

基于模型驱动的车线偏离警告系统^①

朱 峰

(上海城市管理职业技术学院 人文与信息学院, 上海 200438)

摘 要: 车线偏离警告系统(LDWS)是车载主动安全系统的一项主要功能, 研究了 LDWS 中的关键技术, 以车道模型为核心, 采用基于模型驱动的方法探测车道, 为解决弯道探测的问题, 将车线探测区域划分为多个子区域, 独立探测每个子区域中的车线段, 为提高探测效率并减少误识别, 每个子区域的尺寸根据每次探测的结果和车道模型的拟合值动态变化, 为增加探测的稳定性, 对近处子区域采用 Sobel 滤波处理, 对远处子区域采用根据车道几何特征动态设置的 Gabor 滤波处理, 实验表明, 系统能准确探测弯道并精确测算车道偏离范围, 在噪声环境中有良好的鲁棒性。

关键词: 车线检测; 弯道检测; 车线偏离警告系统; 车道模型

Lane Departure Warning System Based on Model Driven Approach

ZHU Feng

(Shanghai Technical College of Urban Management, Shanghai 200438, China)

Abstract: Lane departure warning system(LDWS) is one of the main functions for the vehicle active safety system. This paper researches some critical techniques of LDWS. The road model is made as the core and detect lanes based on the model driven approach. In order to detect curve, the region of detection is divided into multiple sub regions, then lane marks are detected in each region independently. For speeding up the detection and decrease error, the size of each sub region is changed based on the detection result and fitting values of the road model in each frame. For increasing the detection stability, the Sobel filter is adopted for nearby sub regions and Gabor filter is adopted for distant sub regions. Furthermore, the Gabor filter is set adaptively based on the road geometry. Experimental results shows the system can accurate detect the curve and predict lane departure range precisely. Furthermore, the system has good robustness in the noisy environments.

Key words: lane detection; curve detection; lane departure warning system (LDWS); road model

汽车主动安全技术是指为预防汽车发生事故, 避免人员受到伤害而采取的安全设计。典型的系统包括前车防撞系统(FCWS)、车道偏离警告系统(LDWS)、盲点探测系统(BLIS)及行人防撞系统(PCWS)等。在现有的系统中, FCWS、BLIS 及 PCWS 主要还是依赖激光雷达及毫米波雷达探测、视觉识别仅起到辅助判定的作用, 但是车线偏离警告系统(LDWS)只能依赖于视觉传感器进行识别, 由于车道探测相对于车辆及行人探测更加成熟可靠且成本低廉, 因此 LDWS 目前在高端车上已普遍配置。LDWS 仅仅是车道探测的一个

独立应用, 在目前的智能车辆系统中, 车道探测的信息通常还用于 FCWS、PCWS 的探测区域选择, 路径规划等方面, 是智能系统不可缺少的核心模块。

车道探测算法通常分直道探测算法和弯道探测算法。最常用的直道探测算法是采用 Canny 或 Sobel 滤波获取边缘信息, 再采用 Hough 变换探测直线, 如 Assidiq^[1]、Hardzeyeu^[2]的算法, Hough 变换的速度与采样点数量及直线探测范围成正比, 为了提高系统的实时性, 蔡^[3]采用空间约束方法, Satzoda^[4]采用分层自适应 Hough 变换方法进行加速, 为了减少透视造成的影

^① 收稿时间:2015-04-07;收到修改稿时间:2015-05-28

响, Mohamed Aly^[5]使用反向透视图(Inverse Perspective Mapping IPM)进行预处理, 去除透视造成的影响, 然后使用 RANSAC 拟合车线, 这种处理增加了探测的稳定性, 并且能在一定程度上减少路面噪点造成的影响. 关键词库的结合大大提高了信息抽取算法的准确性和通用性, 基于 Web 信息抽取的混合交通出行方案生成与表示系统的成功实验也证明了本文提出的 Web 信息抽取算法的实用性.

