

复习

2019年1月20日 16:46

概论

格式塔法则

1. 图像与背景关系原则 (figure-ground) : 物体/图形比背景更突出
2. 接近原则 (proximity) : 接近/邻近的物体会被认为是一个整体
3. 相似原则 (similarity) : 刺激物的形状/大小/颜色/强度等物理属性方面相似时, 刺激物被认为是一个整体
4. 连续性原则 (continuity) : 若图形的某些部分可以被看作连接在一起的, 则这些部分很可能被认为是一个整体
5. 封闭/闭合原则 (closure) : 对于有些没有闭合的图形, 主体能自行填补缺口使之被认为是一个整体
6. 蕴含律: 对复杂对象进行感知时, 人们倾向于把对象看作对称的, 简单的, 规则的图形

视觉框架的三个阶段

1. Primal Sketch: 处理输入的原始图像, 抽取**基本特征** (角点、边缘、纹理、线条、边界etc.) ——特征的集合称为**基元图**
2. 2.5D Sketch: 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一些深度信息 (深度、法线方向、轮廓etc.) ——不是真正的物体三维表示
3. 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, 恢复表示识别三维物体

二值图像

几何特性:

1. 面积 (零阶矩)
2. 区域中心 (一阶矩)
3. 方向——>最小化问题: $X^2 = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} r_{ij}^2 B[i, j]$, r_{ij} 为点 $[i, j]$ 到直线的距离——最小二乘法求解
4. 伸长率 $E = \frac{X_{\max}}{X_{\min}}$
5. 密集度: $C = \frac{A}{p^2}$, p 为周长, A 为面积
6. 形态比: 区域的最小外接矩形的长宽比
7. 欧拉数: 欧拉数=连通分量数-洞数——平移、旋转、比例不变性

投影计算:

1. 水平/垂直投影: easy
2. 对角线投影计算: 仿射变换——>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 $d = i - j + m$, 其中 $[i, j]$ 为原像素坐标, 图像大小为 $n*m$

连通区域

1. 连通分量标记算法
 - 递归算法
 - 扫描图像直至找到一个没有标记的前景点, 分配标记 L
 - 递归分配标记 L 给该点的邻点

- 若不存在没有标记的点，则停止
 - 返回第一步递归
 - **序贯算法**
 - 扫描图像（左->右，上->下），直至找到一个未标记的前景点，分下面四种情况（E.g. 四连通）
 1. 若上点和左点**有且仅有一个**标记，则**复制**这一标记至当前节点
 2. 若上点和左点有**相同**标记，则**复制**这一标记
 3. 若上点和左点有**不同**标记，则**复制上点**标记，且将两个标记输入**等价表**（构成一个等价集）
 4. 否则，给当前像素点分配一个**新标记**，且**输入等价表**
 - 循环第一步，直到找不到未标记的前景点
 - 在等价表中的每一等价集中找到**最低**的标记
 - 扫描图像，用等价表中的最低标记取代每一标记
2. 区域边界跟踪算法
- 符号定义：1. c：当前点（在边界上）；2. b：当前点的领域点（不在边界上）
 - 过程
 - 扫描图像，求区域s的起始点（左->右，上->下）： $s(k) = (x(k), y(k)), k = 0$
 - 按**逆时针**方向记从**b**开始的c的八个邻点分别为 $n_1, n_2, \dots, n_8, k = k + 1$
 - 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, $b \in \bar{S}$
 - 从b开始，沿逆时针方向**找到一个** $n_i \in S$
 - 置 $c = s(k) = n_i, b = n_{i-1}$ ——保证了b不属于S
 - 重复3,4,5步直至 $s(k) = s(0)$

边缘

1. 模板卷积
2. 边缘产生的位置：
 - 图像深度不连续处
 - 图像（梯度）朝向不连续处
 - 图像光照不连续处
 - 纹理变化处
3. 边缘检测基本思想函数导数反映图像灰度变化显著程度——边缘所在位置：**一阶导数的局部极大值&二阶导数的过零点**
4. 基于一阶导数的边缘检测方法
 - 梯度
 - 幅值： $|G(x, y)| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y| \approx \max(|G_x|, |G_y|)$
 - 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: $G_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$, $G_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$ ——系数 $c=1,2,3$ 时分别对应Prewitt, Sobel, Sethi算子

