Assignment 8

Part 1 Code

参见 SIFT_SURF.cpp。

Part 2 Explanation

（1）

```
SiftFeatureDetector sift_detector(minHessian);
sift_detector.detect(img_obj, keypoints_obj);
sift_detector.detect(img_env, keypoints_env);

SiftDescriptorExtractor sift_extractor;
sift_extractor.compute(img_obj, keypoints_obj, descriptors_obj);
sift_extractor.compute(img_env, keypoints_env, descriptors_env);
```

Sift 算法获得特征点和特征向量。

（2）

```
SurfFeatureDetector surf_detector(minHessian);
surf_detector.detect(img_obj, keypoints_obj);
surf_detector.detect(img_env, keypoints_env);

SurfDescriptorExtractor surf_extractor;
surf_extractor.compute(img_obj, keypoints_obj, descriptors_obj);
surf_extractor.compute(img_env, keypoints_env, descriptors_env);
```

Surf 算法获得特征点和特征向量。
和 Sift 的调用方式类似。

（3）

```
BruteForceMatcher< L2<float> > matcher;
std::vector< DMatch > matches;
matcher.match(descriptors_obj, descriptors_env, matches);

Mat img_matches;
drawMatches(img_obj, keypoints_obj, img_env, keypoints_env, matches, img_matches);
```

使用 BruteForce 匹配两幅图，得到 img_matches。

（4）

```
std::vector<Point2f> obj;
std::vector<Point2f> scene;

for (int i = 0; i < matches.size(); i++)
{
    //-- Get the keypoints from the good matches
    obj.push_back(keypoints_obj[matches[i].queryIdx].pt);
    scene.push_back(keypoints_env[matches[i].trainIdx].pt);
}
Mat H = findHomography(obj, scene, CV_RANSAC);

std::vector<Point2f> obj_corners(4);
obj_corners[0] = cvPoint(0, 0);
obj_corners[1] = cvPoint(img_obj.cols, 0);
obj_corners[2] = cvPoint(img_obj.cols, img_obj.rows);
obj_corners[3] = cvPoint(0, img_obj.rows);
std::vector<Point2f> scene_corners(4);
perspectiveTransform(obj_corners, scene_corners, H);

line(img_matches, scene_corners[0] + Point2f(img_obj.cols, 0), scene_corners[1] + Point2f(img_obj.cols, 0), Scalar(0, 255, 0), 4)
line(img_matches, scene_corners[1] + Point2f(img_obj.cols, 0), scene_corners[2] + Point2f(img_obj.cols, 0), Scalar(0, 255, 0), 4)
line(img_matches, scene_corners[2] + Point2f(img_obj.cols, 0), scene_corners[3] + Point2f(img_obj.cols, 0), Scalar(0, 255, 0), 4)
line(img_matches, scene_corners[3] + Point2f(img_obj.cols, 0), scene_corners[0] + Point2f(img_obj.cols, 0), Scalar(0, 255, 0), 4)
```

