

计算机视觉复习笔记

目录

- [引言](#)
 - [二值图像](#)
 - [边缘](#)
 - [曲线与图像频域](#)
 - [局部特征 Local Feature](#)
 - [图像拼接 Image Stitching](#)
 - [光流 Optical Flow](#)
 - [图像分割](#)
 - [相机模型](#)
 - [相机定标 Camera Calibration](#)
 - [立体视觉](#)
 - [三维数据获取](#)
 - [人脸识别](#)
 - [物体识别](#)
 - [物体识别-CNN](#)
-

引言

计算机视觉五大研究内容

1. **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。
2. **低层视觉 (low level/early)** 主要是对输入的原始图像进行处理。这一过程借用了大量的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，以便从图像中抽取诸如角点、边缘、线条、边界、色彩、纹理、运动等关于场景的基本特征。
3. **中层视觉 (middle level)** 主要任务是恢复场景的深度、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 (stereovision)、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、系统成像模型等研究内容一般也是在这个层次上进行的。分割、拟合等
4. **高层视觉 (high level)** 主要任务是在以物体为中心的坐标系中，在原始输入图像、图像基本特征、2.5维图的基础上，恢复物体的完整三维图，建立物体三维描述，识别物体并确定物体的位置和方向
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。

Gestalt Laws

1. **Law of Proximity (接近原理)**：人们自然而然的对靠近的事物进行分组

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成一组
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 (对称原理)**：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

视觉表示框架的三个阶段

第一阶段(Primal 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)：在以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。

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(如圆)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

$$\chi^2 = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} r_{ij}^2 B[i, j]$$

4. 伸长率

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

5. 密集度:

- p是周长, A是面积
- 给定周长, 密集度越高, 围成的面积就越大: 圆 > 正方形 > 长方形

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

6. 形态比: 区域的最小外接矩形的长宽比

7. 欧拉数: 亏格数(genus), 连通分量减去洞数 $E = C - H$

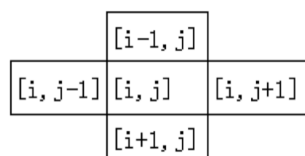
投影计算

水平投影: 计算每一列像素为1的个数。

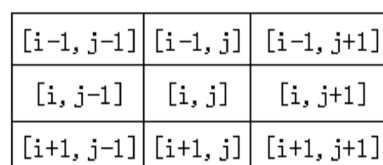
垂直投影: 计算每一行像素为1的个数。

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。

连通区域



四连通邻点



八连通邻点

递归算法

1. 扫描图像, 找到没有标记的一个前景点 (即像素值为1/黑的/和背景的黑块相对), 分配标记L
2. 递归分配标记L给该点的邻点
3. 如果不存在没有标记的点, 则停止
4. 返回第1步

序贯算法(for 4连通)

1. 从左到右、从上到下扫描图像
2. 如果像素点值为1，则（分4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

边界跟踪算法

1. 从左到右，从上到下扫描图像，求区域S的起始点
2. 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
3. 按逆时针方向记从b开始的c的8个8邻点分别为
4. 从b开始，沿逆时针方向找到第一个 n_i in S
5. 置 $c = s(k) = n_i$, $b = n_{i-1}$
6. 重复步骤 3, 4, 5，直到 $s(k)=s(0)$

边缘

Origin of Edges

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

先确定图像中的边缘像素，然后再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，一阶导数的局部极大值，或二阶导数的过零点。

基于一阶的边缘检测

Roberts交叉算子，Sobel算子，Prewitt算子（运算较快）

基于二阶的边缘检测

Laplacian算子, 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

- 图像与高斯函数卷积, 再求卷积的拉普拉斯微分
- 求高斯函数的拉普拉斯微分, 再与图像卷积

使用高斯滤波器的原因 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的平衡

Canny 边缘检测器

算法步骤:

1. 用高斯滤波器平滑图像
2. 用一阶偏导有限差分计算梯度幅值和方向
3. 对梯度幅值进行非极大值抑制 (NMS)
4. 用双阈值算法检测和连接边缘

双阈值的意义和效果

- 高的阈值是将要提取轮廓的物体与背景区分开来 —**阈值太高: 部分轮廓丢失.**
- 低的阈值是用来平滑边缘的轮廓 —**阈值太低: 假边缘;**
- 两个阈值: T_1, T_2 。大于 T_1 的称为强边界。 T_1 和 T_2 之间的为弱边界 —**更有效的阈值方案**

曲线与图像频域

Hough 变换

Hough 变换是基于投票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: 图像中的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 (ρ, θ) 空间。

