

图像分割章节实验报告

1.Canny算子边缘检测

基本步骤如下：

1. 高斯滤波：

对图像进行高斯模糊，平滑图像，去除噪声。

2. 梯度计算：

使用Sobel算子等计算图像中每个像素的梯度幅值和方向，识别可能的边缘。

3. 非极大值抑制（NMS）：

对梯度图像进行细化，只保留在梯度方向上局部最大值的像素点，去除非边缘像素，使边缘更细更清晰。

4. 双阈值检测：

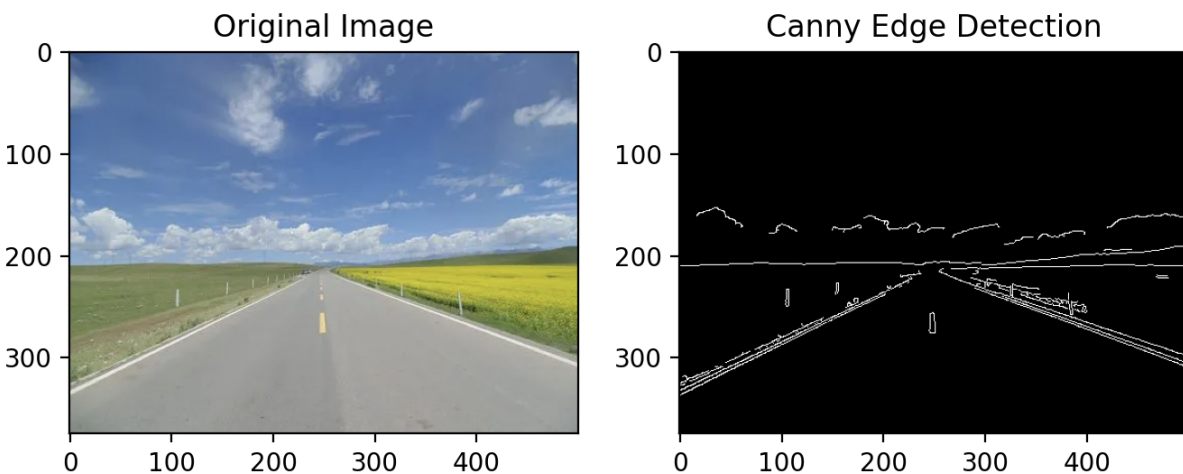
设置一个高阈值和一个低阈值：

- 高于高阈值的像素被认为是“强边缘”；
- 介于高低阈值之间的像素为“弱边缘”；
- 低于低阈值的像素被舍弃。

5. 边缘连接（滞后阈值）：

将与强边缘相连的弱边缘保留下来，其余弱边缘舍弃。

原图像与边缘检测图像对比：



对于本例而言，边缘检测可以推知主要应用于车道、障碍物检测。因此，正确保留车道边缘和中心线即可。展示的结果中，高斯模糊 $\sigma = 0.5$ ；Sobel算子宽度为3；上下阈值分别为200、90，符合书中描述的2:1-3:1规律。经过实验，大约在这个参数范围内，边缘检测的结果最好：清晰展示指示标线、路肩，近处障碍物。当然，天上的一些云混了进去，不过加大高斯模糊将影响车道侧障碍物检测效果，看着也很无聊，因此保留了，这样你才知道你看到的是风景照。

