**深 圳 大 学 实 验 报 告**

|  |
| --- |
| **课程名称 计算机视觉**  **项目名称 实验二：图像处理综合-路沿检测**  **学 院 计算机与软件学院**  **专 业 软件工程（腾班）**  **指导教师 沈琳琳、 周杰**  **报 告 人 黄亮铭 学号 2022155028**  **实验时间 2024年10月14日至2024年11月18日**  **实验报告提交时间 2024年11月18日** |

**教务处制**

# 一、实验目的与要求

**实验目的：**

1．熟悉图像处理基本操作；

2．掌握图像边缘检测原理；

3．掌握图像基本特征抽取以及在实际问题中的应用；

**实验要求：**

1. 实验提交文件为实验报告和相关程序代码，以压缩包的形式提交，实验报告命名规则为“计算机视觉-学号-姓名-实验报告2.doc”，其他文件打包成压缩文件，命名为“计算机视觉-学号-姓名-实验报告2-其他.zip”；

2. 所有素材和参考材料需列明出处，实验报告中的图片和程序代码建议标注个人水印或标识信息：姓名，班级，学号信息；

# 二、实验内容与方法

**实验内容：**针对给定的视频，利用图像处理基本方法实现道路路沿的检测；

提示：可利用Hough变换进行线检测，融合路沿的结构信息实现路沿边界定位（图中红色的点位置）。

****

# 三、实验步骤与过程

我将路沿检测的实现分为如下几个模块：

1. 视频处理：该模块主要负责将视频以帧的形式拆分成图像，将图像送到相应的函数进行处理，然后将处理后的图像重新合成为视频。
2. 图像预处理：该模块的作用是将图像转化为灰度图，减少颜色对后续检测的影响。同时，该模块还对灰度图进行了直方图均衡化处理，进一步增强图像。
3. 边缘提取：该模块利用高斯滤波和Canny算子提取图像的边缘，便于后续检测路沿。
4. Hough变换：该模块主要负责检测经过边缘提取后图像中的直线，并将直线的相关信息送往下一步绘制直线的函数进行处理。
5. 绘制直线：在Hough变换获取的直线中利用空间的结构特征和直线的几何特征对直线进行进一步地筛选，然后才在原始图像上绘制直线，最后将结果返回。

