
63         // Displaying the detected corner points on the checker board
64         cv::drawChessboardCorners(frame, cv::Size(CHECKERBOARD[0],
65 CHECKERBOARD[1]), corner_pts, success);
66
67         objpoints.push_back(objp);
68         imgpoints.push_back(corner_pts);
69     }
70
71     cv::imshow("Image",frame);
72     cv::waitKey(0);
73 }
74
75 cv::destroyAllWindows();
76
77 cv::Mat cameraMatrix,distCoeffs,R,T;
```



相机标定实例

```
77  /*
78  * Performing camera calibration by
79  * passing the value of known 3D points (objpoints)
80  * and corresponding pixel coordinates of the
81  * detected corners (imgpoints)
82  */
83  cv::calibrateCamera(objpoints, imgpoints, cv::Size(gray.rows,gray.cols),
cameraMatrix, distCoeffs, R, T);
84
85  std::cout << "cameraMatrix : " << cameraMatrix << std::endl;
86  std::cout << "distCoeffs : " << distCoeffs << std::endl;
87  std::cout << "Rotation vector : " << R << std::endl;
88  std::cout << "Translation vector : " << T << std::endl;
89
90  return 0;
91 }
```



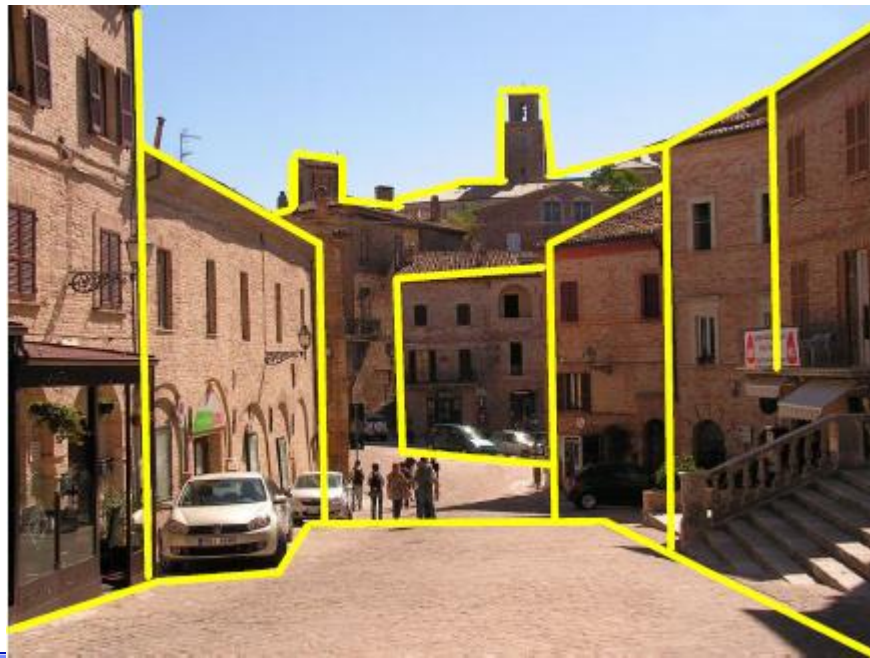
一般而言的视觉

- ◆ 当摄像机记录下一幅图像，**3D 景象**被投影到一个**2D 平面**上 (胶片、光敏阵列)。
- ◆ 为了尝试恢复一些景色中的“有用信息”，通常采用
 - ◆ **边缘检测法**寻找目标的轮廓。
 - ◆ **重建3D景象**所必需的所有信息
 - ◆ 即便对于简单情况，所找到的边缘片断也是不完全的，若想将它们综合形成一条描述目标边缘的清晰线，需要仔细的处理过程。根据**2D 图像**解释**3D 景象**不是轻而易举的工作。然而，采用立体成像或三角测量技术，视觉可以成为一种捕获环境的有利工具。



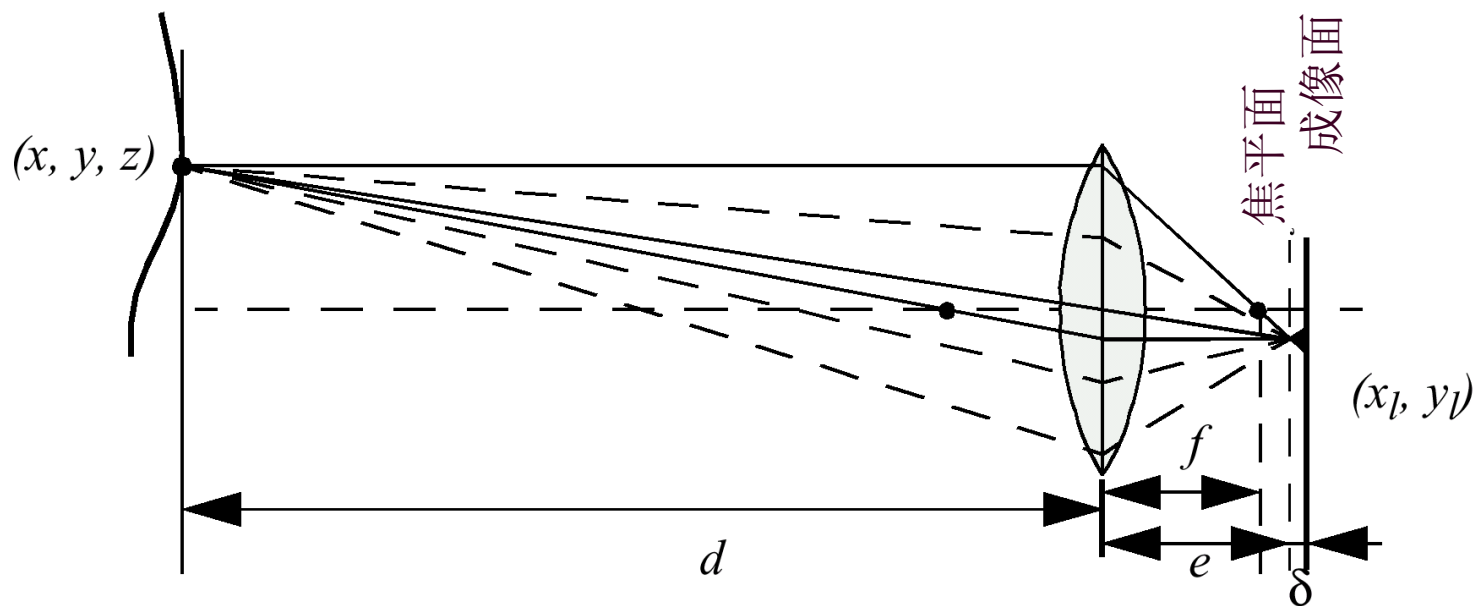
与机器人相关的计算机视觉

- ◆ 自动提取图像中有意义的信息(如语义分割、图像区域的几何分割)等
- ◆ 来源于图像的距离信息（深度信息）





景深 (1)



$$\frac{1}{f} = \frac{1}{d} + \frac{1}{e}$$

$$R = \frac{L\delta}{2e}$$



利用景深进行深度测量 (1) — 锐度比较



◇ 子图像梯度的两种测量:

最大锐度对应正确聚焦

需要搜索，比较费时

$$sharpness_1 = \sum_{x,y} |I(x,y) - I(x-1,y)|$$

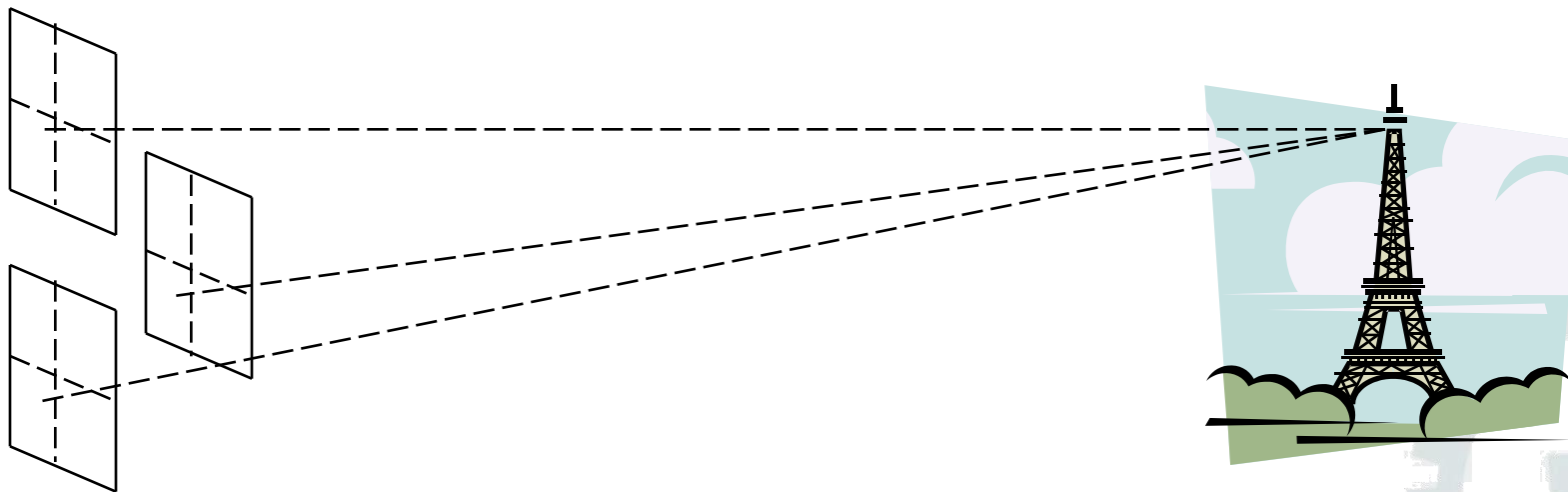
$$sharpness_2 = \sum_{x,y} (I(x,y) - I(x-2,y-2))^2$$

