

Computer Vision Review

引言

格式塔法则

- Law of Proximity, 接近
- Law of Similarity, 相似
- Law of Common Fate
- Law of Symmetry, 堆成
- Law of Continuity, 连续
- Law of Closure, 闭合

Marr视觉表示框架

- 第一阶段: Primal Sketch, 对原始图像处理, 提取例如角点, 边缘, 纹理等基本特征, 这些特征的集合称为基元图
- 第二阶段2.5D: 以观察者为中心恢复**深度**, 法线方向, 轮廓
- 第三阶段3D, 以物体为中心, 恢复, 表示, 识别三维物体

Binary Image

几何特性

- 尺寸和位置
 - 面积, 直接加和 $A = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} B[i, j]$
 - 区域中心, 计算图像的中心, 例如 $y = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} jB[i, j]}{A}$
- 方向
 - 最小问题, 用**最小二乘法**找到一条直线作为方向
- 伸长率
 - 用**两个**极端的方向的最小二乘法结果做比
- 密集度: $C = \frac{A}{p^2}$, 其中 p 是周长, A 是面积
- 形态比, 区域最小外接矩形的长宽比
- 欧拉数 (genus), **连通分量数-洞数**, 即 $E = C - H$ 。这种拓扑特性能保证旋转, 平移, 比例放大下不变。

投影特性

- **定义**：给定直线，用垂直这个直线的一簇等距直线将一幅二二值图像分割成若干条，每一条内统计像素值为1的像素的数量。
- **计算**：

连通区域

- 邻点：包括**四联通邻点**（东西南北）和**八联通邻点**。
- 连通性：从像素 $[i_0, j_0]$ 到像素 $[i_n, j_n]$ 的路径是一个相邻两点为邻点的序列。
- **连通分量(Connected Component)**
 - 两点连通定义：存在一条两点间的路径。连通是种等价关系。
 - 连通分量是**连通像素的集合**。
- **连通分量标记算法**：递归，序贯
 - 序贯算法(for 4连通)，Basic idea:对扫描过程的每个像素点，考虑左边点和上面点的标记，分成四种情况。如果左边和上边存在矛盾，则计入**等价表**，扫描完成后，再扫一次，并根据等价表调整标记。
- **区域边界跟踪算法**：
 1. 从左到右，从上到下扫描，寻找起始点 $s(k), k = 0$;
 2. 图像边界上的跟踪点设置为 c ，对于起始 $c = s(0)$ ，设置 b 为 c 的左邻点。在扫描过程中， b 始终不属于连通分量。
 3. 更新 b 和 c 。具体方法是从 b 开始逆时针扫描 c 的8个邻点。对于找到的第一个属于连通分量的邻点，设置为 c ，并将上一个邻点设置为 b 。

Edge

基本想法

边缘是图像中发生剧烈变化的地方。

- 对应着一阶导的极值点
 - 对应着二阶导的零点
- 需要考虑对噪音的敏感度，导数阶数的增加会导致噪音的扩大。

四种最主要的不连续(discontinuity)

- Surface **normal** discontinuity (瓶盖)
- **Depth** discontinuity
- Surface **color** discontinuity
- **illumination** discontinuity

模板卷积

template and convolution.

正常来说卷积的结果会比原来的图像小。

基于一阶导数的边缘检测

梯度

图像中用差分近似偏导数：

$$G_x = f[x+1, y] - f[x, y]$$

$$G_y = f[x, y] - f[x, y+1]$$

一般用卷积模板进行计算。

- Sobel 算子

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad G_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

- Prewitt 算子:运算较快

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad G_y = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

实际上，上面两个算子说明了邻域加权的方法。

基于二阶导数的边缘检测

- Laplacian算子
- LoG(Laplacian of Gaussian)算法

高斯滤波+拉普拉斯边缘检测：

- 先和高斯卷积，在和拉普拉斯卷积
- 求高斯的拉普拉斯微分，再卷积

Canny 边缘检测

基本步骤

1. 用**高斯滤波器**平滑图像：在平滑去噪和边缘检测的矛盾中获得平衡
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)**
 - 将根据 G_x 和 G_y 得到的方向角离散成邻域8个格子的两个，例如方向角是20度，梯度方向就是左右两个格子。
 - 如果方向上的两个格子的幅值都比中心点的大，那让中心点的值 = 0
 - $N[i, j] = NMS(M[i, j], \zeta[i, j])$
4. 用**双阈值算法**检测和连接边缘
 - 取高、低两个阈值 (T_1, T_2) 作用于 $N[i, j]$ ，得到两个边缘图

