哈爾濱工業大學

数字媒体技术

实验报告

题　　目	Oriented FAST and Rotated BRIEF
学　　院	计算机科学与技术
专　　业	大数据
学　　号	L170300901
学　　生	卢兑玳
任 课 教 师	

哈尔滨工业大学计算机科学与技术学院

2021.3

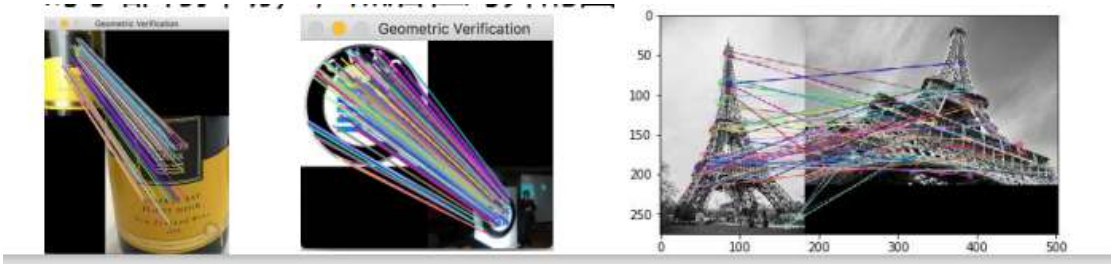
实验二:Oriented FAST and Rotated BRIEF

注意：请按照大家阅读文献的格式进行撰写，确保文档格式的规范性！

一、 实验内容或者文献情况介绍

用手机拍摄任意一段视频，然后对其中的图像进行 SIFT、ORB、HOG 特征提取，并对其中的主要目标进行对齐操作。

- 选择某个目标，从手机或别的摄像机设备对其进行多角度、多距离的拍摄（可以是桌上的某个瓶子或者杯子等常见设备，蹲下拍摄一只蚂蚁爬行的场景，宠物运动的场景，也可以是远场景，如站在天桥上拍摄一段马路上的车流，在寝室拍摄一段地面的视频，在教师拍摄一段楼下人群行走的视频，或车辆的视频等）
- 采用 opencv 等基本平台提取视频帧进行播放控制
- 对视频中的任意两帧，对其中的同一个物体进行特征提取（例如 SIFT 特征，或者 ORB 特征，或者 HOG 特征等，都可以调用现成的特征函数），然后根据特征进行对齐（如随机一致同意 RANSAC，或与 ORB SLAM2 里面类似），注意其中的目标可能有仿射变换，或者别的运动造成的位置、尺度等的变化。（就如视频中有目标移动，或者拍摄时手部有抖动），然后在对齐的图像之间画上直线



在上述匹配的基础上，继续实现以下功能

- 两幅图像上多个目标匹配，然后确认哪些目标出现了异常，例如，停车场上，哪些位置的车辆发生了变化，港口中，哪些船舶发生了移动，几场上，哪些飞机飞走了或者移动了
- 视频中，连续视频帧中运动目标的移动有多少个像素，每个移动目标都检测出来其相对位移，然后估计其相对速度（拍摄者和运动目标的相对速度）
- 在同一幅图像中，尝试用这种匹配的思想来检测复制粘贴操作（将图像的某一个区域拷贝粘贴到同一图像的不同位置，来形成多个目标）

二、 算法简介及其实现细节

1. ORB 的算法原理

ORB 特征是将 FAST 特征点的检测方法与 BRIEF 特征描述子结合起来，并在

它们原来的基础上做了改进与优化。

首先，它利用 FAST 特征点检测的方法来检测特征点，然后利用 Harris 角点的度量方法，从 FAST 特征点从挑选出 Harris 角点响应值最大的 N N 个特征点。其中 Harris 角点的响应函数定义为：

R=detM－α(traceM) 2 R=detM－α(traceM)2

关于 M M 的含义和响应函数的由来可以参考 Harris 角点检测这篇文章。

1.1 旋转不变性

我们知道 FAST 特征点是没有尺度不变性的，所以我们可以通过构建高斯金字塔，然后在每一层金字塔图像上检测角点，来实现尺度不变性。那么，对于局部不变性，我们还差一个问题没有解决，就是 FAST 特征点不具有方向，ORB 的论文中提出了一种利用灰度质心法来解决这个问题，灰度质心法假设角点的灰度与质心之间存在一个偏移，这个向量可以用于表示一个方向。对于任意一个特征点 p p 来说，我们定义 p p 的邻域像素的矩为：

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

其中 I(x,y) I(x,y)为点(x,y) (x,y)处的灰度值。那么我们可以得到图像的质心为：

C=(m 10 m 00 ,m 01 m 00) C=(m10m00,m01m00)

那么特征点与质心的夹角定义为 FAST 特征点的方向：

θ=arctan(m 01 ,m 10) θ=arctan(m01,m10)

