



## 数字图像处理与机器视觉 实验报告

作业名称 \_\_\_\_\_ HW3

姓 名 \_\_\_\_\_ 杨逍宇

学 号 \_\_\_\_\_ 3220105453

电子邮箱 \_\_\_\_\_ 3220105453@zju.edu.cn

联系电话 \_\_\_\_\_ 13518290755

导 师 \_\_\_\_\_ 蔡声泽/曹雨齐/姜伟



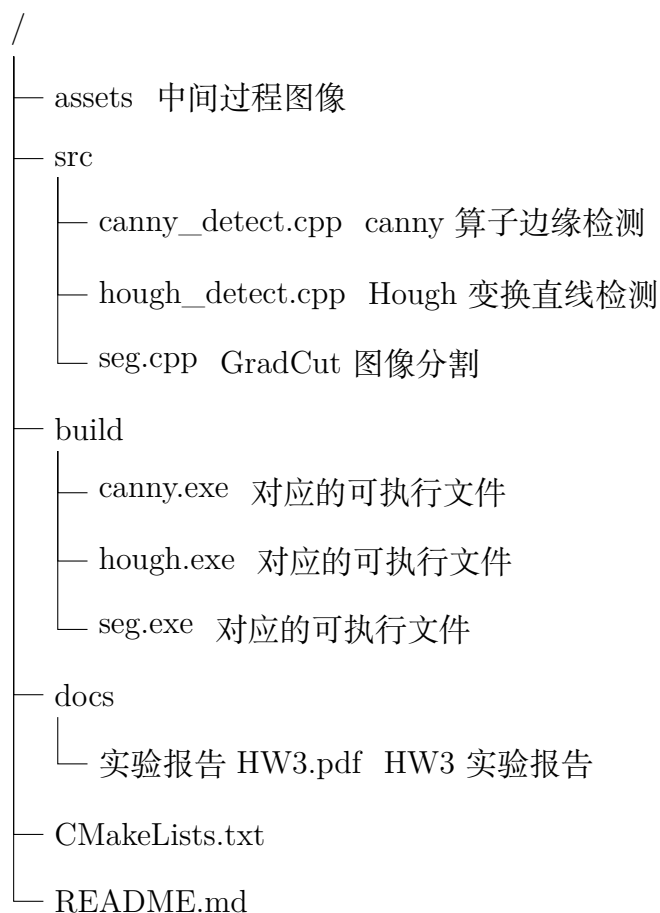
2025 年 4 月 4 日

# 1 已实现的功能简述及运行简要说明

## 1.1 已实现的功能简述:

- (1). `canny_detect.cpp`, `hough_detect.cpp` 分别实现了要求中的图像 A,B,C 的处理和检测。
- (2). 基于 GrabCut 的图像分割方法。
- (3). 运行相关的可执行文件, 会显示中间运行过程中的图像
- (4). 输出路径为 `assets/result` 目录

项目目录树如下:



## 2 开发与运行环境

本实验使用的软件和工具如下:

- 开发环境: Visual Studio Code on Ubuntu22.04

- 编程语言: C++
- 库: OpenCV 4.7.0
- 构建工具: CMake

### 3 算法基本思路

#### 3.1 Canny 边缘检测

- (1). **灰度转换**: 将彩色图像转换为单通道灰度图像
- (2). **高斯模糊**: 使用  $5 \times 5$  高斯核进行平滑处理, 消除噪声干扰
- (3). **梯度计算**: 通过 Sobel 算子计算图像梯度幅值和方向
- (4). **非极大值抑制**: 保留梯度方向上的局部最大值
- (5). **双阈值检测**: 低阈值 50 和高阈值 120 进行边缘连接

