

DOI: 10.19344/j.cnki.issn1671-5276.2019.06.040

# 基于动态划分感兴趣区域的车道线检测算法

谢昌刚,黎向锋,李堃,杨振泰,毕高杰

(南京航空航天大学 机电学院,江苏 南京 210016)

**摘要:** 车道线检测一直是无人驾驶系统的研究热点之一。随着路况复杂程度的增加,对车道线检测效率和检测精度也不断提出更高的要求。提出一种基于动态划分感兴趣区域的车道线隔行检测算法。通过动态划分感兴趣区域将天空、当前车道两旁干扰因素等有效去除,减少了80%左右的检测量,同时也很大程度上避免了误检。在边缘点提取算法中,将检测区域划分成几个小区域,分别提取其边缘点,并采用隔行方法对其检测,最后再将提取到的这些边缘点运用最小二乘法和三次样条曲线拟合出车道线,有效提升了车道线检测算法的实时性和准确性,并且适用于多种试验环境(如雾天、雨天和夜间)。

**关键词:** 车道线;动态;感兴趣区域;隔行;拟合

**中图分类号:** TP274 **文献标志码:** A **文章编号:** 1671-5276(2019)06-0150-05

## Lane Line Detection Algorithm Based on Dynamic Partition of Interested Region

XIE Changgang, LI Xiangfeng, LI Kun, YANG Zhentai, BI Gaojie

(School of Mechanical &amp; Electrical Engineering, Nanjing University of Aeronautics &amp; Astronautics, Nanjing 210016, china)

**Abstract:** Lane line detection is always one of the research hotspots of unmanned driving system. With the increase of the complexity of road conditions, higher requirements are put forward for the efficiency and precision of the lane line detection. This paper proposes an interlaced detection algorithm based on dynamic partition of interested region. By going through the area of interest, the interference factors in the sky and on both sides of the current lane are effectively removed, so as to reduce the detection amount by about 80% and largely avoid the wrong detection. In edge points extraction algorithm, this paper divides the detection area into the several small areas for extracting the edge points, and uses the method of interlaced region for the test, and then, extracts the edge points using the least square method and cubic spline curve fitting out of the lane line, thus effectively improving the real-time accuracy of the lane line detection algorithm. It is suitable for a variety of test environment (e.g., fog, rain and night).

**Keywords:** lane line; dynamic; interested area; interlaced; fitting

## 0 引言

车道线检测是无人驾驶的基础,它能为无人车的驾驶提供行车路线以及变道等一系列辅助功能。车道线检测已经成为无人驾驶重要的一个基础环节,目前车道线检测主要有基于模型的方法和基于特征的方法。基于模型的车道线检测算法主要有直线模型<sup>[1]</sup>、二次曲线模型<sup>[2]</sup>、线性双曲线模型<sup>[3]</sup>、B样条模型<sup>[4]</sup>等。这一类基于模型的车道线检测算法首先对原始图的灰度图用Sobel、Canny等算子<sup>[5]</sup>提取车道线边缘,再进行拟合,从而将车道线以数学模型的形式表达出来,大部分较为成熟的研究都是基于这一种模型。在这种模型研究中,孙朋使用Sobel进行边缘检测,使用霍夫变换和最小二乘法进行直线检测与追踪,这种检测算法虽然实时性好,但是对曲线的检测效果不太理想<sup>[6]</sup>; POLLARD E通过局部阈值方法对车道线区域进行提取,并分割出车道线<sup>[7]</sup>;耿静静使用CHEVP算法确

定天空与路面的水平线,使用B-Spline拟合车道线,此算法受光照影响较大<sup>[8]</sup>;王晓锦使用分段直线拟合车道线,但拟合精度不高<sup>[9]</sup>。另外一种检测方法是基于特征的方法,一般先提取车道线颜色纹理等特征,再用分类等方法获取车道线信息。这种方法受光照变化以及非道路区域的影响很大,故在这里不作深入探讨。

通常,车道线检测方法都是基于固定区域进行,本文提出了一种基于动态感兴趣区域的车道线隔行检测算法,大大减少了检测区域,提升了检测效率和准确性,主要改进如下:1)通过天空到路面的图像灰度值变化从而动态去除天空部分,减少检测区域;2)通过动态划分检测区域约束边界从而将车道线两边的干扰因素去除,极大地减少了不必要的检测区域,也减少了大量干扰因素;3)隔行提取车道线边缘点从而减少循环次数,提高检测速度;4)根据车道线特征提出一种检测车道线边缘点的检测算法。本文车道线检测主要是检测当前行车所在车道的车道线,算法整体流程图如图1所示。

