**目标：**

* 我们将看到ORB算法的基础知识

## api：

### orb = cv.ORB\_create()

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

### # compute the descriptors with ORB

### kp, des = orb.compute(img, kp)

**理论**

作为OpenCV爱好者，关于ORB最重要的是它来自“OpenCV Labs”。这个算法由Ethan Rublee，Vincent Rabaud，Kurt Konolige和Gary R. Bradski在他们的论文ORB中提出：2011年是SIFT或SURF的有效替代方案。如标题所述，它是计算中SIFT和SURF的一个很好的替代方案。成本，匹配性能和主要是专利。是的，SIFT和SURF已获得专利，你应该支付它们的使用费用。但ORB不是!!!

ORB基本上是FAST关键点检测器和Brief描述符的融合，具有许多修改以增强性能。首先，它使用FAST查找关键点，然后应用Harris角点测量来查找其中的前N个点。它还使用金字塔来生成多尺度特征。但有一个问题是，FAST不计算方向。那么旋转不变性呢？作者提出了以下修改。

它计算贴片的强度加权质心，位于中心的角落。从该角点到质心的矢量方向给出了方向。为了改善旋转不变性，用x和y计算矩，该x和y应该在半径为r的圆形区域中，其中r是贴片的大小。

现在对于描述符，ORB使用简要描述符。但我们已经看到，Brief在轮换方面表现不佳。因此，ORB所做的是根据关键点的方向“引导”Brief。对于位置$(x\_i, y\_i)$处的n个二进制测试的任何特征集，定义2×n矩阵，其包含这些像素的坐标。然后使用贴片的方向$\theta$，找到其旋转矩阵并旋转S以获得转向（旋转）版本$S\_\theta$。

ORB将角度离散为$2 \pi /30$（12度）的增量，并构建预先计算的简要模式的查找表。只要关键点方向θ在视图之间是一致的，将使用正确的点集$S\_\theta$来计算其描述符。

BRIEF具有一个重要特性，即每个位特征具有较大的方差，平均值接近0.5。但是一旦它沿着关键点方向定向，它就会失去这个属性并变得更加分散。高差异使得特征更具辨别力，因为它对输入有不同的响应。另一个理想的特性是使测试不相关，因为每次测试都会对结果产生影响。为了解决所有这些问题，ORB在所有可能的二进制测试中运行一个贪婪的搜索，以找到具有高方差和意味着接近0.5的那些，以及不相关的。结果称为rBRIEF。

对于描述符匹配，使用改进传统LSH的多探测LSH。该论文称ORB比SURF快得多，SIFT和ORB描述符比SURF更好。ORB是用于全景拼接等的低功率设备的不错选择。

**OpenCV中的ORB**

像往常一样，我们必须使用函数cv.ORB()或使用feature2d公共接口创建一个ORB对象。它有许多可选参数。最有用的是nFeatures，表示要保留的最大要素数量（默认为500），scoreType表示Harris得分或FAST得分是否对要素进行排名（默认情况下为Harris得分）等。另一个参数WTA\_K决定点数 生成面向简要描述符的每个元素。 默认情况下它是2，即一次选择两个点。在这种情况下，为了匹配，使用NORM\_HAMMING距离。如果WTA\_K为3或4，需要3或4个点来产生BRIEF描述符，则匹配距离由NORM\_HAMMING2定义

## 实例

|  |
| --- |
|  |

### 效果

|  |
| --- |
|  |