

# 合肥工业大学

## 《机器视觉》课程实验

实验题目 课程实验二：车道线检测

学生姓名 高志鸿

学 号 2023217584

专业班级 智能科 23-3 班

## 一、实验目的

- 掌握传统车道线检测的经典流水线: 去噪 → 边缘检测 → ROI 限定 → 直线段检测 → 鲁棒拟合 → 可视化叠加。
- 理解 Canny 阈值自适应设置方法, 降低不同光照条件下的调参成本。
- 理解 HoughLinesP 检测线段的参数含义与局限, 并用 RANSAC 提升抗噪鲁棒性。
- 输出可用于报告展示的中间结果图(边缘图)与最终结果图(车道线叠加)。

## 二、实验环境

- 操作系统: Windows
- 语言与库: Python , OpenCV(cv2)、NumPy、scikit-learn(`RANSACRegressor`)
- 实验脚本: `exp2/detect.py`
- 输入/输出目录(本项目默认):
  - 输入: `exp2/data/`
  - 输出: `exp2/output\_images/`

## 三、实验原理与方法

本实验采用传统几何/统计方法完成车道线检测, 核心思想是: 车道线在图像中通常表现为两条近似直线(或局部直线段), 通过边缘提取与直线拟合可以得到稳定的车道线位置。

### 1. 边缘检测与去噪(`get\_edge\_img`)

- 先对原图进行高斯滤波 `cv2.GaussianBlur` (默认核大小 `gaussian\_ksize=5`, `gaussian\_sigmax=1`), 抑制噪声导致的伪边缘。
- 转灰度后使用 Canny 得到边缘图: `cv2.Canny(gray, threshold1, threshold2)`。

### 2. Canny 阈值自适应(`auto\_canny\_thresholds`)

- 为减少手动调参, 使用灰度中位数  $v = \text{median}(I)$  估计阈值区间:  
为  $T = (1 \pm \sigma) \times \text{median}(I)$   
代码默认 `sigma=0.33` (对于自然图像来说, 有效区域灰度均在此区间内), 并对阈值做边界保护(保证 `high > low` )。
- 本项目示例图 `road.jpg` 的实际统计结果为: `median=102.0`, 因此得到 `low=68`、`high=135` (由脚本同公式计算得到), 使得同一套代码对不同亮度图片更稳健。

### 3. ROI (感兴趣区域) 约束 (`roi\_mask` + `default\_roi\_pts`)

- 车道线主要出现在图像下半部分，使用四边形 ROI 掩膜只保留道路区域边缘，显著减少天空、建筑等无关边缘干扰。
- ROI 顶点由图像宽高按比例生成 (`default\_roi\_pts(width, height)`):
  - $(0.08w, 0.95h), (0.45w, 0.60h), (0.55w, 0.60h), (0.92w, 0.95h)$

### 4. 线段检测 (`cv2.HoughLinesP`)

- 在 ROI 边缘图上执行概率霍夫变换提取直线段:
  - `rho=1`、`theta= \pi / 180`、`threshold=15`
  - `minLineLength=35`、`maxLineGap=20`
- 得到的线段往往包含噪声、断裂段和非车道线段，需要进一步筛选与拟合。

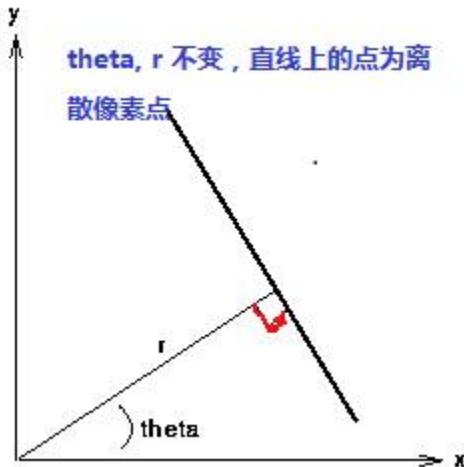


Figure 1 霍夫变换核心原理图

### 5. 斜率分组+异常剔除 (`calculate\_slope` / `reject\_abnormal\_lines`)

- 先按斜率符号将线段分为两类：斜率  $> 0$  视为左侧候选，斜率  $< 0$  视为右侧候选。
- 计算候选线段斜率集合，使用“与均值差异”迭代剔除异常线段（默认阈值 `0.3`），减少离群线段对拟合的影响。