为了提高方法的旋转不变性，需要确保 x x 和 y y 在半径为 r r 的圆形区域内，即 x,y∈[−r,r] x,y∈[−r,r]，r r 等于邻域半径。

1.2 特征点的描述

ORB 选择了 BRIEF 作为特征描述方法，但是我们知道 BRIEF 是没有旋转不变性的，所以我们需要给 BRIEF 加上旋转不变性，把这种方法称为“Steer BREIF”。对于任何一个特征点来说，它的 BRIEF 描述子是一个长度为 n n 的二值码串，这个二值串是由特征点周围 n n 个点对（2n 2n 个点）生成的，现在我们将这 2n 2n 个点(x i ,y i),i=1,2,…,2n (xi,yi),i=1,2,…,2n 组成一个矩阵 S S

S=(x 1 y 1 x 2 y 2 …… x 2n y 2n) S=(x1x2…x2ny1y2…y2n)

Calonder 建议为每个块的旋转和投影集合分别计算 BRIEF 描述子，但代价昂贵。ORB 中采用了一个更有效的方法：使用邻域方向 θ θ 和对应的旋转矩阵 R θ R θ，构建 S S 的一个校正版本 S θ S θ

S θ =R θ S S θ =R θ S

其中

R θ =[cos θ −sin θ sin θ cos θ] R θ =[cos θ sin θ −sin θ cos θ]

而 θ θ 即我们在 1.2 中为特征点求得的主方向。

实际上，我们可以把角度离散化，即把 360 度分为 12 份，每一份是 30 度，然后我们对这个 12 个角度分别求得一个 S θ S θ，这样我们就创建了一个查找表，对于每一个 θ θ，我们只需查表即可快速得到它的点对的集合 S θ S θ。

1.3 解决描述子的区分性

BRIEF 令人惊喜的特性之一是：对于 n n 维的二值串的每个比特位，所有特征点在该位上的值都满足一个均值接近于 0.5，而方差很大的高斯分布。方差越大，说明区分性越强，那么不同特征点的描述子就表现出来越大差异性，对匹配来说不容易误配。但是当我们把 BRIEF 沿着特征点的方向调整为 Steered BRIEF 时，均值就漂移到一个更加分散式的模式。可以理解为有方向性的角点关键点对二值串则展现了一个更加均衡的表现。而且论文中提到经过 PCA 对各个特征向量进行分析，得知 Steered BRIEF 的方差很小，判别性小，各个成分之间相关性较大。

为了减少 Steered BRIEF 方差的亏损，并减少二进制码串之间的相关性，ORB 使用了一种学习的方法来选择一个较小的点对集合。方法如下：

首先建立一个大约 300k 关键点的测试集，这些关键点来自于 PASCAL2006 集中的图像。

对于这 300k 个关键点中的每一个特征点，考虑它的 31×31 31×31 的邻域，我们将在这个邻域内找一些点对。不同于 BRIEF 中要先对这个 Patch 内的点做平滑，再用以 Patch 中心为原点的高斯分布选择点对的方法。ORB 为了去除某些噪声点的干扰，选择了一个 5×5 5×5 大小的区域的平均灰度来代替原来一个单点的灰度，这里 5×5 5×5 区域内图像平均灰度的计算可以用积分图的方法。我们知道 31×31 31×31 的 Patch 里共有 N=(31−5+1)×(31−5+1) N=(31−5+1)×(31−5+1)个这种子窗口，那么我们要 N N 个子窗口中选择 2 个子窗口的话，共有 C 2 N CN2 种方法。所以，对于 300k 中的每一

个特征点，我们都可以从它的 31×31 31×31 大小的邻域中提取出一个很长的二进制串，长度为 M=C 2 N M=CN2，表示为

binArray=[p 1 ,p 2 ,…,p M],p i ∈{0,1} binArray=[p1,p2,…,pM],pi ∈{0,1}

那么当 300k 个关键点全部进行上面的提取之后，我们就得到了一个 300k×M 300k×M 的矩阵，矩阵中的每个元素值为 0 或 1。

对该矩阵的每个列向量，也就是每个点对在 300k 个特征点上的测试结果，计算其均值。把所有的列向量按均值进行重新排序。排好后，组成了一个向量 T T，T T 的每一个元素都是一个列向量。

进行贪婪搜索：从 T T 中把排在第一的那个列放到 R R 中，T T 中就没有这个点对了测试结果了。然后把 T T 中的排下一个的列与 R R 中的所有元素比较，计算它们的相关性，旭果相关超过了某一事先设定好的阈值，就扔了它，否则就把它放到 R R 里面。重复上面的步骤，只到 R R 中有 256 个列向量为止。如果把 T T 全找完也，也没有找到 256 个，那么，可以把相关的阈值调高一些，再重试一遍。

这样，我们就得到了 256 个点对。上面这个过程我们称它为 rBRIEF。

三、 实验设置及结果分析（包括实验数据集）

OpenCV 中的 ORB：

ORB 中有很多参数可以设置，在 OpenCV 中它可以通过 ORB 来创建一个 ORB 检测器。

```
ORB::ORB(int nfeatures=500, float scaleFactor=1.2f, int nlevels=8, int edgeThreshold=31,
int firstLevel=0, int WTA_K=2, int scoreType=ORB::HARRIS_SCORE, int patchSize=31)
```

