

特征匹配

姓名：胡天扬

学号：3190105708

专业：自动化（控制）

课程：数字图像处理与机器视觉

指导教师：姜伟

一、题目要求

编程实现 SIFT、SURF、ORB 等图像特征提取算法，并比较各算法性能。

二、原图



输入是两张不同角度的图像，图像尺寸为 480 × 640。

三、ORB

3.1 原理

具体原理就略去了，毕竟是上机作业，其实原理在代码里都已经体现了，已 *ORB* 为例进行说明。

3.2 代码

```
1  /*
2   * ORB
3   */
4  vector<KeyPoint> key_points1, key_points2;
5  Mat descriptions1, descriptions2;
6  Ptr<ORB> orb = ORB::create(500, 1.2f, 8, 31, 0, 2, ORB::HARRIS_SCORE, 31, 20);
7
8  // (1) 检测 Oriented FAST 角点位置
9  orb->detect(img1, key_points1);
10 orb->detect(img2, key_points2);
11
12 // (2) 由角点位置计算 BRIEF 描述子
13 orb->compute(img1, key_points1, descriptions1);
14 orb->compute(img2, key_points2, descriptions2);
15
16 Mat orb_img;
17 drawKeypoints(img1, key_points1, orb_img, Scalar::all(-1),
18 DrawMatchesFlags::DEFAULT);
18 showImage("ORB特征点", orb_img, image_size, 0,
19 ".../images/ORB_features.png");
20
20 // (3) 用 Hamming 距离匹配两帧中的 BRIEF
21 vector<DMatch> matches_orb;
22 BFMatcher matcher_orb(NORM_HAMMING);
23 matcher_orb.match(descriptions1, descriptions2, matches_orb);
24
25 // (4) 筛选配对点
26 // 找出最大和最小距离
27 double min_dist = 10000, max_dist = 0;
28 for (int i = 0; i < descriptions1.rows; i++)
29 {
30     double dist = matches_orb[i].distance;
31     if (dist < min_dist)
32         min_dist = dist;
33     if (dist > max_dist)
34         max_dist = dist;
35 }
36
37 // 描述子距离大于两倍最小值或经验值时，认为匹配有误
38 vector<DMatch> good_matches_orb;
39 for (int i = 0; i < descriptions1.rows; i++)
40     if (matches_orb[i].distance <= max(2*min_dist, 30.0))
41         good_matches_orb.push_back(matches_orb[i]);
42
43 // (5) 绘制匹配结果
44 Mat img_match, img_good_match;
45 drawMatches(img1, key_points1, img2, key_points2, matches_orb, img_match);
46 drawMatches(img1, key_points1, img2, key_points2, good_matches_orb,
47 img_good_match);
47 showImage("ORB所有匹配点", img_match, image_size, 0,
48 ".../images/ORB_all_matches.png");
```

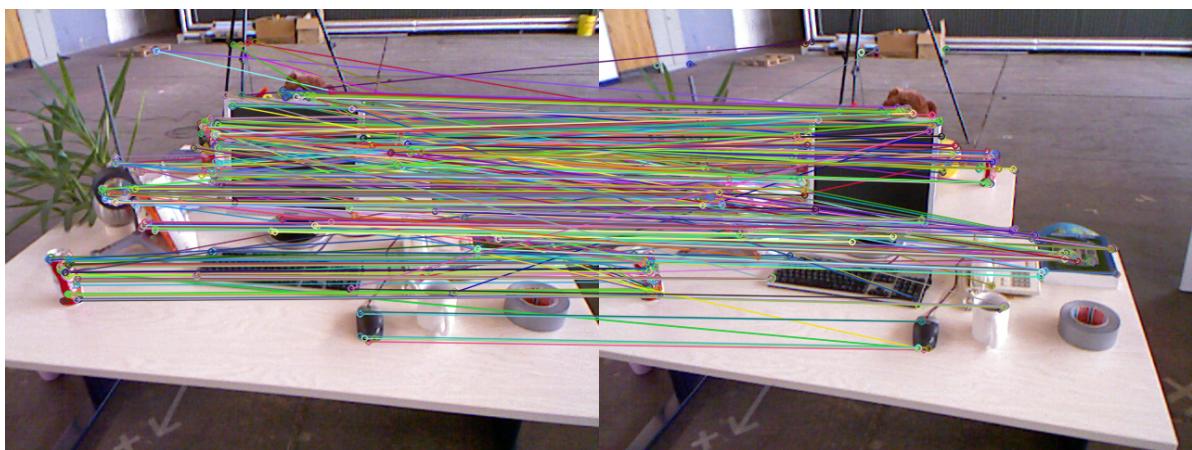
```
48 showImage("ORB优化后的匹配点", img_good_match, image_size, 0,  
49     ".../images/ORB_good_matches.png");  
destroyAllWindows();
```