## 3.1视频处理

视频处理分为三部分，分别为视频读取、视频拆分和视频合成。

### 3.1.1视频读取

视频读取可以使用OpenCV，也可以使用ffmpeg（专业的音视频处理库）。为了方便后续处理图像，这里我选择使用OpenCV实现视频读取。

具体方法为使用videocapture函数读取视频。代码实现如下图所示。

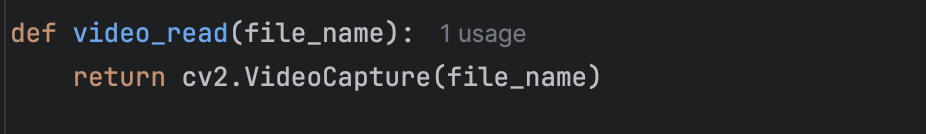


图1:视频读取

### 3.1.2视频拆分

视频拆分使用OpenCV实现。视频拆分实际上是将视频按帧分解成图像。

具体方法为使用read函数读入视频的每一帧，将当前帧作为一张图像进行后续的处理。代码实现如下图所示。

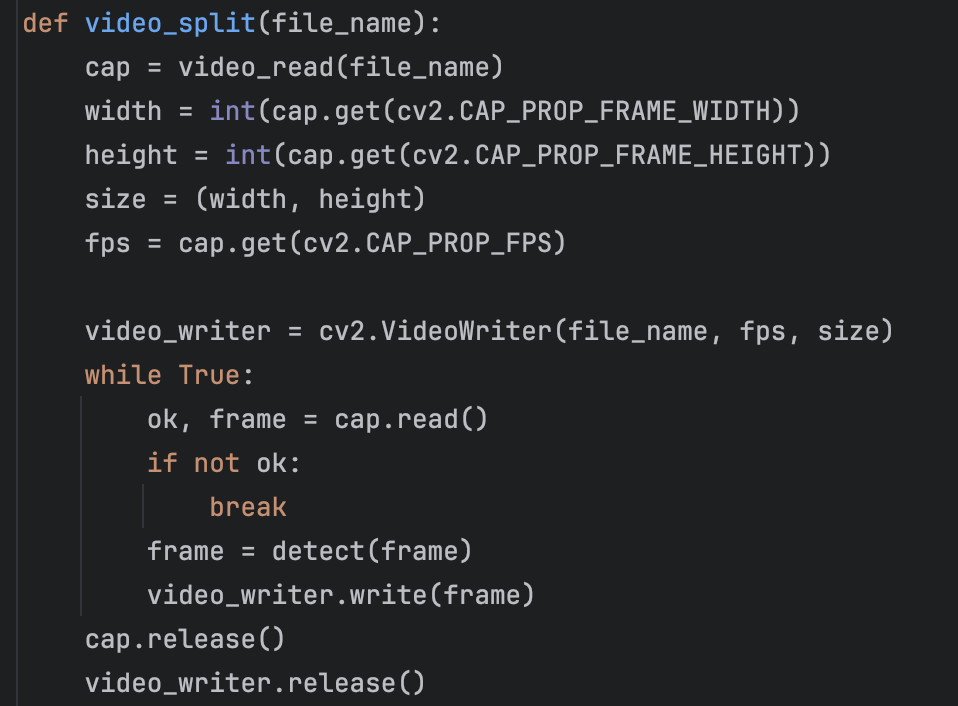


图2:视频拆分

### 3.1.3视频合成

视频合成同样使用OpenCV实现。在图像处理完成后，我们需要将图像重新合成为视频。

具体方法为首先使用videowriter函数创建自定义视频写入器，然后自定义编码格式，最后设置视频文件的存储位置。

实现代码如下图所示。下图所示创建了一个视频写入器，在图像处理完成后使用视频写入器的write函数即可将图像合成为视频。

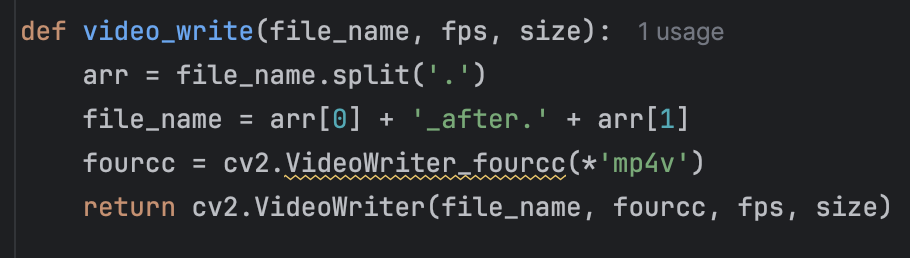


图3:视频合成

## 3.2图像预处理（灰度图、图像增强）

颜色信息对后续的边缘检测提供的有效信息较少，并且颜色本身非常容易收到光照因素的影响，导致边缘检测获取到错误的信息。因此，我们只需要获得灰度图中的信息即可。此外，转化成灰度图后，矩阵存储的信息减少，有利于提高运行速度。

实现彩色图像转化灰度图像非常简单，即将三通道的RGB图像转化为单通道图像。OpenCV库中已经有相应的函数实现。

具体代码实现如下图所示。



图4:彩色图转灰度图

此外，我们还可以使用直方图均衡化对图像进一步增强。该方法在OpenCV库中以有相应的实现，我们直接调用即可： 。

## 3.3边缘检测

### 3.3.1高斯滤波

高斯滤波是一种**线性平滑滤波**，适用于消除高斯噪声。高斯滤波每一个像素点的值，都由其本身和邻域内的其他像素值经过加权平均后得到。高斯滤波的具体操作是：用一个模板扫描图像中的每一个像素，然后用模板确定的邻域内像素的加权平均灰度值去替代模板中心像素点的值。

