

图像特征匹配

Review

- 颜色特征
- 几何特征：矩，Hu矩
- 纹理特征：
 - 灰度共生矩阵：固定位置的两个像素的灰度的统计
 - LBP：图像灰度变化：中心像素和邻域像素比较
 - Haar：图像灰度变化，相邻黑白区域比较
 - HoG：分块，方向梯度直方图，本质是梯度的统计信息

特征匹配

特征匹配，又可以称之为数据关联， Feature Based Matching

在图像领域中，特征匹配的作用非常大。比如：

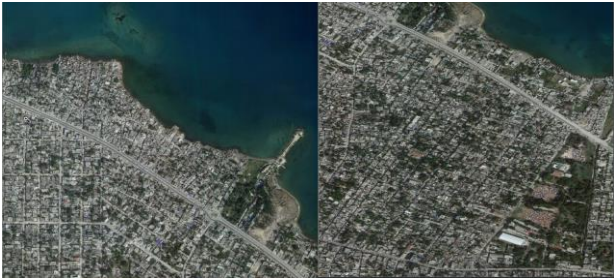
- 在图像拼接中，需要进行特征匹配，方便求出单应矩阵以拼接两幅图像；
- 在三维重建中，需要进行特征匹配，方便求出变换矩阵以及三角化特征点；
- 在图像检索中，可以通过特征点在数据库中检索，查找到特征匹配数量最多的识别结果；

特征匹配

图像拼接



特征匹配



遥感图像

特征匹配

3D重建

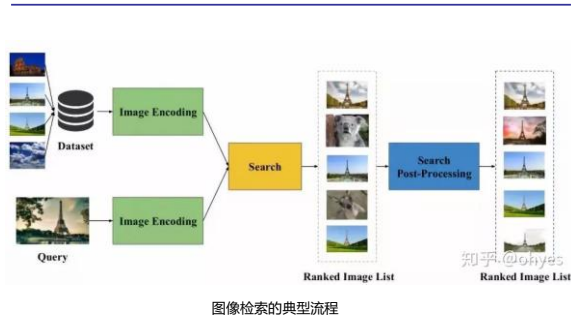


特征匹配

连连看？



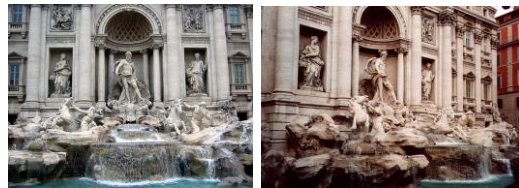
8



7

特征匹配

连连看？



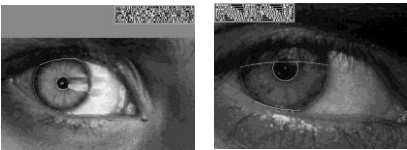
9

特征匹配

是同一个人吗？



通过虹膜对比



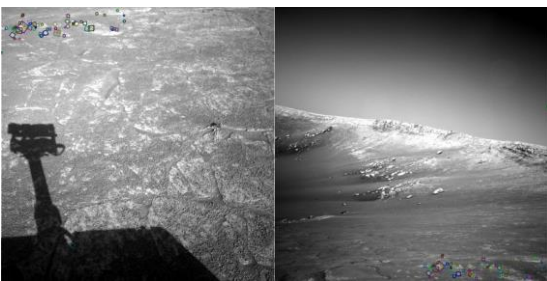
10

特征匹配



11

特征匹配



NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

12

特征匹配

只要图像中有足够多可检测的兴趣点，并且这些兴趣点各不相同且特征稳定，能被精确地定位，上述应用就十分有效。

一幅图像中总存在着其独特的像素点，这些点我们可以认为就是这幅图像的特征，成为特征点。

计算机视觉领域中的很重要的图像特征匹配就是一特征点为基础而进行的，

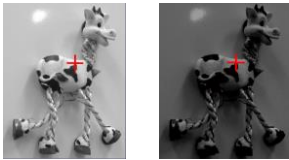
如何定义和找出一幅图像中的特征点就非常重要。

13

特征匹配

特征匹配—基本思路

最简单的描述子：一个像素点，具有不变性吗？



- 对平移、旋转、仿射、尺度都有不变性；
- 但对光照没有任何不变性。

15

特征匹配

特征匹配—基本思路

特征/准确匹配的要求是待匹配特征对平移、2D/3D旋转、光照、对比度、仿射变换等具有不变性。

这需要一个**具有不变性的特征检测子**：

- Harris：对平移、2D旋转、光照具有不变性；
- SIFT：对平移、2D旋转、光照、3D旋转（约60度）、尺度具有不变性；

还需要一个**具有不变性的特征描述子**：

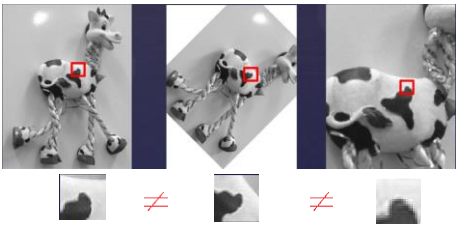
- 描述子用来记录特征点周围的区域信息；
- 描述子要有不变性：图像变化时描述子不变；

