

计算机视觉复习

计算机视觉复习

Chapter1-Introduction

- 什么是计算机视觉
- 计算机视觉的五大研究内容
- Marr视觉计算理论
- 格式塔理论 Gestalt theory

Chapter2-Binary 二值图像

- 几何特性
- 投影计算
- 连通区域

Chapter3-Edge 边缘

- 四种最主要的不连续(discontinuity)
- 边缘检测的基本思想
- 基于一阶导数的边缘检测
- 基于二阶导数的边缘检测
- Canny边缘检测

Chapter4-Feature 局部特征

- Harris角点检测
- SIFT描述子的计算

Chapter5-Curve 曲线

- Hough变换

Chapter6-Eigenface 人脸识别

- PCA主元分析
- Eigenface

Chapter7-Frequency 图像频域与图像分解

- 从图像分解角度，理解傅里叶变换的意义
- 怎么理解拉普拉斯图像金字塔的每一层是带通滤波？

Chapter8-Image Stitching 图像拼接

- RANSAC
- 图像拼接

Chapter9&10- Visual Recognition&BoW 物体识别

- Visual Recognition
- 基于词袋(BoW)的物体分类

Chapter12-Motion(Optical flow) 移动(光流)

- 光流解决的是什么问题
- 光流的三个基本假设是什么
- 推导一个点的约束公式
- 哪些位置光流比较可靠？为什么？
- 提高鲁棒性

Chapter11-Deep Learning 深度学习

- 深度学习
- CNN
- BP算法

Chapter13-Camera 相机

- 基本概念——理解景深/光圈/焦距/视场
- 理想的针孔相机模型(pinhole camera)
- 畸变
- 外参矩阵
- 画图展示所有过程

Chapter14-Camera Calibration 相机定标

- 一般的相机定标
- 基于Homography的相机定标

Chapter15-Stereo Vision 立体视觉
 立体视觉的三角测量基本原理 (Triangulation公式)
 立体视觉的基本步骤
Chapter16-Structured Lighting 3D Scan 结构光三维成像
 结构光成像系统的构成
 利用结构光获取三维数据的基本原理
 ICP算法(Iterative Closest Point)迭代最近点方法
Chapter17-Segmentation 图像分割
 基于k-means聚类的图像分割
 基于Mean Shift的图像分割

Chapter1-Introduction

什么是计算机视觉

人类感知外部世界主要是通过视觉、触觉、听觉和嗅觉等感觉器官，其中**约70%-80%**的信息是由视觉获得的

计算机视觉：an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do. -- Wikipedia

中心任务：对图像进行理解（形状、位置、运动、类别）

计算机视觉的五大研究内容

1. **低层视觉(low level/early)** 对输入的原始图像进行处理。常借用图像处理技术和算法，如图像滤波、图像增强、边缘检测、纹理检测、运动检测，以便从图像中提取角点、边缘、线条、边界、色彩、纹理、运动等关于场景的基本特征。
2. **中层视觉(middle level)** 恢复场景的深度、表面法线方向、轮廓等有关场景的**2.5维信息**，实现的途径有立体视觉(stereo vision)、测距成像(rangefinder)、从X恢复形状(shape from X, X=明暗、纹理、运动)。系统标定、系统成像模拟等研究内容一般也是在这个层次上进行的。**分割、拟合等**
3. **高层视觉(high level)** 在原始输入图像、图像基本特征、2.5维的基础上，恢复物体的完整三维图，建立物体三位描述，识别物体并确定物体的位置和方向等信息。
4. **输入设备(input device)** 包括成像设备和数字化设备，成像设备是指通过光学摄像机或红外、激光、超声、X射线对周围场景或物体进行探测成像，得到关于场景或物体的二维或三维数字化图像。
5. **体系结构(system architecture)** 在高度抽象的层次上，根据系统模型而不是根据实现设计的具体例子来研究系统的结构。为了说明这一点，可以考虑建筑设计中某一时期的建筑风格（如清朝时期）和根据这一风格设计出来的具体建筑之间的区别。体系结构研究涉及一系列相关的课题：并行结构、分层结构、信息流结构、拓扑结构以及从设计到实现的途径等等

Marr视觉计算理论

1. 信息处理分析的三个层次

计算层	表示和算法层	实现层
计算的目的是什么? 为什么这一计算是合适的? 执行计算的策略是什么?	如何实现这个计算? 输入输出的表示是什么 表示与表示之间的变换是什么	在物理上如何实现这些表示算法

2. 视觉表示框架的三个层次

第一阶段(Primary Sketch): 将输入的原始图像进行处理, 抽取图像中诸如角点、边缘、纹理、线条、边界等基本特征, 这些特征的集合称为基元图。

第二阶段(2.5D Sketch): 指在以观测者为中心的坐标系中, 由输入图像和基元图恢复场景可见部分的深度、法线方向、轮廓等, 这些信息包含了深度信息, 但不是真正的物体三维表示, 因此称为二维半图

第三阶段(3D Model): 在一物体为中心的坐标系中, 由输入图像、基元图、二维半图来恢复、表示和识别三维物体

格式塔理论 Gestalt theory

1. **Law of Proximity(接近原理):** 人们自然而然的对靠近的事物进行分类
2. **Law of Similarity(相似原理):** 人们在观察对象时容易把相似的物体分为一组
3. **Law of Common Fate(共同命运原理):** 对象容易被视为行进在光滑路径上的线条
4. **Law of Symmetry(相似原理):** 人们观察对象时容易将对象视为对称的且围绕一个中心
5. **Law of Continuity(连续原理):** 对象的元素容易群聚在一起, 且能形成整体
6. **Law of Closure(闭合原理):** 人们在观察诸如形状、信封、照片时自动补全他们

Chapter2-Binary 二值图像

几何特性

1. 面积(零阶矩)

$$A = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} B[i, j]$$

2. 区域中心(一阶矩)

$$\bar{x} = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} jB[i, j]}{A}$$

$$\bar{y} = -\frac{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} iB[i, j]}{A}$$

3. 方向

某些形状 (圆) 是没有方向的; 假定物体是长形的, 长轴方向为物体的方向

求方向→最小化问题:

$$\chi^2 = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} r_{ij}^2 B[i, j], r_{ij} \text{ 是点 } [i, j] \text{ 到直线的距离}$$

$$\begin{array}{l} \text{直线方程} \quad \rho = x \cos \theta + y \sin \theta \\ \text{距离} \quad r^2 = (x \cos \theta + y \sin \theta - \rho)^2 \end{array}$$

4. 伸长率

$$E = \frac{\chi_{max}}{\chi_{min}}$$

5. 密集度

$$C = \frac{A}{p^2}$$

其中 p : 周长, A : 面积

给定周长, 密集度越高, 围成的面积越大

圆 > 正方形 > 长方形

6. 形态比

区域的最小外接矩形的长与宽之比

7. 欧拉数(亏格数, genus)

连通分量数减去**洞数**

$$E = C - H$$

平稳、旋转和比例不变

A

$$E = 1 - 1 = 0$$

B

$$E = 1 - 2 = -1$$

C

$$E = 1 - 0 = 1$$

投影计算

- 定义**: 给定一条直线, 用垂直该直线的一簇等间距直线将一幅二值图像分割成若干条, 每一条**内像素值为1的像素**的数量
- 水平投影**: 计算每一列像素为1的个数
- 垂直投影**: 计算每一行像素为1的个数
- 对角线投影**: 从左下到右上, 计算每一个对角线像素为1的个数

仿射变换 \Rightarrow 将右上角像素映射成对角线投影的第一个位置, 左下角对应最后一个对角线投影标号对应关系, 其中 $[i, j]$ 为原像素坐标, 图像大小为 $n * m$

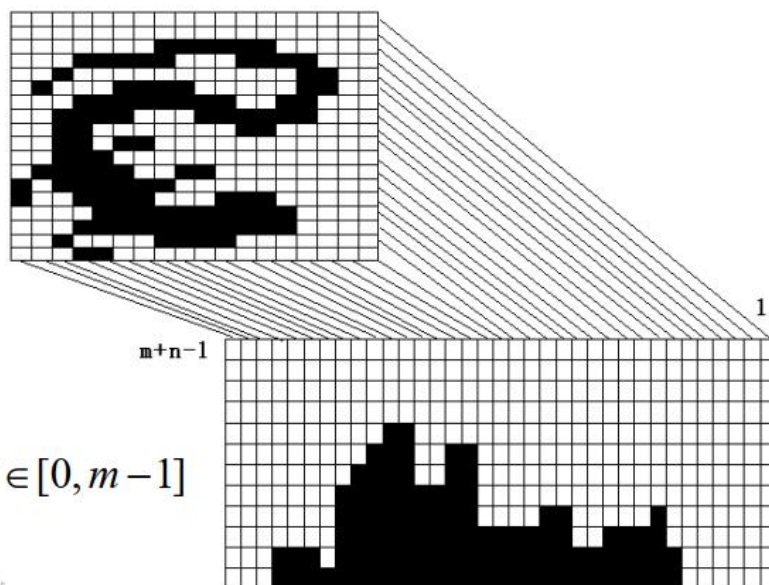
- 计算当前行和列对应的投影分布图位置标号 (nXm)
– 仿射变换

$$d = ai + bj + c$$

⇒

$$d = i - j + m$$

其中 $i \in [0, n-1]$, $j \in [0, m-1]$



浙江大学计算机学院

<{

连通区域

1. 领点概念

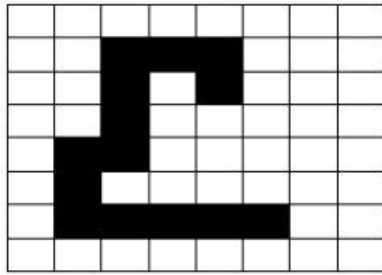
		[i-1, j]	
[i, j-1]	[i, j]	[i, j+1]	
		[i+1, j]	