对于弯道探测, 通常的算法是将探测区域在世界坐标系中等距分割, 然后投射到图像坐标系成不等距区间, 在每个区间内使用直线探测的方法探测车线, 最后再使用模型进行拟合, 如 Wang^[6]基于均匀三次 B 样条曲线拟合车道线. 使用 3 个控制点控制车道线形状, 其中第一个控制点位于地平线上. 将图像分为 5 段, 在每个分段内, 使用 Canny 边缘滤波及霍夫传输获取候选直线段并利用直线段评估地平线、消失点及车道线中线段. 最后通过探测到的车道中线及消失点信息拟合三个控制点从而获取 B 样条曲线. 该算法的主要制约是在探测每个分段中的车道中线时, 必须同时探测到左右车线段才能进行. 对于每个分段仅探测到一侧车线的情况则算法失效. Wang^[7]提出一种分布式弯道检测算法, 使用改进的双曲线模型对弯道标识线进行描述. 将弯道检测算法分为二步: 第一步应用直线模型, 通过 Hough 变换求解其参数; 第二步以第一步所求参数作为双曲线模型的初始参数, 通过一定范围的动态搜索策略完成双曲模型参数的求解. 搜索时每一个候选参数对应于一对双曲线. 采用曲线上对应边缘点数量作为评价标准, 最终选取边缘点最多的模型参数. 该算法弯道探测的准确性直接依赖于边缘点探测的准确性. 但是文中仅采用垂直边缘滤波来获取边缘点, 对于复杂路况抗干扰性差. Chen^[8]通过建立车道边缘投影模型(LBPM)拟合弯道曲线, 对于大曲率的急转弯车道非常有效. Zhou^[9]将探测范围设置为近处直道探测段和远处弯道探测段两部分, 对于弯道探测段的每个小分段, 在多个可能的探测方向上使用不同的 Gabor 滤波器滤波图像并进行探测, 以便选择最优的探测结果. 该方法对于复杂路况有较强的抗干扰性, 但是大量的 Garbar 滤波需要较大的计算开销, 很难满足系统的实时性要求.

本文采用基于模型的可变区域车道探测算法, 每个探测区域由车线探测点及其方差组成. 车道模型仅

需拟合部分车线探测点即可更新. 可以应对由于路况的复杂性和车线的不连续性或者磨损所导致的仅能探测到部分车线的情况. 解决了 Wang^[6]算法的缺陷. 对于弯道探测, 仅在远处的探测区域中才能明显的探测到弯道曲率, 并且由于透视的影响, 远处的车线信息通常较弱且容易受地面噪点的干扰. 本文对远处的探测区域采用变方向 Garbor 滤波的方式探测最佳直线段, 具有较强的抗干扰性, 解决了 Wang^[7]算法的缺陷. 算法根据上一帧的探测结果约束 Garbor 滤波可变探测方向, 结合探测区域的可变性以及采用 Garbor 模板预生成技术减少了 Garbor 滤波的计算开销, 满足了系统的实时性能, 解决了 Zhou^[9]计算量过大的问题.

1 系统方案

目前的大部分手机上都集成了 GPS、GSensor 模块, 可以控制所需的车速、Camera 姿态位置信息, 同时 CPU 基本上都是双核 1.5GHz 以上, 能够满足系统需要的运算能力, 因此系统采用 Android 系统的手机作为算法验证载体. 系统架构如图 1 所示:

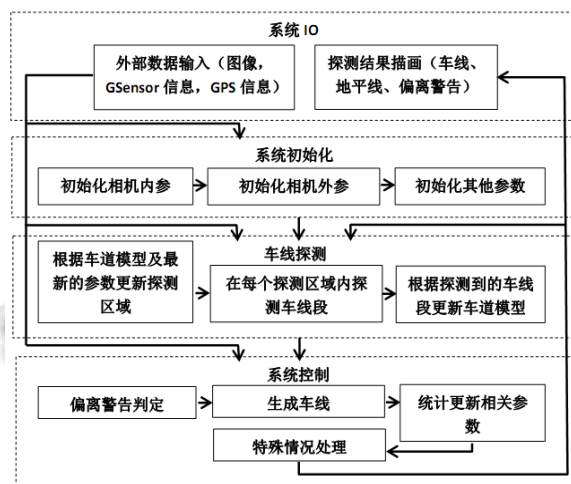


图 1 车载车线偏离警告系统方案

1)输入输出模块: 输入外部信号, 如车速、相机姿态(主要是倾角)及图像, 输出描画的车线、偏离警告信息和其他参考信息, 如地平线.

2)初始化模块: 初始化相机内参、外参、离地高度及车辆宽度信息. 由于手机支架默认放置在挡风玻璃中央后视镜下方, 因此不用考虑相机的左右位移;

3)车线探测模块: 首先根据当前参数和车道模型更新每个子探测区域, 然后在每个子探测区域中探测

车线段,最后通过矩阵拟合运算使用探测到的车线信息更新车道模型及相应的参数;

4)系统控制模块:接收探测模块的结果,综合从输入模块获取的车速、相机倾角等信息进行进一步分析.包括偏离警告判定,根据车道模型生成描画用的车线数据,这些数据将传入到输入输出模块进行描画显示,通过统计更新的方式更新车道宽度、相机倾角等参数,这些参数将用于车线探测模块,最后还需要处理一些特殊情况,比如当车线长时间探测不到时、车速偏低时等情况的处理.

