

基于 DSP 的车道偏离检测与车辆前向车距检测^①



刘金清¹, 陈存弟²

¹(福州外语外贸学院, 福州 350202)

²(福建师范大学 光电与信息工程学院, 医学光电科学与技术教育部重点实验室, 福州 350007)

通讯作者: 刘金清, E-mail: 2060008441@qq.com

摘要: 基于智能交通的快速发展, 研究了在高速路段下基于机器视觉的车道偏离检测与车辆前向安全车距检测技术. 首先固定车载相机, 通过相机标定获取相机的内参数和外参数, 进而设计车距检测模型. 该模型不但能够检测出前方车辆与无人车的距离, 还能计算出前方车辆相对于摄像机光轴的偏转角度. 接着在 CCP 偏离检测算法的基础上, 设定安全和报警区来建立车道偏离模型, 并对当前车辆的偏离结果作出正常行驶的评判. 最后借助 TI 的 DVSDK 组件包将算法移植到嵌入式平台 DSP-DM3730 上测试. 实验表明, 本文设计的车距检测模型和车道偏离模型在解决无人车的前向防撞检测和车道偏离检测等问题上具有较好的参考价值.

关键词: 偏离检测; 防撞检测; DSP

引用格式: 刘金清, 陈存弟. 基于 DSP 的车道偏离检测与车辆前向车距检测. 计算机系统应用, 2020, 29(3): 269-277. <http://www.c-s-a.org.cn/1003-3254/7300.html>

Lane Departure Detection and Vehicle Forward Distance Detection Based on DSP

LIU Jin-Qing¹, CHEN Cun-Di²

¹(Fuzhou University of International Studies and Trade, Fuzhou, 350202, China)

²(Key Laboratory of Optoelectronic Science and Technology for Medicine (Ministry of Education), College of Photonic and Electronic Engineering, Fujian Normal University, Fuzhou 350007, China)

Abstract: Base on the rapid development of intelligent transportation, this work studies lane departure detection and vehicle forward safety distance detection technology under high-speed section based on the machine vision. First fix the car camera, obtain the camera's internal and external parameters through camera calibration, and then design the distance detection model which can not only detect the distance between the front vehicle and the unmanned vehicle, but also calculate the deflection angle of the front vehicle relative to the optical axis of the camera. Secondly, based on the CCP (The Car's Current Position) deviation detection algorithm, the safety and alarm zones are set to establish lane departure models, and the algorithm judges whether the current vehicle is deviated or not. Finally, the algorithm is transplanted to the embedded platform DSP-DM3730 by TI's DVSDK (Digital Video SDK). Experiments show that the vehicle distance detection model and lane departure model designed in this work have good reference value in solving the problems of forward collision detection and lane departure detection of unmanned vehicles.

Key words: departure detection; collision detection; DSP

① 基金项目: 国家自然科学基金 (61179011); 国家自然科学基金青年科学基金 (41701491); 福建省自然科学基金 (2017J01464)

Foundation item: National Natural Science Foundation of China (61179011); Young Scientists Fund of National Natural Science Foundation of China (41701491); Natural Science Foundation of Fujian Province (2017J01464)

收稿时间: 2019-07-24; 修改时间: 2019-08-23; 采用时间: 2019-09-04; csa 在线出版时间: 2020-02-28

车道偏离检测和前向车距检测的目的是防止无人车偏离本车道或者与前方车辆距离较近时发生交通事故. 近年来, 国内外学者基于机器视觉的方法对车道偏离^[1-4]和车距检测技术^[5-7]展开了相应的研究. 本文在前人研究的基础上, 实现了车道偏离和车距检测算法的改进与仿真, 并移植到嵌入式平台上测试.

车道偏离检测算法和车辆前向车距检测算法均需要通过相机采集的图像重建出三维世界, 以此算出客观世界中的真实数据. 在机器视觉中, 三维重建可以定义为根据图像坐标系与世界坐标系的转换关系进而计算出相机参数, 进一步推导出图像坐标与世界坐标系的关系, 因此车道偏离检测与防撞检测的首要条件是相机标定. 本文采用张氏相机标定方法^[8].

1 相机标定原理

相机标定的理论推导涉及 4 种坐标系, 分别是像素为单位的像素平面坐标系 (u, v) ; 表示每毫米包含的像素数量的图像物理坐标系 (x, y) ; 相机坐标系 (X_c, Y_c, Z_c) 和世界坐标系 (X_w, Y_w, Z_w) . 如图 1 所示.

