目 次

[目 次 I](#_Toc36742997)

[1 道路图像预处理 1](#_Toc36742998)

[1.1 道路图像灰度化 1](#_Toc36742999)

[1.2 感兴趣区域选取 3](#_Toc36743000)

[2 基于分块投影变换的快速直道行道线检测 4](#_Toc36743001)

[2.1 基于分块投影变换的行道线候选点提取 4](#_Toc36743002)

[2.1.1 行道线特征及候选点提取思想 5](#_Toc36743003)

[2.1.2 优化算法-积分图快速实现投影变换 6](#_Toc36743004)

[2.1.3 候选特征点提取 8](#_Toc36743005)

[2.2 Hough变换行道线定位 11](#_Toc36743006)

[2.2.1 粗分辨率到精分辨率霍夫变换 11](#_Toc36743007)

[2.2.2 查找表优化 12](#_Toc36743008)

[2.3 候选行道线过滤 12](#_Toc36743009)

[2.4 实验结果与分析 15](#_Toc36743010)

[3 基于分段霍夫变换的行道线弯道检测 16](#_Toc36743011)

[3.1 基于分段直线检测的直道弯道判别 17](#_Toc36743012)

[3.2 弯道曲线拟合 19](#_Toc36743013)

[4 偏航报警 21](#_Toc36743014)

[4.1 偏航报警功能设计 22](#_Toc36743015)

[4.2 功能实现 22](#_Toc36743016)

[5 系统实现及测试 24](#_Toc36743017)

[5.1 实验结果分析与改进 24](#_Toc36743018)

1. 道路图像预处理

图像预处理是图像处理领域中常见又重要的步骤，好的预处理能够减少无用信息、增强有效信息，极大地提升之后的图像处理算法的实时性和准确性。根据不同的任务，往往有不同的预处理方法，就本次课题来说，主要需要的是图像灰度化以及感兴趣区域的选取。

* 1. 道路图像灰度化

一般来说，车载相机得到的是RGB 真彩色图像，包含三个颜色通道的信息。虽然有一些学者在行道线检测研究中利用了车道线的颜色信息，但如果全局直接对彩色图像进行处理需要花费大量的运行时间、占用较大的存储空间，不满足工程的实时性要求。考虑到将要使用的行道线检测算法，本文将彩色图像灰度化后进行处理。

图像灰度化即将三个通道颜色通道的信息映射到一个通道中，概括起来通用计算公式如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

其中*x,y,z* 分别是红、绿、蓝通道的权值，三个通道经过加权求和得到灰度值，不同的图像灰度化实际上就是选取不同的权重组合。由于人眼对红绿蓝三种光谱的光敏感度不同，要得到视觉上较为真实的灰度图像，通常取权重如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

车载相机来内部一般采用Bayer 格式表示图片，其中blue 分量和red分量占比较少，且red 分量容易受到近红外的影响，而green 分量占比最大，是blue和red分量的和，包含信息最多[8]， 因此本文直接使用green 分量作为灰度化图像，即公式如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

通过对不同天气情况的道路图像进行灰度化实验结果如下列图所示，可以看到，多种不同天气情况下，绿通道灰度图像表现的更为稳定。

|  |  |
| --- | --- |
| （aE:\实验结果图\灰度化\image000000.bmp）原图 | （b）加权平均E:\实验结果图\灰度化\gry000000加权平均.bmp |

|  |  |  |
| --- | --- | --- |
| （c）红通道E:\实验结果图\灰度化\gry000000红通道.bmp | （d）蓝通道E:\实验结果图\灰度化\gry000000蓝通道.bmp | （e）绿通道E:\实验结果图\灰度化\gry000000绿通道.bmp |

* 1. 晴天道路图像灰度化

|  |  |
| --- | --- |
| （a）原图E:\实验结果图\灰度化\image000001.bmp | （b）加权平均E:\实验结果图\灰度化\gry000001加权平均.bmp |

|  |  |  |
| --- | --- | --- |
| （c）红通道E:\实验结果图\灰度化\gry000001红通道.bmp | （d）蓝通道E:\实验结果图\灰度化\gry000001蓝通道.bmp | （e）绿通道E:\实验结果图\灰度化\gry000001绿通道.bmp |

* 1. 雨雪天道路图像灰度化

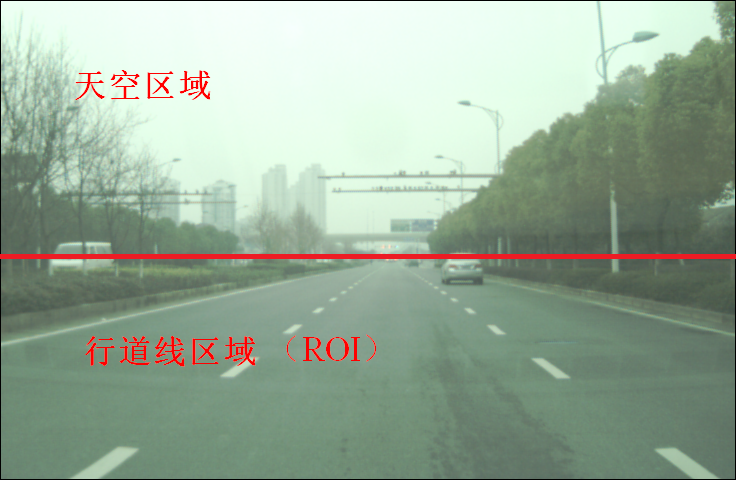
|  |  |
| --- | --- |
| （a）原图E:\实验结果图\灰度化\IMG000002.bmp | （b）加权平均E:\实验结果图\灰度化\gryIMG000000加权平均.bmp |

|  |  |  |
| --- | --- | --- |
| （c）红通道E:\实验结果图\灰度化\gryIMG000000红通道.bmp | （d）蓝通道E:\实验结果图\灰度化\gryIMG000000蓝通道.bmp | （e）绿通道E:\实验结果图\灰度化\gryIMG000000绿通道.bmp |