[i-1, j-1]	[i-1, j]	[i-1, j+1]
[i, j-1]	[i, j]	[i, j+1]
[i+1, j-1]	[i+1, j]	[i+1, j+1]

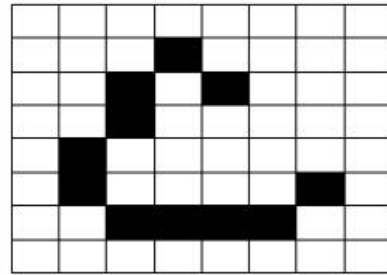
四连通邻点

八连通邻点

2. 连通性



四连通路径



八连通路径

从像素 $[i_0, j_0]$ 到像素 $[i_n, j_n]$ 的路径指：
像素序列 $[i_0, j_0], [i_1, j_1], \dots, [i_n, j_n]$, 且序列
中相邻两点为邻点

3. **连通分量定义**：连通像素的集合

4. **连通分量标记算法(递归算法)**：

- 扫描图像，找到没有标记的一个前景点(即像素值为1)，分配标记L
- 递归分配标记L给该点的邻点
- 如果不存在没标记的点，停止
- 返回第一步

5. **连通分量标记算法(序贯算法)**：

- (1) 从左到右、从上到下扫描图像
- (2) 如果像素值为1，则（分四种情况）
 - 如果上面点和左面点有且仅有一个标记，则复制这一标记
 - 如果两点有相同标记，复制这一标记
 - 如果两点有不同标记，则复制上点的标记且将两个标记输入等价表中作为等价标记
 - 否则给这一个像素点分配一新的标记并将这一标记输入等价表
- (3) 如果需要考虑更多点，则返回(2)
- (4) 在等价表的每一等价集中找到最低的标记
- (5) 扫描图像，用等价表中的最低标记取代每一标记

6. **区域边界跟踪算法**：

符号定义： c ：当前点（在边界上） b ：当前点的邻域点（不在边界上）

过程：

- 左 → 右，上 → 下，扫描图像，求区域S的起始点 $s(k) = (x(k), y(k)), k = 0$
- 用c表示当前边界上被跟踪的像素点，置 $c = s(k)$ ，记c的左邻点为b, $b \in \bar{S}$
- 按逆时针方向记从b开始的c的8个8邻点分别为 $n_1, n_2, \dots, n_8, k = k + 1$
- 从b开始，沿逆时针方向找到第一个 $n_i \in S$
- 置 $c = s(k) = n_i, b = n_{i-1}$
- 重复步骤(3),(4),(5)，直到 $s(k) = s(0)$

Chapter3-Edge 边缘

四种最主要的不连续(discontinuity)

1. surface normal discontinuity
2. depth discontinuity
3. surface color discontinuity
4. illumination discontinuity

边缘检测的基本思想

An edge is a place of rapid change in the image intensity function

先确定图像中的边缘像素，然后把这些像素连接在一起就构成所需的区域边界。

函数导数反映图像灰度变化的显著程度，一阶导数的**局部最大值(Local maxima)**，或二阶导数的**过零点(Zero-crossing)**

基于一阶导数的边缘检测

- 梯度

- 定义：是图像对应二位函数的一阶导数

$$G(x, y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- 幅值： $|G(x, y)| = \sqrt{G_x^2 + G_y^2} = |G_x| + |G_y| \approx \max(|G_x|, |G_y|)$
- 方向： $a(x, y) = \arctan(G_y/G_x)$

- 常用模板

- 差分近似：

$$G_x = f[x+1, y] - f[x, y], G_y = f[x, y] - f[x, y+1] \Rightarrow G_x = [-1 \quad 1], G_y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

- Roberts交叉算子： $G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

- Sobel算子： $G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

- Prewitt算子：运算速度较快 $G_x = G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, G_y = G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

均值差分：一定邻域内灰度平均值之差

↓ 3×3邻域加权

- 均值差分：

$$G_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$
$$G_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$

a_0	a_1	a_2
a_7	$a_{[i,j]}$	a_3
a_6	a_5	a_4

C=1: Prewitt算子

C=2: Sobel算子

C=3: Sethi算子

基于二阶导数的边缘检测

- 拉普拉斯算子

- 是二阶导数的二维等效式

$$\begin{aligned}\nabla^2 f &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ \frac{\partial^2 f}{\partial x^2} &= \frac{\partial G_x}{\partial x} \\ &= \frac{\partial(f[i, j+1] - f[i, j])}{\partial x} \\ &= \frac{\partial f[i, j+1]}{\partial x} - \frac{\partial f[i, j]}{\partial x} \\ &= (f[i, j+1] - f[i, j]) - (f[i, j] - f[i, j-1]) \\ \frac{\partial^2 f}{\partial x^2} &= (f[i, j+1] - 2f[i, j]) + f[i, j-1] \\ \frac{\partial^2 f}{\partial y^2} &= (f[i+1, j] - 2f[i, j]) + f[i-1, j]\end{aligned}$$

用差分近似微分:

- 等效模板

$$\nabla^2 \approx \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- LoG算子

- LoG = Laplacian of Gaussian **高斯滤波+拉普拉斯边缘检测**
- 基本特征:
 - 平滑滤波器是高斯滤波器
 - 采用拉普拉斯算子计算二阶导数
 - 边缘检测判据是二阶导数零交叉点并对应一阶导数的较大峰值
 - 使用线性内插方法在子像素分辨率水平上估计边缘的位置
- 两种等效算法
 - (1) 图像与高斯函数卷积, 再求卷积的拉普拉斯微分
 - (2) 求高斯函数的拉普拉斯微分, 再与图像卷积
- 为什么加上G: 平滑去噪和边缘检测是一对矛盾, 应用高斯函数的一阶导数, 在二者之间获得最佳平衡
- 推导过程:

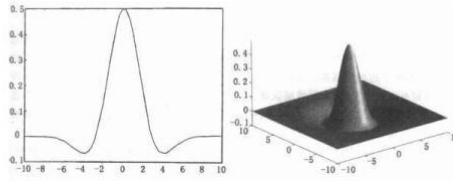
$$h(x, y) = \nabla^2 [g(x, y) * f(x, y)]$$

根据卷积求导法

$$h(x, y) = [\nabla^2 g(x, y)] * f(x, y)$$

$$\nabla^2 g(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

墨西哥草帽算子:



Canny边缘检测

• 算法步骤:

- 高斯滤波器平滑图像

- 用一阶偏导有限差分计算梯度幅值和方向

$$G_x[i, j] \approx (S[i, j+1] - S[i, j] + S[i+1, j+1] - S[i+1, j])/2$$

$$G_y[i, j] \approx (S[i, j] - S[i+1, j] + S[i, j+1] - S[i+1, j+1])/2$$

$$M[i, j] = \sqrt{G_x[i, j]^2 + G_y[i, j]^2}$$

$$\theta[i, j] = \arctan(G_y[i, j]/G_x[i, j])$$

- 对梯度幅值进行非极大值抑制

方向角离散化 (360°分为8个扇区, 对称扇区标号相同)

在离散后的梯度方向 ($\pm 45^\circ, \pm 90^\circ, \pm 135^\circ, 0^\circ, 180^\circ$) 上找到幅值最大的点保留, 其余点置零

- 用双阈值算法检测和连接边缘

取高低阈值T1,T2) ——阈值太低->假边缘; 阈值太高->轮廓丢失

得到高阈值图 $N[i, j] > T_2$, 低阈值图 $N[i, j] < T_1$

连接高阈值边缘; 出现断点时在低阈值边缘图中的八邻点域搜索边缘点

• 双阈值的意义和效果:

- 高阈值是将来提取轮廓的物体与背景区分开—阈值太高: 轮廓丢失
- 低阈值用来平滑边缘的轮廓—阈值太低: 假边缘
- 两个阈值区分强边界、弱边界, 更加有效

Chapter4-Feature 局部特征

Harris角点检测

• 原理

- 使用一个固定窗口在图像上进行任意方向上的滑动

- 比较滑动前与滑动后窗口中的像素灰度上的变化程度
 - 如果存在任意方向上的滑动，都有着较大灰度变化，那么可以认为该窗口中存在角点
- Flat:窗口内图像基本无变化
- Edge:沿Edge平移窗口时，窗口内图像无变化
- Corner:各方向平移时都有较大变化

• 公式推导

- **自相关函数**——描述窗平移后的相似度 $E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$

二元函数的一阶泰勒展开近似为 $f(x + u, y + v) \approx f(x, y) + u f_x(x, y) + v f_y(x, y)$

窗函数 $w(x, y)$ 一般取 $w = \begin{cases} 1 & \text{in window} \\ 0 & \text{else} \end{cases}$ 或高斯分布

(u, v) 代表平移量

平移量很小时，关于 (u, v) 对 E 泰勒展开，得到 $E(u, v) = [u \ v] \left(\sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$

- 推导过程

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

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

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

$$\begin{aligned} E(u, v) &= \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \\ &= \sum_{x,y} w(x, y) [I(x, y) + u I_x(x, y) + v I_y(x, y) - I(x, y)]^2 \\ &= \sum_{x,y} w(x, y) [u I_x(x, y) + v I_y(x, y)]^2 \\ &= [u, v] \left(\sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \end{aligned}$$