5. 基于二阶导数的边缘检测方法

- 拉普拉斯算子 $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$, $\frac{\partial^2 f}{\partial x^2} = f[i, j+1] - 2f[i, j] + f[i, j-1]$, $\frac{\partial^2 f}{\partial y^2} = f[i+1, j] - 2f[i, j] + f[i-1, j]$
- LOG: **高斯平滑+Laplace** $\Rightarrow h(x, y) = \nabla^2 [g(x, y) * f(x, y)] = [\nabla^2 g(x, y)] * f(x, y)$
——>**LOG算子** = $\nabla^2 g(x, y)$ (对高斯模板求拉式算子的结果)

6. Canny边缘检测

- 高斯滤波器平滑图像
- **一阶偏导有限差分**计算梯度 (幅值和方向)
- 对梯度幅值进行**NMS** (非极大值抑制)
简化的NMS步骤
 - 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
- **双阈值法**连接边缘 (取高低阈值 T_1, T_2) ——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

曲线

1. 曲线表示

- 几何属性 (离散化)

曲线长度	切方向 $\phi = \arctan \frac{y_{i+k} - y_i}{x_{i+k} - x_i}$	曲率 $\theta = \phi_l - \phi_r$
------	--	-------------------------------

- 链码: 用相邻边缘点组成的**方向序列**表示边缘 (4方向/8方向链码) ——问题: 无法精确表示边缘方向
- $\psi - s$ 图: 从第一个边缘点开始计算**曲线长度s**和**斜率 ψ** , 然后在图中绘制水平直线 (长度为s, 直线的纵坐标为 ψ) ——>解决链码中将**方向离散化**产生的问题
- 图表达: 考虑边缘点间的拓扑关系: 每个边缘点的数据结构 = {点id, 曲线id, 点坐标, 类型id, 表示边缘点间连接关系的数组}

2. 曲线拟合: 重点掌握Hough变换

- Hough变换: 对每对参数组合进行**voting**
 - 过程
 - 适当量化参数空间 (栅格化) ——问题
 - 对每一个待拟合的点, 其满足的参数方程对应的累加器+1
 - 所有累加器中最大值对应的参数组合即为模型参数
 - 直线拟合: 参数组合 (r, θ)
 - 圆弧拟合: 参数组合 $(a, b): \begin{cases} x = a + r \cos \theta \\ y = b + r \sin \theta \end{cases} \Rightarrow b = a \tan \theta - x \tan \theta + y$, 其中 θ 为边缘点处梯度 (可由 a, b, x, y 表示)

局部特征

1. Harris角点检测

- 原理: 在图像 $I(x, y)$ 对指定大小窗口进行各个方向的平移, 观测窗口内图像的相似程度

- 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
- 自相关函数——描述窗口**平移后的相似度**: $E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2$
 - 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 $E(u, v) = [u \ v] \left(\sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$
- 记自相关矩阵 $M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \Rightarrow$ 特征值分解得 λ_1, λ_2
 - Corner: λ_1, λ_2 都很大, 且 $\lambda_1 \sim \lambda_2$
 - Edge: $\lambda_1 \gg \lambda_2$ 或 $\lambda_1 \ll \lambda_2$
 - Flat: λ_1, λ_2 都很小
- 量化指标: 响应值函数 $R = \det M - k(\text{trace} M)^2$, 其中 $\det M = \lambda_1 \lambda_2, \text{trace} M = \lambda_1 + \lambda_2, k$ 为经验常数
 - Corner: $R > 0$ 且 $|R|$ 很大 ——>可用**阈值作二值化**实现角点检测
 - Edge: $R < 0$ 且 $|R|$ 很大
- 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、平移不变性
 - 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 ——> Solution: Harris-Laplace / SIFT

