

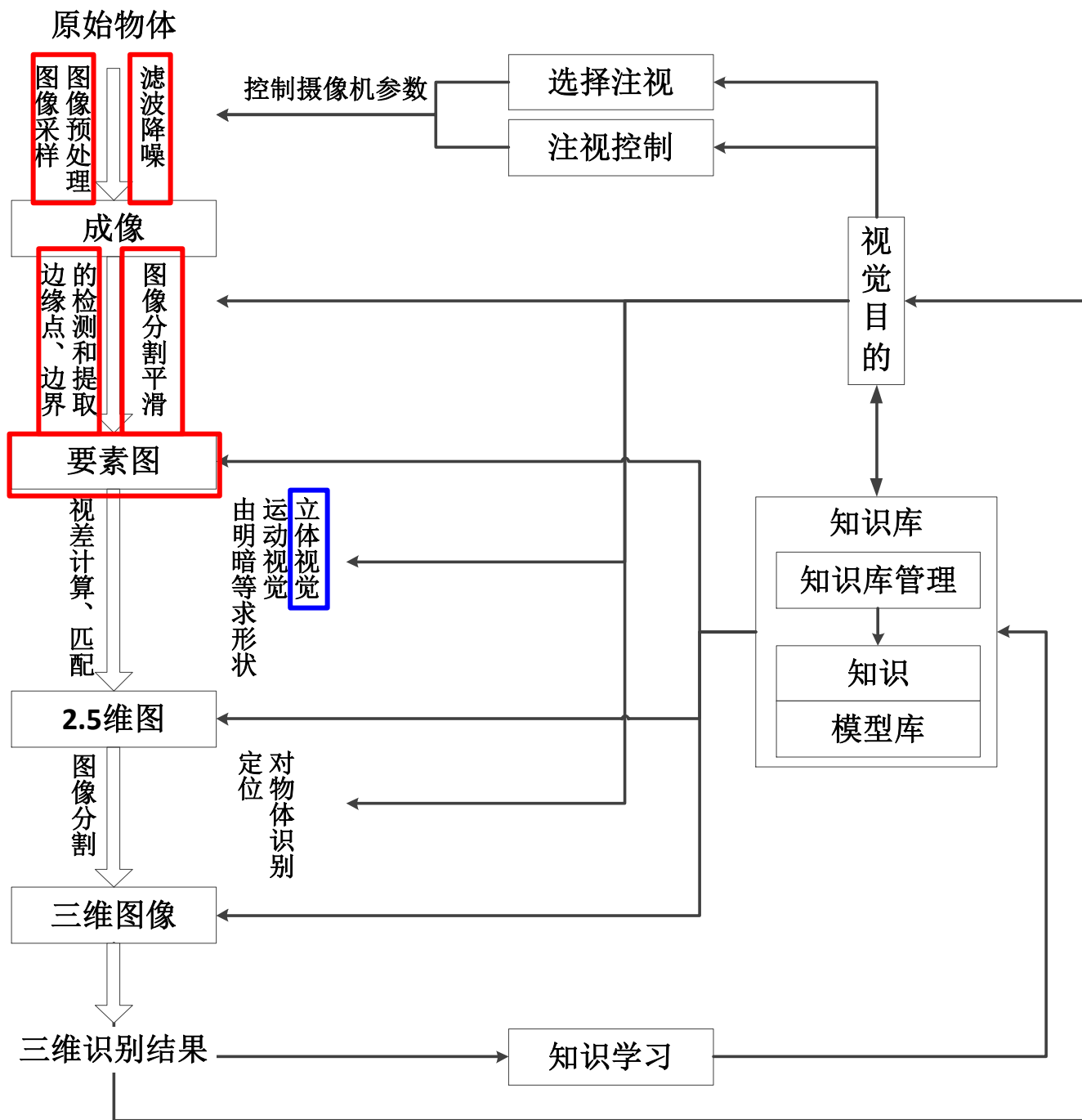
计算机视觉——双目立体视觉

2025春

桑农 王岳环



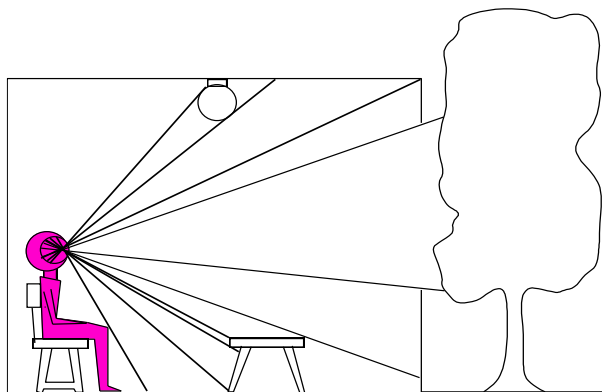
计算机视觉系统框架





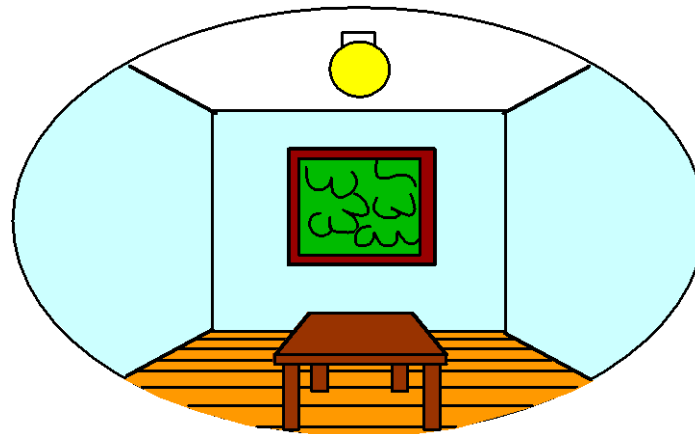
深度感知概念

- 图像是3D世界到2D平面的映射



Point of observation

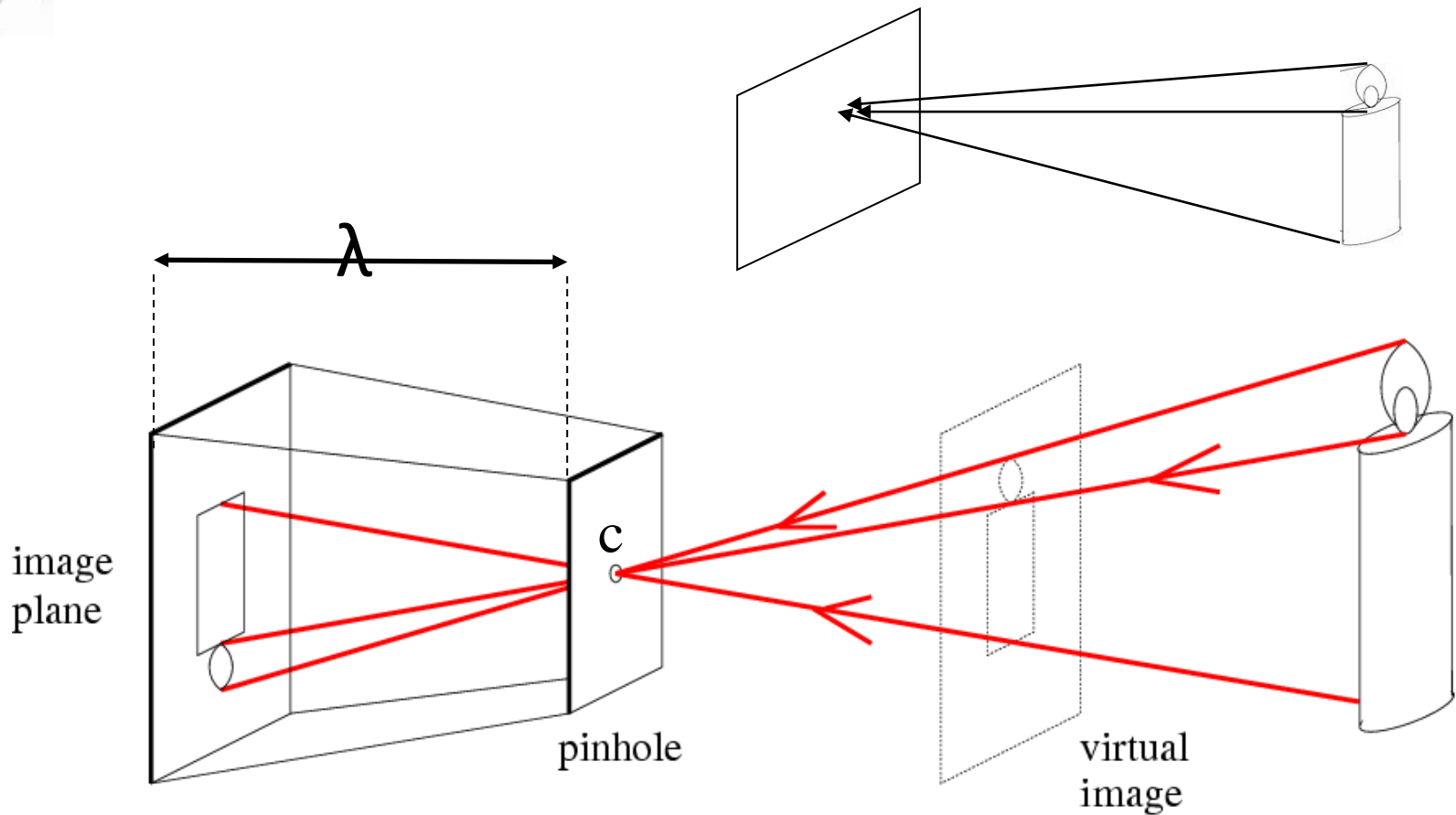
3D 世界



2D 图像



深度感知概念



针孔相机

λ = 焦距, c = 相机中心



深度感知概念

- 深度感知问题



逆问题



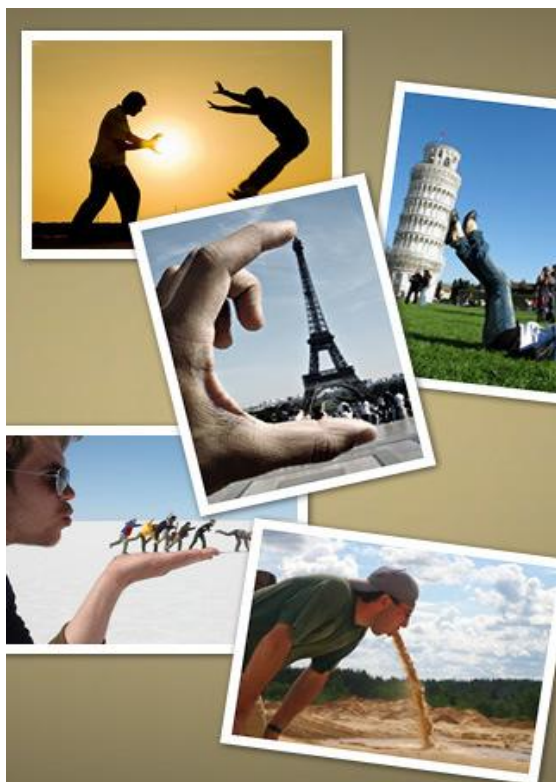
深度感知概念

- 深度感知应用
- 3D电影
 - 偏光眼镜
 - 色差式眼镜（红蓝红青）
 - 液晶快门眼镜
 - 裸眼3D



深度感知概念

- 深度感知应用
- 摄影





深度感知概念

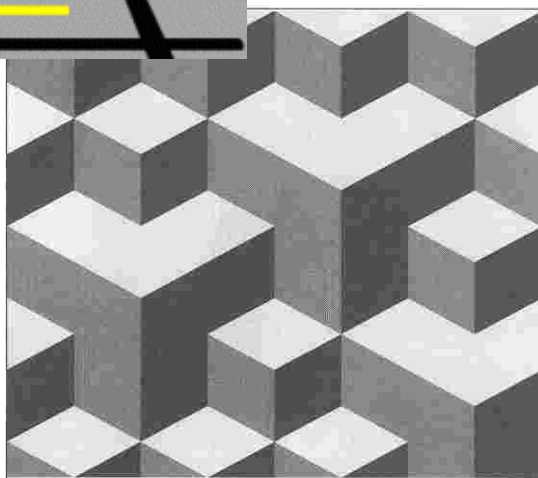
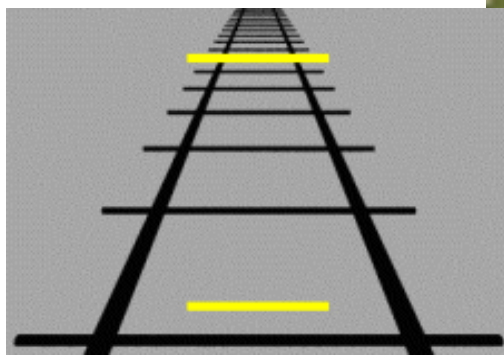
- 绝对深度感知：
 - 在适当的成像条件下，如双或多相机成像且相机参数已知，或拥有物体绝对大小的先验知识同时已知相机参数，估计物体与相机间的绝对距离
- 相对深度感知：
 - 定量感知：利用特定的成像方式（光度、明暗、运动等），定量估计物体间的相对深度
 - 定性感知：利用场景中物体间的相互关系、物体相对大小的先验知识、成像特性等，定性评价场景深度信息
 - 相对深度通常可通过单眼视觉获得



深度感知概念

- 单眼深度感知：通常只能获得定性的相对深度感知，但在某些情况下，如拥有物体绝对大小的先验知识时，也可以获得物体与相机间绝对距离的感知
- 双（多）眼深度感知：能够获得定量的绝对深度感知

单眼深度感知





单眼深度感知

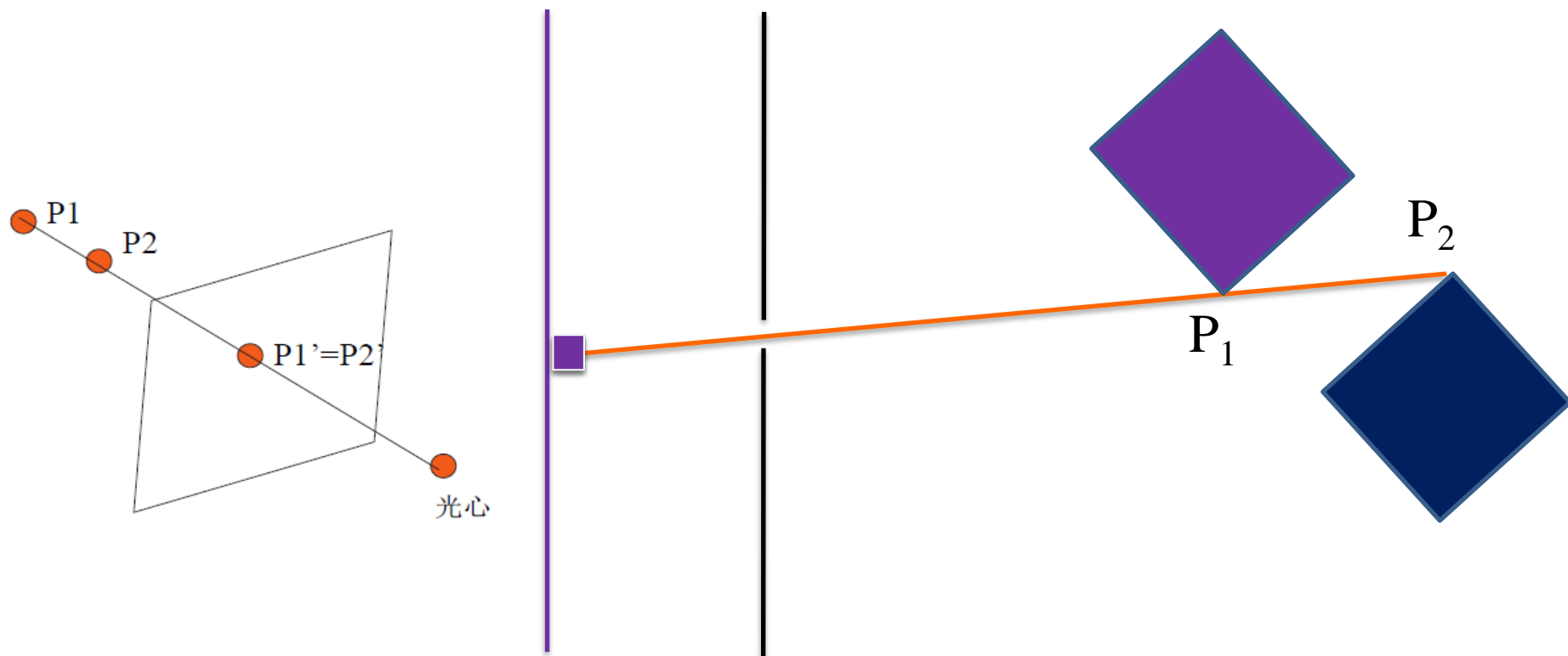
- 单眼视觉在深度感知中的问题





单眼深度感知

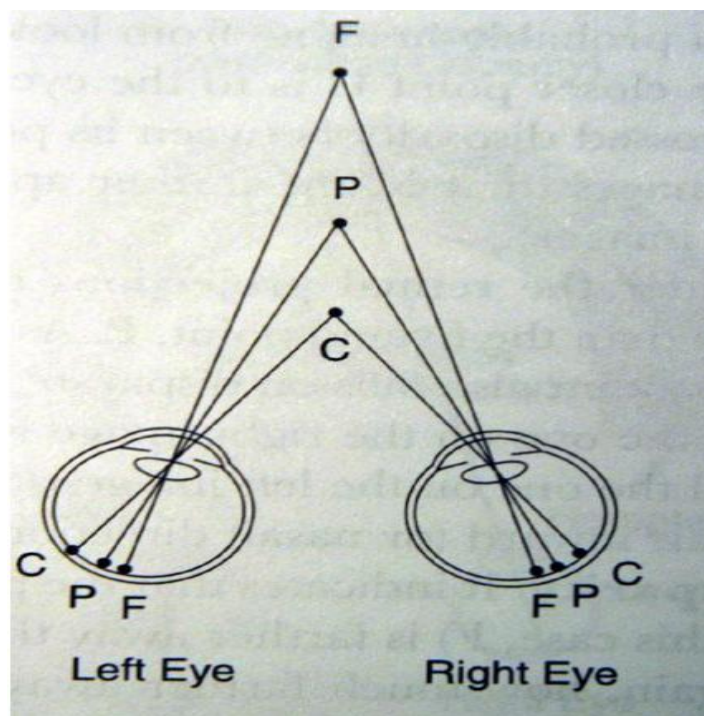
- 为何单眼视觉难以定量感知深度





第9章 双目立体视觉

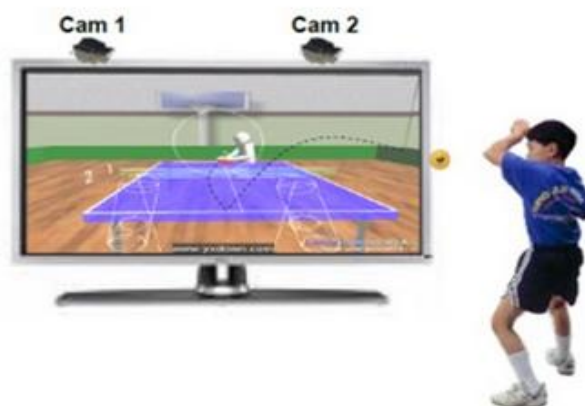
- 除了单眼可以感知深度外，应用更广泛的是通过双眼的观察得到物体定量的深度信息





第9章 双目立体视觉

- 应用



人机交互



3D视频



三维跟踪



机器人



第9章 双目立体视觉

- 9.1 双目立体视觉原理
- 9.2 双目成象和视差
- 9.3 基于区域的双目立体匹配
- 9.4 基于特征的双目立体匹配
- 9.5 视差图误差检测与校正