* 1. 阴影遮挡图像灰度化
  2. 感兴趣区域选取

感兴趣区域简称为ROI （region of interest），即输入图像中拥有有效信息的部分区域，若算法直接对整幅图像处理，则会浪费时间在无效信息区域上，不利于系统的实时性。

观察车载相机拍摄到的图像，可以发现车道线集中在图像的下部分区域，如图 2.4所示，上半部分几乎完全设计不到车道线检测，在图像处理之前应该排除掉，不然会极大影响算法效率。



* 1. ROI示意图

ROI 的选择应根据车载相机的架设角度和位置进行，排除掉非道路区域。本课题程序提供用户接口，用户可根据实际情况提供感兴趣区域位置。

后文中所有的处理皆在ROI区域中进行。

1. 基于分块投影变换的快速直道行道线检测

本章节介绍了一种快速行道线检测方法，并提供了几种优化算法。该方法首先使用分块的投影变化，利用行道线模型特性得到行道线候选点，然后通过Hough 变换定位候选行道线，最后根据车道模型约束进行候选行道线过滤，得到最终结果。本文借鉴和提出了几种优化算法，应用到程序实现中，提高系统实时性。

* 1. 基于分块投影变换的行道线候选点提取

若输入图像的分辨率很高，在原图像上直接检测行道线会相当困难且低效。一般图像处理检测行道线会对行道线进行特征点提取。所谓特征点，就是能够抽象表达行道线的行道线上的点。将整幅图像的像素信息提炼成特征点后能够大大加快运算速度。

边缘检测法是很常见的特征点提取算法，一般算法采用Canny算子、沈俊算子等边缘检测算子检测车道线的边缘特征从而得到特征点。该方法在不良的道路环境下展现不了足够的鲁棒性，如行道线磨损、阴影遮挡、路面积水等边缘特征被破坏的情况。而本文使用的方法由于投影这一加法性质的变换，利用了更多点的灰度信息和行道线特征，将更加具有鲁棒性，同时使用了积分图使得投影变换快速实现，提高算法效率。

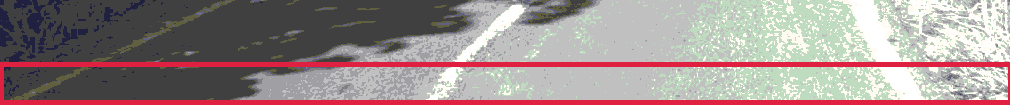
* + 1. 行道线特征及候选点提取思想

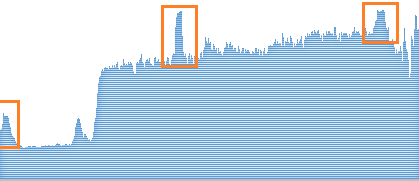
如图3.1 观察行道线图像可以发现，无论是否有阴影遮挡或其他影响，行道线区域的像素亮度总是高于周围路面亮度，且该区域具有一定宽度。



* 1. 道路行道线图像

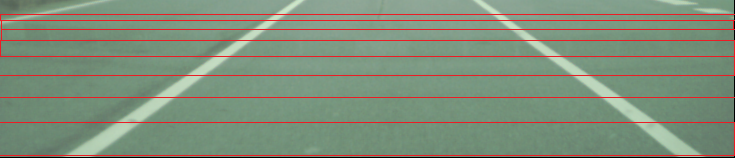
对上图行道线图像中红色矩形区域做垂直投影变换，得到投影直方图，如图3.2 所示。





* 1. 矩形区域垂直投影直方图

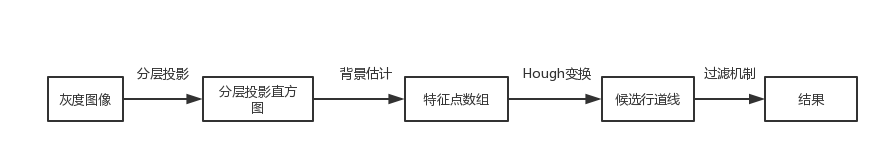
显而易见，无论是有阴影遮挡部分还是正常光照部分，行道线区域都在直方图中形成波峰区域，利用这个特征，再考虑到道路“近大远小”的特点，本文提出如下方案：将行道线图像不重叠地分为高度不等的若干块，由于“近大远小”的视觉现象，每块的高度从下往上依次减小，每块具体高度由车载相机的位置和角度决定，如图 3.3所示。



* 1. 道路图像分块

对每块分别进行垂直投影变换的到各块投影直方图，根据波峰特征进行背景估计得到行道线特征点，根据获得的特征点进行Hough变换得到候选行道线，对候选行道线进行过滤得到最终的结果。

直道检测算法流程图如下：



* 1. 行道线直道检测流程图
     1. 优化算法-积分图快速实现投影变换

根据3.1.1 节中的想法，投影变换是很重要的一个步骤。所谓图像的投影变换，即图像在某一方向上所有像素点灰度值之和。如本文所用垂直投影公式如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

