

## 原 ORB\_SLAM2学习（一）：ORB算法原理

2018年05月18日 16:18:15 Mega\_Li 阅读数 7842 更多

### 前言

目前学习ORB\_SLAM2中，ORB\_SLAM2中使用ORB算子进行特征点的提取与描述，ORB算法原理主要来自于文章《ORB: an efficient alternative to SURF》。这里先就这篇文章做自己的学习过程记录，之后结合文章内容分析ORB\_SLAM2中的代码实现（放到下一篇博客中）。本文把文章《ORB: an efficient alternative to SIFT or SURF》中的涉及到原理知识的部分做了翻译，同时增加了一些个人理解注释。

## 1. 背景

特征匹配是计算机视觉应用场景中的一个基础问题，譬如在物体识别和SFM中。当前的一些方法依赖于计算量很大的描述子计算来进行检测和匹配。作者提出一种基于BRIEF的能够快速运行的二进制描述子计算方法，具有旋转不变性且对噪声有较强的抗干扰能力。通过实验**作者表明该算法在计算时间上要快两个数量级，同时保持相似的检测效果！**

## 2. 算法简介

十几年前提出的SIFT关键点检测与特征描述子算法在使用视觉特征计算的众多应用中已经取得了极大的成功，包括物体识别、图像拼接、视觉地图算法。计算量很大，在实时系统如视觉里程计或者低运算能力的平台如手机上实现有很大困难。因此人们开始研究具有更低运算量的其他算法，其中SURF算法；也有其他人尝试对SIFT进行优化加速，譬如使用GPU。

作者提出一种新的算法，相比于SIFT它具有相似的匹配能力，运算量更小，受噪声的影响更小，能够用于实时运算。作者的主要目的是将该算法应用到各种图像处理场景中，譬如用没有GPU的低功耗设备实现全景拼接和图像跟踪，或者减少在PC上执行基于特征的物体检测运算时间。在这些应用中，和SIFT一样好（优于SURF），同时运行速度比SIFT快两个数量级。

ORB算法基于FAST角点检测算法和BRIEF描述子，因此称为ORB（Oriented FAST and Rotated BRIEF）。这两种方法都具有高性能低运算量的特点。作者把它们和SIFT做对比说明其局限性，特别是BRIEF不具备旋转不变性。论文主要的创新点在于：

- 1) 在FAST算法中增加了快速准确的方向计算
- 2) Steered BRIEF中特征的高效计算
- 3) 对于steered BRIEF描述子的方差和相关性的分析
- 4) 一种在实现旋转不变性时降低BRIEF描述子中各位之间相关度的启发式方法，使得算法在nearest-neighbor应用中具有更好的性能。

作者设计实验比较ORB、SIFT、SURF在图像匹配应用中的表现，同时通过在手机上运行一个图像跟踪的程序来说明其运行效率。ORB的另一个优点是SIFT和SURF它不受licensing限制。

## 3. 相关工作

### 3.1 关键点

FAST算法和它的改进版本可用于在图像特征实时匹配系统中提取关键点，譬如同时跟踪与建图的应用中。FAST算法效率很高且能够获得比较好的结果。**需要配合金字塔图像算法来达到尺度不变性，作者使用Harris响应来滤除边缘影响并提供一个用于评判关键点质量的score。**

许多关键点描述子都包含一个旋转操作（譬如SIFT和SURF），不过FAST没有。有不同的方法来计算关键点的方向，譬如通过梯度直方图（SIFT中）或块纹理的近似（SURF中）。这些方法要么计算量很大，要么像SURF中的那样近似效果不好。**作者使用强度质心法计算方向**，不像SIFT可能在单个关键点方向，强度质心法旋转操作只会得到一个方向。

### 3.2 描述子

BRIEF是一种在平滑后的图像上不同像素间使用简单的二进制操作计算描述子的方法。与SIFT相比它在对光照、模糊、透视失真等方面的鲁棒性都较好。不过它对于平面旋转十分敏感，具体内容文章后面会介绍。

## 4. FAST算子

首先简单介绍下FAST算子（**具体内容可参考<https://blog.csdn.net/lwx309025167/article/details/80234307>**），该算法只有一个参数，圆形邻域内其他点灰度值之间的差值阈值。作者使用FAST-9算法（圆形邻域半径为9），具有不错的性能。（感觉这里对于FAST-9的说法有些问题，应该是选择连续9个点与中心点比较）。

FAST不会给出角点的“评价价值”，且我们发现它对于边缘也有很强的响应。作者使用Harris运算计算角点的“评价价值”并排序（**有关Harris运算可参考<https://blog.csdn.net/lwx309025167/article/details/80236516>**）。假如有N个关键点，作者首先把FAST算法中的阈值设得低一些，获得多