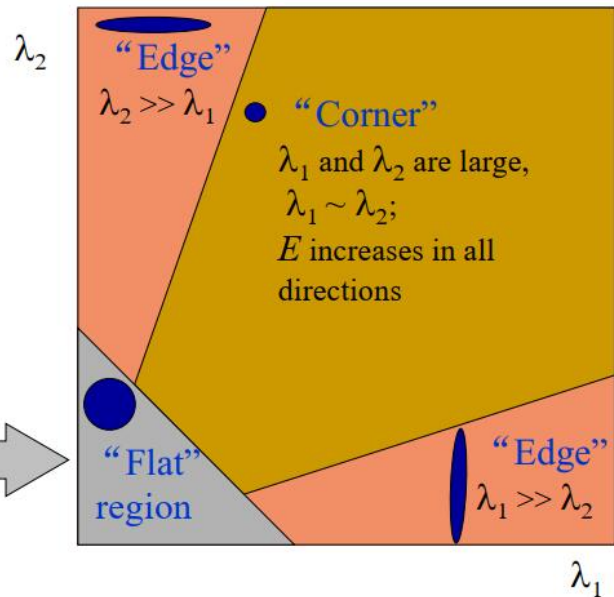
- λ_{max} 、 λ_{min} 的含义

λ_{max} 代表椭圆的长轴， λ_{min} 代表椭圆的短轴

通过 λ_{max} 、 λ_{min} 判断角点：

Classification of image points using eigenvalues of M :

λ_1 and λ_2 are small;
 E is almost constant in all directions



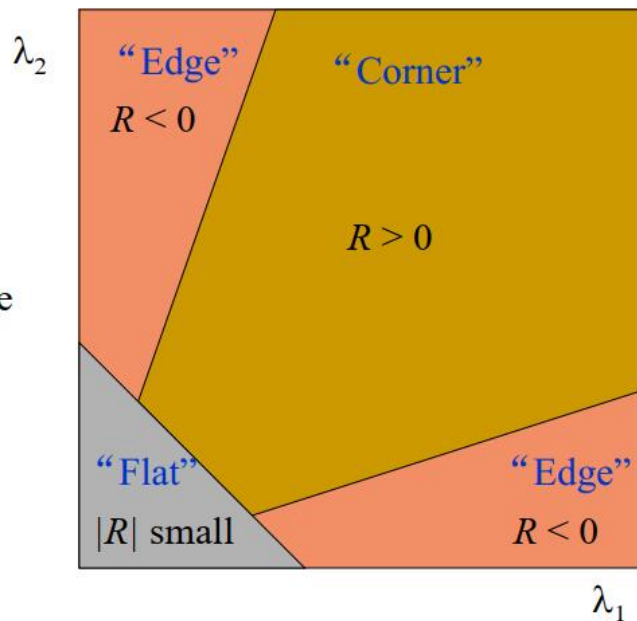
通过响应值函数 R 判断:

$$R = \det M - k(\text{trace} M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace} M = \lambda_1 + \lambda_2$$

- R depends only on eigenvalues of M
- R is large for a **corner**
- R is negative with large magnitude for an **edge**
- $|R|$ is small for a **flat** region



○ 算法过程

计算图像 R 值，并用合适阈值进行二值化
找到 R 值的极值点

○ 算法特点

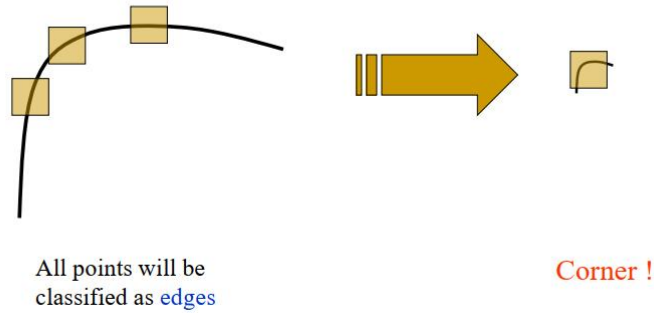
旋转不变性：椭圆可以旋转但是形状还是一样的

平移不变性

仿射强度变化的部分不变性：图像有偏置 ($I \rightarrow I + b$ 或 $I \rightarrow aI$) 时极值点不变

问题：无尺度不变性 \rightarrow Solution: Harris-Laplace / SIFT

- But: non-invariant to *image scale*!



一幅图像的尺度空间可被定义为原图像与可变尺度的高斯核 $G(x, y, \sigma)$ 卷积

SIFT描述子的计算

• 计算基本步骤

1. 构建尺度空间，建立图像金字塔
2. 寻找极值点（相邻的26个点中最大/最小值）
3. 去除不好的特征点：使用近似的harris corner，检测关键点的位置和尺度，并且去除边缘效应点
4. 用 16×16 放在特征点附近
5. 将 16×16 分成16个 4×4 的窗口
6. 计算窗口中每个像素的边的方向（梯度角减去 90° ）
7. 丢掉方向能量小的边（使用阈值）用直方图描述结果
8. 将每个小窗口的所有的方向离散成8个方向，一共 $16 \times 8 = 128$

• 为什么使用梯度信息,好处?

因为梯度信息可以表示边缘信息，并且在光照变化时有抵抗能力

• 不变性的解释

平移不变：SIFT是局部特征，只提取关键点附近矩形区域的sample，所以该物体移动到任何地方提取的feature都是类似的。同时因为是划grid去提取，即便关键点稍微偏离一些也是基本没有变化的。

旋转不变：旋转的时候周围点也会一起旋转，不会影响SIFT向量(在计算grid的梯度bin前需要旋转主方向，因此有了一定的旋转不变性)

光照不变：计算feature vector的时候进行了归一化、卡阈值之后再一次归一化，抵消了部分光照的影响。

尺度不变：金字塔模型，对每一种尺度都能进行检测，所以具有尺度不变（通过前一步计算LoG得到的尺度来确定计算feature的范围，所以不同尺度都能得到类似的feature）

• 尺度不变的原理

见上

Chapter5-Curve 曲线

Hough变换

- 解决什么问题

Hough变换是基于投票(Voting)原理的参数估计方法，是一种重要的形状检测技术

- 基本思想

图像中每一点对参数组合进行表决，赢得多数票的参数组合为胜者（结果）。

用极坐标来表示直线，从(x,y)转换到(p,θ)空间。

- 具体原理

- 基本步骤

- 适当地量化参数空间（合适的精度即可）
- 假定参数空间的每一个单元都是一个累加器，把累加器初始化为零
- 对图像空间的每一点，在其所满足的参数方程对应的累加器上加1
- 累加器阵列的最大值对应模型的参数

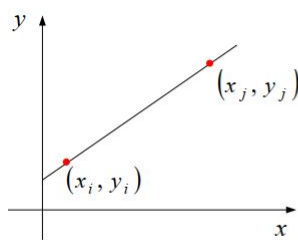
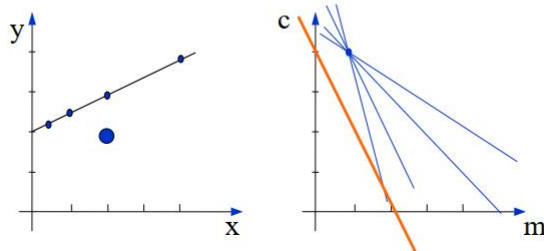
- 直线检测

直线检测：Hough变换

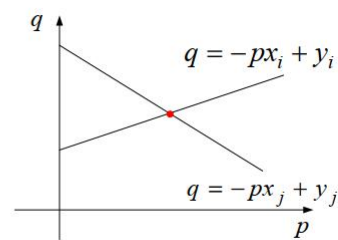
$$y = mx + c \rightarrow c = -xm + y$$

以(x, y)为自变量，(m, c)为因变量

每个点(x,y)对应于空间(m,c)上的一条直线



(a)

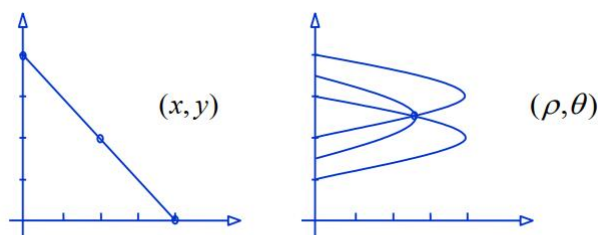


(b)

- 避免垂直直线所带来的问题，采用极坐标表示：

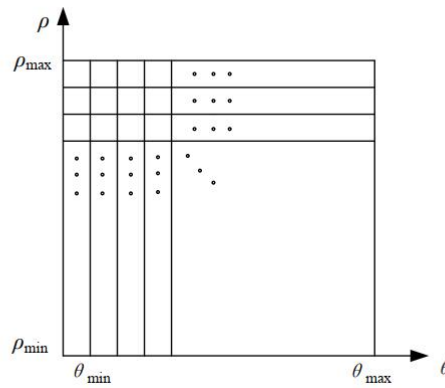
$$\rho = x \cos \theta + y \sin \theta$$

(x,y) 空间到 (ρ,θ) 空间的变换



- 实现：参数空间离散化，累加对应的参数组合

- 累加器



- 圆弧检测6部曲

圆弧检测：Hough变换

- 给定一个圆弧，它应该有三个参数：两个用来确定圆心，一个给出半径

$$x = a + r \cos \theta$$

$$y = b + r \sin \theta$$



$$a = x - r \cos \theta$$

$$b = y - r \sin \theta$$

令 θ 为边缘点处的梯度角



$$b = a \tan \theta - x \tan \theta + y$$



关于 a, b 的参数空间

浙江大学计算机学院

(仅供本课程内部使用)

3. 冬

- (1) 量化关于 a, b 的参数空间到合适精度
- (2) 初始化所有累加器为0
- (3) 计算图像空间中边缘点的梯度幅度 $G_{mag}(x, y)$ 和角度 $\theta(x, y)$
- (4) 若边缘点参数坐标满足 $b = a \tan \theta - x \tan \theta + y$ 则该参数坐标对应的累加器加1
- (5) 拥有最大值的累加器所在的坐标即为图像空间中的圆心之所在
- (6) 得到圆心坐标之后，我们可以很容易反求 r