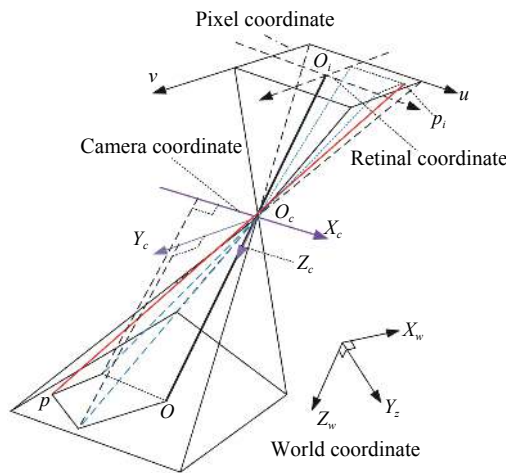


图 1 坐标映射模型

(1) 图像物理坐标系到像素坐标系的转换

图像像素坐标系 (u, v) 中的 u 和 v 分别表示行和列, 其坐标原点在图像的左上角. 图像物理坐标系 (x, y) 的 x 和 y 分别与图像像素的 u 、 v 轴平行, 相机的光心与像平面的交点为坐标原点. 假设光心对应的图像平面坐标为 (u_0, v_0) , d_x 表示 x 方向每个像素的毫米宽度, d_y 表示表示 y 方向每个像素的毫米宽度, 式 (1) 为图像坐标系转换为像素坐标系.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/d_x & 0 & u_0 \\ 0 & 1/d_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

(2) 相机坐标系到物理坐标系的转换

光心是相机坐标系 (X_c, Y_c, Z_c) 的原点, Z_c 轴与光轴重合且方向相同, X_c 与 Y_c 轴各自平行于图像物理坐标系的 x 和 y 轴. 令相机的焦距为 f , 相机坐标系与像素坐标系的关系式为式 (2) 所示.

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2)$$

(3) 世界坐标系到相机坐标系的转换

世界坐标系 (X_w, Y_w, Z_w) 到相机坐标的转换实质上是平移和旋转的过程, 其中旋转矩阵 T 为三维平移矢量, 旋转矩阵 R 为 3×3 的正交单位矩阵, 其转换关系为式 (3) 所示.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ O^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3)$$

整合 4 个坐标系的转换关系, 可以得出像素坐标系与世界坐标系的变换过程, 其转换过程为:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f d_x & 0 & u_0 & 0 \\ 0 & f d_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ O^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = A[R T]M \quad (4)$$

式 (4) 中包含有 $f d_x$ 、 $f d_y$ 、 u_0 、 v_0 等 4 个相关参数, $[R, T]$ 是相机的外参矩阵. 在相机的内参数已知情况下, 就能得到像素坐标与世界坐标之间的投影矩阵 $P=A[R, T]$. 因此, 对于世界坐标上的任意点 $M(X_w, Y_w, Z_w)$ 可以计算出对应的像素坐标 (u, v) , 但是 P 具有不可逆性, 不能用已知的像素坐标推导出世界坐标, 而在车道偏离检测与车距检测中均需要从像素坐标计算出世界中客观的物理量, 这就需要在已知相机内外参数情况下再次建立相应的路面模型计算出相关的物理量.

(4) 求解相机畸变参数

在几何光学和阴极射线管显示中, 畸变是对直线投影的一种偏移, 即一条直线投影到图片上并不是以直线的形式投影在图像上. 一般将畸变分成对图像的投影影响较大的径向畸变和切向畸变, 其表达式为:

$$\begin{bmatrix} x_{crt} \\ y_{crt} \end{bmatrix} = \begin{bmatrix} (1+k_1r^2+k_2r^4+k_3r^6) \begin{bmatrix} x \\ y \end{bmatrix} \\ 2p_1xy+p_2(r^2+2x^2) \\ 2p_1(r^2+2y^2)+2p_2xy \end{bmatrix} \begin{matrix} \text{(径向畸变)} \\ \text{(水平畸变)} \end{matrix} \quad (5)$$

(5) OpenCV 上的相机标定

张氏标定法简单稳定, 因此得到广泛的使用. 具体标定步骤如下:

① 制作棋盘标定板: 在固定的相机角度下, 拍摄不同角度, 多尺度且旋转多变的棋盘图像 20 张, 棋盘格数为 9×5.

② 令棋盘的左上角为世界坐标原点, 初始化棋盘格上所有角点的世界坐标, 且 X_w 轴与 Y_w 轴分别对应棋盘的长和宽, $Z_w = 0$, 每个棋盘格的长宽为 30×30 mm, 如图 2 所示. 图 3 为检测到的棋盘角点结果.