#### 3.2 Hough 直线检测

- (1). **边缘预处理**: 先进行 Canny 边缘检测获取二值边缘图
- (2). **参数空间离散化**: 设置 分辨率 1 像素, 分辨率 1 度
- (3). **累加器投票**: 对每个边缘点进行直线方程参数投票
- (4). **峰值检测**: 设置最小投票数阈值 100 筛选有效直线
- (5). **线段合并**: 合并相邻线段, 设置最小长度 50 像素, 最大间隔 10 像素

#### 3.3 相似性分割方法

- (1). **预处理阶段**:
  - 高斯模糊消除高频噪声 ( $3 \times 3$  核)
  - 自适应阈值二值化 (OTSU 算法)
  - 形态学开闭运算优化目标区域形态
- (2). **标记生成**:
  - 距离变换识别前景种子区域

- 形态学膨胀获取背景区域
- 未知区域计算：原图 - 前景 - 背景

### (3). GrabCut 优化：

- 基于标记图像初始化 GMM 模型
- 迭代优化前景/背景概率分布
- 最终掩膜生成与结果提取

## 4 算法实现要点

### 4.1 Canny 边缘检测核心代码

```
1 // 高斯模糊（代码片段）
2 GaussianBlur(gray, blurred, Size(5, 5), 1.5);
3
4 // Canny边缘检测（关键参数）
5 double lowThreshold = 50;
6 double highThreshold = 120;
7 Canny(blurred, edges, lowThreshold, highThreshold);
```

### 4.2 霍夫变换核心代码

```
1 // 霍夫直线检测参数设置
2 HoughLinesP(edges, lines,
3     1, // rho分辨率
4     CV_PI/180, // theta分辨率
5     100, // 累加器阈值
6     50, // 最小线段长度
7     10); // 最大线段间隙
8
9 // 直线绘制（带抗锯齿）
10 line(result, Point(l[0], l[1]), Point(l[2], l[3]),
11     Scalar(0, 0, 255), 2, LINE_AA);
```

### 4.3 分水岭与 GrabCut 整合

```
1  Mat grab_mask(bg_mask.size(), CV_8UC1, GC_PR_BGD); // 初始化可能背景
2  grab_mask.setTo(GC_FGD, bg_mask); // 确定前景
3  grab_mask.setTo(GC_BGD, fg_mask); // 确定背景
4
5  Mat bgdModel, fgdModel;
6  grabCut(src, grab_mask, Rect(), bgdModel, fgdModel, 3,
7          GC_INIT_WITH_MASK);
8
9  // 生成最终掩膜
10 Mat result_mask;
11 compare(grab_mask, GC_FGD, result_mask, CMP_EQ);
12 // 创建3通道掩膜用于乘法
13 Mat result_mask_3ch;
14 cvtColor(result_mask, result_mask_3ch, COLOR_GRAY2BGR);
15
16 // 与原图合成（按位与操作）
17 Mat result;
18 bitwise_and(src, result_mask_3ch, result);
```