2 基于模型驱动的车线探测算法

2.1 车道模型

在选择车道模型时,需考虑直道和弯道的路况,采用 Southall^[10]、Romuald Aufrère^[11]、Benmansour^[12]的车道表示模型,以相机光轴投影至地面的交叉点为世界坐标系中心,相机光轴投影至地面的轴向为 y 轴,垂直于光轴地面投影的轴向为 x 轴建立世界坐标系,车道模型的几何意义如图 2 所示.结合相机内参和外参,将车道模型投影至图像坐标系 (u,v) 空间,如图 3 所示,则车道模型定义如下:

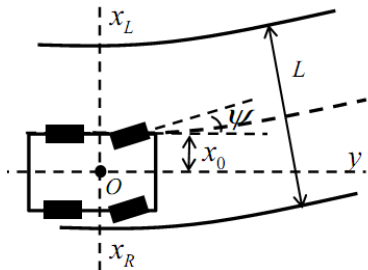


图2 世界坐标系下车道模型

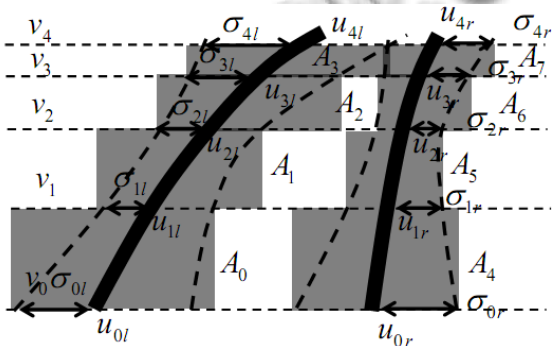


图3 图像坐标系下车线坐标及探测区域分割

$$(u_{il}, u_{ir})^T = g(v_i, x_0, C_1, \psi, L, \alpha) \quad (1)$$

$$u_{ir} = f_u \left(\frac{v_i - f_v \alpha}{f_v z_0} \left(x_0 + \frac{L}{2} \right) - \frac{f_v z_0}{2(v_i - f_v \alpha)} C_1 - \psi \right) \quad (2)$$

$$u_{il} = f_u \left(\frac{v_i - f_v \alpha}{v z_0} \left(x_0 - \frac{L}{2} \right) - \frac{f_v z_0}{2(v_i - f_v \alpha)} C_1 - \psi \right) \quad (3)$$

其中 $(u_{il}, u_{ir})^T$ 为左右车线相对于 v_i 坐标的 u 轴坐标, g 为车道模型, x_0 为车道中心线相对于坐标系中心 O 在 u 轴的偏差, ψ 为车辆转向角, α 为相机倾角, f_u , f_v 分别为在 u 轴和 v 轴上的焦距, z_0 为相机高度, C_1 为车道曲率, L 为车道宽度.该公式成立的前提是 α , ψ 较小,此时可以采用近似的三角函数公式 $\sin(\alpha) = \tan(\alpha) \approx \alpha$, $\cos(\alpha) \approx 1$, $\sin(\psi) = \tan(\psi) \approx \psi$, $\cos(\psi) \approx 1$,将复杂的计算公式简化为公式(1).

2.2 车线探测流程

参考 Romuald Aufrère^[11]、Benmansour^[12]的车线探测算法,将车线在世界坐标系下沿 v 轴等距划分为 n 个区域,使用公式(4)、(5)投影至图像坐标系:

$$\frac{u - c_u}{f_u} = \frac{-x}{y \cos(\alpha) + z_0 \sin(\alpha)} \quad (4)$$

$$\frac{v - c_v}{f_v} = \frac{z_0 \cos(\alpha) - y \sin(\alpha)}{y \cos(\alpha) + z_0 \sin(\alpha)} \quad (5)$$

其中, (u,v) 为图像坐标系坐标, (x,y) 为世界坐标系坐标, c_u , c_v 分别为图像中心坐标.投影效果如图 3 所示,当前算法划分为 4 个探测区域,相应 v 轴坐标 $(v_i | i \in [0,4])$.与 v_i 相交的左右车线 u 轴坐标序列定义为 $X_d = (u_{0l}, \dots, u_{4l}, u_{0r}, \dots, u_{4r})^T = (u_{il}, u_{ir})^T$,其相应的方差序列定义为一个正对角阵 C_{Xd} ,对角线上值为 $(\sigma_{u_{0l}}^2, \dots, \sigma_{u_{4l}}^2, \sigma_{u_{0r}}^2, \dots, \sigma_{u_{4r}}^2)$,其几何意义如图 3 所示.为了后继的算法描述,将车道模型参数定义为 $X_l = (x_0, \psi, C_1, L, \alpha)^T$,其相应的方差序列定义为正对角阵 C_{Xl} ,对角线上值为 $(\sigma_{x_0}^2, \sigma_{\psi}^2, \sigma_{C_1}^2, \sigma_L^2, \sigma_{\alpha}^2)$.定义全局参数为 $X_G = (X_d, X_l)^T$,全局参数方差为 C_{XG} .车线探测流程分以下三步进行:

1)初始化参数 $X_G(0)$, $C_{XG}(0)$.根据设置的初始参数 $X_l(0)$,采用公式(2)(3)算出 $X_d(0) = (u_{il}, u_{ir})^T$,从而获取 $X_G(0)$.根据设置的初始参数 $C_{Xl}(0)$ 采用公式(6)(7)获取 $C_{XG}(0)$.