**基金项目:** 国家自然科学基金(51575269);江苏省科技项目(BY2016003-12)

**作者简介:** 谢昌刚(1991—),男,湖北咸宁人,硕士研究生,主要研究方向为计算机视觉。

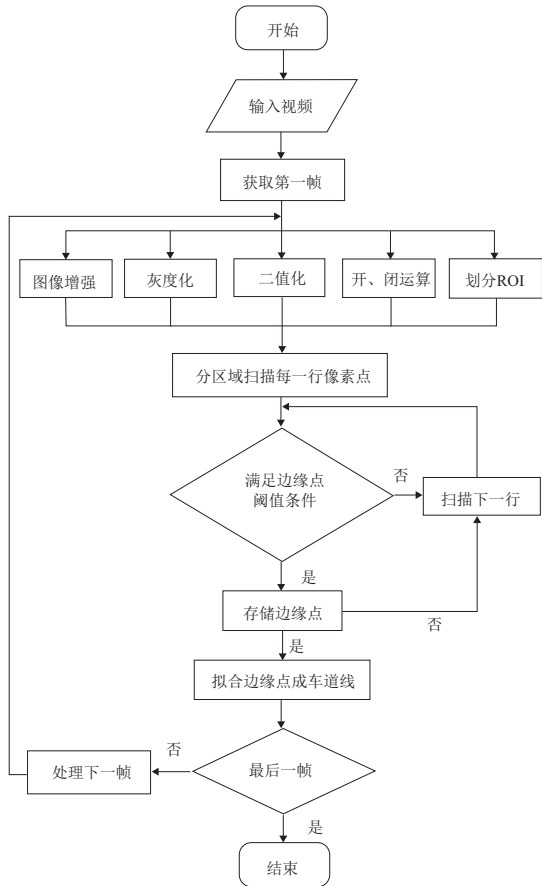


图1 车道线检测整体流程图

## 1 预处理过程

预处理是为了后续的车道线边缘点提取,主要包括增强对比度、二值化、运用形态学滤波、连通域标记等去除噪点、运用动态划分感兴趣区域减少检测区域等步骤。本文针对雾天图像还运用了去雾算法<sup>[10]</sup>。

### 1.1 天空与道路交接处划分感兴趣区域

考虑到天空部分没有车道线,无需检测;故在划分感兴趣区域时将区域划分为天空以下的道路部分。

在将原始图像转化为二值化图像之后,可以发现天空部分是白色的,路面部分是黑色的,故本文采取在二值图上进行划分一次感兴趣区域。从图像的正中间从上往下逐行扫描并获取扫描点的灰度值,当灰度值由 255 突变到 0 且灰度值为 0,持续一定像素点个数(本文取 10),返回此时该像素点的坐标值,在划分感兴趣区域时以此点所在行下面的部分区域为感兴趣区域。

划分感兴趣区域原理用数学表达式如下:

$$\begin{cases} image.at<uchar>(i,j) = 0 \\ image.at<uchar>(i-1,j) = 255 \\ return(i) \end{cases} \quad (1)$$

其中  $i, j$  分别为图像的行坐标和列坐标,此处取  $j = image.cols/2$ ,即图像列坐标的正中间;  $image.at<uchar>(i,j)$  表示点  $(i,j)$  处的灰度值,当灰度值由 255 突变到 0

且灰度值为 0 的像素点连续个数大于设置阈值时,即可以判定由天空变到地面。原图和二值化图分别如图 2 和图 3。



图2 原始视频帧图像



图3 二值化处理后图像

车体部分也不是需要检测的部分,当摄像头固定时,车体部分占图像整体比例也是固定的。可以设定一个参数  $k$ ,需要去除的车体部分为  $k * image.rows$ ,本文中  $k = 0.18$ 。把天空部分和车前部分车体处也不作为检测区域,效果如图 4 所示。



图4 去除天空和车体部分后感兴趣区域

### 1.2 形态学滤波

一般情况下二值化后的图像中还会存在一些干扰因素,比如车道线不一定全部检测出来,不是实心的,很有可能会有一些空洞或者断开的现象,还有的时候会存在一些小点,为了消除这些干扰因素,本文采用形态学滤波的方法。