- 高阈值图: $N[i, j] > T_2$
- 低阈值图: $N[i, j] > T_1$
- 连接高阈值边缘图, 出现断点时, 在低阈值边缘图中的8邻点搜索边缘点
 - 阈值太低, 会得到假边缘
 - 阈值太高, 部分轮廓会丢失
 - 选用两个阈值, 结果更有效。

Local Features

Harris corner Detector

- Basic Idea: 容易识别的特征 -> 往任意方向移动一个小窗口, 都会带来很大变化的
- 形式化表达, 窗口偏移一个向量后的**改变情况**:
 - $E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$, 其中 $w(x, y)$ 是窗函数, u, v 是窗的偏移。
 - 如果 u, v 的值很小, 则有:

$$I(x + u, y + v) \approx I(x, y) + uI_x(x, y) + vI_y(x, y)$$
 - 带入展开, 可以得到

$$E(u, v) = [u, v] \left(\sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} = [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$
- 实际上上式是椭圆的方程: 把 u, v 作为变量, M 的最大最小特征值的平方根刚好是“等高线”的短轴和长轴。
- 根据特征值 λ_1 和 λ_2 的关系可以决定特征的类型, 如果二者差距大, 则是Edge, 如果二者都很大并且近似, 则是Corner。
- 设计函数 R 来得到corner response: $R = \det M + k(\text{trace} M)^2$
 - $R > 0$, 则是corner
 - $R < 0$. 则是Edge
 - R 绝对值小, 则没有明显特征

Harris Detector就是要找到 R 值的局部极大点。

图像改变对特征检测的影响

- 几何
 - 旋转
 - 近似 (旋转+均匀放大或缩小)
 - Affine(Scale dependent on direction)
- 光线(Photometry)
 - 光强的变化, 实际上是对 R 值线性影响

Harris 算法对**Image Scale**的表现很差。

尺度不变的原理

考虑同一点周边的不同大小的区域(circle)，相应的大小的区域在不同尺寸的图像中依然是类似的。问题是如何在每个图像中独立的选择circle，即区域。

- 解决方法

SIFT描述子

- 计算的基本步骤：
 1. 围绕检测到的特征，取一个窗口
 2. 计算内部每个像素的方向（根据梯度的方向）
 3. 除去较小的edges
 4. 根据剩下的Edges计算直方图

Curve

Hough变换

- 基本思想：图像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。
- 基本原理： $y = mx + c \rightarrow c = -xm + y$ ，从而对图像上的每点，可以计算出各自对应的所有的参数的可能（在(m, c)平面构成一条直线），从而能对参数的组合投票。
- 极坐标： m, c 的范围？ c 是有限的，因为图像是有限的。为了避免垂直直线和无限空间的影响，采取极坐标的思考方式，因为图像是有限的， ρ 是有限的。
- 过程：
 1. 适当地量化参数空间（？）
 2. 把参数空间的每个单位都作为一个累加器，把累加器初始化为0
 3. 对图像空间的每个点，在其所满足的参数方程对应的累加器上+1
 4. 累加器阵列的最大值对应所得模型的参数

图像频域

- 傅里叶变换保存了每个频率的大小（多少）和phase（空间信息）
- 图像的低频是图像缓慢变化的地方，是边缘以内的信息，表现了图像的轮廓，高频是图像变化快的地方噪音和细节。高通滤波得到的是锐化结果。
- 拉普拉斯金字塔是从高斯金字塔的每层相减得到的，而高斯金字塔的每层实际上是不同滤波带宽的低通图像，他们相减得到的是经过特定带宽的频率，也就是得到了原来图像的带通滤波。