$$C_{XG} = \begin{bmatrix} J_C \\ I_{5 \times 5} \end{bmatrix} C_{Xl} \begin{bmatrix} J_C \\ I_{5 \times 5} \end{bmatrix}^T \quad (6)$$

其中 $I_{5 \times 5}$ 是单位阵, J_C 是雅可比矩阵,定义为:

$$J_C = \begin{bmatrix} \frac{\sigma_{u_{0l}}}{\sigma_{x0}} & \frac{\sigma_{u_{0l}}}{\sigma_{\psi}} & \frac{\sigma_{u_{0l}}}{\sigma_{C1}} & \frac{\sigma_{u_{0l}}}{\sigma_L} & \frac{\sigma_{u_{0l}}}{\sigma_{\alpha}} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\sigma_{u_{4l}}}{\sigma_{x0}} & \frac{\sigma_{u_{4l}}}{\sigma_{\psi}} & \frac{\sigma_{u_{4l}}}{\sigma_{C1}} & \frac{\sigma_{u_{4l}}}{\sigma_L} & \frac{\sigma_{u_{4l}}}{\sigma_{\alpha}} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\sigma_{u_{0r}}}{\sigma_{x0}} & \frac{\sigma_{u_{0r}}}{\sigma_{\psi}} & \frac{\sigma_{u_{0r}}}{\sigma_{C1}} & \frac{\sigma_{u_{0r}}}{\sigma_L} & \frac{\sigma_{u_{0r}}}{\sigma_{\alpha}} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\sigma_{u_{4r}}}{\sigma_{x0}} & \frac{\sigma_{u_{4r}}}{\sigma_{\psi}} & \frac{\sigma_{u_{4r}}}{\sigma_{C1}} & \frac{\sigma_{u_{4r}}}{\sigma_L} & \frac{\sigma_{u_{4r}}}{\sigma_{\alpha}} \end{bmatrix} \quad (7)$$

$$C_{Xd} = [J_C] C_{Xl} [J_C]^T \quad (8)$$

2) 设定探测区域并在每段区域内探测车线与 v_i 的交点. 采用公式(7)(8)获取 C_{Xd} , 根据 X_d 及 C_{Xd} 以及 v 轴区间 (v_i, v_{i+1}) 设定探测区域 A_i . 本系统中如图3所示, 其中阴影区域即为8个探测区间 $(A_i | i \in [0, 7])$. 在每个探测区间 A_i 内采用 Hough 变换探测直线段并聚类, 进而获取与 (v_i, v_{i+1}) 的交点 (u_i, u_{i+1}) (u_i, u_{i+1}) . 对于上下相邻的探测区域 A_i, A_{i+1} 及其探测到的交点 $(u_i, u_{i+1})^{A_i}, (u_{i+1}, u_{i+2})^{A_{i+1}}$, 则其公共部分的车线探测点采用中值表达方式: $u_{i+1} = ((u_{i+1})^{A_i} + (u_{i+1})^{A_{i+1}}) / 2$. 依次探测每个区域获取 $X_d = (u_{0l}, \dots, u_{4l}, u_{0r}, \dots, u_{4r})^T$.

3) 更新全局参数 X_G, C_{XG} . 将新探测到的车线交叉点 X_d 代入公式(9)(10)(11), 将 X_G, C_{XG} 的状态从第 $k-1$ 帧更新为第 k 帧.

$$X_G(k) = X_G(k-1) + K[X_d - X_d'] \quad (9)$$

$$C_{XG}(k) = C_{XG}(k-1) - KHC_{XG}(k-1) \quad (10)$$

$$K = C_{XG}(k-1)H^T[HC_{XG}(k-1)H^T + R]^{-1} \quad (11)$$

其中 X_d' 并非 $X_d(k-1)$, 而是在第 $k-1$ 帧更新后的 $X_G(k-1)$ 中相对应的 $(u_{il}, u_{ir})^T$ 数据. H 是车线探测点的有效性矩阵. 由于实际探测的车线分段区域中可能只有部分的 u_i 坐标被探测到, 因此需要 H 指示其有效性. R 是探测误差的协方差矩阵, 定义为单位阵 $25I$.

4) 生成左右车线在图像平面的连续坐标序列. 将 X_G 中的 X_l 参数代入公式(2)(3)计算左右车线在图像平面的连续坐标序列.

5) 循环步骤2)3)4), 完成连续探测.