Chapter6-Eigenface 人脸识别

PCA主元分析

1. 基本思想和原理

用于数据集降维，选择一个新的坐标系统进行线性降维，使得第一轴上是最大投影方向，第二轴上是第二大投影方向.....以此类推。

2. 什么样的数据用PCA会比较有效

3. 优化目标函数的推导

d -维空间 $x = (x_1, x_2, \dots, x_d)$

投影方向 $a_1 = (a_1^1, a_1^2, \dots, a_1^d)^T$ where $a_1^T a_1 = 1$

投影值 $z_1 = a_1^T x = \sum_{i=1}^d a_1^i x_i$

需要最大化 $var(z_1)$

求投影方向 $\operatorname{argmax} var(z_1)$

$$\begin{aligned}
 var(z_1) &= E(z_1^2) - [E(z_1)]^2 = E\left[\left(\sum_{i=1}^d a_1^i x_i\right)^2\right] - \left[E\left(\sum_{i=1}^d a_1^i x_i\right)\right]^2 \\
 &= \sum_{i,j=1}^d a_1^i a_1^j E(x_i x_j) - \sum_{i,j=1}^d a_1^i a_1^j E(x_i) E(x_j) \\
 &= \sum_{i,j=1}^d a_1^i a_1^j [E(x_i x_j) - E(x_i) E(x_j)] \\
 &= \sum_{i,j=1}^d a_1^i a_1^j S_{ij} \quad \text{令 } S_{ij} = E(x_i x_j) - E(x_i) E(x_j) \\
 &= \mathbf{a}_1^T \mathbf{S} \mathbf{a}_1 \quad \mathbf{S} ? \quad \operatorname{cov}(x_i, x_j) = E(x_i x_j) - E(x_i) E(x_j)
 \end{aligned}$$

其中 S 代表协方差矩阵，要在限制条件 $a_1^T a_1 = 1$ 最大化 $var(z_1)$,应用拉格朗日乘法

设 λ 为Lagrange乘子，则转为最大化

$$\mathbf{a}_1^T \mathbf{S} \mathbf{a}_1 - \lambda(\mathbf{a}_1^T \mathbf{a}_1 - 1)$$

求微分，得必要条件：

$$\mathbf{S} \mathbf{a}_1 - \lambda \mathbf{a}_1 = 0 \implies \mathbf{S} \mathbf{a}_1 = \lambda \mathbf{a}_1$$

经过计算，得到投影方向 a 是矩阵方向 S 最大特征根对应的特征向量。

Eigenface

1. 什么是Eigenface

使用特征脸进行人脸识别的方法

2. Eigenface人脸识别方法的基本步骤

- 预处理：根据人眼位置进行剪裁，进行灰度均衡化
- 将二维人脸图像按一行行向量排成一列，得到列图像；并把所有列图像拼起来，求出平均人脸
- 图像的协方差矩阵
- 求协方差矩阵的特征值，以及归一化的特征向量，即为特征人脸
- 算法流程
 - 对数据库中人脸图像作归一化处理
 - 用PCA计算得到一组特征脸（特征向量）
 - 计算数据库中每个人脸图像在该特征脸所张成的子空间上的坐标
 - 对每一输入图像，归一化后求解其在特征脸子空间中的坐标，并与库中人脸比较，验证相似性

3. 将重构用于人脸检测的原理

- 识别：将两张图像都投影到人脸空间，比较投影向量的欧氏距离
- 重构：将图像投影到人脸空间，通过左乘特征人脸空间矩阵恢复

对于任意待识别样本 f

- 识别

$$y_f = A^T f$$

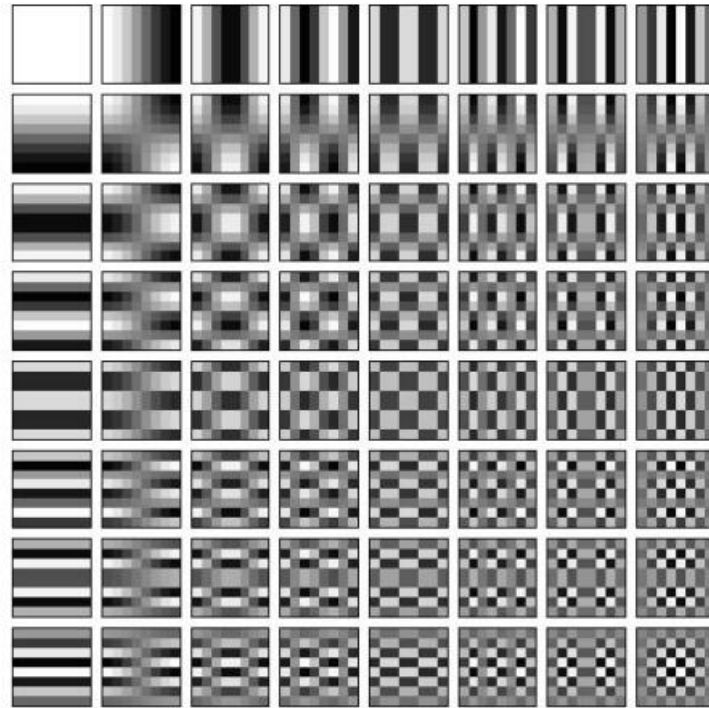
- 重构

$$\hat{f} = A y_f$$

Chapter7-Frequency 图像频域与图像分解

从图像分解角度，理解傅里叶变换的意义

- 变换：用正弦来表示，对于二维图像而言，由以下的基图像表示：



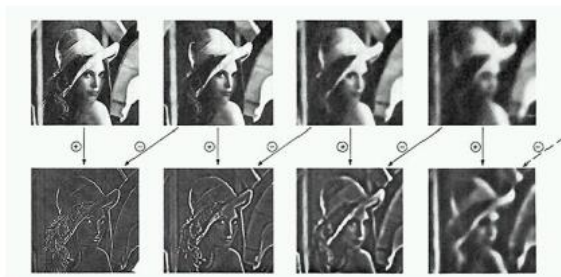
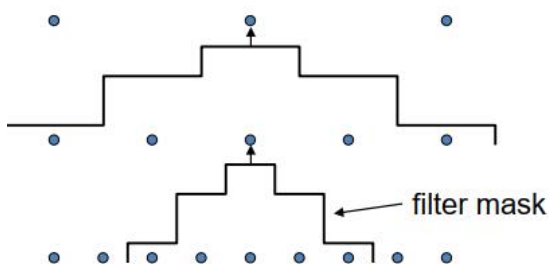
- 低频与高频：亮度灰度剧烈变化的地方是高频（图像边缘和轮廓的度量），对应边缘；变化不大的是低频（图像强度的综合度量），对应大片色块。近处看到的是高频分量，远处观察到的是低频分量。
- 对图像进行二维傅立叶变换得到频谱图，就是图像梯度的分布图。
如果频谱图中暗的点数更多，那么实际图像是比较柔和的（因为各点与邻域差异都不大，梯度相对较小）；如果频谱图中亮的点数多，那么实际图像一定是尖锐的，边界分明且边界两边像素差异较大的。
- 提供了从另一个空间（频域）去理解图像的方式。频域可以更加直观地理解为梯度或者是功率。

怎么理解拉普拉斯图像金字塔的每一层是带通滤波？

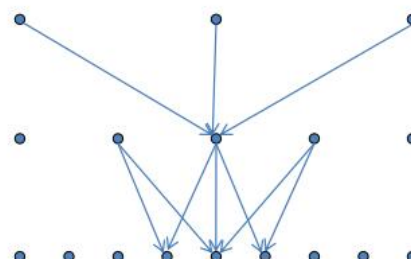
拉普拉斯金字塔是将图像下采样后再上采样得到的差值图像。

相减 保留细节 高通

下采样 降噪 低通



- “Gaussian” Pyramid
- “Laplacian” Pyramid
 - Created from Gaussian pyramid by subtraction
 - $L_i = G_i - \text{expand}(G_{i+1})$



Chapter8-Image Stitching 图像拼接

RANSAC

1. Generally speaking, 可以解决什么样的问题？

解决模型拟合的问题

2. 核心思想

- inliers (内群) 数据可以通过几组模型的参数来叙述其分布, 而outliers (离群) 数据则是不适合模型化的数据
- 数据会受到噪声的影响, 噪声指的是离群, 例如从极端的噪声或错误解释有关数据的测量或不正确的假设
- RANSAC假定, 给定一组 (通常很小) 的内存, 存在一个程序, 可以估算最适用于这一组数据模型的参数

3. 优缺点

- **优点:** 是大范围模型匹配问题的一个普遍意义上的方法, 且运用简单, 计算快
- **缺点:** 只能计算outliers比例不高的情况 (投票机制可以解决outliers高的情况)

4. 基本步骤

- 在数据中随机选择几个点设定为内群(hypothetical inliers)
- 计算拟合内群的模型
- 把刚才没选到的点带入刚才建立的模型中, 计算是否为内群

根据一些模型特定的损失函数, 符合估计模型的那些点被认为是内群的一部分

- 记下内群数量
- 重复以上步骤
- 比较哪次计算中内群的数量最多, 内群最多的那次所建的模型就是要求的解

5. outlier点比例给定的情况下, 则k次采样 (迭代) 后计算成功的概率是

w : inlier点的比例

n : 用于确定模型的样本个数

k : 采样个数

概率 : $1 - (1 - w^n)^k$

图像拼接

1. 实现两张图像自动拼接的基本步骤

1. 检测关键点
2. 建立SIFT描述子
3. SIFT特征匹配
4. 根据匹配的特征点对计算变换矩阵

- 变换矩阵 $T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$ —— 一组点对提供两个方程 \Rightarrow 需要至少三个点对
- 使用RANSAC提高求解准确度