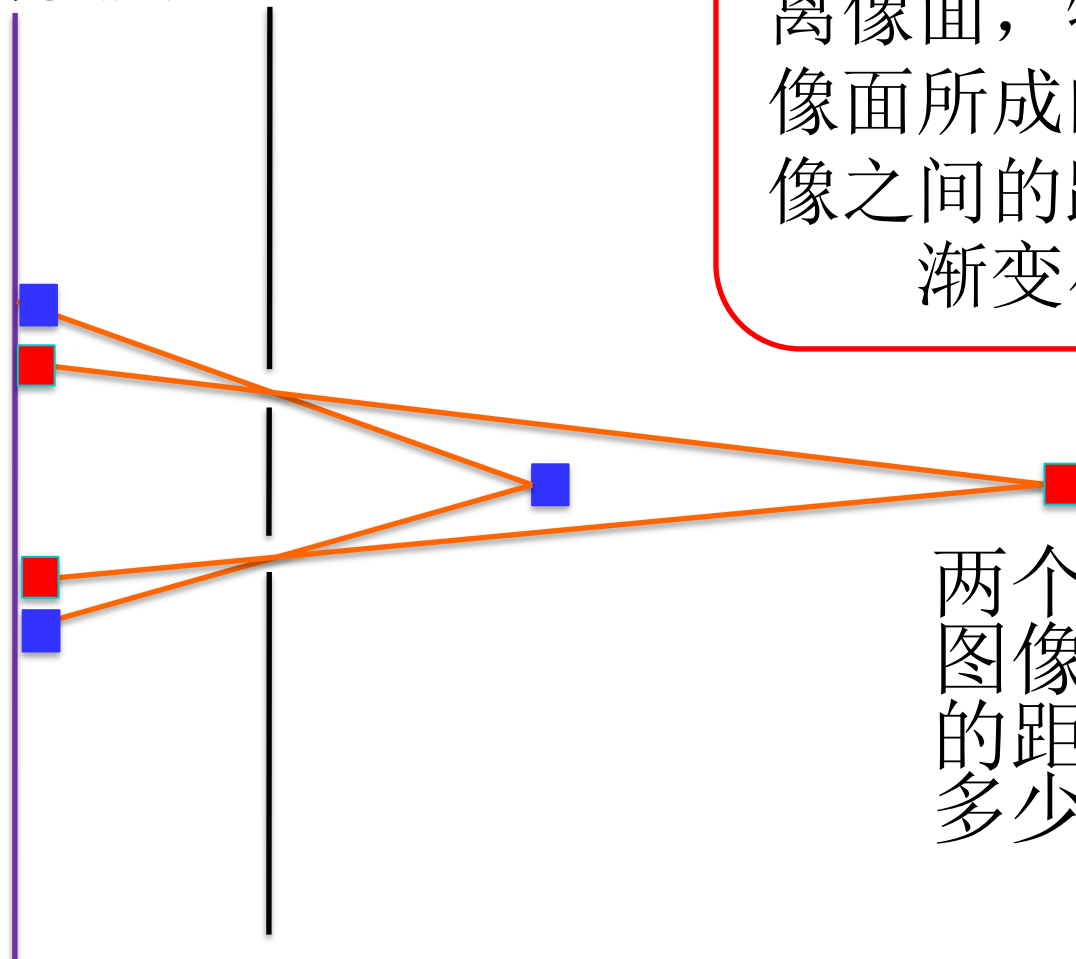
9.1 双目立体视觉原理

- 立体视觉主要研究如何借助（多图象）成像技术从（多幅）图象里获取场景中物体的距离（深度）信息



9.1 双目立体视觉原理

- 小孔成像模型



随着物体逐渐远离像面，物体在像面所成的两个像之间的距离逐渐变小

两个投影
图的像最近
的距离是
多少？



9.1 双目立体视觉原理

六个模块 \longleftrightarrow 六项工作

1. 摄像机标定 (2.4节)
2. 图象获取 (2.1节, 2.3节)
3. 特征提取 (第4章, 第6章)
 - 立体视觉借助不同观察点对同一景物间的视差来帮助求取3-D信息 (特别是深度信息)。所以需要判定同一景物在不同图象中的对应关系选择合适的图象特征以进行多图象间的匹配



9.1 双目立体视觉原理

六个模块 \longleftrightarrow 六项工作

4. 立体匹配

- 根据对所选特征的计算来建立特征间的对应关系，从而建立同一个空间点在不同图象中的象点之间的关系，并由此得到相应的视差图象

5. 3-D信息恢复

- 根据得到的视差图象，可以进一步计算深度图象，并恢复场景中的3-D信息



9.1 双目立体视觉原理

六个模块 \longleftrightarrow 六项工作

6. 后处理

- 3-D信息常不完整或存在一定的误差
- (1) 深度插值
 - 只能恢复出图象中特征点处的视差值
- (2) 误差校正
- (3) 精度改善
 - 像素级视差到亚像素级的视差



9.2 双目成象和视差

- 获得同一场景的两幅视点不同的图像

9.2.1 双目横向模式

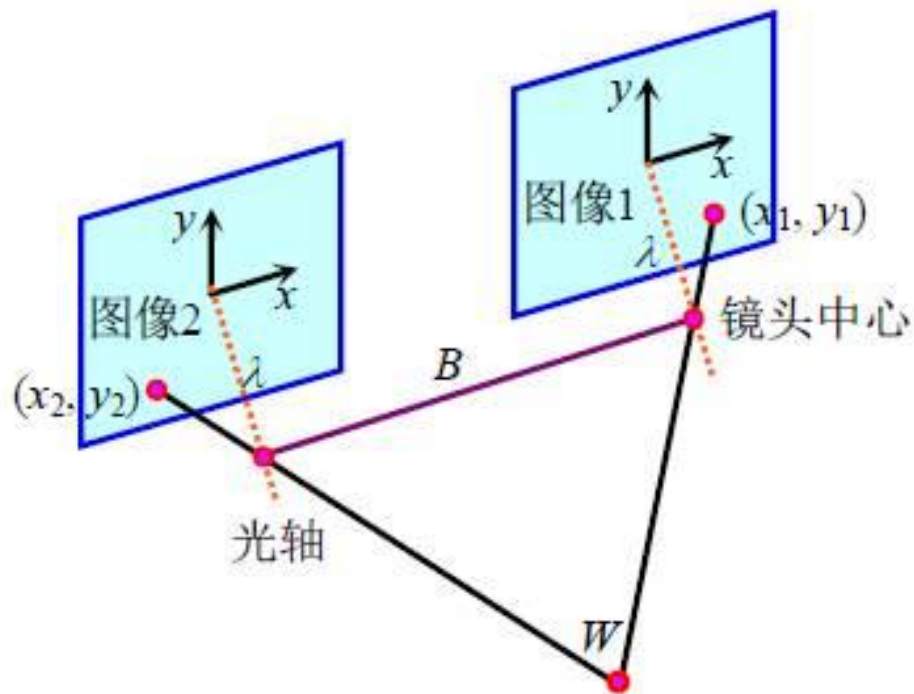
9.2.2 双目横向会聚模式

9.2.3 双目纵向模式



9.2.1 双目横向模式

- 摄像机坐标系统和世界坐标系重合
- 像平面与世界坐标系统的XY平面也是平行
- 两个镜头中心间的连线称为系统的基线B



双目横向成像示意图



9.2.1 双目横向模式

坐标系：

世界坐标系

左相机图像坐标系

右相机图像坐标系

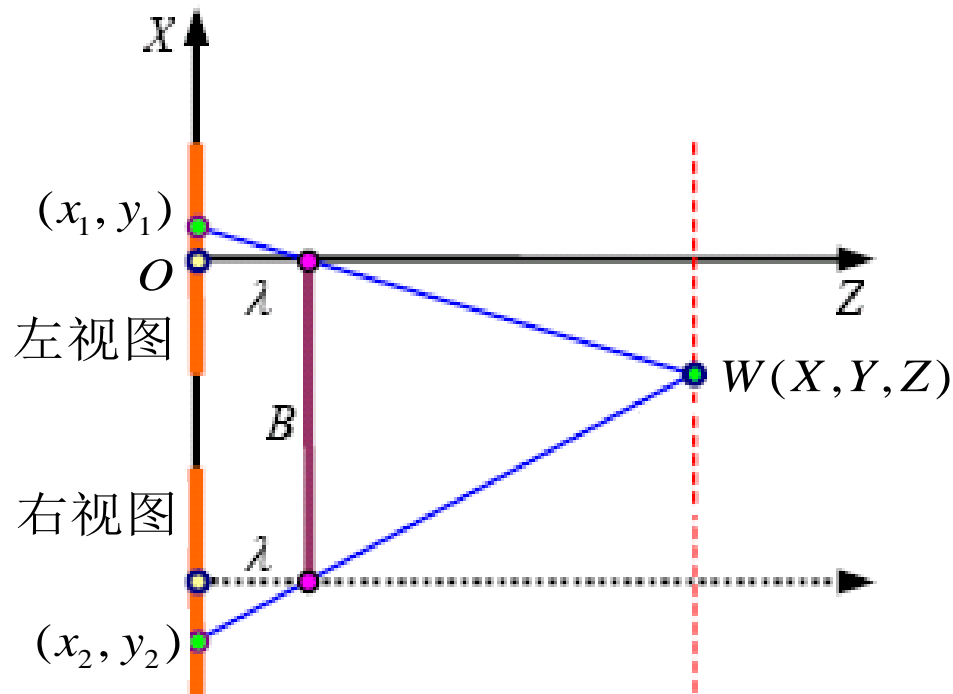
已知：

双目相机的焦距 λ

基线长度 B

W 点在左视图位置 (x_1, y_1)

W 点在右视图位置 (x_2, y_2)



求解：

W 点的深度信息 Z



9.2.1 双目横向模式

- 1. 视差和深度

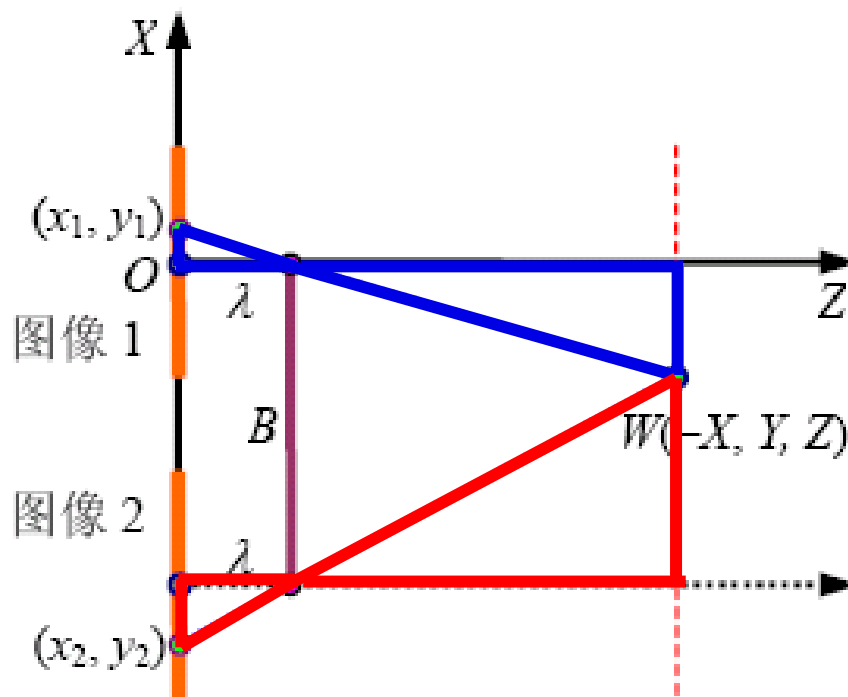
$$-X = \frac{x_1}{\lambda}(Z - \lambda)$$

$$B + X = \frac{-x_2}{\lambda}(Z - \lambda)$$

$$\frac{\lambda B}{Z - \lambda} = x_1 - x_2$$

$$Z = \lambda \left(1 + \frac{B}{d} \right)$$

深度



平行双目成像中的视差

$$d = x_1 - x_2$$

视差



9.2.1 双目横向模式

- 利用视差估计深度



左视图



深度图



右视图

$$Z = \lambda \left(1 + \frac{B}{d} \right)$$

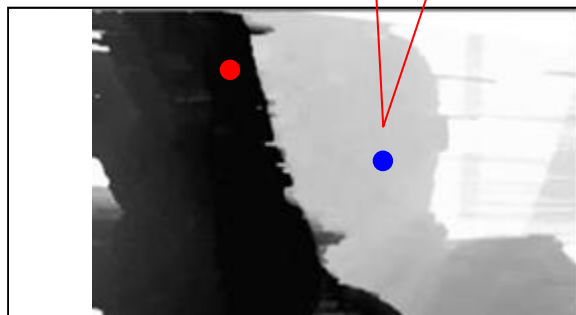


9.2.1 双目横向模式

- 利用视差估计深度



左视图



深度图



右视图

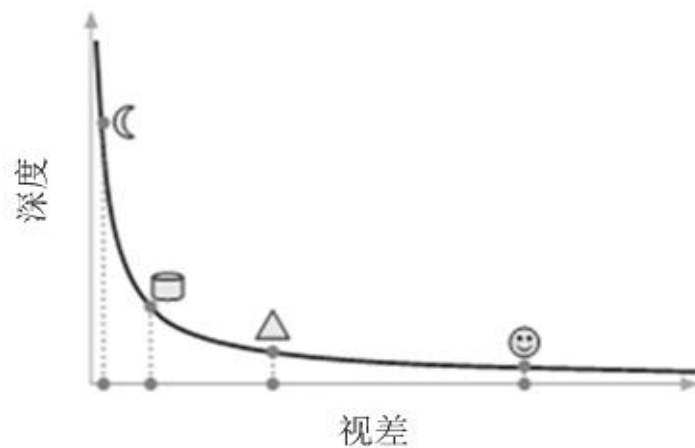
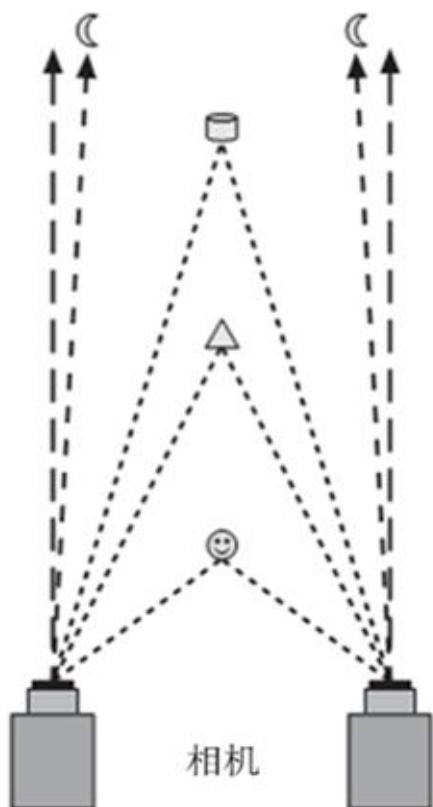
$$Z = \lambda \left(1 + \frac{B}{d} \right)$$