由于投影采用加法运算，体现的是多个像素的信息，因而具有好的鲁棒性。

在本文算法中投影变换将多次用到，若直接在原图中进行多次累加计算，想必是非常费时的。为此我们使用了积分图。

积分图（Integral Image）最早由Paul Viola 等人提出[8]，将全局的信息按照规则组织在一个矩阵中。

一个灰度图像的积分图，其任意坐标(*x*, *y*)上的值为该点坐标与灰度图像坐标原点（一般为左上角）所形成的矩形区域内所有像素点的灰度值之和，如下图中着色区域。

(*x*,*y*)

* 1. 积分图示意

积分图通过遍历一遍图像，将图像从左上角到每个像素点所形成的矩形区域内的像素灰度值之和存储在对应坐标中。步骤如下：

Step1 设原始图像为，积分图为，初始化；水平行方向上像素灰度值累加和为，初始化。

Step2 逐行遍历原始图像，递归计算积分图上每个坐标值，公式如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

Step3 遍历原始图像至右下角像素结束，得到积分图

每当要计算一片矩形区域的灰度和时，只需要知道目标矩形四个顶点的坐标，进行两次减法一次加法即可。如图 3.6中着色区域内所有像素点的灰度值之和计算公式为 。

(*x2*,*y2*)

(*x2*,*y1*)

(*x1*,*y2*)

(*x1*,*y1*)

* 1. 积分图运用

这对3.1.1 中提出的分块投影相当适用，对每个灰度图，仅需扫描一遍，便可获得所有矩形块的投影变换，极大提升了算法运行速度。

* + 1. 候选特征点提取

有了3.1.2 中积分图的使用，很容易得到每层的垂直投影直方图，如图3.7所示。直方图中投影值较大则表示此区域较亮，否则说明该区域较暗。根据行道线的特征，行道线区域较周围路面区域亮，在直方图中成波峰状，本文对波峰进行提取，在波峰中选取点作为特征点。

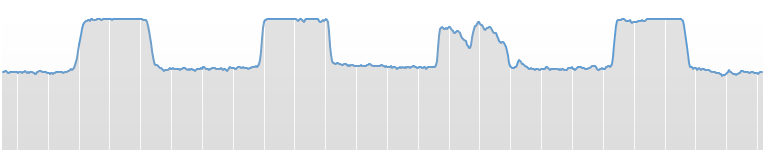
E:\实验结果图\分层投影\Res.bmp

波峰即与周围相比较高的区域，而均值滤波就很好地表示了周围这个概念。均值滤波即用相邻位置的元素的平均值来代替原来位置的元素值。对投影直方图进行均值滤波，均值滤波的窗口半径取决于行道线的宽度。图 3.8为均值滤波之后的投影直方图与原投影直方图对比，系列1为原投影直方图，系列2为均值滤波后的投影直方图。可以发现，原直方图中高于均值滤波后直方图的部分即候选行道线区域，取该部分的中点作为候选特征点，则完成行道线候选特征点的提取。

* 1. 投影折线图

通过行道线的宽度约束，同时能够去除掉一些其他的道路交通标志和噪声。如图 3.9中斑马线，提投影后斑马线区域的波峰宽度明显的大于行道线波峰宽度，依宽度或波峰的高宽比为约束条件便能过滤掉诸如斑马线、转向线、掉头线等等其他交通标志。





* 1. 斑马线及其投影图

将特征点的坐标存储在数组中，作为在之后的Hough变换的输入，这样大大减少了无用的信息，将原始图像高分辨率的像素信息转化为一定数量的特征点信息，使得后续的行道线定位和行道线过滤的计算变得十分迅速。且因为投影变换加法的鲁棒性，特征点的提取可以克服阴影遮挡、路面积水、行道线残缺不齐等不良道路环境，候选特征点提取实验结果如下图所示：

|  |  |
| --- | --- |
| E:\实验结果图\特征点\image.bmp（a） | E:\实验结果图\特征点\Res.bmp（e） |
| （b）E:\实验结果图\特征点\Image000000.bmp | E:\实验结果图\特征点\Res000000.bmp（f） |
| （c）E:\实验结果图\特征点\Image000005.bmp | （g）E:\实验结果图\特征点\Res000005.bmp |
| （d）E:\实验结果图\特征点\IMG000000.bmp | （h）E:\实验结果图\特征点\ResIMG000000.bmp |

* 1. 不同情况下的行道线特征点提取

图(a)~图(d)分别是晴天、小雪雾天、雨天、阴影遮挡等情况下的原始彩色图像，图(e)~图(h)是其分别对应的候选特征点提取结果。可以看到，本文方法对于雨雪天气路面的积水反光、阴影遮挡、行道线残缺不齐等情况都能较好的提取行道线特征点，有着不错的鲁棒性。

* 1. Hough变换行道线定位

霍夫（Hough）变换是一种图像处理中常用的直线检测算法，它有很好的容错性和鲁棒性。Hough空间的原理是构造图像空间和Hough 空间的点坐标到线参数的映射，将原图像空间中的问题转换到参数空间中进行求解。传统的直线检测霍夫变换将图像空间中笛卡尔坐标系中的点（*x*, *y*）转到极坐标系直线的（*ρ*，*θ*）中，对参数空间（*ρ*，*θ*）进行投票，获票超过阈值的（*ρ’*，*θ’*）则为候选直线。阈值选取与车载相机架设位置和与路面夹角有关。