图4显示了一个典型的探测流程. 其中白色点划线表示地平线及根据初始参数预测的左右车线, 黑色实线框表示根据 X_d, C_{Xd} 设定的探测区域生成的动态探测区域 A_i , 白色虚线表示采用 Hough 变换探测直线段并聚类后的直线段, 白色实线为表示根据公式(2)(3)

计算的左右车线, 第 k 帧为初始位置, 随着全局参数 X_G, C_{XG} 的不断更新, 从第 $k+1$ 帧至第 $k+2$ 帧动态探测区域 A_i 逐渐减小, 这种探测策略不仅可以有效节省计算量, 并且可以有效避免车道中的字符、箭头等其他非车线对象的干扰. 由于在公式(11)中车线探测有效性矩阵 H 的存在, 在8个探测区域 A_i 中, 即使只有部分区域探测到车线, 也能进行车道拟合, 这种情况在实际路测中大量存在. 如图5显示的部分路测结果, 其中(a)、(b)仅一边车线探测到; (c)、(d)仅部分区域探测到. 此时, 车道模型参数都能准确拟合.

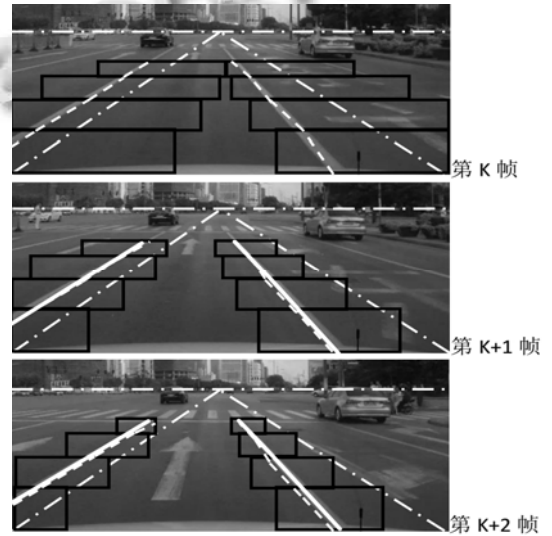


图4 车线探测流程

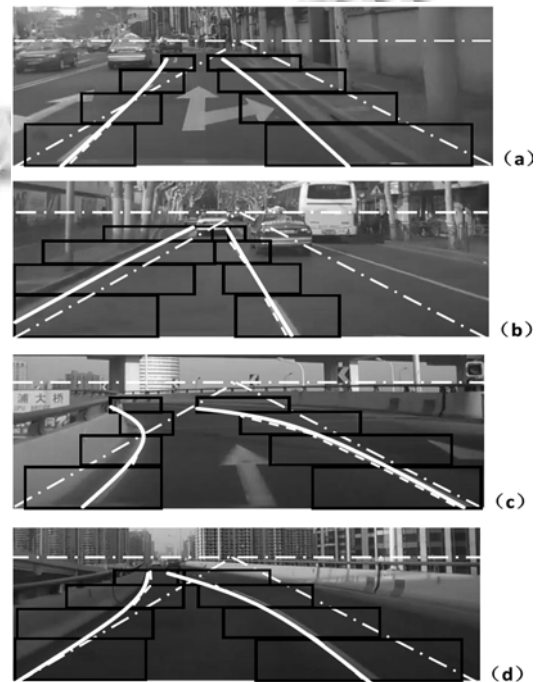


图5 典型车线探测结果

2.3 基于不同滤波器的区域内车线段探测算法

为了探测弯道, 根据 v 轴坐标将总的探测区域划分为8个子探测区域 A_i (如图3所示), 由于透视投影的关系, 车线宽度从近至远由粗到细, 探测区域的高度也逐渐变短, 这就导致采用 Hough 变换在每个 A_i 内拟合直线段时, 远处的探测区域内的车线探测准确度明显低于近处的探测区域内的车线探测, 另外, 根据公式(2)、(3)拟合的车道曲率参数 C_1 的值很小, 一般为 1.0×10^{-4} 级. 因此, 远处探测区域内车线探测的不准确会导致 C_1 在小范围内波动, 从而导致所计算的远处车线轨迹产生较大的误差. 为了解决该问题并兼顾处理速度, 针对不同的探测区域, 采用了3种滤波器, 具体算法如下:

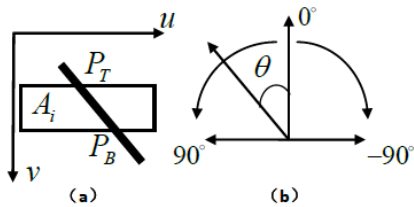


图6 Gabor 滤波器方向设定

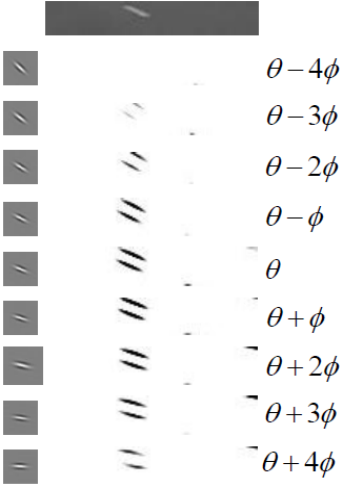


图7 检测区域中 A_7 典型场景的 Gabor 滤波情况

对于 A_0 、 A_4 探测区域: 由于此段车线较粗, 采用 5×5 垂直 Sobel 滤波处理后, 使用 Hough 变换查找直线段后合并.