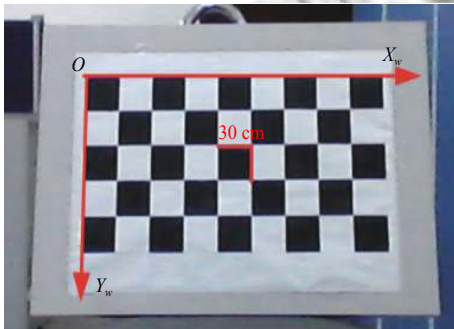


图 2 棋盘世界坐标

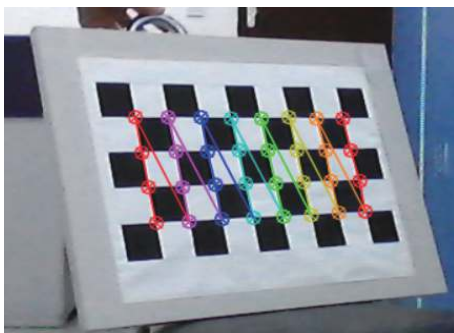


图 3 棋盘角点坐标

③ 引用 calibrateCamera() 函数计算相机的内外参数和畸变参数得到的标定参数为:

畸变系数:

[0.2051, -0.7335, -0.018 52, -0.039 42, 1.571 67]

相机内参矩阵:

$$\begin{bmatrix} 885.783, & 0, & 268.6169 \\ 0, & 882.7993, & 192.251 95 \\ 0, & 0, & 1 \end{bmatrix}$$

2 车距检测算法的实现

2.1 构建车距检测模型

车距检测模型, 如图 4 所示. 假设相机能够获取的路面区域为 ABCD, 路面 E 点投影在图像最顶边的 E_o 点上; F 点是相机刚好拍到靠近相机的路面边界处, 其投影在图像最底边 F_o 点上, G 点是相机光心射向的路面焦点, 该点投影在图像的 o 点处; 路面上随意的 P 点投影在 P_o 点处; Q 点表示相机的位置, h、ZF 和 ZG 可用尺子测量. $\angle ZQF = \gamma$, $\angle FQG = \beta$, $\angle TQG = \alpha$. 假设 P 点是检测到的车辆, 计算车距即是求解路面上 ZP 距离, Z 是相机在路面上的投影点.

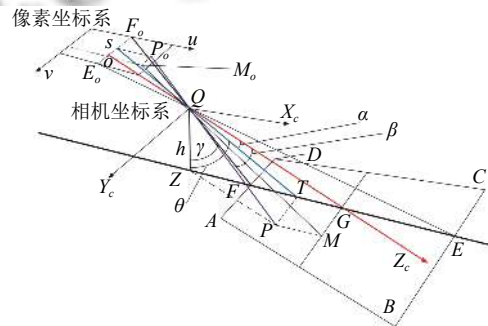


图 4 路面车距检测模型

根据图 4 的路面建模图, 结合相机成像规则, 可以得到以下关系:

(1) $\angle OQS = \alpha = \arctan(SO/f_{d_y})$;

$\angle OQF_o = \beta = \arctan(F_oO/f_{d_y})$;

$\gamma = \arctan(ZF/h)$.

(2) if $P_o.v > v_0 \Rightarrow \theta = \gamma + \beta - \alpha$;

if $P_o.v \leq v_0 \Rightarrow \theta = \gamma + \beta + \alpha$;

$\Rightarrow ZT = h \cdot \tan \theta$.

(3) 又: $PT = MG; S P_o = O M_o; QG = \sqrt{h^2 + (ZG)^2}$;

$\frac{QG}{MG} = \frac{f_{d_x}}{O M_o}$;

$\Rightarrow PT = MG = O M_o \cdot QG / f_{d_x}$.

(4) 无人车与前车距离: $\Rightarrow ZP = \sqrt{(ZT)^2 + (PT)^2}$.

2.2 车距检测试验

本文在实验室通过模拟车道来测试算法的可靠性. 相机光心投影在图像上的像素坐标为 $(u_0, v_0) = (268.62, 192.25)$, x 方向的焦距为 $f_{d_x} = 885.78$, y 方向的焦距为 $f_{d_y} = 882.80$, 相机距离地面的高度为 $h = 690$ mm, 通过实地测量相机投影在路面与图像最低边在真实路

面上的距离为 $ZF=2050$ mm.

图5中随机的路面坐标 $P_o(300,295)$, 投影在真实路面上的数据为: $PT=197.13$ mm, $ZT=5520.95$ mm, 通过欧氏公式随机路面点到相机的距离为: $ZP=5524.47$ mm, 实际距离为 5400 mm. 模拟的车辆位置与相机的偏离角度为右偏 2.05 度. 计算结果与实际测量的结果之差仅为 $|5524.47 - 5400| = 124.47$ mm.