现实中的噪声都是随机分布的，因此使用高斯滤波去除高频噪声的效果会比较好。

高斯滤波在OpenCV已有对应的函数实现。

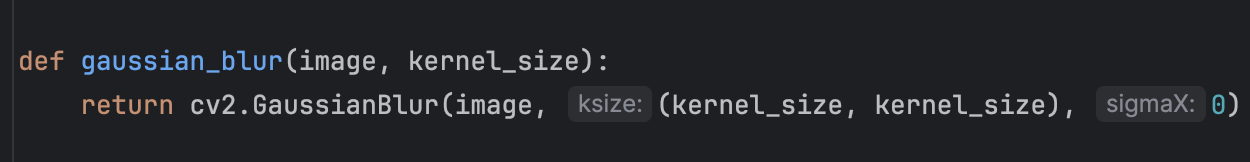


图5:高斯滤波

### 3.3.2Canny算子

Canny算子是一种**非微分边缘检测**算子，目标是找到一个最优的边缘检测解或找寻一幅图像中灰度强度变化最强的位置。

具体的检测步骤如下：

1. 使用高斯滤波器对图像进行平滑处理；
2. 使用Sobel算子来近似图像在x方向和y方向的梯度，然后计算梯度的强度和方向；
3. 使用非极大值抑制寻找局部像素点最大值，将非极大值对应的像素点灰度值置0，极大值对应的像素点置1，从而剔除非边缘像素点，确保边缘检测得到一条细线；
4. 使用双阈值连接，即选取两个阈值，将小于低阈值的点认为是假边缘置0，将大于高阈值的点认为是强边缘置1，介于两者之间的被标记为弱边缘。

OpenCV库中已有对应的函数实现，因此这里直接调用OpenCV的库函数进行边缘检测。具体代码如下图所示。

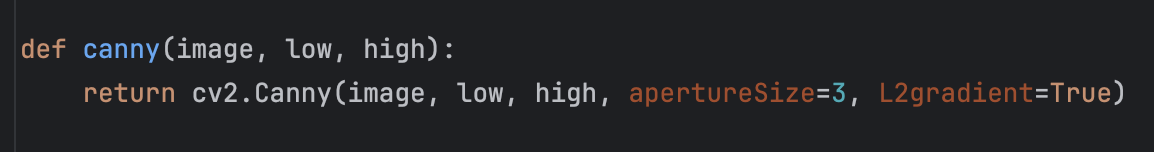


图6:Canny算子

### 3.3.3掩码矩阵

观察本实验给出的视频，发现路沿均在图像的右侧。说明我们可以使用掩码矩阵排除其他边缘的影响，只关注图像的右侧。

掩码矩阵的实具体现步骤如下：

1. 生成一个与原图像大小相同的矩阵，并将其初始化为0，即一张全黑的图像。
2. 将图像右侧的数值全部填充为255，即：一张左侧全黑右侧全白的图像。
3. 将掩码矩阵与边缘检测后的图像按位与，得到我们感兴趣区域的边缘信息。

具体代码实现如下图所示。

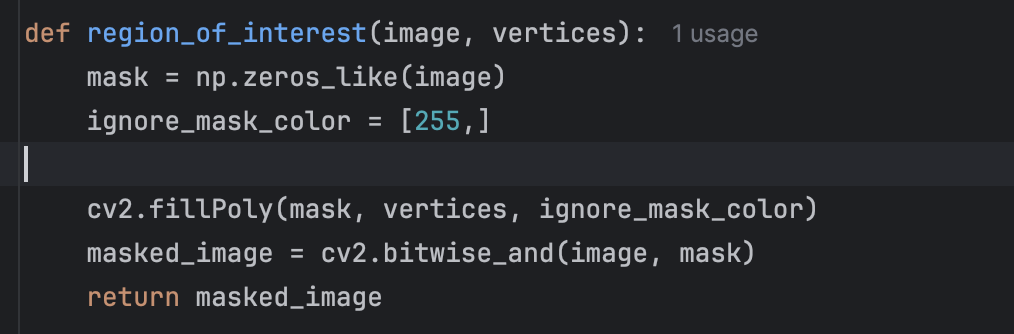


图7:掩码矩阵生成

## 3.4Hough变换

Hough变换是一种用于检测图像中几何形状的技术，将图像由图像空间变换为参数空间。它最初是由保罗·霍夫（Paul Hough）在1962年提出的，用于在图像中检测直线。

Hough变换的原理是利用图像空间和Hough参数空间的线-点对偶性，把图像空间中的检测问题转换到参数空间中进行。

在OpenCV库中，有标准Hough变换的函数实现，也有统计概率霍夫变换的函数实现。因为统计概率霍夫变换执行效率更高，所以在实现路沿检测中采用统计概率霍夫变换实现。

