

Proj1 问题陈述

花振东 jackeyhua@sjtu.edu.cn

2023 年 12 月 2 日

本项目的名字是多边形面积计算的应用，核心目标是读入用户输入的一张图片，然后读入图片的比例尺和图片上用来表示目标多边形区域的点，最后高亮并计算所围成多边形的面积。在本项目中，图形界面的排版显示和图片的读入等都依靠 **FLTK** 库实现。下面分别从图片读入、比例尺读入、数据点的读入和表示、多边形面积计算、窗口显示这五个方面进行详细的问题陈述与分析。

针对图形读入，我们的问题和目标在于使用一种数据结构去读入并且操作用户输入的图片。我们可以使用 **FLTK** 的图片读入功能。根据题目要求，用户输入的图片只可能是 **jpg** 或者 **png** 格式的图片。我们可以利用一个 **string** 变量来读取用户的输入，并对图片格式进行解析。用 **F1_JPEG_Image** 来读入 **jpg** 格式的图片，用 **F1_PNG_Image** 来读入 **png** 格式的图片。对于不满足格式要求的输入，会输出错误信息并退出程序。

针对比例尺读入，我们的问题在于如何让用户输入比例尺，我想到有三种可以实现的方法。第一种是命令行输入的方法，输入一个 **double** 类型的数字，用来表示每 100m 或者 200m 在窗口中是多少像素。第二种方式更加直观，因为输入的图片是地图，所以通常自带有比例尺，而且通常一格或者两格表示 100m。因此在图片上用鼠标选取比例尺中实际距离为 100m 的两个端点即可。再由 **FLTK** 的屏幕进行鼠标点击事件的捕捉，从而获取比例尺。第三种方式是在 **GUI** 中输入比例尺的方式，利用 **F1_Input** 读入用户输入的字符串，并且进行解析，从而获得比例尺。

针对数据点的读入和表示，我们的核心问题在于如何读入多边形的顶点，并且用一个数据结构去存储它们。我想到两种办法。第一种方式是命令行输入，在这种方式下，输入的点必须是以顺时针或者逆时针的方式可以构成多边形的。输入的点以 **std::vector<std::pair<int, int>>** 的形式存储。第二种方式更加合理且直观的去复原凸多边形，用户可以直接在地图上用鼠标选取多边形的顶点，而且更加重要的是，这种方式下我们无需按照顺时针或者逆时针的方式输入顶点。程序通过选取最右下角的点，然后用扫描线算法，根据每个顶点与最右下角的点的斜率进行排序，从而还原多边形。这种方式下，输入的点同样以 **std::vector<std::pair<int, int>>** 的形式存储。为了在这种方式下能够实现点集的读入和排序，我设计了 **polygon** 类来存储点集，并且利用扫描线算法实现点集的排序。为了保证算法的正确性，我使用 **Google Test** 框架对该类的功能进行了单元测试。在排完序后，只需使

用 `fl_begin_polygon`，并且依次用 `fl_vertex` 来表示每一个多边形的顶点，最后用 `fl_end_polygon` 来结束多边形的表示即可。这样就能够在 GUI 中高亮所围多边形。

针对多边形面积计算，我们的问题在于如何利用已知的多边形顶点的坐标去计算所围成多边形的面积。为了解决这个问题，可以利用行列式计算面积的方法，以顺时针或者逆时针的顺序分别计算相邻节点与原点所构成三角形的“有向”面积之和，最后再取绝对值即可。实际面积的值就是图中面积的值与比例尺平方的商，然后需要进行一定的单位变换，以及精度保留。这部分是上课所学的计算几何的直接应用。

针对窗口显示，我设计了针对命令行输入和 GUI 输入的两种窗口，分别是 `windemo` 和 `myWindow`，两者都继承了 `Fl_Window`。前者仅提供最基本的图片显示、多边形显示的功能，应对的是纯命令行输入的情况。后者主要应对需要在 GUI 窗口进行输入的情况，设置了两个 `button`。前者用来表示比例尺已读入（无论是通过点击输入还是通过打字输入），后者用来表示数据点已读入。在读入比例尺之后，屏幕上会即时显示比例尺，在读入数据点之后，会在屏幕上高亮显示所围成的多边形，并且在左下角显示图形的计算面积和真实面积。为了保证窗口的美观，窗口的大小取决于输入图片的尺寸，避免了图片过小且窗口过大的情况。