

计算机视觉基础实验报告（三）

1811361 物联网 郭宇

一、实验要求

对于两张图片，找出其中的特征点，并进行特征点匹配，在其匹配的特征点中连线证明其匹配。

二、实验环境

opencv4.5.1

opencv contrib 4.5.1

visual studio 2019 x64

三、实验步骤

下载并重新编译了带有扩展库的opencv4.5.1，使用扩展库里已经写好的SURF特征点检测进行实验。

对于opencv的扩展的编译，可以参见博客：《[在Windows下编译扩展OpenCV 3.1.0 + opencv contrib](#)》

编译成功后，代码如下：

```
1  //-- Step 1: Detect the keypoints using SURF Detector, compute the
   descriptors
2      int minHessian = 400;
3      Ptr<SURF> detector = SURF::create(minHessian);
4      std::vector<KeyPoint> keypoints1, keypoints2;
5      Mat descriptors1, descriptors2;
6      detector->detectAndCompute(img1, noArray(), keypoints1, descriptors1);
7      detector->detectAndCompute(img2, noArray(), keypoints2, descriptors2);
```

创建SURF检测器和特征点描述容器。

```
1  //-- Step 2: Matching descriptor vectors with a FLANN based matcher
2      // Since SURF is a floating-point descriptor NORM_L2 is used
3      Ptr<DescriptorMatcher> matcher =
   DescriptorMatcher::create(DescriptorMatcher::FLANNBASED);
4      std::vector< std::vector<DMatch> > knn_matches;
5      matcher->knnMatch(descriptors1, descriptors2, knn_matches, 2);
```

使用FLANN算法进行特征点匹配。

```

1  //-- Filter matches using the Lowe's ratio test
2      const float ratio_thresh = 0.7f;
3      std::vector<DMatch> good_matches;
4      for (size_t i = 0; i < knn_matches.size(); i++)
5      {
6          if (knn_matches[i][0].distance < ratio_thresh * knn_matches[i]
7              [1].distance)
8              {
9                  good_matches.push_back(knn_matches[i][0]);
10             }
11     }

```

去除反应不大的特征点。

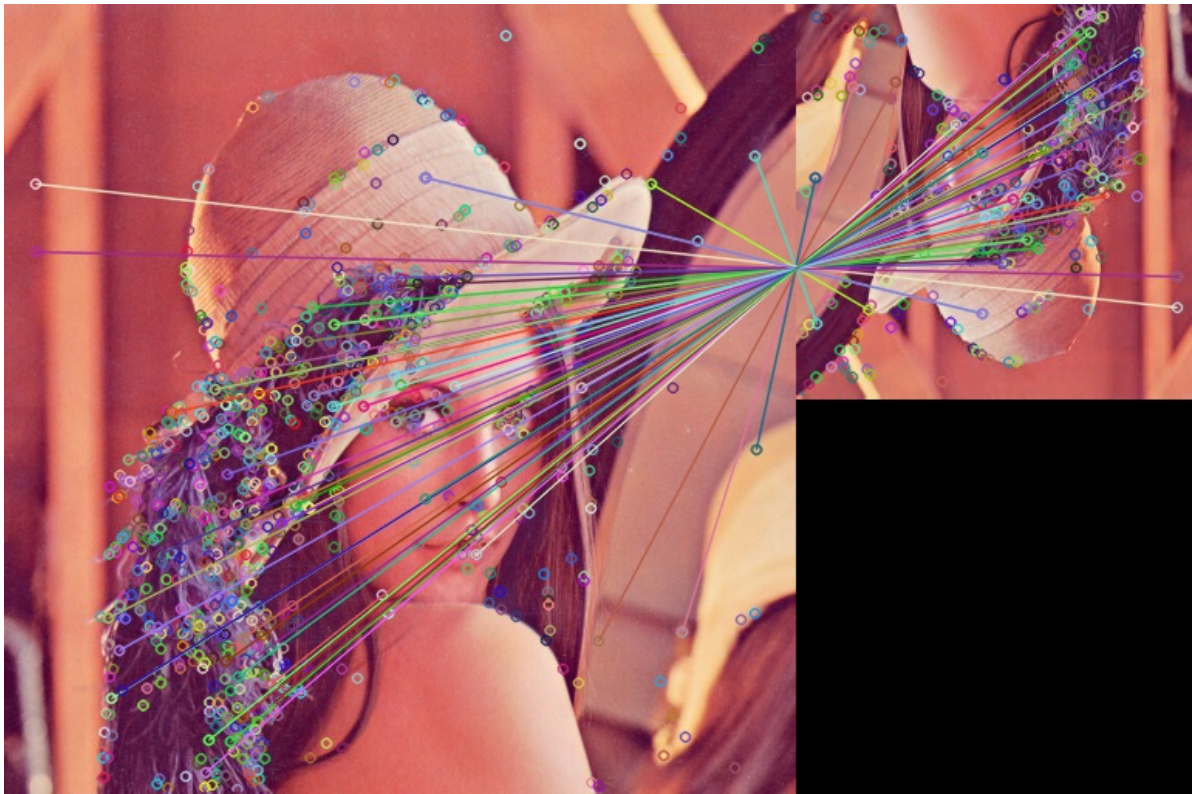
```

1  //-- Draw matches
2      Mat img_matches;
3      drawMatches(img1, keypoints1, img2, keypoints2, good_matches,
4                  img_matches, Scalar::all(-1),
5                  Scalar::all(-1), std::vector<char>(),
6                  DrawMatchesFlags::NOT_DRAW_SINGLE_POINTS);
7
8      //-- Show detected matches
9      imshow("Good Matches", img_matches);

```

画出关键点之间的连线并显示图片。

四、实验结果



五、实验感想

调用函数很快就解决了问题，同时测试了SIFT和SURF两种检测方法，特征点检测并无明显差异。

