

专业限选系列课程

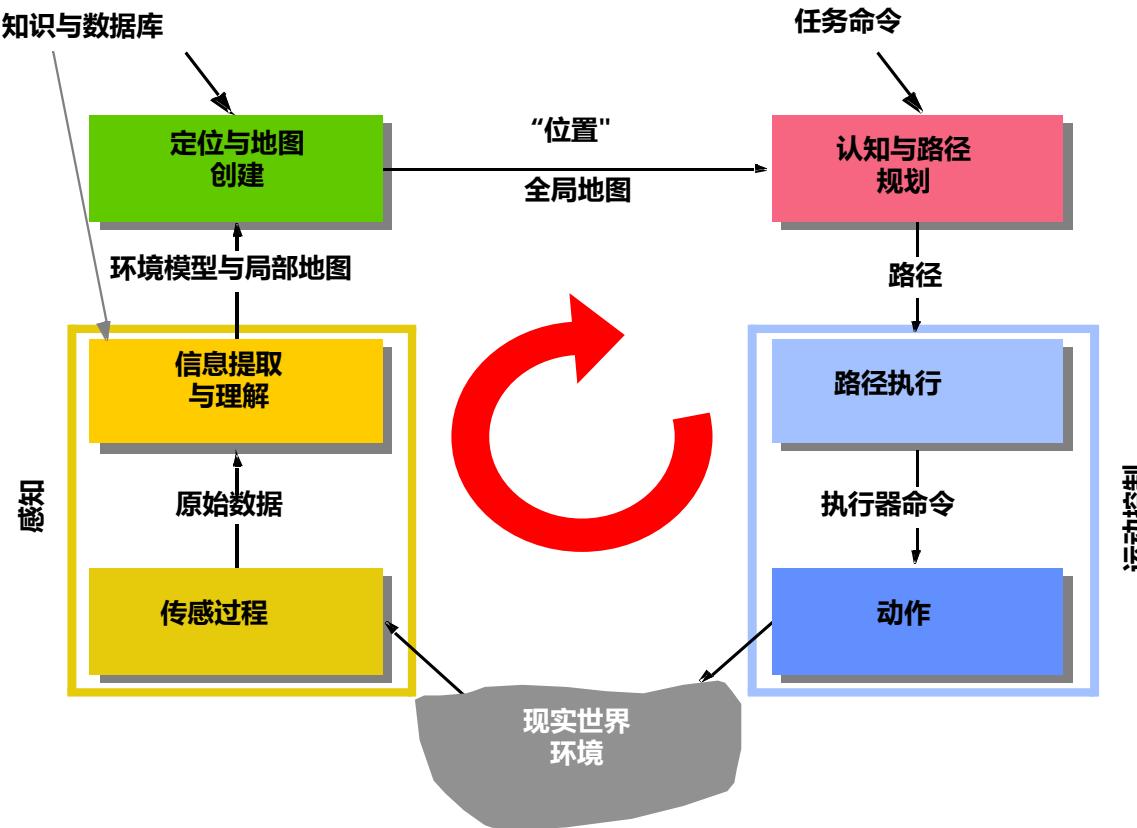
智能机器人技术

---Introduction to Intelligent Robotics

赵振刚 gavin@ustc.edu.cn

机器人学的知识架构

- 1. 感知**: 视觉传感器、图像传感器、触觉和力传感器、惯导等；
- 2. 认知**: 人工智能、知识表达、规划、任务调度、机器学习等；
- 3. 行为**: 运动学、动力学、控制、manipulation和locomotion等；
- 4. 数学基础**: 最优估计、微分几何、计算几何、运筹学等；

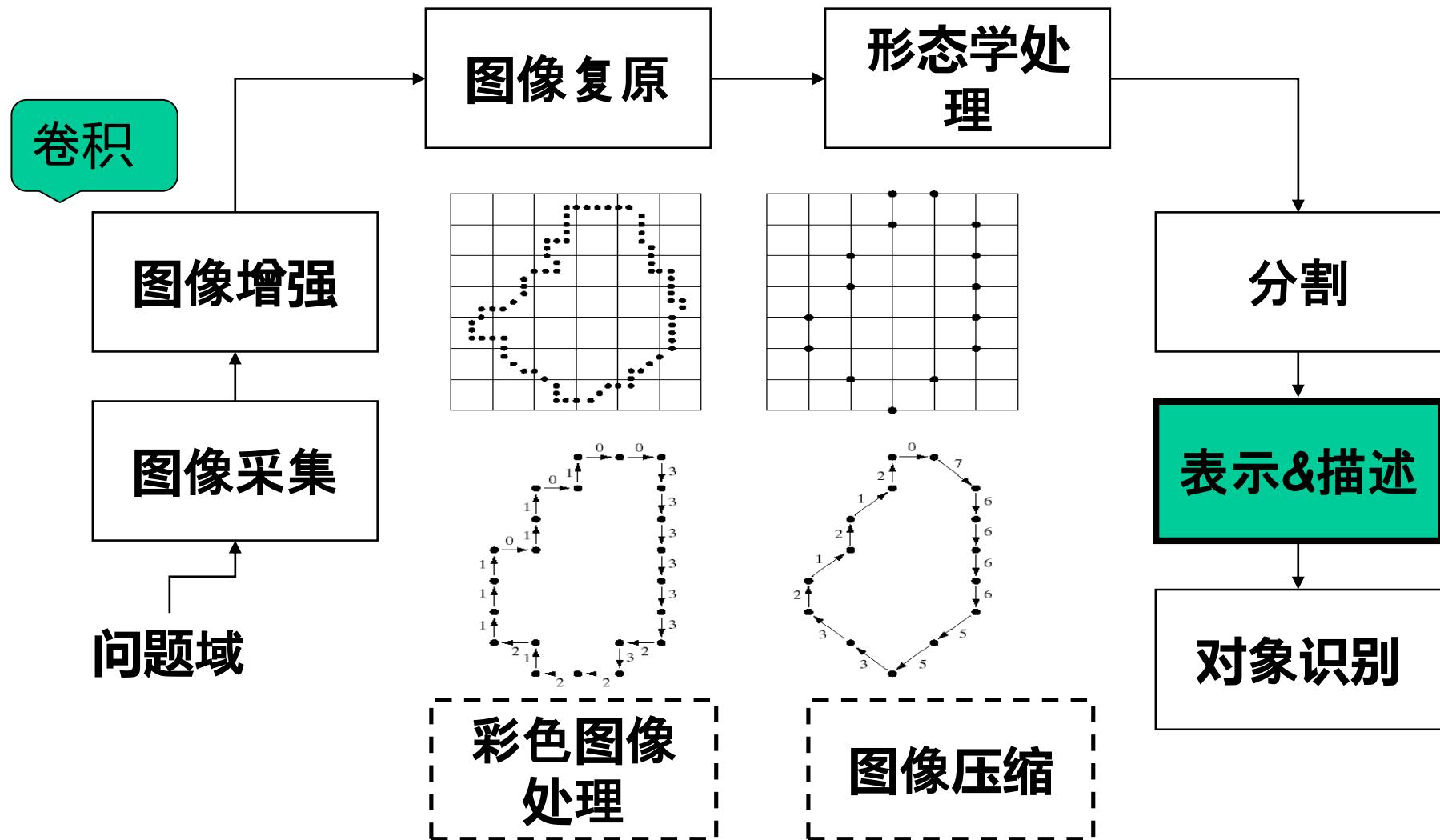


回顾上节 2D数字图像处理

- ❖ 数字图像:用二维阵列表示的影像
 - ❖ 颜色空间, 图像通道, 坐标系, 边缘



回顾上节 2D数字图像处理



2D视觉

【CV竞赛】百度车道线检测，正在报名中，8.9万元奖金

我爱计算机视觉 11月19日



视觉概述/Vision in General

视觉是最有优势的感知方式：
信息获取量大，支持与环境的非接触交互

视觉是最复杂的感知方式
对人脑处理人眼影像信息的理解仍然处于初级阶段

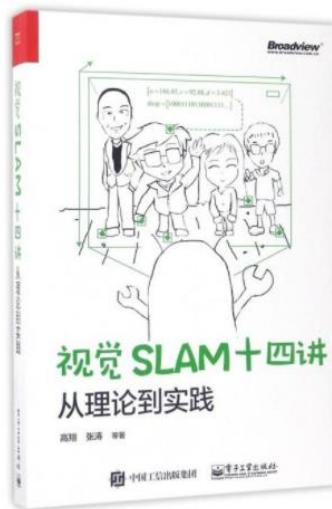
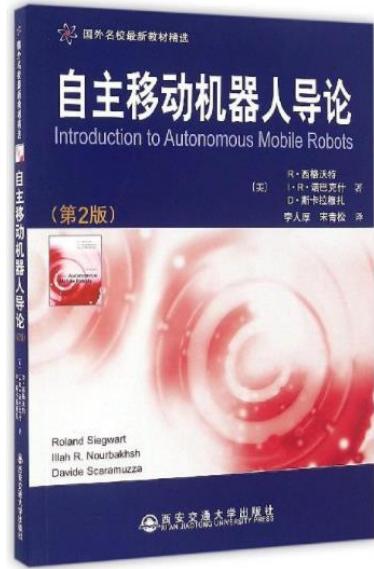
成像是三维场景在二维平面投影的结果
从2维影像解读三维景象是机器视觉的核心任务

如何获取、处理包含2.5/3维信息的图像？

第四章 立体视觉系统

目标:

- ❖ 掌握常见立体视觉系统的基本原理，点云图拼接实践
- ❖ 了解同时定位与地图构建SLAM原理

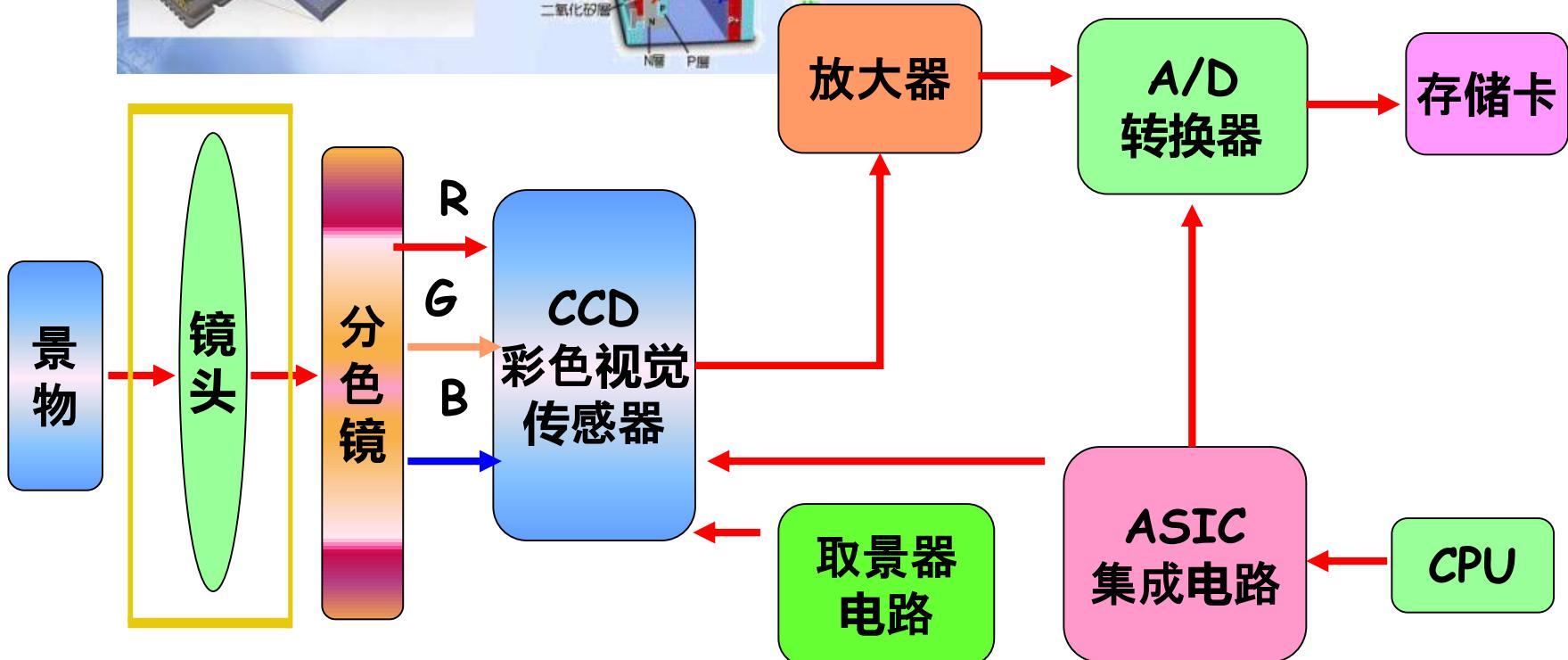
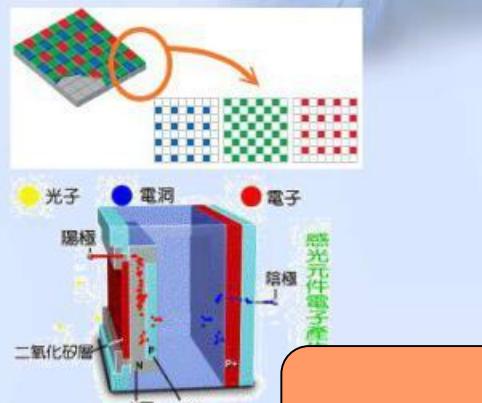
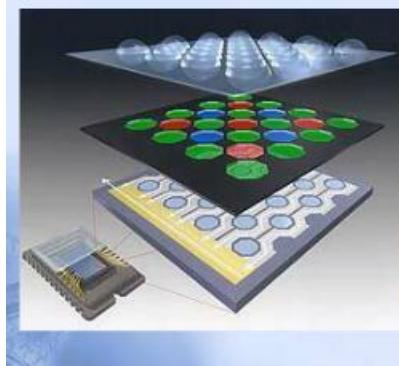


目录:

- ❖ 相机模型
 - 双目相机模型
 - 深度相机模型
- ❖ 特征点法视觉里程计
 - 特征点提取
 - 特征点比对
 - 对极几何
 - 三角化与深度估计
- ❖ 光流法视觉里程计
- ❖ 直接法视觉里程计
- ❖ 编程实践
 - 点云拼接
 - 视觉距离估计: 标定与测量

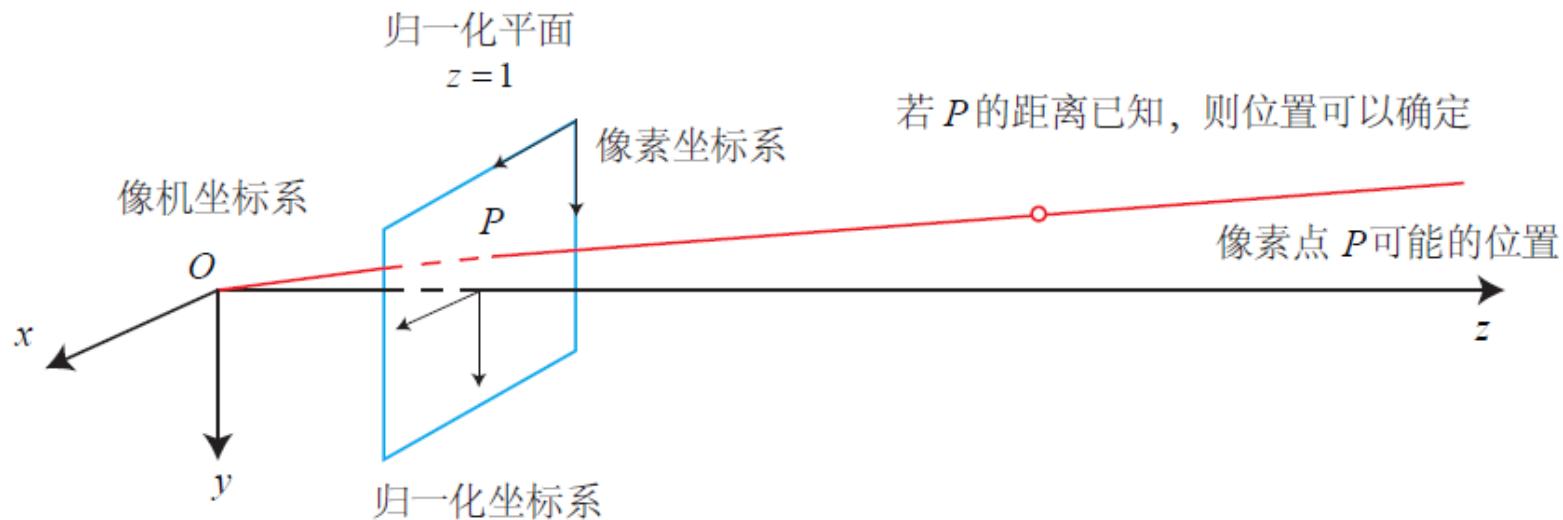
视觉传感器

CCD 的三层结构：上：增光镜片、中：色块网格
下：感应线路



1. 相机模型

- 小孔成像模型
- ❖ 投影顺序: 世界——相机——归一化平面——像素



激光数据的观测模型更加简单

1. 相机模型

- 小孔成像模型小结

1. 首先，世界坐标系下有一个固定的点 P ，世界坐标为 \mathbf{P}_w ；
2. 由于相机在运动，它的运动由 \mathbf{R}, \mathbf{t} 或变换矩阵 $\mathbf{T} \in SE(3)$ 描述。 P 的相机坐标为：
$$\tilde{\mathbf{P}}_c = \mathbf{R}\mathbf{P}_w + \mathbf{t}.$$
3. 这时的 $\tilde{\mathbf{P}}_c$ 仍有 X, Y, Z 三个量，把它们投影到归一化平面 $Z = 1$ 上，得到 P 的归一化相机坐标： $\mathbf{P}_c = [X/Z, Y/Z, 1]^T$ ^①。
4. 最后， P 的归一化坐标经过内参后，对应到它的像素坐标： $\mathbf{P}_{uv} = \mathbf{K}\mathbf{P}_c$ 。

1. 相机模型

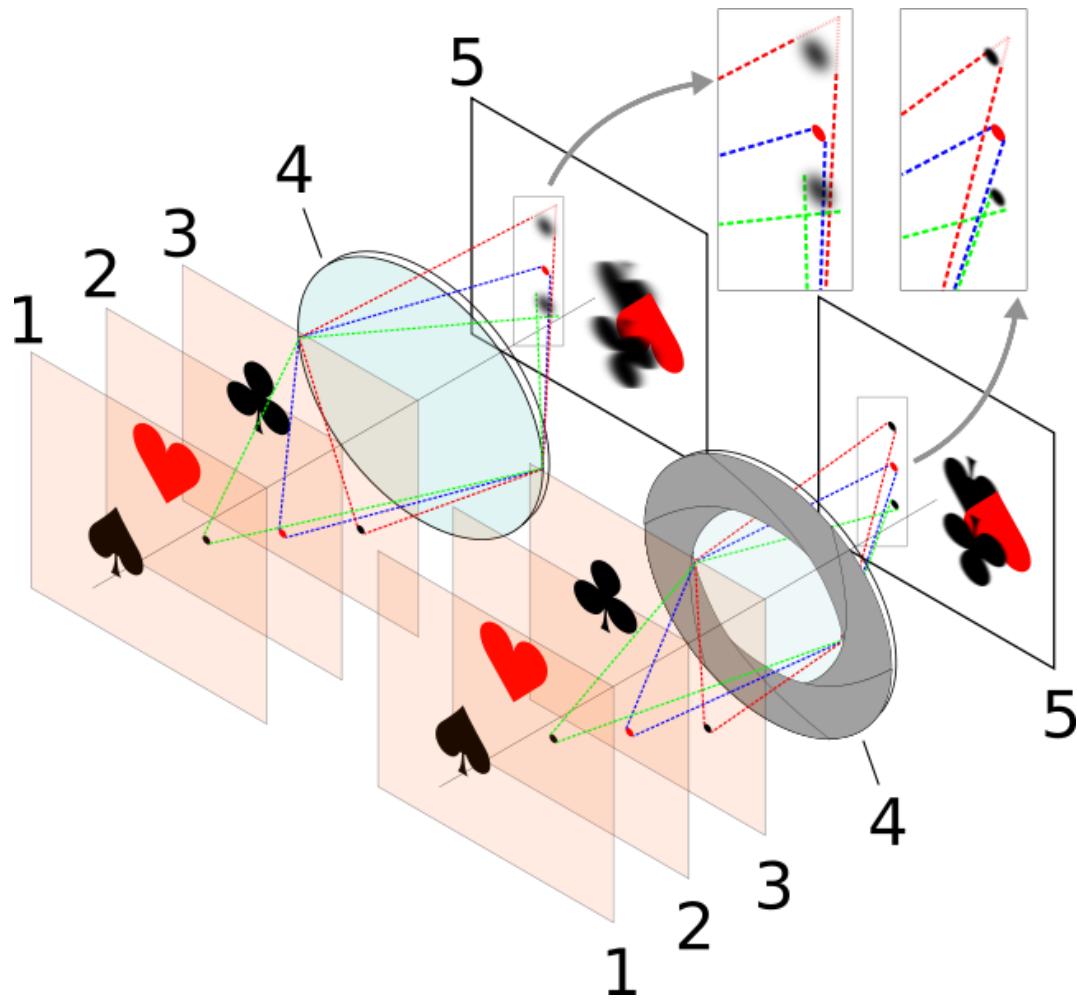
深度获取方法

一般的解决方案是通过观察情景的几张图像以取得更多的信息，它足以有希望地，至少部分地恢复深度。所用的图像必须不一样，以使得合在一起时它们能提供附加信息。它们可以从不同的观察点获得立体或运动算法。另一种选择是建立不同的图像，但不改变观察点，而是改变摄像机的几何特性，诸如焦点位置或镜头的光圈。这是支持景深和散焦深度的基本概念。