根据Harris计算的“评价” 排序，取前N个点。

5. 质心法计算旋转角度

作者使用了一种简单有效的方法计算角点的方向（方向主要用于实现特征描述子的旋转不变性）——强度质心法。一般一个图像块的强度质心是偏的，我们可以用这个偏离计算得到一个方向。Rosin定义了如下的图像矩：

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y)$$

由此我们得到图像的强度质心：

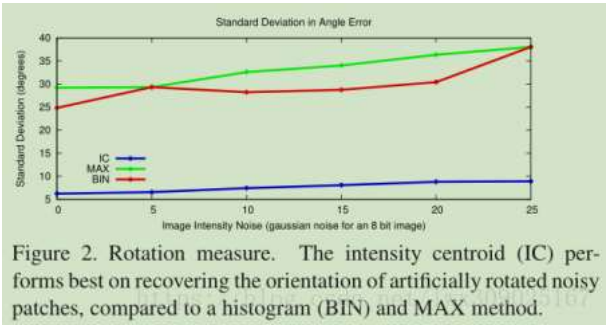
$$C = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

之后得到从角点所在图像块中心O指向强度质心C的向量  $\vec{OC}$ ，之后计算得到角点的角度

$$\theta = \text{atan2}(m_{01}, m_{10})$$

为了提高这种计算的旋转不变性（对旋转的不敏感程度）作者划定了一个半径为r的圆形区域计算上述图像矩（个人理解是圆形区域旋转后区域范性），我们把r叫做patch size。这里当O和C十分接近时，计算角度变得不太稳定，不过作者实际测试时发现这种情况比较少见。

之后作者把这种方法和另外两种基于梯度的方法BIN、MAX做了比较（这两种方法我都没接触过，这里就不翻译了，免得误人子弟），结果表明在加了高斯噪声的图像，计算其实际旋转角度方面，强度质心法要优于上述两种方法。比较结果如下图所示。



6. rBRIEF

作者首先介绍Steered BRIEF，说明如何计算它，同时告诉大家这个描述子在图像发生旋转时表现还不够优秀。之后引入改进的方法，通过寻找相二进制算子来得到更好的描述子，称为rBRIEF，并把它和SIFT和SURF做了测试对比。

6.1 BRIEF的高效旋转计算

简要介绍下BRIEF，BRIEF是对关键点所在的一个区域做一系列的二进制操作后得到的一个二进制字符串描述子。一个二进制操作如下定义：

$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & : \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & : \mathbf{p}(\mathbf{x}) \geq \mathbf{p}(\mathbf{y}) \end{cases}$$

其中p(x)代表在x处的强度p，因此描述子就是n个二进制操作结果的组合，如下所示：

$$f_n(\mathbf{p}) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i)$$

注意这里对于挑选比较对 (x,y) 并没有给出明确定义，因此我们有很多种挑选方法（或者说分布），有文章测试了不同的挑选方法的测试结果，表现还不错的挑选方法——高斯分布，同时作者定义了描述子长度为256。

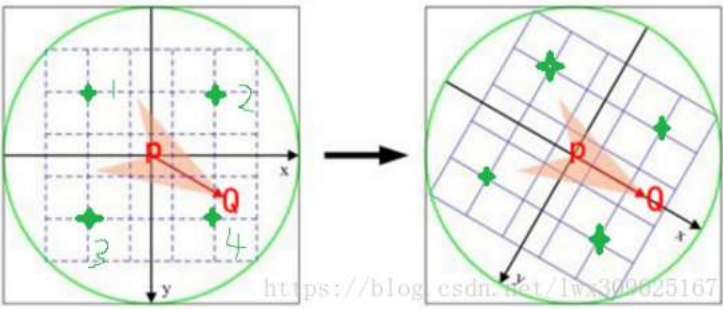
作者说在计算描述子之前需要对图像做平滑操作（应该是减少噪声的影响），作者通过定义p(x)不是点x的灰度，而是x为中心的5x5的一个图像块减少噪声的影响，也算是一种平滑操作。

6.2 Steered BRIEF

之前花力气计算的角点旋转方向这里可以发挥作用了，原始版本的BRIEF在图像发生旋转时其效果就会大打折扣（同一幅图像旋转了，结果肯定会有偏差，肯定会后面的匹配计算产生不好的影响）。图像旋转时，每个关键点所在图像块中的强度质心也会跟着旋转，由此我们可以借用它来定义坐标方向，做到这个坐标方向会随着图像旋转而旋转。这样我们就能够做到图像旋转后，根据同一规则选出的二进制操作比较对仍然不变，这样结果就会一致了，也就是说实现了旋转不变性。