## 5 实验结果及分析

### 5.1 Canny 边缘检测结果

不同阈值对比实验：



(a) 图像 A 的边缘检测

(b) 图像 B 的边缘检测

(c) 图像 C 的边缘检测

Figure 1: Canny 算子边缘检测效果分析

实验表明：

- 低阈值过小会保留过多细节噪声（如 50）
- 高阈值过大会丢失重要边缘（如 200）
- 双阈值比例建议保持在 1:2 到 1:3 之间

## 5.2 霍夫直线检测结果



(a) 图像 A 的直线检测

(b) 图像 B 的直线检测

(c) 图像 C 的直线检测

Figure 2: Hough 变换直线检测效果分析

结果分析：

- 累计器阈值过低会导致误检增多
- 线段最小长度设置过小会产生碎片化线段
- 最大间隔参数影响断线连接效果

### 5.3 分割流程可视化

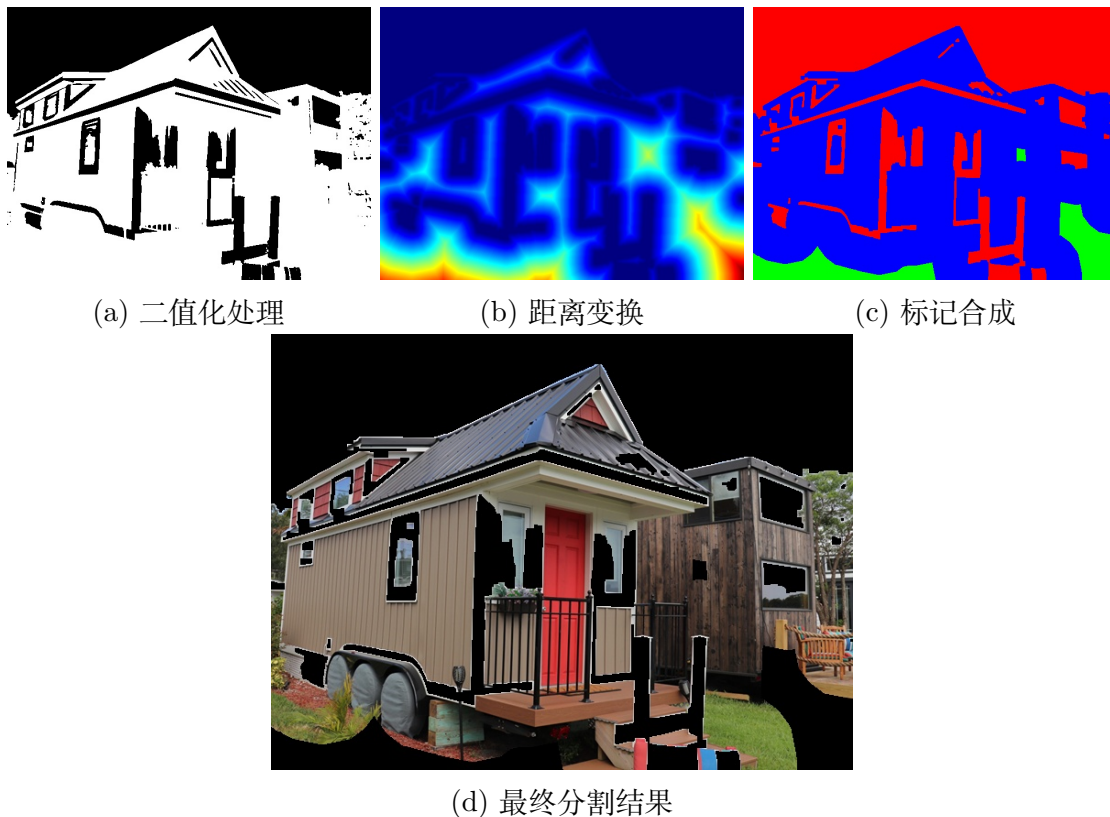


Figure 3: 图像分割处理流程

关键参数影响分析：

- 高斯模糊核大小： $3 \times 3$  核在保留边缘的同时有效抑制噪声
- 形态学迭代次数：2 次开运算消除细小噪点，2 次闭运算填充孔洞
- 距离变换阈值：0.5 倍最大值平衡前景区域完整性
- GrabCut 迭代次数：3 次迭代实现快速收敛

## 6 结论与心得体会

通过本实验，我们得出以下结论：

- Canny 算子的双阈值机制能有效平衡噪声抑制与边缘连续性
- 高斯模糊核大小显著影响边缘检测的精度与抗噪性
- 霍夫变换参数需要根据具体场景进行针对性调整

- GrabCut 结合颜色相似性与空间连续性，有效改善复杂背景分割
- 距离变换可有效识别目标中心区域，为标记生成提供可靠依据

实验中的改进思考：

- 可尝试自适应阈值确定方法（如 OTSU 算法）
- 结合边缘方向信息优化霍夫投票策略
- 对检测到的直线进行角度聚类分析