山东大学 计算机科学与技术 学院

计算机视觉 课程实验报告

|  |  |  |
| --- | --- | --- |
| 学号：201900130151 | 姓名： 莫甫龙 |  |
| 实验题目：图像结构 | | |
| 实验过程中遇到和解决的问题：  （记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）   1. 连通域   首先先将图像转化为二值图像，方便后面的处理，这边阈值设为127。      接着就是实现8连通的快速连通域算法，和4连通的快速连通域算法差不多，都是要进行两遍遍历，第一次是将每个白色的像素点都给上一个标记，第二次就是处理等价对。而处理等价对，我使用的是并查集。八连通的标记算法用的是如下算法，在其上稍微做了一点改变：    对于第二和第三种情况，可以看出是会存在等价对的情况的，在出现这种情况的时候就将当前点标记为最小的标记点，如果这两个点的标记不一样，那么就将这两个点的标记加入并查集。    那么在进行完这一系列操作以后，我们就能得到很多棵树，这时候，只需要将每个像素点的标为置为根结点的标记即可，也就是合并连通域。    接着就是统计每个连通域的个数和连通域的面积，然后将最大面积的连通域输出即可。        下面这张图是通过opencv自带的connectedComponentsWithStats函数计算的连通域来实现的    通过不断地调开始转换为二值图像的阈值发现，当阈值变大的时候，马身下的草地就越少。  这是阈值为200时的效果图。    又换了张图来运行。      2.距离变换：  直接调用distanceTransformhanshu就行。 | | |
| 结果分析与体会：  八连通的快速连通算法，和四连通的快速连通算法比起来，就是要多考虑左上角和右上角两个点，虽然只是多了两个点，但是其中涉及的情况却多了12种，而上面的八邻域标记算法中，它按照左边，左上，上面，右上的顺序来标记当前点，只需要考虑左上和右上、左边和右上两种特殊情况即可，这就大大减少了考虑的情况。  而distanceTransform这个函数主要用于计算非零像素到最近零像素点的最短距离。一般用于求解图像的骨骼。 | | |