* + 1. 粗分辨率到精分辨率霍夫变换

命令*n*为特征点数量，*count*为计数数组，霍夫变换的伪代码如下：

for θ = θ1; θ< θ2; θ+=stepθ

for i = 1 ; i < n ; i++

ρ = Xi × cosθ + Yi × sinθ;

count[ρ][ θ]++;

end

end

为了加快运算速度，可以先进行粗精度的霍夫变换，实现起来即先用较大的步长*stepθ*和*stepρ*进行迭代，得到候选直线后，再用较小的步长在候选直线附近小角度范围内中进行第二次霍夫变换，得到更加精细的直线参数。

* + 1. 查找表优化

查找表本是图像处理系统中常用的一种数据结构，目的在于加速图像点运算，所谓点运算，就是像素点的计算结果仅与该点的灰度值有关，由于图像像素点数量巨大，而灰度值较小，则构建一个灰度值映射表，之后不需要对图像中的每个像素点进行运算，根据像素灰度值查表得到结果即可。

由于涉及到的并不是点运算，所以灰度值查找表在本文中并不适用，但其空间换时间的思想值得利用。在霍夫变换中，大量地涉及到三角函数运算，若每个特征点都调用三角函数，无疑是非常费时的，因此本文构建了正玄函数和余玄函数查找表，将获取三角函数的时间复杂度变为O(1)，大大加快运算效率。构建查找表过程中，在一定精度下，甚至避免了耗时的浮点数运算。

例如，在构建正弦函数查找表时的伪代码如下：

for i = θ1 ; i < θ2 ; i++

sinLUT[i] = sin(i) × 2*x*;

end

在使用到正弦函数查找表时，伪代码如下：

A = B × sinLUT[θ] >> *x*;

即在构建查找表时乘上2的*x*次以保证运算精度，在使用查找表时通过移位运算代替除法运算来恢复偏差。此过程中，会丢失一定的运算精度，可根据精度要求来改变*x*的取值。

* 1. 候选行道线过滤

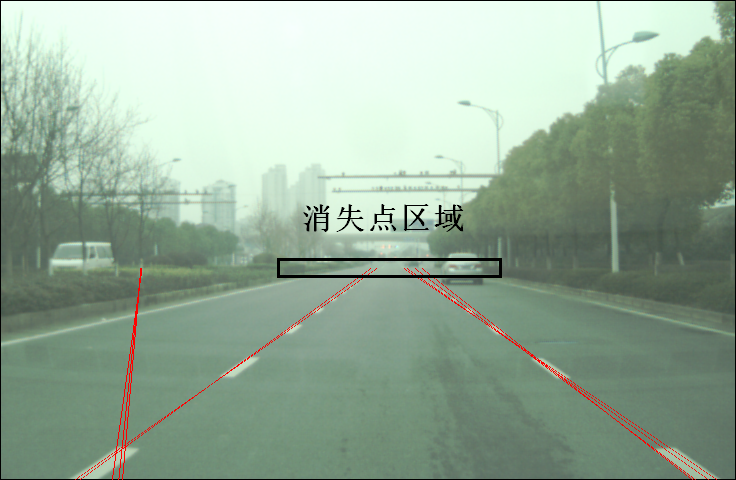
经过上文中的方法得到很多候选直线，如图 3.11。其中有很多不是真正的行道线，需要对候选的直线进行过滤。

* 1. 霍夫变换得到候选直线

首先根据道路模型，车载相机获取的道路图像中，行道线总是有一定的角度范围，若不在适当的角度范围中则舍去，如图 3.12。

* 1. 过滤掉角度不合理的候选直线

其次，车载相机获得的道路图像，由于“近大远小”的视觉原理，车道线会在远端某处汇聚，通常由于车载相机的安装位置和俯仰角固定，消失点的位置将限制在一定范围内。因此，本文在图像中依据经验设定了消失点区域，即下图中黑色标注区域，若不经过该区域的直线则舍去。在实际应用到车辆上时，消失点区域与相机标定有关。



* 1. 通过消失点区域过滤候选直线

经过上面两次过滤后剩下的候选直线基本上为真实行道线，但由于角度轻微的偏差，出现相近的直线重叠问题。为解决这一问题，本文改进了霍夫变换。步骤如下：

Step1 完成霍夫空间投票；

Step2 遍历（*ρ*，*θ*），步长都为1，若（*ρ’*，*θ’*）得票数超过阈值则转Step3；

Step3 检测（*ρ’*，*θ’*）相近的直线，即遍历ρ范围为[*ρ*’+Δ*ρ* , *ρ’*-Δ*ρ*] , θ范围为[*θ’,* *θ’*+Δ*θ*] 的参数空间中的得票数，取其中得票数最大的（*ρ’’*，*θ’’*）作为目标直线记录下来，*θ*=*θ*+Δ*θ*，转至Step2继续遍历；

Step4 霍夫空间遍历完成。

这样，相邻近似的多条直线则选取其中经过特征点最多的直线为目标行道线，其余直线舍弃。如图 3.14。实际上这一过程在3.2小节Hough变换中同步完成。

* 1. 改进霍夫变换去除近似重叠直线
  2. 实验结果与分析

对几种典型的道路环境图像进行直道检测实验，得到如下结果：

|  |  |
| --- | --- |
| （a）晴天E:\实验结果图\Hough\Rescheimage.bmp | （b）车辆遮挡G:\ADAS_2_Student_Res\binkangwest\image000614.bmp |
| （c）雨天E:\实验结果图\Hough\Resimage000006.bmp | （d）小雪加雾且行道线残缺E:\实验结果图\Hough\Resimage000000.bmp |
| （e）阴影E:\实验结果图\Hough\ResIMG000003.bmp | （f）阴影E:\实验结果图\Hough\ResIMG000000.bmp |