对于 A_1 、 A_5 探测区域: 采用 3×3 垂直 Sobel 滤波处理后, 使用 Hough 变换查找直线段后合并. 对于 A_2 、 A_6 、 A_3 、 A_7 探测区域: 类似于 Zhou^[9] 的方法, 根据上一帧的探测结果采用变方向 Garbor 滤波, 使用 Hough 变换查找直线段后合并. 在进行多方向的探测和比较后获取最佳的车线探测结果. 具体处理步骤如

下:

1) 根据车道模型生成车线并计算与检测区域 A_i 上下边界的交点 P_T 和 P_B , 如图 6(a) 所示.

2) 计算 Gabor 滤波器中的滤波方向 θ :

$$\theta = 90 - \arctan\left(\frac{P_T(y) - P_B(y)}{P_T(x) - P_B(x)}\right) \quad (12)$$

图 6(b) 显示了 Gabor 滤波器中的 θ 与 (u, v) 空间中车线方向的关系.

3) 以 θ 为基方向, ϕ 为变化步长, 在 $[\theta - n/2 \times \phi, \theta + n/2 \times \phi]$ 范围生成 $n+1$ 个 Gabor 滤波器进行滤波. Gabor 滤波器生成公式如下:

$$f_G = f_{G_Real} + f_{G_Imaginary} \quad (13)$$

$$f_{G_Real}(x, y) = e^{\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right)} \cos\left(2\pi \frac{x'}{\lambda}\right) \quad (14)$$

$$f_{G_Imaginary}(x, y) = e^{\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right)} \cos\left(2\pi \frac{x'}{\lambda} + \frac{\pi}{2}\right) \quad (15)$$

其中 $x' = x \cos \theta + y \sin \theta$, $y' = -x \sin \theta + y \cos \theta$, λ 为波长, γ 为空间纵横比. 在实际应用中设置 $n=8$ 、 $\phi=5$, 图 7 显示了探测区域 A_7 中一个典型场景的滤波结果, 滤波结果随着 θ 的变化有明显的不同.

4) 对于每个方向的滤波结果, 使用 Hough 变换查找直线段后合并. 对于所有方向的直线探测结果, 沿着查找到的直线段附近累加滤波值, 选择滤波值最大的直线探测结果为最终探测结果.

表1 不同路况下车线方向范围

路况	方向变化范围		
	$A_0 A_4$	$A_1 A_5$	$A_2 A_6 A_3 A_7$
直道	$10^\circ \leq \theta \leq 50^\circ$	$10^\circ \leq \theta \leq 50^\circ$	$10^\circ \leq \theta \leq 50^\circ$
弯道	$20^\circ \leq \theta \leq 65^\circ$	$20^\circ \leq \theta \leq 65^\circ$	$20^\circ \leq \theta \leq 80^\circ$
急转弯道	$25^\circ \leq \theta \leq 70^\circ$	$25^\circ \leq \theta \leq 70^\circ$	$25^\circ \leq \theta \leq 85^\circ$

滤波器自适应的准确性与各探测区域内车线方向变化范围相关. 不同路况下的车线方向变化范围如表 1 所示. 对于中近探测区域内的线段最大变化角度不大于 70° , 一般在 50° 左右, 具有较强的垂直梯度分量, 完全满足边缘提取的需要. 远处探测区域内的线段最大变化角度不大于 85° , 较倾斜一侧的车线段一般大于 65° , 此时必须采用变方向 Gabor 滤波器才能满足滤波需要. 实际采用的 Gabor 滤波器方向在上一帧探测方向的 $\pm 20^\circ$ 范围内变化, 滤波结果完全满足边缘提取的需要.

Gabor 滤波器的运算优化及系统开销. 由于滤波模板的生成涉及指数及三角函数运算, 且四个运算区域需要 36 个模板, 若实时运算会产生庞大的系统开销. 本文采用预计算模板的方式, 利用存储空间置换计算时间. 若滤波方向 $|\theta| \leq 30^\circ$, 则统一采用 $\theta = 15^\circ$ 的滤波模板, $|\theta| \in (30^\circ, 85^\circ]$ 则以 5° 为单位, 生成 11 个不同角度的模板. 考虑到 (A_2, A_6) , (A_3, A_7) 探测域采用不同的波长以及 θ 的正负方向, 一共预生成 $(11+1) \times 2 \times 2 = 48$ 个模板. 运行时, 将实时生成的 θ 归一化为 5° 间隔区间内选取合适的模板进行滤波操作. 系统在 CPU 单核 1.5GHZ 手机上测试, Gabor 滤波时间如表 2 所示.