避免隔行扫描带来的偏差



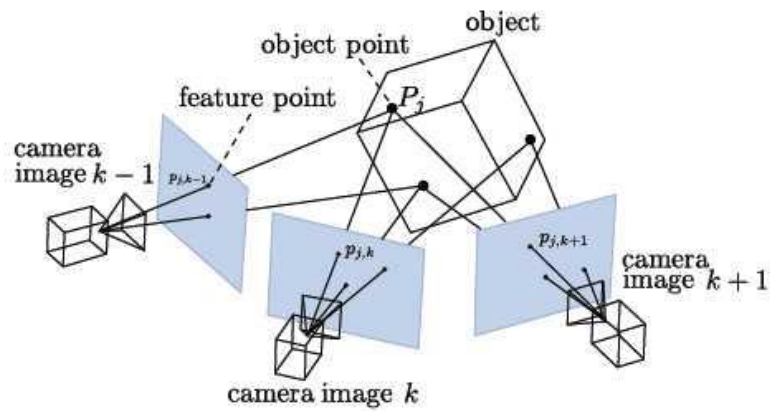
利用景深进行深度测量

- ◆ 用3台摄像机对同一场景拍摄，三台摄像机聚焦分别为远、中、近。这样有利于得到搜索的方向，提高深度测量效率。
- ◆ 对3台摄像机拍摄的图像划分多个子区域，分别对应相同场景。分别对子区域深度进行寻优，可以得到粗糙的场景深度信息。





立体重建





立体视觉

◆ 对于立体视觉，理想化的摄像机几何特性

◆ 两副图像的不一致 -> 计算深度

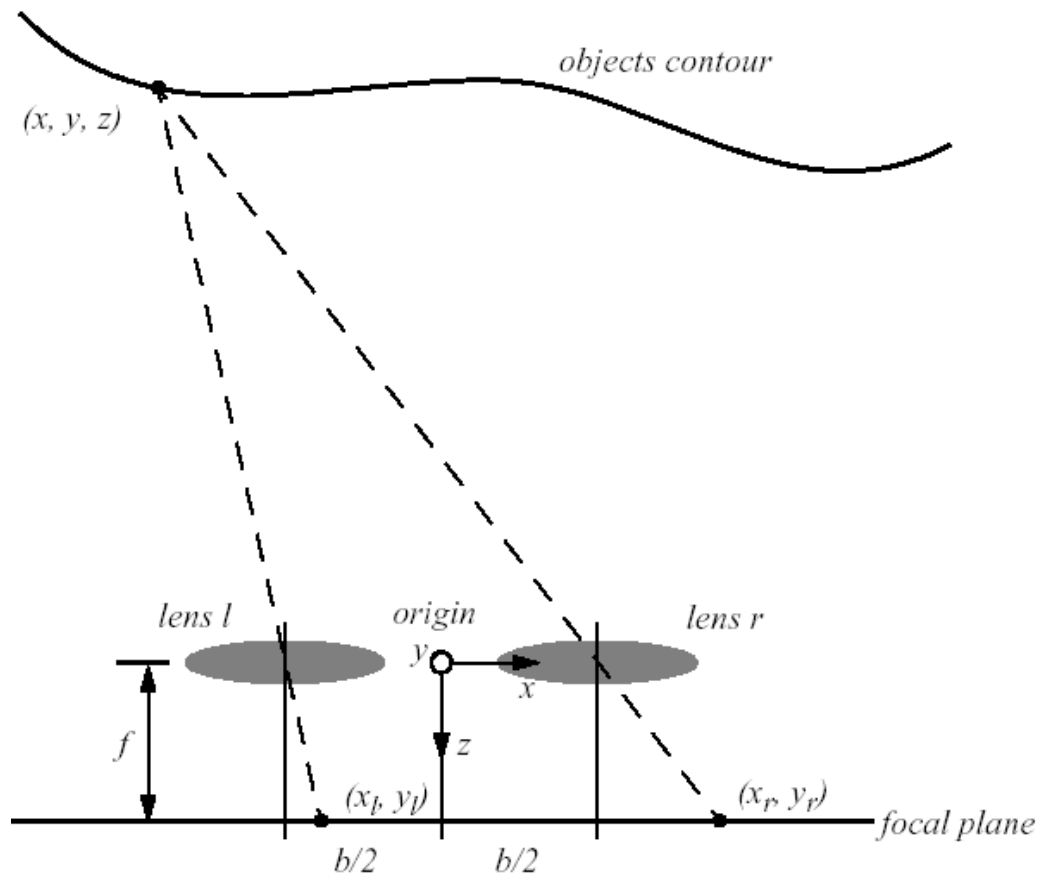
◆ 从图中，可以看到

$$\frac{x_l}{f} = \frac{x + b/2}{z} \quad \text{and} \quad \frac{x_r}{f} = \frac{x - b/2}{z}$$

$$\frac{x_l - x_r}{f} = \frac{b}{z} \quad \frac{y_l}{f} = \frac{y_r}{f} = \frac{y}{z}$$

$$x = b \frac{(x_l + x_r)/2}{x_l - x_r} ; \quad y = b \frac{(y_l + y_r)/2}{x_l - x_r}$$

$$z = b \frac{f}{x_l - x_r}$$





立体视觉

1. 距离与“视差”成反比

- ◆ 近物体测量更准确

2. “视差”与 b （基线）成正比

- ◆ 深度估计的精度随基线 b 的增加而增加。
- ◆ 当 b 增加时，某些物体或许会只出现在一个摄像机视野中。

3. 两个摄像机都可见的点，产生一个共轭对。

- ◆ 共轭对处在表偏振线上 (表偏振线平行于 x -轴)。



立体视觉 – 一般情况

◇ 同一点 P 分别被左右摄像机测量:

$$r'_r = R \cdot r'_l + r_0$$

$$\begin{bmatrix} x'_r \\ y'_r \\ z'_r \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{21} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x'_l \\ y'_l \\ z'_l \end{bmatrix} + \begin{bmatrix} r_{01} \\ r_{02} \\ r_{03} \end{bmatrix}$$

◇ 其中,

- ◇ R 是 3×3 旋转矩阵
- ◇ r_0 = 偏移向量

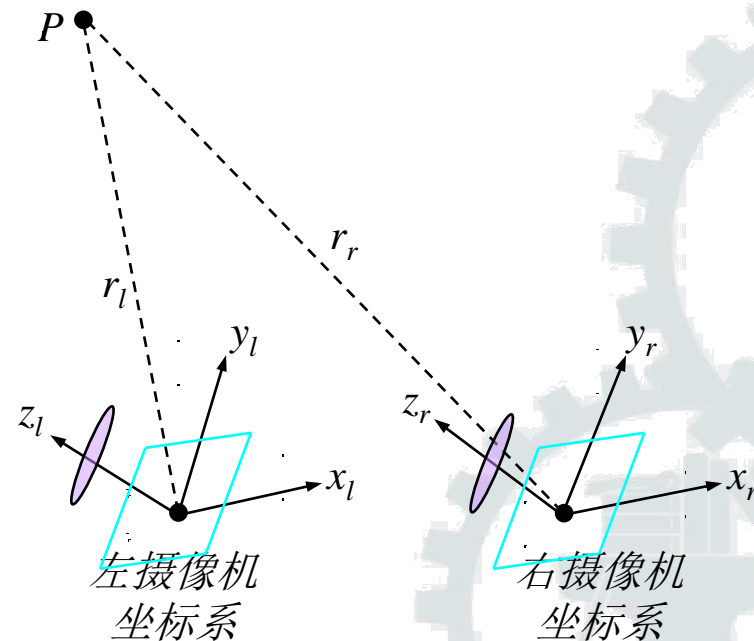
同一点在两个坐标系中表示之间的关系

◇ 注释:

- ◇ 对于完全对准的两台摄像机, $R=I$ (单位阵)
- ◇ 给定对应的 x_l, y_l, z_l, x_r, y_r 和 z_r , 我们可以校准系统, 找到 $r_{11}, r_{12} \dots$

◇ 我们有12个未知, 需要12个方程

- ◇ 为完全校正, 需要4个共轭对
- ◇ 注: 另外, 还有图像的光学畸变





立体视觉 – 一般情况

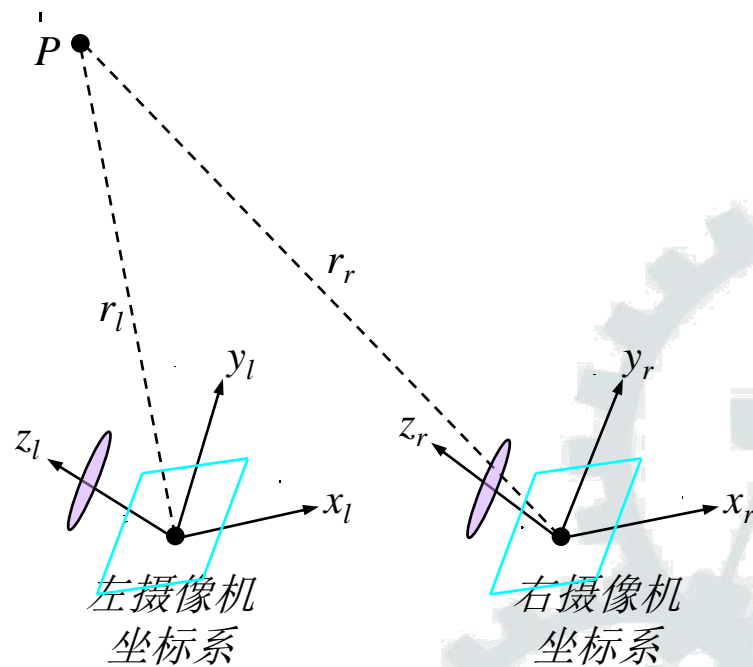
◇ 关于校准:

◇ 问题: 根据四个以上共轭对相对两个摄像机的表示:

$$(x'_l \ y'_l \ z'_l)_i, (x'_r \ y'_r \ z'_r)_i, i = 1, 2, 3, 4, \Lambda$$

求解二摄像机的参数:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \begin{bmatrix} r_{01} \\ r_{02} \\ r_{03} \end{bmatrix}$$



◇ 此校准与图像无关, 只是获取坐标变换关系 (平移、旋转) !