注: 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**

2. Harris-Laplace

- Harris角点检测中的自相关矩阵 $M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = g(\sigma_I) \otimes \begin{bmatrix} I_x^2(x, y) & I_x I_y(x, y) \\ I_x I_y(x, y) & I_y^2(x, y) \end{bmatrix}$
- 定义尺度自适应的自相关矩阵 $M = \sigma_D^2 g(\sigma_I) \otimes \begin{bmatrix} I_x^2(x, y, \sigma_D) & I_x I_y(x, y, \sigma_D) \\ I_x I_y(x, y, \sigma_D) & I_y^2(x, y, \sigma_D) \end{bmatrix}$

$g(\sigma_I)$ 表示尺度为 σ_I 的高斯卷积核	I_x 和 $I_y(x, y, \sigma_D)$ 表示图像使用高斯函数 $g(\sigma_D)$ 进行平滑后取LOG	σ_I : 积分尺度, 决定Harris角点当前尺度;	σ_D : 微分尺度, 决定角点附近微分值变化, 通常 $\sigma_I > \sigma_D$
---------------------------------------	--	------------------------------------	--

- 流程
 - 确定尺度空间的一组取值 $\sigma_I = (\sigma_0, \sigma_1, \dots, \sigma_n) = (\sigma, k\sigma, k^2\sigma, \dots, k^n\sigma), \sigma_D = s\sigma_I$, 经验值 $s = 0.7$
 - 对于确定的尺度空间值 σ_D , 计算Harris角点响应值
 - 尺度空间搜索: 计算候选点的Laplace响应值, 并对于给定阈值作比较 $F(x, y, \sigma_i) = \sigma_i^2 |L_{xx}(x, y, \sigma_i) + L_{yy}(x, y, \sigma_n)| \geq \text{threshold}$
 - 将响应值 F 与相邻两个尺度空间对应的拉式响应值进行比较, 找到 $F(x, y, \sigma_i) > F(x, y, \sigma_l), l = i - 1, i + 1$ 对应的 i

3. SIFT特征提取

- 特征点提取
 - 对图像做降采样得到一系列Octave: 前一个Octave中图像的大小为后一个的两倍
 - 对每个Octave对应的图像应用不同 σ 的高斯核卷积 $G(x, y, \sigma)$, 则一个Octave中能够得到一组图像

- iii. 对每个Octave中的图像（设共有S个）作差分，就可得到S-1个DOG图像构成的图像金字塔
 - iv. 对每个Octave下的DOG金字塔中图像的每个点，比较其8邻域内的点以及上下层9*2个点（共26个点），若该点在26点空间中为极大值/极小值，则该点为当前尺度（Octave）下的一个特征点，记录关键点信息为如下格式：KeyPoint = (x, y, σ)
- 关键点定位及尺度确定
 - 关键点精确定位
 1. DOG函数D(x)的Taylor展开为 $D(X) = D + \frac{\partial D}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X$
 2. 令D(x)的导数为0，得到极值点偏移量 $\hat{X} = -\left(\frac{\partial^2 D}{\partial X^2}\right)^{-1} \frac{\partial D}{\partial X}$
 3. 若 $\hat{X} = (x, y, \sigma)^T$ 在任意维度大于0.5则说明极值点精确位置离另一个点更近，则改变当前关键点位置，定位到新点后重复上一步操作，迭代一定次数后仍不收敛则不认为该检测点为关键点
 4. 精确关键点处函数值 $D(\hat{X}) = D + \frac{1}{2} \frac{\partial^2 D}{\partial X^2} \hat{X}$
 - 消除边缘效应——DOG算子有较强的边缘效应，边缘点的特征表现：某个防线有较大的主曲率，而其垂直方向主曲率较小
 1. 计算主曲率：使用 2×2 的Hessian矩阵 $H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix}$ ，D的主曲率与H特征值成正比
 2. 设H的较大特征值为 α ，较小者为 β ，且记 $\frac{\alpha}{\beta} = r$ ，则边缘点对应r较大的情况。而 $\frac{\text{Tr}(H)^2}{\det(H)} = \frac{(r+1)^2}{r}$ 在两个特征值相等时最小且随r增大而增大
 3. 因此设定阈值 r_t ，若 $\frac{\text{Tr}(H)^2}{\det(H)} < \frac{(r_t+1)^2}{r_t}$ ，则认为当前关键点不是边缘，可以保留；否则应予以剔除
 - 关键点方向确定
 1. 在关键点为中心的8邻域窗口内计算关键点的梯度幅值和方向
 2. 对每个关键点（尺度为 σ ），用**直方图**统计其一定领域内的像素梯度分布（梯度方向可近似到 $\pm 45^\circ, \pm 90^\circ, \pm 135^\circ, 0^\circ, 180^\circ$ 上）
 3. 找到直方图中对应个数最多的梯度方向，即为当前关键点的方向
 - 特征向量生成
 1. 将坐标轴旋转到关键点方向上：旋转后新坐标 $\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$
 2. 对关键点的8*8领域内所有像素点计算其对应的梯度幅值和方向
 3. 将8*8领域划分为4个4*4的小领域，在每个子领域上计算统计**每个梯度方向上**对应所有**梯度幅值的累加**⇒得到属于关键点的4*8共32个特征（每个子领域8个方向对应8个特征）
 4. 将32个特征的值归一到(0,1)内，最后将关键点用上述得到的32个值描述，即 $L = (l_1, l_2, \dots, l_{128}), l_i \in (0,1)$ ——有128维是因为原论文中作者推荐对 16×16 邻域进行切分
 - SIFT特征的匹配——度量两幅图像中关键点的相似性