- 散焦深度法
- 双目视觉法
- 深度相机法

1. 相机模型-深度获取

感光元件尺寸、焦距与景深的关系



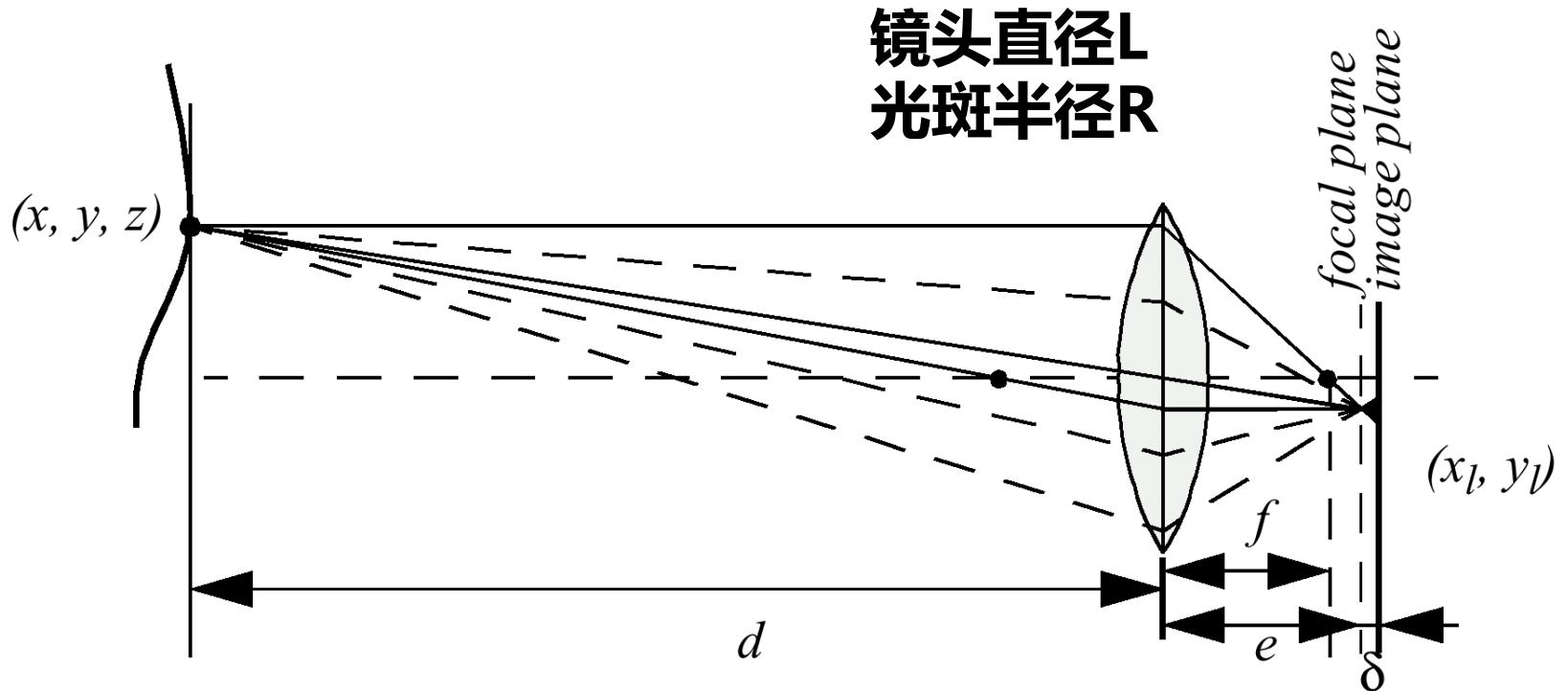
在 $f/32$, 背景很清楚



在 $f/5.6$, 光圈开大
使背景模糊, 让花
与背景分离

1. 相机模型-深度获取

散焦深度原理

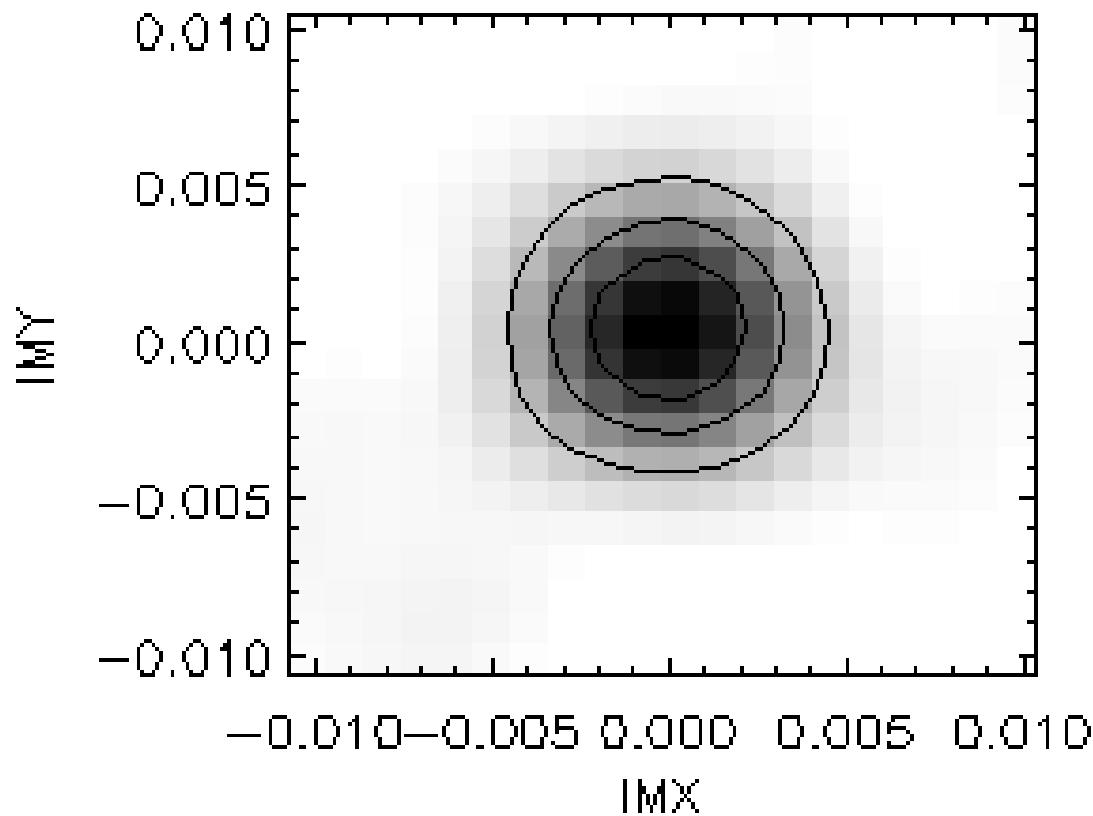


$$\frac{1}{f} = \frac{1}{d} + \frac{1}{e}$$

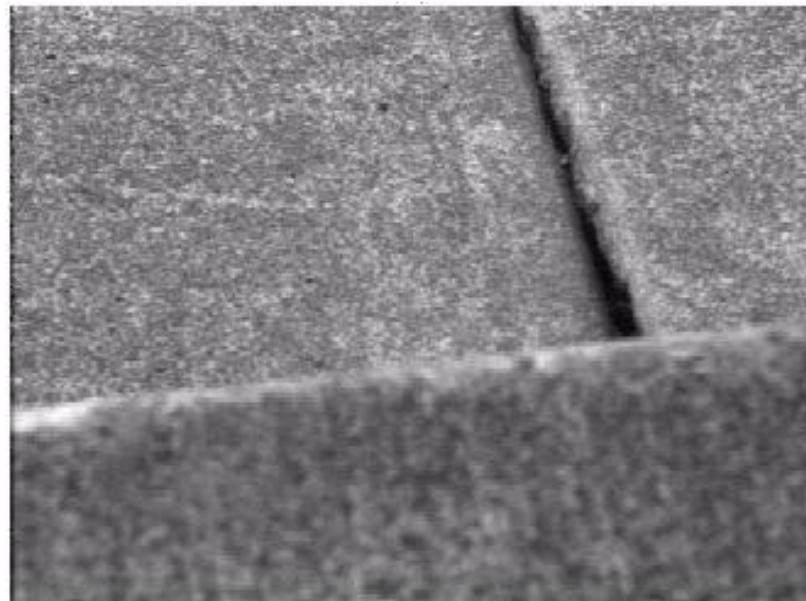
$$R = \frac{L\delta}{2e}$$

1. 相机模型-深度获取

点扩散函数/*Point Spread Function h*

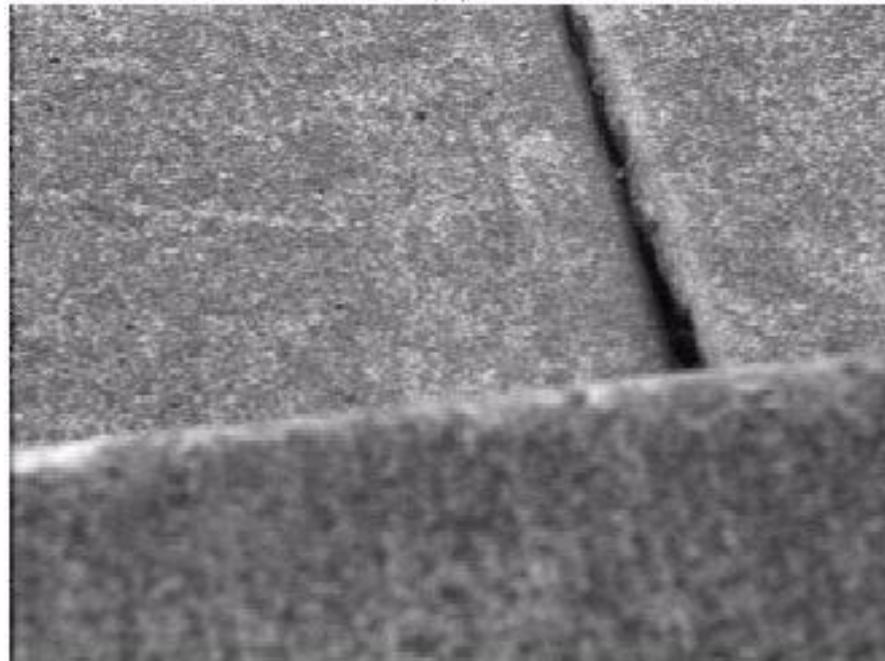


1. 相机模型-深度获取



景深方法是最简单的视觉测距的技术之一。为了确定离物体的距离，传感器简单地将图像平面(通过聚焦)移动，直到将物体锐度最大化。当锐度被最大化时；图像平面的相应位置直接产生距离。某些自动对焦的摄像机和实际上所有视频摄像机都使用该技术。当然，需要有一个方法来测量图像的锐度或图像内的物体。最普通的技术是子图像的强度(I)梯度的近似测量：

1. 相机模型-深度获取



**聚焦深度原理
锐度最大化
纹理锐度计算：
子图梯度**

$$\begin{aligned} \text{sharpness}_1 &= \sum_{x,y} |I(x,y) - I(x-1,y)| \\ \text{sharpness}_2 &= \sum_{x,y} (I(x,y) - I(x-2,y-2))^2 \end{aligned}$$

1. 相机模型-深度获取



Cheshem 机器人使用 3 个单色摄像机作为它在人类环境中避障唯一的测距传感器，静态障碍物如灌木丛，凸障碍如边缘和台阶

1. 相机模型-深度获取

$$\text{锐度}_2 = \sum_{x,y} (I(x,y) - I(x-2,y-2))^2 \quad (4.22)$$

所有 3 个帧同步的摄像机每秒几次同时捕捉同一场景的 3 幅图像。图像各分成 5 列和 3 行,或 15 个子区域。各区的近似锐度用方程(4.22)的变形计算,总共产生 45 个锐度值。注意,方程(4.22)是沿着对角但隔行地计算锐度的。这是由于一个巧妙但是重要的问题引起的。许多摄像机以隔行扫描的模式产生图像,这意味着首先捕捉奇数行,接着捕获偶数行。当在动态环境中使用这种摄像机时,例如,在移动机器人上,则相邻行表示 2 个不同的时间点(差别多至三十分之一秒)的动态场景,由于运动和非光学散焦,结果造成一个人为的模糊(图像混乱)。如果采用只比较偶数编号的行,则可以避免这个隔行扫描的副作用。

1. 相机模型-深度获取

记住,3个图像是用不同的焦点位置,各由一个摄像机拍摄的。根据焦点位置,我们分别称各图像为近、中、远图像。简单地比较3个相应区的锐度值,很快地构成了 3×5 场景的粗糙景深图像。因而,利用近、中、远的值,深度图像只把深度信息的二位分配给各区域。关键的一步是调整所有3个摄像机的焦点位置,使得障碍物前面的平地在深度图的行中产生中等读数。然后,不希望的,或是近的,或是远的读数会分别指明凹或凸的障碍物,使之能在地面的物体以及掉在地面上的物体的周围实现基本的避障。

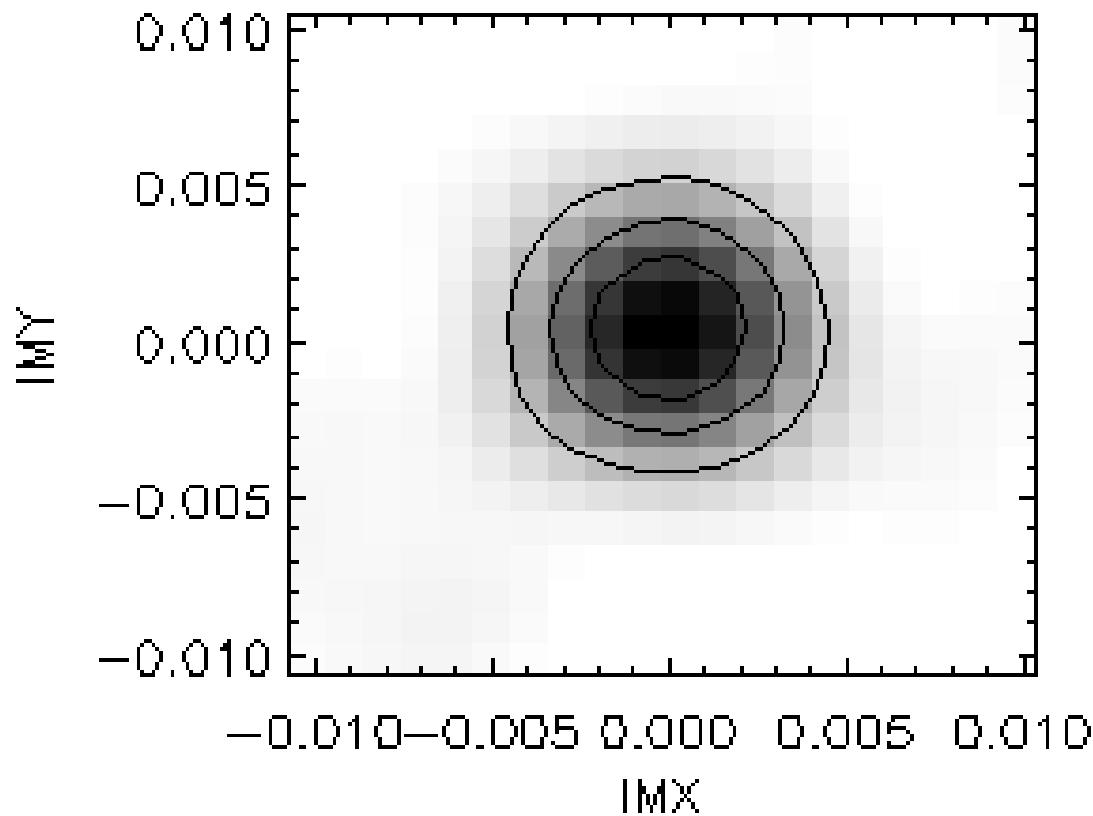
虽然足以应付避障,但上面景深算法只提供不满意的、粗糙的距离信息。另一个选择是用散焦深度方法,它是最期望的基于视觉的技术。

1. 相机模型-深度获取

散焦深度方法把同一场景的 2 个或多个图像作为输入,这些图像是用不同的已知摄像机几何特性拍摄的。给定图像和摄像机几何特征的设置,目标是恢复图像所表示的 3D 场景的深度信息。我们从推导实际场景性质(辐照度和深度)、摄像机几何设置和在图像平面组成的图像 g 之间关系开始。

1. 相机模型-深度获取

点扩散函数/*Point Spread Function h*



1. 相机模型-深度获取

点扩散函数/Point Spread Function h

$$h(x_g, y_g, x_f, y_f, R_{x,y}) = \begin{cases} \frac{1}{\pi R^2} & \text{if } ((x_g - x_f)^2 + (y_g - y_f)^2) \leq R^2 \\ 0 & \text{if } ((x_g - x_f)^2 + (y_g - y_f)^2) > R^2 \end{cases}$$

成像模型

$$g(x_g, y_g) = \sum_{x, y} h(x_g, y_g, x, y, R_{x,y}) f(x, y)$$

求解 R 获得 单目相机场景中地图的深度信息

1. 相机模型-深度获取

点扩展函数 $h(x_g, y_g, x_f, y_f, R_{x,y})$ 定义为从场景 P 点(相应于 (x_f, y_f))到聚焦的图像 f 的辐照总量, 在观察到的散焦图像 g 中, f 提供了点 (x_g, y_g) 。注意, 点扩展函数不仅取决于源 (x_f, y_f) 和目标 (x_g, y_g) , 也取决于 R 。

$$g(x_g, y_g) = \sum h(x_g, y_g, x, y, R_{x,y}) f(x, y)$$

这个方程, 通过 R 建立了场景点深度对观察到的图像 g 的关系, 求解 R 就向我们提供地图深度。但是该函数有另外的未知聚焦图像 f 。所以, 假如我们不知道全聚焦图像如何看, 单独一个图像不足以解决深度恢复问题。

给定用不同摄像机几何特性拍摄的同一场景的两个图像, 在理论上有可能求解 g 和 R , 因为 f 保持恒定。存在许多可以准确快速完成这种解答的算法。经典的方法是所谓逆滤波

1. 相机模型-深度获取

基于点扩散函数 h , 用逆滤波法求解景深

Pentland, A.P., “A New Sense for Depth of Field.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9:523–531, 1987.

基于统计技术, 把散焦问题特征化为线性方程组

Ens, J., Lawrence, P., “An Investigation of Methods for Determining Depth from Focus.” *IEEE Transactions. on Pattern Analysis and Machine Intelligence*, 15: 97–108, 1993.