步骤:

1. 适当地量化参数空间 (合适的精度即可)

2. 假定参数空间每一个单元都是一个累加器，把累加器初始化为 0
3. 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
4. 累加器阵列最大值对应模型的参数

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。

怎么理解拉普拉斯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
- 高通减低通

$$\begin{cases} LP_l = G_l - G_{l+1}, & \text{当 } 0 \leq l < N \text{ 时} \\ LP_N = G_N, & \text{当 } l = N \text{ 时} \end{cases}$$

N为拉普拉斯金字塔顶层的层号，LP_l是拉普拉斯金字塔分解的第L层图像。由LP₀，LP₁、LP₂... LP_N构成的金字塔即为拉普拉斯金字塔。它的每一层LP_l图像是高斯金字塔本层G_l图像与其高一层图像G_{l+1}经内插放大后图像G_{l+1}的差，此过程相当于带通滤波，因此拉普拉斯金字塔又称为带通金字塔分解。

局部特征 Local Feature

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，那么我们可以认为该窗口中存在角点。

公式推导

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

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

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

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

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

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

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

- $R > 0$ （大于某一阈值），则为角点；
- $R < 0$ ，则为边；
- R 绝对值很小，则为平面区域。

选取 R 得到的符合条件点的局部最大值作为结果。

SIFT描述子 Descriptor

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，并且去除边缘响应点。
4. 用 16x16 的窗口放在特征点附近
5. 将 16x16 分成 16 个 4x4 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. 将每个小窗口中的所有方向离散成 8 个方向，一共 16x8=128 个

为什么使用梯度信息而不直接用像素值

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

各种不变性的解释

1. **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，即便关键点稍微偏移一下feature也基本没有变化，有点类似于HOG或者CNN的pooling。
2. **旋转不变**：旋转的时候每一个关键点周围的点也会跟着旋转，不会影响SIFT向量。所以SIFT对旋转不敏感（在计算grid里面的梯度bin前需要旋转到主方向，因此有了一定的旋转不变性）
3. **光照不变**：计算feature vector的时候进行了归一化、卡阈值之后又一次归一化，抵消了部分光照的影响。

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

图像拼接 Image Stitching

实现两张图像自动拼接的基本步骤

找出两张图片中相似的点 (至少四个，因为 homography 矩阵的计算需要至少四个点)，然后计算一张图片可以变换到另一张图片的变换矩阵 (homography 单应性矩阵)，用这个矩阵把那张图片变换后放到另一张图片相应的位置 (就是相当于把两张图片中定好的四个相似的点给重合在一起)。如此，就可以实现简单的全景拼接。当然，因为拼合之后图片会重叠在一起，所以需要重新计算图片重叠部分的像素值，否则结果会很难看。

总结：

1. ~~找两张图片中相似的点，计算变换矩阵~~
2. 变换一张图片放到另一张图片合适的位置，并计算重叠区域新的像素值 (这里就是图片融合所需要采取的策略)

RANSAC

核心思想（基本假设）

- “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。

基本步骤

1. 在数据中随机选择几个点设定为内群 (hypothetical inliers)
2. 计算拟合内群的模型。
3. 把其他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(model specific loss function)，符合估计模型的那些点被认为是内群(consensus set)的一部分

4. 记下内群数量。
5. 重复以上步骤多做几次。
6. 比较哪次计算中内群数量最多，内群最多的那次所建的模型就是所要求的解。

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不多的情况（投票机制可以解决outliers高的情况）

光流 Optical Flow

光流解决的是什么问题？

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。

光流三个基本假设是什么？

- 亮度恒定性 brightness constancy
$$I(x+u, y+v, t+1) = I(x, y, t)$$
- 空间相干性 spatial coherence
- 细微运动 small motion

哪些位置的光流比较可靠，为什么

纹理复杂区域，因为梯度比较大且方向不同，求出来的特征值比较大

推导一个点的约束公式

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

图像分割

基于K-MEANS聚类的图像分割

1. 任意选择 k 个 sift 特征点作为初始聚类质心。
2. 对于每个 sift 特征点，计算它们与 k 个聚类质心的欧式距离，找到最小的那个聚类质心，将该特征点放入此聚类质心集合中。
3. 对于每个聚类质心集合，用所有元素均值来更新质心。
4. 比较更新前后聚类质心集合，满足 No points are re-assigned，否则返回 2，如果迭代次数太多聚类失败。