### 6. RANSAC 鲁棒直线拟合 (`ransac\_fit`)

将候选线段端点展开为点集(x,y)，用 `RANSACRegressor(residual\_threshold=10, max\_trials=200)` 拟合线性模型  $y = ax + b$ 。

- RANSAC 通过随机采样与内点投票机制，在存在大量噪声点时仍能得到稳定的直线估计。拟合完成后对内点再用 `np.polyfit(..., deg=1)` 得到最终直线参数，并输出端点用于绘制，相比最小二乘法而言，具有更好的鲁棒性。

RANSAC 通过反复选择数据中的一组随机子集来达成目标。被选取的子集被假

设为局内点，并用下述方法进行验证：

- 一个模型适用于假设的局内点，即所有的未知参数都能从假设的局内点计算得出。

- 用 1 中得到的模型去测试所有的其它数据，如果某个点适用于估计的模型，认为它也是局内点。

- 如果有足够的点被归类为假设的局内点，那么估计的模型就足够合理。

- 然后，用所有假设的局内点去重新估计模型，因为它仅仅被初始的假设局内点估计过。

最后，通过估计局内点与模型的错误率来评估模型。

这个过程被重复执行固定的次数，每次产生的模型要么因为局内点太少而被舍弃，要么因为它比现有的模型更好而被选用。

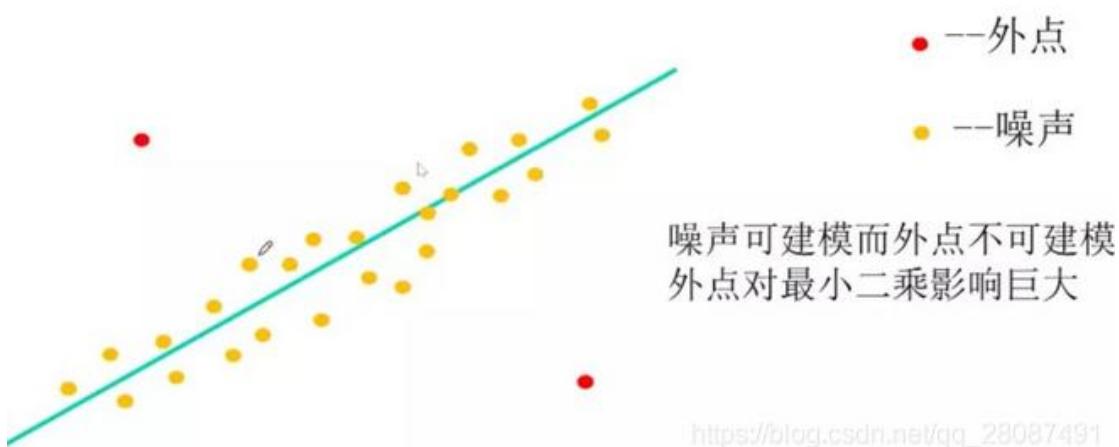


Figure 2 RANSAC 运作原理

## 7. 结果绘制与可视化 (`draw\_lines`)

- 将左右车道线绘制到原图副本上：

- 红色粗线：RANSAC 拟合端点连线（`thickness=3`）

- 绿色细线：按 y 范围做延长后的直线（`thickness=1`），使车道线更直观连续。

## 四、实验内容与步骤

- 将待处理图片放入 `exp2/data/`（本实验示例为 `road.jpg`）。

- 运行脚本进行批处理（脚本会遍历目录下所有常见图片格式）：

- `python exp2/detect.py --data-dir exp2/data --output-dir exp2/output\_images`

- 对每张图片，`process\_one\_image()` 依次执行：

- 读取原图 `cv2.imread`

- 计算 ROI 顶点 `default\_roi\_pts`