5. 图像混合: 高斯金字塔 \rightarrow 拉普拉斯金字塔 \rightarrow 左右各一半 \rightarrow 上采样恢复图像

Chapter9&10- Visual Recognition&BoW 物体识别

Visual Recognition

1. 基本任务大概可以分为哪几大类

- 图片分类
- 检测和定位物体/图片分割
- 估计语义和几何属性
- 对人类活动和事件进行分类

2. 都有哪些挑战

- 视角变换
- 光线变化
- 尺度变化
- 物体遮挡
- 背景凌乱
- 内部类别多样

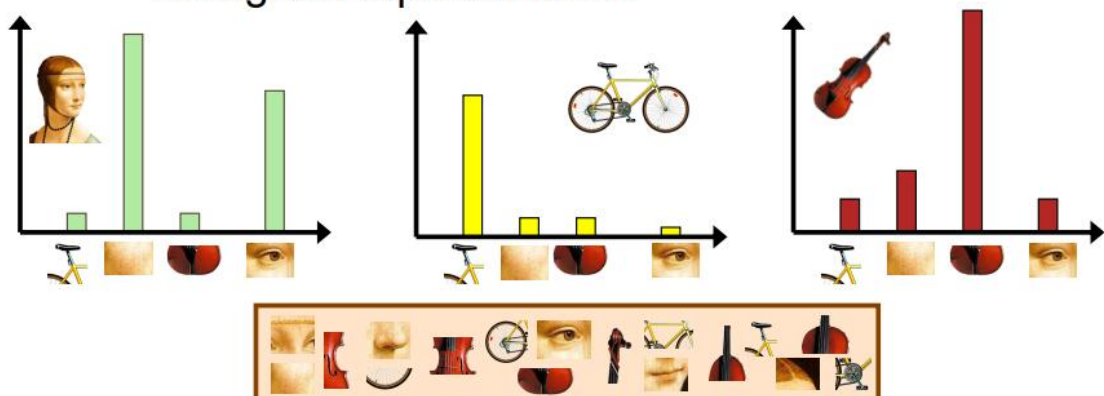
基于词袋(BoW)的物体分类

1. 图像的BoW(bag-of-words)是指什么意思

图像中的单词被定义为一个图像块的特征向量，图像的BoW模型即图像中所有图像块的特征向量得到的直方图

- Stricter definition

- Independent features
- histogram representation



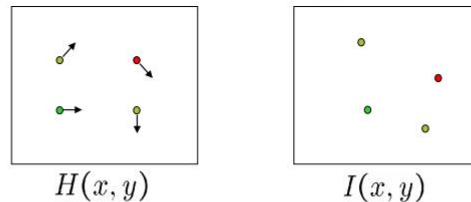
2. 基本步骤

- 特征提取与表示
- 通过训练样本聚类来建立字典
- 用字典的直方图来表达一张图像
- 根据bag-of-words来分类未知图像

Chapter12-Motion(Optical flow) 移动(光流)

光流解决的是什么问题

1. **定义:** optical flow is the apparent motion of brightness patterns in the image
apparent motion can be caused by lighting changes without any actual motion
2. 评估从H到I的像素运动，给出图像H中的一个像素，找到图像I中相同颜色的相近像素。解决的是像素对应问题。



- How to estimate pixel motion from image H to image I?
 - Solve pixel correspondence problem
 - given a pixel in H, look for nearby pixels of the same color in I
- Key assumptions
 - **brightness constancy**
 - **spatial coherence**
 - **small motion**
- This is called the **optical flow** problem

光流的三个基本假设是什么

1. **空间相干性 spatial coherence**
实际物体表面上不同两点在图片中具有一致的运动趋势。
同一子图像的像素点具有相同的运动
2. **细微运动 small motion**
3. **亮度恒定性 brightness constancy**
相邻帧之间的亮度恒定
$$I(x + u, y + v, t + 1) = I(x, y, t)$$

推导一个点的约束公式

$$\begin{aligned} I(x + u, y + v) &\approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v \\ 0 &= I(x + u, y + v) - I(x, y) \\ &\approx I(x, y) + I_x u + I_y v - I(x, y) \quad \text{shorthand: } I_x = \frac{\partial I}{\partial x} \\ &\approx (I(x, y) - I(x, y)) + I_x u + I_y v \\ &\approx I_t + I_x u + I_y v \\ &\approx I_t + \nabla I \cdot [u \ v] \end{aligned}$$

哪些位置光流比较可靠？为什么？

纹理复杂区域，因为梯度较大且方向不同，求出来的特征值比较大

提高鲁棒性

- 提高鲁棒性：用领域内的多个像素点计算光流——Lucas-Kanade算法
 - 使用5*5的窗，
$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} \Leftrightarrow (A^T A) d = A^T b$$
 - 可解性
 1. $A^T A = \sum \nabla I (\nabla I)^T$ 需可逆
 2. $A^T A$ 的特征值不能太小——防止被噪声干扰
 3. $A^T A$ 的特征值 $\lambda_1, \lambda_2 (\lambda_1 > \lambda_2)$ 需满足 $\frac{\lambda_1}{\lambda_2}$ 不能太大
- 使用技巧：尽量避免用边缘上的点计算光流——使用纹理复杂区域，梯度比较大且方向不同，求出来的特征值比较大

Chapter11-Deep Learning 深度学习

深度学习

1. 怎么理解被称为end-to-end的学习？

端到端的学习方式，从一端输入原始数据，另一端产生输出结果。模型的输出就是我们想要的结果。而模型如何具体实现并不重要，类似于一个黑箱。端到端的网络的出现使得特征可以自己去学习，不需要像之前一样认为干预了。

2. 神经网络的学习，数学本质上是求解什么？常用的基本方法？

求解一个函数，这个函数利用矩阵的线性变换加上激活函数的非线性变换，将原始输入空间投向线性可分、稀疏的空间去分类、回归。

基本方法：激活函数、矩阵变换、卷积处理、损失函数等

CNN

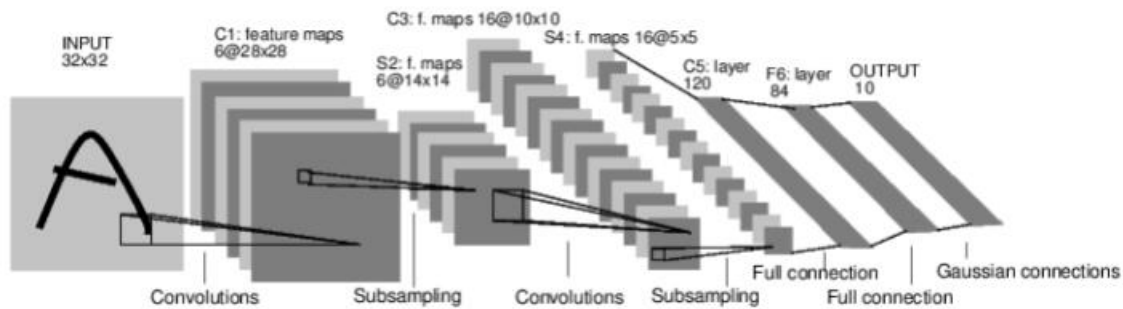
1. 理解卷积层与Pooling层的作用

- 卷积层：构建卷积神经网络的核心层，产生了网络中的大部分计算量。卷积的意义在于从数据中提取出特征。
- pooling层：逐渐降低数据体的空间尺寸，这样可以减少网络中的参数数量，使得计算资源耗费变少，也能有效减少过拟合现象

单图池化：减少参数数量 & 获得更大的感受野 & 使特征对微小变换更鲁棒 & 图像平滑作用

多图间池化：减少参数数量 & 找到多图间最显著的特征

2. 会计算第一个卷积层的各种weight个数



[LeNet-5, LeCun 1980]

参数计算：

- 输出图像大小计算:记输出为 的feature map，输入为 ，卷积核为 ，步长为 S ，padding为 P ，则有 $M = (N - K + 2P) / s + 1$
- 权重个数和神经元数目计算：
 1. 每一次 区域内的卷积都对应一个神经元，因此每个神经元有 k^2 个权重，1个偏置——多通道时权重要乘上通道数 图中第一层权重个数为 $(5 \times 5 + 1) \times 6 = 156$ 个参数
 2. 总神经元数为 M （输出图像大小），连接数为 $(K^2 + 1) \times M$ （此处未考虑多通道）——一般使用多个卷积核，则神经元个数乘上卷积核数 图中c1层共有 $(5 \times 5 + 1) \times 6 \times (28 \times 28) = 122,304$ 个连接
- 更新的参数个数：CNN中一个卷积核下对应的所有神经元共享参数，因此需要更新的参数个数为 $(k^2 + 1) \times \text{通道数} \times \text{卷积核数}$