立体视觉 – 一般情况

◇ 将

$$\begin{bmatrix} x'_r \\ y'_r \\ z'_r \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x'_l \\ y'_l \\ z'_l \end{bmatrix} + \begin{bmatrix} r_{01} \\ r_{02} \\ r_{03} \end{bmatrix}$$

$$r = [r_{11} \quad r_{12} \quad r_{13} \quad r_{21} \quad r_{22} \quad r_{23} \quad r_{31} \quad r_{32} \quad r_{33} \quad r_{01} \quad r_{02} \quad r_{03}]^T$$

◇ 写成关于待估计向量

的方程:

$$\begin{bmatrix} x'_{li} & y'_{li} & z'_{li} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & x'_{li} & y'_{li} & z'_{li} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x'_{li} & y'_{li} & z'_{li} & 0 & 0 & 1 \end{bmatrix} r = \begin{bmatrix} x'_{ri} \\ y'_{ri} \\ z'_{ri} \end{bmatrix}$$

OR

$$\Phi_i r = Y_i$$



立体视觉 – 一般情况

◇ 用 4 组共轭对，构成理论上可解的方程形式：

$$\begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \end{bmatrix} r = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix}$$

◇ 如考虑到测量误差，应采用最小二乘等方法求解。变为求以下极小问题：

$$\min_r J = \sum_{i=1}^N \|\Phi_i r - Y_i\|^2$$

◇ N 可以远大于 4 !



立体视觉 – 一般情况

◇ 求解:

◇ 将
$$\sum_{i=1}^N \|\Phi_i r - Y_i\|^2 = \sum_{i=1}^N (\Phi_i r - Y_i)^T (\Phi_i r - Y_i)$$

◇ 写成
$$\left\| \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_N \end{bmatrix} r - \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix} \right\|^2 = \|\Phi r - Y\|^2, \quad \Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_N \end{bmatrix}, \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix}$$

◇ 当 $N > 4$, 这是一个超定方程。最优解为: $r = (\Phi^T \Phi)^{-1} \Phi^T Y$



立体视觉 – 一般情况

◇ 证明: $2J = (Y - \Phi r)^T (Y - \Phi r)$

$$= Y^T Y - Y^T \Phi r - r^T \Phi^T Y + r^T \Phi^T \Phi r$$

◇ 写成 $2 \frac{\partial J}{\partial r} = -2\Phi^T Y + 2\Phi^T \Phi r = 0 \Rightarrow r = (\Phi^T \Phi)^{-1} \Phi^T Y$

◇ 证明极值点是极小点: $\frac{\partial^2 J}{\partial r^2} = \Phi^T \Phi > 0$



立体视觉 – 一般情况

◇ 假设校准步骤完整，求解距离恢复问题：

◇ 问题：根据已校准的二摄像机参数：

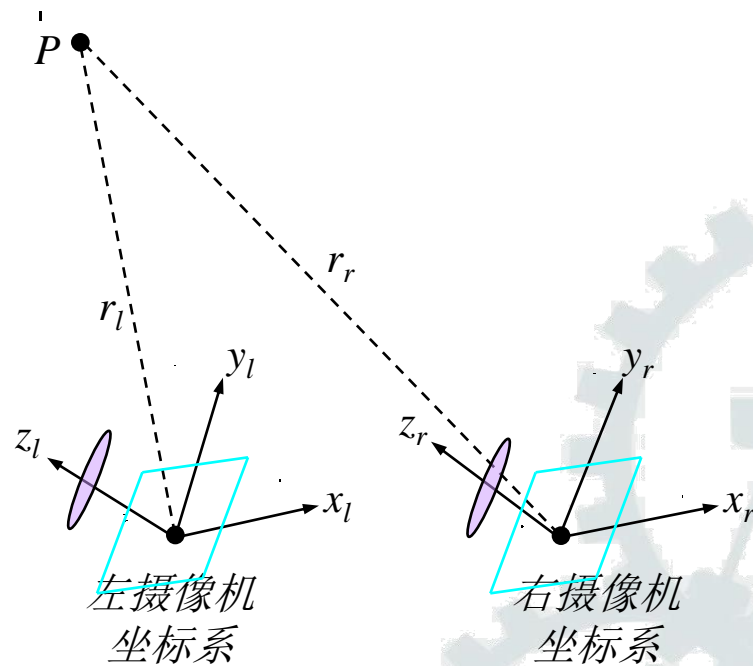
$$f, \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \begin{bmatrix} r_{01} \\ r_{02} \\ r_{03} \end{bmatrix}$$