对图像A中某一关键点，找出图像B中与其距离最近的两个关键点，采用 $ratio = \frac{\text{最近邻距离}}{\text{次近邻距离}}$ 的评价方法：ratio的阈值越小说明匹配准确度越高

 - ratio = 0.4：对准确度要求较高的匹配
 - ratio = 0.5：一般情况的取值
 - ratio = 0.6：对匹配点数目要求较多的匹配

一般地，ratio = 0.8则认为当前匹配为错误匹配
 - 优点
 - 尺度/光照/旋转不变性

- 在刚体的表征上尤其有效
- 局部表征能力强
- 缺点
 - 耗时
 - 处理非刚性边缘时表现较差
 - 在严重的仿射扭曲下效果较差

图像拼接

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

RANSAC

- 基本假设
 - 正确数据：内点，可以被模型描述
 - 异常数据：偏离正常范围很远
- 基本步骤
 - 随机采样：在样本集中随机抽取n个样本，构成S，基于S中的样本对初始模型进行估计
 - 模型验证：计算样本集中其他样本到模型的误差—— \rightarrow 误差小于阈值的样本+S中样本共同构成内点集 S^*
 - 重复上述步骤并保留到目前为止最好的内点集：数据点最多&其他样本到该集合的残差最小——称为最好
 - 输出最佳模型的参数
- 参数确定：每次的随机采样数n（用于确定模型的样本个数，例如对直线， $n=2$ ）和重复次数K
 - 定义内点比例 $\omega = \text{内点数} / \text{样本总数}$
 - 结论1: $K = \frac{\log(1-P)}{\log(1-\omega^n)}$ —— ω 不高时，n过大会导致K急剧增大
 - 实际应用中，一般 ω 难以预先估计——采用自适应估计法
 - 初始化 $K = \infty, \text{count} = 0, P = 0.99$
 - 随机采样n个样本、计算模型并检查内点数
 - 计算 $\omega = \text{上一步中求得的内点数} / \text{样本总数}$
 - 计算 $K = \frac{\log(1-P)}{\log(1-\omega^n)}, \text{count}++$
 - 重复上述步骤，直到 $K < \text{count}$
- 优点：
 - 适用性强（能够解决很多模型拟合问题）
 - 易于实现
- 缺点：
 - 保证代价（迭代次数，计算耗时等）不过大的基础上，只能处理外点比例不高的数据——In contrast, Hough变换能处理外点比例很高的数据集

- 实际问题中很多数据集的外点比例很高——maybe可以通过随机选择子集提高性能

运动跟踪

光流法

- 基本假设
 - 亮度恒常性：目标像素强度在相邻帧不发生变化—— $I(x+u, y+v, t+1) = I(x, y, t)$
 - 空间一致性：相邻像素拥有相似运动
 - 时间规律：相邻帧的时间间隔足够短
- 公式
 - 假设一个目标像素在 t 时刻亮度为 $I(x, y, t)$ ， $t + \delta t$ 时刻亮度为 $I(x+u, y+v, t + \delta t)$ ，则两者相等
 - Taylor展开得 $I_x u + I_y v + I_t = 0$, $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $I_t = \frac{\partial I}{\partial t}$ ，即 $-I_t = \nabla I \cdot \begin{pmatrix} u \\ v \end{pmatrix}$
- 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
 - $A^T A = \sum \nabla I (\nabla I)^T$ 需可逆
 - $A^T A$ 的特征值不能太小——防止被噪声干扰
 - $A^T A$ 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