图像拼接 Image Stitching

基本步骤

- Detect feature points in both images (在各自图中寻找特征点) -> **DOG**
 - Detect key points
 - build the SIFT descriptors
- find corresponding pairs (寻找对应的特征点)
 - Match SIFT descriptors(根据描述子之间的欧氏距离)
- Use these pairs to align the images (用这些点来对其图像)
 - Fitting the transformation
 - fit the transformation matrix $\begin{Bmatrix} h_1 1 & h_1 2 & h_1 3 \\ h_2 1 & h_2 2 & h_2 3 \\ 0 & 0 & 1 \end{Bmatrix}$
 - 6个变量，每个点给出两个方程，所以至少需要三个点
 - 最小二乘法
 - RANSAC

RANSAC(RANdom SAMple Consensus随机采样一致算法)

- 目标：排除outliers的干扰
- **基本思想**：如果一个outlier被选择用来计算当前的fit，那么得到的结果不会获得很多来自其他点的支持。
- **基本步骤**：
 1. 随机选择一组点作为种子来估计变形
 2. 从选取的点来计算变形
 3. 找到这组变形的inliers
 4. 如果inliers的数量足够大，重新计算基于所有inliers的对变形的最小二乘估计。
 5. 保存有最大数目Inliers的变形
- **优点**：
 - 处理很多model fitting问题的通用方法
 - 容易实现，容易计算失败率

光流

- 解决的问题：如何估计从一个图像到另一个图像的像素移动？光流解决了像素的Correspondence问题，给定image H上的一个像素点，寻找在image I中周围的拥有同样颜色的像素。
- 基本假设：
 - Brightness Constancy: $I(x + u, y + v, t + l) = I(x, y, t)$, 保持颜色的一致性
 - Spatial Coherence: 某一个点和邻域的点的空间关系保持不变（如果在同一个表面的话）。
 - Temporal Persistence: small motion, 变化在图像中是缓慢发生的
- 纹理复杂的地方光流的计算结果比较可靠

一个点的约束公式推导：

两个图像：image H and image I；H中的某点(x,y), 在I中移动到(x+u, y+v)。

- 由亮度恒定有约束： $H(x, y) = I(x + u, y + v)$
- 由微小运动, $u, v \leq 1$, 所以根据泰勒展开： $I(x + u, y + v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$
- 将上面两个式子合起来, 可以得到 $0 = I_t + \nabla I * [u, v]$, $I_t = I(x, y) - H(x, y)$, $\nabla I = [I_x, I_y]^T$

图像分割

基于k-means聚类的图像分割

- 用聚类进行图像分割的基本原理：Bottom-up，将相似的点组成一组，然后用一个单独的变迁表征他们
 - 区域之间的像素差别比较大，在边界上具有某种不连续性
 - 区域内部的像素具备一定的相似性
 - 根据像素进行聚类，并根据聚类结果分割
- 基本步骤：
 1. 预处理
 2. 随机的选取K个像素点作为center
 3. 将图像的每个像素点分配给一个center
 4. 对当前的每个cluster，计算新的center（均值）
 5. 重复第3，4步知道没有点再被重新标记
 6. 这时得到的每个cluster就是一个区域，可以用区域边界跟踪算法来标记出边界。

基于Mean Shift的图像分割

- 基本原理：找到图像中最密集的区域
- 基本思路：
 1. 选定kernel和bandwidth
 2. 对于每个点，
 1. 以该点为中心放一个窗口
 2. 计算这个窗口内点的重心
 3. 将这个窗口的中心移动到算出的中心上
 4. 重复第2，3步直到收敛

相机模型

景深，光圈，焦距，视场

- 光圈控制景深：光圈越小，DOF景深越大，原理是小光圈，光束更密集，角度更小，可以画图解释

- **焦距控制视场Zoom**：视场越小，焦距越大。

针孔相机模型

- **基本投影公式**： $\frac{-x}{f} = \frac{X}{Z}$, Z 是光轴上的距离
- **齐次坐标系下的透视投影公式**：
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/d \end{bmatrix}, \text{ 即 } (-d\frac{x}{z}, -d\frac{y}{z})$$
- **内参**： (f_x, f_y, c_x, c_y) , 其中前二者代表焦距，后二者代表光学中心
- **内参矩阵**：
$$\begin{bmatrix} -f_u & 0 & u_0 & 0 \\ 0 & -f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} U^{(new)} \\ V^{(new)} \\ S \end{bmatrix}$$

