

创建时间: 2019/8/13 9:02

更新时间: 2019/8/24 9:23

作者: Min Xia

URL: <https://blog.csdn.net/vonzhoufz/article/details/46594369>

1: 有哪些特征描述子

[Feature Detection Methods List](#)

图像处理的基础就是要进行特征点的提取, feature(interest points) detect , 边检测, 角点检测, 直线检测, 圆检测, SIFT特征点检测, 同时描述符也在发展, 为了匹配的高效, 逐渐从高维特征向量到二进制向量等发展。主要的有以下几种, 在一般的图像处理库中(如opencv, VLFeat, Boofcv等)都会实现。

通过以下方法进行特征匹配: 暴力(Brute-Force)匹配法; 基于FLANN匹配法;

- Canny Edge Detect, [A Computational Approach to Edge Detection 1986](#), 1986. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.
- Harris, [A combined corner and edge detector, 1988](#). considering the differential of the corner score with respect to direction directly.
- GFTT, [Good Features to Track,1994](#), Determines strong corners on an image.
- Matas-2000, Robust Detection of Lines Using the Progressive Probabilistic Hough Transform. 霍夫变换检测直线.
- MSER, [Robust Wide Baseline Stereo from Maximally Stable Extremal Regions,2002](#),斑点检测
- SIFT,[Distinctive Image Features from Scale-Invariant Keypoints,2004](#), invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion
- SURF,[Speeded Up Robust Features,2006](#),受SIFT启发, 比SIFT快, 健壮
- FAST ,[Machine Learning for High-speed Corner Detection, 2006](#)
Very fast, not robust to high level noise.
- STAR, [Censure: Center surround extremas for realtime feature detection and matching 2008](#),引用次数不高
- ORB: [an efficient alternative to SIFT or SURF,2011](#), 基于FAST, 比SIFT快两个数量级,
32B binary descriptor.可作为SIFT的替代
- BRISK: [Binary Robust Invariant Scalable Keypoints](#)
64B binary descriptor.
- GFTT, [Good Features to Track,1994](#),Determines strong corners on an image

- HARRIS, [Harris and M. Stephens \(1988\). "A combined corner and edge detector"](#), 也是一种角点检测方法

下图表示对两张图片对应的的时间、找到的特征点及匹配的特征点

image pair	SIFT	SURF	ORB	FAST	BRISK
1.jpg 2.jpg	2.77	3.22	0.11	0.22	None
1.jpg 2.jpg	1639-1311- 697	2802-2606- 1243	500-500- 251	1196-1105- 586	607-491- 287

2: cv2.cornerHarris

- 角点的检测主要有两类基于图像边缘的方法和基于图像灰度的方法。前者很大程度上依赖于图像的分割和边缘提取，一旦待检测目标发生局部变化，很可能导致操作失败，因此该类方法使用范围较小；后者有很多方法，包括Harris算子，Moravec算子，Susan算子等等。
- Harris 角点算子是对Moravec角点检测算子的改进，扩展了检测方向，检测结果具有旋转不变性；对滑块窗口使用了高斯系数，对离中心越近的点赋予更高的权重，以增强对噪声的干扰；
- 角点在各个方向上的变化大小： $w(x,y)$ 窗口函数， $[u, v]$ 表示方向，及位移。公式表示 $E(u,v)$ 在某个方向上图像灰度变化。

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

- 求解方法：泰勒展开式 ($I(x+u, y+v) = I(x, y) + uI_x + vI_y + O(x, y)$)

$$E(u, v) \approx [u, v] * M * \begin{bmatrix} u \\ v \end{bmatrix}$$

- 其中

$$M = \sum_{x,y} W(x, y) \begin{bmatrix} I_x/I_x & I_x/I_y \\ I_x/I_y & I_y/I_y \end{bmatrix}$$

- 其中 I_x 和 I_y 分别为图像在X方向和Y方向上的导数。通过sobel算子求解。
- $R = \det(M) - k(\text{trace}(M))^2$, 这个函数用来确认一个窗口中是否含有角点
 $\det(M) = \lambda_1 \lambda_2$, $\text{trace}(M) = \lambda_1 + \lambda_2$ k 是一个控制参数。
 当 $|R|$ 很小，说明区域是平坦的，没有角点
 当 $|R|$ 小于零，说明 $\lambda_1 \gg \lambda_2$, 是边缘区域
 当 $|R|$ 大于零，说明 $\lambda_1 \ll \lambda_2$, 是角点区域
- 不具有尺度不变性
- 参数说明：
 blockSize - 角点检测领域大小，滑块窗口的尺寸

ksize - Sobel边缘检测滤波器大小

k - 角点检测的自由度

[python 实现 harris](#)

[原理详解](#)

3: HOG

- 分析: 假如图像大小(128, 64), stride=8, blocks= 16 * 16 pixes, cell = (8 * 8) pixes
一个检测窗口有 $((128-16)/8+1) * ((64-16)/8+1)=105$ 个Block, 一个Block有4个Cell, 一个Cell的Hog描述子向量的长度是9, 所以一个检测窗口的Hog描述子的向量长度是 $105 * 4 * 9=3780$ 维。
- step 1: Gamma correct and to gray
$$G_x(x, y) = H(x+1, y) - H(x-1, y)$$
$$G_y(x, y) = H(x, y+1) - H(x, y-1)$$

式中 $G_x(x, y)$, $G_y(x, y)$, $H(x, y)$ 分别表示输入图像中像素点(x,y)处的水平方向梯度、垂直方向梯度和像素值。像素点(x,y)处的梯度幅值和梯度方向分别为:

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$$
$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right)$$

- step 2: 计算每一个像素的方向和幅值
- step 3: 计算每一个cell 的梯度直方图
- step 4: 计算每一个block 的梯度直方图
- step 5: 归一化, 所有特征点

[HOG 从理论到实现](#)

[HOG实现](#)

4: classical image stitching

- SIFT+RANSAC

[SIFT descriptions](#)

[角点获取的各种算法介绍](#)

[SURF算法进行图像拼接](#)