视差越大说明
目标离相机越近



9.2.1 双目横向模式

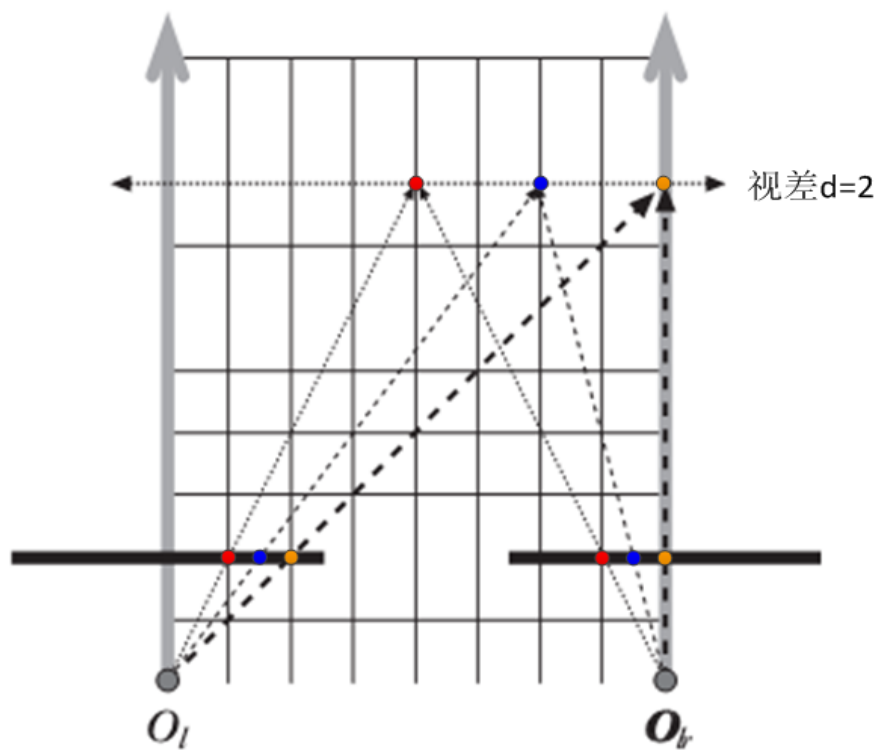
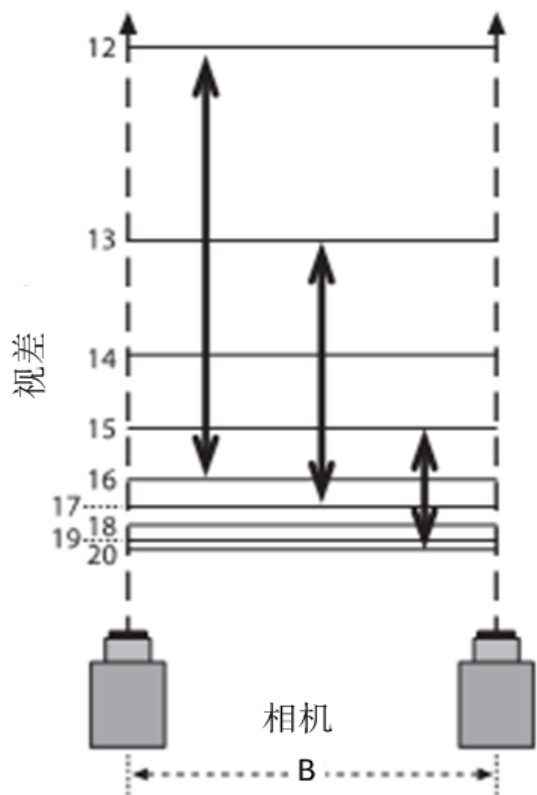
- 视差和深度成反比关系：
$$Z = \lambda \left(1 + \frac{B}{d} \right)$$





9.2.1 双目横向模式

- 同一深度下的视差一样





9.2.1 双目横向模式

- 1. 视差和深度
- 条件：3-D空间点在两个摄像机的公共视场
- 由于摄像机视角不同，以及被摄物体形状和拍摄环境的影响，可能一些3D点在一个摄像机是可见的，在另一个摄像机是不可见的
- 对应点不确定问题



9.2.1 双目横向模式

- 1. 视差和深度
- 测距精度

$$x_{1e} = x_1 + e, \quad d_{1e} = x_1 + e - x_2 = d + e$$

视差d增加，则深度Z减小

$$\Delta Z = Z_{1e} - Z = \lambda \left(1 + \frac{B}{d_{1e}} \right) - \lambda \left(1 + \frac{B}{d} \right) = \frac{-\lambda B e}{d(d + e)}$$

$$\frac{\lambda B}{Z - \lambda} = x_1 - x_2 = d$$

$$\Delta Z = \frac{-e(Z - \lambda)^2}{\lambda B + e(Z - \lambda)} \approx \frac{-eZ^2}{\lambda B + eZ}$$



9.2.1 双目横向模式

- 1. 视差和深度

- 测距精度

$$\Delta Z = \frac{-e(Z - \lambda)^2}{\lambda B + e(Z - \lambda)} \approx \frac{-eZ^2}{\lambda B + eZ}$$

- 测距精度与焦距、基线长度、物距有关。焦距、基线越长精度越高，物距越大，精度越低
- 直观理解：
 - 两个相机距离“越远越好”
 - 更适用于近距离测距



9.2.2 双目横向会聚模式

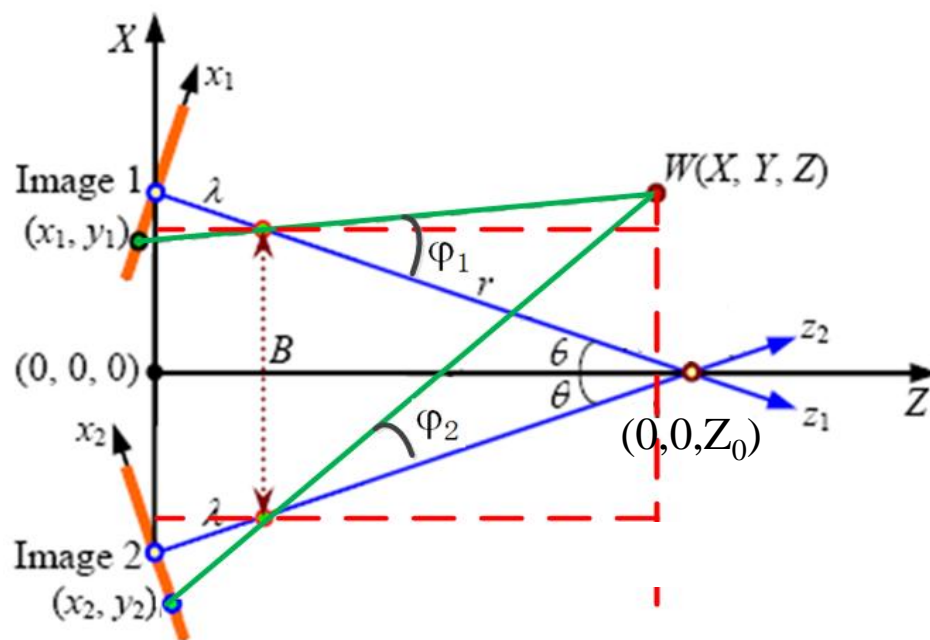
- 两个单目系统绕各自中心相向旋转

$$\frac{X - \frac{B}{2}}{(Z - \lambda \cos \theta)} = \tan(\phi_1 - \theta)$$

$$\frac{X + \frac{B}{2}}{(Z - \lambda \cos \theta)} = \tan(\phi_2 + \theta)$$

$$\phi_1 = \arctan\left(\frac{x_1}{\lambda}\right)$$

$$\phi_2 = \arctan\left(\frac{-x_2}{\lambda}\right)$$



$$Z = \frac{B}{\tan(\phi_2 + \theta) - \tan(\phi_1 - \theta)} + \lambda \cos \theta$$

会聚双目成像中的视差



双目立体匹配

- 立体匹配主要难点

(1) 光学失真和噪声（亮度、色调、饱和度等失衡）



(2) 平滑表面的镜面反射

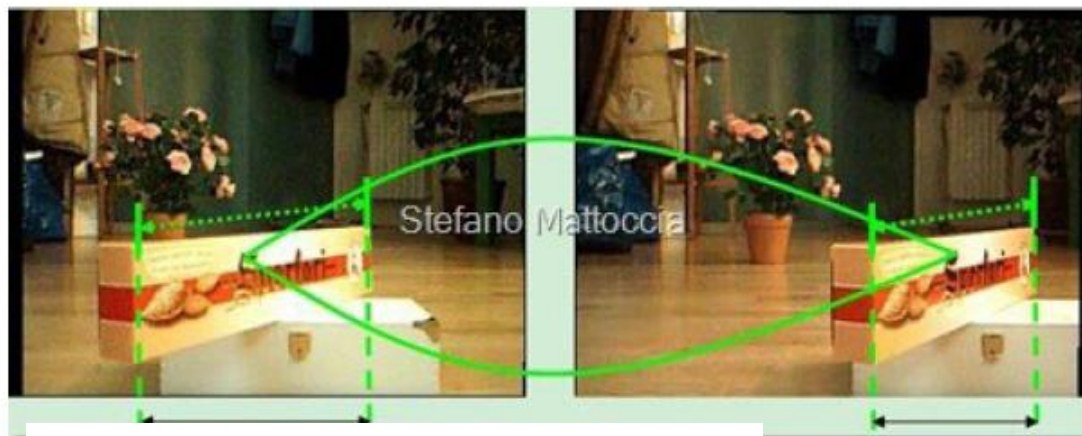




双目立体匹配

- 立体匹配主要难点

(3) 投影缩减 (Foreshortening)



(4) 透视失真 (Perspective distortions)

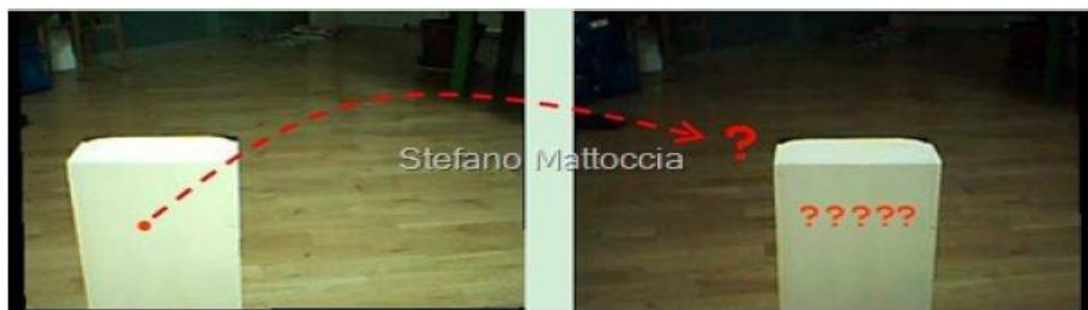




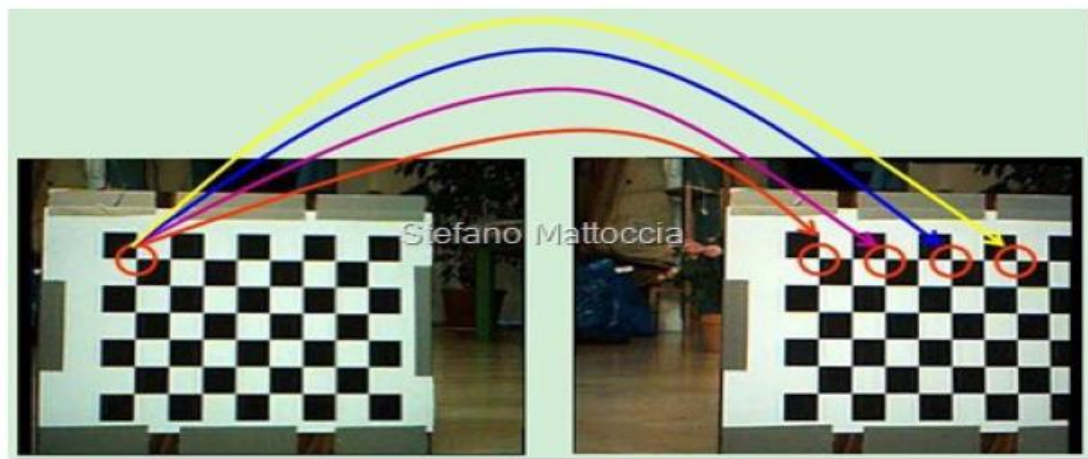
双目立体匹配

- 立体匹配主要难点

(5) 低纹理 (Low texture)



(6) 重复纹理 (Repetitive/ambiguous patterns)





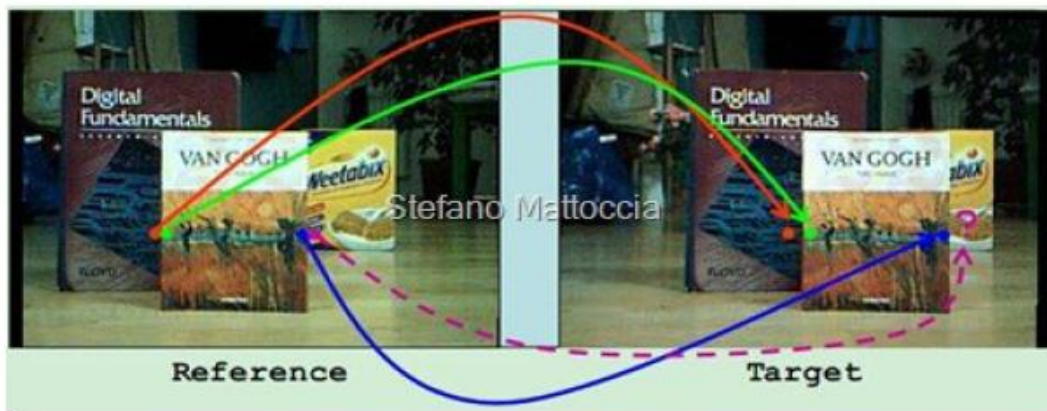
双目立体匹配

- 立体匹配主要难点

(7) 透明物体



(8) 重叠和非连续





双目立体匹配

■ 立体匹配约束条件

- 匹配的约束条件，就是根据所选匹配基元将现实物理世界的某些固有属性表示成匹配所必须遵循的若干规则，用以提高系统的去歧义匹配能力和计算效率。优良的匹配算法应能充分利用各种辅助信息或者约束条件来提高匹配的质量。

■ 1: 唯一性约束

- 也就是说一幅图像上的一个像素在另一幅图像上最多只能对应第二幅图像中的一个像素。例外情况是存在遮挡等条件下，将不存在对应点。这条约束条件是立体匹配必须满足的。



双目立体匹配

■ 立体匹配约束条件

■ 2. 连续性约束

- 物体表面一般都是光滑的，因此物体表面上各点在图像上的投影是连续的，其视差也是连续的。但是，由于遮挡问题，在物体边界处，比如边界两侧的两个点，连续性约束并不成立。

■ 3. 相似性约束

- 物体表面上一点在两幅或多幅图像上的投影在某些物体度量上(如灰度，灰度梯度变化等几何形状上)具有相似性。比如空间某一个点在一条直线上，它在图像中的投影也应该在一条直线上。此约束限制了寻找对应点时的搜索范围。

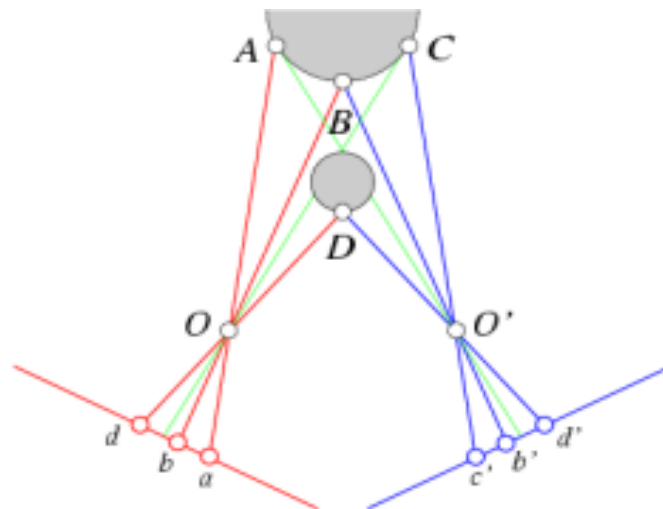
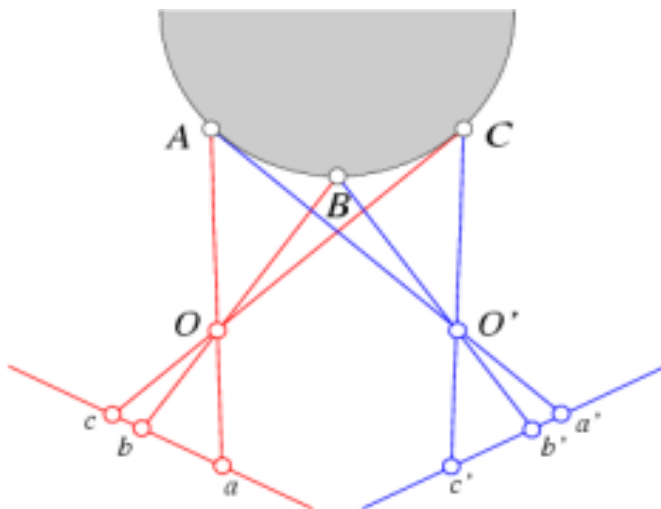


