

**计算机视觉实验报告**

**实验题目**  车道线检测

**学生姓名**  杜述超

**学 号**  2021214745

**专业班级**  21计科-1班

**指导教师**  吴晶晶

**完成日期**  2023.12.20

**合肥工业大学 计算机与信息学院**

# 实验目的

**车道线检测的目的**：车道线检测是自动驾驶和辅助驾驶系统中的关键部分，旨在从车辆摄像头拍摄的路面图像中识别和定位车道线，以辅助车辆导航。

**实验内容**：车道线检测是自动驾驶的基本模块。请使用霍夫变换实现车道线的检测。

**具体要求：**

➢ 任务输入：自己拍摄的校园中道路图像（画有车道线的路）。

➢ 任务输出：图像中车道线的位置。

➢ 代码语言不限，方法不限，要求提交整个算法源代码，模型结果，算法分析等内容。

# 实验原理

1. **图像边缘检测**：首先，需要对输入的道路图像进行边缘检测。这通常通过诸如Canny边缘检测器之类的算法实现，以识别图像中的路面和车道线边缘。
2. **霍夫变换**：

* **变换空间**：霍夫变换是一种特征提取技术，用于从图像中识别几何形状，特别是直线、圆等。对于直线检测，霍夫变换将图像空间中的每个点映射到霍夫空间中的一条线上。霍夫空间是一个参数空间，对于直线可以用两个参数表示，通常选择直线的斜率和截距，或者极坐标中的角度和距离。
* **投票机制**：在霍夫空间中，一条直线上的所有点会在相同的位置（或附近）累积投票。这些累积的投票可以用于识别原始图像中的直线，即多个点共线的情况。

1. **车道线检测**：

* **候选直线提取**：通过霍夫变换，从边缘检测后的图像中提取候选直线。
* **车道线识别**：由于道路图像中的车道线具有特定的方向和位置特征，可以通过设置适当的阈值和约束条件（如直线的长度、方向等）来从候选直线中筛选出实际的车道线。

# 实验内容

**霍夫变换算法思路：**

**HoughLinesP**函数是霍夫变换的一个变种，用于在二值图像中检测直线段。这个函数的核心思路涉及以下几个关键步骤：

1. **霍夫空间映射**：
   * 对于图像中的每一个非零像素点（边缘点），考虑所有可能通过该点的直线。
   * 这些直线在霍夫空间中用极坐标表示（ρ-θ空间），其中**rho**是直线到原点的垂直距离，**theta**是直线的倾斜角度。
   * 对于每个边缘点，将ρ-θ空间中的相应点增加一个计数（累积数组）。
2. **阈值判断和直线选择**：
   * 在累积数组中，任何超过指定**threshold**的值都被认为是直线候选。
   * 这些候选被进一步处理以确定实际的直线，根据**minLineLength**（最小线长）和**maxLineGap**（线段间最大间隙）参数。
3. **直线提取**：
   * 最终，函数返回由各直线段端点坐标构成的集合，每个直线段由起点和终点两个点表示。

**霍夫变换核心代码如下：**

void HoughLinesP(cv::InputArray image, cv::OutputArray lines,

double rho, double theta, int threshold,

double minLineLength = 0, double maxLineGap = 0)

{

// 确保输入是二值图像

cv::Mat binary;

if (image.channels() == 1) {

binary = image.getMat();

}

else {

cv::cvtColor(image, binary, cv::COLOR\_BGR2GRAY);

cv::threshold(binary, binary, 128, 255, cv::THRESH\_BINARY);

}

// 累积数组的初始化

int rows = binary.rows, cols = binary.cols;

int numThetas = int(std::round(CV\_PI / theta));

int maxRho = std::sqrt(rows \* rows + cols \* cols);

cv::Mat accum = cv::Mat::zeros(2 \* maxRho, numThetas, CV\_32SC1);

// 遍历图像的每个像素

for (int y = 0; y < rows; ++y) {

for (int x = 0; x < cols; ++x) {

if (binary.at<uchar>(y, x) > 0) { // 边缘点

for (int t = 0; t < numThetas; ++t) {

double currentTheta = t \* theta;

int rhoValue = int(x \* std::cos(currentTheta) + y \* std::sin(currentTheta));

rhoValue += maxRho; // 调整范围以避免负索引

accum.at<int>(rhoValue, t)++;

