

图像配准和拼接

SIFT 提取特征点

首先需要构建高斯金字塔，高斯金字塔的每一层中图片大小一致，每增加一层，图片的长与宽变为上一层图片的一半。在每一层中，后一张图像由前一张图像经过滤波后得到。在构造完高斯金字塔后，在高斯金字塔的每一层中，用后一张图片减去前一张图片得到高斯差分金字塔。假设我们需要在高斯差分金字塔的每一层中得到 S 个能够用于提取特征点的图片，则差分金字塔每一层需要有 $S+2$ 个图片，因为特征点需要与在空间中与它相邻的 26 个点进行比较。为了得到每一层有 $S+2$ 个图片的差分金字塔，我们构造的高斯金字塔每层需要有 $S+3$ 个图片。

然后在高斯差分金字塔中，寻找非 0 的极值点。在找到极值点后，由于图片上的像素是离散的，所以我们找到的极值点不是连续意义上的极值点。由此我们利用泰勒展开来模拟连续函数，在点 (x, y, σ) 处的泰勒展开为

$$D(x) = D + \frac{\partial D}{\partial x} \Delta x + \frac{1}{2} \Delta x^T \frac{\partial^2 D}{\partial x^2} \Delta x$$

令上式一阶导数等于 0，得到 $\Delta x = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}$ ， Δx 即为偏移量。如果偏移量大于 0.5，

则极值点对应的坐标应该发生变化。重复 5 次后，若极值点收敛，位置不再发生变动，则认为其为特征点，否则剔除该点。在纠正完极值点后，边缘处会产生很多特征点。为了进一步筛选这些特征点，需利用该特征点的 Hessian 矩阵进行一些判定。

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

其中二阶导可以用差分形式近似求得，公式如下

$$D_{xx} = D(x+1, y) + D(x-1, y) - 2D(x, y)$$

$$D_{yy} = D(x, y+1) + D(x, y-1) - 2D(x, y)$$

$$D_{xy} = (D(x+1, y+1) + D(x-1, y-1) - D(x+1, y-1) - D(x-1, y+1)) / 4$$

得到 Hessian 矩阵后，进行如下判断

$$\frac{Tr(H)^2}{Det(H)} < \frac{(\gamma+1)^2}{\gamma}$$

其中 γ 取 10。如果符合判定，则保留特征点，否则剔除。

然后进行特征点主方向的确定。对于特征点 (x, y, σ) ，在所属差分图片的 3σ 邻域内统计其他像素点的梯度和梯度方向。将 360° 分为 8 份，即

$$[7\pi/8, 5\pi/8, 3\pi/8, \pi/8, -\pi/8, -3\pi/8, -5\pi/8, -7\pi/8]$$

构建梯度方向直方图，直方图的纵坐标从个数变为梯度的和，选取梯度和最大的梯度方向作为特征点的方向。

最后一步是生成特征点描述子，先将特征点周围 16×16 的邻域分为 4×4 的子区域，每

个子区域包含 16 个点。统计每个子区域的梯度方向直方图，此时需要将每个像素点计算出的梯度方向减去特征点的方向，这样做是为了保证方向不变形。4×4 个 8 维向量拼接在一起构成 128 维向量，进行归一化后就生成了一个特征点的描述子。

特征点匹配

计算图一中每个特征点描述子与图二每个特征点描述子之间的距离，并进行升序排序。如果最小距离<第二小距离×比例，则认为最小距离的点为匹配点。这样匹配出来的特征点对中会出现很多误匹配的坏点，所以我们利用 RANSAC 算法进行剔除。

我们首先从已经配对的特征点组中随机选出四组，计算他们对应的单应性矩阵 H

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

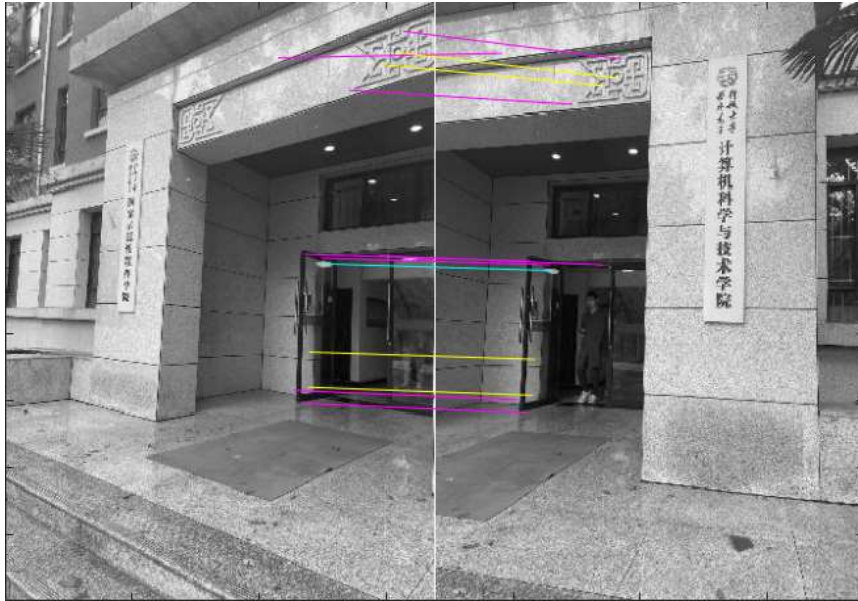
其中 x', y' 为图二中的坐标， x, y 为图一中的坐标，经转化后 H 计算公式如下