BP算法

1. 知道BP算法的作用

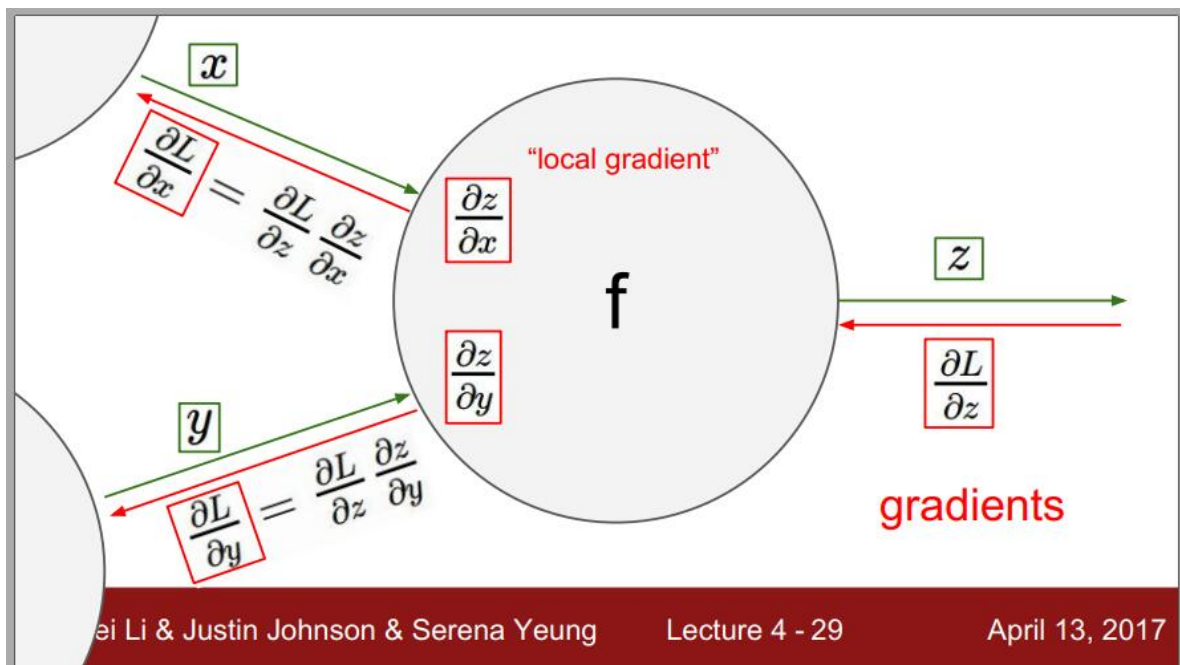
BP网络能学习和存储大量的输入-输出模式映射关系，而无需事先描述这种映射关系的数学方程。它的学习规则是使用梯度下降法，通过反向传播来不断调整网络的权值和阈值，使网络的误差平方和最小。

BP神经网络模型拓扑结构包括输入层、隐层和输出层

2. 理解“梯度下降法”与BP算法的关系

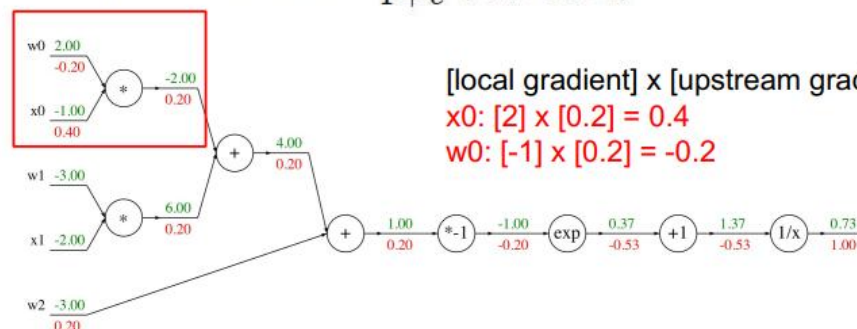
BP算法中需要用到梯度下降法，用来配合反向传播，BP算法就是提供了给梯度下降法所需要的所有值。梯度下降法是求局部最好的权重 w 。

3. 给一个具体例子，会计算梯度反向传播的过程



Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2 x_2)}}$$



[local gradient] x [upstream gradient]

$$x_0: [2] \times [0.2] = 0.4$$

$$w_0: [-1] \times [0.2] = -0.2$$

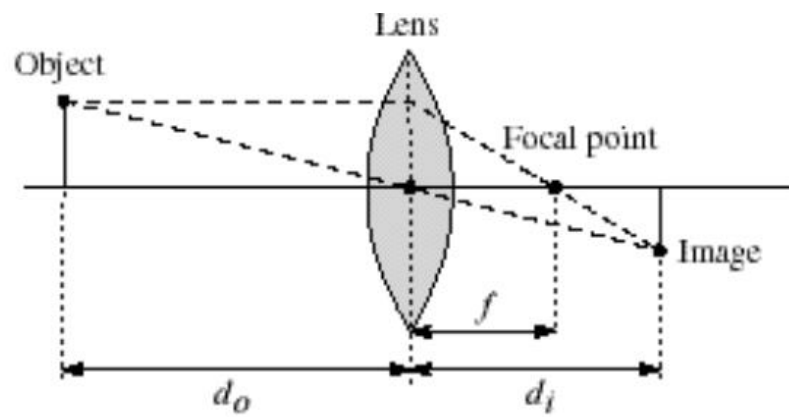
$$\begin{array}{lcl} f(x) = e^x & \rightarrow & \frac{df}{dx} = e^x \\ f_a(x) = ax & \rightarrow & \frac{df}{dx} = a \end{array} \quad \left| \quad \begin{array}{lcl} f(x) = \frac{1}{x} & \rightarrow & \frac{df}{dx} = -1/x^2 \\ f_c(x) = c + x & \rightarrow & \frac{df}{dx} = 1 \end{array} \right.$$

Chapter13-Camera 相机

基本概念——理解景深/光圈/焦距/视场

1. 景深？视场？焦距？视场？

- 景深：相机镜头能够取得清晰图像的成像所测定的被摄物体前后范围距离
- 光圈：镜头中用于控制光线透过镜头并进入机身内感光面光量的装置
- 焦距：从镜片中心到底片等成像平面的距离
- 视场：镜头能够观察到的最大范围

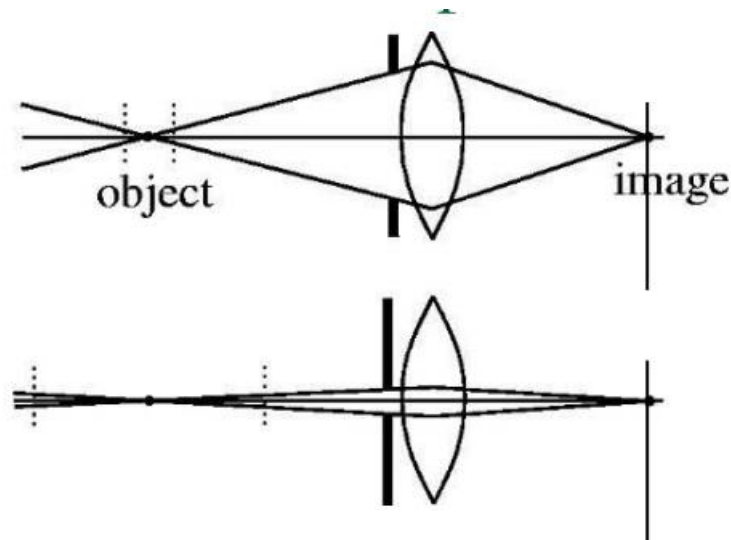


Thin lens equation:

$$\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}$$

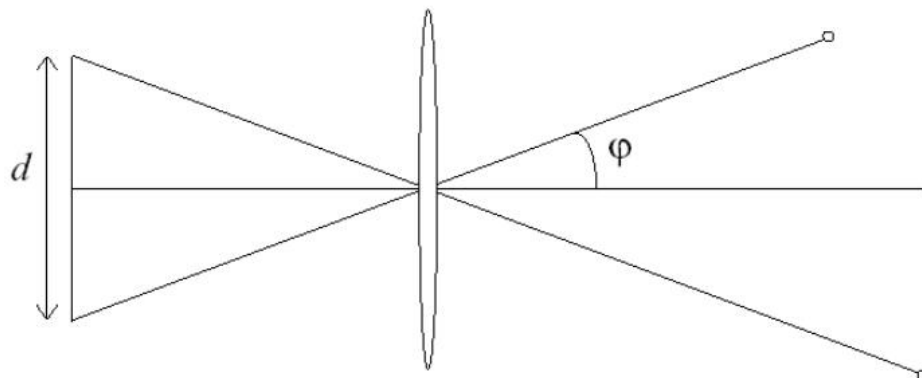
1. 光圈对景深的影响？理解原理

小光圈 → 大景深，大光圈 → 小景深



2. 焦距对视场的影响？理解原理

短焦距 → 大视场角: $\varphi = \arctan \frac{d}{2f}$



Size of field of view governed by size of the camera retina:

$$\varphi = \tan^{-1}\left(\frac{d}{2f}\right)$$

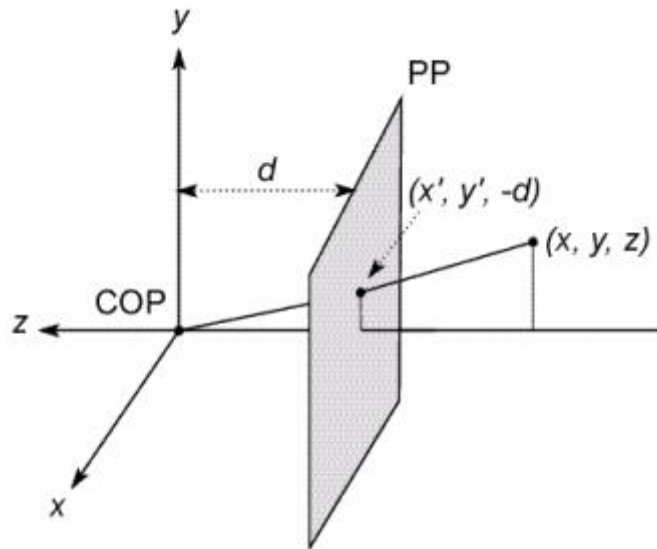
大焦距 离得近：整个场景被缩短，远处的东西被拉到近处而且很大，但是虚化了焦距内的物体也能看到

小焦距 离得远：整个场景被拉长，远处的东西很小，但是都很清楚，焦距内的东西会在视野之外

理想的针孔相机模型(pinhole camera)