开运算<sup>[11]</sup>是先腐蚀后膨胀的过程,可以去小物体,在纤细点处分离物体,并且在平滑较大物体时不明显改变其面积。其表达式如下:

$$dst = open(src, element) = dilate(erode(src, element)) \quad (2)$$

其中:  $src$  为输入图像,  $dst$  为输出图像,  $erode()$  为腐蚀函数,  $dilate()$  为膨胀函数,  $open()$  为开运算函数。

开运算之后往往图像会出现车道线中有断开或者小白点的情况,为了消除这些干扰,还需对图像进一步进行闭运算处理。

闭运算<sup>[11]</sup>是先膨胀后腐蚀的过程,能够排除小型黑洞,其数学表达式如下:

$$dst = cclose(src, element) = erode(dilate(src, element)) \quad (3)$$

其中  $cclose()$  为闭运算函数。

原图如图 5 所示,先开运算后闭运算效果如图 6 所

示。



图5 原图



图6 开、闭运算后效果图

由图5-图6可见,开、闭运算后消除了很多干扰因素。

### 1.3 动态划分车道线检测区域

经过初步处理的二值图像仍然含有很多不需要检测的目标,如车道中的车辆、相邻两边车道线、道路两旁的房屋树木等。要想检测的结果准确可靠,去除这些干扰因素也是必不可少的。

有效去除这些非检测目标最直观有效的方法就是将要检测的两条车道线之外的其他非检测目标尽可能多地变成背景,就能极大地减少检测目标的范围。一般情况下只要确定车道线左右外侧边界直线即可,在这里将左右车道线内侧边界直线确定出来,经过这样处理之后基本就没有多少干扰因素了。

为了确定这些边界直线,本文采用从底端中点开始分别向两边寻找白色点再确定斜率 $k$ 的方式,即点斜式的方法来确定内侧边界直线。一般情况下 $k$ 的取值范围<sup>[12]</sup>如下:

$$\begin{cases} 0.5 \leq k \leq 6, \text{左车道线} \\ -6 \leq k \leq 0.5, \text{右车道线} \end{cases} \quad (4)$$

提取左车道线内边界线时,以0.1为单位来改变 $k$ 值,然后在 $k$ 变化过程中,统计直线上有多少白点并记录下来,最后比较白点数目,白点数目最大值对应的 $k$ 值就是要找的 $k$ 值。这样已知了 $k$ 和直线上一点,就可以求出直线方程,再将 $b$ 值加30和减60个像素单位分别作为该车道线的内外边界,右车道线边界线提取算法原理也是类似的。最后将感兴趣区域外部分变成背景色,其结果如图7所示。



图7 感兴趣区域外部分变成背景色

为了进一步减少检测误差,将内侧边界线交点以上部分去除(尤其是夜间图像这一步很重要),以进一步缩小检测区域,其效果如图8所示。



图8 去除交点以上部分检测区域

为了进一步减少左右两边不必要的检测,缩减检测耗时,以左外界的下起点为新的感兴趣区域的下起点,以右外界的下终点为感兴趣区域的下终点,最后得到的检测区域如图9所示。



图9 去除左右部分后检测区域

每一步重新确定检测区域后与原图面积对比均有所减少,如表1所示。

表1 检测区域变化表

图片名称	图片大小	图片大小占原图百分比/(%)
原图	1 280×720	100.00
去除天空图	1 280×370	51.39
去除车体图	1 280×220	30.56
去除车道两旁图	642×198	13.79

从表1中可以看到最终检测区域减少为原图的13.79%,减少了大量不必要的检测,从而使检测速度更快。

## 2 车道线边缘点提取算法

### 2.1 划分检测区域

从图10中可以看出图像下端车道线较宽的情况下所包含的像素点较多,由下到上呈现减少趋势,故可以采用将待检测图像由上向下划分成3个相等区域,每一个区域的大小根据输入图像大小进行参数化,即 $k * image.rows$ ;划分区域是为了接下来分别设置检测阈值的隔行检测做准备。