表 2 Gabor 平均滤波时间
(不包含系统初始化时的模板预生成时间)

运行状况	平均时间	占总时间比率
系统重启(长时间探测失败时)	70ms	60%
车线探测成功(探测区域缩小)	27 ms	40%
车线探测失败(探测区域扩大)	55 ms	55%
综合(所有运行状况)	40 ms	48%

综合状态下 Garbor 滤波的卷积运算约为总运算时间的一半. 若车线长时间探测不到引起系统重启(探测区域重新设置)或探测区域扩展至较大状态, 则 Garbor 滤波的卷积运算为系统的主要运算开销. 若不采用 Gabor 模板预生成技术, 根本无法达到实时运行状态.

3 综合多种信号的相机倾角统计更新

系统采用 Android 系统的手机作为算法验证载体, 在将手机安装至支架时, 一般安置在后视镜下方, 即左右车辆中间处, 此时系统的精度受三个外部参数影响: 相机倾角 α , 方位角 β 及倾斜角 φ . 其中 β 只能通过视觉目测, α 及 φ 可以在车辆静止时通过 GSensor 估算初值. 由于 α 对系统的影响最大, 为了简化计算, 在公式(1)定义的车道模型中仅包含了 α .

在车辆行驶的过程中, 由于路面经常会起伏不定, 需要根据车线的探测情况, 对 α 值进行持续更新, 以保证车道模型拟合的准确性. 但是在车载情况下, 由于某些情况下的车线持续探测错误或者由于严重的抖动使得支架偏离, α 角发生较大变化, 就可能使得车道模型无法准确拟合, 且这种无法探测的情况会一直

持续. 为了保证系统的稳定稳定性, 综合 GSensor、GPS 信号和车线探测结果对 α 角进行统计更新. 更新策略如下:

1)在系统启动时, 通过 GSensor 统计多帧评估 α 角作为车道模型的初始化值. 车辆在行驶过程中, 从 GSensor 获取的角度值易受车辆加减速、转向的影响, 但是通过均值和方差的方式对获取的角度序列进行统计分析, 通过去除变化过大的角度, 取值统计平均角度还是相对稳定的.

2)检测 GPS 信号, 在车速大于 15KM/H 时, 启动车线探测系统, 反之停止车线探测, 之所以加上速度限制, 是由于在低速时前方车辆可能过于接近, 导致车线探测无法进行并且容易将车身的一部分误识别为车线. 在车线探测启动状态下, 根据车线探测结果和车道模型拟合结果, 通过统计分析获取 α 角.

3)检测 GPS 信号, 在确认车线探测启动状态下, 检查车线探测结果, 若长时间探测失败, 则对比从 GSensor 获取的 α 角和从之前车线探测结果获取的 α 角, 若两角度相差 $> \pm 5^\circ$, 则认为从之前车线探测结果获取的 α 角有误, 重新设定从 GSensor 获取的 α 角为车道模型统初始值, 重新启动车线探测, 若两角度相差 $< \pm 3^\circ$, 则认为当前车道无车线或者车线不清晰导致车线探测失败, 继续探测.

4)检测 GPS 信号, 在确认车线探测启动状态下, 检查车线探测结果, 若长时间探测失败, 且从 GSensor 获取的 α 角和系统运行初始化阶段获取的 α 角相差 $> \pm 5^\circ$, 则认为可能是支架发生移动导致系统无法获取有效的探测区间, 此时发出调整支架的警报.

另外, 在上下坡时间较长时, 从 GSensor 会获取错误的 α 角, 此时, 若地面车线能探测到, 则不会发出误报, 否则, 可能会发出调整支架的误报.

4 实验结果与分析

系统在 CPU 单核 1.5GHZ 手机上测试, 平均处理速率为 10~12FPS, 基本满足实时性要求. 对于远处的 A_2 , A_6 , A_3 , A_7 四个探测区域采用变方向 Garbor 滤波的探测方法探测车线占了 50%左右的运算时间. 车线探测测试结果如表 3 所示.

图 8 显示了夜间探测的部分结果, 在有路灯的情况下, 可以探测到弯道, 若仅前车灯照明, 则 A_3 , A_7 探测区域大部分情况下无效.