计算 object 的 corner，在 img_matches 中画出方框，在 environment 中标记出 object。

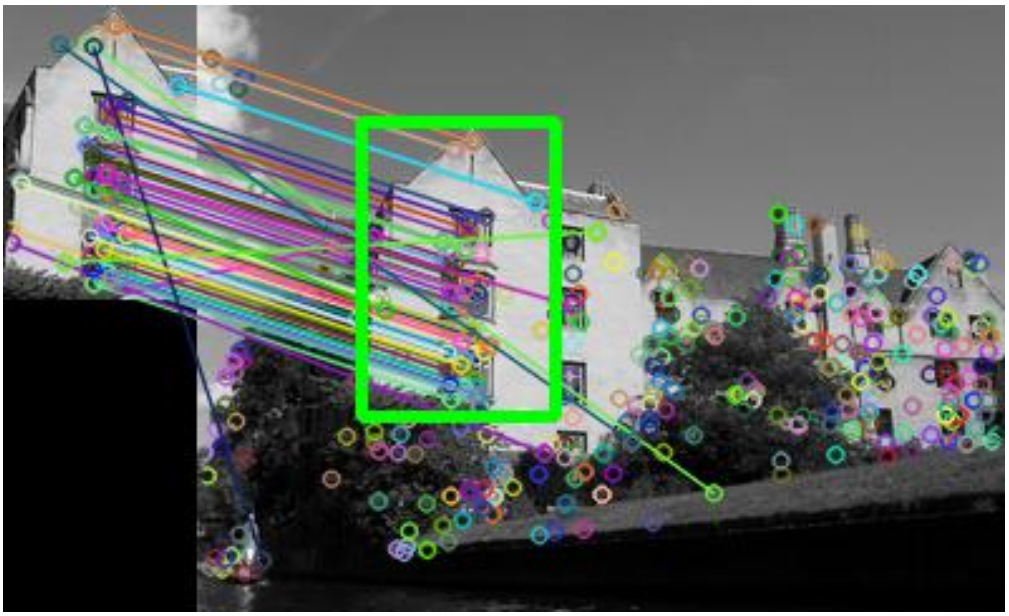
Part 3 Output

测试的图片涉及了不同的尺度、方向和光照。