在获取两幅图像的描述子后，根据最大汉明距离进行筛选，得到优化后的匹配。

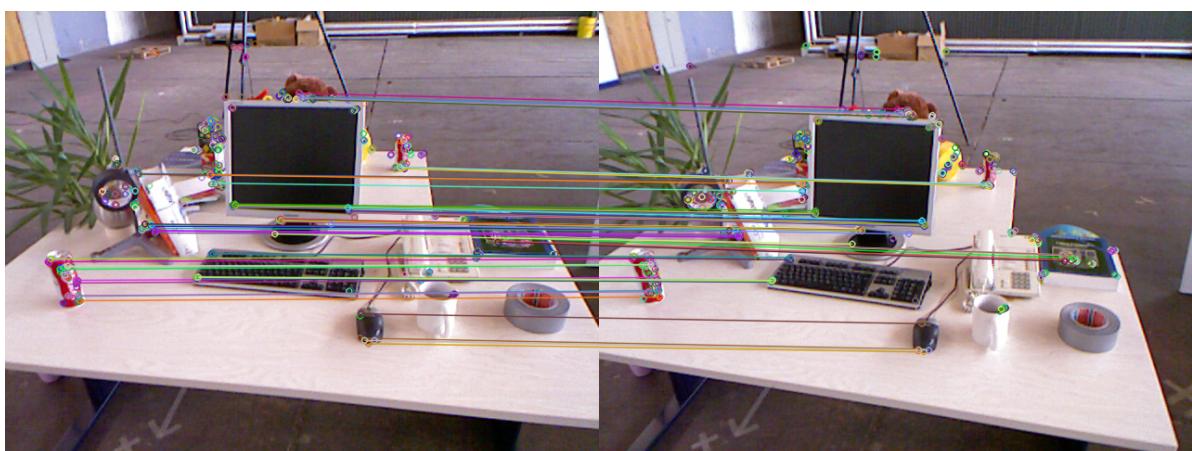
3.3 运行结果



ORB特征点



ORB全部特征匹配



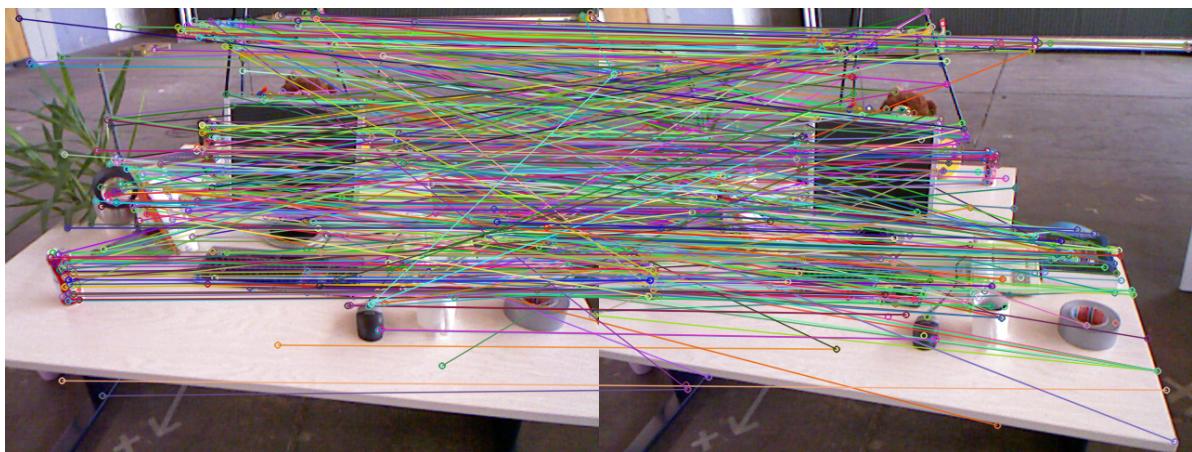
ORB优化后的特征匹配

四、SIFT

4.1 运行结果



SIFT特征点



SIFT得全部特征匹配



SIFT优化后的特征匹配

五、SURF

5.1 运行结果



SURF特征点



SURF得全部特征匹配



SURF优化后的特征匹配

六、算法比较

可以说 SIFT 是尺度不变特征检测的开山之作，但是 SIFT 的缺陷也很明显：实时性不高、特征点少、对边缘光滑的图像效果差。而ORB、SURF 都是在 SIFT 上做出的改进，SURF 的运算简单，效率更高，检测的特征点也更多；ORB则效果更为显著，效率也更高。

特征点检测数量：SURF > ORB > SIFT

检测特征点时间：ORB < SURF < SIFT

七、总结

opencv3 之后 SURF 被移出去了，要多装个 `opencv_contrib`，代码上由于这部分之前接触过，所以没有什么难度。