双目立体匹配

■ 立体匹配约束条件

■ 4. 顺序一致性约束

- 图像是对现实世界的投影，假如在现实世界中，P点在Q点的左边，则在图像上，P点仍然在Q点的左边。但是，如果视点的方位变化很大，这个约束条件可能不被满足。





双目立体匹配

■ 立体匹配约束条件

■ 5. 互对应约束

- 假设搜索从左图像点A开始，找到右图像上对应的点B。如果任务反过来，搜索从B点开始，但是没能找到A，则匹配不可靠，应该被排除。这个约束有助于排除由于遮挡，高光或噪声原因而不存在对应的那些点。

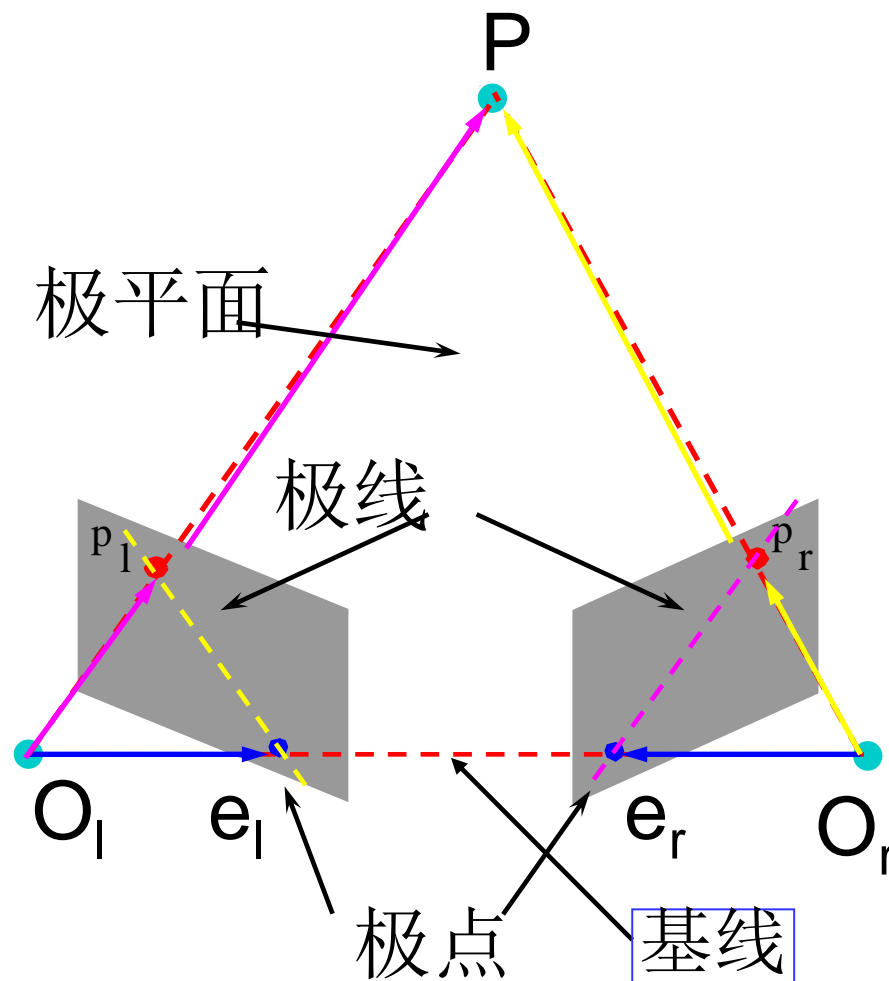
■ 6. 视差范围约束

■ 7. 极线约束



双目立体匹配

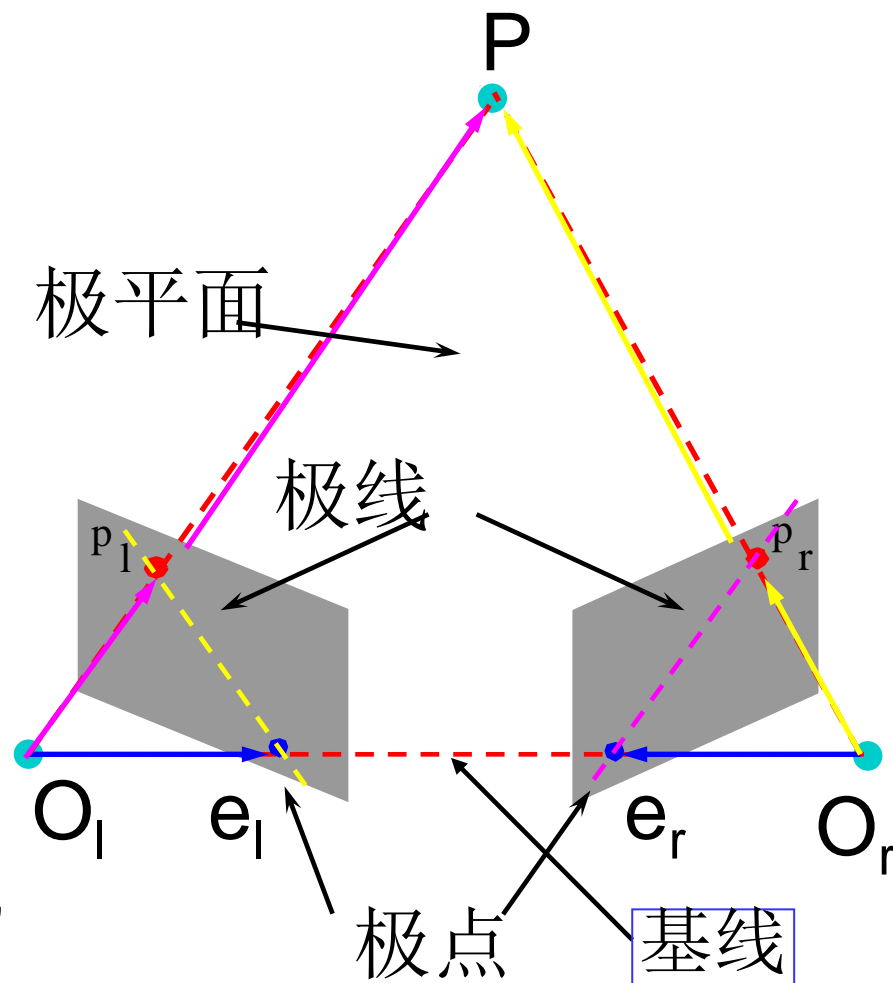
- 极线约束
- 基线：左右两相机光心的连线；
- 极平面：空间点、两相机光心决定的平面；
- 极点：基线与两摄像机图像平面的交点；
- 极线：极平面与图像平面的交线。





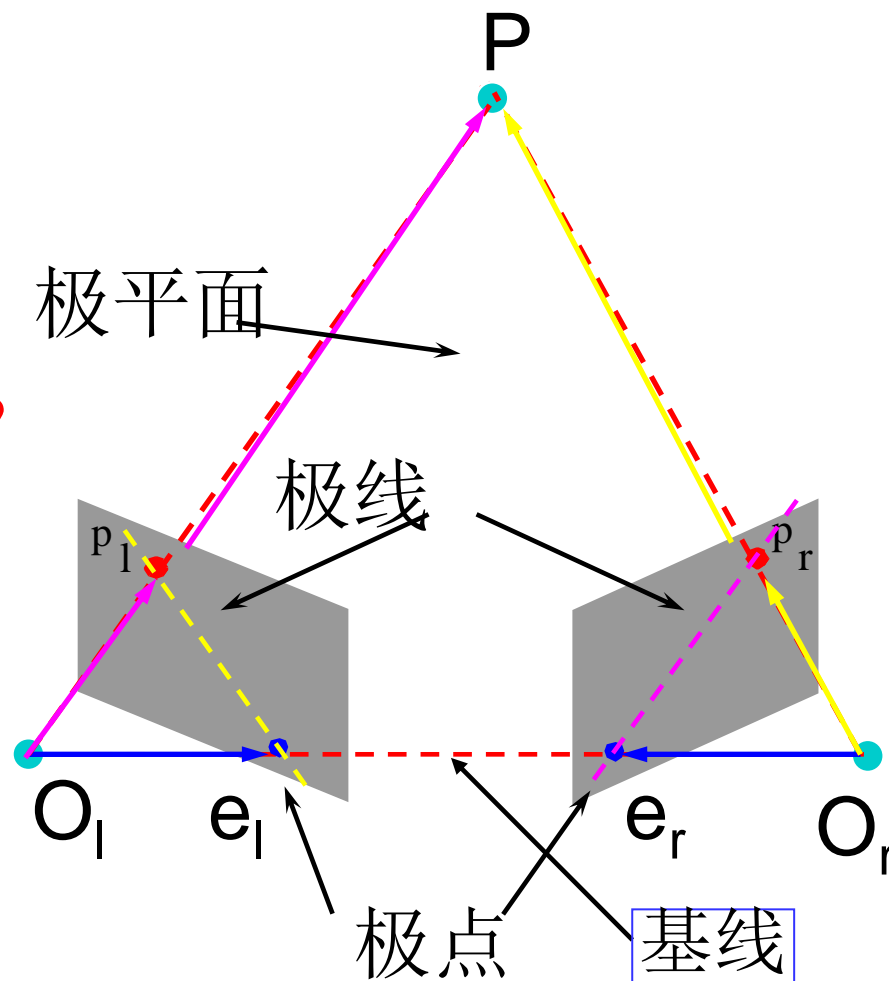
双目立体匹配

- 极线约束
- 思考:
- 动机: 在哪寻找匹配点?
 - 极平面
 - 极线
 - 极点
- 极线约束
 - 匹配点必须在极线上
 - 举例: p_r 在左图黄色极线





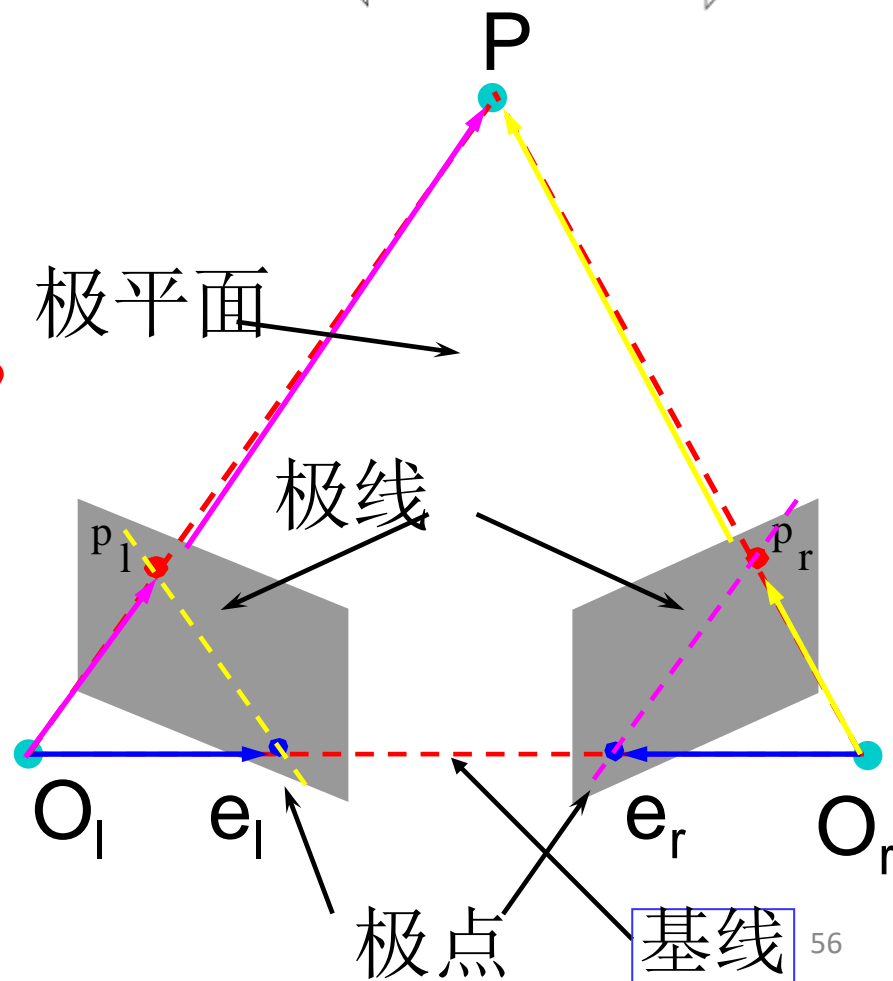
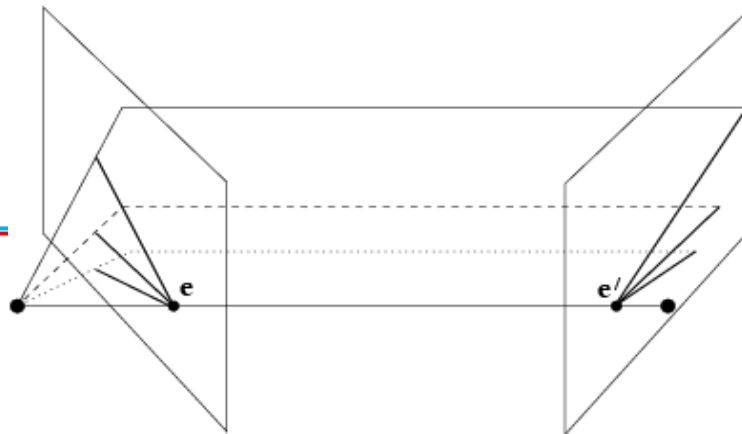
- 极线约束
- 思考:
- 极线有什么共同点?
- 极点是否一定在图像上?
- 所有的极线汇聚到极点
- 极点在像平面但是不一定在图像上

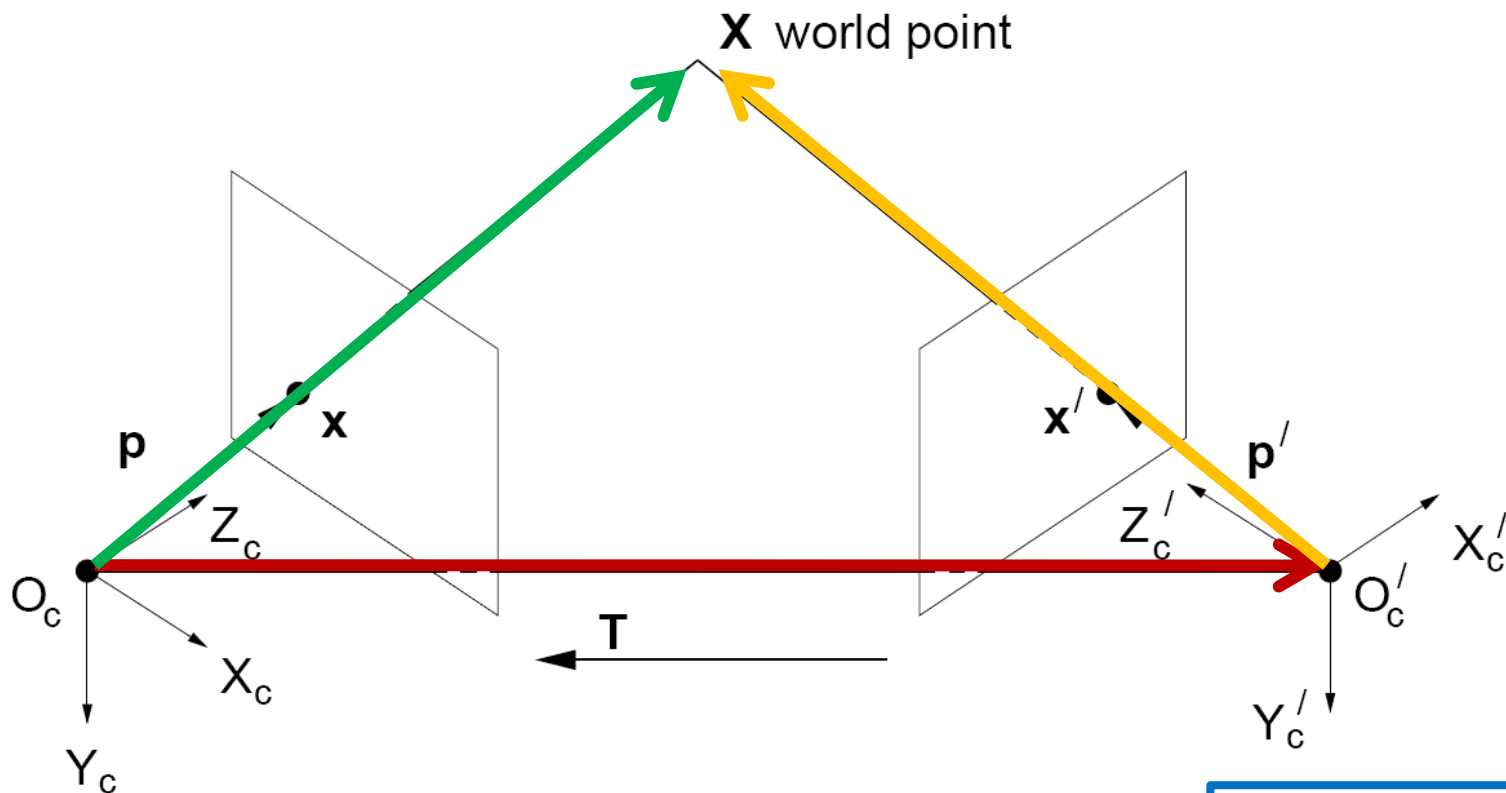




双目立体匹配

- 极线约束
- 思考:
- 极线约束关系什么时候确定?
- 极线约束关系如何确定?
- 极线对应关系是在相机标定时完成的
- 可以由相机的投影矩阵确定





$$\boxed{\mathbf{X}'} = \boxed{\mathbf{R}}\mathbf{X} + \boxed{\mathbf{T}}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{与极平面正交}} = \mathbf{T} \times \mathbf{R}\mathbf{X} + \mathbf{T} \times \mathbf{T} \\ = \mathbf{T} \times \mathbf{R}\mathbf{X}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X}) \\ = 0$$

