

计算机视觉 课程实验报告

学号：	姓名：	班级：
实验题目：图像匹配 2		
<p>实验内容：</p> <p>实验 9.1 特征检测与匹配</p> <ul style="list-style-type: none">了解 OpenCV 中实现的 SIFT, SURF, ORB 等特征检测器的用法，并进行实验。将检测到的特征点用不同大小的圆表示，比较不同方法的效率、效果等。了解 OpenCV 的特征匹配方法，并进行实验。		
<p>实验过程中遇到和解决的问题：</p> <p>（记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）</p> <p>一、opencv 特征检测和匹配的通用步骤及 Code</p> <pre>//步骤一：读取图片并将图片灰度化 //code: Mat src1, src2; src1 = imread("图片路径"); src2 = imread("图片路径"); Mat graySrc1, graySrc2; cvtColor(src1, graySrc1, CV_BGR2GRAY); cvtColor(src2, graySrc2, CV_BGR2GRAY); //步骤二：提取特征并描述 //code: vector<KeyPoint> keys1; vector<KeyPoint> keys2; Ptr<xfeatures2d::SURF> detector = xfeatures2d::SURF::create(1500); Mat descriptorMat1, descriptorMat2; detector->detectAndCompute(src1, Mat(), keys1, descriptorMat1); detector->detectAndCompute(src2, Mat(), keys2, descriptorMat2); //步骤三：特征点匹配 //code: cv::BFMatcher matcher; std::vector<DMatch> matches; matcher.match(descriptorMat1, descriptorMat2, matches); //步骤四：获取优秀匹配点</pre>		

```

//code:
double max_dist = 0; double min_dist = 100;
for (int i=0; i<descriptorMat1.rows; i++)
{
    double dist = matches[i].distance;
    if (dist < min_dist) min_dist = dist;
    if (dist > max_dist) max_dist = dist;
}
cout<<"-- Max dist :"<< max_dist<<endl;
cout<<"-- Min dist :"<< min_dist<<endl;

vector< DMatch > good_matches;
for (int i=0; i<descriptorMat1.rows; i++)
{
    if (matches[i].distance < 0.2*max_dist)
    {
        good_matches.push_back(matches[i]);
    }
}

//步骤五：绘制特征匹配图
//code:
Mat img_matches;
drawMatches(src1, keys1, src2, keys2, good_matches, img_matches,
            Scalar::all(-1), Scalar::all(-1), vector<char>(),
            DrawMatchesFlags::NOT_DRAW_SINGLE_POINTS);

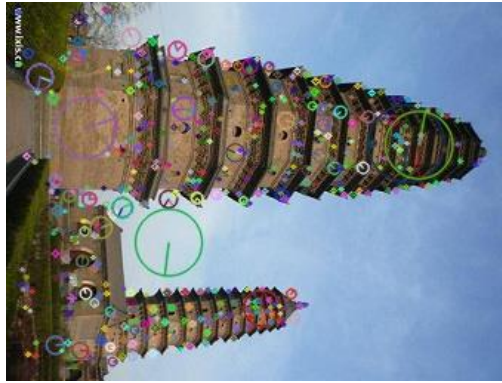
```

注：需要添加#include<opencv2/xfeatures2d.hpp>，
#include<opencv2/features2d.hpp>，其中 SIFT 和 SURF 在 xfeatures2d 中，ORB 在 feature2d 中。