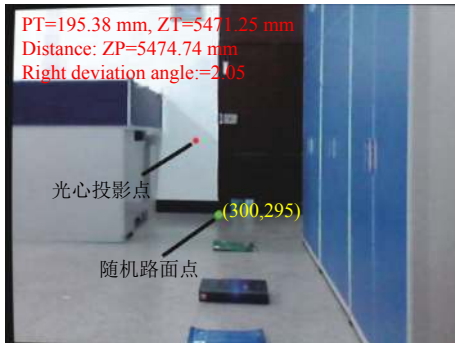


图5 畸变校正与路面测距模拟结果图

表1中的距离结果以 mm 为单位, 角度单位为度, 从模拟路面上的 6 组数据中可以看出, 实际值与测量值的误差较小, 在 10 m 内的误差控制在 200 mm 内, 说明本文建立的车距测量模型较准确. 在实际的高速路面上, 安全车距不少于 50 米, 可以通过本文的算法设置固定的车距安全阈值来实现车距安全检测.

表1 无人车相对于光轴的偏航角和车距测量表

序号	坐标 (x, y)	实际值	测量值	误差	偏向角	偏向
1	274,412	2630	2639.66	9.66	0.36	右
2	82,410	2700	2761.44	61.44	12.27	左
3	293,344	3700	3773.73	73.73	1.47	右
4	300,295	5400	5524.47	124.47	2.05	右
5	318,273	6700	6842.66	142.47	3.21	右
6	325,420	10 700	10 864.6	164.6	3.65	右

2.3 构建车道偏离模型

由于无人车驾驶过程中容易出现偏航现象, 如果不及调整方向便会偏离到其他车道. 此外, 车辆压到分道线导致左轮或者右轮偏离在其他车道中, 将影响其他车辆的正常运行, 甚至导致不可预知的交通事件, 因此需要车道偏离提示. 车辆准备超车时会提前打开左转向灯, 所以, 无人车在没有检测到左转向灯打开时, 车道偏离报警系统才会发挥相应的警报作用.

无人车正常行驶时, 行驶方向与左右分道线趋近平行, 因而, 可以利用车道俯视图模型来计算出偏航的

角度和车辆与左右分道的边距, 以做出相应的报警信息反馈.

$$\Delta y = \begin{cases} \Delta y_1 = \frac{b_l}{2} - \left(y_0 + \frac{b_c}{2}\right) \\ \Delta y_2 = \frac{b_l}{2} + \left(y_0 - \frac{b_c}{2}\right) \end{cases} \quad (6)$$

CCP 算法需要预知无人车的横向宽度 b_c 和车道宽度 b_l , 标准高速公路的车道宽度为 3.75 m, 可以通过式 (6) 判断车道偏离情况. y_0 表示无人车中心与车道中线的距离. 如果 $\Delta y_1 < 0$ 或 $\Delta y_2 < 0$, 则认为是车道偏离; 如果 $\Delta y_1 > 0$ 和 $\Delta y_2 > 0$ 则认为是无人车在当前行驶的车道内. 如果判别到无人车是左偏离, 那么如图6所示, 可知 Δy_1 是车辆与左分道线的距离, Δy_2 是车辆与右分道线的距离; 如果判别到无人车是右偏离, 那么 Δy_1 是车辆与右线的距离, Δy_2 是车辆与左线的距离.

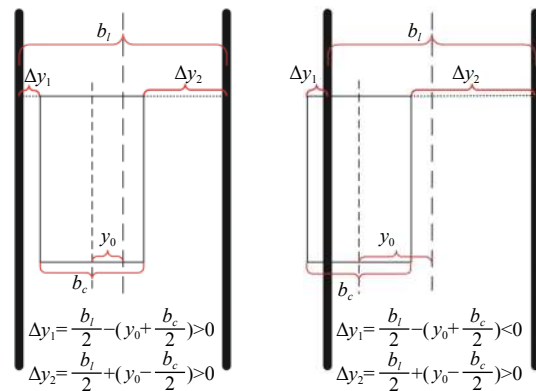


图6 CCP 车道偏离检测

相机的光心在图像上的坐标为 (u_0, v_0) , 因此图像中坐标点 $F_o(u_0, I_h)$ 为相机光轴投影在地面上投影的点, I_h 是图像的高, I_w 为图像的宽. 如图7所示, 是一张车身右偏而行驶方向偏左的图像, 图中 L_1 和 L_2 是左右车道线, L_3 是 L_1 和 L_2 交叉角的平分线, $P_t(u, v)$ 是 L_3 上任意一点, L_3 与图像底边的交点为 $V(u, v)$.