以及一点 P 在两个摄像机成像位置：

$$(x_l \ y_l), (x_r \ y_r)$$

求该点分别相对两个摄像机的位置

$$(x'_l \ y'_l \ z'_l), (x'_r \ y'_r \ z'_r)$$





立体视觉 – 一般情况

◇ 按右图，根据三角形相似性，得：

$$\frac{x_l}{f} = \frac{x'_l}{z'_l}, \quad \frac{y_l}{f} = \frac{y'_l}{z'_l}$$

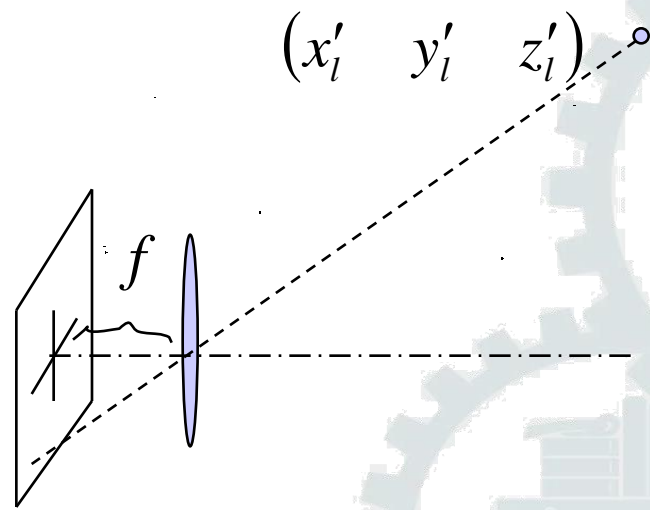
◇ 这样可把 x'_l, y'_l 表示为 z'_l 的函数：

$$x'_l = \frac{x_l}{f} z'_l, \quad y'_l = \frac{y_l}{f} z'_l$$

$$\begin{pmatrix} x'_l & y'_l & z'_l \end{pmatrix} = z'_l \begin{pmatrix} \frac{x_l}{f} & \frac{y_l}{f} & 1 \end{pmatrix}$$

◇ 同理对右摄像机

$$\begin{pmatrix} x'_r & y'_r & z'_r \end{pmatrix} = z'_r \begin{pmatrix} \frac{x_r}{f} & \frac{y_r}{f} & 1 \end{pmatrix} \begin{pmatrix} x_l & y_l \end{pmatrix}$$





立体视觉 – 一般情况

◇ 代入方程组后得(4.32), (4.33), (4.34):

$$\left(r_{11} \frac{x_l}{f} + r_{12} \frac{y_l}{f} + r_{13} \right) z'_l + r_{01} = \frac{x_r}{f} z'_r$$

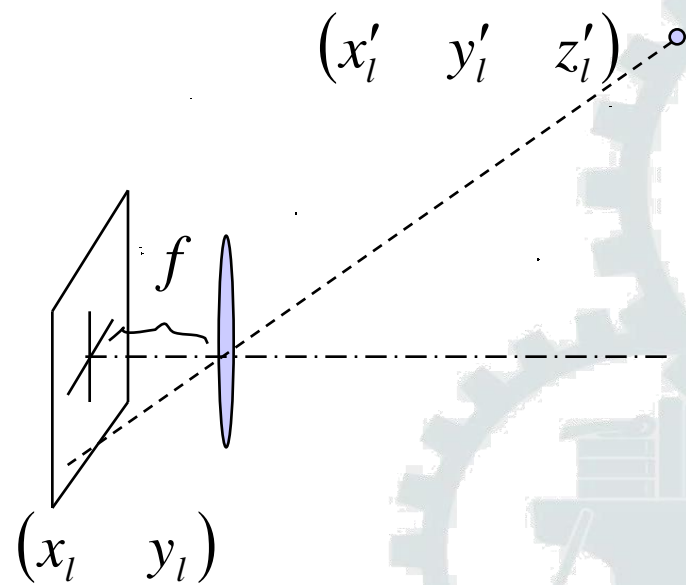
$$\left(r_{21} \frac{x_l}{f} + r_{22} \frac{y_l}{f} + r_{23} \right) z'_l + r_{02} = \frac{x_r}{f} z'_r$$

$$\left(r_{31} \frac{x_l}{f} + r_{32} \frac{y_l}{f} + r_{33} \right) z'_l + r_{03} = z'_r$$

◇ 从以上三个方程中任意两个就可以解出深度信息 z'_l 和 z'_r

◇ 从而计算得到

$$(x'_l \quad y'_l \quad z'_l), (x'_r \quad y'_r \quad z'_r)$$





立体视觉 – 一般情况的进一步讨论

◆ 同一点 P 分别被左右摄像机测量:

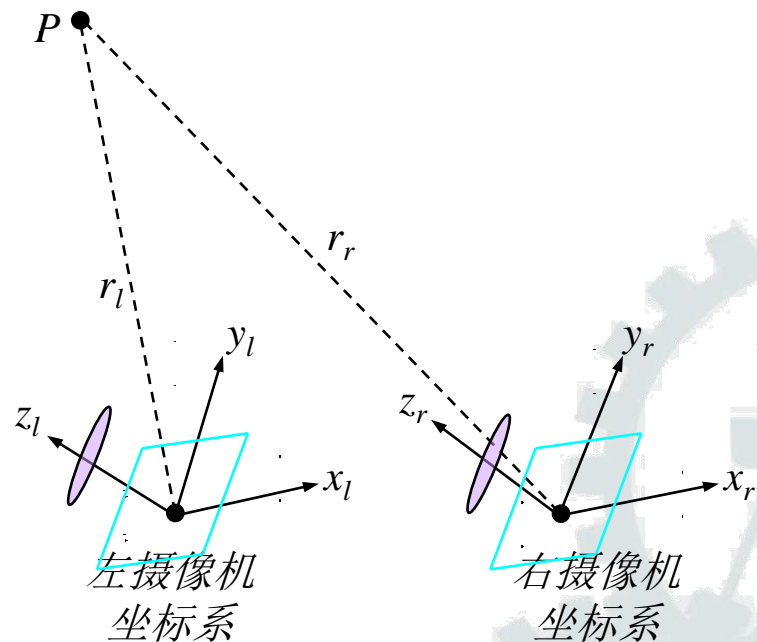
$$r'_r = R \cdot r'_l + r_0 \quad \begin{bmatrix} x'_r \\ y'_r \\ z'_r \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{21} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x'_l \\ y'_l \\ z'_l \end{bmatrix} + \begin{bmatrix} r_{01} \\ r_{02} \\ r_{03} \end{bmatrix}$$

◆ 根据旋转阵 R 和偏移向量 r_0

后, 可计算深度值 z'_l, z'_r :

$$\begin{cases} \left(r_{11} \frac{x_l}{f} + r_{12} \frac{y_l}{f} + r_{13} \right) z'_l + r_{01} = \frac{x_r}{f} z'_r \\ \left(r_{21} \frac{x_l}{f} + r_{22} \frac{y_l}{f} + r_{23} \right) z'_l + r_{02} = \frac{x_r}{f} z'_r \\ \left(r_{31} \frac{x_l}{f} + r_{32} \frac{y_l}{f} + r_{33} \right) z'_l + r_{03} = z'_r \end{cases}$$

同一点在两个坐标系中表示之间的关系