1. 基本投影公式

相机坐标系 → 成像坐标系(3D TO 2D)



根据相似三角形：

$$(x', y', z') \rightarrow (-d \frac{x}{z}, -d \frac{y}{z}, -d)$$

2. 推导齐次坐标形式下的透视投影公式(矩阵形式)

z不是齐次的，那就转化成齐次的：

- Is this a linear transformation? $(x, y, z) \rightarrow (-d \frac{x}{z}, -d \frac{y}{z})$

□ no—division by z is nonlinear

- Trick: add one more coordinate:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image
coordinates

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

homogeneous scene
coordinates

- Converting *from* homogeneous coordinates

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

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

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

3. 齐次坐标表示的好处

齐次坐标缩放一个常量因子是不变的，齐次坐标升维后，使得之前的平移运算（矩阵加法）也变成了矩阵乘法，这样旋转平移缩放变换都只要进行矩阵乘法运算就可以了。

4. 相机模型有哪几个内参(不包括畸变参数)，会写内参矩阵

4个 f_x, f_y 和相机的焦距、像素大小有关； c_x, c_y 是平移的距离，和相机成像平面的大小有关

成像坐标系 → 像素坐标系(2D TO 2D): 考虑度量单位 (mm pixel) 的不同，以及两坐标系原点不同，将一个成像平面的坐标 (x, y) 在水平上缩放 α 倍，在竖直方向上缩放 β 倍，同时平移 (c_x, c_y) ，就可以得到像素坐标系的坐标 (μ, ν)

$$\begin{cases} \mu = \alpha \cdot x + c_x \\ \nu = \beta \cdot y + c_y \\ x = f \frac{X}{Z} \\ y = f \frac{Y}{Z} \end{cases}$$

得到

$$\begin{cases} \mu = \alpha \cdot f \frac{X}{Z} + c_x \\ \nu = \beta \cdot f \frac{Y}{Z} + c_y \\ \text{其中 } f_x = \alpha \cdot f, f_y = \beta \cdot f \end{cases}$$

写成齐次坐标系：

$$\begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

坐标，缩放一个常量因子仍然是相等的，将 Z 挪到左边

$$\begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix} = Z \begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

通过上面的推导，就得到了相机的**内参数矩阵 (Camera Intrinsics) K** ,

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

畸变

1. 径向畸变与切向畸变各是什么原因引起的？

- 径向畸变：原因是光线在远离透镜中心的地方比靠近中心的地方更加弯曲
- 切向畸变：原因是透镜不完全平行于图像平面

2. 径向畸变常见的有哪两种？

- 桶形畸变：中间向外凸起
- 枕形畸变：中间向内凹陷

外参矩阵

世界坐标系→相机坐标系(3D TO 3D)

$$P_c = RP_w + t$$

P_c 是 P 在相机坐标系的坐标, P_w 是其在世界坐标系下的坐标, 可以使用旋转矩阵 $R(3 \times 3)$ 和一个平移向量 $t(3 \times 1)$ 将 P_c 转为 P_w

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

其齐次坐标的形式

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

将旋转矩阵 R 和平移向量 t 带入

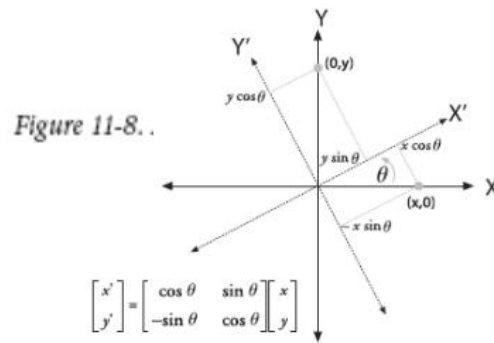
$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

上面就推导得到相机的外参数 (Camera Extrinsics) T

$$T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$$

其中, R 是旋转矩阵, t 是平移向量.

Rotation and Translation



$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{bmatrix} \quad R_y(\varphi) = \begin{bmatrix} \cos \varphi & 0 & -\sin \varphi \\ 0 & 1 & 0 \\ \sin \varphi & 0 & \cos \varphi \end{bmatrix} \quad R_z(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P' = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{t}_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix} P \quad R = R_z(\theta), R_y(\varphi), R_x(\psi)$$

$$\mathbf{t}_{3 \times 1} = (t_x, t_y, t_z)'$$

Called **extrinsic parameters**

$$(\theta, \varphi, \psi, t_x, t_y, t_z)$$

Called **intrinsic parameters**

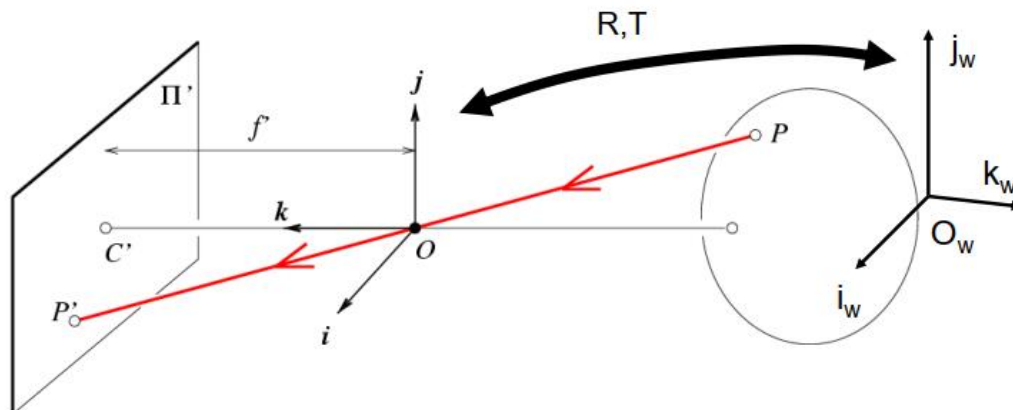
$$(f_x, f_y, c_x, c_y)$$

Called **distortion parameters**

$$(k_1, k_2, p_1, p_2, k_3)$$

画图展示所有过程

Projection



$$\mathbf{x} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \mathbf{X}$$

\mathbf{x} : Image coordinates: $(u, v, 1)$

\mathbf{K} : Intrinsic matrix (3×3)

\mathbf{R} : Rotation (3×3)

\mathbf{t} : Translation (3×1)

\mathbf{X} : World coordinates: $(X, Y, Z, 1)$

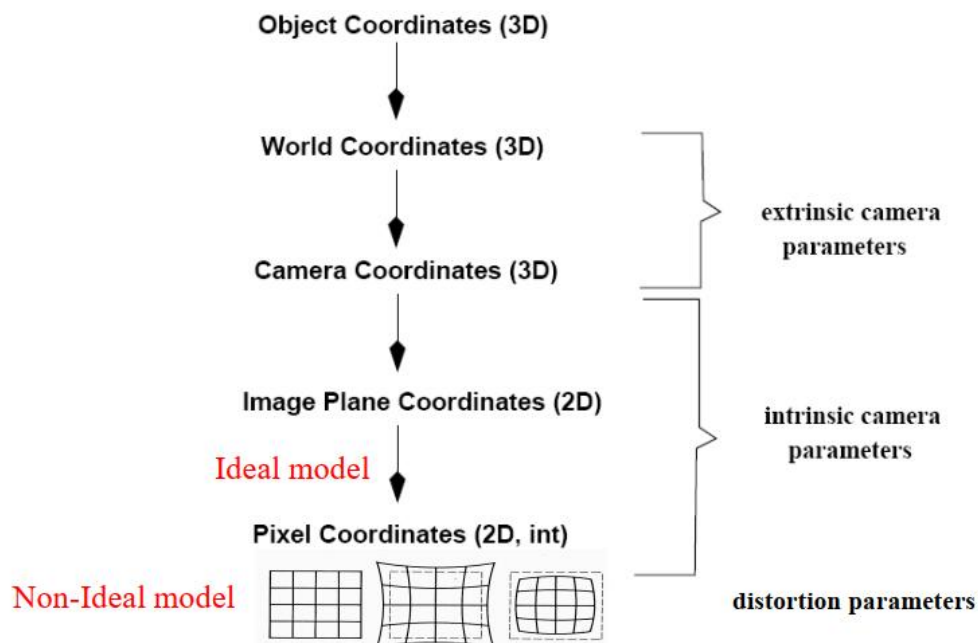
- 世界坐标系 \Rightarrow 相机坐标系: 外参数
- 相机坐标系 \Rightarrow 图像坐标系 \Rightarrow 像平面坐标系(2D): 内参数
- 非理想模型:

像素 \Rightarrow 畸变参数

- k 和径向畸变有关
- p 和切向畸变有关

畸变参数有五个 k_1, k_2, k_3 为径向畸变系数, p_1, p_2 是切向畸变系数

Camera parameters



Chapter14-Camera Calibration 相机定标

一般的相机定标

1. 需要求解哪些参数

求解内外参(3D) 畸变参数(2D)

2. 解决这个问题的基本思路是什么

通过最小二乘法进行求解, 类似解方程

• Linear Least-Squares Methods(线性最小二乘法)

— A system of p linear equations in q unknowns:

$$\begin{cases} u_{11}x_1 + u_{12}x_2 + \dots + u_{1q}x_q = y_1 \\ u_{21}x_1 + u_{22}x_2 + \dots + u_{2q}x_q = y_2 \\ \dots \\ u_{p1}x_1 + u_{p2}x_2 + \dots + u_{pq}x_q = y_p \end{cases} \Leftrightarrow \mathbf{U}\mathbf{x} = \mathbf{y}$$
$$\mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1q} \\ u_{21} & u_{22} & \dots & u_{2q} \\ \dots & \dots & \dots & \dots \\ u_{p1} & u_{p2} & \dots & u_{pq} \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_q \end{pmatrix} \quad \text{and} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_q \end{pmatrix}$$

when $p < q$, the set of solutions to this equation forms a $(q - p)$ -dimensional vector subspace of \mathbb{R}^q

when $p = q$, there is unique solution

when $p > q$, there is no solution

We focus on the overconstrained case $p > q$ and assumes that \mathbf{U} has maximal rank q