* 1. 不同道路情况下行道线直道检测

从结果可以发现，本文的方法对于车辆遮挡、雨雪天路面积水、雾天视线受阻、行道线残缺、阴影遮挡等情况都有着不错的鲁棒性，算法实时性和准确性将在第6.2节中进行分析。

1. 基于分段霍夫变换的行道线弯道检测

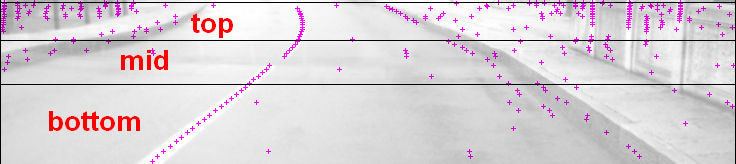
上一章介绍了快速行道线直道检测算法，直道检测在一定程度上已经能够实现偏航报警功能，因为在判别车辆是否跨线行驶或压线行驶时，往往是用车头到往前不远的一段行道线作为依据的，而这一段短距离的行道线通常可以近似为直线。

但本文重点在于行道线检测，偏航报警仅是基于行道线检测能够实现的功能之一，一个有效的行道线检测系统不希望仅限于此，要能够一定程度地适用于辅助驾驶系统甚至无人驾驶系统。直道检测不能应对所有的道路情况，因此弯道检测成为了必不可少的一步，弯道检测对于智能驾驶车辆的规划有关键的作用。本章提出了一种基于分段直线检测的行道线弯道检测算法。

* 1. 基于分段直线检测的直道弯道判别

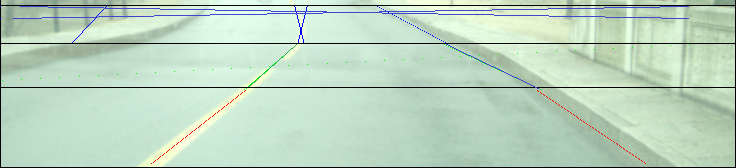
根据微积分的思想，弯道曲线可近似看成若干直线段拼接而成。本文将道路图像在竖直方向上分段，分段数和每段高度根据相机的架设位置和与路面夹角而异。本文根据实验数据集，将图像分为上中下三段。再对每一段分别作第3章中的直线检测，对直线段进行拼接得到折线，根据折线进行弯道直道判别。具体步骤如下：

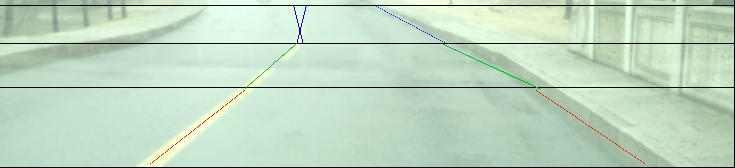
Step1 对获得特征点后的道路图像感兴趣区域进行分段，本文分为上中下三段。根据3.2小节，由于“近大远小”的视觉现象，分块投影获得特征点时，图像底部的分块高度较高，分块数较少，提取的特征点在竖直方向上较为稀疏，而顶部特征点较为密集，在分段时保证每一段中的竖直方向上特征点数量基本相等。如图 4.1，每段中的行道线可近似看作直线段。



* 1. ROI区域分段

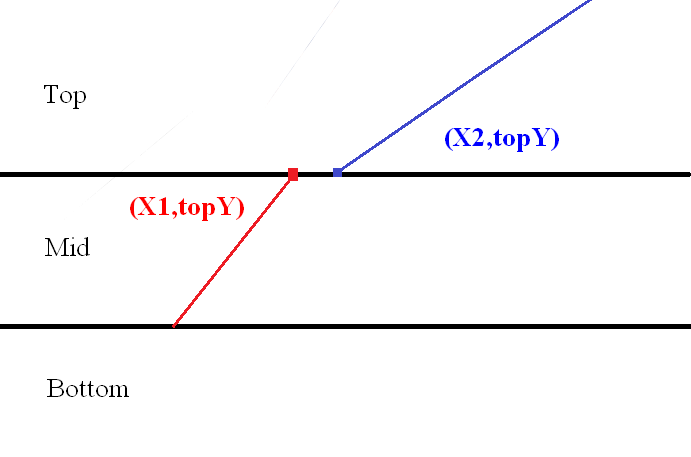
Step2 对每段分别进行第3章中的快速行道线直道检测操作，包括不良行道线、虚假行道线过滤。详细见第3章。





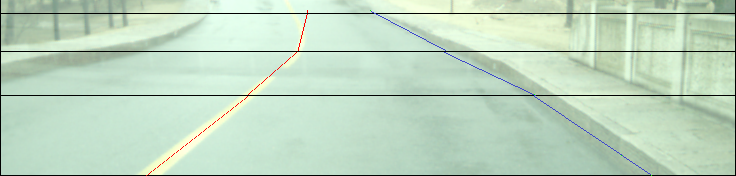
* 1. 分段直线检测和除杂

Step3 对每段中的直线线段进行首尾拼接。通过图 4.2可以观察到一条弯道行道线被分成了三个直线段。要做曲线拟合操作，首先要判别那些直线段属于同一条行道线。属于同一条行道线的直线段首位间空间位置相近，根据此特性，计算相邻两段中行道线与分段直线的交点，若欧式距离小于阈值，则判定为同一行道线上的线段，将其拼接起来。如图 4.1所示，顶部段的直线段与段分界线*y*=*topY*的交点横坐标为*X2*，而中部段的直线段与分界线*y*=*topY*的交点横坐标为*X1*，则计算，若*dist*小于设定的阈值（一般为行道线宽度），则认为两条直线段可以拼接为一条折线。