使用三个单色摄像机实现深度测量的机器人Cheshem

Nourbakhsh, I.R., Andre. D., Tomasi, C., Genesereth, M.R., “Mobile Robot Obstacle Avoidance via Depth from Focus,” *Robotics and Autonomous Systems*, 22:151–158, 1997.

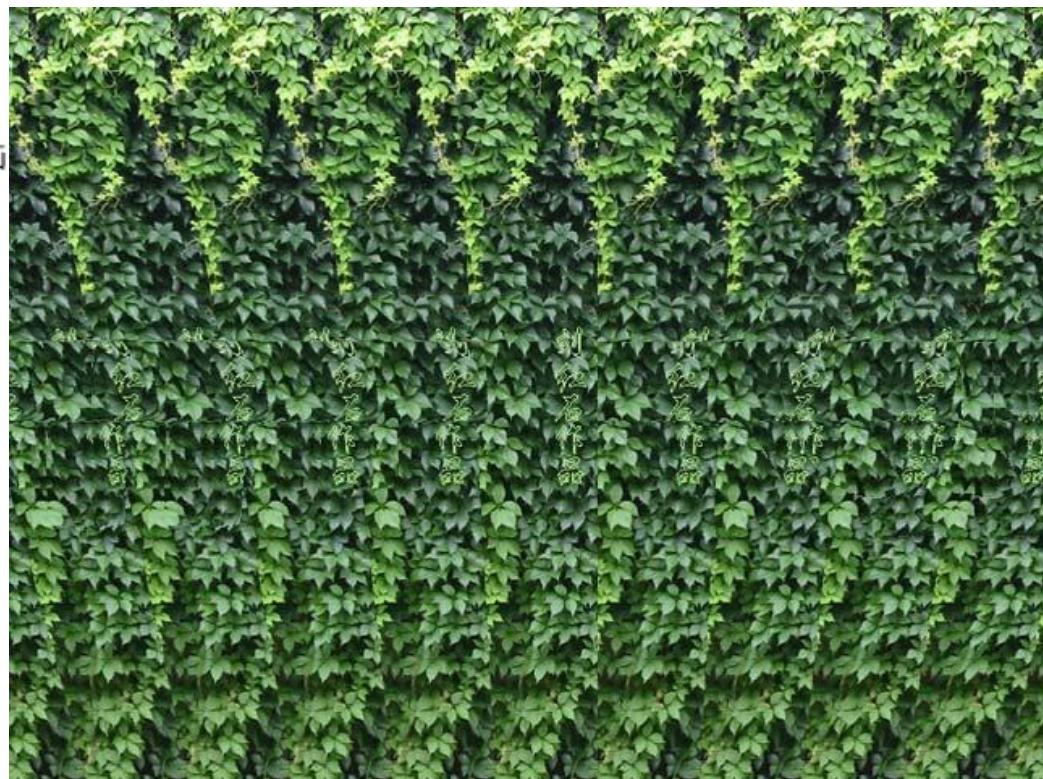
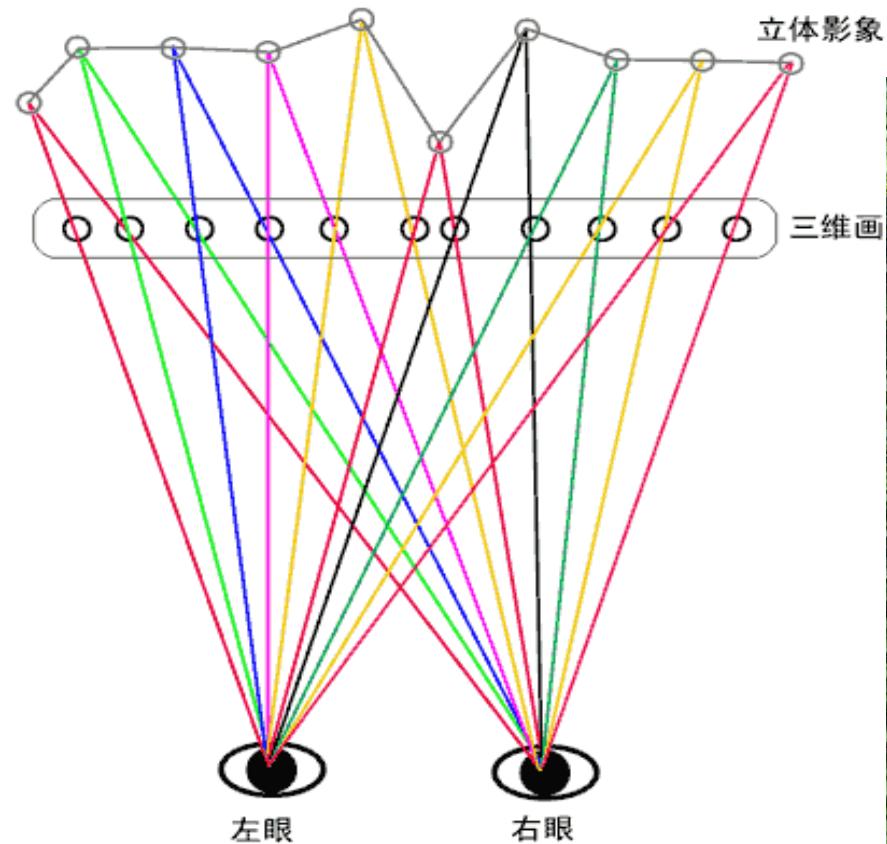
1. 相机模型

深度获取方法

- 散焦深度法
- 双目视觉法
- 深度相机法

1. 相机模型-双目法深度获取

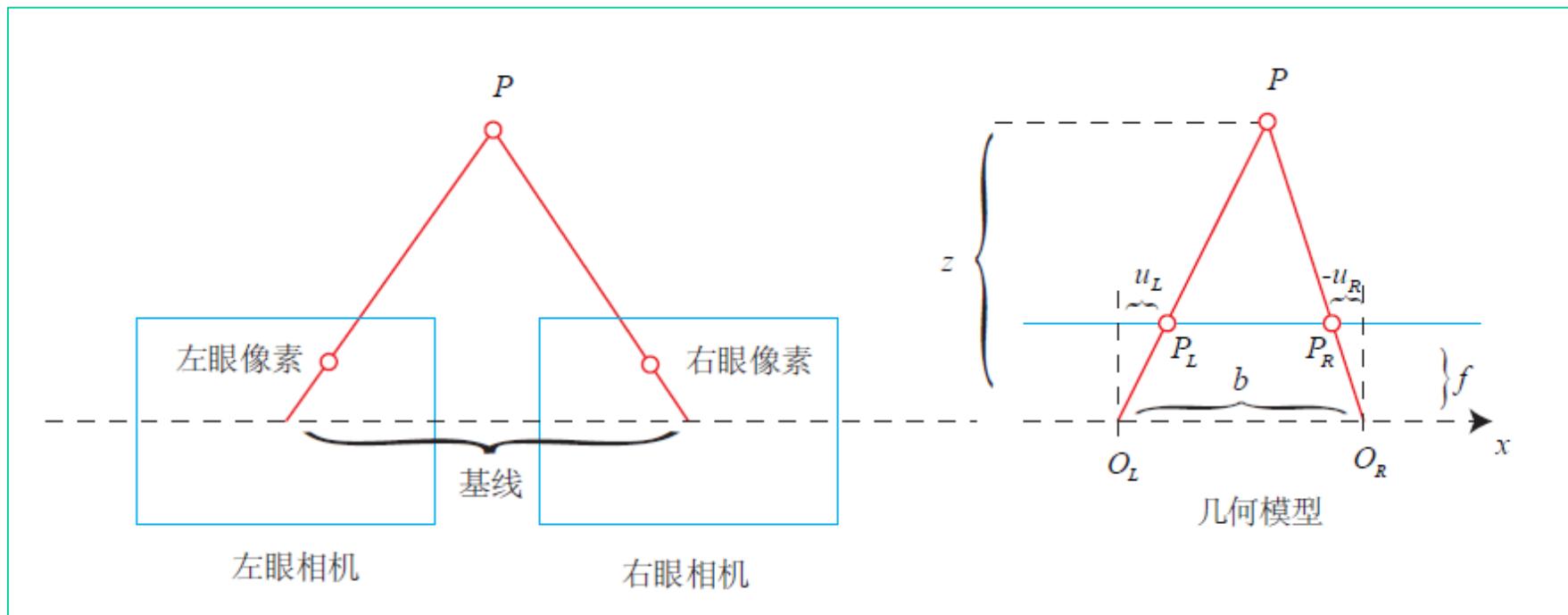
三维画的原理



1. 相机模型

❖ 双目模型

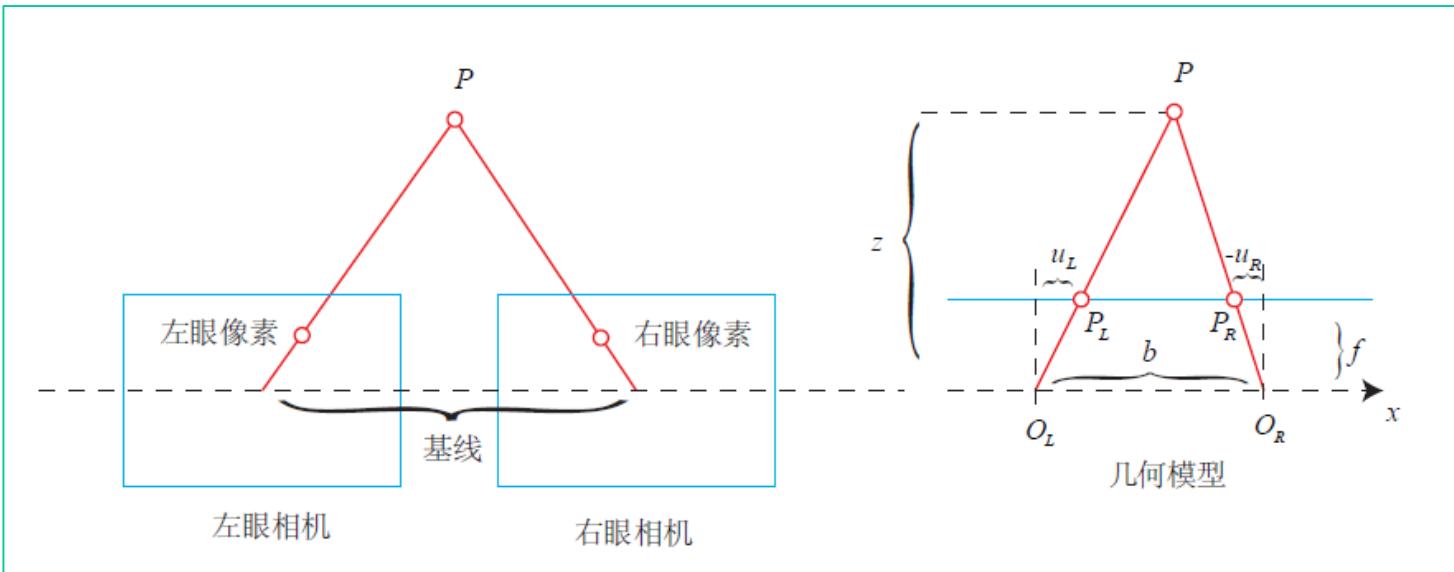
- 左右相机中心距离称为基线



d 称为视差 (disparity)，描述同一个点在左右目上成像的距离， d 最小为 1 个像素

1. 相机模型

❖ 双目模型



■ 左右像素的几何关系：

$$\frac{z - f}{z} = \frac{b - u_L + u_R}{b}.$$

$$\text{■ 整理得 } z = \frac{fb}{d}, \quad d = u_L - u_R.$$

立体视觉/双目模型

理想的立体视觉成像几何模型/

Idealized camera geometry for stereo vision

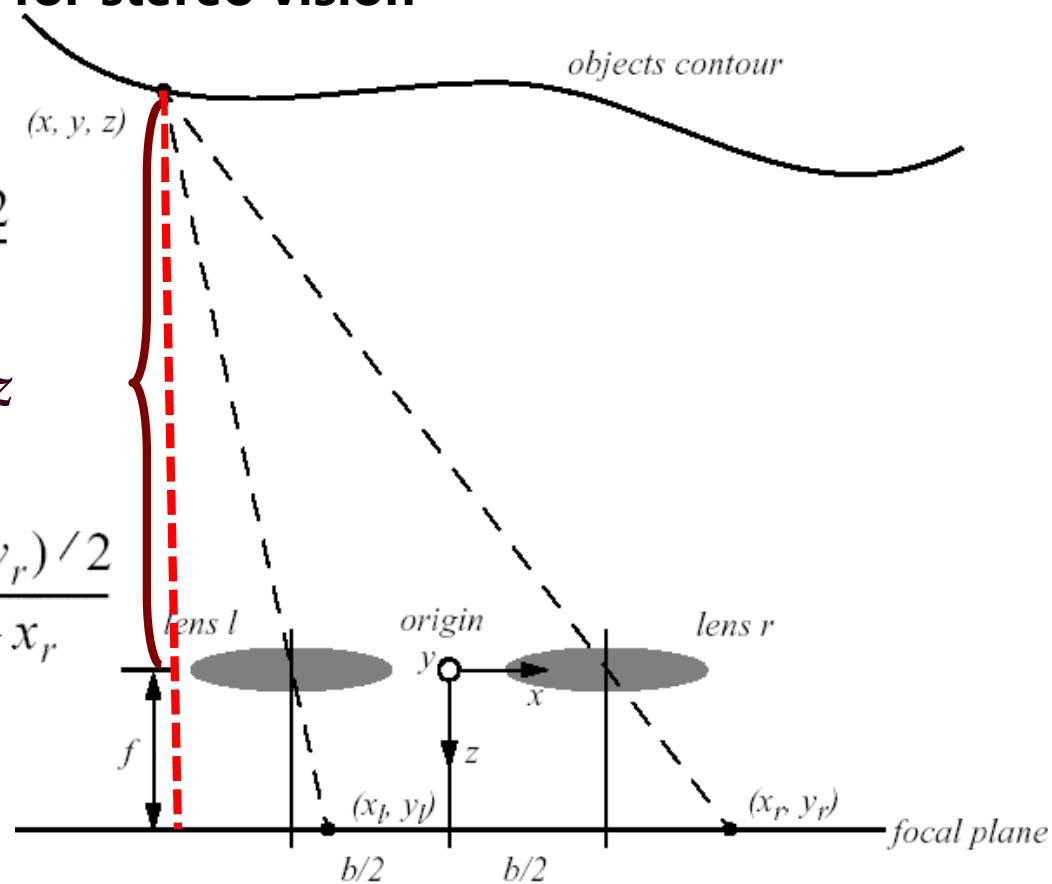
两幅像的差异 → 计算景深
如图所示

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

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

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

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



1. 相机模型

双目模型 三角化像差图像分析 (Binocular Disparity map)

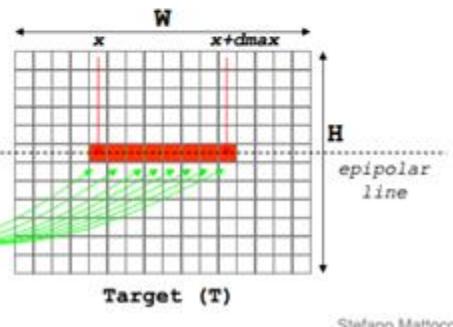
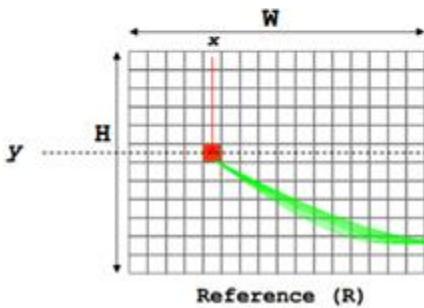
The simplest (naive and unused) local approach:



Reference (R)

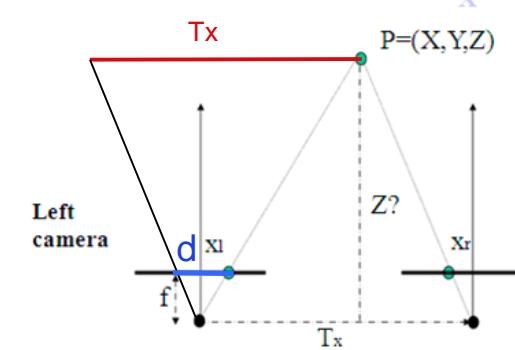
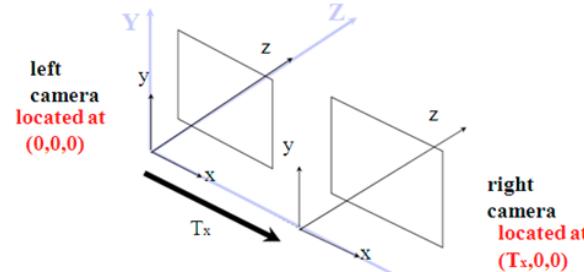


Target (T)



Triangulation

获得(u_l, u_r, v)之后，计算特征点在相机坐标系下的坐标



$$Z = T^* f / d$$

Left
camera

Right
camera

T 为相机间距，单位m
 x_l, x_r 分别为principal point
在图像的横坐标
 d 为视差，单位均为pix

基于线性搜索的视差计算

立体视觉/Stereo Vision

1. 距离反比于差异

越近的目标测量的越准

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

2. 差异正比于(左右相机间)基线宽度

给定差异误差，增加基线宽度可以提高深度估计的精度。
但增加基线宽度，会出现目标在某个相机中消失的情形。

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

3. 出现在两台相机中的同一点构成共轭对.

共轭对位于核线

立体视觉/Stereo Vision - the general case

同一点 P 在左相机成像的测量结果可以表示成:

$$r'_r = R \cdot r'_l + r_0 \quad \begin{bmatrix} x'_r \\ y'_r \\ z'_r \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x'_l \\ y'_l \\ z'_l \end{bmatrix} + \begin{bmatrix} r_{01} \\ r_{02} \\ r_{03} \end{bmatrix}$$

其中

R 是 摄像机坐标系左到右的 3×3 旋转矩阵

r_0 = 摄像机之间的偏置平移向量

观察上述方程知道:

若知道 R, r_l 和 r_0 则可以确定 r_r .

特别地, 当 相机完全对准时 $R=I$

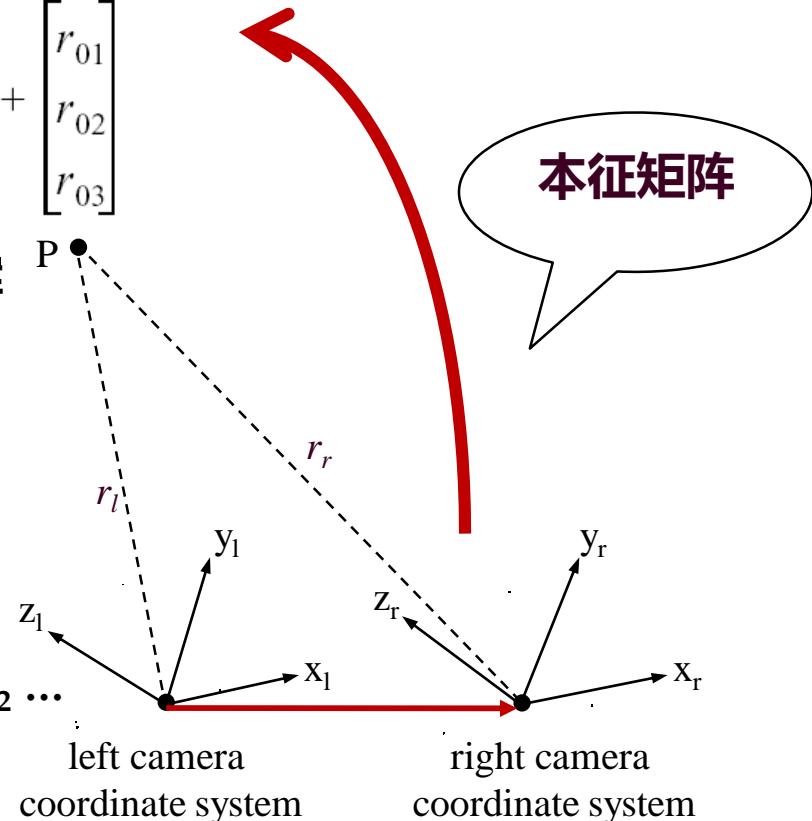
给定相应的 x_l, y_l, z_l, x_r, y_r 和 z_r 的值

我们能标定坐标变换系统, 即确定出 $r_{11}, r_{12} \dots$

我们需要确定 12未知量, 需要 12方程:

需要 4 共轭点.