图像分割

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）

基于K-means聚类算法

- 点的表示方式（特征空间）
 - 灰度值
 - RGB值
 - 坐标值&灰度值
 - 坐标值&RGB值
- 基本步骤
 - 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
 - 重新计算每一簇中新的中心（一般取当前类内所有样本在每一维度上的均值）
 - 重复2,3两步直至没有点被重新分配
 - 退出迭代后，将每一簇中的所有点赋予簇中心的类别标记
- 缺点
 - K 的选取没有明确规则
 - 每次迭代要遍历整个样本集，开销大
 - 基于距离划分——只适用于凸分布的数据集

基于Mean Shift的聚类

- 基本步骤
 - 在空间内初始化几个聚类中心，以及一定大小 h 的邻域（簇）

2. 计算每个簇的质心 $c_h(x) = \sum_{i=1}^n G\left(\left\|\frac{x_i - x}{h}\right\|^2\right) x_i / \sum_{i=1}^n G\left(\frac{x_i - x}{h}\right)$, h 为核函数带宽, x 为当前簇中心坐标
3. 将簇中心移动至质心所在位置: $x = c_h(x)$, 并重复2,3步骤直至收敛, 即 $\|m_h(x) - x\| < \varepsilon$
- 偏移量及偏移方向计算
 - 偏移量 $f(x) = \frac{1}{n} \sum_{i=1}^n K(x - x_i)$, 其中 K 称为核函数——代表不同的质心计算方式

相机

相机模型

基本概念

- 概念
 - 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
- 相互联系
 1. 小光圈→大景深
 2. 短焦距→大视场角: $\varphi = \arctan \frac{d}{2f}$

内参模型: 相机坐标系→成像坐标系→像素坐标系

1. 相机坐标系→成像坐标系(3D TO 2D)

$$\begin{cases} x_{screen} = f \frac{X}{Z} \\ y_{screen} = f \frac{Y}{Z} \end{cases} \Rightarrow Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

其中 f 为焦距

2. 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

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_u & 0 & c_u \\ 0 & s_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \text{ 其中 } u, v \text{ 为像素坐标系坐标, } x, y \text{ 为成像坐标系坐标, } s_u, s_v \text{ 为表示度量单位转换的参数, } c_u, c_v \text{ 为成像坐标系相对于像素坐标系的偏移量}$$

3. 内参矩阵 $M: q = MQ \Rightarrow \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$, 最终物体在像素坐标系中坐标为 $\left(\frac{u}{w}, \frac{v}{w}\right)$

其中 $f_u = f s_u, f_v = f s_v \Rightarrow$ 内参 (f_u, f_v, s_u, s_v)

畸变模型

1. 径向畸变

- 原因: 由于透镜的几何形状不完美或安装位置引起的畸变
- 分类: 枕形畸变/桶形畸变
- 校正模型: $\begin{cases} x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{cases}$

2. 切向畸变

- 原因: 透镜平面和成像平面不平行引起的畸变
- 校正模型: $\begin{cases} x_{corrected} = x + [2p_1 y + p_2(r^2 + 2x^2)] \\ y_{corrected} = y[2p_2 x + p_1(r^2 + 2y^2)] \end{cases}$

3. 畸变参数: $(k_1, k_2, k_3, p_1, p_2)$

外参模型: 世界坐标系→相机坐标系(3D TO 3D)

外参矩阵: $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \end{bmatrix} \Rightarrow$ 外参: $(\theta, \varphi, \psi, t_x, t_y, t_z)$ 分别表示世界坐标系相对相机坐标系的旋转、平移量

相机标定

求解目标: 求解内外参(3D)、畸变参数(2D)