- 自适应 Canny 阈值 `auto\_canny\_thresholds`

- 边缘提取 `get\_edge\_img`

- ROI 掩膜 `roi\_mask`

- Hough 线段检测 + 筛选 + RANSAC 拟合 `get\_lines`
- 叠加绘制 `draw\_lines`
- 输出文件（以 `stem=road` 为例）保存在输出目录：
  - `road\_edges.png`：全图 Canny 边缘结果
  - `road\_lane.png`：最终车道线叠加结果
  - 若加 `--save-debug`，还会额外保存：
    - `road\_roi\_edges.png`：ROI 内边缘图（调试用）

## 五、实验数据记录

- 输入图像：`exp2/data/road.jpg`
- 自适应 Canny 阈值 ( $\sigma=0.33$ )：
  - 灰度中位数 `v=102.0`
  - `threshold1=68`，`threshold2=135`
- 输出结果文件：
  - `exp2/output\_images/road\_edges.png`
  - `exp2/output\_images/road\_lane.png`

## 六、实验结果与分析

### 1. 实验结果



Figure 3 input\_img

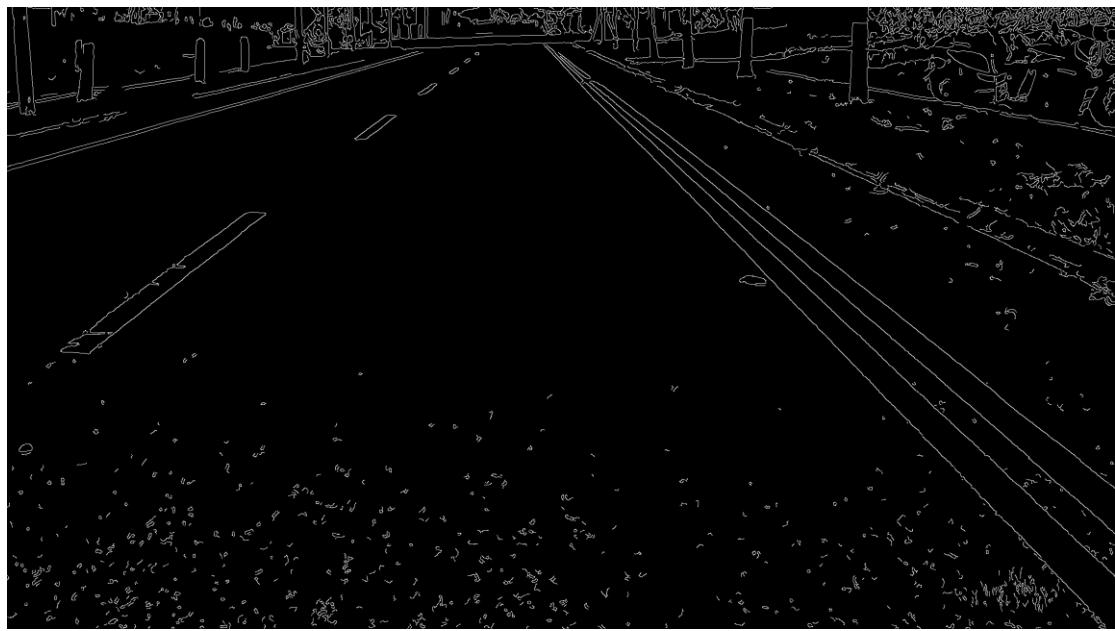


Figure 4 road\_edges



Figure 5 road\_lane

## 2.结果分析

- 从 `road\_edges.png` 可以观察到：车道线、路缘、阴影边界、路面纹理等都会产生边缘响应。若直接在全图做直线检测，极易被无关边缘干扰，导致误检或车道线抖动。
- 引入 ROI 后，检测范围聚焦在道路区域，干扰边缘显著减少，Hough 得到的线段更集中于车道线附近，后续拟合更稳定。
- HoughLinesP 输出的是大量短线段，且可能包含离群线段（来自裂纹、阴影、标线噪声）。通过斜率分组与 `reject\_abnormal\_lines` 先做粗筛，再用 RANSAC 做鲁棒拟合，可以有效抑制离群点影响，使最终 `road\_lane.png` 中车道线更连续、更符合直观几何结构。