需要说明的是: 影像是存在光学岐变的



1. 相机模型

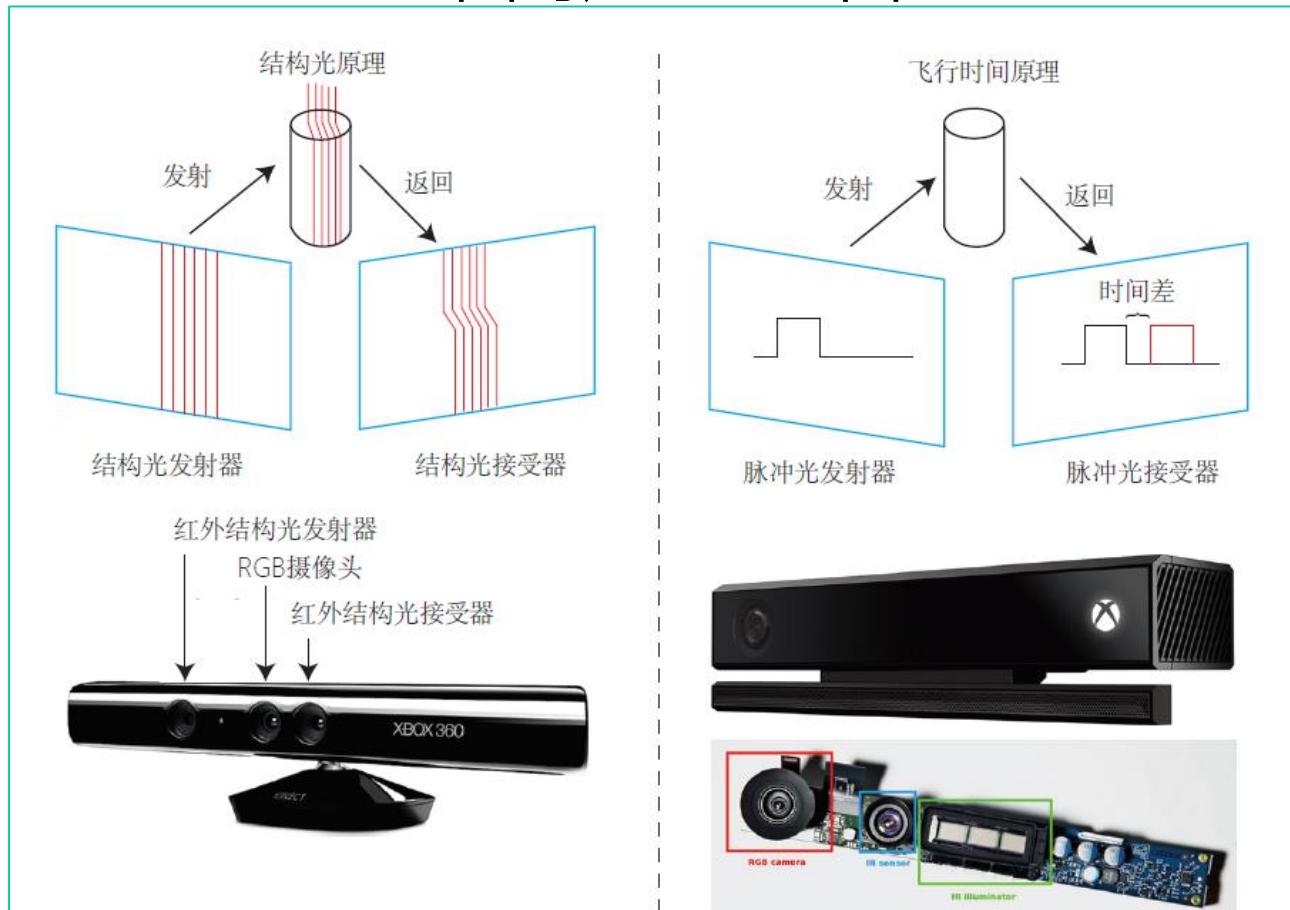
深度获取方法

- 散焦深度法
- 双目视觉法
- 深度相机法

1. 相机模型

❖ RGB-D相机：物理手段测量深度

- ToF或结构光两种主要原理
- 通常能得到与RGB图对应的深度图



1. 相机模型

- ❖ RGB-D相机：物理手段测量深度



Microsoft和PrimeSense合作的Kinect1

1. 相机模型

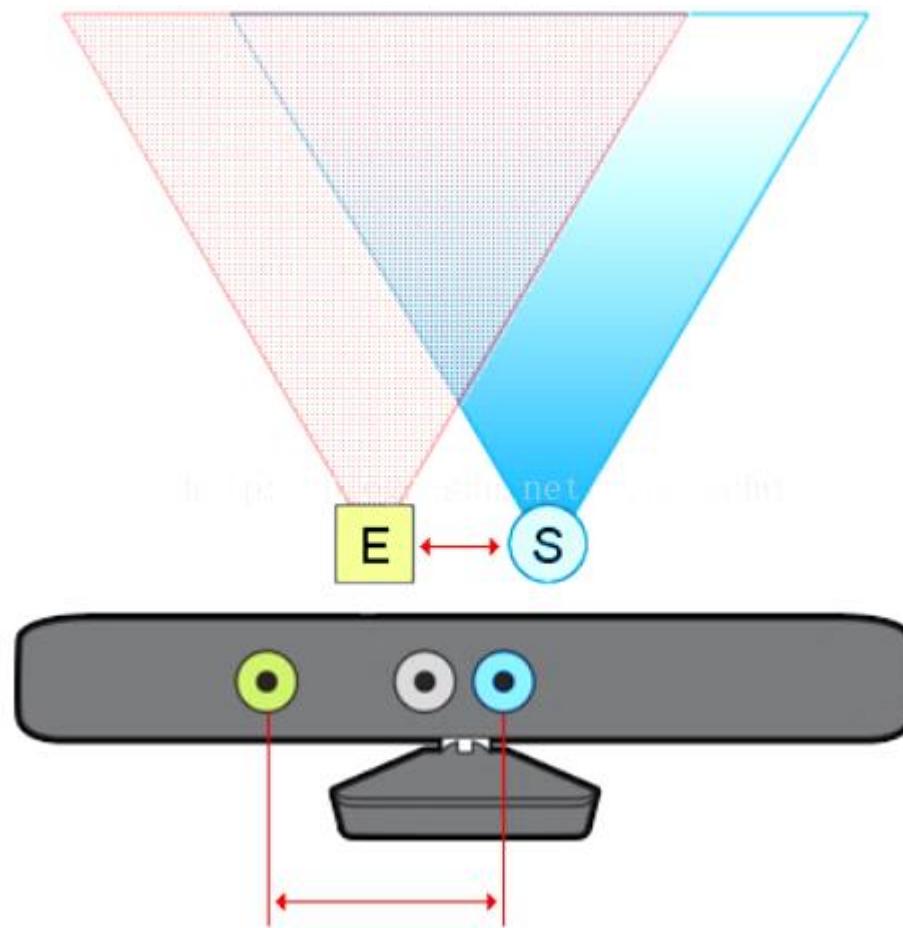
- ❖ RGB-D相机：物理手段测量深度



Microsoft和PrimeSense合作的Kinect1

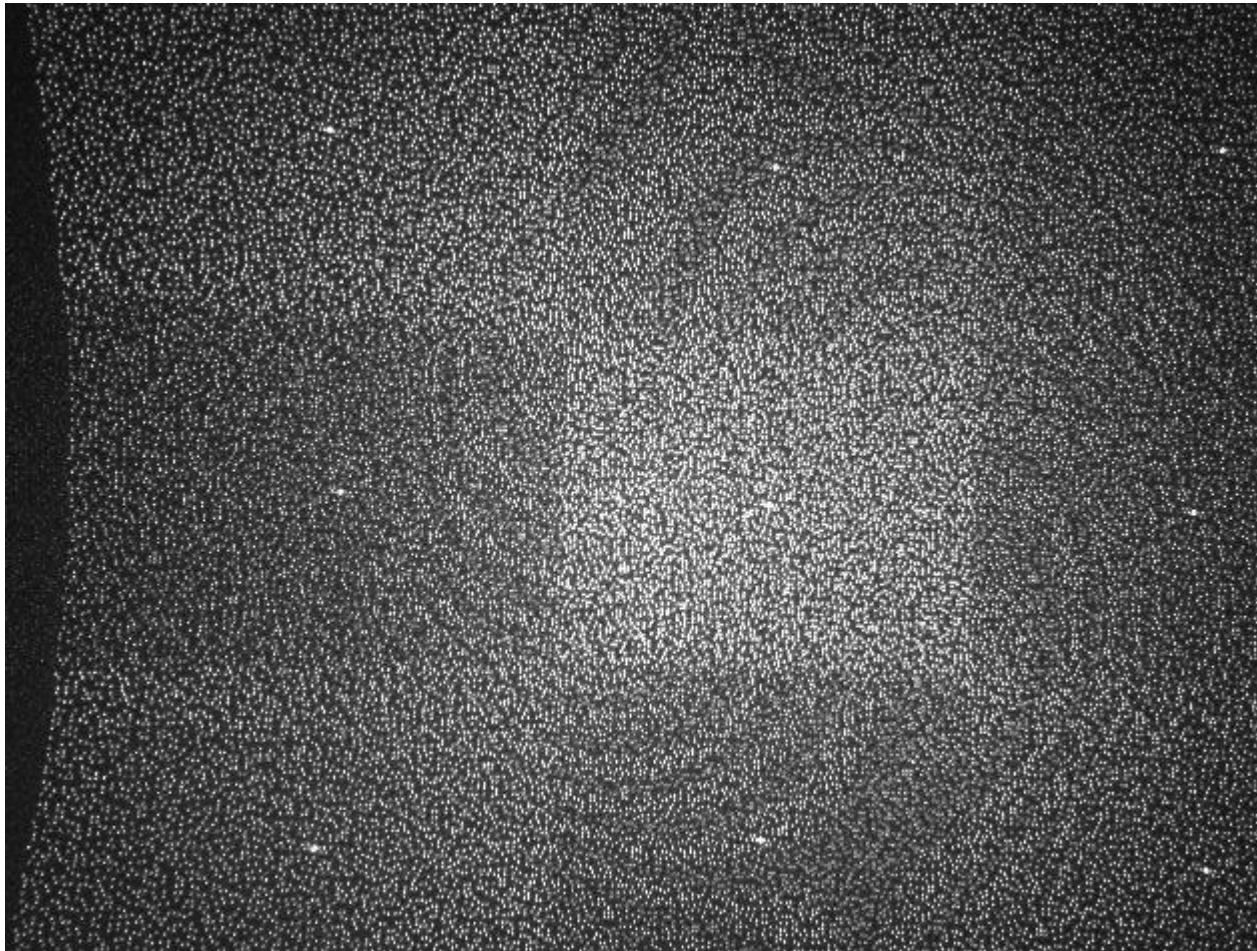
1. 相机模型

- ❖ RGB-D相机：物理手段测量深度



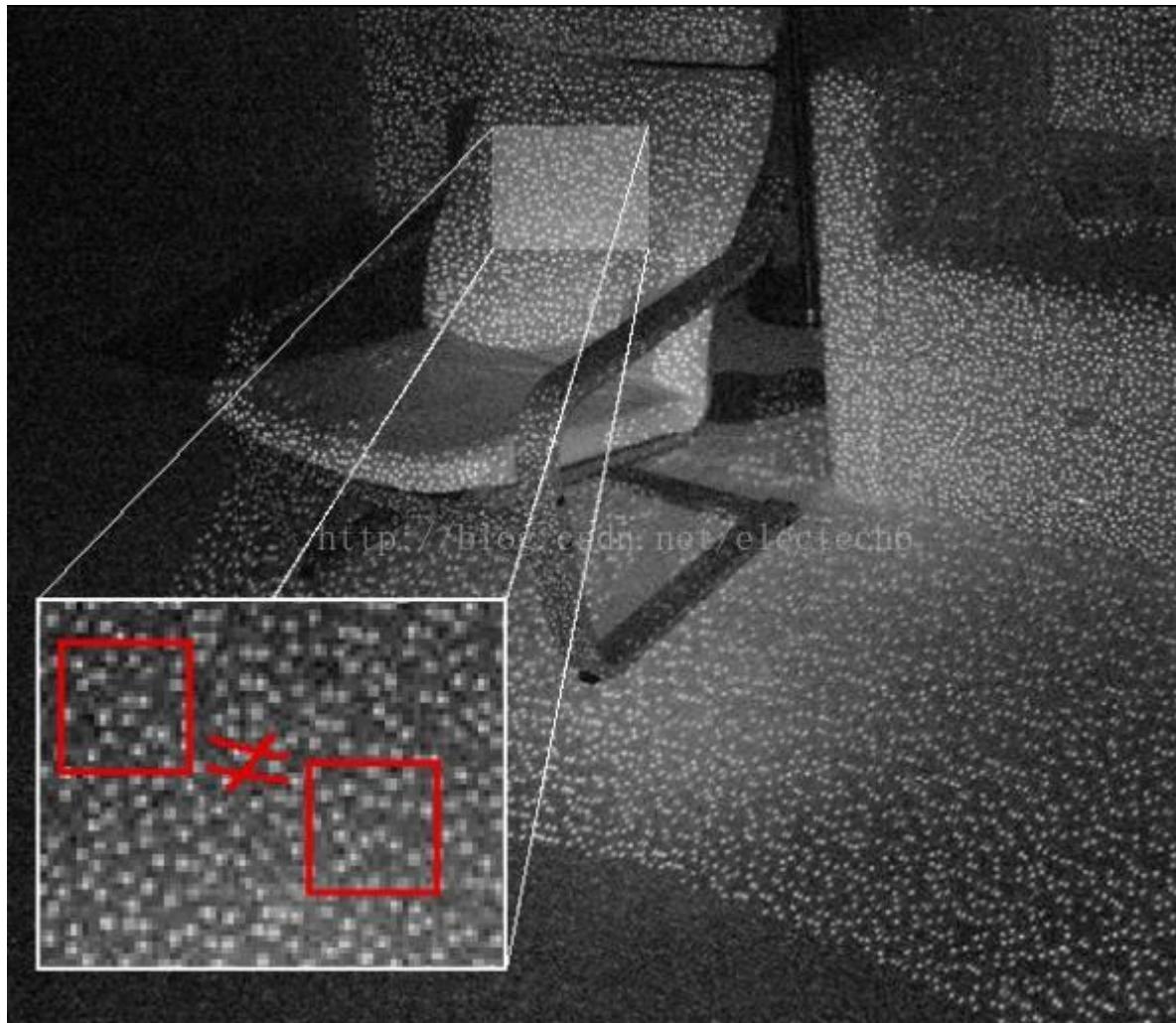
1. 相机模型

- ❖ RGB-D相机：物理手段测量深度—散斑分布



1. 相机模型

- ❖ RGB-D相机：物理手段测量深度—散斑反射



1. 相机模型

❖ RGB-D相机：物理手段测量深度

Intel RealSense R200 (480×360 px,
 $70^\circ \times 46^\circ$, 0.5 – 4 m)



Intel RealSense SR300 (640×480 px,
 $72^\circ \times 55^\circ$, 0.2 – 2 m)



HP 3D Scanner Pro S3 (6 – 50 cm, up to 0.05 mm accuracy, > 3000 €) Microsoft Kinect v1 (640×480 px, $57^\circ \times 43^\circ$, 0.4 – 4 m / 8 m)



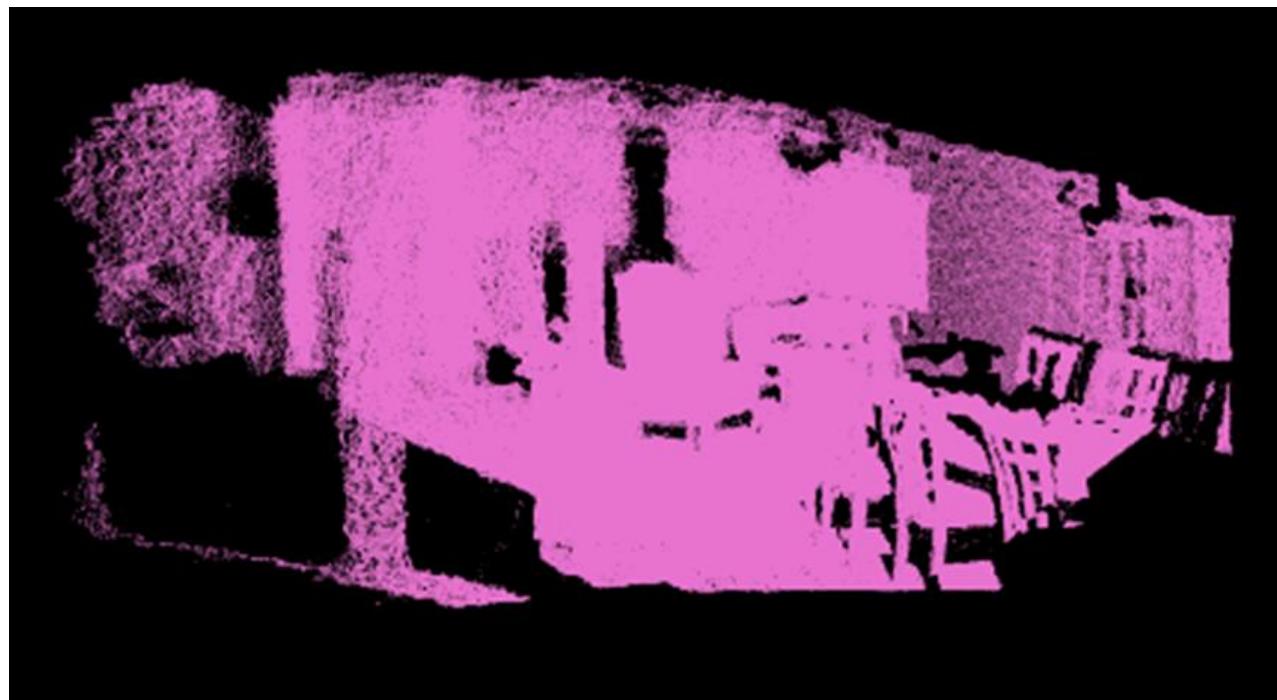
几种结构光深度相机的参数

Lab2

❖ 视觉3D点云图构建与深度测量

实验内容：

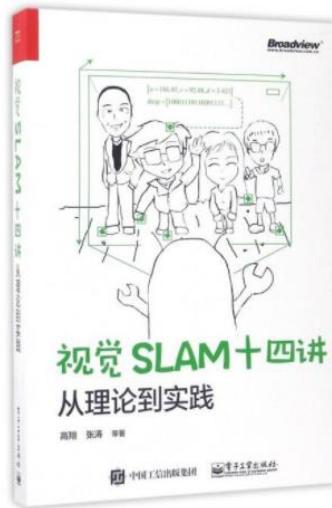
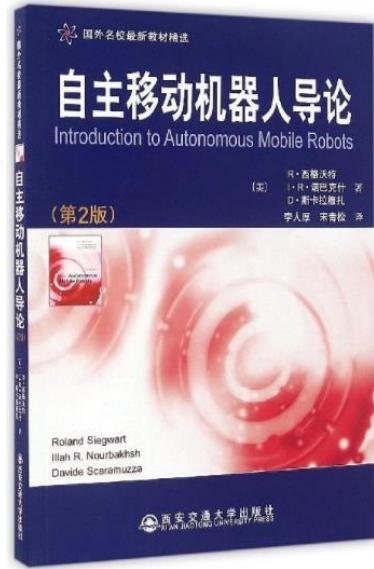
1. 彩色图、深度图的读取与参数提取
2. 相机参数加载与使用
3. 点云地图拼接
4. 双目视觉系统标定与深度测量(选做)



第四章 立体视觉系统

目标:

- ❖ 掌握常见立体视觉系统的基本原理，点云图拼接实践
- ❖ 了解同时定位与地图构建SLAM原理



目录:

- ❖ 相机模型
 - 双目相机模型
 - 深度相机模型
- ❖ 特征点法视觉里程计
 - 特征点提取
 - 特征点比对
 - 对极几何
 - 三角化与深度估计
- ❖ 光流法视觉里程计
- ❖ 直接法视觉里程计
- ❖ 编程实践
 - 点云拼接
 - 视觉距离估计: 标定与测量

视觉里程计

里程计(Odometry)是一种利用从移动传感器获得的数据来估计物体位置随时间的变化而改变的方法。该方法被用在许多机器人系统(轮式或者腿式)上面，来估计，而不是确定这些机器人相对于初始位置移动的距离。

1. 常用的里程传感装置是**旋转编码器**，借助机器人在其轮子或腿关节处配备的**旋转编码器**等设备，可以测量出轮子旋转的圈数，如果知道了轮子的周长，便可以计算出机器人移动的距离。但轮子的打滑就会导致产生机器人移动的距离与轮子的旋转圈数不一致的问题；
2. **视觉里程计**是一种利用连续的图像序列来估计机器人移动距离的方法，按照原理不同，可分为**特征点法**，**光流场法**及其改进的**直接法**。

2. 特征点法视觉里程计

像差图像 (Binocular Disparity map)

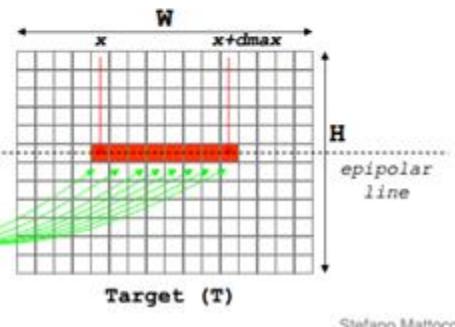
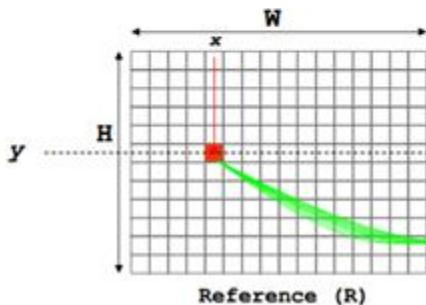
The simplest (naive and unused) local approach:



Reference (R)



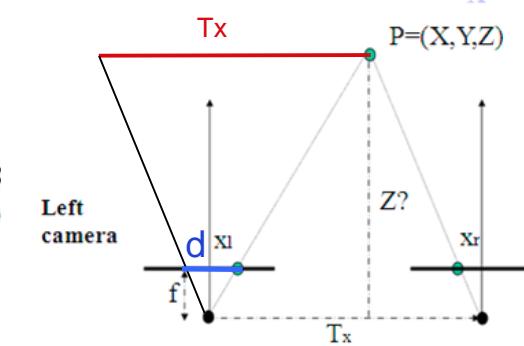
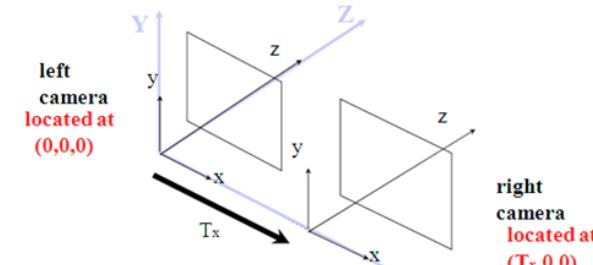
Target (T)



基于线性搜索的视差计算

Triangulation

获得(u_l, u_r, v)之后，计算特征点在相机坐标系下的坐标

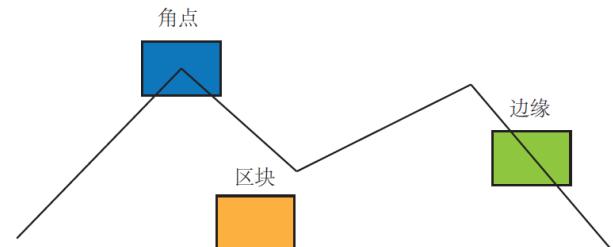


$$Z = T^* f / d$$

T 为相机间距，单位m
 X_l, X_r 分别为principal point
在图像的横坐标
 d 为视差，单位均为pix

2. 特征点法视觉里程计

- 特征点：图像当中具有代表性的部分
 - 可重复性：在不同图像中能重现
 - 可区别性：不同的点有不同的表达
 - 高效：特征点数量应远小于像素的数量
 - 本地：特征仅与一小片图像区域相关
- 特征点的信息
 - 位置、大小、方向、评分等——关键点(*Key point*)
 - 特征点周围的图像信息——描述子(*Descriptor*)
- 例子：SIFT/SURF/ORB 见OpenCV features2d模块



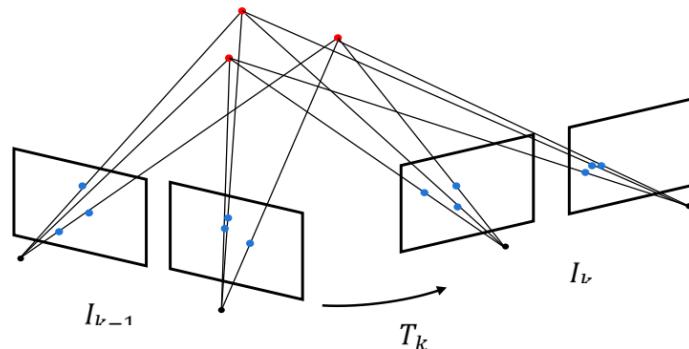
特征描述应该在光照、
视角发生少量变化时
仍能保持一致

2. 特征点法视觉里程计

图像特征点

要求detector找到的特征点具有重复性和稳定性

- different viewpoint
光照、尺度、旋转…



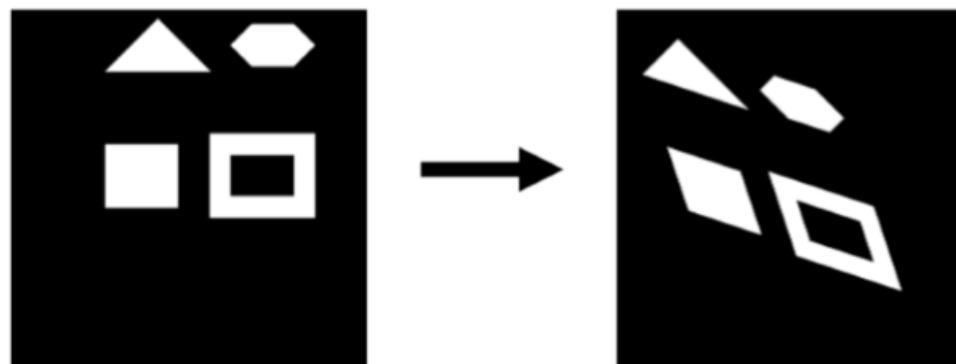
	Corner Detector	Blob Detector	Rotation Invariant	Scale Invariant	Affine Invariant	Repeatability	Localization Accuracy	Robustness	Efficiency
Haris	x		x			+++	+++	++	++
Shi-Tomasi	x		x			+++	+++	++	++
FAST	x		x	x		++	++	++	++++
SIFT		x	x	x	x	+++	++	+++	+
SURF		x	x	x	x	+++	++	++	++
CENSURE		x	x	x	x	+++	++	+++	+++

Figure 1. Comparison of feature detectors: properties and performance.

2. 特征点法视觉里程计

affine

2D的仿射变换具有六自由度，这种变化是保平行的

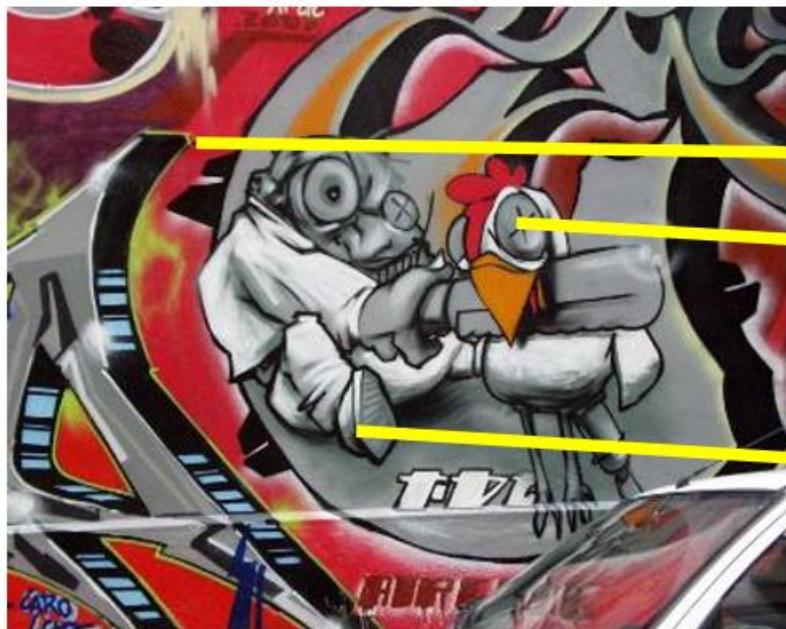


$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = R(\theta)R(-\phi)SR(\phi) \quad S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

2. 特征点法视觉里程计

affine



2. 特征点法视觉里程计

Harris Detector通过数学计算在图像上发现角度特征的一种算法

将图像窗口平移 $[u, v]$ 产生灰度变化 $E(u, v)$

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

由： $I(x + u, y + v) = I(x, y) + I_x u + I_y v + O(u^2, v^2)$ ， 得到：

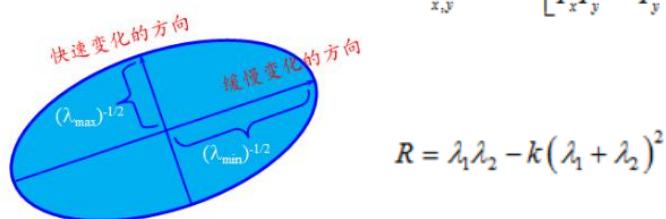
$$E(u, v) = [u, v] \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

对于局部微小的移动量 $[u, v]$ ，近似表达为：

$$E(u, v) \cong [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

其中 M 是 2×2 矩阵，可由图像的导数求得：

$$M = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



详见

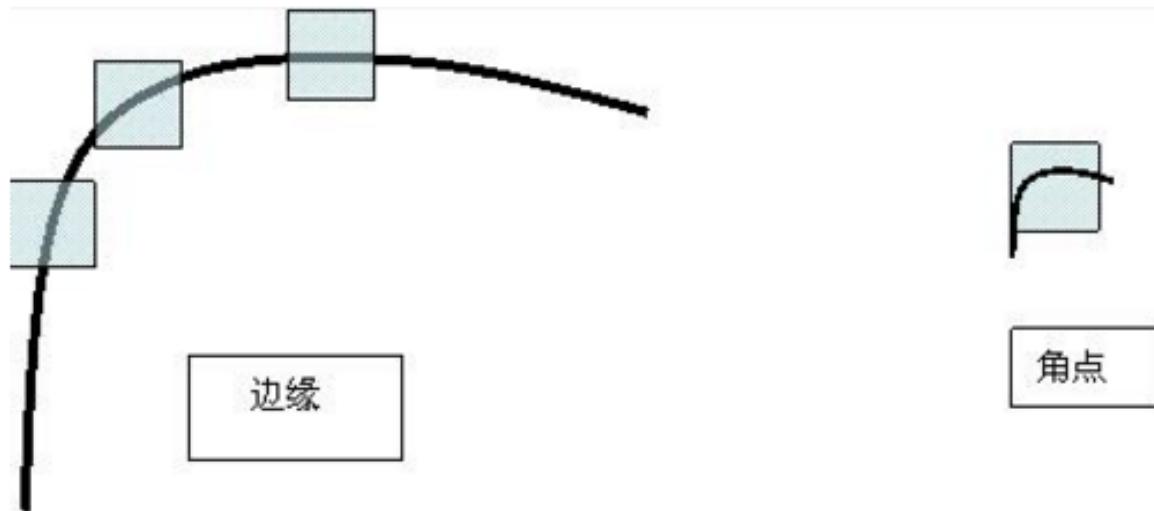
CSE486_lecture06_Harris corner of Penn State University

Harris and M. Stephens_A Combined Corner and Edge Detector

2. 特征点法视觉里程计

Harris Detector通过数学计算在图像上发现角度特征的一种算法

Harris算子对灰度的平移是不变的,因为只有差分,对旋转也有不变性,但是对尺度很敏感,在一个尺度下是角点,在另一个尺度下可能就不是了.



2. 特征点法视觉里程计

- FAST(Features from Accelerated Segment Test)特征

- 快速检测角点
- 连续N个点的灰度有明显差异

$$S_{\text{bright}} = \{x | I_{p \rightarrow x} \geq I_p + t\}$$

$$S_{\text{dark}} = \{x | I_{p \rightarrow x} \leq I_p - t\}$$

如果一个像素与邻域的像素差别较大
(过亮或过暗), 那么它可能是个角点

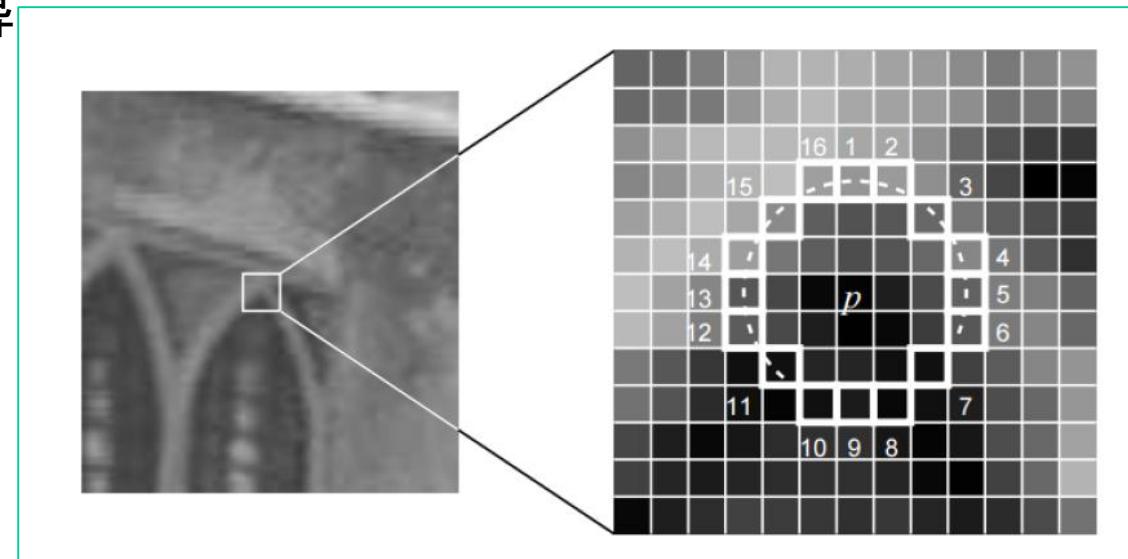


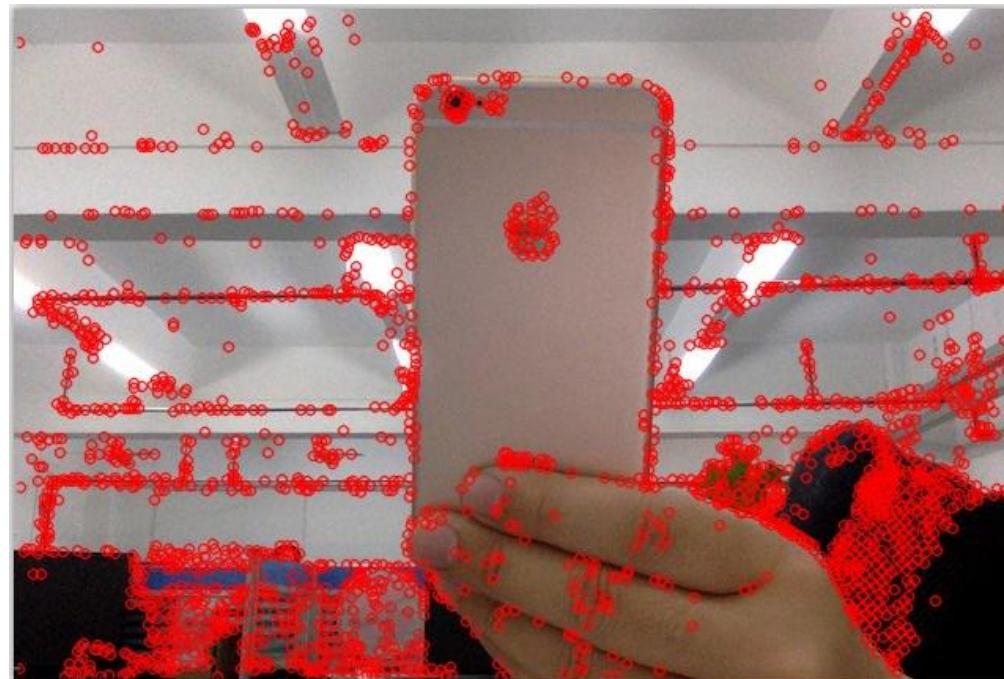
Figure 1. 12 point segment test corner detection in an image patch. The highlighted squares are the pixels used in the corner detection. The pixel at p is the centre of a candidate corner. The arc is indicated by the dashed line passes through 12 contiguous pixels which are brighter than p by more than the threshold.