举例如下，对图中左边图像的关键点p，我们使用强度质心法得到Q，向量方向就是该关键点方向；然后根据BRIEF，我们选取 (1,2) 和 (3,4) 计算描述子。之后图像发生了旋转变为右边图像，此时我们在选取 (1,2) 和 (3,4) 时对应的坐标系也要做一定的旋转，才能保证得到的描述子与左边

标系旋转的大小和方向的变化是相关的。



对于长度为n的描述子，我们需要挑选得到n个匹配对，定义为2xn的矩阵

$$S = \begin{pmatrix} x_1, \dots, x_n \\ y_1, \dots, y_n \end{pmatrix}$$

得到了关键点的旋转方向后可以得到对应的旋转矩阵，之后构造一个旋转的匹配对矩阵

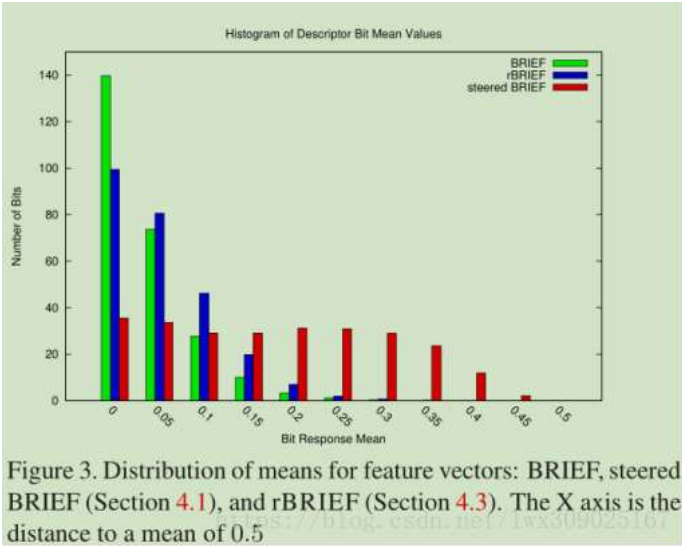
$$S_\theta = R_\theta S$$

此时的steered BRIEF操作如下

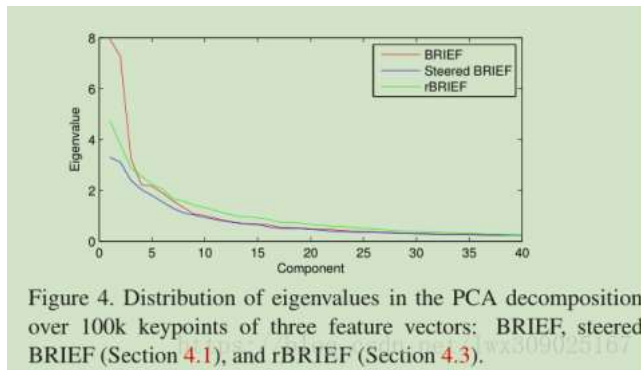
$$g_n(p, \theta) := f_n(p) | (x_i, y_i) \in S_\theta$$

6.3 Variance and Correlation

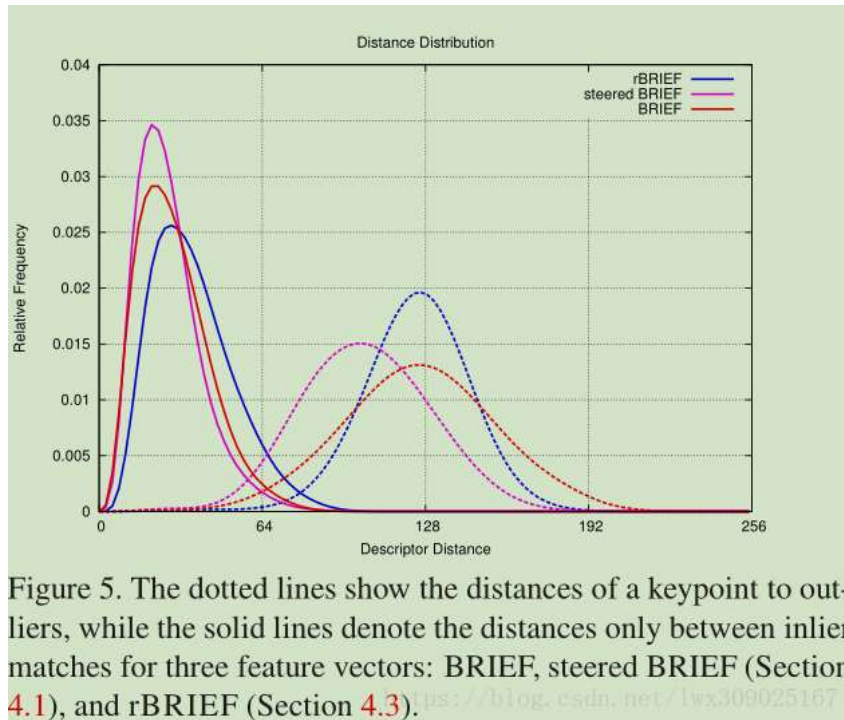
BRIEF中一个好的特性就是对于描述子中的每一位，它的均值接近0.5且方差很大。作者通过一个实验说明了该点，作者对100K个关键点都求取了BRIEF描述子，我们把它看做为100K X 256的矩阵，之后把每一列看做一组数组（只有0和1两种分布），对这组数据计算其均值和方差。一个数学规律接近0.5，方差越大，方差大的意义在于这组数据的分布更有区分度，也就是说这一位描述子更具有区分度，对于后面基于描述子的匹配计算有更好的帮助。从图中我们可以看到BRIEF中大部分位的均值都接近0.5，但是steered BRIEF（基于高斯分布选取匹配对）中位位的均值反而变得平均了，这会导致每个位的区分度小了，不利于后面的匹配计算。