本征矩阵

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X}) = 0$$

$$\mathbf{X}' \cdot (\mathbf{T}_x \mathbf{R}\mathbf{X}) = 0$$

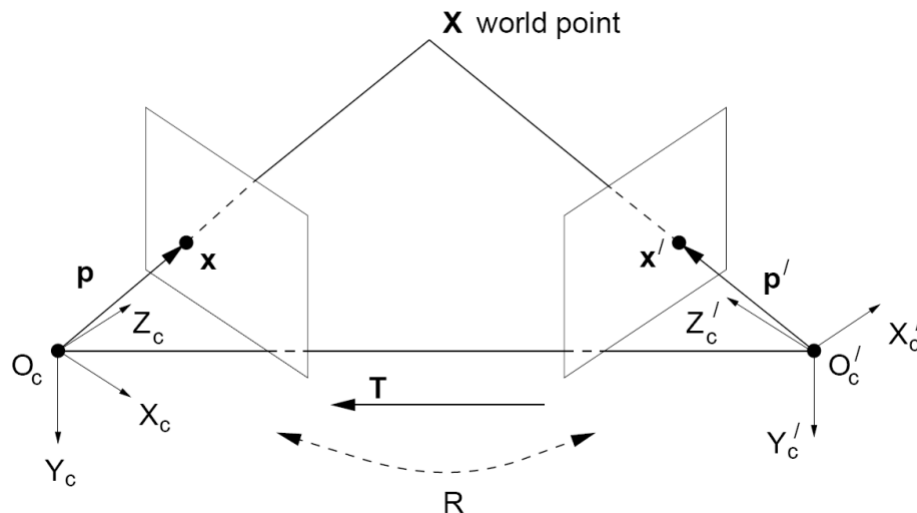
Let $\mathbf{E} = \mathbf{T}_x \mathbf{R}$

$$\mathbf{X}'^T \mathbf{E} \mathbf{X} = 0$$

对视线 \mathbf{p} 和 \mathbf{p}' 有:

$$\mathbf{p}'^T \mathbf{E} \mathbf{p} = 0$$

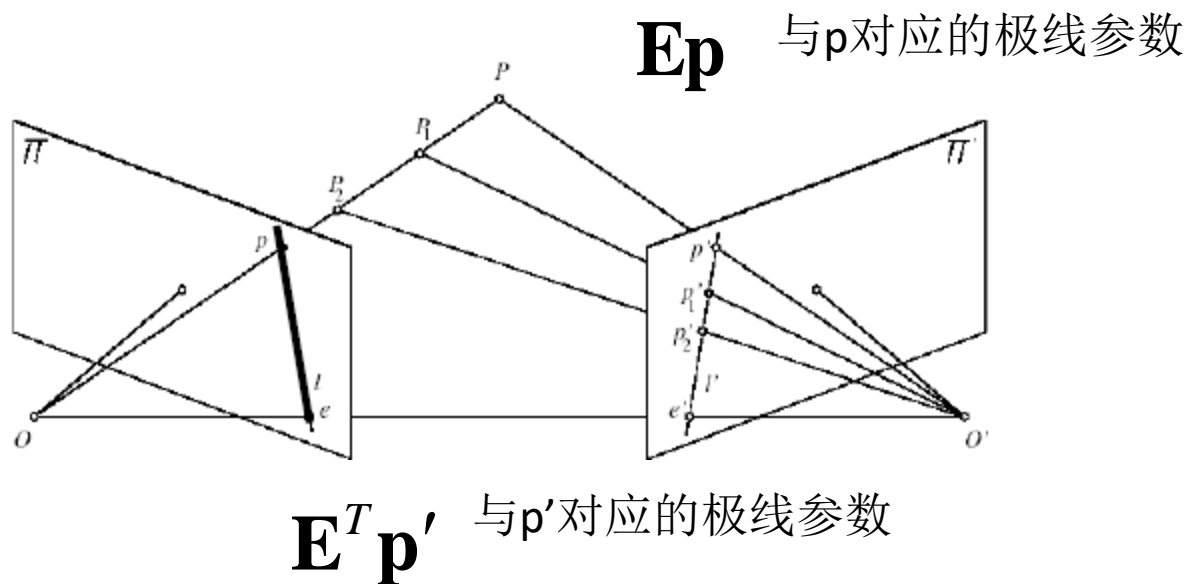
\mathbf{E} 称为本征矩阵[Longuet-Higgins 1981]



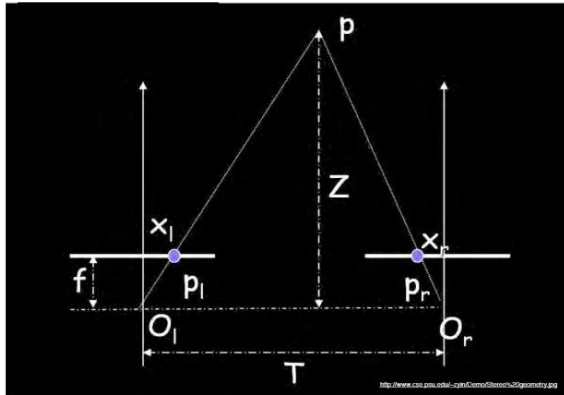
本征矩阵与极线

$$\mathbf{p}'^T \mathbf{E} \mathbf{p} = 0$$

对极约束: 一幅图像中的点 \mathbf{p} , 在另一幅图像中对应的点 \mathbf{p}' 应满足对极约束公式.



本征矩阵示例: 双目横向模式



$$\mathbf{R} = \mathbf{I}$$

$$\mathbf{T} = [-d, 0, 0]^T$$

$$\mathbf{E} = [\mathbf{T}_x] \mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix}$$

$$\mathbf{p}'^T \mathbf{E} \mathbf{p} = 0$$

$$\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

$$\Leftrightarrow \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ df \\ -dy \end{bmatrix} = 0$$

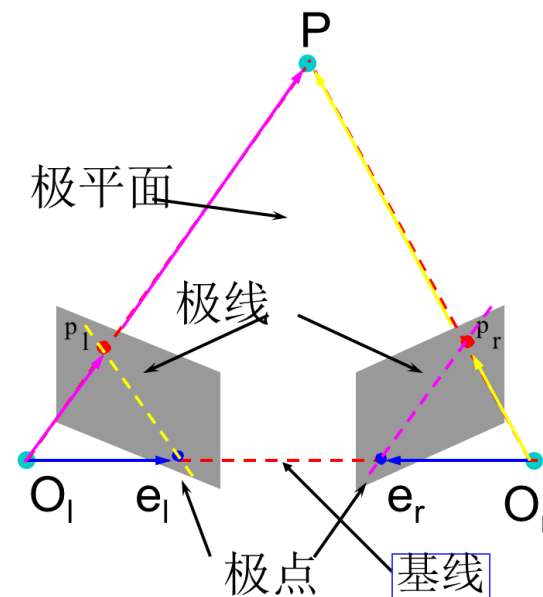
$$\Leftrightarrow y = y'$$

For the parallel cameras,
image of any point must lie on
same horizontal line in each
image plane.



双目立体匹配

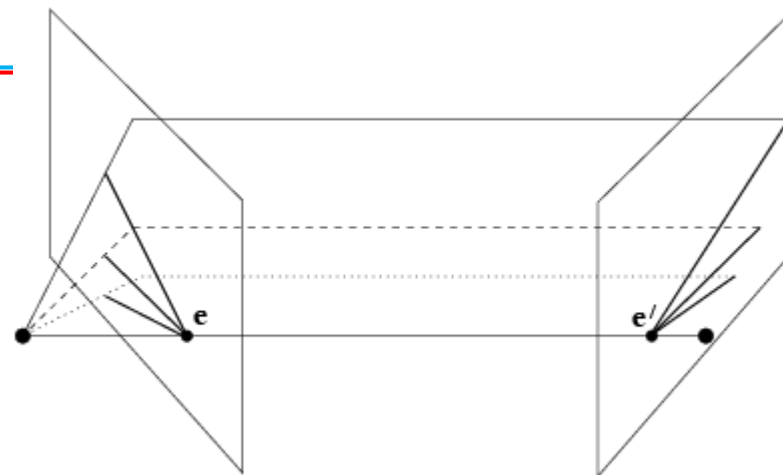
- 极线约束
- 思考：
- 极线约束的作用？
- 将立体匹配的对应点搜索从二维降到一维