2.Hough变换直线检测

步骤如下：

1. 边缘检测：

对原始图像使用Canny等算法提取边缘。

2. 建立参数空间：

将图像空间中的每个边缘点 (x,y) 映射到极坐标空间 (ρ,θ) 。

3. 参数累加投票：

在参数空间中对每个可能的 (ρ,θ) 进行累加计数。

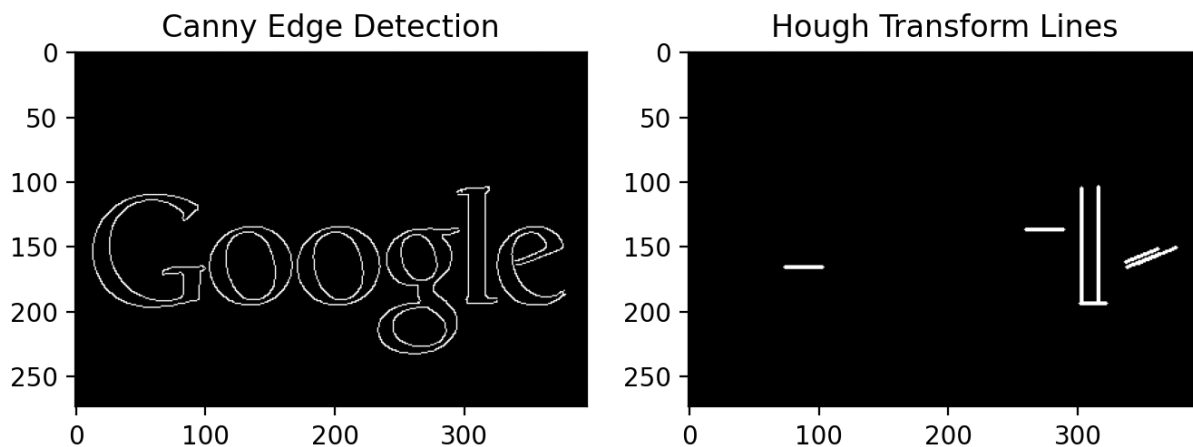
4. 寻找局部最大值：

找到累加器中高于设定阈值的点，表示图像中存在对应直线。

5. 返回直线信息：

将参数空间中的结果转换为图像中的直线。

边缘检测图像与直线图像对比：



原始的Hough变换直线检测方法会划出——这很讨厌——贯穿长虹的直线，因此使用其变种：概率Hough变换。相对于原始方法，他会根据给定的间距长度检测直线起讫端，由此给出符合原图长度的边缘。在本例中，像素分辨率选择1，角度分辨率设为 $\frac{\pi}{8}$ ，线段最短设20像素，最大间隔设5像素，结果不错——除了一些不知道为什么它不认为是直线的部分（比如l的最上沿），他几乎都检测出来了。