基于Mean Shift图像分割的基本原理与基本思路

基本原理

Mean Shift算法,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。

基本思路

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

相机模型

景深/光圈/焦距/视场

光圈对景深的影响

大光圈景深小, 小光圈景深大 光路图里把上下两条线放近一点

焦距对视场的影响

$$\varphi = \tan^{-1}(d/2f)$$

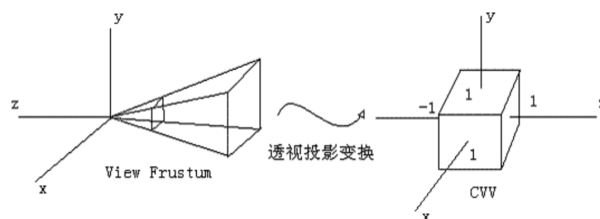
(φ 是视角的二分之一)

- 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外

总结: 焦距越大, 视场越小

理想针孔相机 (pinhole camera) 模型

齐次坐标形式下的透视投影图和公式 (矩阵形式)



$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{d} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ -\frac{z}{d} \end{pmatrix}$$

$$\left(-d\frac{x}{z}, -d\frac{y}{z}\right)$$

内参

4个内参: (f_x, f_y, c_x, c_y)

内参矩阵:

$$M = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

畸变

径向畸变和切向畸变

1. **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
 - 桶形畸变: 中间向外凸起
 - 枕形畸变: 中间向内凹陷
2. **切向畸变**: 原因是透镜不完全平行于图像平面

外参有哪些? 分别代表什么意义

(每拍一张照片, 就有一组外参)

刚体变换可用6个参数来描述, 这6个参数就称为相机的外参(Extrinsic), 相机外参决定了空间点从世界坐标系转换到相机坐标系的变换, 也可以说外参描述了相机在世界坐标系中的位置和朝向。

这六个参数为: 3个旋转参数, 3个位移参数

- 三个轴的旋转参数分别为 (ω, δ, θ) , 然后把每个轴的 3×3 旋转矩阵进行组合 (即先矩阵之间相乘), 得到集合三个轴旋转信息的 R , 其大小还是 3×3 ;
- T 的三个轴的位移参数 (T_x, T_y, T_z) 。
- R, T 组合成成的 3×4 的矩阵, 其是转换到标定纸坐标的关键。其中绕X轴旋转 θ

外参矩阵

$$P = \begin{pmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{pmatrix} P_0$$

三者在校准各阶段中的角色

(从三维物体到真实图像的过程)

- 世界坐标系 => 相机坐标系：外参数
- 相机坐标系 => 像平面坐标系(2D)：内参数
- 非理想模型：=> 像素：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 为切向畸变系数)

相机定标 Camera Calibration

相机定标要求解哪些参数

Given:

N correspondences b/w scene (黑白场景) and images

Recover the camera parameters (恢复相机要求的参数) :

- Distortion coefficients (畸变参数)
- Intrinsic para (内参)
- Extrinsic para (外参)

基于Pattern/Reference Object的相机定标

How Many Chess Corners (*pattern_size*) for Calibration?

■ How many parameters we have?

- 6 extrinsic parameters $(\theta, \varphi, \psi, t_x, t_y, t_z)$
- 4 intrinsic parameters (f_x, f_y, c_x, c_y)
- 5 distortion parameters $(k_1, k_2, p_1, p_2, k_3)$

3D
geometry

■ 2D geometry -- 5

- 3 points yield 6 constraints (in principle)
- Enough for 5 parameters!
- More for robustness