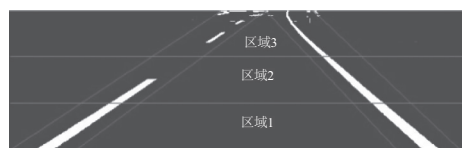


图10 划分检测区域

### 2.2 隔行检测边缘点

对于划分好的3个检测区域,分别有不同的检测条件。检测以内界的交点为横坐标起点,从图像最下端分别向两边进行隔行检测并存储满足各自区域阈值条件的边缘点。

在最下端的区域1,本文采用每隔3行检测边缘点。实际上可以根据输入图像的不同,所隔的行数也可以不同。理论上检测到的点越多越好,但是考虑到检测的耗时问题,故进行隔行检测。在区域1内可以看到车道线的宽度较宽,因此其阈值的设置也较大,以左侧车道线边缘点提取为例,当检测到第一个像素点为白色时:

$$image.at<uchar>(i,j) = 255 \quad (5)$$

记下该点横坐标并继续向左扫描,并记下白色像素点的宽度,即每扫描一次累积加一次,最后得出当灰度值由 255 突变成 0 之前灰度值为 255 的像素点的个数即  $Lwidth$ ,当  $Lwidth$  满足所设置的宽度条件时就将其存储到向量  $vector<Point>\&Lpoints$  中。本区域采用的阈值判别条件是:

$$16 < Lwidth \leq 32 \quad (6)$$

具体算法流程图如下:

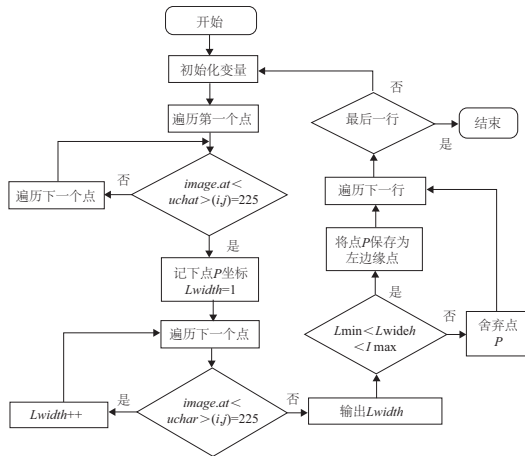


图 11 左车道线边缘点提取算法流程图

对于区域 2、3 也是采用类似的算法进行隔行遍历像素点,只是遍历时车道线的宽度发生了变化,即判断条件发生了变化,区域 2 和 3 分别取  $8 < Lwidth < 16$  和  $4 < Lwidth < 8$ ,最后将这些符合条件的像素点坐标保存下来,为下一步拟合做准备。

### 3 车道线边缘点拟合算法

车道线边缘点虽然已经提取出来,但还要对其进行拟合,以计算出车道线数学模型的参数。本文采用曲线多项式模型来进行拟合,包含最小二乘法和三次样条法。最终将对这两种拟合效果的差异,选择一种效果较好的拟合方法。

#### 3.1 最小二乘法

最小二乘法<sup>[13]</sup>是 Adrien-Marie Legendre 于 1806 年最早独立发现的,它是一种运用领域非常广的拟合算法,它通过用拟合函数和观测值之间的误差平方和来求解。

#### 3.2 三次样条法<sup>[14]</sup>

三次样条法原理:在区间  $[x_1, x_2, \dots, x_n]$  上构造函数  $s(x)$ ,使其满足如下条件:

- 1)  $s(x)$  在任一衔接点  $[x_1, x_2, \dots, x_n]$  上二阶连续且可微;
- 2)  $s(x)$  在任一子区间  $[x_i, x_{i+1}]$  上为三次多项式;
- 3)  $s(x_i) = f(x_i), i = 1, 2, 3, \dots, n$ 。

### 3.3 拟合效果对比

分别用上面两种拟合方法对之前保存的车道线边缘点进行拟合,并最终在原图上显示出拟合后的车道线,效果如图 12-图 13 所示。



图 12 最小二乘法拟合效果



图 13 三次样条曲线拟合效果

可以看出最小二乘法明显比三次样条曲线拟合效果好,因此本文最终采用最小二乘法进行车道线拟合。

## 4 多环境拟合方法效果展示

本文所有程序都是运用 C++ 编写,并且基于 OpenCV 2.4.9 库,是在 VS2012 环境下调试的。所用电脑配置是:

- 1) 处理器:第三代智能英特尔®酷睿™ i5-3337U CPU 1.8 GHz;
- 2) 显卡:Nvidia GeForce GT 750M DDR3 2 GB;
- 3) 内存:4 GB DDR3 SDRAM, 1 600 MHz;
- 4) 操作系统:Windows 10 家庭普通版 64 位(简体中文)。

#### 4.1 多环境拟合效果

本文分别对常见的晴天(第 3 节已经展示)、雾天(包含阴天)、夜间、雨天等行车环境进行了测试,采用的是自适应阈值进行二值化处理,选择的是拟合效果较好的最小二乘法,最终效果如图 14-图 16 所示。