14

特征匹配

特征匹配—基本思路

另一个简单的描述子：以特征点为中心的一个小方窗区域，具有不变性吗？

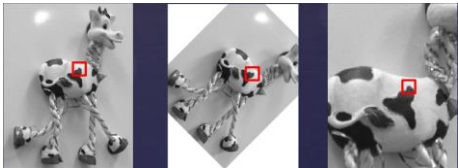


16

特征匹配

特征匹配—基本思路

另一个简单的描述子：以特征点为中心的一个小方窗区域，具有不变性吗？



- 对平移、光照（灰度归一化）有不变性；
- 但对旋转、尺度没有不变性。

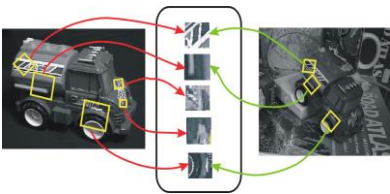
17

特征匹配

We need good features

Find **local features** that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, ...



18

特征匹配

What are good features?

Invariance Local Features:

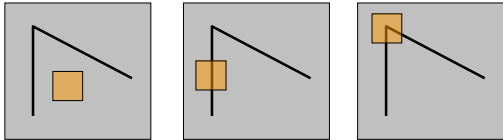
- Locality: features are local, so robust to occlusion and clutter
- Distinctiveness: can differentiate a large database of objects
- Quantity: hundreds or thousands in a single image
- Efficiency: real-time performance achievable
- Generality: exploit different types of features in different situations

19

特征匹配

Harris 角点

角点指的是两条边的交点

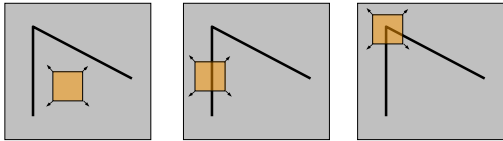


21

特征匹配

Harris 角点

角点指的是两条边的交点



基本思想: 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两种情况, 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。

23

特征匹配

Good Features

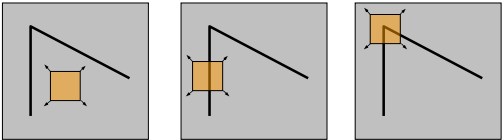
- Harris: 检测角点
- FAST: 检测角点
- SIFT: Scale-invariant feature transform, 检测斑点, 专利保护
- SURF: Speeded Up Robust Features, 检测斑点, 专利保护
- BRIEF: 检测斑点
- ORB: 带方向的FAST算法与具有旋转不变性的BRIEF算法
- 还有线 ...

20

特征匹配

Harris 角点

角点指的是两条边的交点



"flat" region:
no change in all
directions

"edge":
no change along
the edge direction

"corner":
significant change
in all directions

22

特征匹配

Harris 角点

$$c(x, y; \Delta x, \Delta y) = \sum_{u,v \in Window(x,y)} w(u,v) (I(u,v) - I(u + \Delta x, v + \Delta y))^2$$

使用泰勒展开对图像 $I(x, y)$ 在移动 $(\Delta x, \Delta y)$ 处一阶展开

$$I(u + \Delta x, v + \Delta y) = I(u, v) + I_x(u, v)\Delta x + I_y(u, v)\Delta y + O(\Delta x^2, \Delta y^2)$$

$$C(x, y; \Delta x, \Delta y) \approx \sum_{Window} (I_x(u, v)\Delta x + I_y(u, v)\Delta y)^2$$

$$c(x, y; \Delta x, \Delta y) \approx \sum_w (I_x(u, v)\Delta x + I_y(u, v)\Delta y)^2 = [\Delta x, \Delta y] M(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

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

24

特征匹配

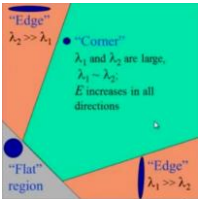
Harris 角点

使用打分函数 $R = \det(M) - k(\text{trace}(M))^2$ 来判断是否是角点,

k值一般取0.04到0.06

M的特征值:

- 当 λ_1 和 λ_2 都小时, R也小, 此区域平坦
- 其中一个远大于另外一个时, R小于0, 此区域为边缘
- 都很大的时候, R也很大, 判断为角点

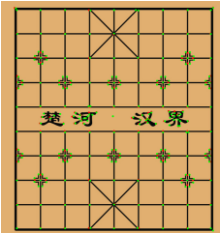


25

特征匹配

Harris 角点

- `cv2.cornerHarris`(img,blocksize,ksize,k)
- img 输入图像, 数据类型为float32
- blockSize 角点检测当中的邻域值。
- ksize 使用Sobel函数求偏导的窗口大小
- k 角点检测参数, 取值为0.04到0.06



26

特征匹配

SIFT 斑点 (main idea)

DoG: difference of Gaussian: 对一幅图像f (x, y) 进行不同参数的高斯滤波

$$g_1(x,y) = G_{\sigma_1}(x,y) * f(x,y)$$

$$g_2(x,y) = G_{\sigma_2}(x,y) * f(x,y)$$

$$g_1(x,y) - g_2(x,y) = DoG * f(x,y)$$

27

特征匹配

SIFT: main idea

高斯金字塔:

- 对图像做高斯平滑
- 对平滑后的图像做下采样
- Octave: 一组图像, 用不同的参数滤波

将相邻的两个高斯空间的图像相减即可得到DoG响应图像

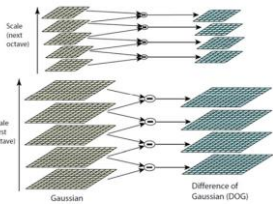


图 1 高斯金字塔和 DoG 金字塔

28

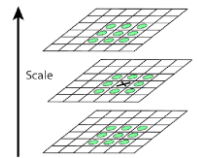
特征匹配

SIFT: main idea

DoG空间极值检测 (关键点): 关键点是一些十分突出的点, 不会因光照条件的改变而消失, 比如角点、边缘点、暗区域的亮点和亮区域的暗点

既然两幅图像中有相同的景物, 那么使用某种方法分别提取各自的稳定点, 这些点之间会有相应的匹配点。

所谓关键点, 就是在不同尺度空间的图像下检测出的具有方向信息的局部极值点



中间的检测点和它同尺度的8个相邻点和上下相邻尺度对应的9x2个点共26个点比较, 以确保在尺度空间和二维图像空间都检测到极值点

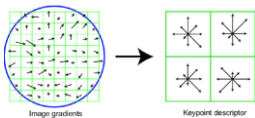
29

特征匹配

Step2: 如何产生不变的特征描述?

A descriptor captures the information in a region around the detected feature point

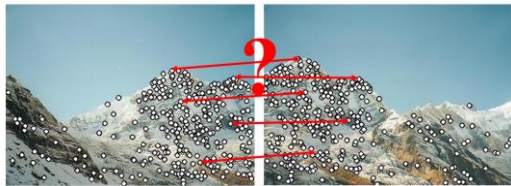
- 以特征点为中心根据主方向取一个16x16的窗口;
- 将窗口主方向旋转至水平;
- 将窗口划分成4x4个子区域;
- 计算每个子区域梯度直方图;
- 得到一个4x4x8=128维向量构成描述子。



30

特征匹配

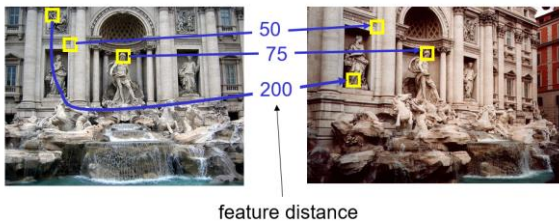
Step3: Matching Algorithm



31

特征匹配

Matching Algorithm

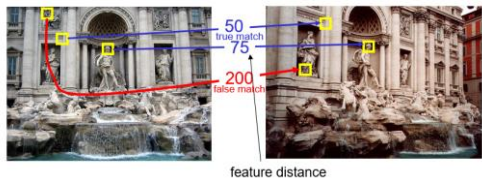


33

特征匹配

Matching Algorithm

Threshold affects the performance



35

特征匹配

Matching Algorithm

特征匹配的基本假设：可以在特征空间通过特征描述子之间的欧氏距离判断匹配程度，距离越小匹配度越高。

特征匹配基本思路：

- 1) 在特征空间定义特征描述子之间的某种距离度量函数；
- 2) 找出I2中与I1特征点距离最小的作为匹配点。

常用距离度量函数：SSD (Sum of Square Differences)
两个描述子对应值差的平方和

32

特征匹配

Matching Algorithm

用于定量衡量匹配质量的指标：

- TP: True Positive, 被判定为匹配，事实上也是匹配；
- FN: False Negative, 被判定为不匹配，**但**事实上匹配；
- FP: False Positive, 被判定为匹配，**但**事实上不匹配；
- TN: True Negative, 被判定为不匹配，事实上也是不匹配。

