

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`

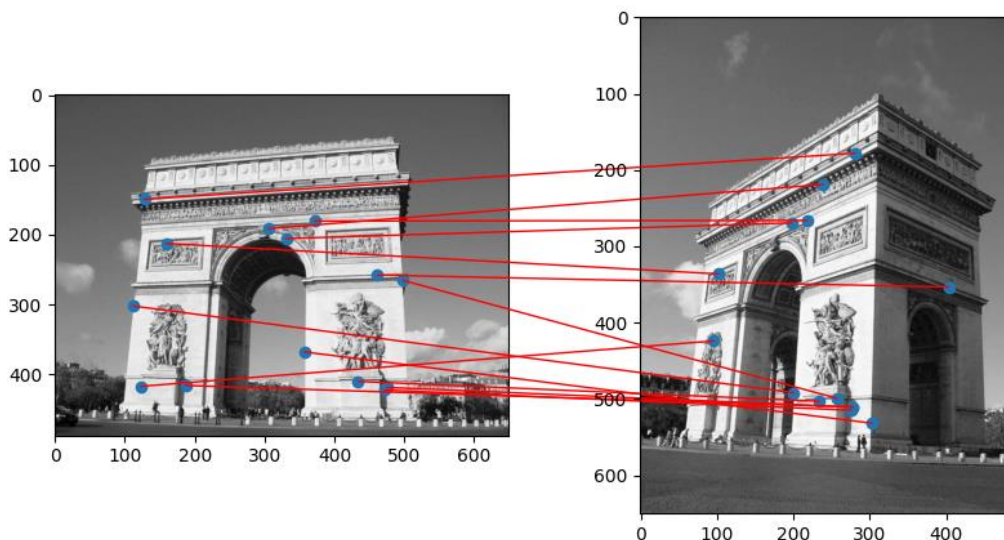
`python sift.py`

2. 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。

2. 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>。

3. 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*.