2D
geometry

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）

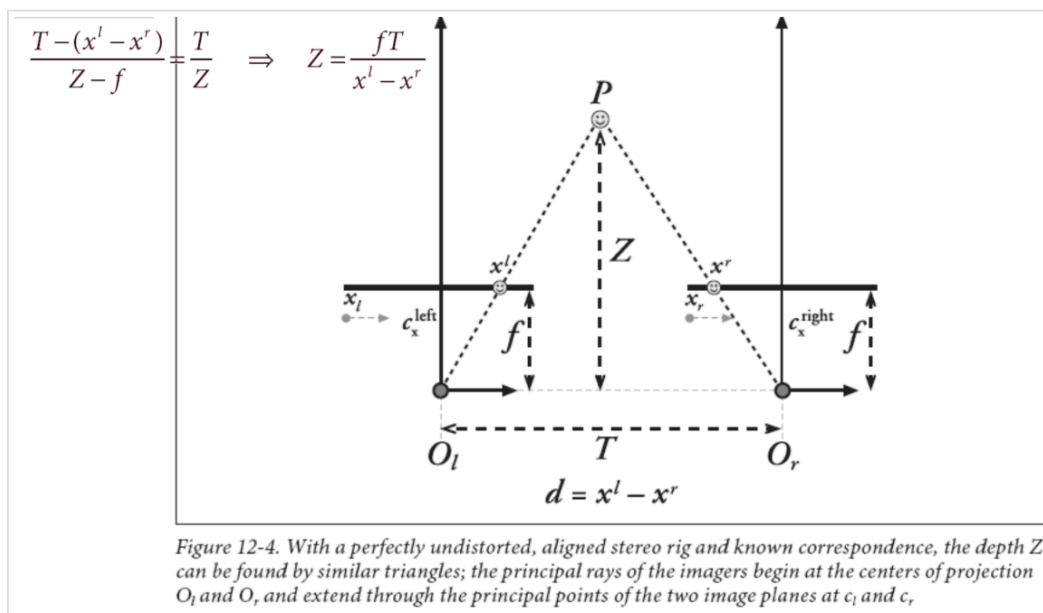
求解：所有的参数。N 个点 K 个视角可以列出 $2NK$ 个等式，会带来 $6K+4$ 个参数。需要 $2NK > 6K+4$ 。

基本过程简述

1. 获取标定物体网格的角点在坐标系的位置
2. 找到图片的角点
3. 根据图像空间坐标系到世界坐标系列出等式
4. 求解相机参数

立体视觉

立体视觉的三角测量基本原理



公式推导：使用三角形相似推导出 $Z = \frac{fT}{x^l - x^r}$

立体视觉的基本步骤

1. 消除畸变
2. 校正相机 (Rectification)
3. 两幅图中找到相同特征
4. 三角测量

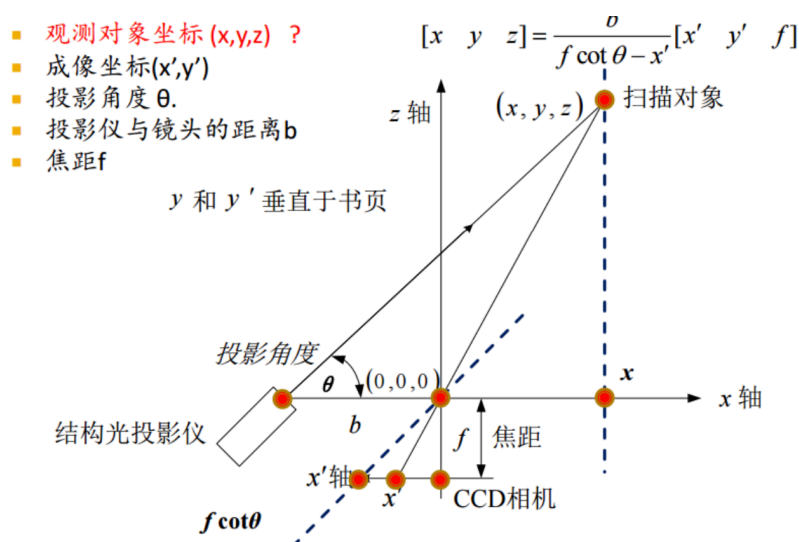
三维数据获取

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

projector (one or more), CCD camera (one or more), and depth recovery system

利用结构光获取三维数据的基本原理



ICP算法的作用和基本步骤

ICP (Iterative Closet Point): 迭代最近点方法，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。

基本步骤： 给定两个三维点集 X 与 Y ，将 Y 配准到 X

1. 建立两个扫描结果之间的对应关系
2. 通过迭代获得一个仿射变换函数能够描述1中对应点之间的变换关系
3. 对 Y 应用上一步求得的仿射变换，更新 Y
4. 两个结果中距离最近的点作为对应点，计算对应点的距离如果大于阈值，重复 2, 3，否则停止计算

寻找F就变成了找到使Cost最小的点的搜索过程

人脸识别

PCA（主元分析）

基本思想及原理

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。

优化目标函数的推导

d维空间 $x=(x_1, \dots, x_d)$ ，投影方向 $a_1=(a_{11}, a_{12}, \dots, a_{1d})^T$ ， $a_1 a_1^T = 1$ 投影值

$$z_1 = a_1^T x = \sum a_{1i} x_i$$

问题：最大化 $\text{var}(z_1)$ ，求投影方向 $\text{argmax}_{a_1} \text{var}(z_1)$ 因为 $\text{var}(z_1) = a_1^T S a_1$ 其中 $S = E(x_i, y_i) - E(x_i)E(y_i) = \text{Cov}(x_i, y_i)$ 所以就是最大化 $a_1^T S a_1$ ，其中 $a_1^T a_1 = 1$