以 p 为中心, 以3为半径的圆上16个像素点, 有N个点符合判据, 则 p 为特征点, N 取12,11或9

2. 特征点法视觉里程计

- FAST(Features from Accelerated Segment Test)特征
 - 实时性高
 - 原始角点比较集中, 进行非极大值抑制, 避免角点过于集中

$$V = \max \left(\sum_{x \in S_{\text{bright}}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{\text{dark}}} |I_p - I_{p \rightarrow x}| - t \right)$$



图片引用<https://www.cnblogs.com/denqiaojun/p/5302778.html>

2. 特征点法视觉里程计

- FAST

- 连续N个点的灰度有明显差异

- 约束较为单一

- 方向问题

- FAST角点不具有方向信息，固定选取半径为3的圆

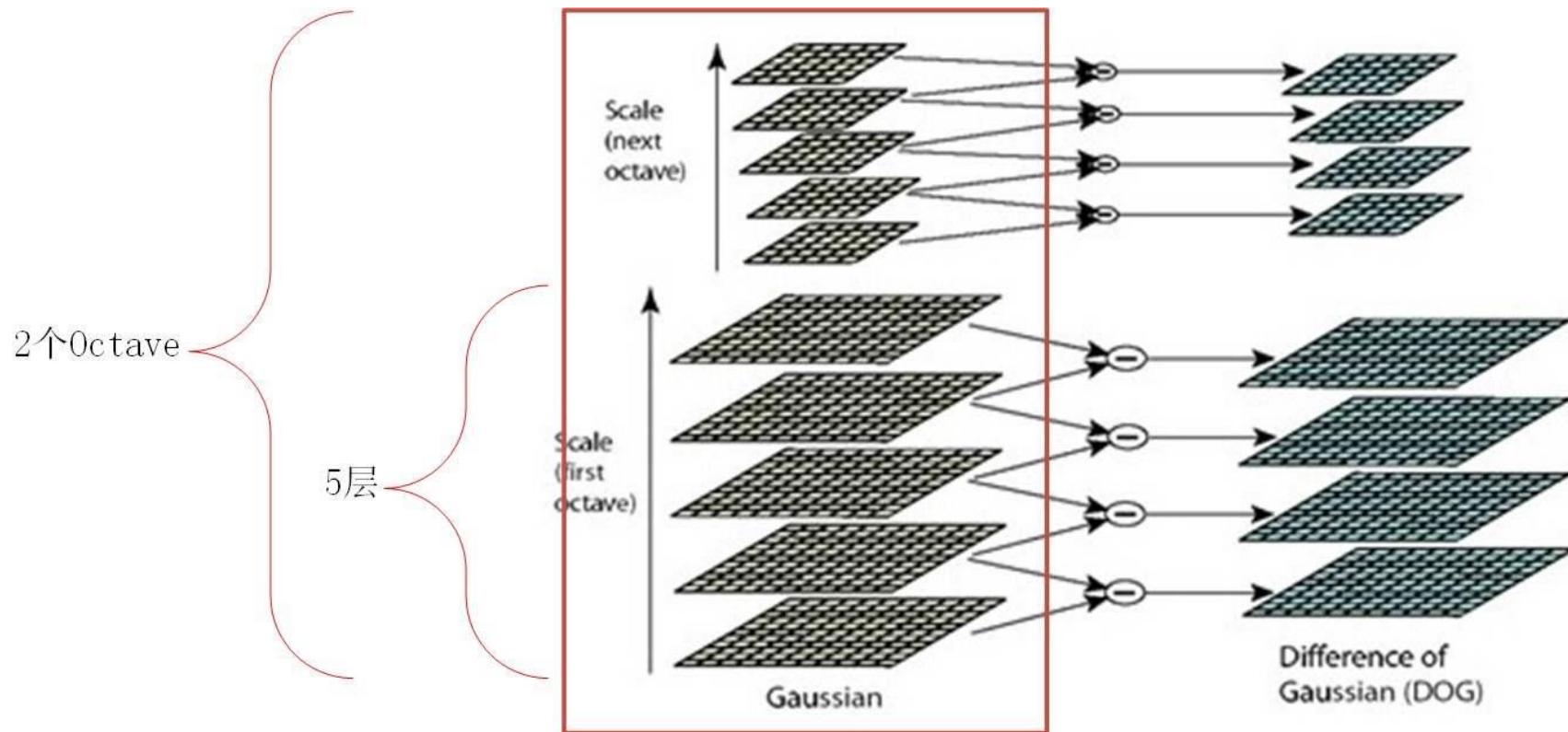
- 尺度问题

- 远看是角点的地方，近看就不是了

2. 特征点法视觉里程计

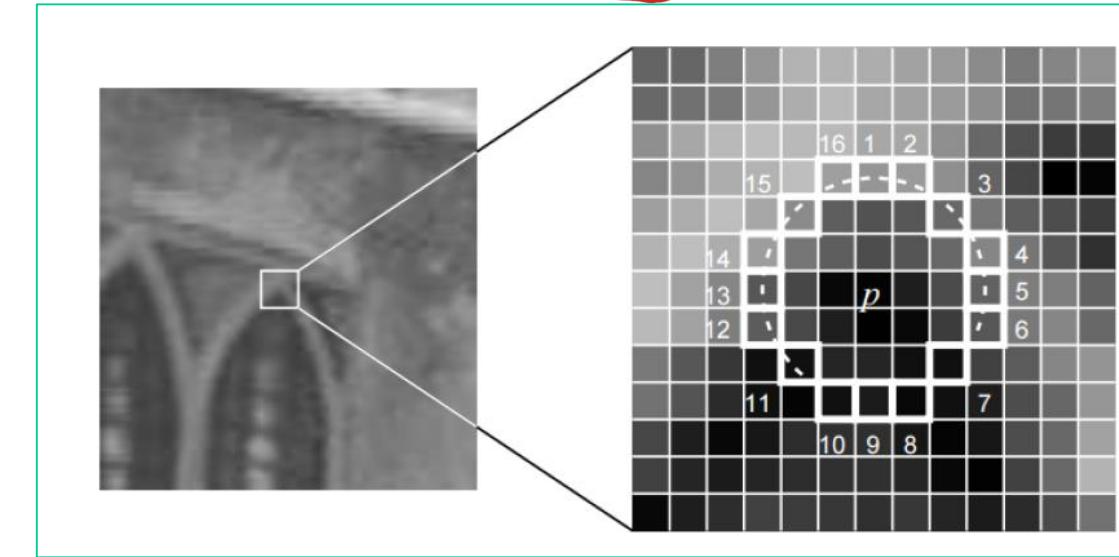
- 改进: ORB特征 Oriented FAST

- 尺度上引入图像金字塔
- 在FAST基础上计算旋转
- 添加描述:BRIEF



2. 特征点法视觉里程计

- 改进: ORB特征
 - 关键点: Oriented FAST
 - 尺度上引入图像金字塔
 - 在FAST基础上计算旋转
 - 添加描述: BRIEF



1. 在一个小的图像块 B 中, 定义图像块的矩为:

$$m_{pq} = \sum_{x,y \in B} x^p y^q I(x,y), \quad p, q = \{0, 1\}.$$

2. 通过矩可以找到图像块的质心:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right).$$

3. 连接图像块的几何中心 O 与质心 C , 得到一个方向向量 \overrightarrow{OC} , 于是特征点的方向可以定义为:

$$\theta = \arctan(m_{01}/m_{10}).$$

2. 特征点法视觉里程计

- BRIEF

- BRIEF是一种二进制描述，其描述向量由0和1组成，0和1表示关键点附近两个像素的大小关系
- BRIEF-128:按某种概率随机的方式，在特征点附近取128次像素比较

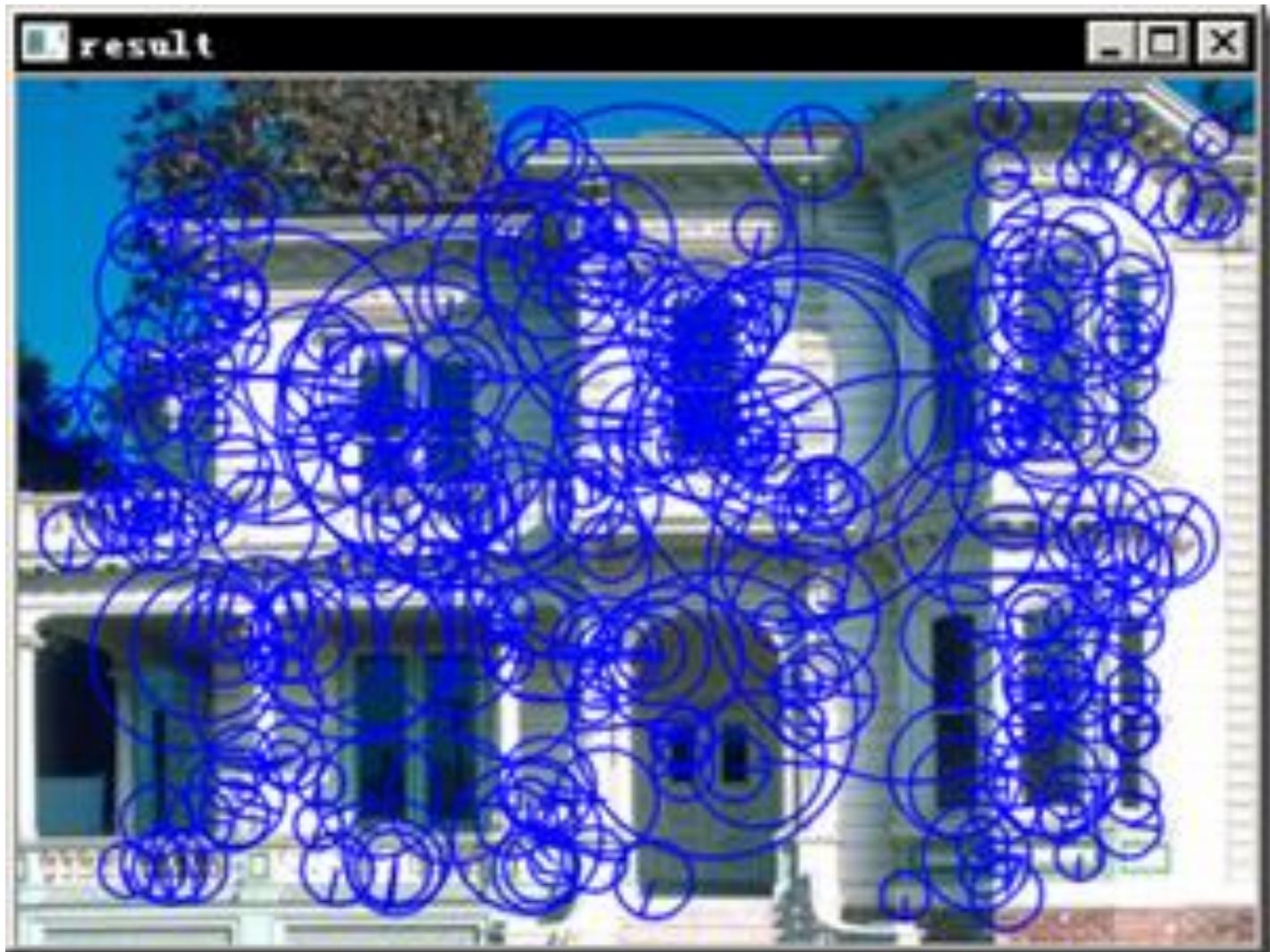


BRIEF的比较pattern

- ORB:旋转之后的BRIEF描述
- BRIEF是一种二进制描述，需要用汉明距离度量

2. 特征点法视觉里程计

- 例子: ORB特征



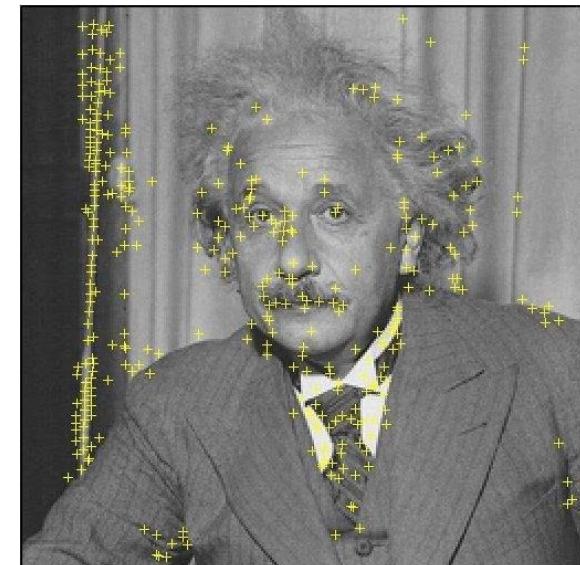
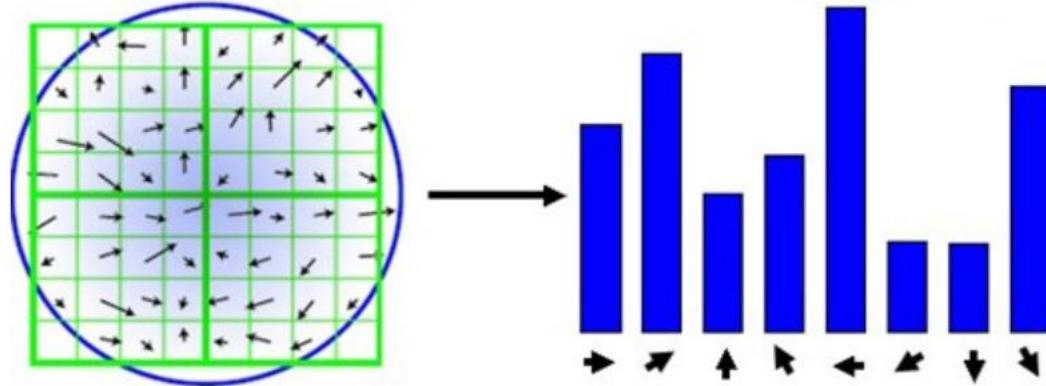
2. 特征点法视觉里程计

- 例子: ORB特征



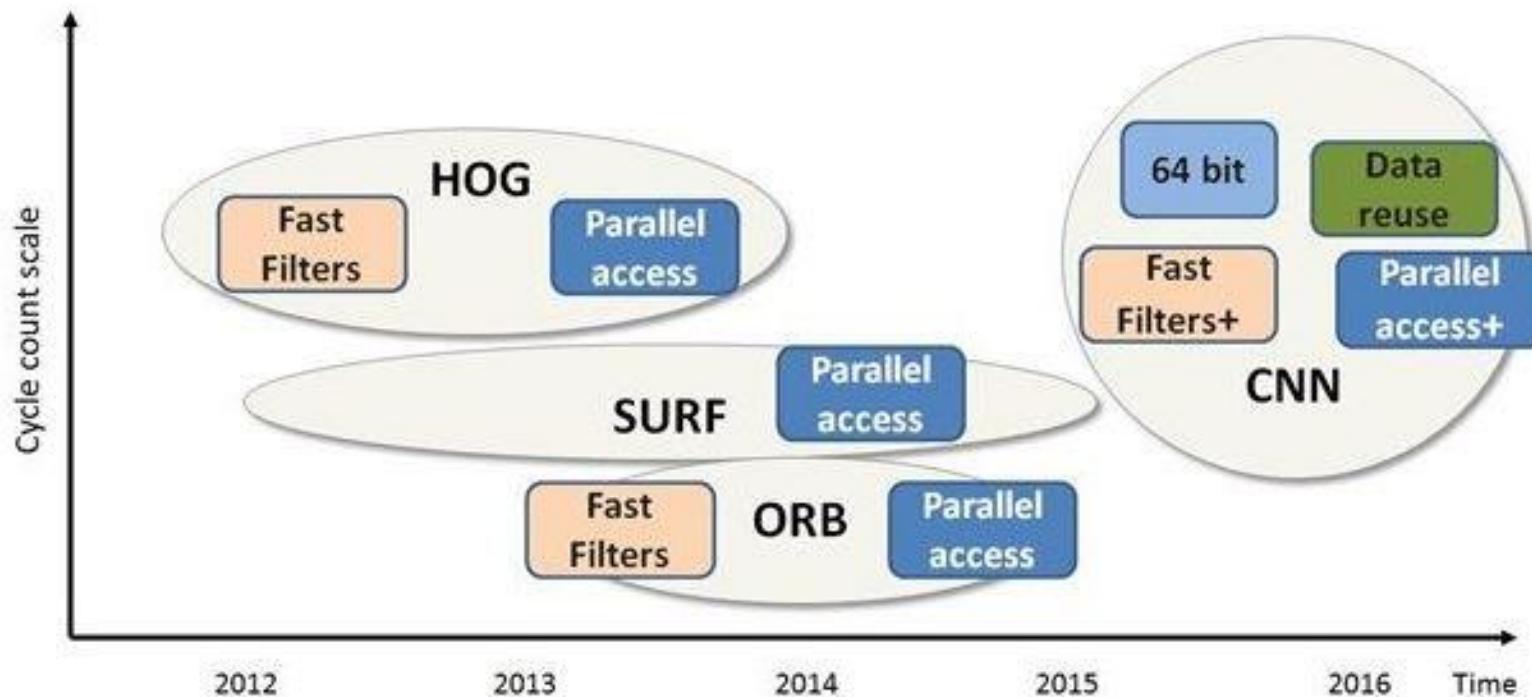
2. 特征点法视觉里程计

- 更多例子: SIFT 特征
- SIFT 描述子是关键点邻域高斯图像梯度统计结果的一种表示。
- 每一个关键点, 拥有三个信息: 位置、尺度以及方向
- 通过对关键点周围图像区域分块, 计算块内梯度直方图, 生成具有独特性的向量, 这个向量是该区域图像信息的一种抽象, 具有唯一性。
- 作者 Lowe 建议描述子使用在关键点尺度空间内 4×4 的窗口中计算的 8 个方向的梯度信息, 共 $4 \times 4 \times 8 = 128$ 维向量表征



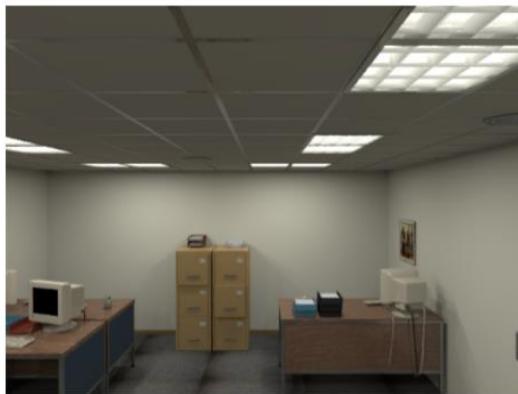
2. 特征点法视觉里程计

SURF和ORB等计算效率超高的算法为 CNN 之类的功能更强大的框架提供了实现的可能

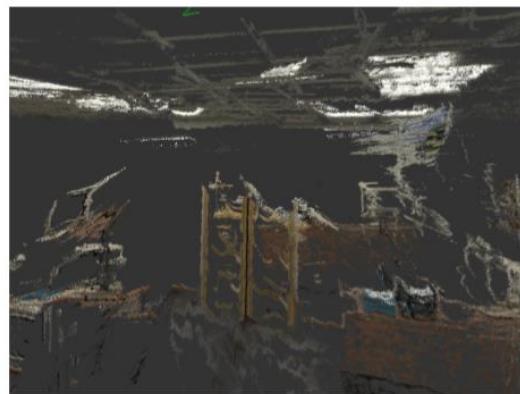


2. 特征点法视觉里程计

- Direct Multichannel Tracking, International Conference on 3D Vision 2017



(a) Color image at $t = 0$



(b) Point cloud using Gray (baseline case)



(c) Point cloud using AlexNet

Figure 3. 3D point cloud comparison for the Office Seq. N° 00 from the ICL-NUIM dataset. AlexNet features produced a denser and more accurate reconstruction. For example, the computer desks, file cabinets, and ceiling regions are better aligned for the most part.

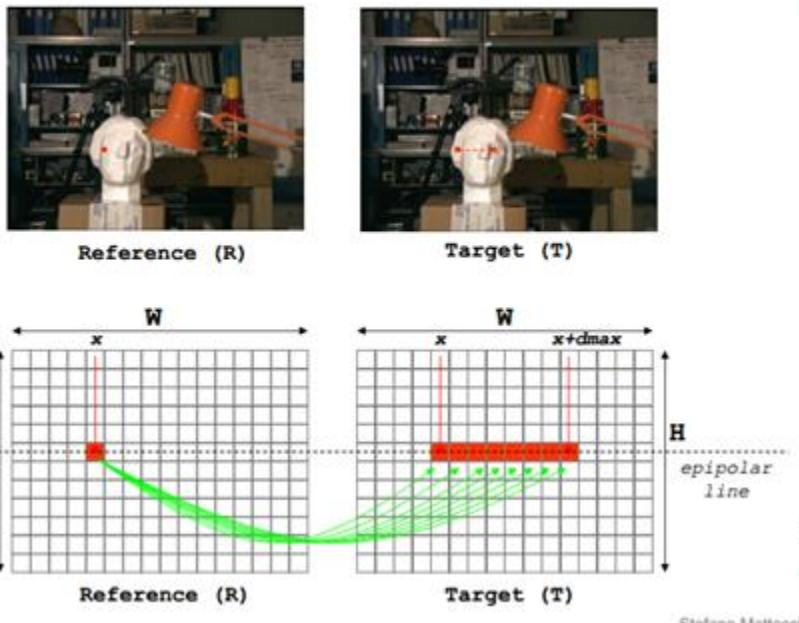
Instead of using a fixed feature transformation such as SIFT, learning based methods search among a large functional space of feature transformations for one that generalizes well to different cases using a large amount of training data.