同时，作者提出了相关性方面的要求，虽然每个描述子由256位组成，但是我们无法保证每个位都能做出很大的贡献（都具有很好的区分度），因存在相关性。然后作者使用PCA（主成分分析，相关原理可参考<https://www.cnblogs.com/hadoop2015/p/7419087.html>）对上述的100K X 256矩阵作，结果表明对于BRIEF和steered BRIEF来说，描述子中都存在位之间相关性的情况，结果上来看只有40位是最具有代表性的。不过相对于steered BRIEF的协方差对应的特征值更大一些（表明对应特征向量上的分布方差更大，更具有区分度），也算是再一次佐证了steered BRIEF在区分度上相对较差。



作者又使用另一种方法证明这种说法（真的严谨），就是把基于描述子匹配计算后产生的匹配成功点之间的距离（基于描述子的距离，譬如汉明匹配点之间的距离统计了一下，如图所示。结果表明steered BRIEF中匹配点和非匹配点之间的距离均值差距变小了。



## 6.4 Learning Good Binary Features

前面分析对比了半天，目的是提出：对于steered BRIEF，我们要尽量提高其描述子各个位的方差，同时也要尽量减少各个位之间的相关性。当然PCA，不过由于对比对的选择空间十分之大，PCA计算量太大了。作者用的是一种基于启发式规则的贪心搜索算法。

作者从一个图像测试集中提取出300K个关键点作为测试集，选取31X31的图像块为对比对选取范围，5X5的大小来计算该点的强度累加值，那么就是5 X (31-5) 个选择，从其中选择两个作为对比，同时去除对比对之间干涉的情况，就存在205590种对比对的选择可能。目标是要从中选取256组对的方差尽量大，且之间的相关性尽量低。作者提出的算法如下：

Step1.对于各个关键点根据各个对比对进行二进制操作计算；结果可以看为一个300K X 205590的矩阵，每个元素只有0/1的分布。

Step2.计算矩阵每一列的均值，根据fabs（均值-0.5）从小到大排列各个列（实际上是排列各个对比对），得到向量T

Step3.贪心搜索：

- (1) 把T中的第一个对比对取出放入R
- (2) 从T中取出一个对比对，并计算其代表的序列（大小为1 X 300K）和R中各个对比对对应的序列（大小也是1 X 300K）计算其相关度，如果相关度超过阈值，则丢弃它，否则从T中取出加入R
- (3) 重复第（2）步直到R中累积了256个对比对。如果最终少于256，则可以把（2）中的阈值提高再计算一次

最终得到了一个固定的256个对比对，一般在ORB实现算法中使用数组保存为全局变量。作者把使用这种方法计算得到的对比对序列应用到rBRIEF。作者同时证明了相对于steered BRIEF，rBRIEF在方差、相关度上的优势，同时也具备了steered BRIEF的旋转不变性。

之后作者把ORB和SIFT、SURF做了各种对比，结果表明大部分情况下ORB能够在表现不输的前提下计算时间比SIFT低两个数量级，这部分了。

其他

- (1) 文中关于ORB描述子尺度不变性的实现没有做详细阐述（原理就是构造金字塔图像序列然后提取各个图像上的ORB特征）
- (2) ORB\_SLAM2中ORB的实现中一些部分和论文中有稍许的不同，基本原理没变。
- (3) 有关ORB\_SLAM2中ORB的实现分析，放到下一篇博客中讲解。。。