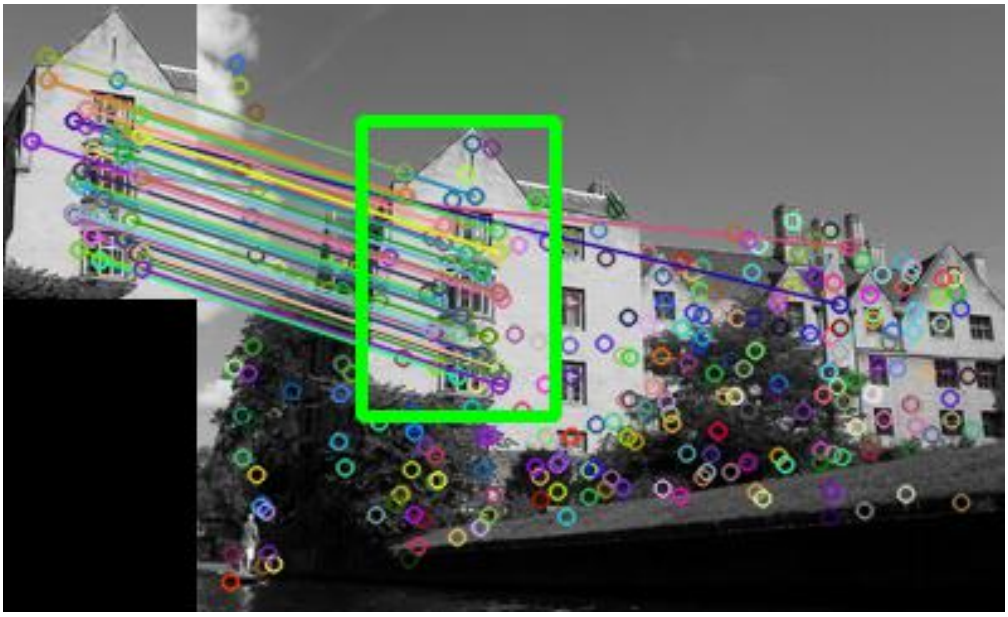
（1）

environment

object



Sift 匹配结果

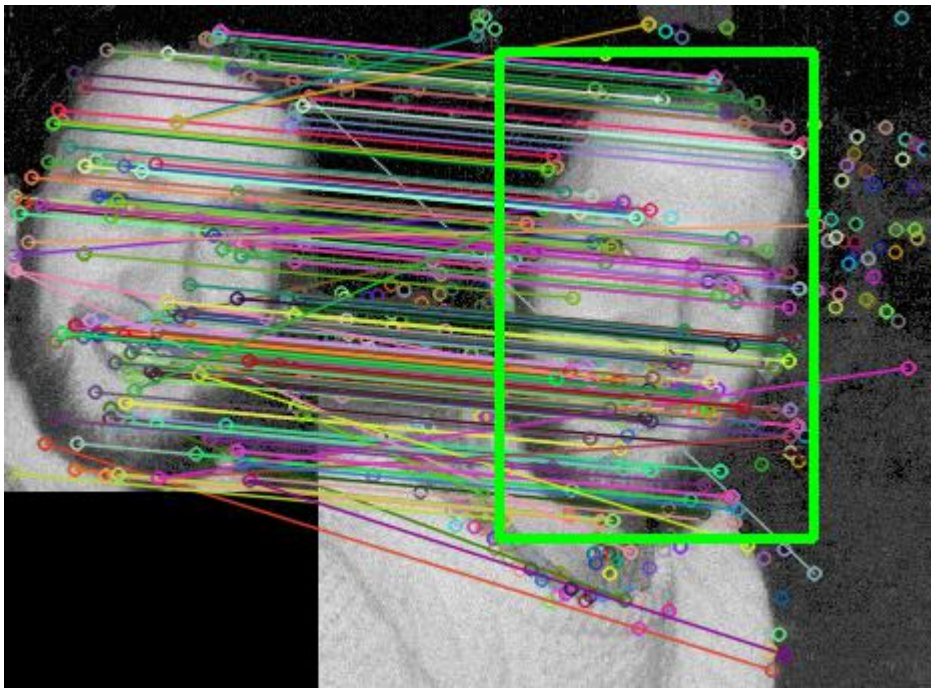


Surf 匹配结果

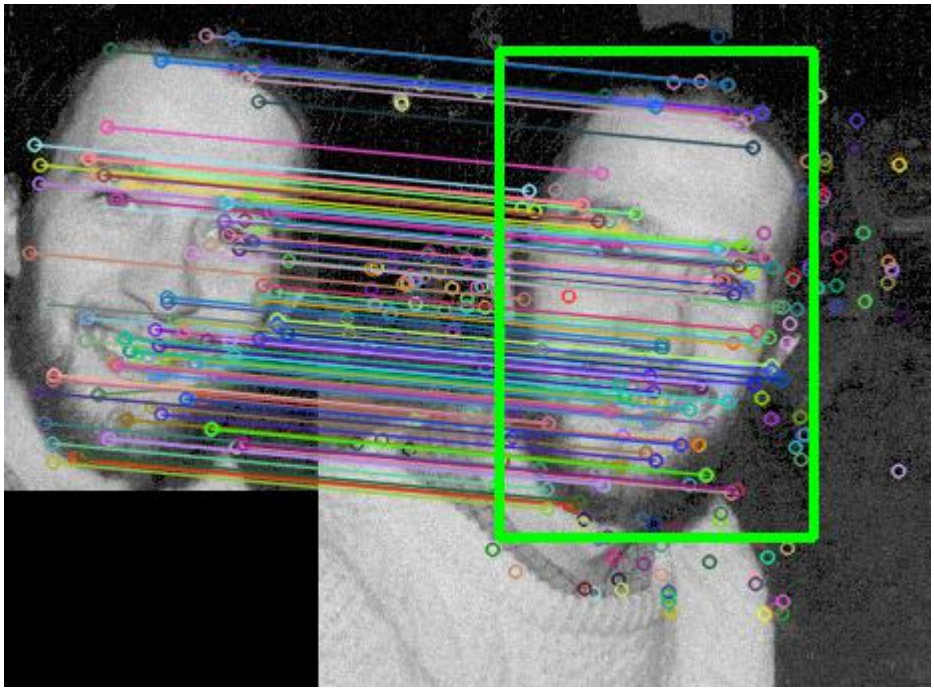
(2)