$TPR = TP / (TP + FN)$ ，真阳性率；
 $FPR = FP / (FP + TN)$ ，假阳性率；

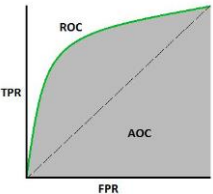
34

特征匹配

Matching Algorithm

用于定量衡量匹配质量的指标—ROC曲线, Receiver Operating Characteristic curve

- 阈值最高时，ROC曲线在 左下角；
- 阈值最低时，ROC曲线在 右上角；
- ROC曲线越靠近左上角 (ROC曲线下面积越大) 性能越好。



36

特征匹配

Matching Algorithm

Brute-force matcher (cv::BFMatcher) 暴力方法

- 找到点集1中每个描述子在点集2中距离最近的描述子
- 浮点描述子-欧氏距离；二进制描述符-汉明距离。

在第一幅图像中选取一个关键点然后依次与第二幅图像的每个关键点进行（描述符）距离测试，最后返回距离最近的关键点

37

特征匹配

Matching Algorithm

Brute-force matcher (cv::BFMatcher) 暴力方法

- 找到点集1中每个描述子在点集2中距离最近的描述子
- 浮点描述子-欧氏距离；二进制描述符-汉明距离。

```
matcher = cv2.BFMatcher_create(cv2.NORM_HAMMING, crossCheck=True)
matchePoints = matcher.match(queryDescriptors, trainDescriptors)
```

One match:

- distance:3.0
- imgIdx:0 # 目标图像的索引|
- queryIdx:1 # 查询图像中描述符的索引|
- trainIdx:0 # 目标图像中描述符的索引|

norm Type：取模方法，有四种可选：

- NORM_L1，对于SIFT和SURF描述符是较好的选择
- NORM_L2，对于SIFT和SURF描述符是较好的选择，默认
- NORM_HAMMING，应该与ORB，BRISK和BRIEF一起使用
- NORM_HAMMING2，应该与WTA_K=3或4时的ORB使用

38

特征匹配

Matching Algorithm

FLANN匹配器: Fast Library for Approximate Nearest Neighbors的简称。它是一个对大数据集和高维特征进行最近邻搜索的算法的集合，而且这些算法都已经被优化过了。在面对大数据集时它的效果要好于 BFMatcher。

```
flann = cv2.FlannBasedMatcher(index_params, search_params)
```

39

特征匹配

```
cv2.ORB_create(
    nfeatures = 500,          确定要查找的最大要素（关键点）数
    scaleFactor = 1.2,        ORB使用图像金字塔来查找要素
    nlevels = 8,              金字塔等级的数量
    edgeThreshold = 31,
    firstLevel = 0,
    WTA_K = 2,
    scoreType = HARRIS_SCORE,
    patchSize = 31,
    fastThreshold = 20
)
```

40

特征匹配

拟合模型

得到大量匹配点之后，如何拟合模型（如配准、拼接）？

解决方法：RANSAC（RANdom SAmple Consensus）

基本的假设就是数据是由“内点”和“外点”组成的。“内点”就是组成模型参数的数据，“外点”就是不适合模型的数据。同时RANSAC假设：在给定一组含有少部分“内点”的数据，存在一个程序可以估计出符合“内点”的模型。

41

特征匹配

RANSAC

1. 选择出可以估计出模型的最小数据集；（对于直线拟合来说就是两个点，对于计算Homography矩阵就是4个点）
2. 使用这个数据集来计算出数据模型；
3. 将所有数据带入这个模型，计算出“内点”的数目；（累加在一定误差范围内的适合当前迭代推出模型的数据）
4. 比较当前模型和之前推出的最好的模型的“内点”的数量，记录最大“内点”数的模型参数和“内点”数；
5. 重复1-4步，直到迭代结束或者当前模型已经足够好了（“内点数目大于一定数量”）。

42

特征匹配

通过匹配点找到旋转矩阵

```
m, mask=cv2.findHomography(src_pts,dst_pts,cv2.RANSAC,5,0)
```

通过矩阵m把旋转点集

```
dst=cv2.perspectiveTransform(pts,m)
```

通过矩阵旋转图像

```
im_out = cv2.warpPerspective(img1, m, img1.shape)
```