One type of CNNs, such as the above-mentioned Siamese networks [34, 33, 4, 23, 15], was designed and trained specifically for the low-level vision task of patch matching:

2. 特征点法视觉里程计

特征匹配

The simplest (naive and unused) local approach:



基于线性搜索的视差计算

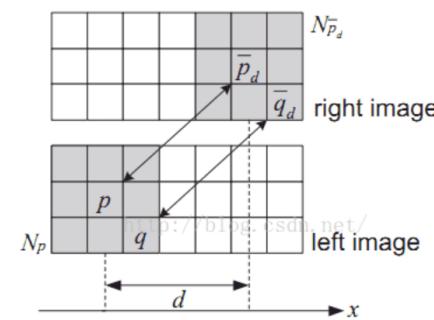
Stereo matching algorithm

对于校正的左右图像，其基线是水平的，选取左图特征点临近的 $N \times N$ 个像素组成patch，进行相似性检验

$$\text{SAD} = \sum_{(i,j) \in W} |I_1(i,j) - I_2(x+i, y+j)|$$

Sum of absolute differences, 将每个像素对应数值之差的绝对值求和，据此评估两个图像块的相似度

- (1) 构造一个小窗口,类似于卷积核;
- (2) 用窗口覆盖左边的图像,选择出窗口覆盖区域内的所有像素点;
- (3) 同样用窗口覆盖右边的图像并选择出覆盖区域的像素点;
- (4) 左边覆盖区域减去右边覆盖区域，并求出所有像素点灰度差的绝对值之和;
- (5) 移动右边图像的窗口，重复 (3) - (4) 的处理 (这里有个搜索范围,超过这个范围跳出);
- (6) 找到这个范围内SAD值最小的窗口,即找到了左图锚点的最佳匹配的像素块。



2. 特征点法视觉里程计

特征匹配

Stereo matching algorithm

对于校正的左右图像，其基线是水平的，选取左图特征点临近的N*N个像素组成patch，进行相似性检验

$$\text{SAD} \quad \sum_{(i,j) \in W} |I_1(i,j) - I_2(x+i, y+j)|$$

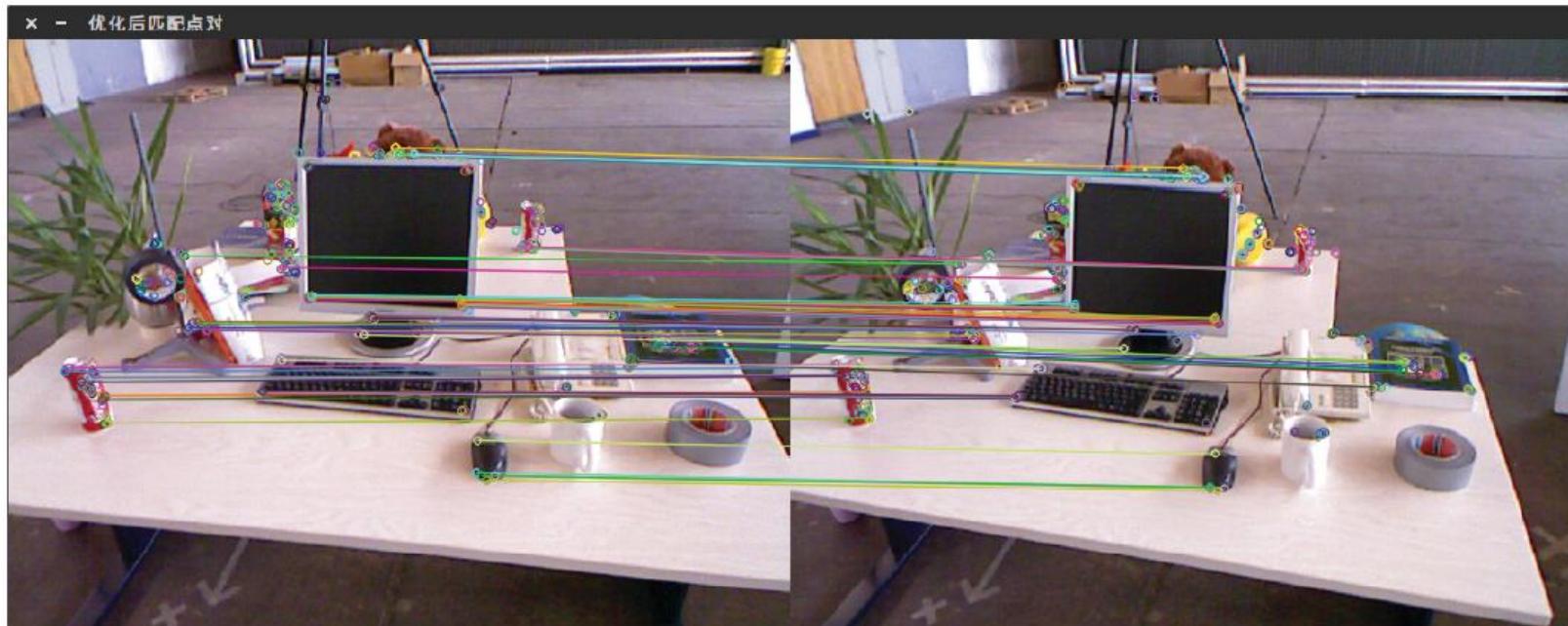
$$\text{SSD} \quad \sum_{(i,j) \in W} (I_1(i,j) - I_2(x+i, y+j))^2$$

$$\text{NCC} \quad \frac{\sum_{(i,j) \in W} I_1(i,j) \cdot I_2(x+i, y+j)}{\sqrt{\sum_{(i,j) \in W} I_1^2(i,j) \cdot \sum_{(i,j) \in W} I_2^2(x+i, y+j)}}$$

2. 特征点法视觉里程计

- 特征匹配

- 通过描述子的差异判断哪些特征为同一个点
- 暴力匹配: 比较图1中每个特征和图2特征的描述符距离 (Brute-Force Matcher)
- 加速: 快速最近邻 (FLANN)



2. 特征点法视觉里程计 工作流程

❖ Input

左右图像、前后图像序列、左右相机的内参、外参

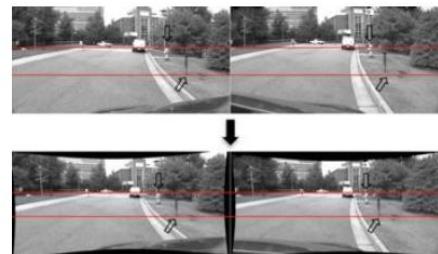
❖ Output

图像对之间的位姿关系、相机轨迹

(也可以生成基于特征点的稀疏3d地图, optional)

1. 获得左右图像、图像校正
2. 提取图像特征点
3. 左右图像特征匹配
获取特征点坐标
4. 前后图像特征匹配
5. 增量式的运动估计

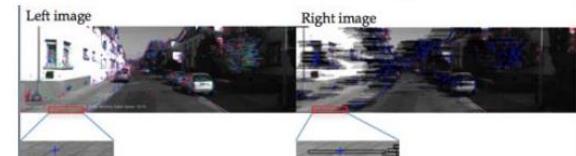
1. Rectification



2. Feature Extraction



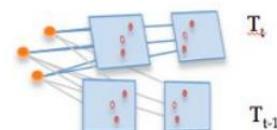
3. Stereo Feature Matching



4. Temporal Feature Matching

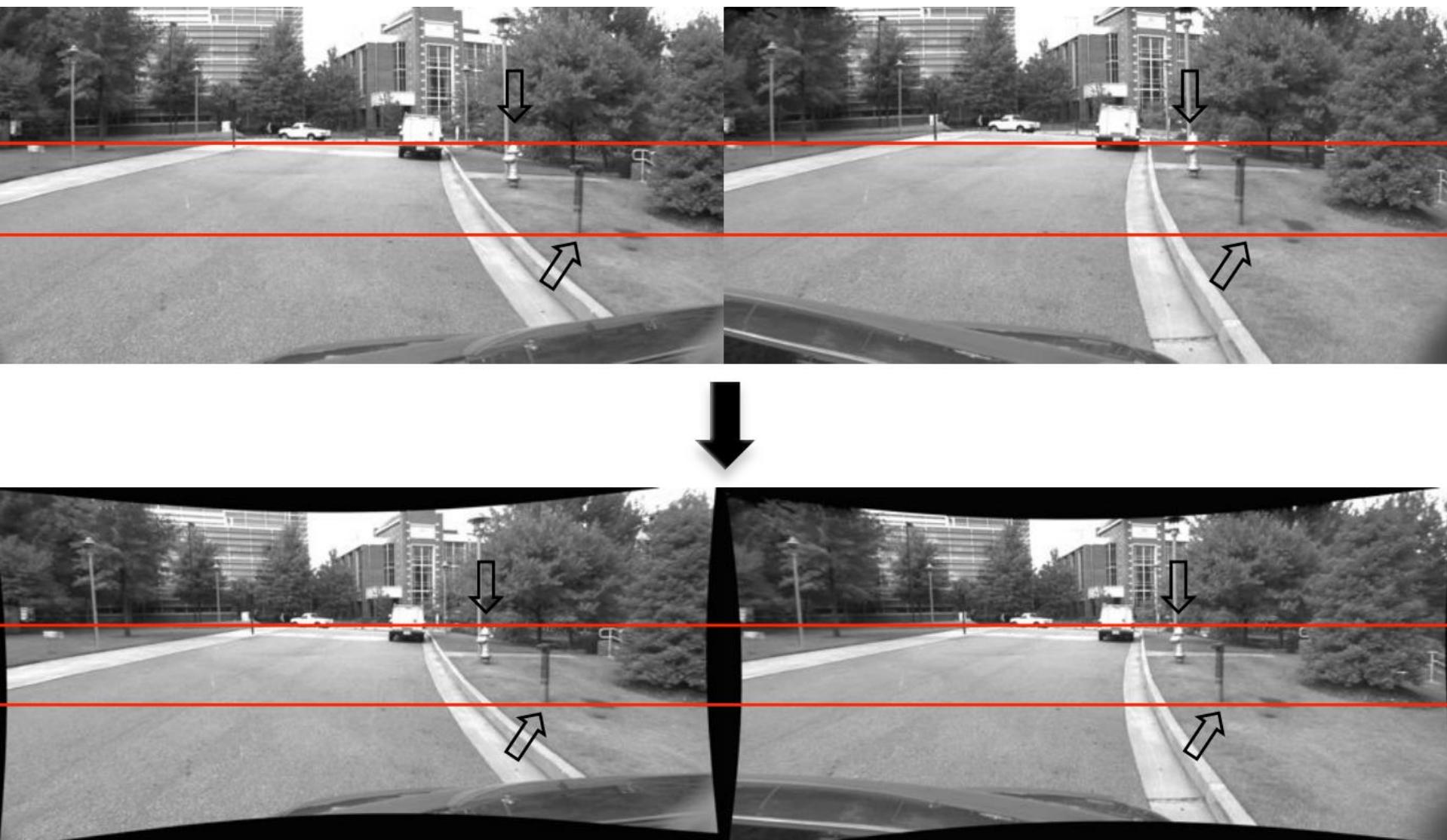


5. Incremental Pose Recovery/RANSAC



2. 特征点法视觉里程计

1. 获得左右图像、图像校正



2. 特征点法视觉里程计

2. 提取图像特征点

Haris角点检测

将图像窗口平移 $[u, v]$ 产生灰度变化 $E(u, v)$

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

由: $I(x+u, y+v) = I(x, y) + I_x u + I_y v + O(u^2, v^2)$, 得到:

$$E(u, v) = [u, v] \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

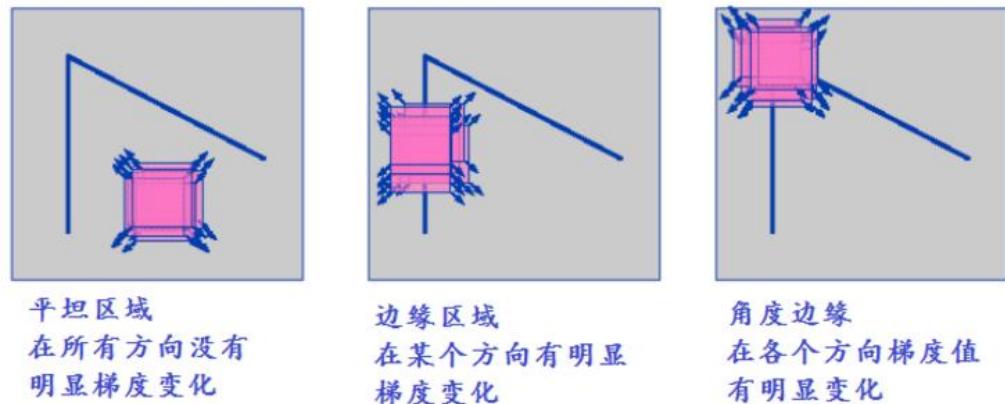
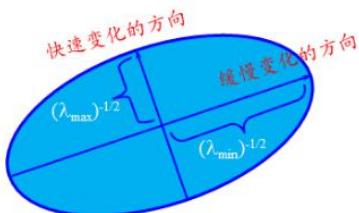
对于局部微小的移动量 $[u, v]$, 近似表达为:

$$E(u, v) \approx [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

其中 M 是 2×2 矩阵, 可由图像的导数求得:

$$M = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

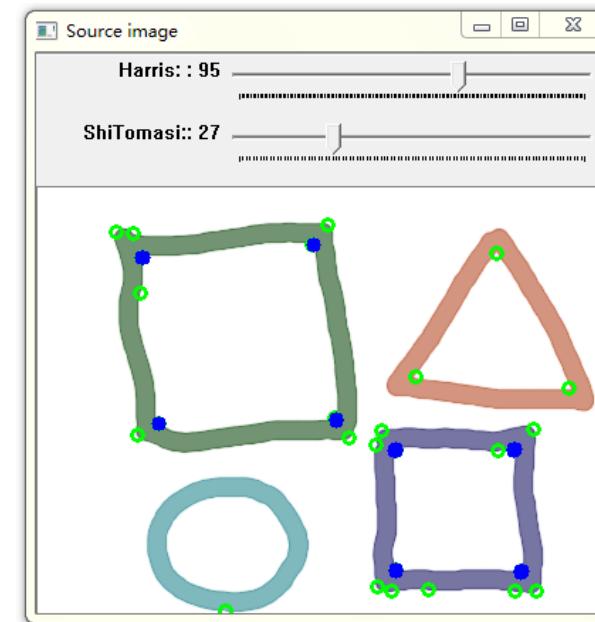
$$R = \lambda_1 \lambda_2 - k (\lambda_1 + \lambda_2)^2$$



平坦区域
在所有方向没有明显梯度变化

边缘区域
在某个方向有明显梯度变化

角度边缘
在各个方向梯度值有明显变化



http://blog.csdn.net/xiaowei_cqu/article/details/7805206

Shi and C. Tomasi. Good Features to Track. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, June 1994

2. 特征点法视觉里程计

3. 左右图像特征匹配、获取特征点坐标

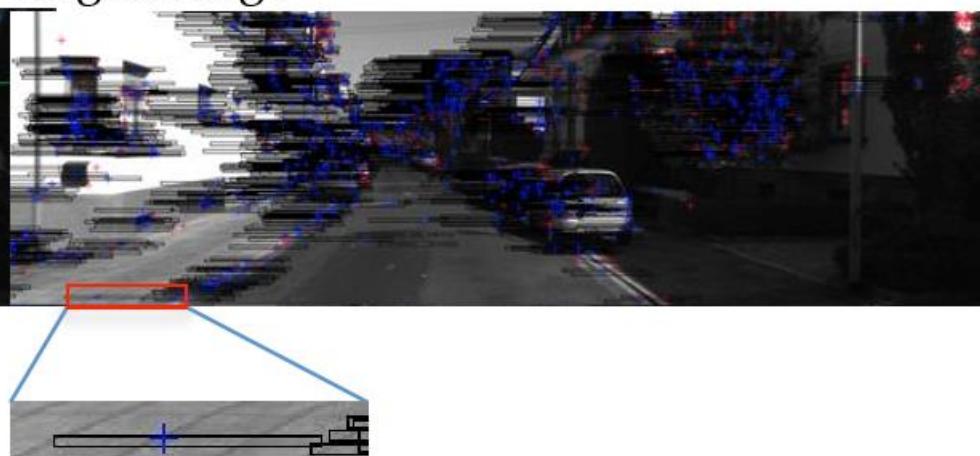
Stereo matching

- Match features between left/right images
- Since the images are rectified, we can restrict the search to a bounding box on the same scan-line

Left image



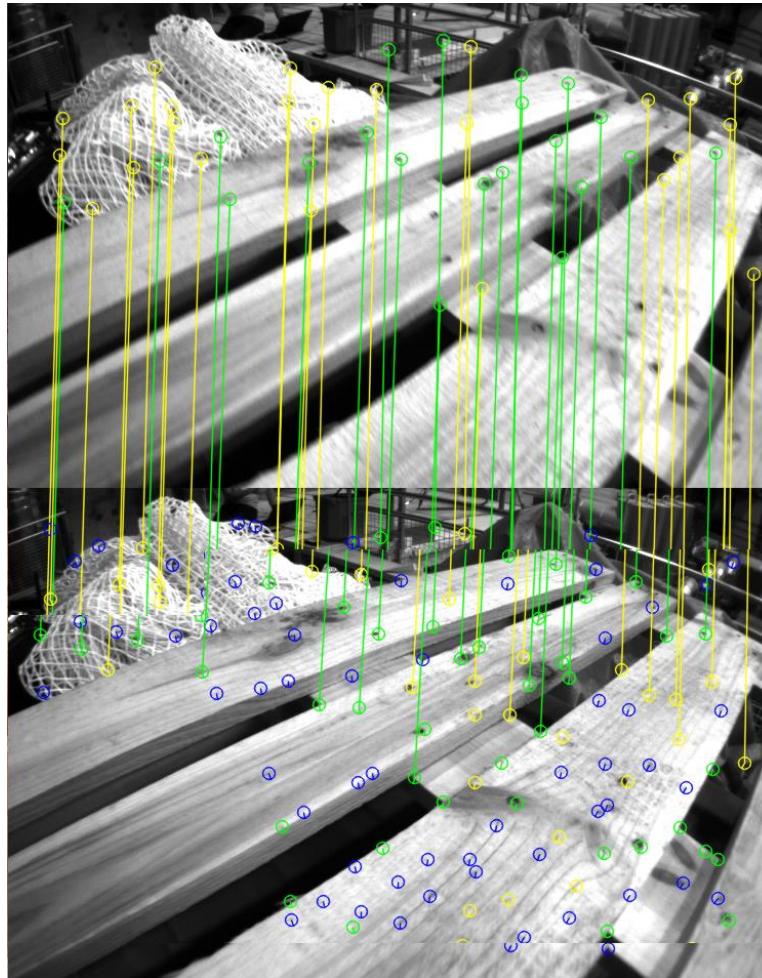
Right image



2. 特征点法视觉里程计

4. 前后图像特征匹配

根据描述子距离在限定区域内进行匹配



2. 特征点法视觉里程计

5. 增量式的运动估计

根据匹配后的特征点和特征点的坐标，恢复两个时刻相机的运动

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$

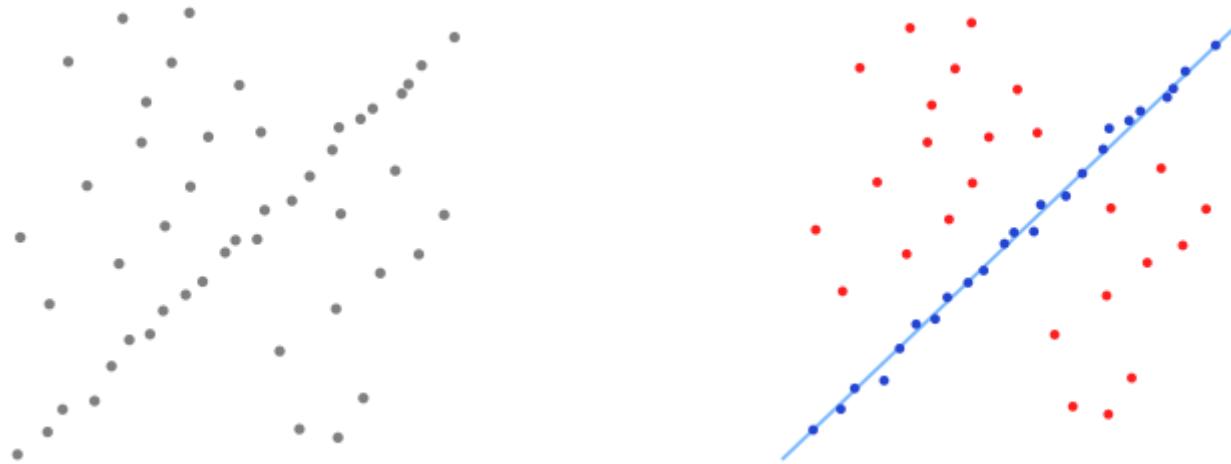
在匹配过程中，存在大量的误匹配，需要用RANSAC(RANdom SAmple Consensus,随机抽样一致，一般用MLESAC)！

- 从M中选取3个点对
- 由3个点对估计参数
- 根据估计参数得到的模型计算误差，如果小于给定误差，则认为是inlier
- 重复S次，找到inliers最大的点集
- 对该点集进行计算，得到最终T

2. 特征点法视觉里程计

RANSAC

采用迭代的方式从一组包含离群的被观测数据中估算出数学模型的参数



- ①考虑一个最小抽样集的势为n的模型(n 为初始化模型参数所需的最小样本数)和一个样本集P, 集合P的样本数 $\#(P) > n$, 从P中随机抽取包含n个样本的P的子集S初始化模型M;
- ②余集 $SC = P \setminus S$ 中与模型M的误差小于某一设定阈值t的样本集以及S构成 S^* 。 S^* 认为是内点集, 它们构成S的一致集(Consensus Set);
- ③若 $\#(S^*) \geq N$, 认为得到正确的模型参数, 并利用集 S^* (内点inliers)采用最小二乘等方法重新计算新的模型 M^* ; 重新随机抽取新的S, 重复以上过程。
- ④在完成一定的抽样次数后, 若未找到一致集则算法失败, 否则选取抽样后得到的最大一致集判断内外点, 算法结束。

Fischler, M.A. and Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM , 24(6): 381-395, 1981.