图 14 雾天行车拟合效果



图 15 雨天行车拟合效果



图 16 夜间行车拟合效果

可以看出本文车道线检测算法针对各种检测环境都能比较准确地拟合出相应的车道线。

## 4.2 与现有算法对比

随机提取各种环境下视频图像的 500 视频帧作为检测对象,并对比现有效果较好的 Bertozzi、Aly<sup>[15]</sup> 车道线检测算法,检测结果对比见表 2。

表 2 检测效果对比

算法名称	识别率/(%)	运算速度/(帧/s)
Bertozzi	84.45	25.6
Aly	89.97	28.5
本文算法	88.57	37.9

由表 2 可以看出,本文检测识别率与现有较好效果算法相差不多,但是运算速度却提升了 40% 左右,有着不小的优势,所以本文车道线检测算法的效果还是比较理想的。

## 5 结语

本文介绍了车道线检测的方法,主要包括图像的预处理、车道线边缘点的提取、车道线拟合等步骤。在预处理阶段主要创新点在于动态划分感兴趣区域以减少检测区域,最终减少了 80% 以上检测区域,且减少了干扰因素和检测时间,检测速度提升了近 40%;在边缘点提取算法中运用划分区域设定不同阈值提取边缘点以及进行隔行检测,提高了检测的实时性和准确性;在拟合算法中对比了

最小二乘法和三次样条拟合算法,最终选择了效果较好的最小二乘法拟合出车道线。试验结果表明本文的车道线检测算法对晴天、阴天、雾天、雨天、夜间等各种环境下直线、曲线车道线以及有干扰情况下较快车速的车道线均有较好的检测效果。

### 参考文献:

- [1] COLLADO J M, HILARIO C, DE la Escalera A, et al. Detection and classification of road lanes with a frequency analysis [C]. Proceedings of the 2005 IEEE Intelligent Vehicles Symposium. Nevada, USA, 2005: 78-83.
- [2] LU W N, ZHENG Y C, MA Y Q, et al. An integrated approach to recognition of lane marking and road boundary [C]. Proceedings of the 1st International Workshop on Knowledge Discovery and Data Mining, Adelaide, Australia, 2008: 649-653.
- [3] CHEN Q, WANG H. A real-time lane detection algorithm based on a hyperbola-pair model [C]. Proceedings of the 2006 IEEE Intelligent Vehicles Symposium. Tokyo, Japan, 2006: 510-515.
- [4] WANG Y, TEOH E K, SHEN D. Lane detection and tracking using B-snake [J]. Image and Vision Computing, 2004, 22(4): 269-280.
- [5] 段瑞玲,李庆祥,李玉和. 图像边缘检测方法研究综述 [J]. 光学技术, 2005, 31(3): 415-419.
- [6] 孙朋. 结构化道路中的车道偏离预警系统研究 [D]. 济南: 山东大学, 2014.
- [7] POLLARD E, GRUYER D, TAREL J P, et al. Lane marking extraction with combination strategy and comparative evaluation on synthetic and camera images [C]. Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on IEEE, 2011: 1741-1746.
- [8] 耿静静. 基于单目视觉的车道线检测与识别 [D]. 哈尔滨: 哈尔滨工业大学, 2007.
- [9] 王晓锦, 王增才, 赵磊. 基于消失点检测与分段直线模型的车道线识别 [J]. 机电一体化, 2017(5): 41-46.
- [10] 肖剑峰, 王敬东, 许莺, 等. 基于暗元先验图像去雾速度优化技术 [J]. 机械制造与自动化, 2017, 46(1): 197-201.
- [11] 毛星云. OpenCV3 编程入门 [M]. 北京: 电子工业出版社, 2015.
- [12] 刘公俊. 基于单目视觉的车辆检测与跟踪 [D]. 哈尔滨: 哈尔滨工业大学, 2015.
- [13] PAPPAS T N, NEUBHOFF D L. Least-squares model-based half-toning [J]. IEEE Transactions on Image Processing, 1992, 8(8): 1102-1116.
- [14] 陈浩, 华灯鑫, 张毅坤, 等. 基于三次样条函数的激光雷达数据可视化插值法 [J]. 仪器仪表学报, 2013, 34(4): 831-837.
- [15] ALY M. Real time detection of lane markers in urban streets [C]. [S.I.]: Proceedings of IEEE Intelligent Transportation Systems, 2008: 7-12.

收稿日期: 2018-10-16