Code 5 实验报告

PB16050087 叶嘉沅

1 实验内容

本次实验利用最大流算法进行图像分割，并进行纹理合成。参考（https://www.cc.gatech.edu/cpl/projects/graphcuttextures/ ）进行实现，实验的主要原理及步骤是：

（1）构建无向有权图，来模拟图像直接的融合度，设计不同拼图在指定像素位置的 cost 函数，记拼图 A1，A2 在 pixel1, pixel2 之间的 cost 为 M(1,2,A1,A2)。本次实验中使用了最简单的一种 cost 函数，即

$$M(s,t,A,B) = |A[s]-B[s]| + |A[t]-B[t]|$$

（2）根据最大流最小割原理，此时可以通过求最大流来求出最小割，且最小割对应的恰为二拼图的最佳拼接线（最佳的含义是 cost 最小）

（3）为了考虑旧有拼接线的影响，需要对切割线做特殊处理，包括增加节点，设计满足三角形法则的权重等等，具体见论文，不赘述

（4）为了得到拼图的位置，可以使用不同的方法不断向输出图上摆放拼图，具体的方法有：随机置放法；最小权重选择法（权重评估了拼图与输出图中某一位置上的已有图像之间的差异，差异越小代表权重越小，即位置越适合摆放该拼图）。

（5）为了增加图像的随机性，可以对图像进行一些变换，包括增大/缩小，旋转等等，由于时间来不及，没有做完此项内容。

2 实验中遇到的问题

此次实验遇到的主要问题如下：

（1）对重复区域的记录较为复杂，由于拼图的位置可变动范围超过图像的边界（可以以部分形式参与拼接），故记录的参数较多，实验中采用 7 元整数组 pos 来记录拼图在输出图中的具体位置，和它参与拼接的部分的具体大小位置参数。

（2）重叠区域图构建较为复杂，且在节点的问题上容易出现纰漏，实验中使用 img_mask 矩阵来存储输出图中每个位置是否已经被着色过，以及它是重叠区域图的第几个节点。使用 TextureSyn.h 文件中的 void TextureSyn::buildgraph()函数来进行处理

（3）对切割的记忆较为复杂，主要体现在处理切割点对于图构建的影响，以及对切割点的维护。使用 TextureSyn.h 文件中的 void TextureSyn::cut_and_record()函数来进行处理。

（4）在实验过程中中，绘图比较慢（大约 3 分钟），使用循环来保证图像一定能够填满输出不现实，由于时间问题，还没有考虑出怎么解决这个问题，故自己的输出可能有部分残缺。