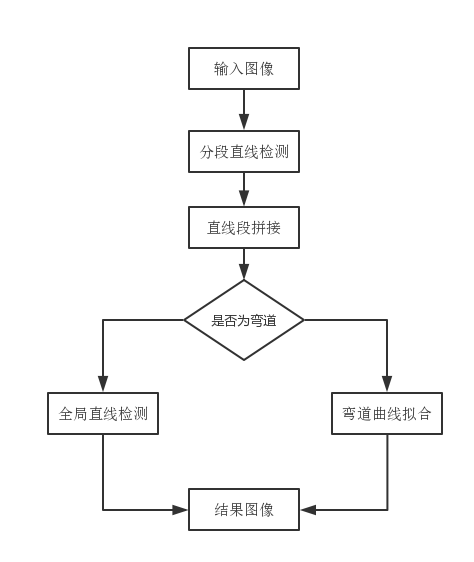
* 1. 拼接示意

考虑到行道线为虚线的情况，分段后底部段图像可能无法检测到行道线，所以从中间段的直线段开始遍历，先后计算与顶部段、底部段内直线在分段线上的交点距离，若小于阈值则拼接在一起，由此得到若干折线，如图 4.2。

* 1. 直线段拼接成折线

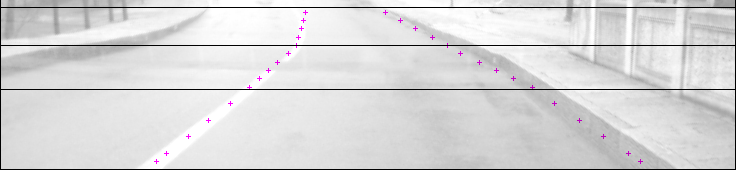
Step4 若折线上的所有直线段角度接近，则认为是直道，直接进行第3章直道行道线检测操作，否则认为是弯道，进行后文中提出的曲线拟合。

流程图如下：



* 1. 行道线检测流程图
  2. 弯道曲线拟合

上一节中已经获得了位于行道线上的折线，本小节将讨论如何将折线拟合成弯道曲线。首先要获得待拟合点，在上一节得到的折线中，在竖直方向上每隔一定间隔取一个点作为特征点，图像顶部取较小间隔，底部取较大间隔，使得每个段中的待拟合点数量相等，如图所示：

* 1. 拟合点选取

得到拟合点后则进行曲线拟合算法，本文采用多项式最小二乘法进行拟合。

所谓曲线拟合，即对于给定的n 个样本点，求得一条曲线，使得曲线尽量逼近所有样本点。如何判断逼近的量化指标之一是偏差，计算公式如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

最小二乘法曲线拟合方法即以偏差的平方和最小为原则得到曲线参数，即：

|  |  |  |
| --- | --- | --- |
|  |  |  |

进行拟合之前需要选择的函数类型，一般常用的类型有多项式、三角函数、指数函数等，根据行道线弯道的特点，本文选择多项式函数类，即：

|  |  |  |
| --- | --- | --- |
|  |  |  |

其中 为待定系数，拟合的过程即求取这些待定系数，使得偏差的平方和最小。最小二乘法是具有唯一最优解的问题，利用导数极值法求解，可得一个唯一的使得式成立 。推到过程如下：

Step1. 样本点与目标曲线偏差平方和如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

Step2. 对求导，*i*=0,1,…n-1 。得到：

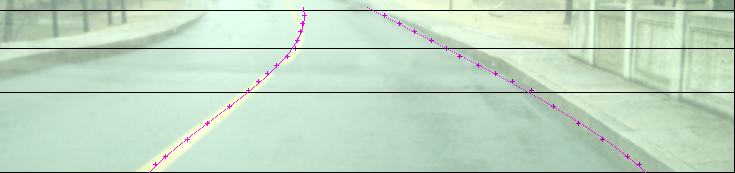
|  |  |  |
| --- | --- | --- |
|  |  |  |

Step3. 将上面的方程组写成矩阵形式并化简：

|  |  |  |
| --- | --- | --- |
|  |  |  |

Step4. 高斯消元法解方程，求解参数向量，便得到目标函数的参数。

多项式的次数m越高，往往越能拟合出更加复杂的曲线函数，但是耗费的时间代价也会因此大大增加。对于车载相机获得的道路图像来说，弯道行道线一般不会出现过于复杂的情况，一般来说2次或3次多项式就能很好的拟合大多数弯道行道线，本文采用3次多项式进行拟合，效果如下：

* 1. 曲线拟合效果

1. 偏航报警

行道线检测是智能驾驶系统中很重要的一环，它不仅能作为智能辅助驾驶系统工作，也能用在无人驾驶系统中。偏航报警基于行道线检测实现，是行道线检测的实际应用之一。在现今发生的各种事故中，无意识的车道偏离是危害人们生命的首要原因[14],偏航报警功能有着重要的实际意义。本章介绍偏航报警系统的设计思想和应注意的问题，并讲述了本文提出的简单功能实现方法。