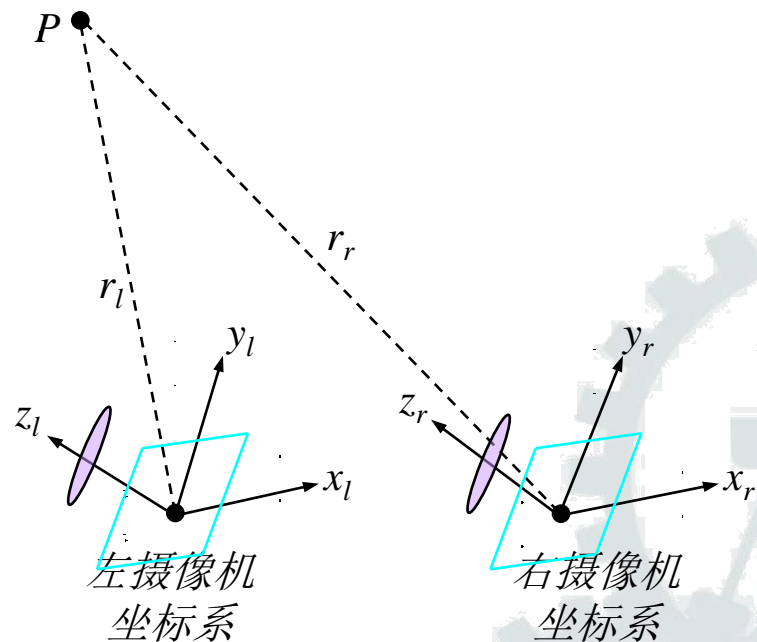




立体视觉 – 一般情况的进一步讨论

- ◆ 但为计算 \mathbf{R} 和 \mathbf{r}_0 , 测量 4 个以上点在两个坐标系中的 $x'_l, y'_l, z'_l, x'_r, y'_r, z'_r$ 并不容易, 尤其对 x'_l, y'_l, x'_r, y'_r 的测量比较困难。
- ◆ 根据计算深度值的三个方程可得:

$$\begin{aligned} \left(r_{11} \frac{x_l}{f} + r_{12} \frac{y_l}{f} + r_{13} \right) z'_l + r_{01} &= \frac{x_r}{f} z'_r \\ \left(r_{21} \frac{x_l}{f} + r_{22} \frac{y_l}{f} + r_{23} \right) z'_l + r_{02} &= \frac{y_r}{f} z'_r \\ \left(r_{31} \frac{x_l}{f} + r_{32} \frac{y_l}{f} + r_{33} \right) z'_l + r_{03} &= z'_r \end{aligned}$$
$$\Rightarrow \frac{1}{f} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_l z'_l \\ y_l z'_l \\ f z'_l \end{bmatrix} = \frac{1}{f} \begin{bmatrix} x_r z'_r \\ y_r z'_r \\ f z'_r \end{bmatrix} - \begin{bmatrix} r_{01} \\ r_{02} \\ r_{03} \end{bmatrix}$$





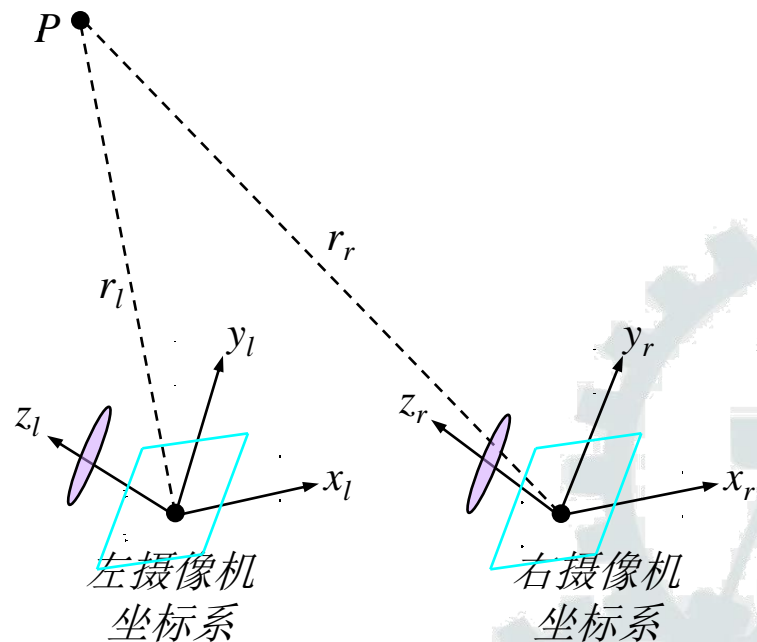
立体视觉 – 一般情况的进一步讨论

◇ 但为计算 \mathbf{R} 和 \mathbf{r}_0 ，测量 4 个以上点在两个坐标系中的 $x'_l, y'_l, z'_l, x'_r, y'_r, z'_r$ 并不容易，尤其对 x'_l, y'_l, x'_r, y'_r 的测量比较困难。

◇ 计算深度值的三个方程将两个摄像机成像面中的点的 \mathbf{x}, \mathbf{y} 坐标 x_l, y_l, x_r, y_r 与深度值 z'_l, z'_r 建立了关系：

$$x_l, y_l, x_r, y_r \Rightarrow z'_l, z'_r$$

仅根据这个关系也是有可能计算 \mathbf{R} 和 \mathbf{r}_0 中的全部系数。





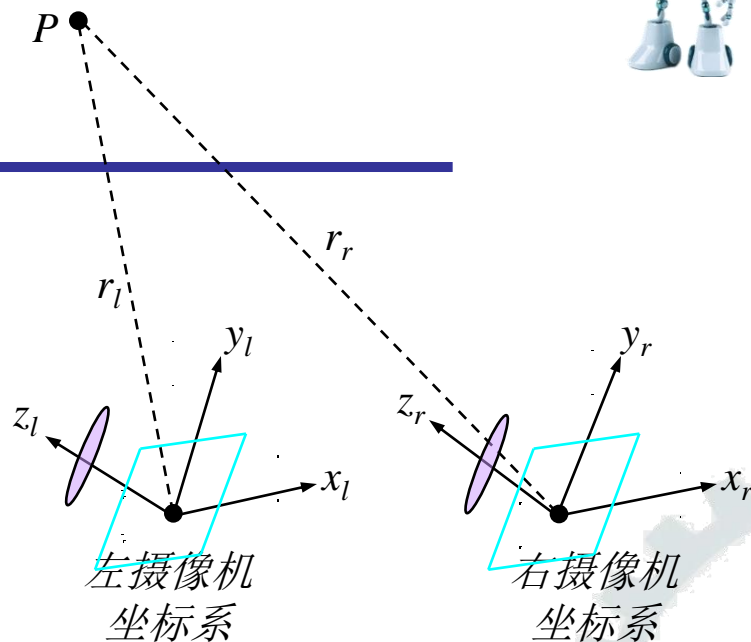
立体视觉 – 一般情况的进一步讨论

$$\frac{1}{f} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_l z'_l \\ y_l z'_l \\ f z'_l \end{bmatrix} = \frac{1}{f} \begin{bmatrix} x_r z'_r \\ y_r z'_r \\ f z'_r \end{bmatrix} - \begin{bmatrix} r_{01} \\ r_{02} \\ r_{03} \end{bmatrix}$$

- ◆ 根据这个方程组，仅测出 4 个点的深度值 z'_l, z'_r ，以及两个摄像机成像面中的点 x_l, y_l, x_r, y_r 就可以计算 \mathbf{R} 和 \mathbf{r}_0 中全部元素。

$$\mathbf{r} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{21} & r_{22} & r_{23} & r_{31} & r_{32} & r_{33} & r_{01} & r_{02} & r_{03} \end{bmatrix}^T$$