具体实现代码如下图所示。

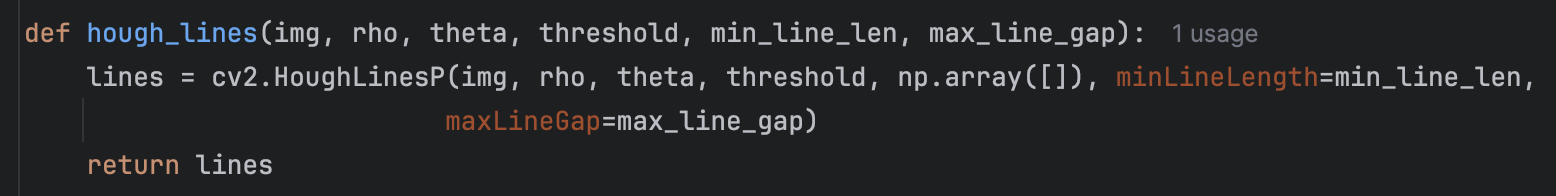


图8:Hough变换

## 3.5绘制直线

经过上述处理，我们可以得到图像中我们感兴趣的区域的直线。但是，Hough变换存在过度检测和过度判断等问题，此外直接绘制Hough变换检测到的直线会发现多条直线相邻平行的情况。

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 过度检测 | 过度判断 | 多条直线 |

图9:问题图像

接下来我将依次解决上述问题。

### 3.5.1过度检测

过度检测即检测直线的条件过于宽松，检测出多余的直线（非路沿直线）。

对于过度检测，我们可以利用空间的结构特征和直线的几何特征进行处理。空间的结构特征即前文提到的掩码矩阵。通过掩码矩阵，我们可以只关注右半侧图像，而不必关注左半侧图像，减少了过度检测的可能性。直线的几何特征这里特指直线的斜率，由给定的测试图像可以看出，路沿的斜率范围大约在2-4之间。因此，我们可以根据斜率筛选直线，再加入到集合中，从而减少过度检测的可能性。

### 3.5.2过度判断

过度判断即检测直线的条件过于严格，没有检测出直线。

对于过度判断，一种简单的办法是使用上一帧图像检测到的直线，因为一帧的变化并不大。此外，我们还可以在不过度检测的前提下，适当地调整参数以检测路沿。

### 3.5.3多直线相邻平行

Hough变换后如果如果直接绘制直线，会出现多条直线相邻平行的情况。这样的结果并不美观，并且我们所需要的是单一直线车道路沿检测。为了直观的体验和后续可能的处理，我们需要对检测到的直线进一步判断：

1. 计算在斜率范围内的直线的斜率平均值。
2. 取在斜率范围内线段端点的最高点作为绘制的起始点。
3. 根据步骤1的斜率平均值和步骤2的起始点计算出绘制的结束点（与图像顶部或右边相交）。
4. 使用OpenCV的库函数绘制直线。