* 1. 偏航报警功能设计

偏航报警系统不断监视着车辆相对于行道线的位置，在车辆偏离车道时警告驾驶员，预防危险情况的发生。危险的车道偏离情况有以下几种：

1. 车辆较长时间压着行道线行驶；
2. 车辆较长时间跨线行驶；
3. 车辆在未打指示灯的情况下，快速地接近行道线；

这些情况不仅需要正确地检测行道线，系统还需要获取车辆的速度、加速度、航向角等一系列数据，需要使用到惯导传感器。

另外，还需考虑到驾驶员的用户体验。在大多数情况下，驾驶员的注意力是集中的，能够自主的发现车辆的偏航，如果报警系统过于敏感，或者虚假报警率太高，使驾驶员频繁地收到警报，将会很是打击用户体验。因此，是否发出警报的判别应该以低虚警率为原则。

* 1. 功能实现

如上一个小节中提到的，车辆内置的辅助驾驶系统或无人驾驶系统中的偏航报警功能，通过惯导传感器获得车辆行驶时速度、航向角等一系列数据，根据行道线检测的结果，判断偏航与否。实际上，行道线检测是偏航报警的基础和关键，行道线检测功能完善和准确，偏航报警是很容易添加的功能，但不同的车辆、不同的车载相机获得的道路图像参数不一，偏航报警需要根据具体的车辆和车载相机进行设计。本文的目的在于行道线检测的实时性、鲁棒性和通用性上，希望能够成为便于后安装的实用系统，因此对于本文获取的实验数据集进行简化的偏航报警功能实现。

本文根据实验数据集，认为行道线在水平方向上处于图像中间位置，如图 5.1中绿色矩形区域，则认为属于跨线或压线行驶。

****

* 1. 假定跨线区域

考虑到低虚警率的原则，本文当连续4帧以上的图像处于跨线压线行驶时，触发警报。当然，在实际应用到车辆中时，这一判别依据应与车辆速度、车载相机每秒获取图像帧数有关，需要获得这些数据的输入。

1. 系统实现及测试
   1. 实验结果分析与改进

本文的测试数据集有四个，列出了个数据集来源以及其特点。图 6.1给出了各数据集的例图。

* 1. 各数据集来源和特点

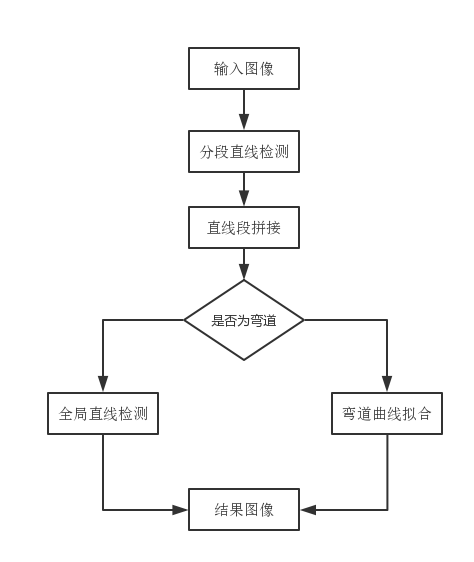
|  |  |  |
| --- | --- | --- |
| 数据集 | 来源 | 特点 |
| Dataset1 | IKTTTI | 阴影遮挡、形势多样 |
| Dataset2 | 车载相机 | 晴天、结构化道路 |
| Dataset3 | 车载相机 | 雨天、行道线残缺 |
| Dataset4 | 车载相机 | 小雪天、雾天、路面积水 |

|  |
| --- |
| G:\ADAS_2_Student_SourceImage\ROMA\IMG000085.bmpG:\ADAS_2_Student_SourceImage\ROMA\IMG000033.bmpG:\ADAS_2_Student_SourceImage\ROMA\IMG000058.bmp  （a）Dataset1例图 |
| G:\ADAS_2_Student_SourceImage\binkangwest\image000394.bmpG:\ADAS_2_Student_SourceImage\binkangwest\image002129.bmpG:\ADAS_2_Student_SourceImage\binkangwest\image000000.bmp  （b）Dataset2例图 |
| G:\ADAS_2_Student_SourceImage\湘湖_小雨\image000028.bmpG:\ADAS_2_Student_SourceImage\湘湖_小雨\image000147.bmpG:\ADAS_2_Student_SourceImage\湘湖_小雨\image000417.bmp  （c）Dataset2例图 |
| G:\ADAS_2_Student_SourceImage\小雪_1\image000000.bmpG:\ADAS_2_Student_SourceImage\小雪_1\image000446.bmpG:\ADAS_2_Student_SourceImage\小雪_1\image000944.bmp  （d）Dataset3例图 |

* 1. 各数据集例图

Dataset1选自KITTI数据集[20]，KITTI是目前自动驾驶领域比较著名和有代表性的数据集，通常用于计算机视觉的环境感知算法评估。Dataset1选取了KITTI中[ROMA (ROad MArkings)](http://www.lcpc.fr/english/products/image-databases/article/roma-road-markings-1817)的部分图像，该数据集主要用于路标记提取算法的性能评估，其中的道路各种各样，如图 6.1（a）高亮度光照、（b）中行道线残缺不齐、（c）弯道曲率大，以及3.4小节中图 3.15（f）中阴影遮挡。