$$\begin{bmatrix} x_{li} z'_{li} & y_{li} z'_{li} & f z'_{li} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & f & 0 & 0 \\ 0 & 0 & 0 & x_{li} z'_{li} & y_{li} z'_{li} & f z'_{li} & 0 & 0 & 0 & 0 & 0 & f & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{li} z'_{li} & y_{li} z'_{li} & f z'_{li} & 0 & 0 & 0 & f \end{bmatrix} \mathbf{r} = \begin{bmatrix} x_{ri} z'_{ri} \\ y'_{ri} z'_{ri} \\ f z'_{ri} \end{bmatrix}$$





立体视觉

深度的计算

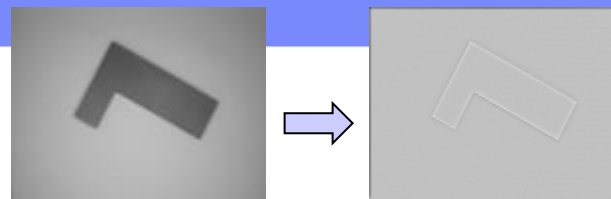
- ◇ 当前，对于立体视觉的关键问题，是如何解决对应性问题！

灰度级别匹配

- ◇ 在对应的表偏振线上，匹配灰度波形
- ◇ “亮度” = 图像辐射照度 $I(x,y)$
- ◇ 高斯的拉普拉斯零相交（ZLoG），是在左、右摄像机图像中辨别特征的广泛使用方法



高斯的拉普拉斯零相交 (ZLoG)



◆ 辨别特征的策略，稳定且匹配良好 (一定程度地突出特征)

◆ 具有强度 I 的图像的Laplacian:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

$$L = P \otimes I$$

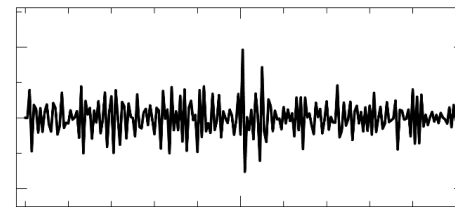
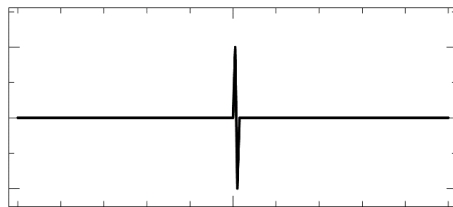
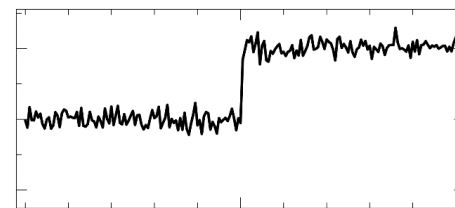
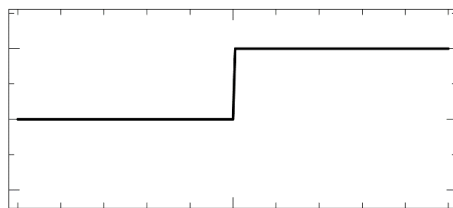
$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

◆ 与 P 卷积:

$$\begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}$$

◆ 在含噪声图像中
检测台阶/边缘

◆ 采用高斯平滑进行滤波





高斯的拉普拉斯零相交 (ZLoG)

```
import torch
import torch.nn.functional as F
import numpy as np

from torch.autograd import Variable
from math import exp
from PIL import Image
from torchvision.utils import save_image, make_grid

#将最后的矩阵中的元素归一化到0~1之间
def minmaxscaler(data):
    min = torch.min(data)
    max = torch.max(data)
    return (data - min)/(max-min)

#LoG变换
def LoG(img>window>window_size>mode="RGB"):
    img1_array = np.array(img>dtype=np.float32)#Image -> array
    img1_tensor = torch.from_numpy(img1_array)# array -> tensor
    # 处理不同通道数的数据
    if mode == 'L':
        img1_tensor = img1_tensor.unsqueeze(0).unsqueeze(0)#h,w -> n,c,h,w
    else:#RGB or RGBA
        img1_tensor = img1_tensor.permute(2,0,1)# h,w,c -> c,h,w
        img1_tensor = img1_tensor.unsqueeze(0)#c,h,w -> n,c,h,w
```



高斯的拉普拉斯零相交 (ZLoG)

```
channel = img1_tensor.size()[1]
window = Variable(window.expand(channel, 1, window_size, window_size).contiguous())
output = F.conv2d(img1_tensor, window, padding = window_size//2, groups = channel)
output = minmaxscaler(output)# 归一化到0~1之间
