一、算法和软件说明

**解压SIFT.zip**后主目录下sift.py为主程序文件，sift.exe为可执行文件。

**直接双击sift.exe即可运行**。

1. 算法说明

算法实现基于Numpy从零开始实现SIFT，参照了两篇论文Distinctive Image Features from Scale-Invariant Keypoints[1]和The Anatomy of the SIFT Method[2]，其中后者对SIFT进行了详细的分析和参数配置分析，故算法实现参数配置参照[2]。

遵循SIFT的实现过程，将算法分为五个模块：八度图(octave)、关键点、参考方向、描述子的构建，特征点的匹配。具体而言:

*octaves.py 创建逐渐增强模糊的高斯和DoG的octaves；*

*keypoints.py 根据DoG octave的极值生成候选关键点；*

*orientations.py 根据局部梯度给关键点分配参考方向；*

*descriptors.py 根据局部最大邻域内梯度分配描述子；*

*match.py 匹配两图像中SIFT特征并可视化匹配结果。*

上述五个模块都在路径SIFT/src内。基于五个模块，在主程序文件sift.py中构建SIFT实例，然后载入两张灰度图像，匹配并可视化匹配结果。

*此外，SIFT/src/miscs.py整合了各个模块中用到的辅助函数。*

**可直接在命令行中运行算法主程序：**

*cd/d ‘path\_unzip’/SIFT # 切换到解压SIFT的路径*

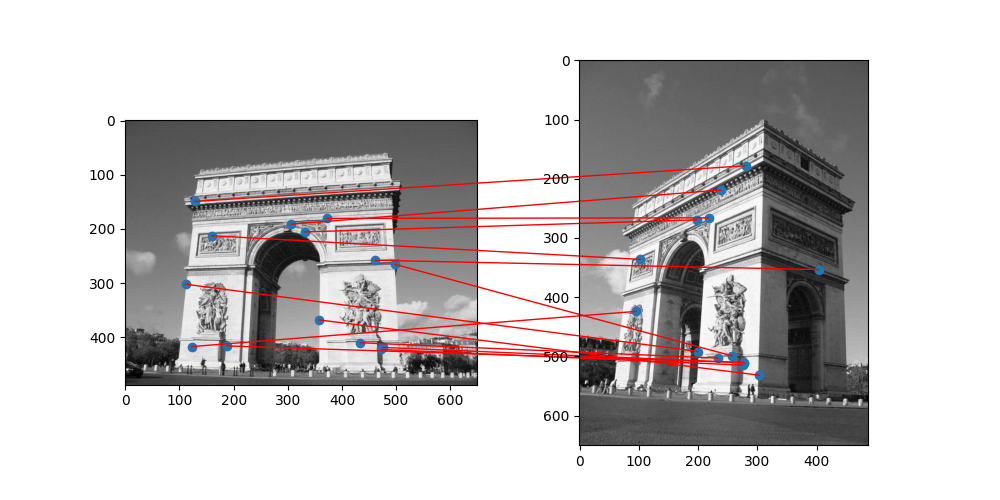
*python sift.py*

1. 软件说明

可执行文件sift.exe由pyinstaller打包，**可直接在windows系统上双击运行sift.exe**。运行后会自动弹出命令行运行界面，可查看运行消息，SIFT检测并匹配结束后会弹出匹配结果图，关闭匹配结果图后命令行窗口10秒后自动关闭。

二、运行结果

在两张尺寸分别为(488, 650), (650, 488)的法国凯旋门灰度图像上，SIFT检测和匹配用时2分32秒，分别检测到1777和1551个特征点，匹配到14个特征点，匹配结果如下图。



作为对比试验，调用开源库cv2中的SIFT检测接口，对同样两张图像进行检测和匹配。结果为：用时2秒，分别检测到2057和1693个特征点，匹配到83个特征点，对匹配程度最高的20个点可视化，如下图。



对比两种方法的匹配结果可以发现，在测试的两张图像上，官方SIFT算法匹配精度并不高，明显左图左上角的很多特征点错误地匹配到了右图右下角的特征点，但是明显地官方方法在效率上高很多，基本上快了两个数量级。

附录

1. 算法依赖说明

本算法实现仅使用3个第三方库：Numpy，pillow和matplotlib。其中Numpy用于实现中数组的各种操作，pillow用于图像的读入和尺寸调整，matplotlib用于匹配结果可视化。具体版本要求见SIFT/src/requirements.txt。可执行文件已集成安装各依赖库。

1. 算法补充说明

参照README.md。完整算法实现和软件已上传至GitHub:

<https://github.com/brian-zZZ/SIFT-Numpy/tree/master>。

1. 对比实验说明

对比实验中cv2的SIFT实现代码文件：SIFT/src/cv2\_sift.py。直接运行即可，匹配结果图会弹出并保存至本地SIFT/results/cv2\_matching.png。注意，依赖于cv2库，需安装，限于内存要求，可执行文件中并未安装。

引用

[1] LoweDavid, G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision.

[2] Otero, I. R. , & Delbracio, M. . (2014). The Anatomy of the SIFT Method. Image Processing On Line.