**目标：**

* 了解FAST算法的基础知识
* 使用OpenCV功能为FAST算法找到角点

## api：

### fast = cv.FastFeatureDetector\_create()

### kp = fast.detect(img,None)

### # Disable nonmaxSuppression

### fast.setNonmaxSuppression(0)

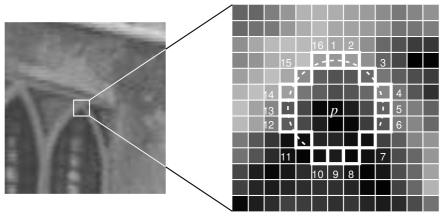
**理论**

我们看到了几个特征探测器，其中很多都非常好用。但从实时应用程序的角度来看，它们还不够快。一个最好的例子是具有有限计算资源的SLAM（同步构建与地图定位）移动机器人。

为了解决这个问题，Edward Rosten 和 Tom Drummond 在 2006 年提出里 FAST 算法。我们下面将会对此算法进行一个简单的介绍。你可以参考原始文献获得更多细节（本节中的所有图像都是曲子原始文章）。

**使用 FAST 算法进行特征提取**

1. 在图像中选取一个像素点 p，来判断它是不是关键点。$I\_p$等于像素点 p的灰度值。
2. 选择适当的阈值 t。
3. 如下图所示在像素点 p 的周围选择 16 个像素点进行测试。

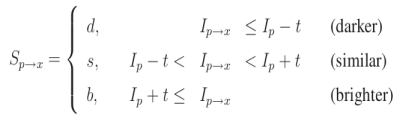
[](https://camo.githubusercontent.com/d7a0ecc40572870bbd1e02cc3031d0f961b7659e19a82fc2228bf3359aa38ab0/68747470733a2f2f646f63732e6f70656e63762e6f72672f342e302e302f666173745f7370656564746573742e6a7067)

1. 如果在这 16 个像素点中存在 n 个连续像素点的灰度值都高于$I\_p + t$，或者低于$I\_p - t$，那么像素点 p 就被认为是一个角点。如上图中的虚线所示，n 选取的值为 12。
2. 为了获得更快的效果，还采用了而外的加速办法。首先对候选点的周围每个 90 度的点：1，9，5，13 进行测试（先测试 1 和 19, 如果它们符合阈值要求再测试 5 和 13）。如果 p 是角点，那么这四个点中至少有 3 个要符合阈值要求。如果不是的话肯定不是角点，就放弃。对通过这步测试的点再继续进行测试（是否有 12 的点符合阈值要求）。这个检测器的效率很高，但是它有如下几条缺点：
   * 当 n<12 时它不会丢弃很多候选点 (获得的候选点比较多)。
   * 像素的选取不是最优的，因为它的效果取决与要解决的问题和角点的分布情况。
   * 高速测试的结果被抛弃
   * 检测到的很多特征点都是连在一起的

前 3 个问题可以通过机器学习的方法解决，最后一个问题可以使用非最大值抑制的方法解决。

**机器学习的角点检测器**

1. 选择一组训练图片（最好是跟最后应用相关的图片）
2. 使用 FAST 算法找出每幅图像的特征点
3. 对每一个特征点，将其周围的 16 个像素存储构成一个向量。对所有图像都这样做构建一个特征向量 P
4. 每一个特征点的 16 像素点都属于下列三类中的一种。

[](https://camo.githubusercontent.com/bf25cb1087dab7e996b9f10b9325b96ba8bf517e2b86df75d59c695fa27adbde/68747470733a2f2f646f63732e6f70656e63762e6f72672f342e302e302f666173745f65716e732e6a7067)

1. 根据这些像素点的分类，特征向量 P 也被分为 3 个子集：P d ，P s ，P b
2. 定义一个新的布尔变量$K\_p$，如果 p 是角点就设置为 Ture，如果不是就设置为 False。
3. 使用 ID3 算法（决策树分类器）使用变量$K\_p$查询每个子集，以获得有关真实类的知识。它选择x，其产生关于候选像素是否是拐角的最多信息，通过$K\_p$的熵测量。
4. 这递归地应用于所有子集，直到其熵为零。
5. 将构建好的决策树运用于其他图像的快速的检测。

**非极大值抑制**

使用极大值抑制的方法可以解决检测到的特征点相连的问题

1. 对所有检测到到特征点构建一个打分函数 V。V 就是像素点 p 与周围 16个像素点差值的绝对值之和。
2. 计算临近两个特征点的打分函数 V。
3. 忽略 V 值最低的特征点

**总结**

FAST 算法比其它角点检测算法都快。但是在噪声很高时不够稳定，这是由阈值决定的。

**OpenCV中的FAST特征检测器**

它被称为OpenCV中的任何其他特征检测器。如果需要，你可以指定阈值，是否应用非最大抑制，要使用的邻域等。

对于邻域，定义了三个标志，cv.FAST\_FEATURE\_DETECTOR\_TYPE\_5\_8，cv.FAST\_FEATURE\_DETECTOR\_TYPE\_7\_12和cv.FAST\_FEATURE\_DETECTOR\_TYPE\_9\_16。

## 实例

|  |
| --- |
|  |

### 效果

|  |
| --- |
|  |