基于Homography的相机定标

1. 有哪些优点

- Only require the camera to observe a planar pattern shown at a few (minimum 2) different orientations
- Pattern can be printed and attached on planer surface
- Either camera or planar pattern can be moved by hand
- More flexible and robust than traditional techniques
 - Easy setup
 - Anyone can make calibration pattern
- Considered flexibility, robustness, and low cost

2. 已知什么, 求解什么

- **已知** 给定标定物体的N个角点, K个视角(棋盘格子两个点可以得出四个等式)
- **求解** 所有的参数。N个点K个视角可以列出 $2NK$ 个等式, 会带来 $6K+4$ 个参数。需要 $2NK \geq 6K+4$

3. 简述基本过程(4个步骤)

- 获取标定物体网格的角点在坐标系的位置
- 找到图片的角点
- 根据图像空间坐标系到世界坐标系建立等式
- 求解相机参数

4. Homography矩阵有几个自由度? 求解需要至少几个特征点?

8个自由度

求解最少需要4个特征点

Chapter15-Stereo Vision 立体视觉

立体视觉的三角测量基本原理 (Triangulation公式)

1. 会画“视差disparity”

$$\frac{T - (x^l - x^r)}{Z - f} = \frac{T}{Z} \Rightarrow Z = \frac{fT}{x^l - x^r} \quad \text{Triangulation}$$

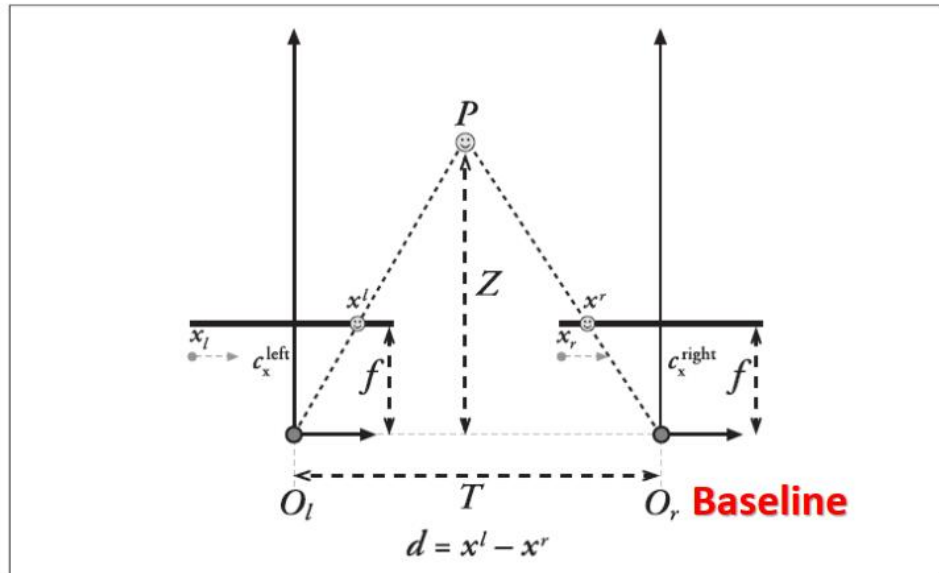


Figure 12-4. With a perfectly undistorted, aligned image pair and known correspondence, the depth Z can be found by similar triangles on the projected lines of the image pair at the constant of magnification.

2. 并能推导公式

$$\frac{T - (x^l - x^r)}{Z - f} = \frac{T}{Z} \Rightarrow Z = \frac{fT}{x^l - x^r}$$

立体视觉的基本步骤

1. 消除畸变：根据畸变模型消除畸变
2. 矫正相机
 - 原因：根据对极几何，左右观测对于同一物体的投影处在同一水平线上
 - 目的：使左右观测所得图像行对齐
3. 两幅图中找到相同特征
4. 三角测量，得到深度图

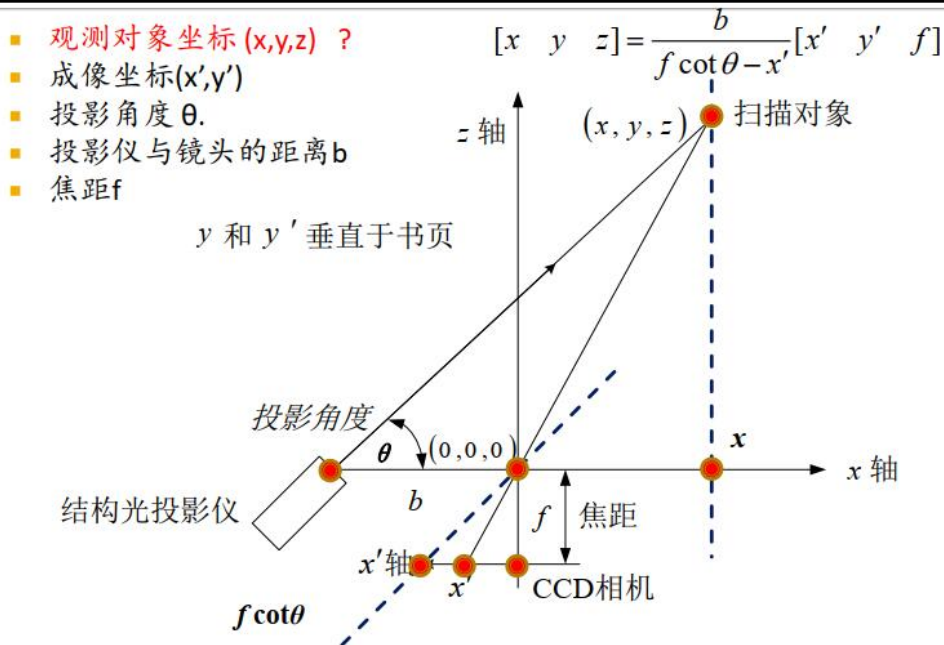
Chapter16-Structured Lighting3D Scan 结构光三维成像

结构光成像系统的构成

结构光投影仪+CCD相机+深度信息重建系统

利用结构光获取三维数据的基本原理

1. 会画图



2. 推导公式

由相似三角形有

$$\begin{cases} \frac{z}{x} = \frac{f}{x'} \\ \frac{b+x}{z} = \cot \theta \end{cases} \Rightarrow [x \ y \ z] = \frac{b}{f \cot \theta - x'} [x' \ y' \ z']$$

ICP算法(Iterative Closest Point)迭代最近点方法

1. 要解决什么问题

用于多个摄像机的配准问题，即把多个扫描结果拼接在一起形成对扫描对象的完整描述

2. 基本步骤

给定两个三维点集 X 与 Y ，将 Y 配准到 X ：

- 计算 Y 中每一点在 X 中的对应最近点
- 求使得上述对应点对的平均距离最小的刚体变换，获得刚体的变换参数(平移参数与旋转参数)，类似于变换矩阵。
- 对 Y 应用上一步求得的刚体变换(平移与旋转)，更新 Y
- 如果 X 与 Y 的对应点对平均距离大于阈值，回到第一步，否则停止计算

这样寻找配准函数 F 的过程就变成了找到使得Cost最小的点的搜索过程。

Chapter17-Segmentation 图像分割

基于k-means聚类的图像分割

1. 理解用聚类进行图像分割的基本原理

基于区域间的不连续性（不同区域间）和相似性（同一区域内）

2. 给定图像，描述使用k-means进行分割的所有步骤

- 将图像转换到特征空间，例如灰度空间、HSV空间、坐标值&灰度值、坐标值&RGB值
- 随机选择k个聚类中心 c^0
- 对图像上所有点，根据其与聚类中心的距离，将其划分为距离最近对应的中心的聚类簇
- 重新计算每一簇新的中心(一般取当前类内所有样本在每一维度上的均值)
- 重复3、4两步直到没有点被重新分配
- 退出迭代后，将每一簇中的所有点赋予簇中心的类别标记
- 遍历图像中的点，根据其的类别标记对图像进行分割

基于Mean Shift的图像分割

1. 基本原理

一般是指一个迭代的步骤，即先算出当前点的偏移均值，移动该点到其偏移均值，然后以此为新的起始点，继续移动，直到满足一定的条件结束。

2. 基本思路

- 选择一点为圆心，选定一个半径画圆
- 获得落在圆内的其他点离圆心的方向向量
- 求向量的和，获得质心转移向量（新向量的终点为新的质心）
- 如果质心转移向量够小，就认为找到正确的质心，重新选择圆心；否则对新的质心重复操作

- The mean shift algorithm *seeks modes of the given set of points*

1. Choose *kernel* and *bandwidth*

2. For each point:

- a) Center a *window* on that point
- b) Compute the *mean* of the data in the search window
- c) Center the search window at the *new* mean location
- d) Repeat (b,c) until *convergence*

3. Assign points that lead to nearby modes to the same cluster

3. 跟k-mean图像分割相比，有什么好处

Mean shift pros and cons

- Pros
 - Good general-purpose segmentation
 - Flexible in number and shape of regions
 - Robust to outliers
 - Cons
 - Have to choose kernel size in advance
 - Not suitable for high-dimensional features
 - When to use it
 - Oversegmentation
 - Multiple segmentations
 - Tracking, clustering, filtering applications
- 良好通用的segmentation
 - 在数量和区域形状上具有灵活性
 - 对异常值具有鲁棒性