双目立体匹配

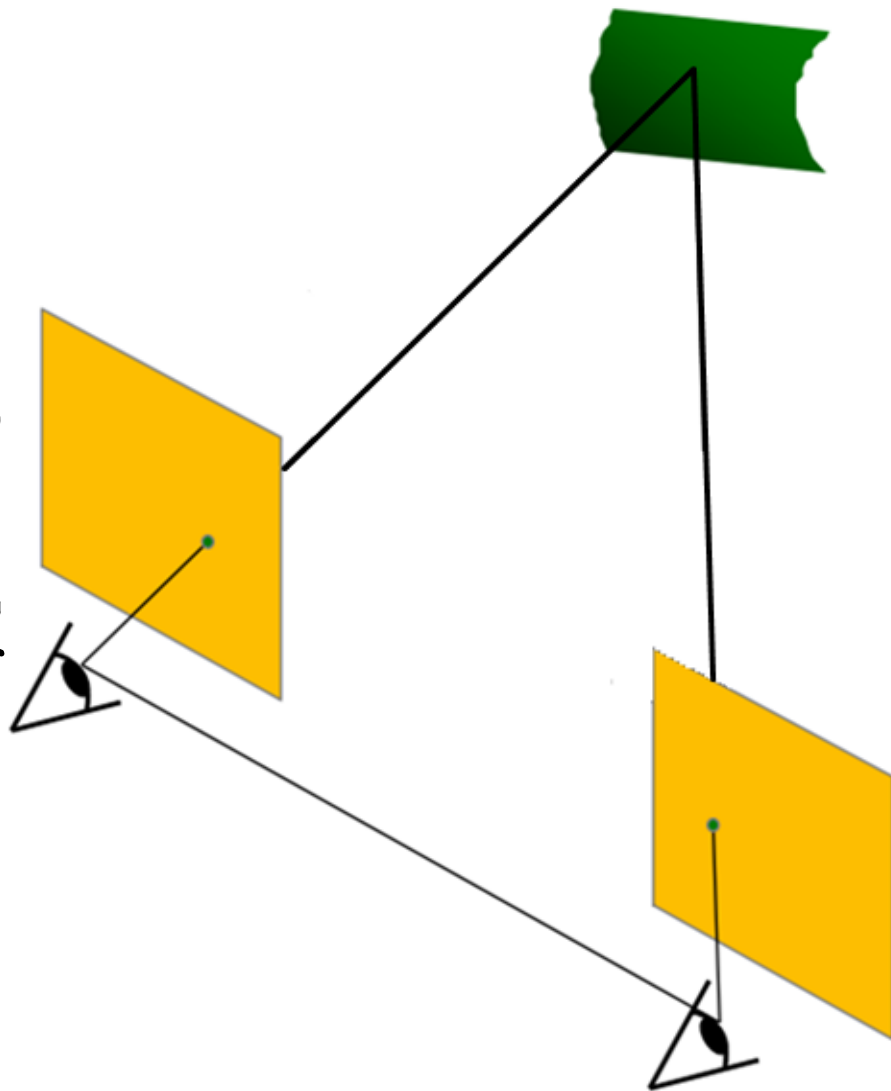
- 极线约束





双目立体匹配

- 极线约束
- 思考：
- 双目平行模式的极线约束，有什么特点？
- 对应的匹配点在图像相对应的扫描线上

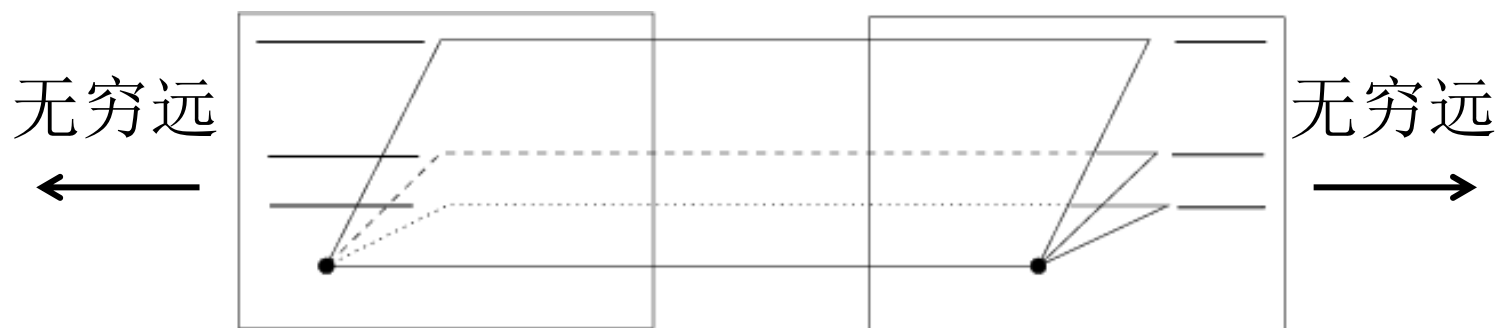




双目立体匹配

- 极线约束

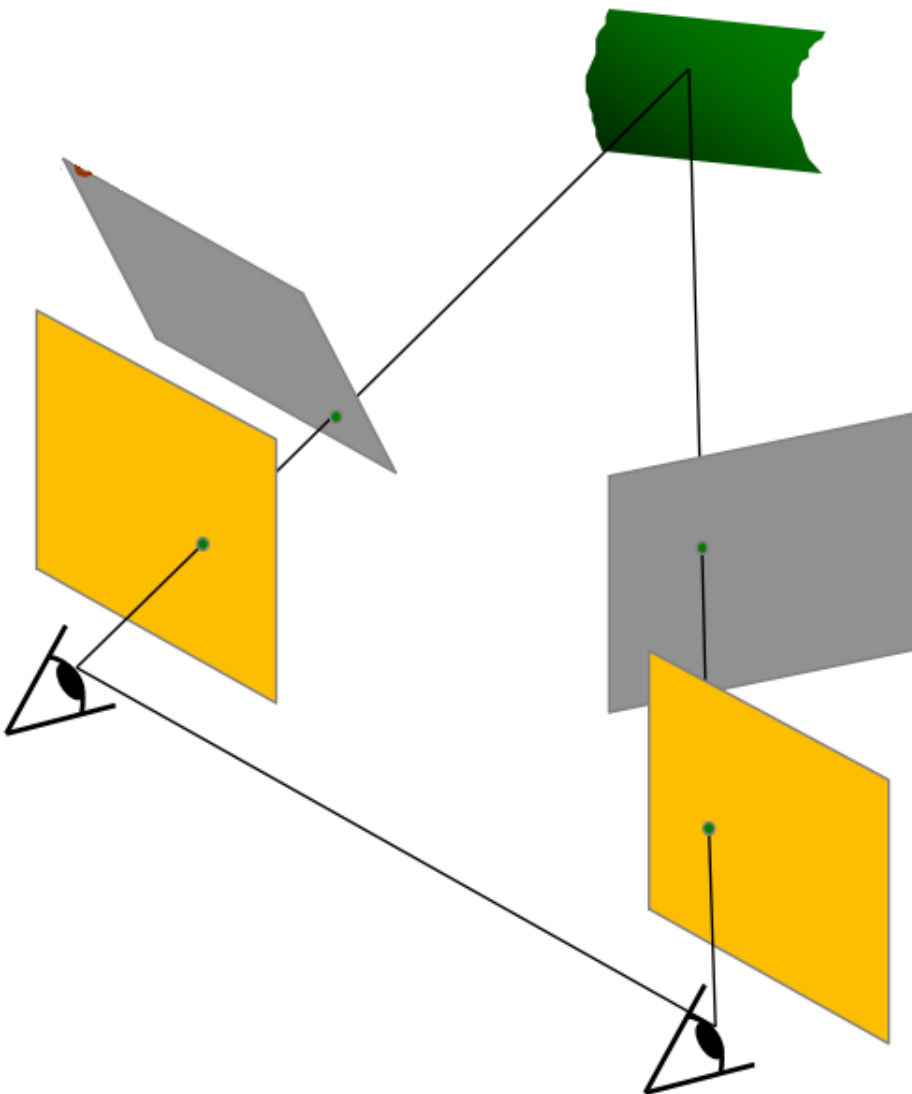
极点在哪里





双目立体匹配

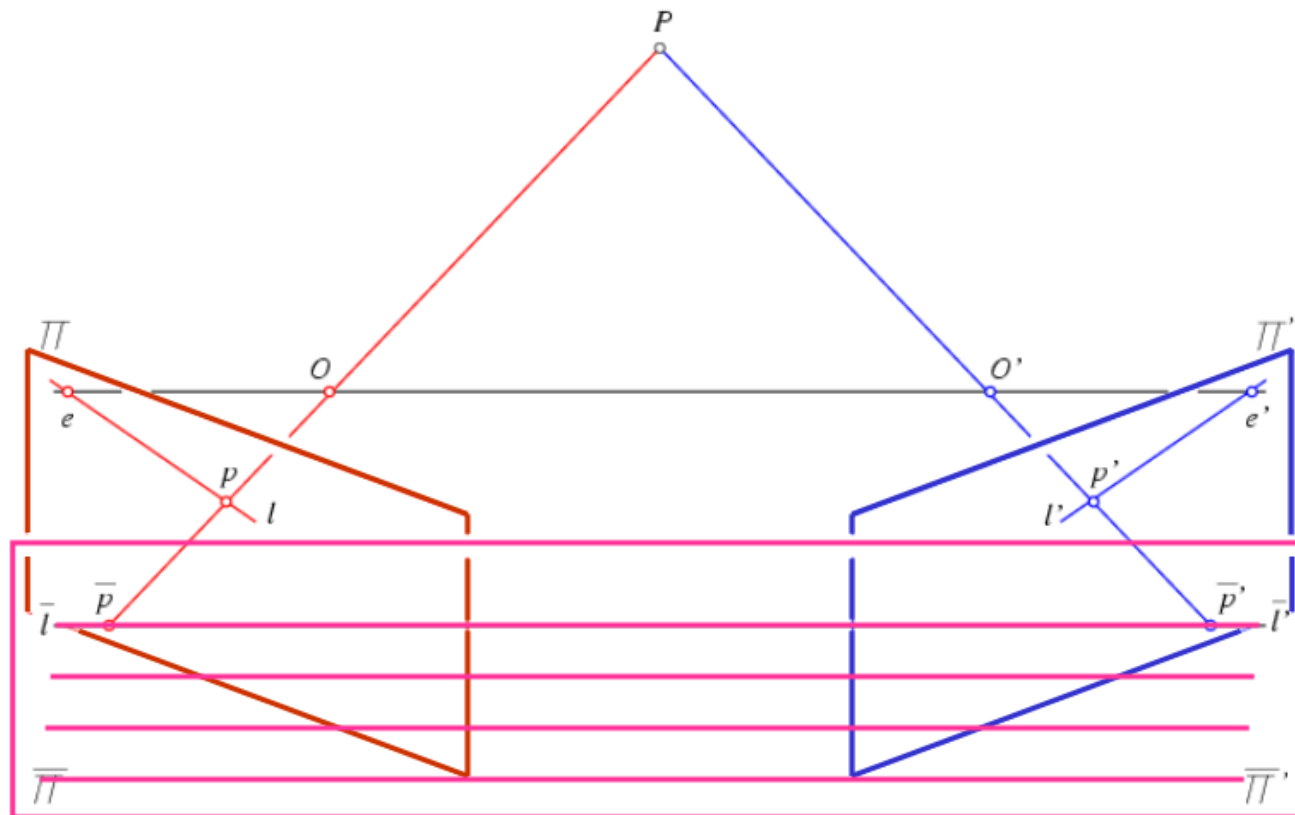
- 图像校正（重投影）
- 目的：规范化极线约束中的极线分布，使得匹配效率得到进一步的提高。
- 校正后，对应的匹配点在图像相对应的扫描线上





双目立体匹配

- 图像校正
- 在校正图像中所有极线都平行

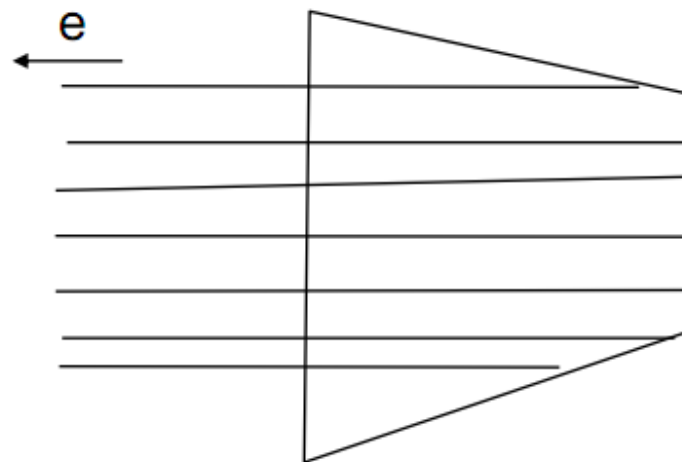
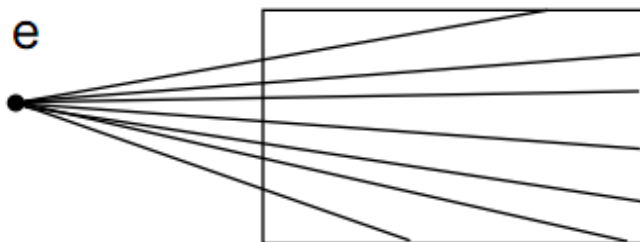




双目立体匹配

- 图像校正
- 在校正图像中所有极线都平行

把极点拉向无穷远处





双目立体匹配

- 图像校正

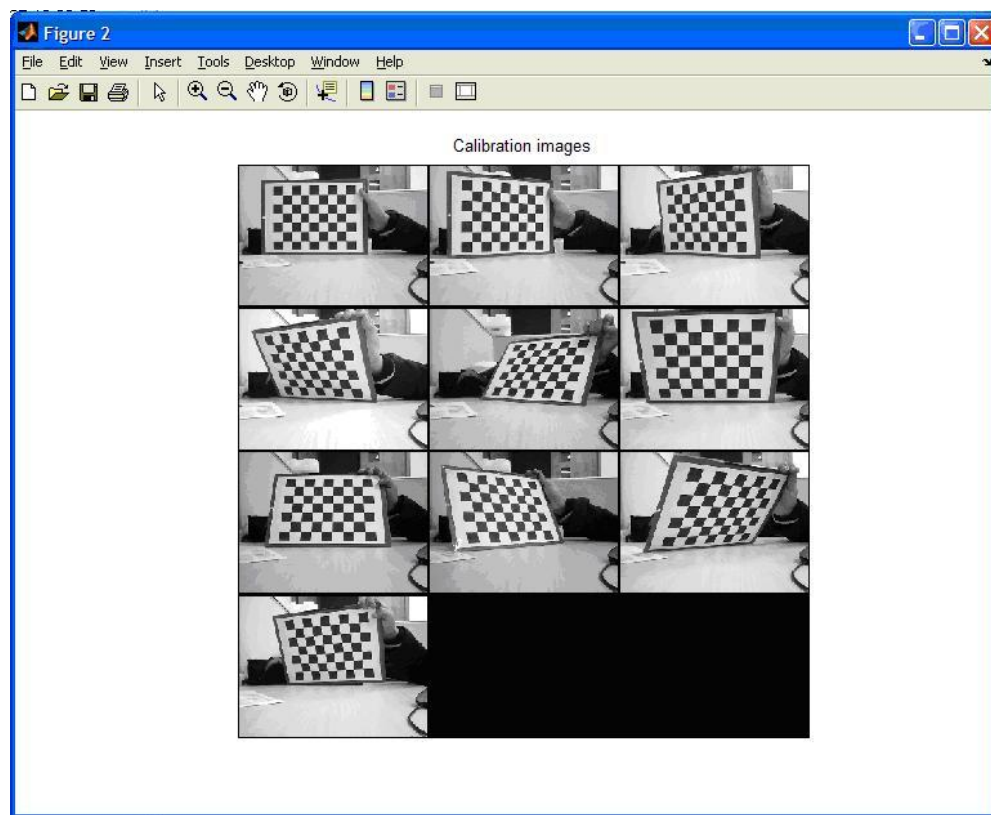


校正
后



双目立体匹配

- 双目相机标定过程示意

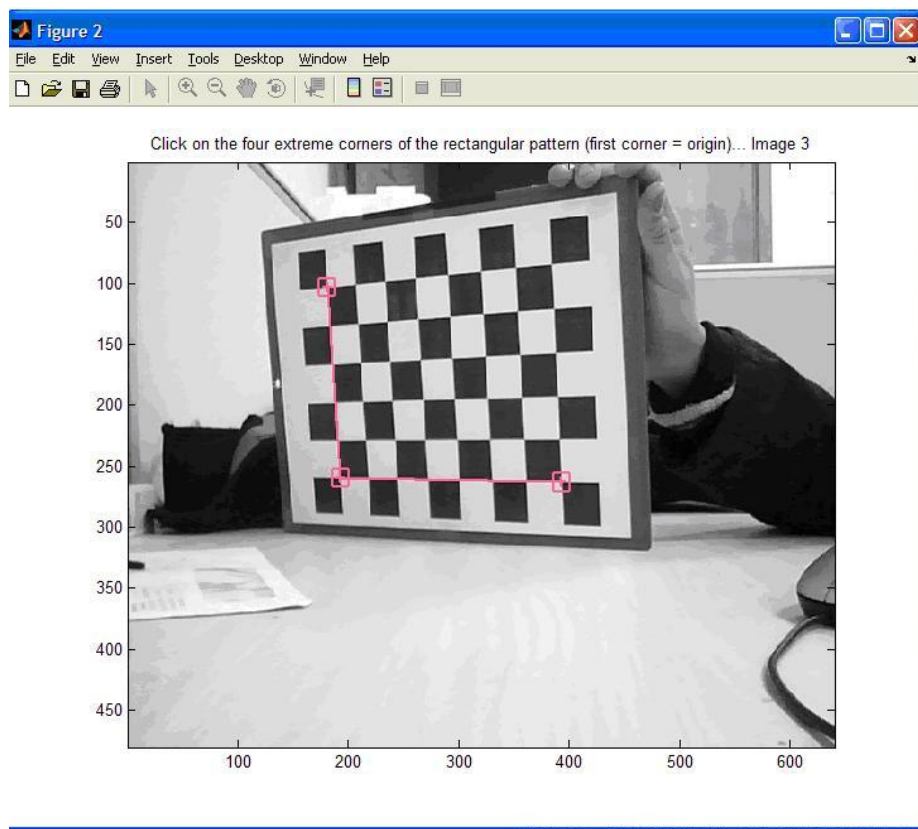


读入的棋盘图



双目立体匹配

- 双目相机标定过程示意

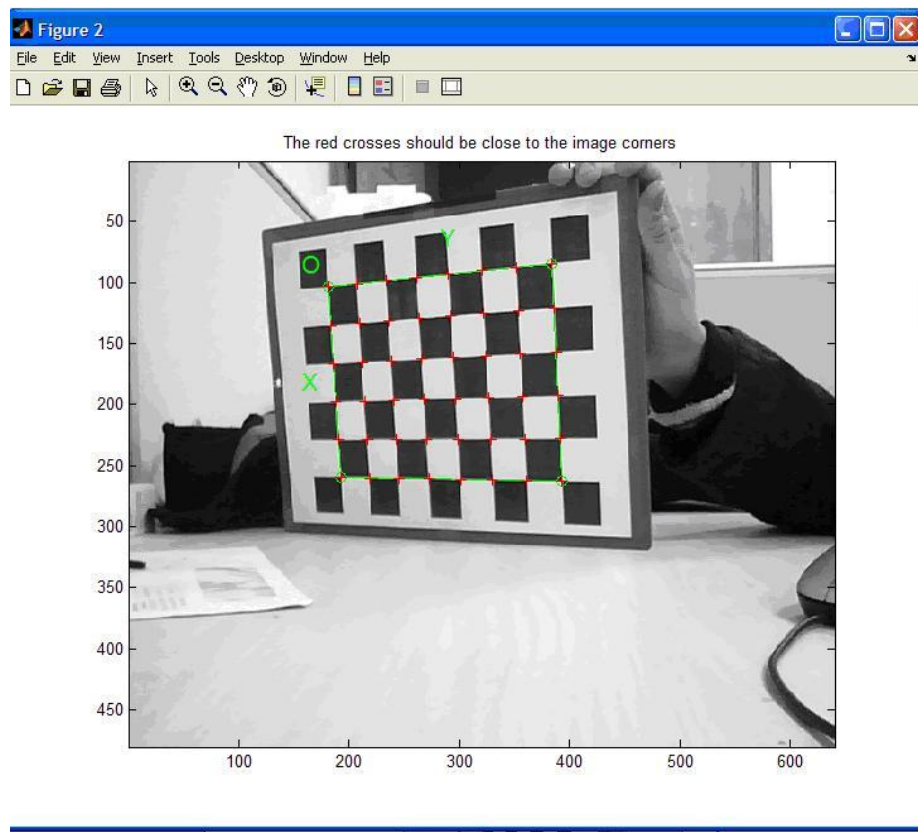


提取角点



双目立体匹配

- 双目相机标定过程示意



提取角点



双目立体匹配

- 双目相机标定过程示意

```
Stereo calibration parameters after optimization:

Intrinsic parameters of left camera:

Focal Length:      fc_left = [ 823.33854  823.69693 ] ?[ 6.27019  6.31344 ]
Principal point:    cc_left = [ 355.58113  211.36465 ] ?[ 16.59578  10.66193 ]
Skew:              alpha_c_left = [ 0.00000 ] ?[ 0.00000 ] => angle of pixel axes = 90.00000 ?0.00000 degrees
Distortion:         kc_left = [ -0.06537  0.22581  0.00199  -0.00055  0.00000 ] ?[ 0.06512  0.45806  0.00392

Intrinsic parameters of right camera:

Focal Length:      fc_right = [ 825.88737  825.95586 ] ?[ 6.18792  6.09389 ]
Principal point:    cc_right = [ 338.84525  224.54243 ] ?[ 16.31391  10.61733 ]
Skew:              alpha_c_right = [ 0.00000 ] ?[ 0.00000 ] => angle of pixel axes = 90.00000 ?0.00000 degrees
Distortion:         kc_right = [ -0.20431  0.65081  -0.00931  0.00599  0.00000 ] ?[ 0.06993  0.75854  0.0031

Extrinsic parameters (position of right camera wrt left camera):

Rotation vector:     om = [ 0.01631  -0.01954  0.00262 ] ?[ 0.01572  0.02714  0.00095 ]
Translation vector:  T = [ -231.48234  -1.10073  -6.42098 ] ?[ 1.71999  1.72043  14.19753 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).

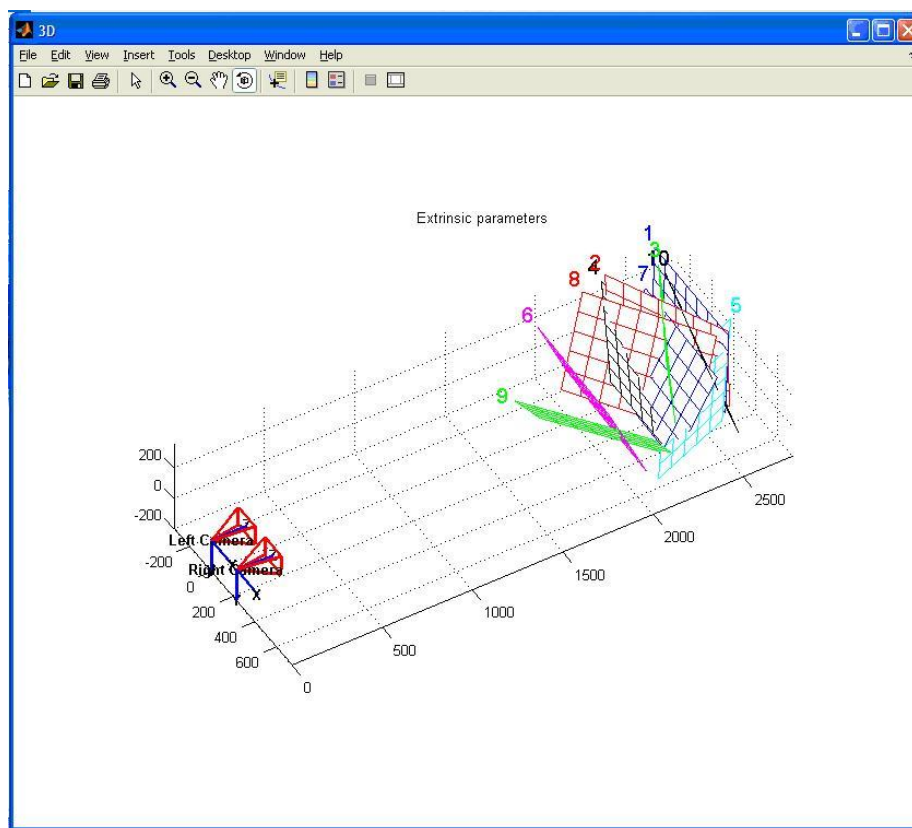
>> |
```