畸变

- **径向畸变**：因为棱镜形状和光圈的位置导致
- **切向畸变**：光学组件的偏移，比如CMOS歪了
- **外参**

相机定标

- **需要求解**：畸变参数，外参和内参

Pattern/Reference Object的相机定标

- **已知**：几何形状（ N 个corner, k 个关于这个定标物的view）； **求解**：相机参数
- **步骤**：
 1. 确定定标物体
 2. 从图像中寻找corners
 3. 构造方程组
 4. 解方程组，得到相机参数

立体视觉

三角测量基本原理

基于几个假设

- 我们有两个相同焦距的相机他们没有畸变，aligned，可测量
- 我们知道3d物体在2的图像中的位置

然后可以根据相似三角形来构造

立体视觉的基本步骤

- 标定相机
- 纠正图像
 - 让相机raw-aligned
- 计算视差
- 估计深度

三维数据获取

结构光成像系统

三个部分：

- 结构光投影仪（一或多）
- CCD相机（一或多）
- 深度信息重建系统

利用结构光获取三维数据基本原理

根据：

- 成像坐标 (x', y')
- 投影角度 θ
- 投影仪和镜头的距离 b
- 焦距 f

来计算观测对象坐标 (x, y, z)

ICP算法

迭代最近点方法，给定两个三维点集X和Y，将Y配准到X

1. 计算Y中每个点在X中的对应最近点
2. 求使得上述对应点平均距离最小的刚体变换，获得刚体变换参数（平移+旋转）
3. 对Y应用刚体变化，更新Y
4. 如果X和y之间对应点平均距离大于阈值，则从1继续开始，否则停止

Eigen Face

随机变量的数据特征

- 方差: $var(x) = E\{[x - E(x)]^2\} = E(x^2) - [E(x)]^2$
- 标准差: $\sigma(x) = \sqrt{var(x)}$
- 协方差
 - $cov(x_i, x_j) = E\{[x_i - E(x_i)][x_j - E(x_j)]\} = E(x_i * x_j) - E(x_i)E(x_j)$
 - 描述了变量间的相关程度, 单个变量的协方差就是方差

估计数据特征

- $cov(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$
- 协方差的符号表明了是正相关还是负相关
- 对于多元, 我们有**协方差矩阵**
 - 对角线是各个变量的方差
 - 这是个对称矩阵
 - n 个变量, $n * n$ 矩阵

主成分分析(PCA)

定义

- d-维空间: 每个数据 x 是d维向量
- 投影方向: d维单位向量 a
- 投影值 z : 投影方向和数据向量的点积
- 目标:
 - max: $var(z)$
 - 求**投影方向**

求解(注意推导过程)

化简 $var(z)$, 根据方差的公式和 z 的定义, 并在中间用协方差定义替换, 可以得到: $var(z) = \vec{a}^T S \vec{a}$

这样成了一个等式约束下的最优化问题, 可以用**Lagrange算子法**。

结果是需要让 a 取得协方差矩阵最大特征值对应的特征向量(特征值对应的就是**方差**)。

以上求得了一个投影方向, 为了继续获得投影方向, 需要增加约束

希望新求得的投影和已求得的方向投影**不相关** -> 也就是协方差为0。

不同特征值的特征向量互不相关!

Eigen Face步骤

- 预处理: 归一化 (直方图均衡化, 直方图拉伸)

- Image -> Vector
- 训练，对k个Face Vector，求得估计的协方差矩阵。根据这个矩阵的特征值和其对应的特征向量，构建转换矩阵。

识别和重构

- 识别：将未知样本经过转换矩阵后和训练集中的样本比较
- 重构：将经过转换矩阵的向量重新恢复为样本。

物体识别

- **基本任务**
 - 给图像和视频分类
 - 检测和定位物体
 - 估计语义和几何属性
 - 区分人类活动和事件
- **挑战因素**
 - 不同的视角
 - 光照
 - 尺寸
 - 形变
 - 遮挡
 - 背景混乱
 - 自身的多样性

基于词袋的物体分类

- **定义**：提取独立的特征，用直方图的形式表征
- **基本步骤**：
 1. feature extraction and representation
 2. building codebook from training samples with clustering
 3. represent an image with histogram of codebook
 4. classify an unknown image with its BoW

CNN

CNN

BP算法