基于Pattern/Reference Object的相机标定方法

- 已知: 给定标定物体(棋盘格)的N个角点和K个视角
- 求解: 所有参数
- N,K计算: 方程数 $2NK \geq$ 未知参数数 $2K + 4$: 对所有视角, 外参变化但内参不变
- 步骤
 1. 检测棋盘格中的角点(得到角点在图片中的位置)
 2. 根据棋盘格尺寸获取角点在空间中的位置
 3. 根据对应关系代入公式求解相机参数

立体视觉

双目视觉

基本步骤:

1. 去畸变: 根据畸变模型消除畸变
2. 矫正
 - 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(2D搜索 TO 1D)
 - 目的: 使左右观测所得图像行对齐
3. 角点匹配: 在1D的直线上搜索匹配值函数的极值位置
4. 计算深度——Triangulation: $\frac{T - (x^l - x^r)}{Z - f} = \frac{T}{Z}$

三维数据获取

方法: 结构光/TOF系统

结构光:

- 组成部分: 结构光投影仪+CCD相机+深度信息重建系统
- 基本原理——公式推导: $\begin{bmatrix} x & y & z \end{bmatrix} = \frac{b}{f \cot \theta - x'} \begin{bmatrix} x' & y' & f \end{bmatrix}$, x', y' 为物体在CCD上的投影坐标, b 为投影仪距原点位置, θ 为投影角度
- 编码方式: 直接编码/时分复用编码/空分复用编码

多视角图像匹配——ICP算法

- 目标: 计算两组数据(两帧图像)间的旋转平移量, 使之形成最佳匹配
- 输入: 点集 P, P' 输出: 最佳匹配的旋转平移量 R, t s.t. $\forall i, p_i = R p'_i + t$
- 步骤:
 - 根据最近邻域规则建立 P 和 P' 中点的关联——即初始化一个 R 和 t , 一般根据传感器和机器人移动参数得到
 - 两帧图像之间不能相差过大——会导致通过机器人运动学得到的 R_0 和 t_0 不准确
 - 利用线性代数/非线性优化的方式估计旋转平移量
 - 使用估计得到的旋转平移量对点集 P' 的点进行旋转平移
 - 若旋转平移后的均方差小于阈值, 则结束; 否则迭代重复
- 线性代数求解法

- 构建最小二乘问题 $\min_{R,t} J = \frac{1}{2} \sum_{i=1}^n \|p_i - (Rp'_i + t)\|_2^2$, 其中 $e_i = p_i - (Rp'_i + t)$ 为第 i 个匹配点的误差
- 旋转平移分解 (去质心化)
 - 根据 $\arg\min_R \frac{1}{2} \sum_{i=1}^n \|q_i - Rq'_i\|_2^2$, 其中 $q_i = p_i - p$, $q'_i = p'_i - p'$, $p = \frac{1}{n} \sum_{i=1}^n p_i$, $p' = \frac{1}{n} \sum_{i=1}^n p'_i$
 - 根据 R 计算 t : $t^* = p - R^* p'$
- 求解 R : 优化目标函数可变为 $\max_R \sum_{i=1}^n q_i^T R q'_i$
 - 奇异值分解: 令矩阵 $W = \sum_{i=1}^n q_i q_i'^T$, 对 W 进行 SVD 分解, 可得 $W = U \Sigma V^T$ (Σ 为奇异值从大到小排列组成的对角阵, U, V 为正交矩阵), 则 $R = V U^T$

人脸识别

主成分分析 (PCA)