3.图像分割

基于图的方法可以很好地利用图像像素点间的相似性。

1. 图构建：

将图像表示为一个无向图，图中的每个像素或像素块为一个节点，相邻像素之间建立边。

2. 计算边权重：

为图中的边分配权重，通常基于颜色、灰度或纹理差异等特征，表示像素间的相似度。

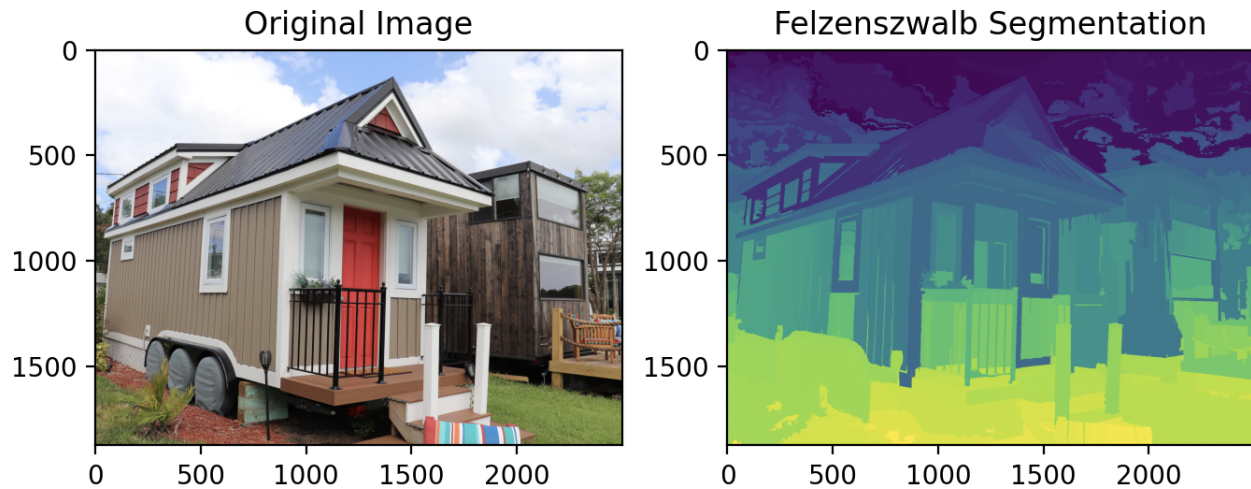
3. 图划分（分割）：

根据边的权重，使用图论算法（如最小割、区域合并等）将图划分为若干子图，每个子图对应图像中的一个区域。

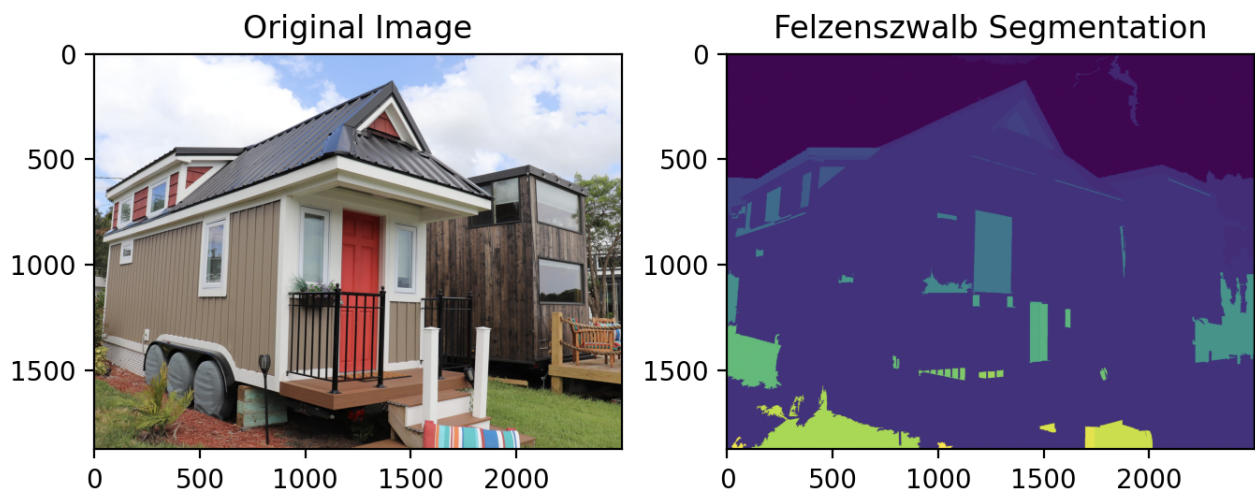
4. 生成分割结果：

将每个子图还原为图像中的区域，得到最终的分割图像。

原始图像与分割图像对比：



在眼花缭乱的各种图分割方法中，本例中采用是Felzenszwalb分割方法。尽管也是一个比较老的算法，他的复杂度已经接近线性，适合比较大的图像处理；同时，它相对而言更加考虑邻接因素。这类算法还有一个好处——不需要考虑要把图像分割为多少块。本例中，考虑到“区域要是什”这个问题并不明显，最小块大小及规模大小均设为比较符合大多数色块大小的500，高斯模糊的 $\sigma = 0.5$ 。如果更为完整的区域，譬如整个房子，可以将规模大小调大。如下图。最小块调成1000，规模调成3000。



值得注意的是，图算法的时间开销相对比较大。因此，对于比较简单的分割任务，还是以阈值处理这样的传统方法更为格算。