立体标定结果



双目立体匹配

- 双目相机标定过程示意

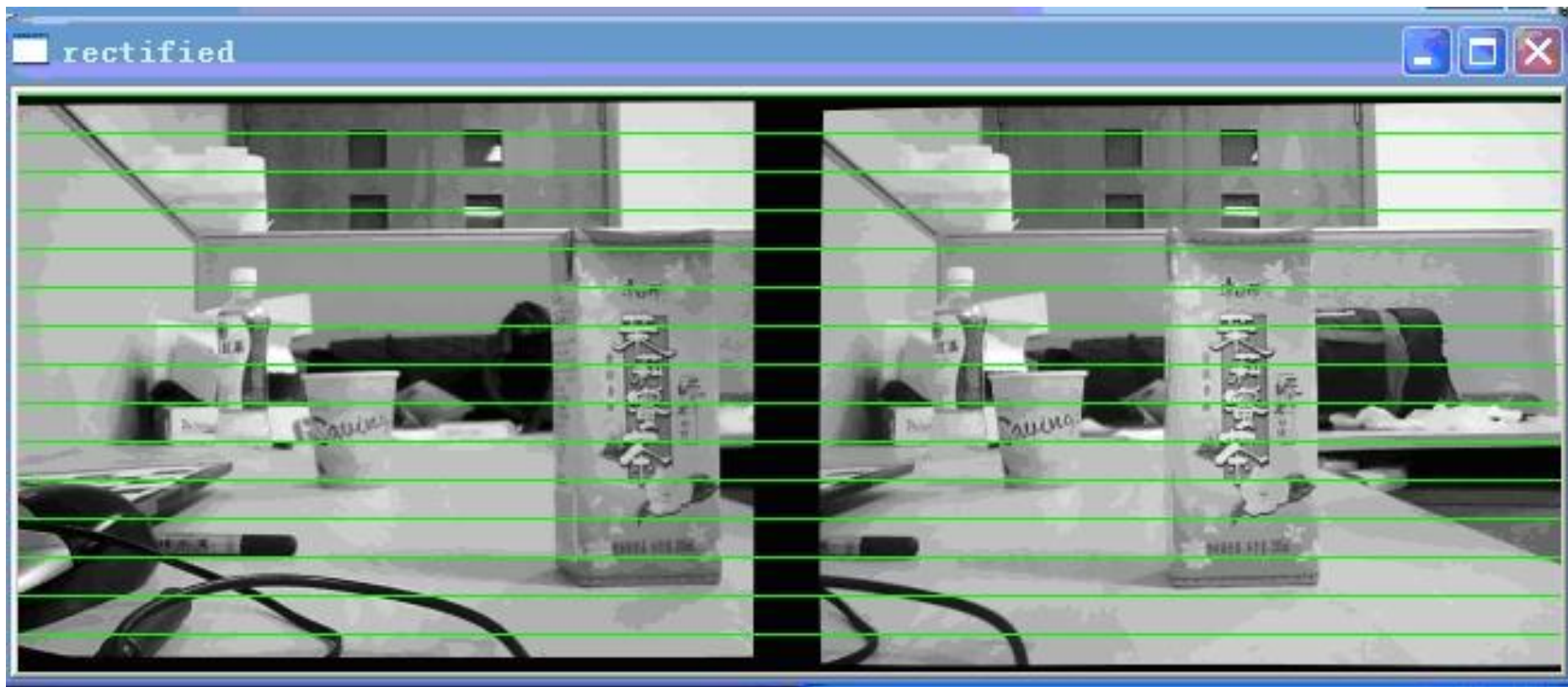


双摄像头与定标棋盘间的位置关系



双目立体匹配

- 双目相机标定过程示意



校正后的左右视图



9.3 基于区域的双目立体匹配

- 在双目视觉中，图像匹配的目的在于给定一幅图像上的已知点（或称为源匹配点）后，在另一幅图像上寻找与之对应的目标匹配点（或称为同名像点）。
- 图像匹配的方法通常有基于灰度（区域）的匹配、基于图像特征的匹配和基于解释的匹配或者多种方法结合



9.3 基于区域的双目立体匹配

- 图像匹配的方法通常有**基于灰度（区域）的匹配**、**基于图像特征的匹配**和**基于解释的匹配**或者多种方法结合
- 模板匹配
 - 直接用单点灰度搜索会受到图像中许多点会有相同灰度、图像噪声等因素影响而不实用



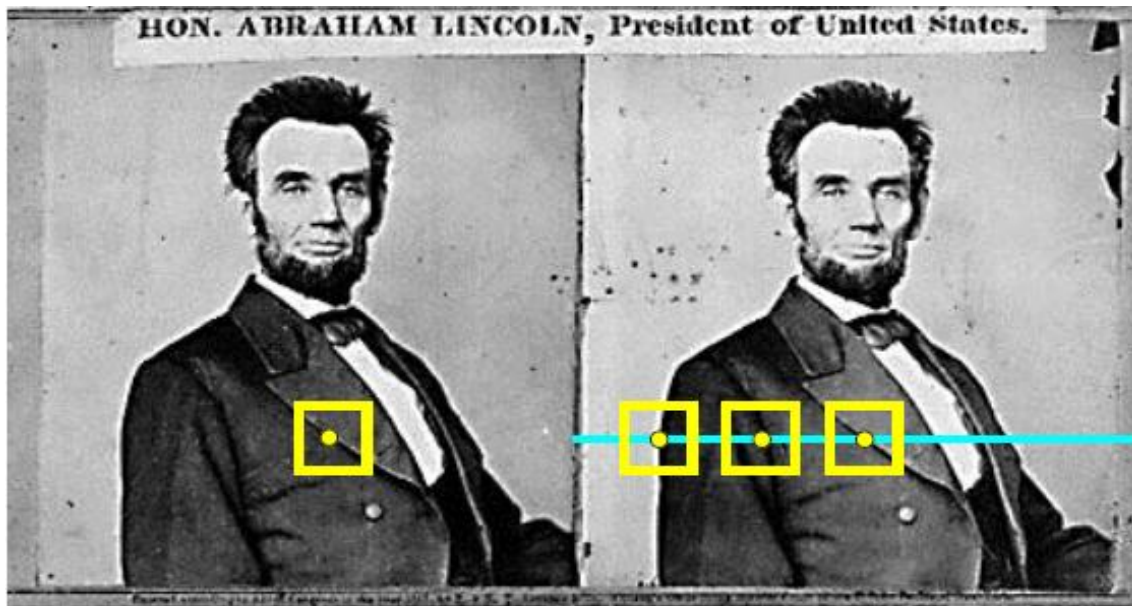
9.3 基于区域的双目立体匹配

- 基于区域的匹配
- 原理：在其中一幅图像中选取一个子窗口图像，然后在另一幅图像中的一个区域内，根据某种匹配准则，寻找与子窗口图像最为相似的子图像。常用的匹配准则有最大互相关准则、最小均方误差准则等。
- 区域匹配方法的计算量很大，而且匹配率较高，匹配精度较差。



9.3 基于区域的双目立体匹配

- 基于区域的匹配
- 原理：



- 区域匹配方法的计算量很大，而且匹配率较高，匹配精度较差。



9.3 基于区域的双目立体匹配

• 基于区域的匹配

最小绝对差算法 (SAD)

$$\sum_{(u,v) \in W} |I_1(u, v) - I_2(x + u, y + v)|$$

最小均差绝对算法 (ZSAD)

$$\sum_{(u,v) \in W} |(I_1(u, v) - \bar{I}_1) - (I_2(x + u, y + v) - \bar{I}_2)|$$

最小平方差算法 (SSD)

$$\sum_{(u,v) \in W} (I_1(u, v) - I_2(x + u, y + v))^2$$

最小均方差算法 (ZSSD)

$$\sum_{(u,v) \in W} (I_1(u, v) - \bar{I}_1 - (I_2(x + u, y + v) - \bar{I}_2))^2$$

归一化互相关算法

$$\frac{\sum_{(u,v) \in W} I_1(u, v) \cdot I_2(x + u, y + v)}{\sqrt{\sum_{(u,v) \in W} I_1^2(u, v) \cdot \sum_{(u,v) \in W} I_2^2(x + u, y + v)}}$$

去均值归一化互相关算法

$$\frac{\sum_{(u,v) \in W} (I_1(u, v) - \bar{I}_1) \cdot (I_2(x + u, y + v) - \bar{I}_2)}{\sqrt{\sum_{(u,v) \in W} (I_1(u, v) - \bar{I}_1)^2 \cdot \sum_{(u,v) \in W} (I_2(x + u, y + v) - \bar{I}_2)^2}}$$



9.3.1 模板匹配

■ 相关函数

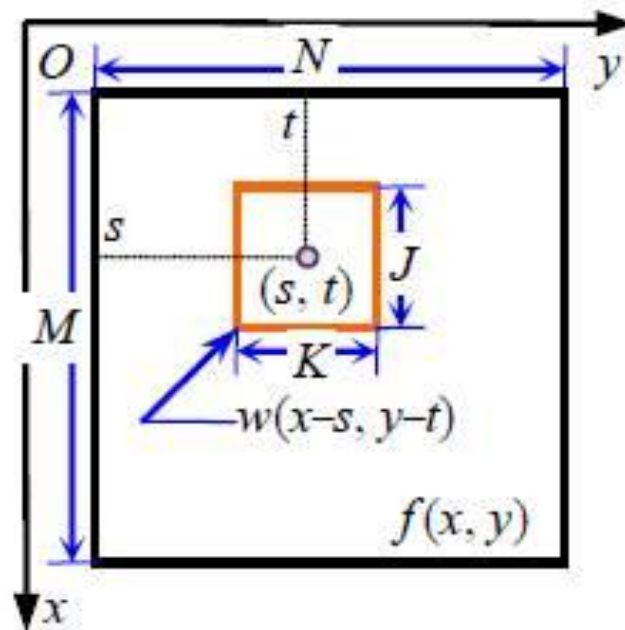
$$c(s, t) = \sum_x \sum_y f(x, y) w(x - s, y - t)$$

对 $f(x, y)$ 和 $w(x, y)$

幅度值的变化比较敏感

■ 相关系数

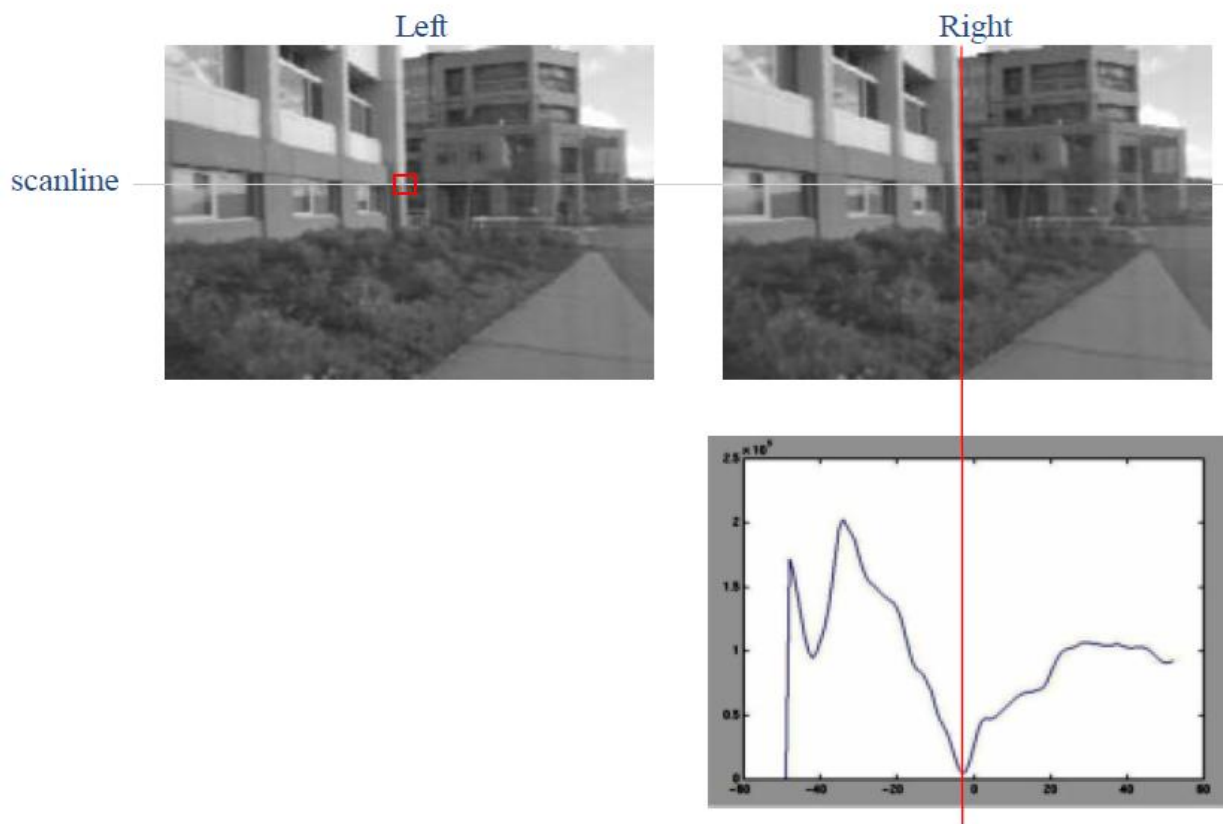
$$\gamma(s, t) = \frac{\sum_x \sum_y [f(x, y) - \bar{f}(x, y)] [w(x - s, y - t) - \bar{w}]}{\left\{ \sum_x \sum_y [f(x, y) - \bar{f}(x, y)]^2 \sum_x \sum_y [w(x - s, y - t) - \bar{w}]^2 \right\}^{1/2}}$$





9.3 基于区域的双目立体匹配

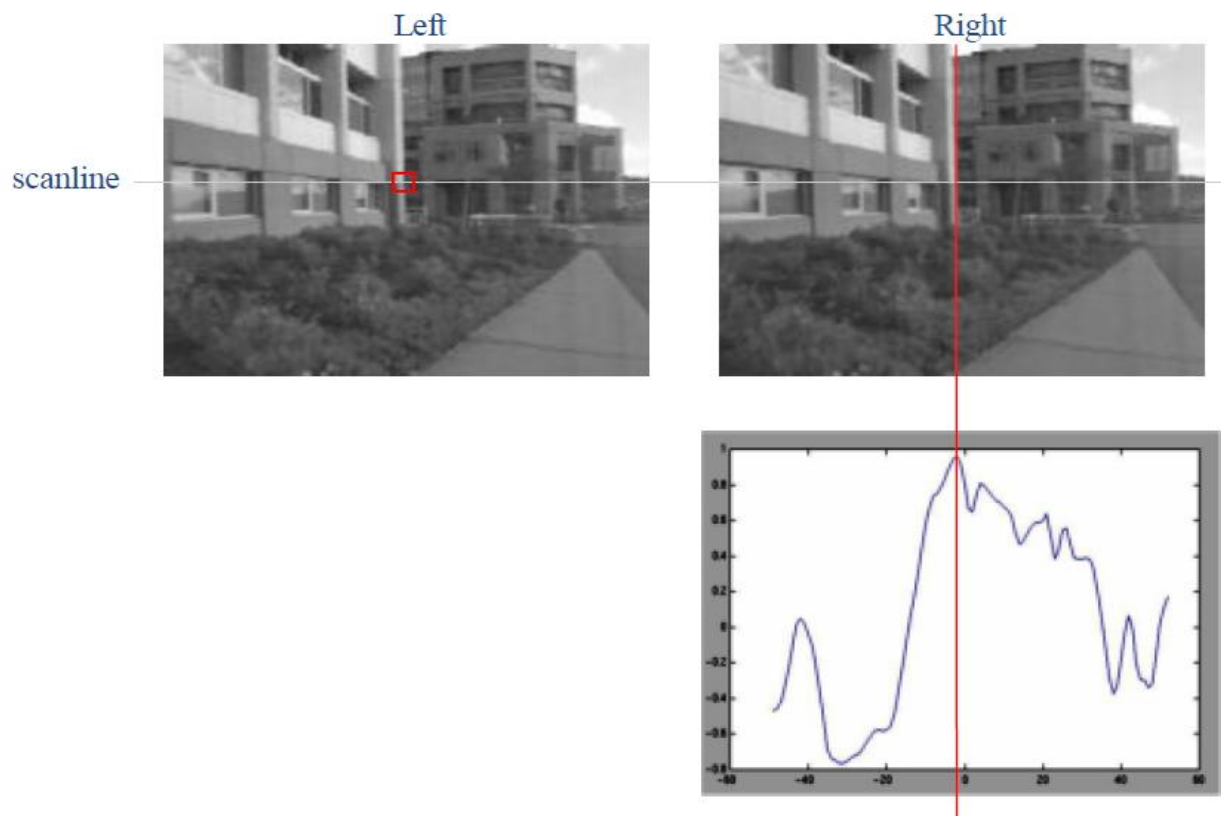
- 基于区域的匹配
- 最小均方差





9.3 基于区域的双目立体匹配

- 基于区域的匹配
- 归一化互相关



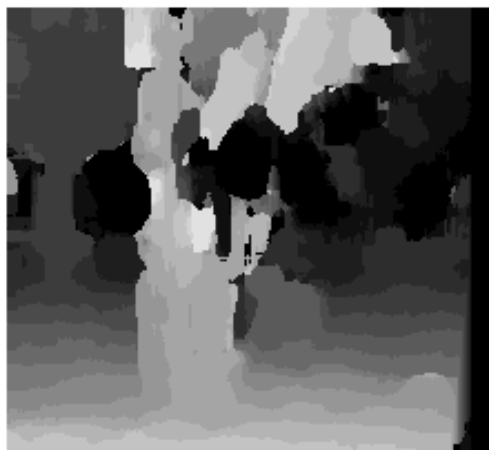


9.3 基于区域的双目立体匹配

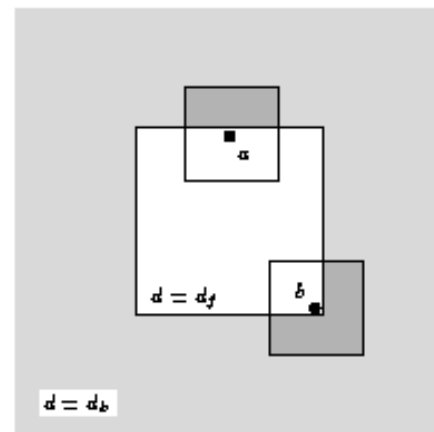
- 基于区域的匹配
- 窗口大小影响
 - 小窗口：更多细节
 - 大窗口：更少孤立的错误点