- 问题定义:
 - d 维空间 $X = (x_1, x_2, \dots, x_d)$
 - 投影方向 $a_1 = (a_1^1, a_1^2, \dots, a_1^d)^T$, 且 $a_1^T a_1 = 1$ (单位向量)
 - 目标 $\arg\max_{a_1} \text{var}(z_1) = \arg\max_{a_1} \text{var}(a_1^T x)$
 - $\text{var}(z_1) = \sum_{i,j=1}^d a_1^i a_1^j S_{ij} = a_1^T S a_1$, 其中 $S_{ij} = E(x_i x_j) - E(x_i) E(x_j) = \text{cov}(x_i, x_j) \rightarrow S$ 为协方差矩阵
- 数学化表达 $\arg\max_{a_1} (a_1^T S a_1)$, s. t. $a_1^T a_1 = 1$
 - \Rightarrow Lagrange 乘子法解得必要条件: $S a_1 = \lambda a_1$ 求解协方差矩阵特征值
 - 为了使 $\text{var}(z_1)$ 最大化, 则 a_1 取最大特征值对应的特征向量
- 取多个主成分 $a_1, a_2, \dots, a_k \Rightarrow$ 问题: $\arg\max_{a_k} \text{var}(z_k)$, s. t. $a_k^T a_k = 1, \text{cov}(z_k, z_l) = 0$ (for $k \geq l \geq 1$) \Rightarrow 必要条件 $S a_k = \lambda a_k$, a_k 为第 k 大的特征值对应的特征向量

Eigenface 人脸识别算法

- 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
- 预处理
 - Mask: 根据人脸两只眼睛的中心位置 旋转/平移/缩放, 使所有训练人脸图像与模板对齐——根据模板切出人脸区域
 - 灰度值归一化: 直方图均衡 & 直方图拉伸
- 训练过程
 - 向量一维化: x_i 为 $M \times N$ 的人脸图像对应的一维化矩阵, $1 \leq i \leq K$
 - 求协方差矩阵: $\Sigma = \frac{1}{K} \sum_{i=1}^K (x_i - u)(x_i - u)^T, u = \text{mean}(x_1, x_2, \dots, x_K)$
 - 求矩阵 Σ 的特征值和相应的单位特征向量 (MN 维)
 - 构建转换矩阵, 求出子空间中的坐标: $y_i = A^T x_i, A = [v_1, v_2, \dots, v_k], k \ll K$
- 识别与重构 (待识别样本 f)
 - 识别: $y_f = A^T f$, 比较 y_f 和 y_i
 - 重构: $\hat{f} = A y_f$
- 特征脸个数 k 的选取: $\arg\min_k \left(\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^K \lambda_i} \geq \alpha \right)$, α 常取 0.95~0.99 (即选取的特征值在总特征值中占比已经足够大)

物体识别

词袋 (Bag-of-words) 模型

- 原因：单单根据SIFT特征描述/匹配图像的计算量巨大
- 码本构建步骤（from 训练集）
 1. 图像预处理
 2. 每张图像提取SIFT特征——SIFT详细见“局部特征”
 3. 对所有SIFT特征聚类（常用K-means），得到K个视觉词——Kmeans详细见“图像分割”
 4. 对每个图像构建基于视觉词的直方图，表征一张图像中不同视觉词出现的次数
- 识别步骤（from 未知图片）：基于K个视觉词对未知图片建立直方图，并比较其与训练集的直方图的距离，取距离最短即为最佳匹配

CNN

反向传播（back-propagation）算法

- 本质：复合求导
- 关键：计算图的理解和使用
 - 节点：运算符
 - 连线上方：前向计算值
 - 连线下方：反向梯度值
 - 常用节点：加法/乘法/最大值节点

CNN结构

- 卷积层：用于提取图像中的特征
 - 概念
 - 卷积核（kernel）：扫描整个图像进行卷积，得到输出图像——感受野（receptive field）
 - 步长（stride）：卷积核移动的间隔大小
 - 填充（padding）：用于处理卷积时无法处理图像边缘的问题——在图像边缘外补一定宽度的0
 - 特征图（feature map）：每一个卷积层后的输出图像
 - 参数计算
 - 输出图像大小计算
记输出为 $M \times M$ 的feature map，输入为 $N \times N$ ，卷积核为 $K \times K$ ，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 $K \times K$ 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**
 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
 - 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 $(K^2 + 1) \times \text{通道数} \times \text{卷积核数}$
 - 作用：用于提取图像中特征
- 池化层
 - 原理：同样是用一定大小的卷积核对输入图像进行卷积
 - 与卷积层区别
 1. 参数不可更新
 2. stride一般较大
 - 池化方式
 1. 单幅图像中池化——最大池化/平均池化
 2. 多幅特征图间池化——只使用最大池化

- 作用

1. 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
2. 多图间池化：减少参数数量 & 找到多图间最显著的特征