**1.实验原理**

本断裂线连接算法使用PyCharm开发平台，基于Python编程语言完成算法开发，具体过程涉及到Numpy、Matplotlib第三方库。具体代码见同文件夹下的BrokenLine.py。

**完成实验的原理为：**

**a.**将原始数据利用Numpy分割成两个数组，其中前两列为首端点数组head\_lines和第三列和第四列为尾端点tail\_lines，使用两层for循环，计算每一个首端点到所有尾端点的距离和角度；

**b.**但计算距离和角度时，还需要考虑不应该连接同一条直线上的两个端点，故设置布尔值数组to\_delete，若计算完成，将值设置为False，则以后都不再计算该端点；

**c.**设置距离阈值distance\_threshold和角度阈值angle\_threshold，若计算的两个点满足距离和角度阈值，则视为应当连接点，置入连接点数组connected\_lines\_start和connected\_lines\_end；

**d.**利用Image库，读取tif影像作为底图并将其转换成Numpy数组，使用Matplotlib创建子图，分别显示tif影像，原始的断裂线(红色)，计算后得到的连接线(蓝色)。

**2.实验结果**

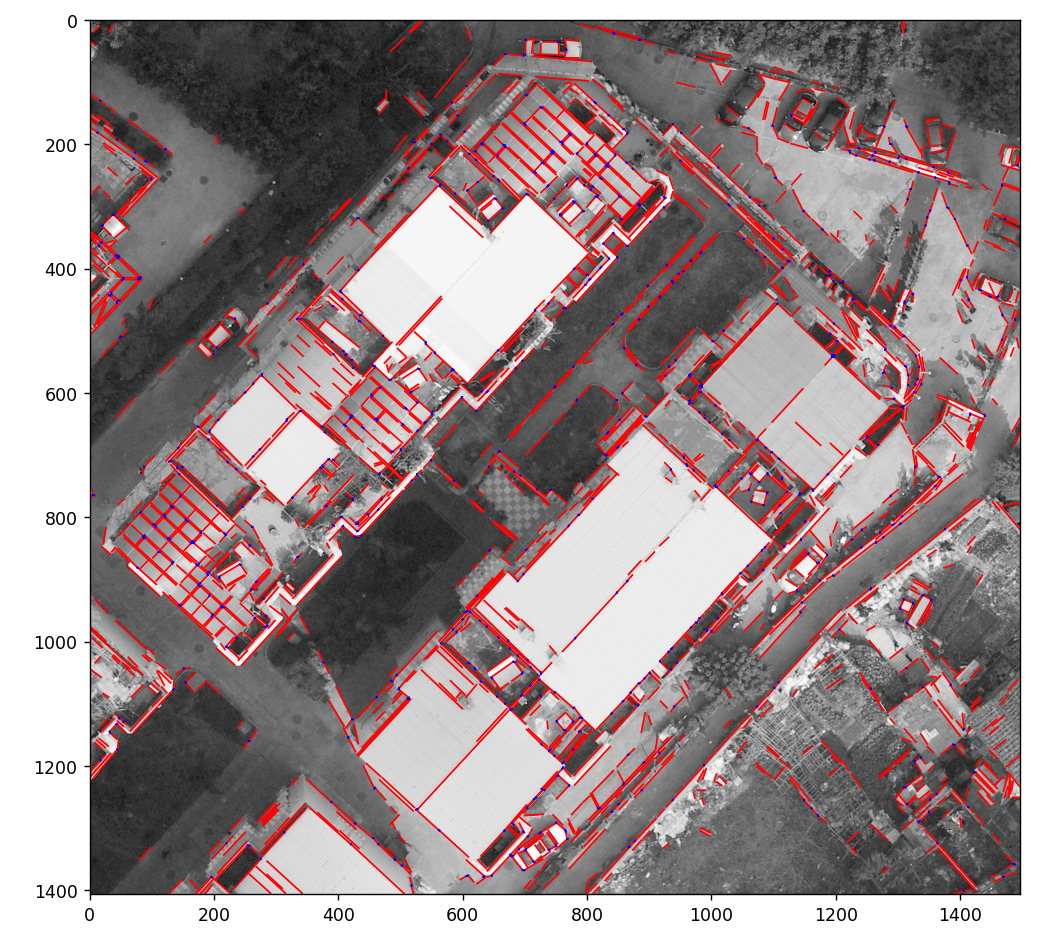