$w = 3$



$w = 20$





9.3 基于区域的双目立体匹配

- 2. 匹配影响因素
 - (1) 由于景物自身形状或景物互相**遮挡**的原因，用左图象确定的某些模板不一定能在右图象中找到完全匹配的位置。此时常需根据其他匹配位置的匹配结果来**插值**这些无法匹配点的数据
 - (2) 不同模板图像应有不同模式。但在**平滑区域**得到的模板图象具有相同或相近的模式，使匹配有不确定性，并导致产生误匹配。此时可需要将一些**随机的纹理投影**到这些表面上以将平滑区域转化为纹理区域



9.4 基于特征的双目立体匹配

- 基于区域方法的缺点：
 - 区域匹配要求在每个相关窗口中都存在可探测纹理特征，对于较弱特征和存在重复特征情况匹配容易失效
 - 区域匹配对绝对光强、对比度和照明条件敏感
 - 区域匹配不适用于深度变化剧烈的场合
 - 区域匹配计算量大，匹配速度慢



9.4 基于特征的双目立体匹配

- 6.4.1 基本方法
- 6.4.2 动态规划匹配（不讲）



9.4.1 点对点的方法

- 基于特征匹配方法

基于抽象的几何特征，如角点、边缘及边缘直线等几何基元的形状及参数等进行特征点匹配。由于几何特征的稀疏性和不连续性，特征匹配方式只能获得稀疏的深度图，需要各种内插方法才能完成整幅深度图提取。

- (1) 参与匹配的特征少于区域匹配所需要点，因此速度快。
- (2) 特征匹配需要对两幅图像进行特征提取，增加计算量。
- (3) 几何特征提取可达到“亚像素”级，特征匹配精度高。
- (4) 匹配元素为物体几何特征，对照明变化不敏感。



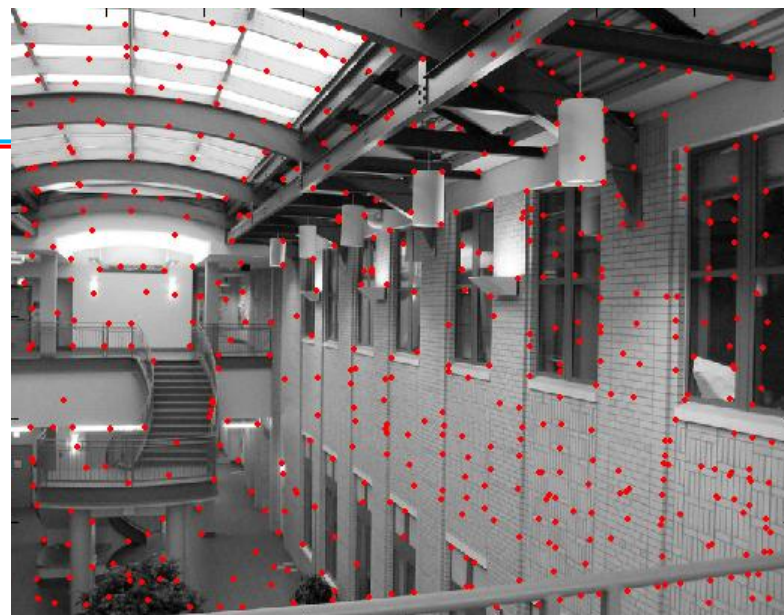
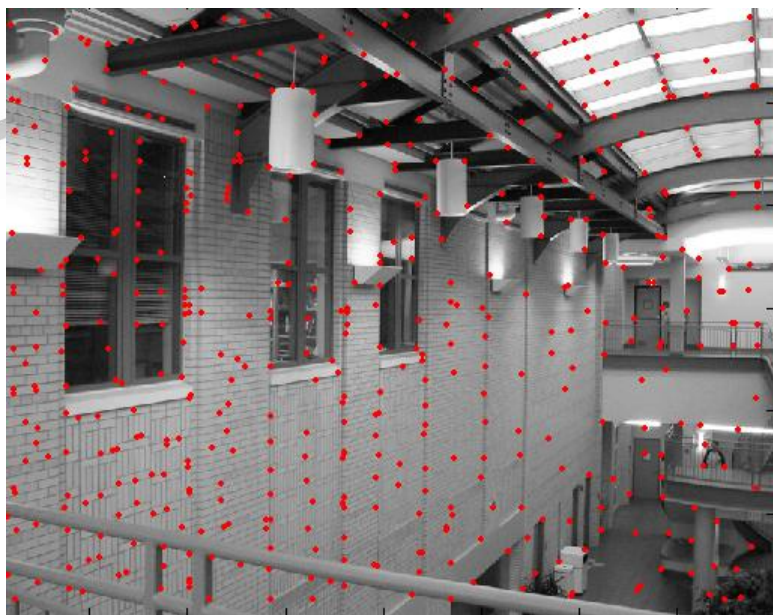
9.4.1 点对点的方法

- 典型方法的主要步骤
 - (1) 确定特征点（比如，用边缘检测寻找物体的轮廓线，并在轮廓线上确定特征点）
 - (2) 利用立体匹配方法匹配各特征点
 - (3) 对匹配点求视差，获取匹配点的深度
 - (4) 利用获得的匹配点进行深度插值，以进一步得到其它各点的深度



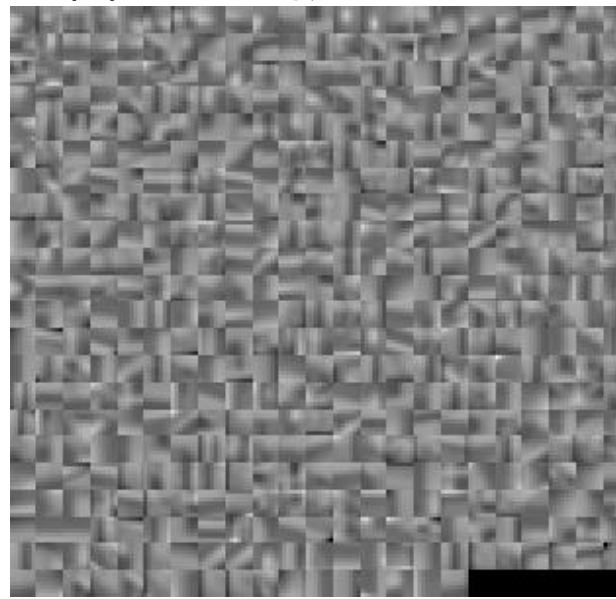
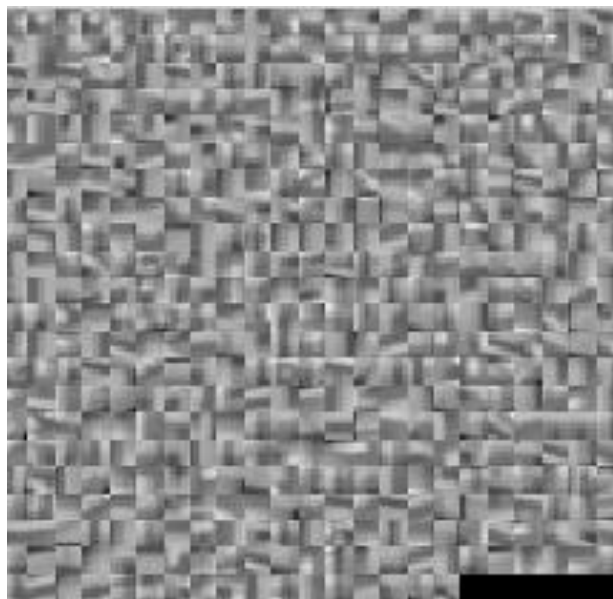
9.4.1 点对点的方法

- 特征点提取：平移、旋转、缩放、仿射不变性
 - 1.SUSAN角点提取；
 - 2.harris角点提取；
 - 3.SIFT尺度不变特征提取
- 特征点匹配：距离度量



左图特征点对应(区域)特征

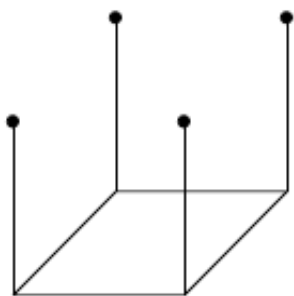
右图特征点对应(区域)特征



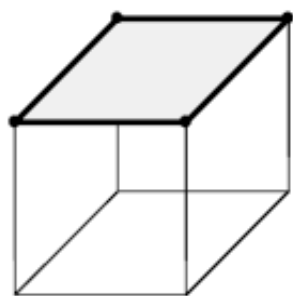


9.4.1 点对点的方法

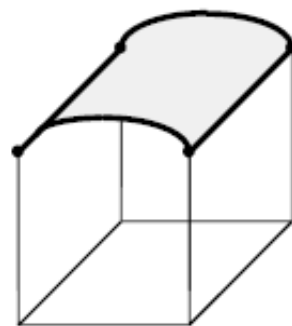
- 注意：仅由稀疏的匹配点并不能唯一地恢复 物体外形
- 示例：



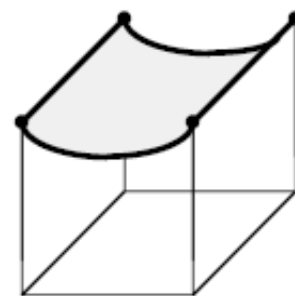
(a)



(b)



(c)



(d)

- 过这4个点的曲面可以有无穷多个



特征匹配 VS 区域匹配

- **特征匹配 (Feature match):**
 - 速度快，匹配效率高；
 - 特征的提取可以到亚像素级别，精度较高；
 - 匹配元素为物体的几何特征，对照明变化不敏感；
 - 重建需要拟合。
- **区域匹配 (Dense match):**
 - 重建不需要拟合；
 - 速度慢，效率低；
 - 对于无纹理，纹理不明显的图像匹配效果不理想；
 - 对光强、对比度、照明条件敏感。

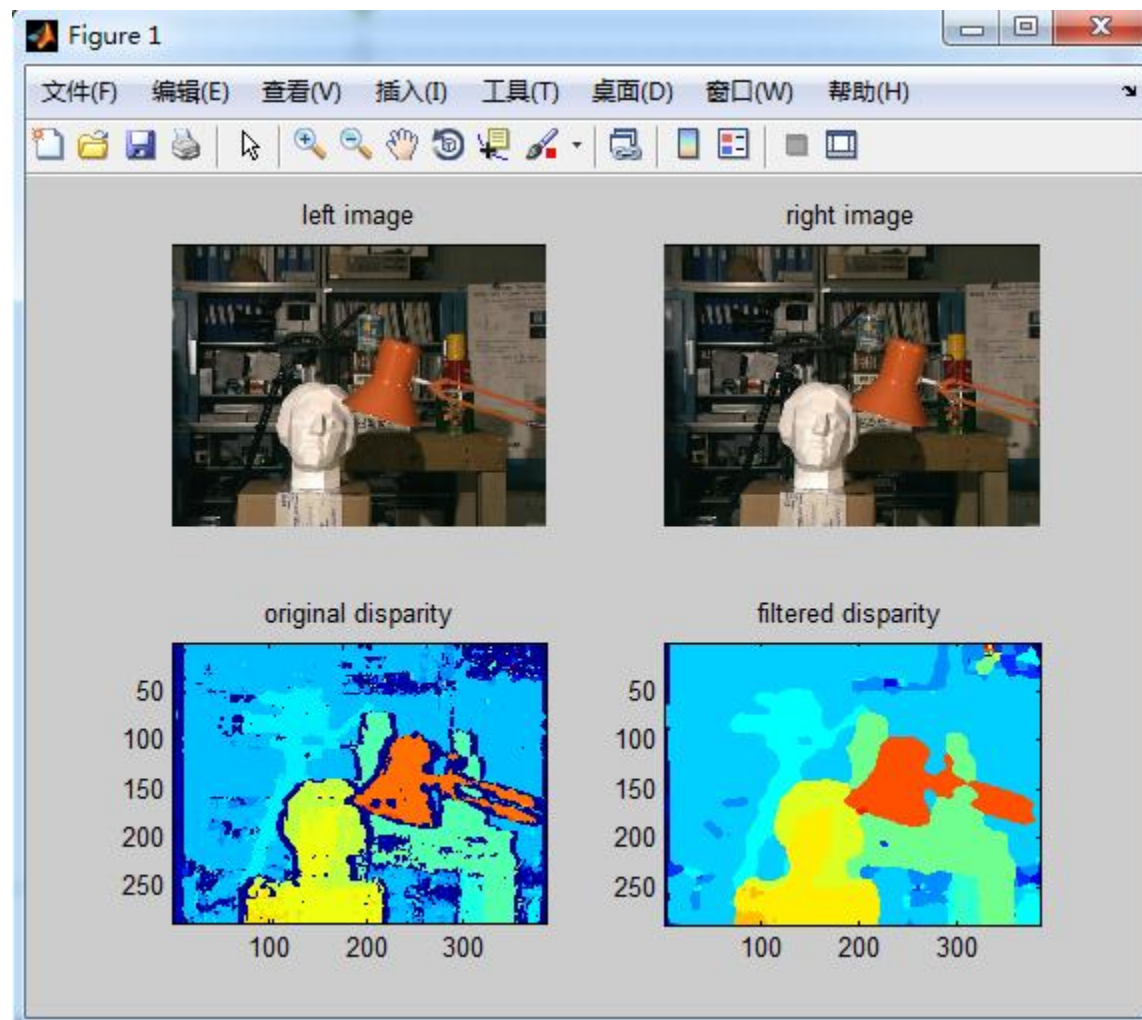


立体匹配

- **局部算法 (Local/window-based algorithms):**
 - 在匹配点的一个特定窗口中计算相似度。
 - SSD, SAD, MSE, NCC等
- **全局算法 (Global algorithms):**
 - 能量方程:
 - 模拟退火(Simulated annealing), 动态规划(Dynamic Programming), 图割(graph-cut), 等



- 实现
 - Opencv
 - Matlab





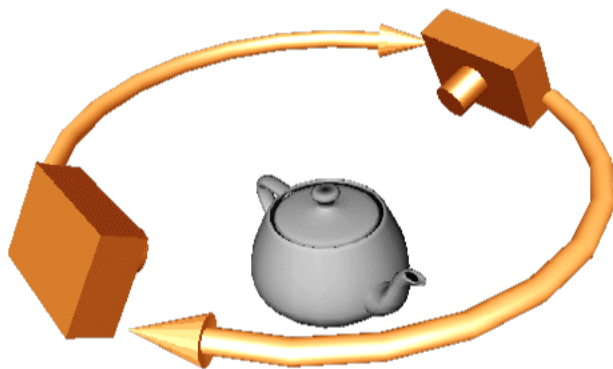
三维重建

- 完整的三维重建
 - 多个相机同时
 - 序列图像

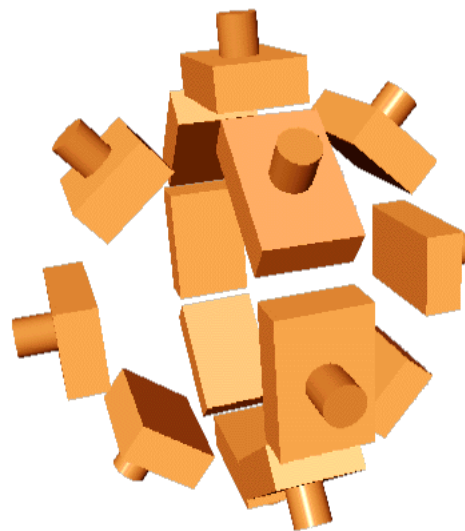




三维重建



向内



向外



三维重建



平台旋转





作业

- 9.2 在下图中，如设坐标原点在两个光心连线的中点处，试求 Z 的表达式

如 $\lambda = 0.05\text{m}$, $B = 0.4\text{m}$, $x_1 = 0.02\text{m}$,
 $x_2 = -0.03\text{m}$, 算出 W 点的 X 和 Z 坐标。