### 3.5.4代码实现

通过上述分析，我们明确了代码实现的思路。具体实现的代码如下图所示。

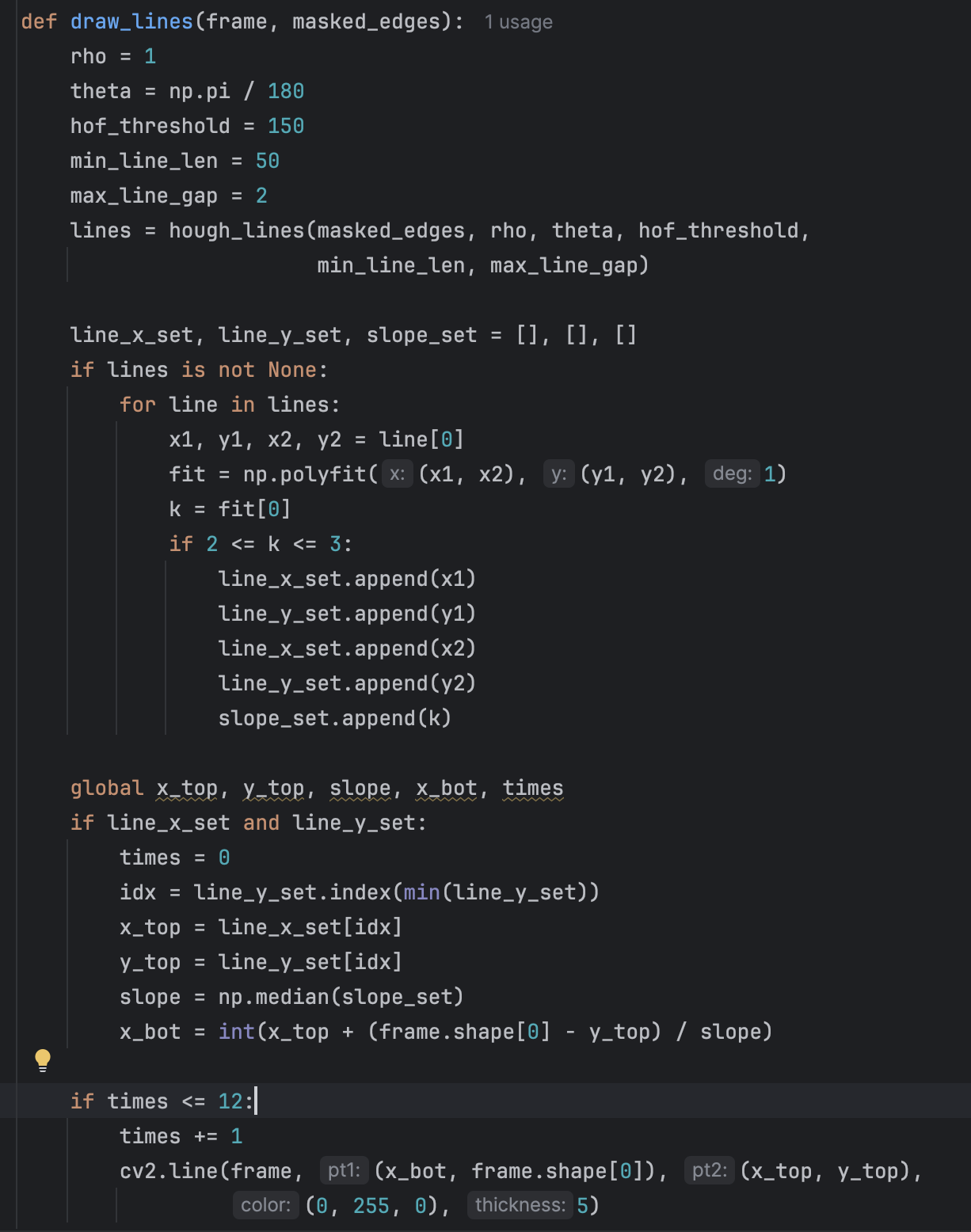


图10:绘制直线

## 3.6结果展示

下面以图片的形式展示路沿检测的结果。路沿检测的视频文件放在压缩包中。

|  |
| --- |
|  |
| 图11a：01.avi路沿检测结果 |
|  |
| 图11b：02.avi路沿检测结果 |
|  |
| 图11c：03.avi路沿检测结果 |

## 3.7 其他工作

### 3.7.1滤波选择

我们可以选择高斯滤波、均值滤波和中值滤波。我们从视频中任意获取获取一帧图像，然后分别使用高斯滤波、均值滤波和中值滤波进行处理，得到的图像结果如图12所示。

由此，我们可以看出：经过均值滤波和中值滤波处理后的图像比较模糊，而经过高斯滤波处理后的图像表现比较好。产生这个问题的原因是算法的实现决定的。

综合上述分析，我们最终选定了高斯滤波作为滤波器。

|  |
| --- |
|  |
| 图12a：高斯滤波 |
|  |
| 图12b:均值滤波 |
|  |
| 图12c：中值滤波 |

### 3.7.2算子选择

Sobel算子的主要优势是计算速度快，在水平和垂直方向上的检测效果比较好，缺点是对噪声敏感并且对其他方向的检测效果比较差。Canny算子的主要优势是检测效果好，错误率和遗漏率比较低，缺点是计算速度比较慢。

综合上述优势和缺点，我最终选择了Canny算子进行边缘检测。

### 3.7.3Hough变换函数选择

在OpenCV库中，存在两个Hough变换函数的实现，分别为HoughLines和HoughLinesP。

前者是标准的Hough变换实现，后者是基于概率的Hough变换。因为视频的每一帧都是一张图像，需要计算直线，所以计算量非常大。

综合上述分析，我选择了计算速度更快地HoughLinesP。

# 四、实验结论或体会

## 4.1结论

1. 实验结果显示，通过高斯滤波和Canny算子的边缘提取，结合Hough变换的直线检测，能够有效地从视频中检测出路沿。
2. 通过掩码矩阵的使用，我们成功地排除了图像左侧的干扰，专注于右侧路沿的检测。
3. 通过对直线的斜率进行筛选，我们进一步优化了检测结果，减少了过度检测和过度判断的问题。

## 4.2体会

1. 通过本次实验，我深入理解并掌握了视频和图像处理的基本操作，包括视频的读取、拆分和合成以及图像的相应处理。
2. 通过本次试验，我学会了如何使用OpenCV库中的高斯滤波、Canny算子等来处理图像，以及如何应用Hough变换进行直线检测和如何绘制直线。

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：沈琳琳、周杰  2024年11月 18日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。