解：设 λ 为Lagrange乘子，转为最大化 $a_1^T S a_1 - \lambda(a_1^T S a_1)$ 以 a_1^T 为变量，对上式求微分，令微分为零。 $S a_1 - \lambda a_1 = 0$ $S a_1 = \lambda a_1$ 观察上述表达式，就是矩阵特征值的定义

所以必须用协方差矩阵最大特征值对应的特征向量，转化为求协方差矩阵

Eigenface

使用特征脸进行人脸识别的方法

基本步骤

1. 预处理：根据人眼位置进行裁剪，进行灰度均衡化。
2. 将二维人脸图像按一行行向量拼成一行，得到列图像；并把所有列图像拼起来，并求出平均人脸。
3. 图像的协方差矩阵。
4. 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。

将识别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。

物体识别

Visual Recognition

基本任务大概可以分为哪几大类

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

都有哪些挑战因素

- 视角变换
- 光线变化

- 尺度变化
- 物体形变
- 物体遮挡
- 背景凌乱
- 内部类别多样

基于词袋 (BoW) 的物体分类

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

基本步骤

1. 特征提取与表示
2. 通过训练样本聚类来建立字典 (codewords dictionary)
3. 用字典的直方图来表达一张图像
4. 根据 bag of words 来分类未知图像

物体识别-CNN

CNN

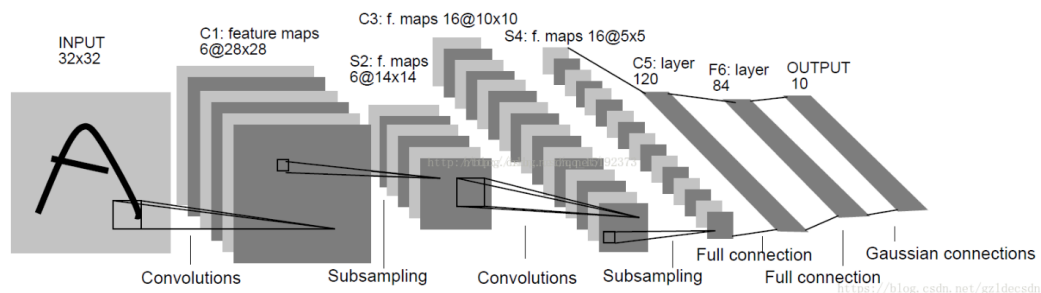
一个简单的卷积神经网络是由各种层按照顺序排列组成, 网络中的每个层使用一个可以微分的函数将激活数据从一个层传递到另一个层。卷积神经网络主要由三种类型的层构成: **卷积层**, **汇聚 (Pooling) 层**和**全连接层**。通过将这些层叠加起来, 就可以构建一个完整的卷积神经网络。

卷积层和pooling层的作用

- **卷积层**是构建卷积神经网络的核心层, 它产生了网络中大部分的计算量。
- 在连续的卷积层之间会周期性地插入一个**polling层**。它的作用是逐渐降低数据体的空间尺寸, 这样的话就能减少网络中参数的数量, 使得计算资源耗费变少, 也能有效控制过拟合。

计算第一个卷积层各种Weight个数

以经典的LeNet-5例子来逐层分析各层的参数及连接个数:



C1卷积层：有 6 个 feature map，每个 feature map 中的每个神经元（每个点）与一个 5*5 的卷积核相连接；该卷积核大小为 5*5，padding 为 0，步长（stride）为 1。

1. 特征图的大小：

$\text{feature_map} = (\text{input_size} + \text{padding} * 2 - \text{卷积核的边长} + \text{步长} \text{stride}) / \text{步长}$

input_size 在图中标明为 32 所以 feature_map 为： $(32 + 2 * 0 - 5 + 1) / 1 = 28$ ，大小为 28*28

2. 参数的个数：

公式： $(\text{卷积核的大小} + \text{步长}) * \text{该层卷积核的个数}$ 所以 c1 层有 $(5 * 5 + 1) * 6 = 156$ 个参数，

3、链接个数： 公式： $[(\text{卷积核的大小} + \text{步长}) * \text{该层卷积核的个数}] * \text{feature_map大小}$

所以 c1 层有 $(5 * 5 + 1) * 6 * (28 * 28) = 122,304$ 个链接。

BP算法

作用

BP网络能学习和存贮大量的输入-输出模式映射关系，而无需事前揭示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（input）、隐层(hidden layer)和输出层(output layer)。

与梯度下降法的关系

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w（权重）