if (channel==4):
    save_image(output, "output.png", normalize=False)
else:
    save_image(output, "output.jpg", normalize=False)
return output

#近似卷积核
window = torch.Tensor([[[[0,1,1,2,2,2,1,1,0],
                          [1,2,4,5,5,5,4,2,1],
                          [1,4,5,3,0,3,5,4,1],
                          [2,5,3,-12,-24,-12,3,5,2],
                          [2,5,0,-24,-40,-24,0,5,2],
                          [2,5,3,-12,-24,-12,3,5,2],
                          [1,4,5,3,0,3,4,4,1],
                          [1,2,4,5,5,5,4,2,1],
                          [0,1,1,2,2,2,1,1,0]]]])

window_size = 9
img = Image.open("./input.png")
img = img.convert('L')
LoG(img,window,window_size,img.mode)
```



立体视觉例子

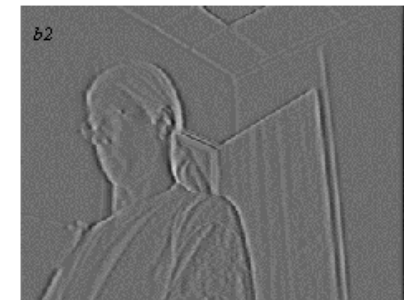
◇ 从一组立体图像中提取深度信息

◇ a1 和 a2: 左、右图像

◇ b1 和 b2: 过滤出竖直边缘的
左、右图像;
滤波器 = $[1 \ 2 \ 4 \ -2 \ -10 \ -2 \ 4 \ 2 \ 1]$

◇ c: 置信图像:
亮处 = 高置信度 (纹理好)

◇ d: 深度图像:
亮处 = 近; 暗处 = 远





安装在全地形机器人上的SVM 公司立体头

◇ 立体摄像机

◇ **Videre Design (公司)**

◇ www.videredesign.com

◇ 机器人

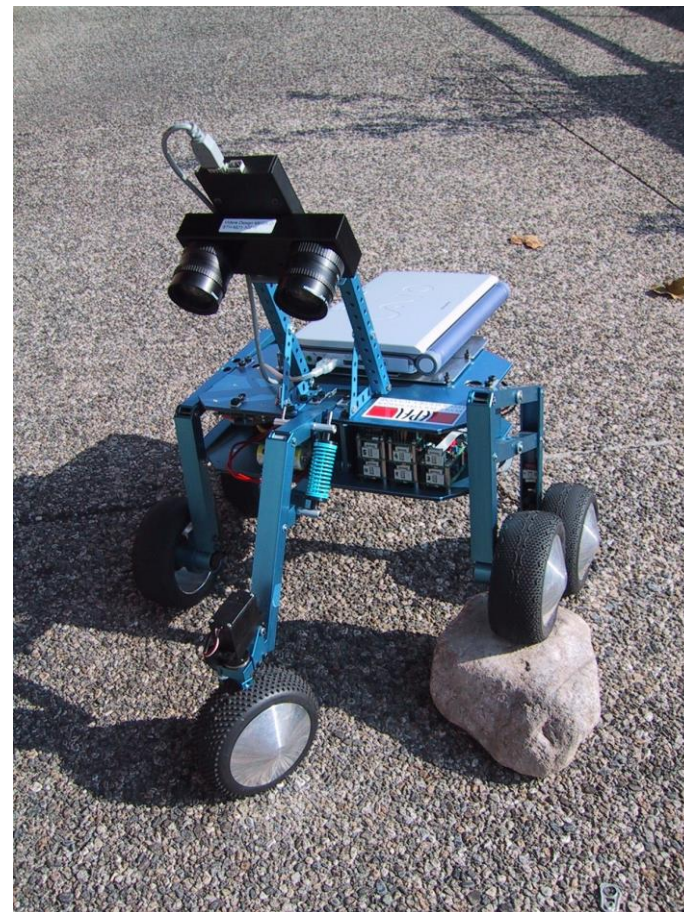
◇ 小虾, EPFL

◇ 立体视觉的应用

◇ 用于室外导航的，基于立体视觉的
可遍历性计算

**Traversability calculation based on
stereo images for outdoor navigation**

◇ 运动跟踪



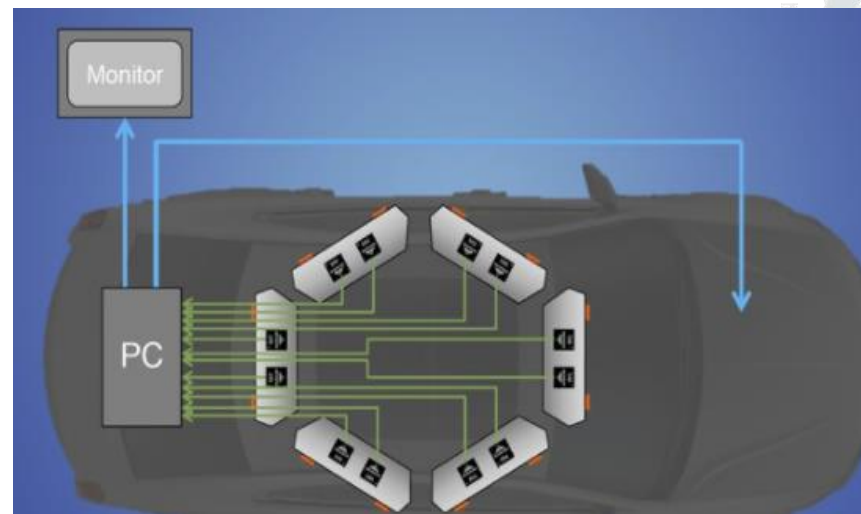


智能驾驶中的双目立体视觉



Subaru EyeSight

安霸 (Ambarella)





光流 (1)

- ◇ $E(x, y, t)$ = 时刻 t 在图像点 (x, y) 上的辐射照度
- ◇ $u(x, y)$ 和 $v(x, y)$ = 在 (x, y) 点的光流向量的分量
 - ◇ 在新图像 ($t + \delta t$ 时刻图像) 中搜索一点, 使得该点的辐射照度与 t 时刻时的相同 (获得恒定照度斑点的运动)。

$$E(x + \delta x, y + \delta y, t + \delta t) = E(x, y, t)$$

- ◇ 如图像亮度随 x, y 和 t 变化缓慢, 可以把上式左边按泰勒级数展开:

$$E(x + \delta x, y + \delta y, t + \delta t) =$$

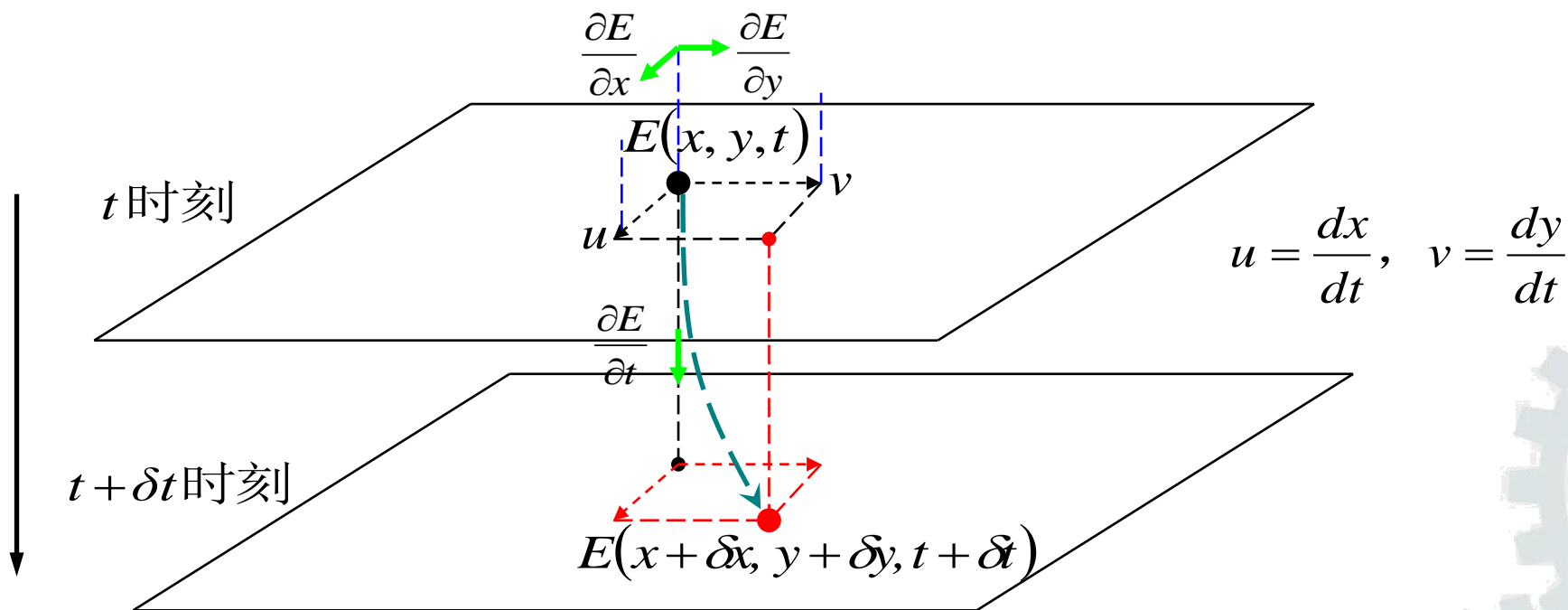
$$E(x, y, t) + \delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} + e = E(x, y, t)$$

- ◇ e = 包含 $u\delta$ 、 $v\delta$ 、 δ 二阶及以上更高阶项
- ◇ 令 $u(x, y)$ 和 $v(x, y)$ 分别表示 (x, y) 处的恒定照度斑点移动的两个速度分量:

$$u = \frac{dx}{dt}, \quad v = \frac{dy}{dt}$$



光流 (1)



$$\begin{aligned}