3 使用说明

运行 main.cpp 即可，等待 3 分钟会出现原图和合成的纹理图，按 0 出现切割线图，再按一次 0 退出程序。

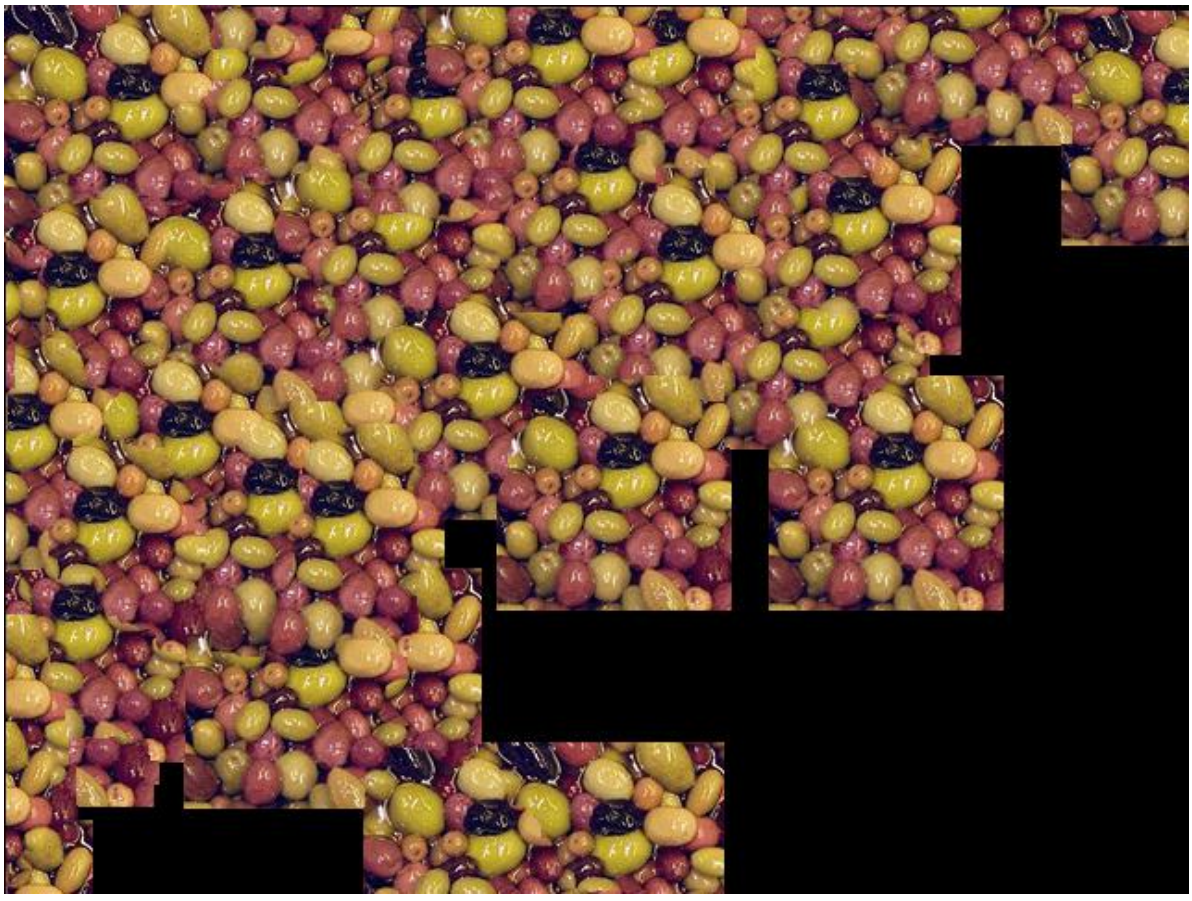
4 实验结果总结

以网页中的 fruit.png 和 lily.bmp 为例（可以于 main 函数中调整测试）