environment

object



Sift 匹配结果

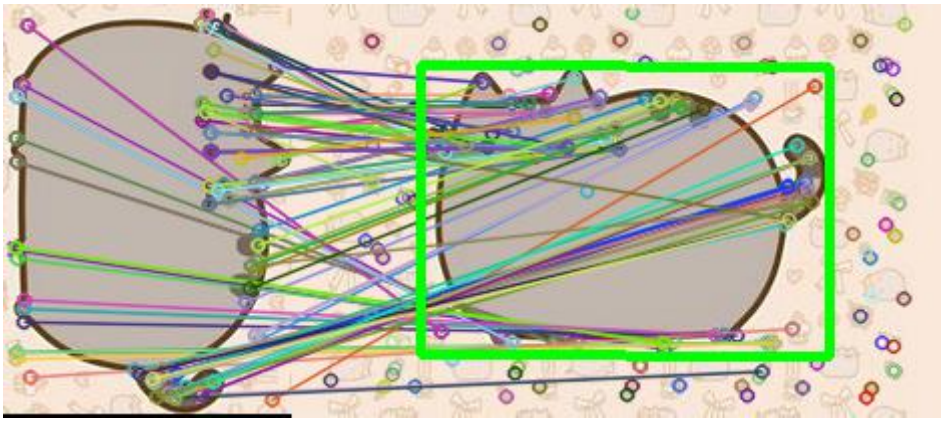


Surf 匹配结果

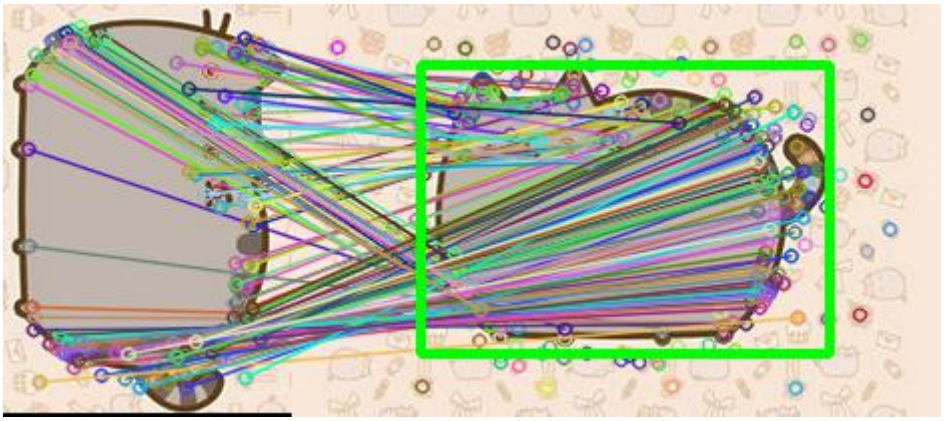
（3）

environment

object



Sift 匹配结果

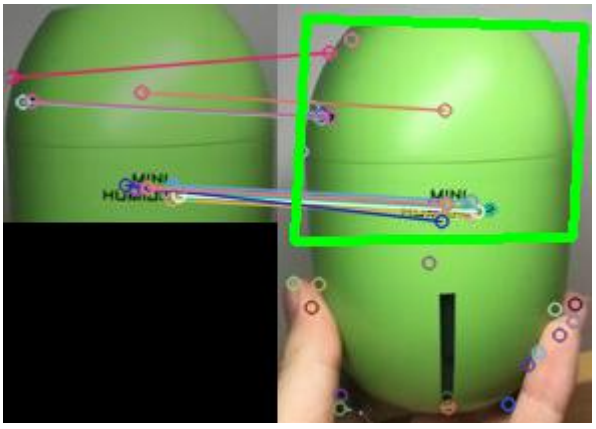


Surf 匹配结果

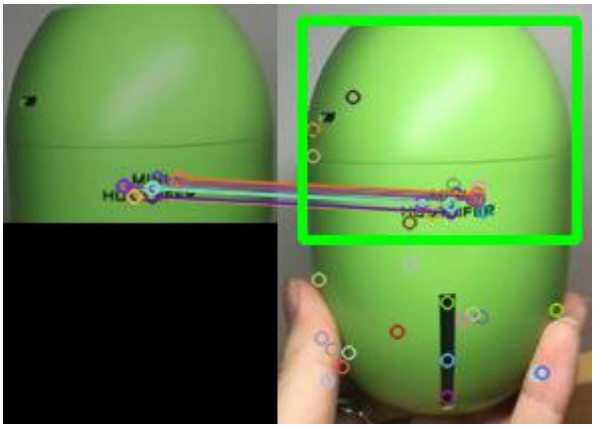
（4）

environment

object



Sift 匹配结果



Surf 匹配结果