 E(x + \delta x, y + \delta y, t + \delta t) &= E(x, y, t) + \delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} + e \\
 &= E(x, y, t) + \left(u \frac{\partial E}{\partial x} + v \frac{\partial E}{\partial y} + \frac{\partial E}{\partial t} \right) \delta t + e
 \end{aligned}$$



光流 (2)

$$\begin{aligned} E(x, y, t) + \left(\frac{\partial E}{\partial x} \frac{dx}{dt} + \frac{\partial E}{\partial y} \frac{dy}{dt} + \frac{\partial E}{\partial t} \right) \delta t + e \\ = E(x, y, t) + \left(u \frac{\partial E}{\partial x} + v \frac{\partial E}{\partial y} + \frac{\partial E}{\partial t} \right) \delta t + e = E(x, y, t) \end{aligned}$$

◇ 随着 $\delta t \rightarrow 0$, $e \rightarrow 0$, 有:

$$u \frac{\partial E}{\partial x} + v \frac{\partial E}{\partial y} + \frac{\partial E}{\partial t} = 0$$

◇ 上式可缩写为“光流约束方程”:

$$E_x u + E_y v + E_t = 0$$

◇ 偏导数 E_x , E_y 和 E_t 从图像中估计得到。

◇ 从这个方程, 只能得到向量 (u, v) 的方向, u 和 v 数值是不惟一的。

◇ 于是要引入附加的约束条件: 光流平滑性



光流 (3)

◆ 附加约束条件

◆ 相邻像素运动是相似的，因此全部像素的总光流是平滑的（变动不大）

◆ 总光流计算（光流矢量长度平方的积分）：

$$e_s = \iint (u^2 + v^2) dx dy$$

◆ 光流约束方程的误差：

$$e_c = \iint (E_x u + E_y v + E_t)^2 dx dy$$

◆ 解优化问题：

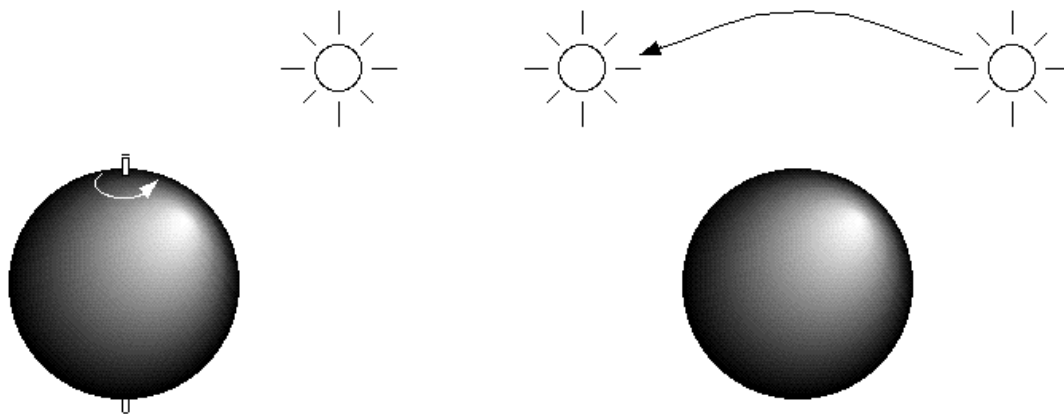
$$\min_{u,v} J = e_s + \lambda e_c = \iint [u^2 + v^2 + \lambda (E_x u + E_y v + E_t)^2] dx dy$$

$$\lambda > 0$$



光流的问题

- ◆ 这里，球体或光源的运动显示了：光流并不总是与运动场一致。（自转时光流到处为零；球不动但光源移动时光流到处非零）

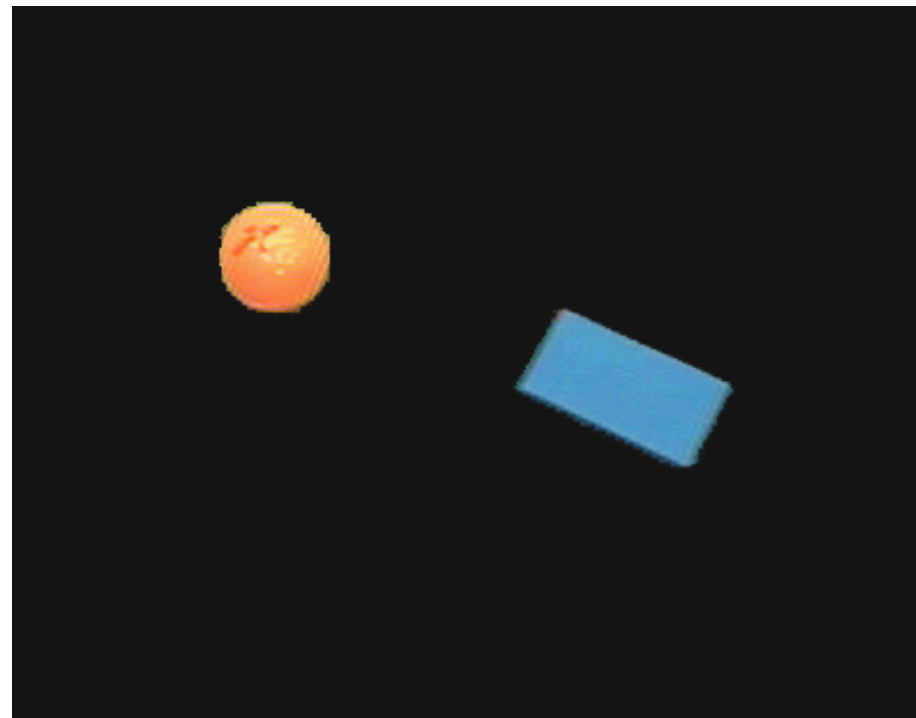
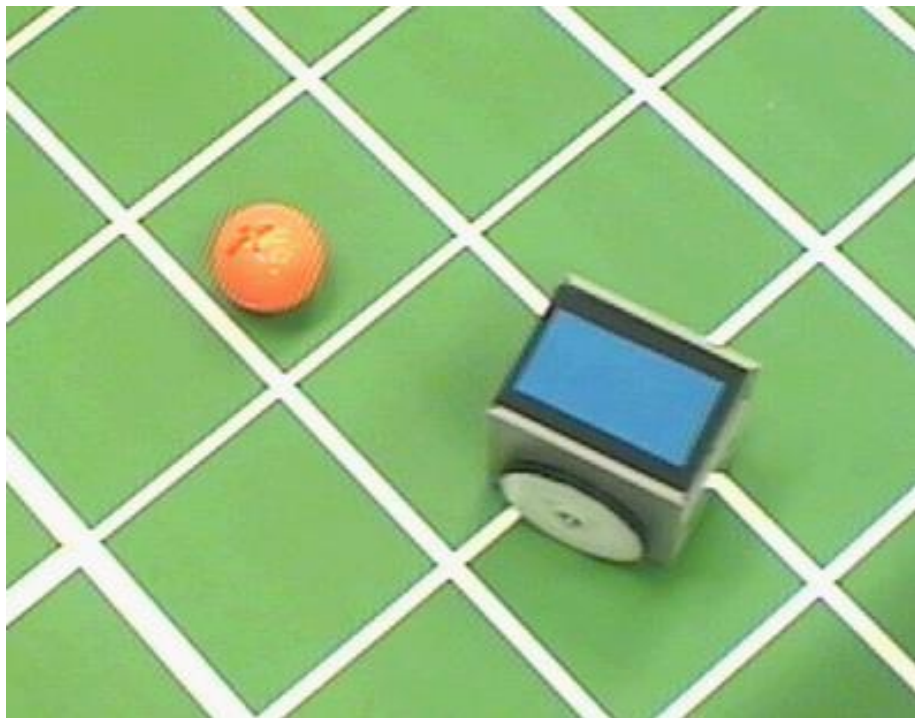


- ◆ 光流的不连续性
 - ◆ 侧面轮廓 (一个物体遮挡另一个)
 - ◆ 光流的不连续性
 - ◆ 找到这些点
 - ◆ 停止平滑性假设
- ◆ 球体运动，移动光源



颜色跟踪传感器

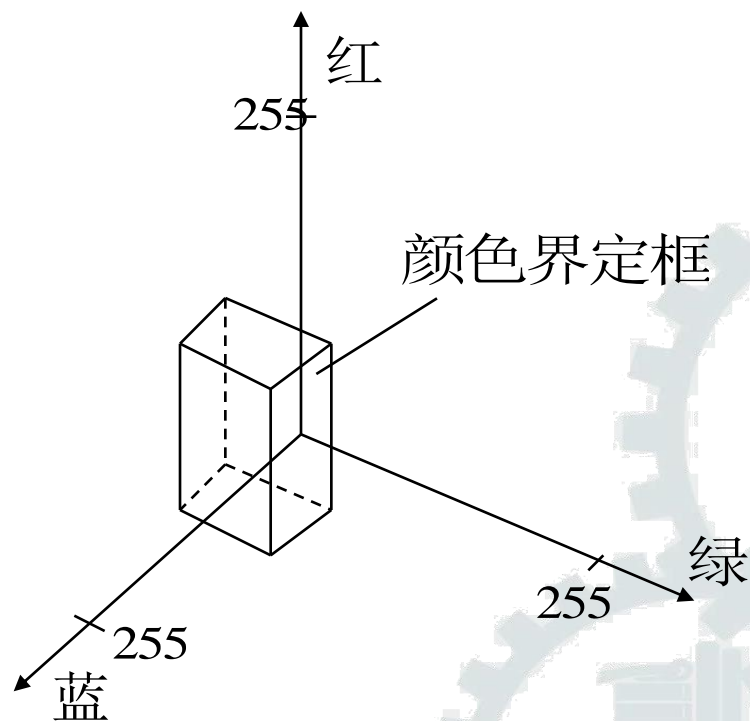
◇ 足球赛中，使用颜色跟踪方法估计球和机器人的运动





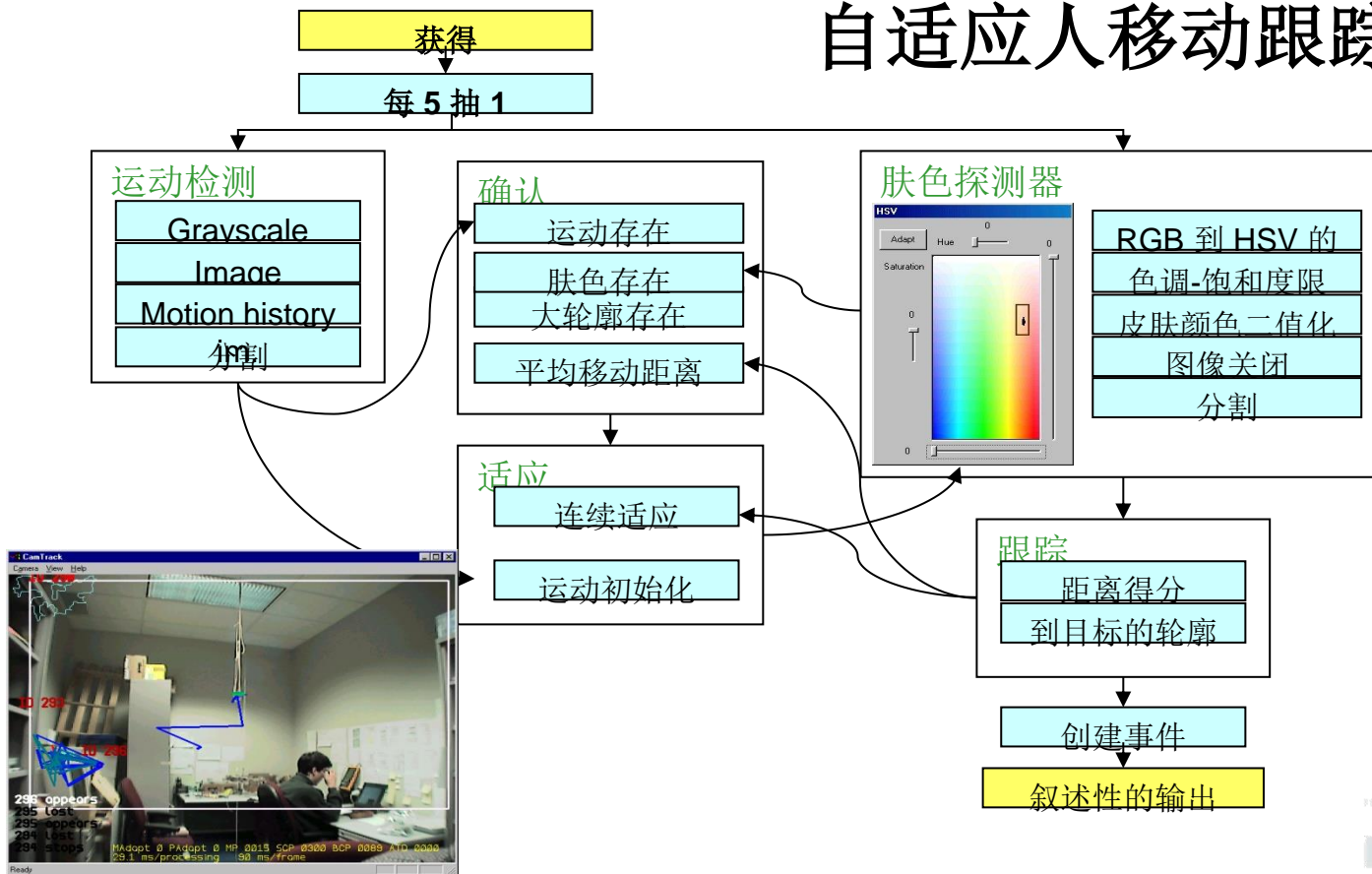
颜色跟踪传感器（续）

- ◆ 在红绿蓝三维区间（ $256 \times 256 \times 256$ ）中划出一块，用顶点6个参数描述
 - ◆ 找出所有落入该颜色界定框中的像素
 - ◆ 聚类形成一个表示目标的区域
 - ◆ 找出该区域的质心位置——目标物的质心位置
-
- ◆ 亮度改变后，颜色界定框也应自适应移动
 - ◆ 识别的颜色相对比较简单（如应用在小机器人色块和球的识别、跟踪）



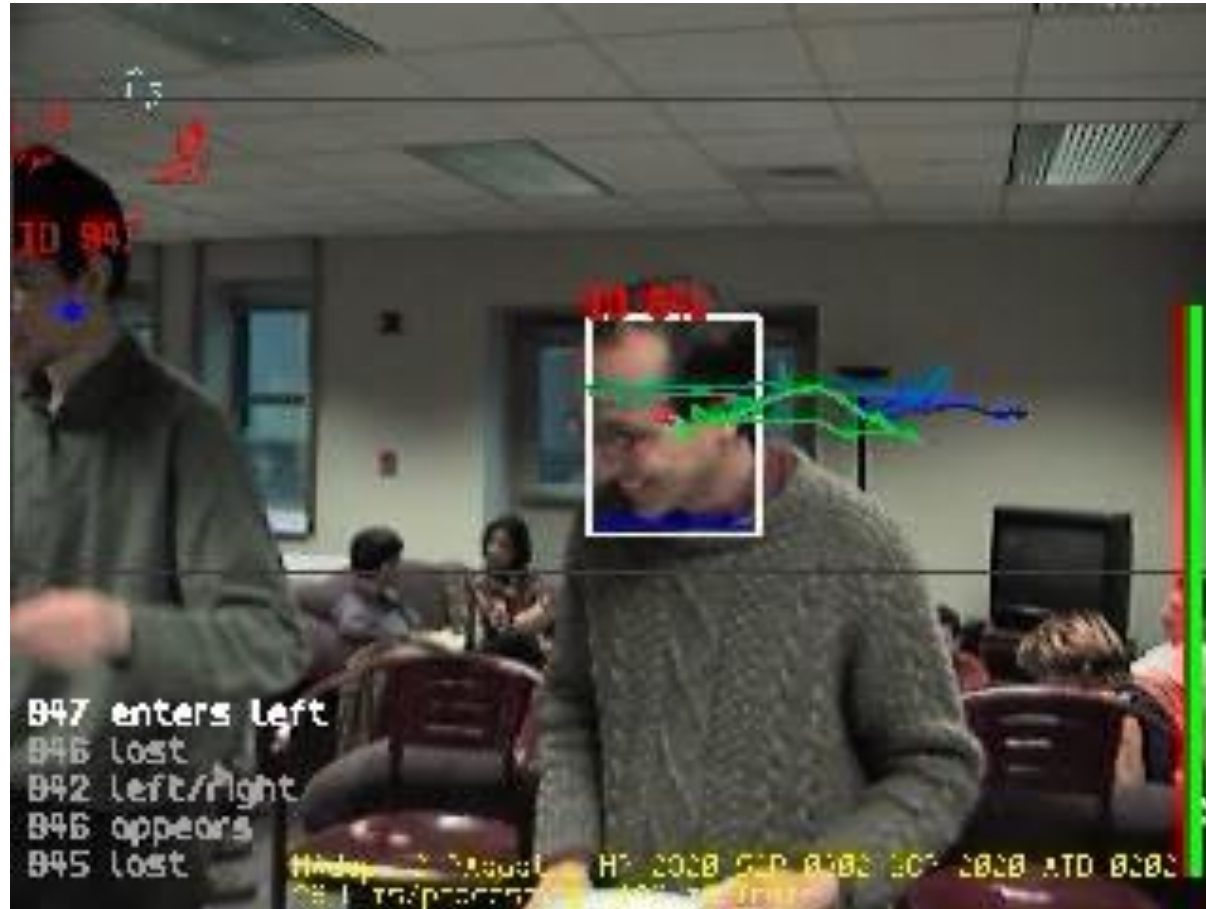


自适应人移动跟踪





自适应人移动跟踪





自适应人移动跟踪（续一）





自适应人移动跟踪（续二）

基于贝叶斯最大后验概率的
跟踪算法
(Face Tracking)



结束