图1 整体实验结果展示（蓝色线为后续计算得到的线）

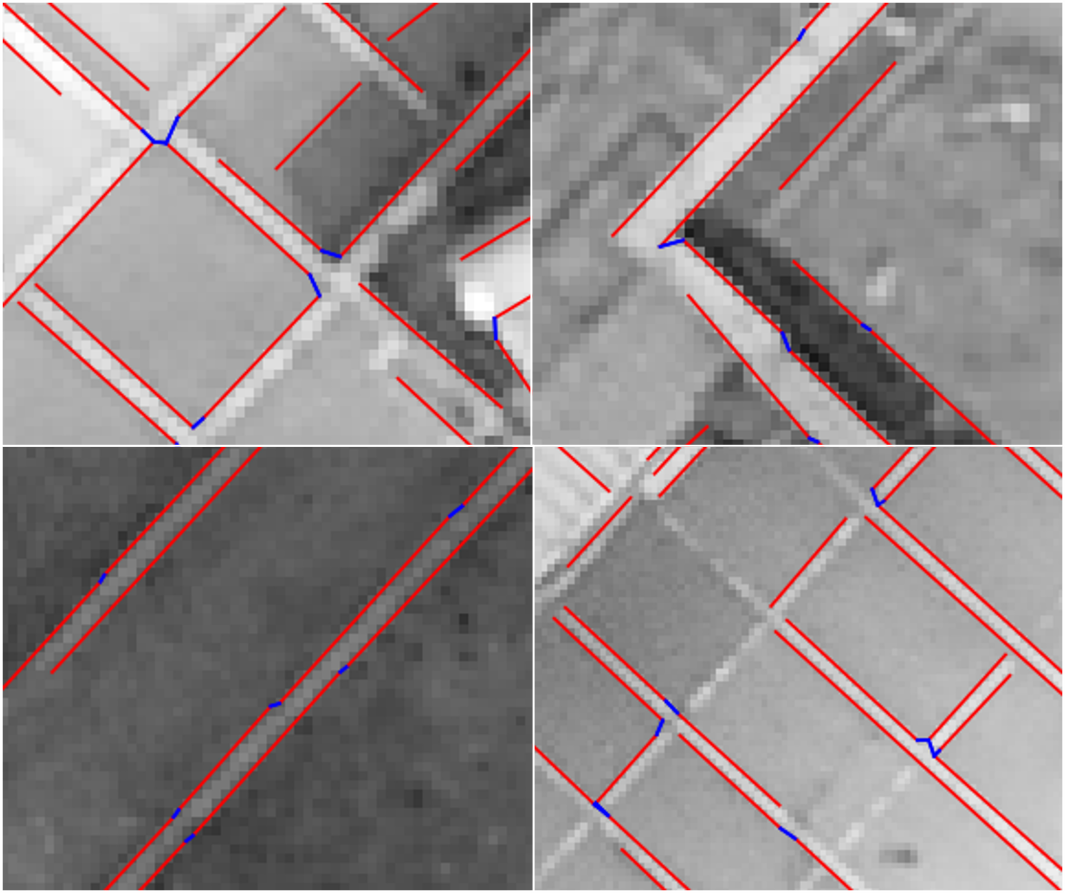


图2 局部实验结果展示（蓝色线为后续计算得到的线）

**3.实验分析**

**该算法还存在一些问题：**

**a.**对于垂直的直线，需要将一端端点沿直线延长，若相交，则应当在交点处连接。本算法还未考虑到该种情况，后续应当做调整。

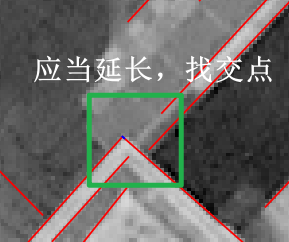


图3 漏洞：需要延长直线找交点

**b.**一些距离其他原始断裂线很远的线条，同时也未在tif影像中能明显看出是明显边界的，应当予以剔除，或者重新对tif影像处理，增加对比度来获取原始线条。



图4 漏洞：周围没有其他直线，可以考虑剔除

**c.**对于道路转弯处，连接后的线条是直直的从绿地上穿过去的，若有必要，应对连接后的线条进行直线拟合，改善实验结果。

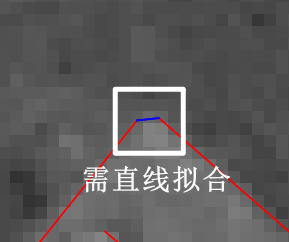


图5 漏洞：没有弧度，是直直地连过去

**d.**后续可以将原始断裂线矢量数据转换成栅格数据，通过膨胀进行断裂线连接。