Dataset2、Dataset3、Dataset4为车载相机所拍摄的各类行道线图像。Dataset2是在天气良好、光照条件不错的情况下拍摄的结构化道路连续图像集，如图 6.1（b）所示，行道线完整清晰。。Dataset3是在雨天拍摄的道路连续图像集，如图 6.1（c）所示，路面有积水，行道线以弯道、虚线为主。Dataset4是在小雪天拍摄的道路连续图像集，如图 6.1（d）所示，路面积水严重，有雾，可见度不高，部分行道线残缺不齐。



* 1. 实验流程

实验平台和编译环境在6.1小节中已有介绍，本文对上文中的数据集进行测试，同时包括直道、弯道检测。算法流程如图 6.2所示。

|  |
| --- |
| G:\ADAS_2_Student_Res_测\ROMA1\IMG000085.bmpG:\ADAS_2_Student_Res_测\ROMA1\IMG000033.bmpG:\ADAS_2_Student_Res_测\ROMA1\IMG000058.bmp  （a）Dataset1结果图 |
| G:\ADAS_2_Student_Res_测\binkangwest\resImage000402.bmpE:\实验结果图\Hough\image002129.bmpG:\ADAS_2_Student_Res_测\binkangwest\resImage000000.bmp  （b）Dataset2例图 |
| G:\ADAS_2_Student_Res_测\湘湖_小雨\resImage000028.bmpG:\ADAS_2_Student_Res_测\湘湖_小雨\resImage000147.bmpG:\ADAS_2_Student_Res_测\湘湖_小雨\resImage000417.bmp  （c）Datase3例图 |
| G:\ADAS_2_Student_Res_测\小雪_1\resImage000000.bmpG:\ADAS_2_Student_Res_测\小雪_1\resImage000446.bmpE:\实验结果图\Hough\resImage000994.bmp  （d）Datase4例图 |

* 1. 各数据集测试结果图例
  2. 数据集测试统计结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据集 | 图像帧数 | 图像大小 | 误判率（FPR） | 漏检率（FNR） | 平均处理时间 |
| Dataset1 | 95 | 1280×1024 | 3% | 8% | 9.8ms |
| Dataset2 | 4689 | 736×480 | 5% | 3% | 2.3ms |
| Dataset3 | 2000 | 736×480 | 13% | 17% | 2.1ms |
| Dataset4 | 3600 | 736×480 | 10% | 15% | 2.5ms |

图 6.3是各数据集的测试结果图例，表 6.2展示各数据集上的抽样统计结果。

可以看到，在Dataset1中，本文方法能够对强光照、阴影遮挡、行道线残缺等不良环境有着不错的鲁棒性。

对于Dataset2中的结构化的城市道路来说，行道线清晰可见，本文方法有着非常不错的性能。

对于Dataset3中雨天路面，且多虚线行道线的情况，容易出现误检或漏检。因为虚线作为行道线没有很好的连续性，在本文第4章提到的分段直线检测中，很可能出现某一图像段是行道线虚线空缺的情况，如图 6.4（a），Mid段中无法检测到行道线，解决方法可以是将第4章中的分段算法和直线段拼接改进，调整每段的高度，这与车载相机的安装以及行道线的国标有关。另外，如图6.4（b）由于路面积水或其他污渍，容易出现误检，对图6.4（b）中蓝色矩形框部分进行投影变换，得到直方图，分析该直方图可以发现，进行本文第3章中分块投影的操作后，由于积水等污渍的影响，非行道线区域也会形成波峰，且刚好符合宽度约束，从而出现误检，图6.4（c）中雨刷器同理。目前能够思考到的解决方案是通过颜色特征加以过滤。通常行道线标记的颜色为白色或黄色，检测候选直线区域的彩色通道信息可以进行过滤，但是如何判别像素彩色通道值是否为白色或黄色需要设置阈值，要面对不同的路面情况，阈值的选取是很难的，且对于彩色图像的操作是比较费时的。在灰度图像中，行道线区域的颜色信息体现为灰度值大小，可以通过计算候选直线和直线两边的对比度来判别，判别依据与图像道路区域亮度有关。

Dataset4中路面积水非常严重，部分道路行道线残缺，本文方法对其而言有着不错的鲁棒性，但是由于路面积水反光，容易将白色路灯杆的倒影误检测成行道线，如图 6.4（d），无论是角度、宽度、还是颜色特征，灯杆倒影都是非常接近行道线的，有待研究更好的行道线过滤机制。

|  |
| --- |
| E:\实验结果图\错误\a特征点aa.bmp  （a）  E:\实验结果图\直方图类似\resA.bmp  E:\实验结果图\直方图类似\YGR2ZIVXD%4`9_5NZFYG72D.png  （b）  G:\ADAS_2_Student_Res_测\湘湖_小雨\resImage000198.bmp  （c） |
| G:\ADAS_2_Student_Res_测\小雪_1\resImage000603.bmp  （d） |

* 1. 不良结果分析

所有测试数据集都满足实时性的要求，还有很充裕的时间复杂度可以应用到今后的进一步完善中去，例如，加入连续图像帧之间的关系，根据上一帧的行道线检测结果来约束下一帧的检测范围，提高准确性和效率；增加多种后验算法，对各种误检的行道线进行筛选。

另外考虑到第5章提出的偏航报警设计原则和思路，再依据本文测试数据集中车辆的平均行驶速度，本文监视连续4帧中的检测结果，若连续4帧都处于“压线跨线”情况，则触发警报，对于Dataset1、2、3连续图像集，其误报率能够做到低于5%，达到了低虚警率的要求。