2. 特征点法视觉里程计

RANSAC通过反复选择数据中的一组随机子集来达成目标。被选取的子集被假设为局内点，并用下述方法进行验证：

- 1.首先我们先随机假设一小组局内点为初始值。然后用此局内点拟合一个模型，此模型适应于假设的局内点，所有的未知参数都能从假设的局内点计算得出。
- 2.用1中得到的模型去测试所有的其它数据，如果某个点适用于估计的模型，认为它也是局内点，将局内点扩充。
- 3.如果有足够多的点被归类为假设的局内点，那么估计的模型就足够合理。
- 4.然后，用所有假设的局内点去重新估计模型，因为此模型仅仅是在初始的假设的局内点估计的，后续有扩充后，需要更新。
- 5.最后，通过估计局内点与模型的错误率来评估模型。

整个这个过程为迭代一次，此过程被重复执行固定的次数，每次产生的模型有两个结局：

- 1、要么因为局内点太少，还不如上一次的模型，而被舍弃，
- 2、要么因为比现有的模型更好而被选用。

2. 特征点法视觉里程计

OpenCV中滤除误匹配对采用RANSAC算法寻找一个最佳单应性矩阵H，矩阵大小为 3×3 。RANSAC目的是找到最优的参数矩阵使得满足该矩阵的数据点个数最多，通常令 $h_{33} = 1$ 来归一化矩阵。由于单应性矩阵有8个未知参数，至少需要8个线性方程求解，对应到点位置信息上，一组点对可以列出两个方程，则至少包含4组匹配点对。

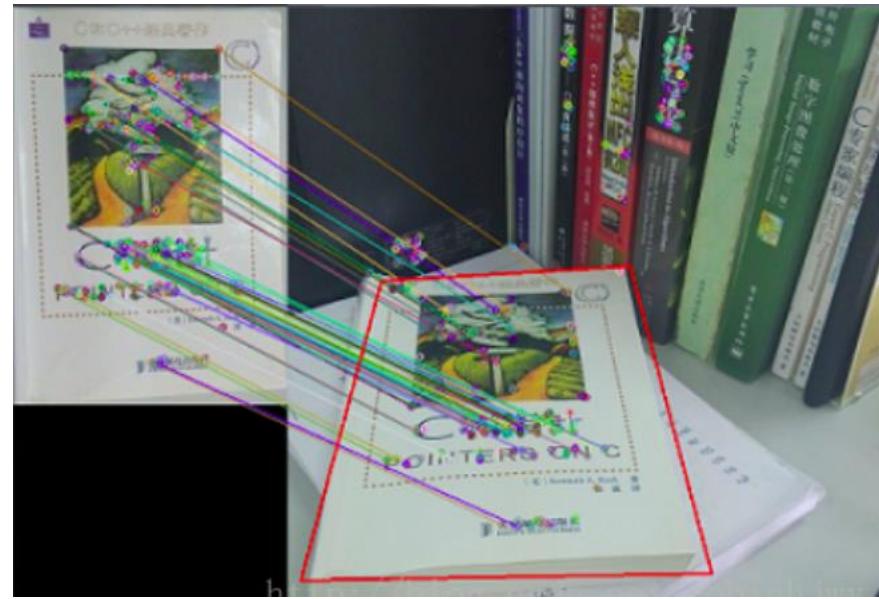
$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

其中 (x,y) 表示目标图像角点位置， (x',y') 为场景图像角点位置， s 为尺度参数。

RANSAC算法从匹配数据集中随机抽出4个样本并保证这4个样本之间不共线，计算出单应性矩阵，然后利用这个模型测试所有数据，并计算满足这个模型数据点的个数与投影误差(即代价函数)，若此模型为最优模型，则对应的代价函数最小。

$$\sum_{i=0}^n (x'_i - \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}})^2 + (y'_i - \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}})^2$$

2. 特征点法视觉里程计



1. 随机从数据集中随机抽出4个样本数据 (此4个样本之间不能共线), 计算出变换矩阵 H , 记为模型 M ;
2. 计算数据集中所有数据与模型 M 的投影误差, 若误差小于阈值, 加入内点集 I ;
3. 如果当前内点集 I 元素个数大于最优内点集 I_{best} , 则更新 $I_{best} = I$, 同时更新迭代次数 k ;
4. 如果迭代次数大于 k ,则退出 ; 否则迭代次数加1, 并重复上述步骤;

2. 特征点法视觉里程计

5. 增量式的运动估计-单目相机

根据匹配后的特征点和特征点的坐标, 恢复两个时刻相机的运动

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$

2D-2D：对极几何(八点法,Eight-point-algorithm) 三角化

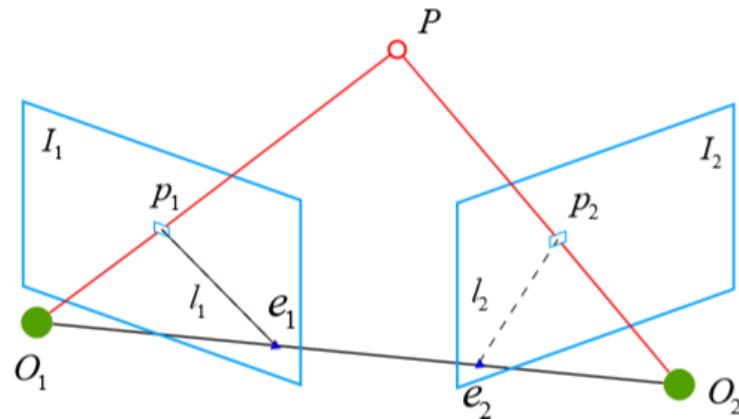


图 7-7 对极几何约束。

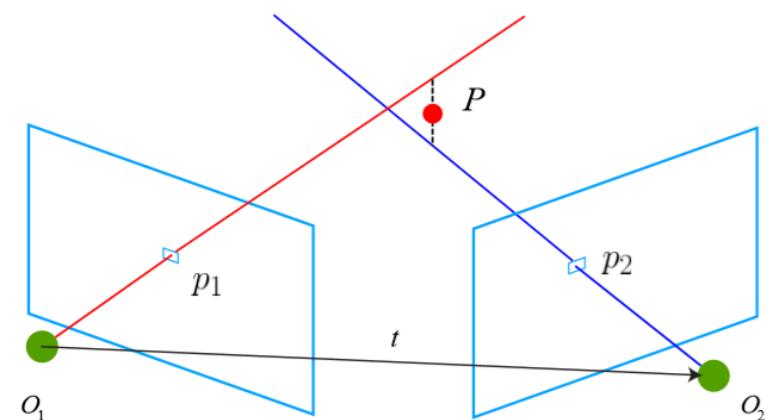


图 7-9 三角化获得地图点深度

2. 特征点法视觉里程计

5. 增量式的运动估计-深度相机

根据匹配后的特征点和特征点的坐标, 恢复两个时刻相机的运动

在匹配过程中, 存在大量的误匹配, 需要用RANSAC(MLESAC)

- 从M中选取3个点对
- 由3个点对估计参数
- 根据估计参数得到的模型计算误差, 如果小于给定误差, 则认为是inlier
- 重复S次, 找到inliers最大的点集

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$

Motion Estimation		
2-D-to-2-D	3-D-to-3-D	3-D-to-2-D
X	X	X

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2. 特征点法视觉里程计

5. 增量式的运动估计

根据匹配后的特征点和特征点的坐标, 恢复两个时刻相机的运动

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$

3D-2D PnP (Perspective-n-Point)

线性DLT解法

$$\arg \min_{T_k} \sum_i \| p_k^i - \hat{p}_{k-1}^i \|^2$$

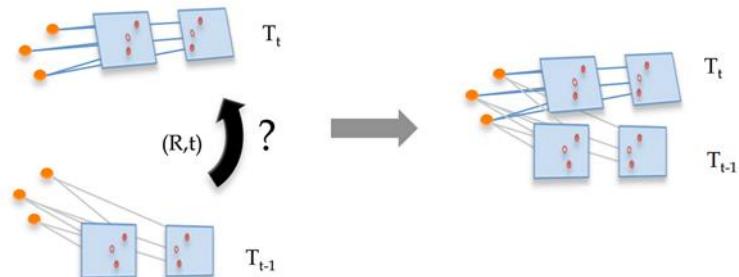
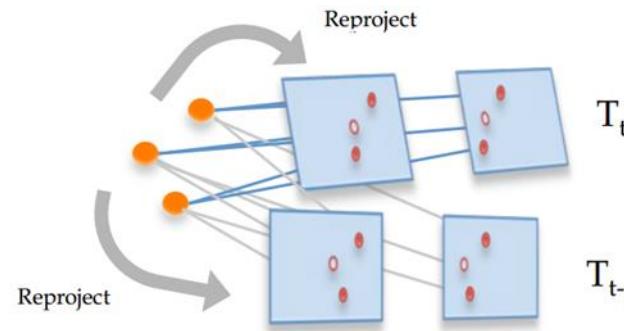
非线性BA解法

3D-3D ICP(Iterative Closest Point)

线性SVD解法

非线性BA解法

$$\arg \min_{T_k} \sum_i \| \tilde{X}_k^i - T_k \tilde{X}_{k-1}^i \|^2$$



2. 特征点法视觉里程计

5. 增量式的运动估计

3D-2D PnP

已知3D点的空间位置和相机上的投影点，求相机的旋转和平移(外参)

DLT直接线性解法

设空间点 $P = (X, Y, Z, 1)^T$

最下一行为

投影点为: $x = (u, v, 1)$ 归一化坐标

$s = [t_9, t_{10}, t_{11}, t_{12}][X, Y, Z, 1]^T$

投影关系: $sx = [R|t]p$

用它消掉前两行中的s

展开:

$$s \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

$$u_1 = \frac{t_1 X + t_2 Y + t_3 Z + t_4}{t_9 X + t_{10} Y + t_{11} Z + t_{12}}$$

$$v_1 = \frac{t_5 X + t_6 Y + t_7 Z + t_8}{t_9 X + t_{10} Y + t_{11} Z + t_{12}}.$$

2. 特征点法视觉里程计

5. 增量式的运动估计

3D-2D PnP

DLT直接线性解法

$$s \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\mathbf{t}_1 = (t_1, t_2, t_3, t_4)^T, \mathbf{t}_2 = (t_5, t_6, t_7, t_8)^T, \mathbf{t}_3 = (t_9, t_{10}, t_{11}, t_{12})^T$$

$\mathbf{t}_1^T \mathbf{P} - \mathbf{t}_3^T \mathbf{P} u_1 = 0$, 共有 N 个特征点, 可以列出线性方程组:

$$\mathbf{t}_2^T \mathbf{P} - \mathbf{t}_3^T \mathbf{P} v_1 = 0.$$

由于 \mathbf{t} 一共有 12 维, 因此最少通过六对匹配点, 当匹配点大于六对时可以使用 SVD 等方法对超定方程求最小二乘解。

$$\begin{pmatrix} \mathbf{P}_1^T & 0 & -u_1 \mathbf{P}_1^T \\ 0 & \mathbf{P}_1^T & -v_1 \mathbf{P}_1^T \\ \vdots & \vdots & \vdots \\ \mathbf{P}_N^T & 0 & -u_N \mathbf{P}_N^T \\ 0 & \mathbf{P}_N^T & -v_N \mathbf{P}_N^T \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = 0$$

2. 特征点法视觉里程计

5. 增量式的运动估计

3D-2D PnP

已知3D点的空间位置和相机上的投影点，求相机的旋转和平移(外参)

P3P解法

$$\Delta Oab - \Delta OAB, \quad \Delta Obc - \Delta OBC, \quad \Delta Oac - \Delta OAC.$$

考虑 Oab 和 OAB 的关系。利用余弦定理，有：

$$OA^2 + OB^2 - 2OA \cdot OB \cdot \cos \langle a, b \rangle = AB^2.$$

$$OA^2 + OB^2 - 2OA \cdot OB \cdot \cos \langle a, b \rangle = AB^2$$

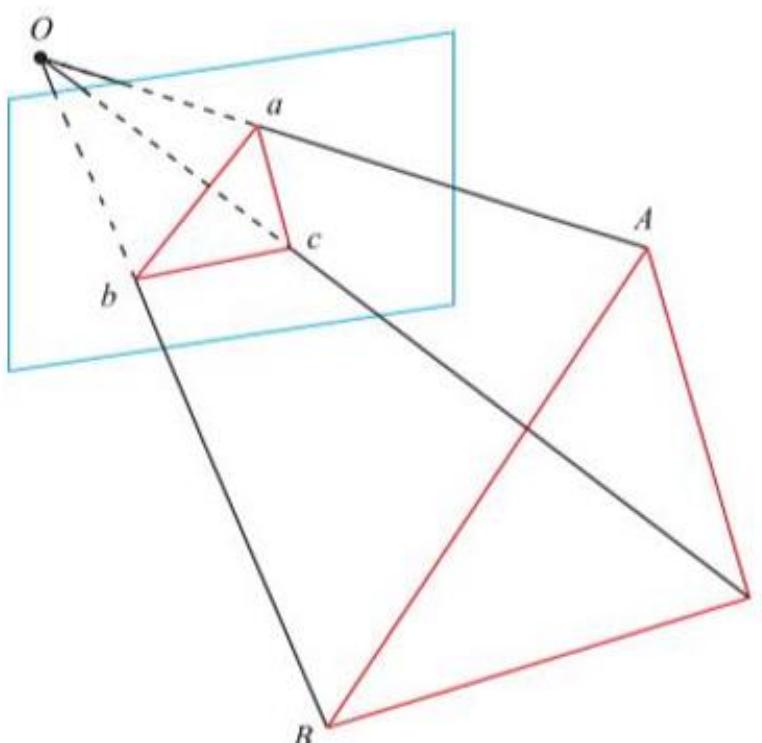
$$OB^2 + OC^2 - 2OB \cdot OC \cdot \cos \langle b, c \rangle = BC^2$$

$$OA^2 + OC^2 - 2OA \cdot OC \cdot \cos \langle a, c \rangle = AC^2.$$

$$v = AB^2/OC^2, uv = BC^2/OC^2, wv = AC^2/OC^2, \text{ 有:}$$

$$(1 - u) y^2 - ux^2 - \cos \langle b, c \rangle y + 2uxy \cos \langle a, b \rangle + 1 = 0$$

$$(1 - w) x^2 - wy^2 - \cos \langle a, c \rangle x + 2wxy \cos \langle a, b \rangle + 1 = 0.$$



2. 特征点法视觉里程计

5. 增量式的运动估计

3D-3D ICP解法

给定配对好的两组3D点，求其旋转和平移，可用迭代最近点（Iterative Closest Point, ICP求解）

思想：

按照一定的约束条件，找到最邻近点(p_i, q_i)，然后计算出最优匹配参数 R 和 t ，使得误差函数最小(无相机模型)

计算流程：

假设给两个三维点集 X_1 和 X_2 , ICP方法的配准步骤如下：

第一步，计算 X_2 中的每一个点在 X_1 点集中的对应近点；

第二步，求得使上述对应点对平均距离最小的刚体变换，求得平移参数和旋转参数；

第三步，对 X_2 使用上一步求得的平移和旋转参数，得到新的变换点集；

第四步，如果新的变换点集与参考点集满足两点集的平均距离小于某一给定阈值，则停止迭代计算，否则新的变换点集作为新的 X_2 继续迭代，直到达到目标函数的要求。

2. 特征点法视觉里程计

5. 增量式的运动估计

3D–3D ICP

给定配对好的两组3D点，求其旋转和平移，可用迭代最近点（*Iterative Closest Point, ICP*求解）

设： $P = \{p_1, \dots, p_n\}, P' = \{p'_1, \dots, p'_n\}$,

运动关系： $\forall i, p_i = Rp'_i + t.$

同样定义误差项： $e_i = p_i - (Rp'_i + t).$

以及最小二乘问题：

$$\min_{R,t} J = \frac{1}{2} \sum_{i=1}^n \|(p_i - (Rp'_i + t))\|_2^2.$$

2. 特征点法视觉里程计

5. 增量式的运动估计

3D–3D ICP

给定配对好的两组3D点，求其旋转和平移，可用迭代最近点（Iterative Closest Point, ICP求解）

定义质心： $p = \frac{1}{n} \sum_{i=1}^n (p_i)$, $p' = \frac{1}{n} \sum_{i=1}^n (p'_i)$.

改写目标函数：

$$\begin{aligned}\frac{1}{2} \sum_{i=1}^n \|p_i - (Rp_i' + t)\|^2 &= \frac{1}{2} \sum_{i=1}^n \|p_i - Rp_i' - t - p + Rp' + p - Rp'\|^2 \\ &= \frac{1}{2} \sum_{i=1}^n \|(p_i - p - R(p_i' - p')) + (p - Rp' - t)\|^2 \\ &= \frac{1}{2} \sum_{i=1}^n (\|p_i - p - R(p_i' - p')\|^2 + \|p - Rp' - t\|^2) +\end{aligned}$$

交叉项部分求和为零 $\Rightarrow 2(p_i - p - R(p_i' - p'))^T (p - Rp' - t)$.

目标函数简化为：

$$\min_{R,t} J = \frac{1}{2} \sum_{i=1}^n \|p_i - p - R(p_i' - p')\|^2 + \|p - Rp' - t\|^2$$

只和R有关

和R,t都有关

2. 特征点法视觉里程计

5. 增量式的运动估计

3D–3D ICP

给定配对好的两组3D点，求其旋转和平移，可用迭代最近点（Iterative Closest Point, ICP求解）

旋转的求取

- 定义去质心坐标：

$$\mathbf{q}_i = \mathbf{p}_i - \bar{\mathbf{p}}, \quad \mathbf{q}'_i = \mathbf{p}'_i - \bar{\mathbf{p}'}$$

$$\min_{\mathbf{R}, \mathbf{t}} J = \frac{1}{2} \sum_{i=1}^n \| \mathbf{p}_i - \bar{\mathbf{p}} - \mathbf{R}(\mathbf{p}'_i - \bar{\mathbf{p}'}) \|^2 + \| \bar{\mathbf{p}} - \mathbf{R}\bar{\mathbf{p}'} - \mathbf{t} \|^2$$

- 最小化：

$$\text{转换为} \Rightarrow \mathbf{R}^* = \arg \min_{\mathbf{R}} \frac{1}{2} \sum_{i=1}^n \| \mathbf{q}_i - \mathbf{R}\mathbf{q}'_i \|^2 = \frac{1}{2} \sum_{i=1}^n \mathbf{q}_i^T \mathbf{q}_i + \mathbf{q}'_i^T \mathbf{R}^T \mathbf{R} \mathbf{q}'_i - 2 \mathbf{q}_i^T \mathbf{R} \mathbf{q}'_i$$

$$\text{转换为} \Rightarrow \sum_{i=1}^n -\mathbf{q}_i^T \mathbf{R} \mathbf{q}'_i = \sum_{i=1}^n -\text{tr}(\mathbf{R} \mathbf{q}'_i \mathbf{q}_i^T) = -\text{tr} \left(\mathbf{R} \sum_{i=1}^n \mathbf{q}'_i \mathbf{q}_i^T \right)$$

- SVD解法：
 $\mathbf{W} = \sum_{i=1}^n \mathbf{q}_i \mathbf{q}_i^T, \quad \mathbf{W} = \mathbf{U} \Sigma \mathbf{V}^T, \quad \mathbf{R} = \mathbf{U} \mathbf{V}^T$

2. 特征点法视觉里程计

5. 增量式的运动估计

3D–3D ICP

给定配对好的两组3D点，求其旋转和平移，可用迭代最近点（*Iterative Closest Point, ICP*求解）

解得 R 后，求解 t 即可

$$t^* = p - Rp'$$

Thanks