- 自适应 Canny 阈值相对固定阈值更适合不同亮度场景：当图像整体变亮/变暗时，中位数会随之变化，阈值自动调整，避免“阈值偏高导致漏检”或“阈值偏低导致噪声爆炸”的问题。

## 七、结论

- 本实验实现了基于传统视觉方法的车道线检测完整流程，并能输出边缘中间结果与最终叠加结果图。
- ROI 约束与 RANSAC 鲁棒拟合是提升稳定性与抗噪性的关键：ROI 负责“减少干扰来源”，RANSAC 负责“在残留干扰下稳定拟合”。
- 自适应 Canny 阈值策略减少了跨图像场景的人工调参需求，使流程更具工程可用性。

## 八、实验体会与个人理解

这次车道线检测实验让我对“传统视觉流水线”的价值有了更深的认识：它并不是简单地把几个算子拼在一起，而是通过明确的先验假设把问题逐步收敛。车道线大多位于画面下方、呈近似直线，因此先用 Canny 提取可能的边缘，再用 ROI 把注意力集中在道路区域，最后用 Hough+拟合把大量局部线索整合成两条稳定车道线。整个过程每一步都有清晰的可解释输出（如 edges 图、ROI 后的边缘、最终叠加图），这让调试和答辩表达都非常直接：哪里出了问题就看对应中间结果，很容易定位是阈值、ROI、线段检测还是拟合环节导致。

我体会最明显的是“参数敏感性”和“自适应策略”的重要性。固定的 Canny 阈值在光照变化时很容易失效：画面变暗会漏掉车道线，画面变亮又会引入大量噪声边缘。本实验用灰度中位数配合  $\left(1\mu m\sigma\right)$  的方式自动生成阈值，使阈值会随整体亮度变化而同步调整，明显降低了反复手调参数的成本。虽然这种方法并不能解决所有复杂情况（例如强阴影、反光、雨雾），但它体现了一种工程化思路：让系统尽量依赖数据统计量而不是依赖人工经验，从而提升跨场景泛化能力。

在直线拟合部分，我进一步理解了“鲁棒性”为什么是工程里必须考虑的核心指标。HoughLinesP 得到的是许多短线段，其中混杂着阴影边缘、路面裂纹、标线断裂产生的离群线段；如果直接用最小二乘拟合，很容易被少量异常点带偏。

RANSAC 的内点投票机制等于在“噪声很大”的条件下主动寻找可信证据，把拟合建立在大多数一致的点上，因此车道线会更稳定、更贴近真实结构。与此同时，代码里对检测失败的回退（沿用上一条线）也让我感受到系统设计不仅要追求“理想情况下最准确”，更要保证“异常情况下不崩溃”，输出仍然可用、连续。

最后，这个实验也让我明确了传统方法的边界与改进方向：当车道线弯曲、标线磨损严重、光照极端或背景干扰很强时，单纯依靠边缘与直线模型会出现误检、断裂或抖动。后续若要提升效果，可以从两条路径入手：一是继续在传统框架内增强稳健性（更好的颜色/亮度归一化、更合理的 ROI 自适应、更强的时序平滑）；二是引入学习方法做语义分割/检测，再用几何拟合做后处理。通过本次实验，我不仅掌握了完整流程，也形成了“用中间可视化结果驱动调参、用鲁棒估计抵抗

噪声、用容错机制保证稳定输出”的工程思维。