}

}

}

}

// 提取直线

std::vector<cv::Vec4i> lineSegments;

for (int r = 0; r < accum.rows; ++r) {

for (int t = 0; t < accum.cols; ++t) {

if (accum.at<int>(r, t) > threshold) {

// TODO: 根据minLineLength和maxLineGap提取直线段

// 此处省略直线提取的具体实现

}

}

}

// 将检测到的线段转换为输出格式

cv::Mat(lineSegments).copyTo(lines);

}

**检测车道线核心算法思路：**

1. 边缘检测：使用Canny算法从灰度图像中检测边缘。Canny边缘检测器是一种流行的边缘检测算法，用于识别图像中的边缘。
2. 区域感兴趣(ROI)分割：定义一个三角形区域，该区域大致对应于图像中道路的前方部分。然后创建一个掩膜，以该区域为三角形形状，并使用cv::fillPoly函数填充。之后，使用cv::bitwise\_and函数将掩膜应用于Canny边缘检测后的图像，以便专注于该特定区域。
3. 霍夫线变换：在分割后的图像上应用霍夫线变换，以检测图像中的直线。这是通过HoughLinesP函数实现的，该函数能够识别图像中的直线片段。
4. 绘制检测到的线：在原始彩色图像上绘制检测到的线条。使用cv::line函数在图像上绘制每条检测到的直线。

**核心代码：**

int main()

{

// 读取道路图像

std::string path = "./image/road4.png";

cv::Mat img = cv::imread(path, cv::IMREAD\_GRAYSCALE);

cv::Mat clo = cv::imread(path, cv::IMREAD\_COLOR);

if (img.empty())

{

std::cout << "read image failed" << std::endl;

system("pause");

return 0;

}

// 使用Canny算法进行边缘检测

cv::Mat edges;

int lowThreshold = 100, highThreshold = 200;

int kernel\_size = 3;

Canny(img, edges, lowThreshold, highThreshold, kernel\_size);

// 提取感兴趣区域

cv::Mat segment\_result = segment(edges);

// 应用霍夫线变换检测直线

std::vector<cv::Vec4i> result;

HoughLinesP(segment\_result, result, 1, CV\_PI / 180, 20, 30, 10);

// 在原始彩色图像上绘制检测到的线条

for (int i = 0; i < result.size(); i++)

{

cv::Vec4i line = result[i];

cv::line(clo, cv::Point(line[0], line[1]), cv::Point(line[2], line[3]), cv::Scalar(255, 0, 0), 2);

}

// 显示和保存结果

cv::imshow("result", clo);

cv::imwrite("./image/result.jpg", clo);

cv::waitKey(0);

return 0;

}

**运行结果：**

****

图表 1 车道线检测结果

这里可以看出成功检测出了车道线的直线，这里的右边的线只检测了一定长度，这是因为霍夫变换函数参数的设置导致的，可以变化参数来调整长度。

# 总结和感想体会

**实验目的与重要性：**

进行这个实验的主要目的是理解和实现车道线检测的基本概念和方法，这对于自动驾驶和驾驶辅助系统至关重要。通过识别和定位车辆摄像头捕获的路面图像中的车道线，我们可以为车辆提供重要的导航和定位信息。这不仅增强了安全驾驶，也是自动驾驶技术中的一个关键步骤。

**实验过程与方法：**

我选择使用霍夫变换来实现车道线检测。霍夫变换是一种经典的图像处理技术，用于从图像中检测各种形状，尤其适用于直线检测。实验分为几个步骤：

1. 图像采集：首先，我拍摄了校园内有明显车道线的道路图像。
2. 边缘检测：使用Canny算法对图像进行边缘检测，以准确识别出图像中的车道线边缘。
3. 霍夫线变换：将边缘检测后的图像应用霍夫变换，检测图像中的直线，这些直线代表车道线。
4. 结果分析：最后，我分析了检测到的车道线并将其绘制在原始图像上，以验证检测的准确性。

**实验结果与体会：**

实验结果显示，通过霍夫变换能够有效地检测出道路图像中的车道线。在实验中，我学习到了如何处理和分析图像数据，以及如何使用图像处理技术来解决实际问题。这个过程不仅提高了我的编程技能，也加深了我对自动驾驶技术中图像处理应用的理解。

此外，实验过程中遇到的挑战包括正确设置边缘检测的阈值和霍夫变换的参数。这些参数的选择对于最终结果的准确性至关重要。通过调整这些参数，我能够更好地理解它们对车道线检测效果的影响。

**总结：**

这次实验不仅让我对车道线检测技术有了深刻的理解，还让我认识到了图像处理技术在自动驾驶中的重要性。通过实践操作，我得以将理论知识应用于解决实际问题，这对我的学习和未来的研究都有极大的帮助。