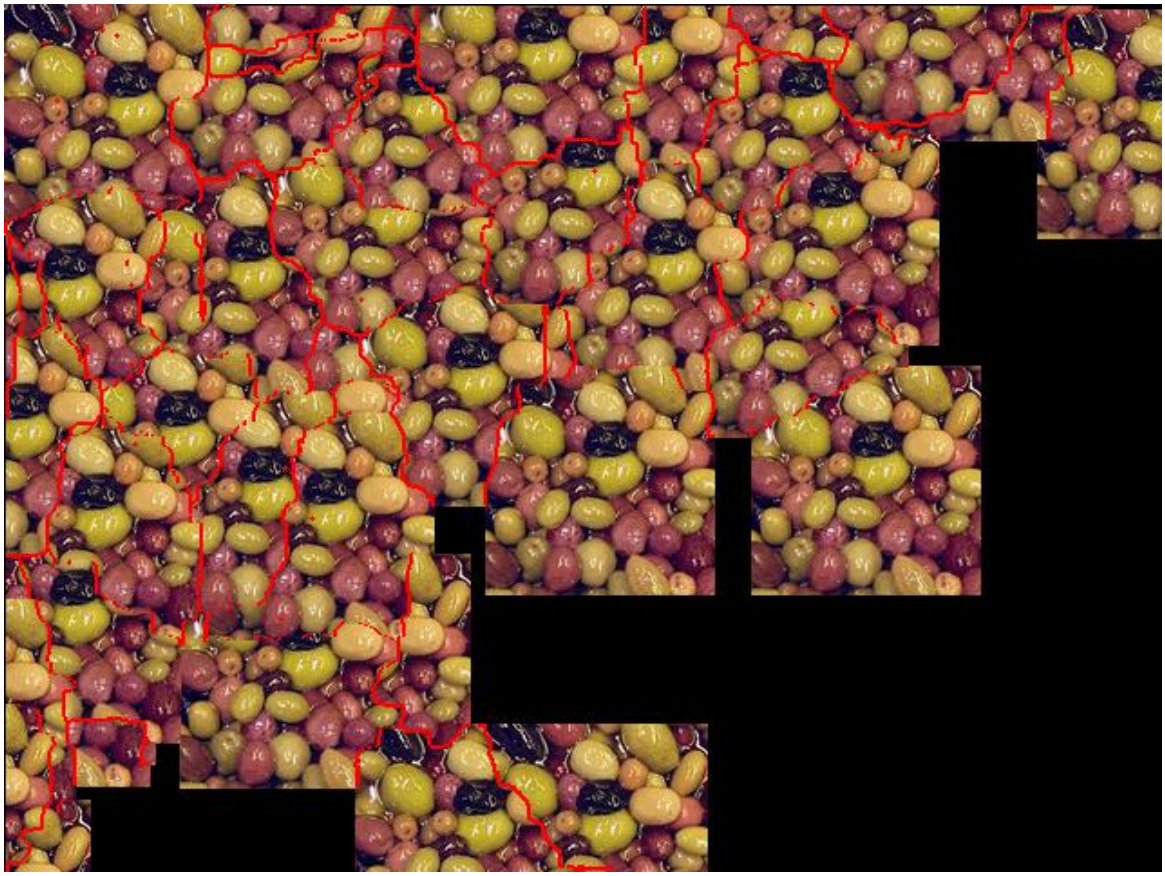
（1）Fruit.png 实验结果

原图

纹理图



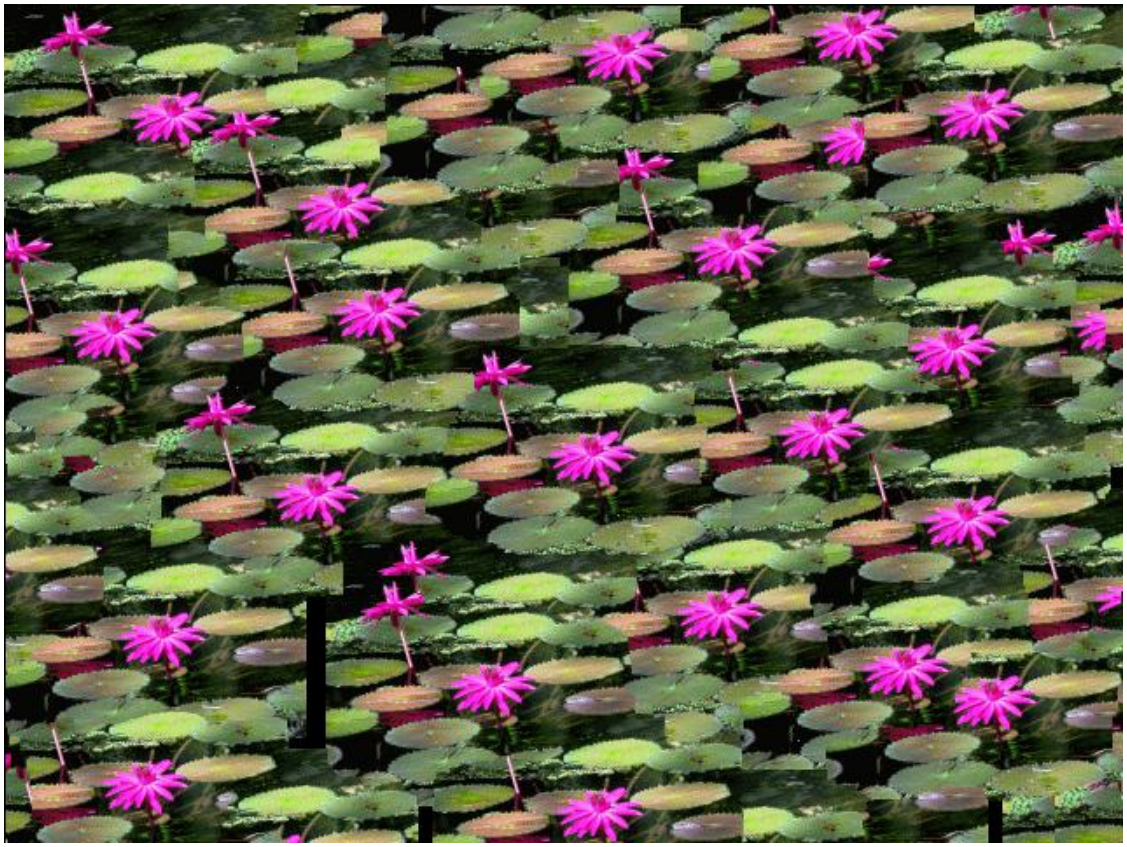
切割线图



（2）Lily.bmp 实验结果

原图

纹理拼接图



纹理切割图