下面介绍一下各个参数的含义：

nfeatures - 最多提取的特征点的数量；

scaleFactor - 金字塔图像之间的尺度参数，类似于 SIFT 中的 k k；

nlevels - 高斯金字塔的层数；

edgeThreshold - 边缘阈值，这个值主要是根据后面的 patchSize 来定的，靠近边缘 edgeThreshold 以内的像素是不检测特征点的。

firstLevel - 看过 SIFT 都知道，我们可以指定第一层的索引值，这里默认

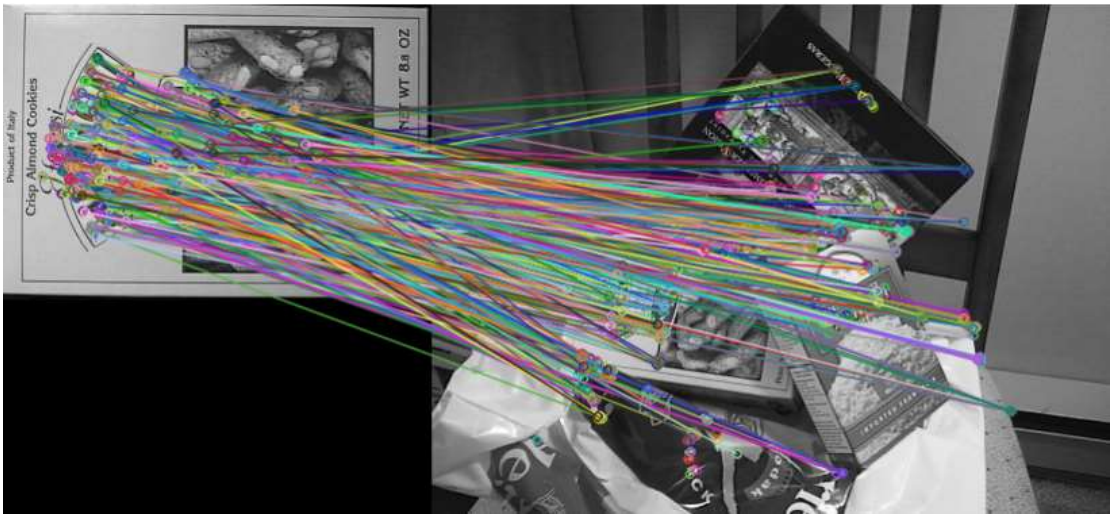
为 0。

WET_K - 用于产生 BIREF 描述子的 点对的个数，一般为 2 个，也可以设置为 3 个或 4 个，那么这时候描述子之间的距离计算就不能用汉明距离了，而是应该用一个变种。OpenCV 中，如果设置 WET_K = 2，则选用点对就只有 2 个点，匹配的时候距离参数选择 NORM_HAMMING，如果 WET_K 设置为 3 或 4，则 BIREF 描述子会选择 3 个或 4 个点，那么后面匹配的时候应该选择的距离参数为 NORM_HAMMING2。

scoreType - 用于对特征点进行排序的算法，你可以选择 HARRIS_SCORE，也可以选择 FAST_SCORE，但是它也只是比前者快一点点而已。

patchSize - 用于计算 BIREF 描述子的特征点邻域大小。

```cpp
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/features2d/features2d.hpp>

using namespace cv;

int main(int argc, char** argv)
{
    Mat img_1 = imread("box.png");
    Mat img_2 = imread("box_in_scene.png");

    // -- Step 1: Detect the keypoints using STAR Detector
    std::vector<KeyPoint> keypoints_1,keypoints_2;
    ORB orb;
    orb.detect(img_1, keypoints_1);
    orb.detect(img_2, keypoints_2);

    // -- Stpe 2: Calculate descriptors (feature vectors)
    Mat descriptors_1, descriptors_2;
    orb.compute(img_1, keypoints_1, descriptors_1);
    orb.compute(img_2, keypoints_2, descriptors_2);

    //-- Step 3: Matching descriptor vectors with a brute force matcher
    BFMatcher matcher(NORM_HAMMING);
    std::vector<DMatch> mathces;
    matcher.match(descriptors_1, descriptors_2, mathces);
    // -- dwaw matches
    Mat img_mathes;
    drawMatches(img_1, keypoints_1, img_2, keypoints_2, mathces, img_mathes);
    // -- show
    imshow("Mathces", img_mathes);

    waitKey(0);
    return 0;
}
```



四、 结论

通过此实验， 图像 SIFT、 ORB、提取了 HOG 的特征， 并更深入地了解了其中主要目标的排序工作

五、 参考文献

[1] Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary R. Bradski: ORB: An efficient alternative to SIFT or SURF. ICCV 2011: 2564-2571.

[2] 看 ORB 特征，一些理解和解释

[3] OpenCV Tutorials