表 3 车线探测测试结果

路况	识别率(%)	误识别率(%)
白天/车线清晰	>95	<5
夜间/车线清晰	>95	<5
车线磨损	由磨损度确定一 般<60	不定
近处车辆干扰	<80	>20
林荫道(严重树影干扰)	<85	>15

在长期路测中,发现车线探测产生的误识别主要由以下几个因素造成:



图 8 夜间车线探测结果

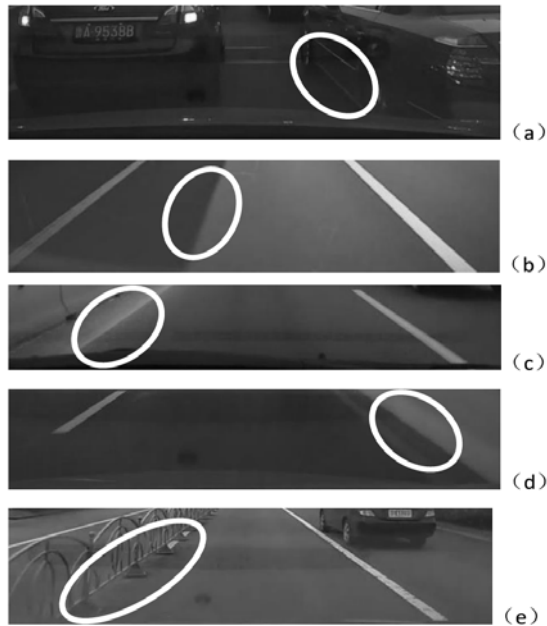


图 9 路测中车线探测失败的典型场景

1)近处车辆干扰,尤其是超车至当前车道的车辆,

容易将其一侧误识别为车线的一部分.如图 9(a)中的典型场景,后方车辆切入到合适的车线探测位置,且车身一部分有明显的直线投影,且投影角度符合车线角度范围,此时在椭圆区域容易产生误识别,一个有效的规避方法是设定偏离警告启动速度限制,如超过 30KM/H 时启动车线探测系统.

2)路面有直线型投影,如某些建筑物在地面的长条投影,如图 9(b)中的典型场景,一个可能的解决方案是采用 BWB(Black-White-Black)车线模式验证方式去除该类误识别. BWB 能解决大部分该类误识别,但是也会对某些场景造成干扰,导致车线无法识别.如图 9(c)(d)的场景,由于车线和车道边缘融合在一起,若采用 BWB 模式,会造成车线长时间无法识别. (3)路上的树木投影干扰,城市道路中的护栏隔离带等,会造成一侧的车线长时间无法识别.如图 9(e)中的典型场景.

5 结论

文章基于模型驱动方式实现了车道检测算法,并在此基础上实现了 LDWS 的功能,算法通过根据上帧探测的结果自动调整本帧探测区域的大小,以此减少计算量,提高探测的实时性,在探测过程中,使用多方向 Garbor 滤波的方式增加远处车线探测的稳定性,从而提高弯道曲率拟合的精确性,算法在车载设备上(Android 系统)进行的大量路测,实验表明能满足大部分路况,具有很高的探测性能,由于使用了多方向 Garbor 滤波,消耗了大量的计算量,算法的实时性还有待进一步提高.

参考文献

1 Assidiq AA, Khalifa OO, Islam MR, Khan S: Real Time Lane Detection for Autonomous Vehicles. International Conference Computer and Communication Engineering (ICCCE), 2008: 82-88.

2 Hardzeyeu V, Klefenz F. On using the hough transform for driving assistance applications. Proc. of the 4th Int. Conf. Intell. Comput. Commun. Process. Shanghai, China, 2008: 91-98.

3 Cai AS, Ren MW. Robust method for extracting of vehicle-tract line. Computer Engineering and Design, 2011, 32(12):4164-4168.

- 4 Satzoda RK, Sathyanarayana S, Srikanthan T. Hierarchical additive hough transform for lane detection. *Embedded Systems Letters*, 2010, 2(2): 23–26.
- 5 Mohamed A, Pasadena CA. Real time detection of lane markers in urban streets. *Intelligent Vehicles Symposium*, IEEE, 2008: 7–12.
- 6 Wang Y, Teoh EK, Shen D. Lane detection and tracking using B-Snake. *Image and Vision Computing*, 2004, 22: 269–280.
- 7 Wang JW, An XJ. A multi-step algorithm for curved lane detection. *2010 Chinese Conference on Pattern Recognition (CCPR)*. 2010: 21–23.
- 8 Chen Y, He MY. Sharp curve lane boundaries projective model and detection. *INDIN*, 2012: 1188–1193.
- 9 Zhou S, Jiang Y, Xi J, Gong J, Xiong G, Chen H. A novel lane detection based on geometrical model and gabor filter. *2010 IEEE Intelligent Vehicles Symposium (IV)*. 2010: 59–64.
- 10 Southall J, Taylor C. Stochastic road shape estimation. *International Conference on Computer on Computer Vision*. Vancouver, BC, Canada: IEEE, 2001:205–212.
- 11 Aufrère R, Chapuis R, Chausse F. A model-driven approach for real-time road recognition. *Mach. Vis. Appl*, 2001, 13(2): 95–107.
- 12 Benmansour N, Labayrade R, Aubert D, Glaser S. Stereovision-based 3D lane detection system: a model driven approach, *Intelligent Transportation Systems*. 11th International IEEE Conference on Intelligent Transportation Systems. Beijing, China, 2008: 182–188.