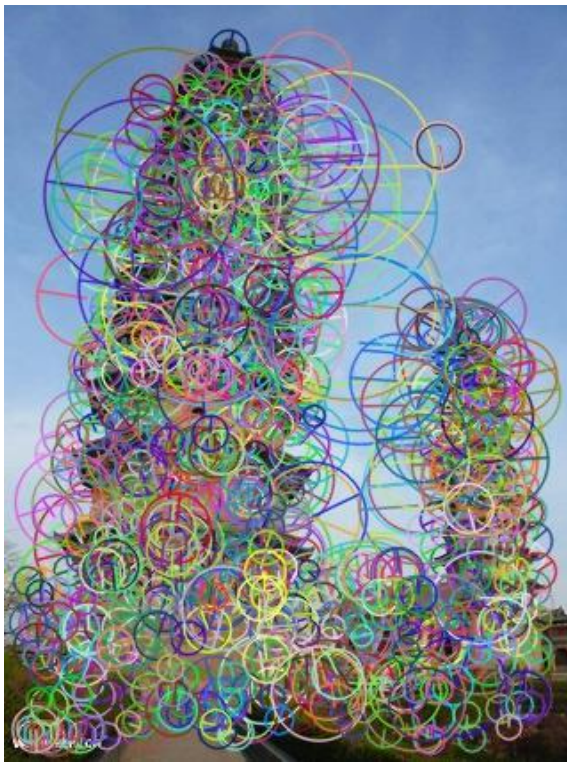
二、测试结果及对比展示

1、原图 1 (340*256) 的特征检测结果：（按 ORB→SURF→SIFT 顺序）



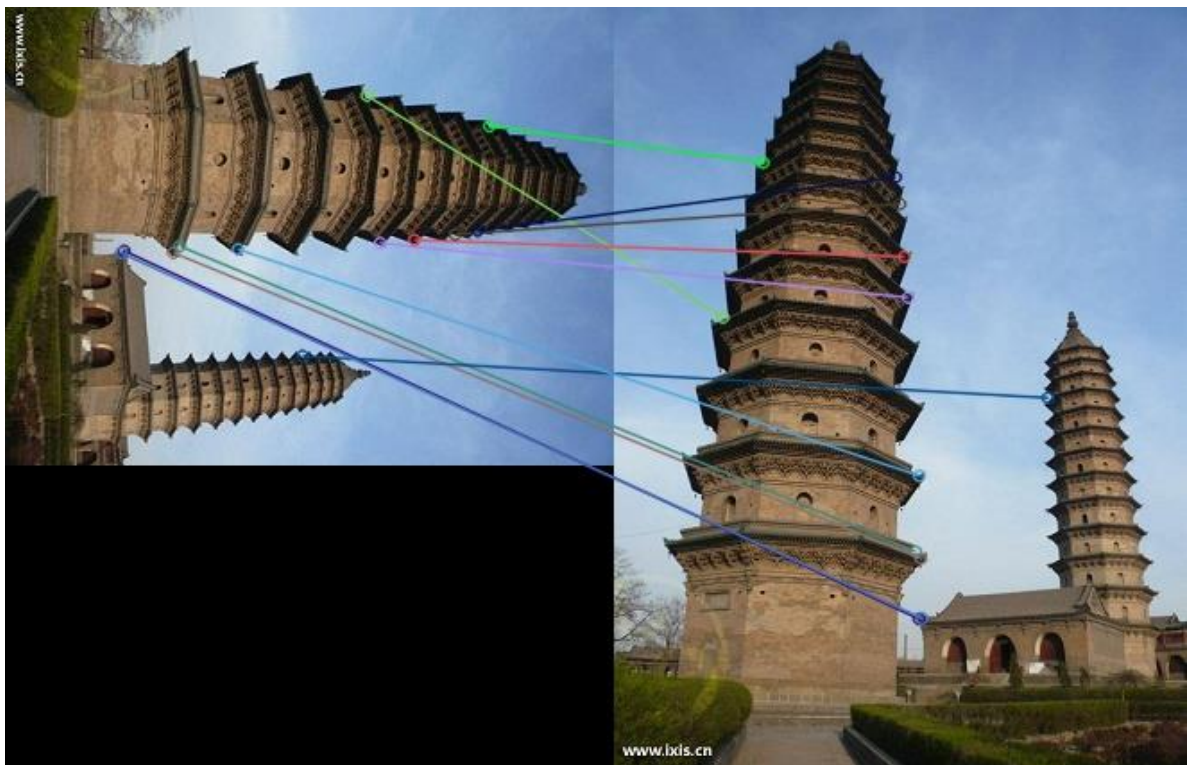


2、原图 2 (320*426) 的特征检测结果：（按 ORB-→SURF-→SIFT 顺序）

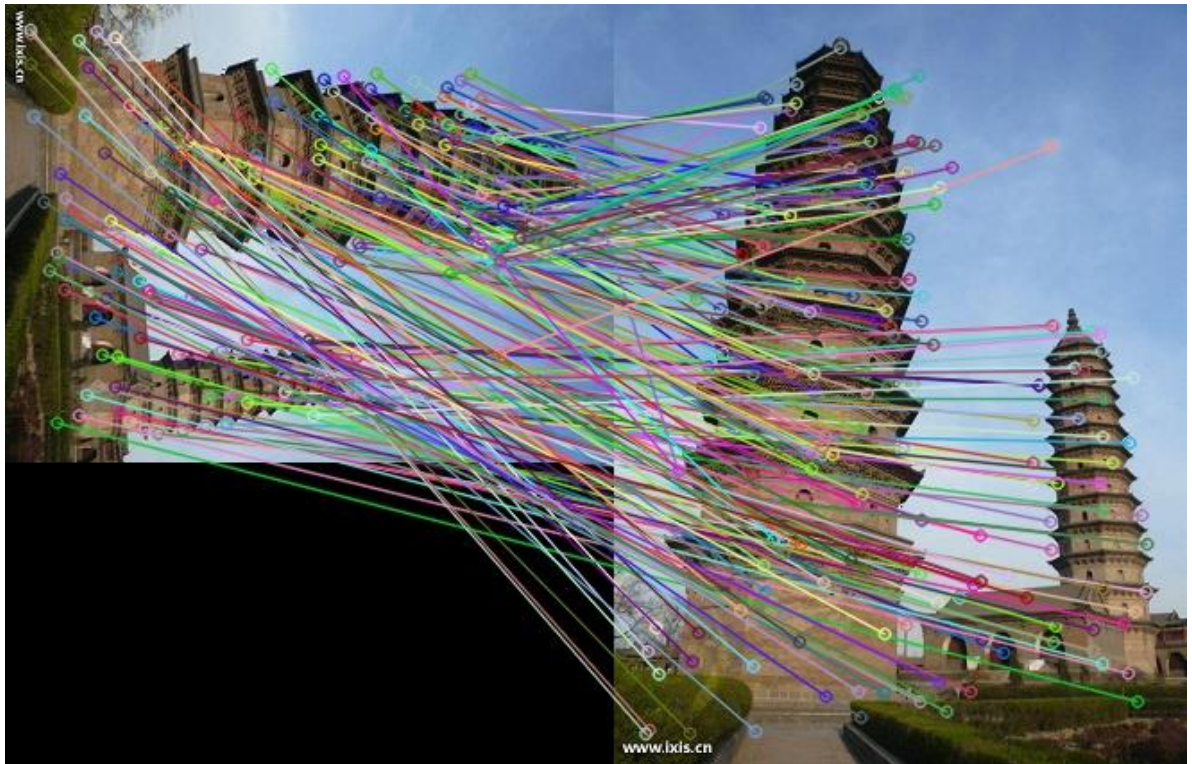


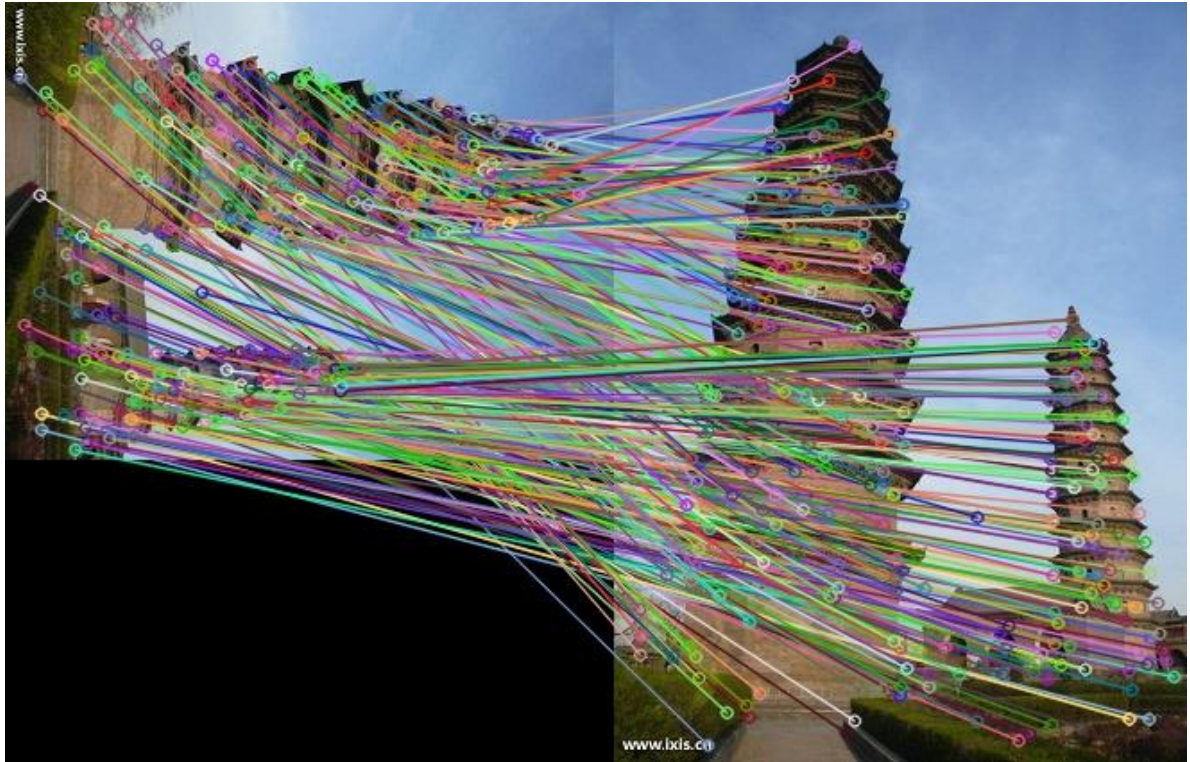


3、源图 1 和 2 的特征匹配结果：（按 ORB→SURF→SIFT 顺序）



对 ORB 的结果不进行 matches 的筛选结果如下：





4、运行时间如下：

```
[ INFO:0] Initialize OpenCL runtime...
ORB-- Max dist :456.863
ORB-- Min dist :57.0351
ORB运行时间为672ms!

SURF-- Max dist :0.559342
SURF-- Min dist :0.0139962
SURF运行时间为1313ms!

SIFT-- Max dist :391.801
SIFT-- Min dist :10.1489
SURF运行时间为719ms!
```

```
[ INFO:0] Initialize OpenCL runtime...
ORB运行时间为1125ms!

SURF-- Max dist :0.755144
SURF-- Min dist :0.0117634
SURF运行时间为3484ms!

SIFT-- Max dist :454.62
SIFT-- Min dist :18.3576
SURF运行时间为1782ms!
```

6、总结以上，从运行效率和效果上进行排序，可以发现 $SIFT > SURF > ORB$, 其中 SIFT 的方法最好。