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -y'_1x_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x'_1y_1 & -y'_1y_1 & -y'_1 \\ & & & \vdots & \vdots & \vdots & & & \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -y'_4x_4 & -x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x'_4y_4 & -y'_4y_4 & -y'_4 \end{bmatrix} * \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

解上述方程即可得到单应性矩阵 H。得到 H 后，对每一组匹配点进行验证，如果计算值与实际值误差小于阈值，则计数加一，否则不加。进行 10000 次上述操作，得到使得所有匹配点计数最大的 H 矩阵作为图片之间的变化矩阵。



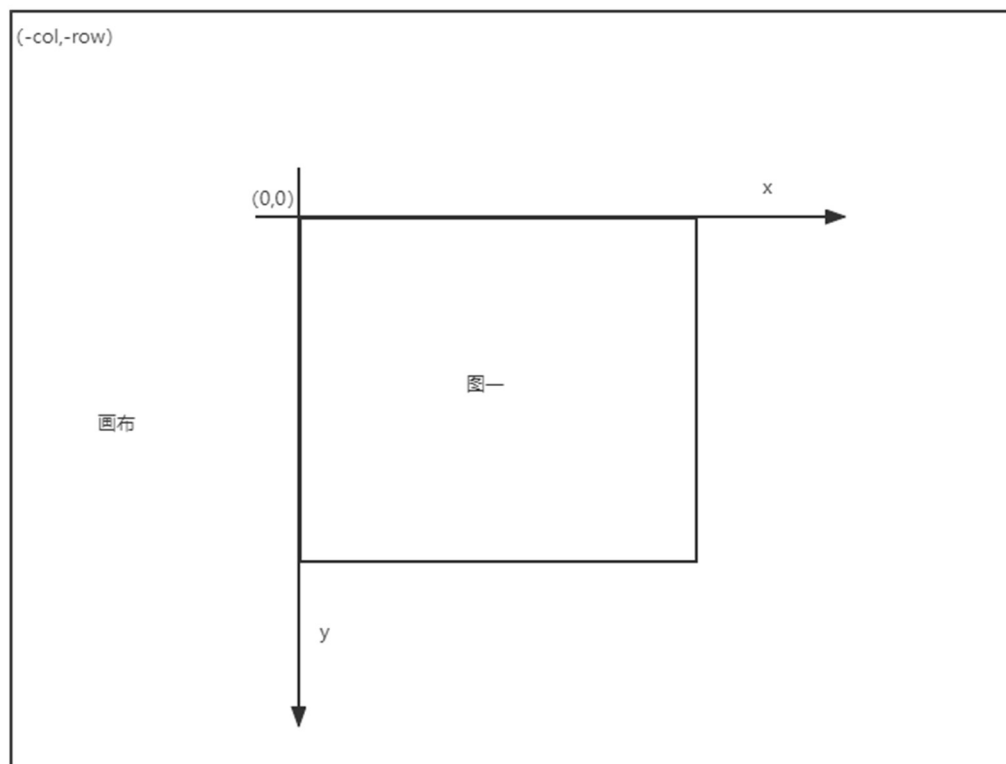
匹配的特征点



经过 RANSAC 过滤的匹配的特征点

图片的拼接

首先创建一副大画布 $img = zeros(3 * size(img_1))$ ，长宽都为图一的三倍。将图一的图片矩阵放在画布中间。然后对画布中的空白部分进行遍历，此时画布的坐标还应以图一坐标系为准。



在遍历画布中空白的点时，将每个空白点的 x 与 y 坐标构成一个向量，再与 H 运算计算出该点在图二中对应的点的坐标。

$$\begin{bmatrix} x^p \\ y^p \\ c \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

如果计算出的坐标没有超过图二的范围，就将图二对应坐标的像素值赋给画布。如此重复，直到遍历完画布，最终将画布中的全 0 行与全 0 列删去，就得到了合并后的图像。