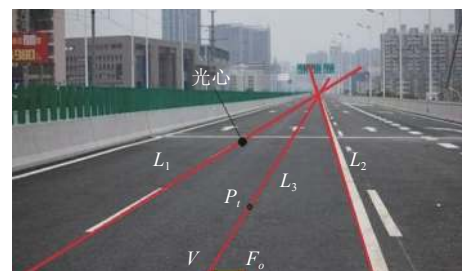


图7 车向偏离示意图

如图8所示,在路面上建立一个路面坐标,以分道线平行的中线为正向y轴,无人车所在的位置横向为x轴,右方向为正,其中 θ 是无人车的偏航角, $\angle FV_oP_o = \varphi$, I_b 是相机成像后靠近车辆的底边, V_o 、 P_o 分别是图像中V、 P_t 点的实际路面投影点, P_o 为随机投影点,像素坐标中直线 $u = u_0$ 与车道平分线的交点投影在路面上,即是图中 P_o 点。

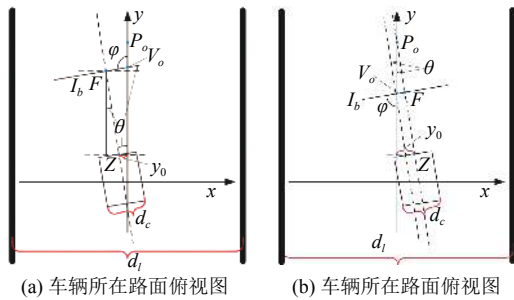


图8 车辆偏离俯视图模型

(1) 无人车偏航角的计算

① 余太武^[9]计算左右车道的角平分线作为车道中线,而本文计算车道中线的方法是基于左右车道斜率和截距求平均的方法。

$$\text{车道中线} : y = \frac{k_1 + k_2}{2} \cdot x + \frac{b_1 + b_2}{2} \quad (7)$$

如图9所示,基于角平分线求解车道中线的方法存在误差,而基于式(7)求解的结果正确。

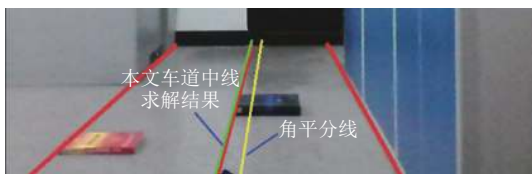


图9 求解车道中线的比较

② 基于2.1节介绍的路面建模图像,我们随机取车道中线上的某点 P_t , 接下来,计算出车道中线上V点坐标,分别计算出 F_o , V , P_t , 分别对应在地面上的 F , V_o , P_o 点,以此计算出实际地面上的距离 $|FV_o|$ 、 $|P_oV_o|$ 和 $|FP_o|$ 。

根据余弦定理求 $\angle FV_oP_o = \varphi$:

$$\varphi = \arccos\left(\frac{|FV_o|^2 + |P_oV_o|^2 - |FP_o|^2}{2|FV_o||P_oV_o|}\right) \quad \varphi \in [0, \pi] \quad (8)$$

③ 计算偏航角:

$$\theta = \begin{cases} \varphi - \pi/2 & \varphi \geq \pi/2 \\ \pi/2 - \varphi & \varphi < \pi/2 \end{cases}, \quad \theta \in \left[0, \frac{\pi}{2}\right] \quad (9)$$

if $\theta > \pi/36 \Rightarrow$ 车辆行驶方向开始偏离车道

(2) 无人车与左右分道线的距离计算

① 计算 $y_0 = \begin{cases} |FV_o| \cdot \cos\theta - |FZ| \cdot \sin\theta, & \varphi \geq \pi/2 \\ |P_oZ| \cdot \sin\theta, & \varphi < \pi/2 \end{cases}$

② 若 $F_o.u > V.u$, 可初步判断为车辆向左偏离车道中线。

无人车距左分道线距离为:

$$D_L = \frac{b_l}{2} - \left(y_0 + \frac{b_c}{2}\right) \quad (10)$$

无人车距右分道线距离:

$$D_R = \frac{b_l}{2} + \left(y_0 - \frac{b_c}{2}\right) \quad (11)$$

③ 若 $F_o.u < V.u$, 可初步判断为车辆右偏离车道中线
无人车距左分道线距离:

$$D_L = \frac{b_l}{2} + \left(y_0 - \frac{b_c}{2}\right) \quad (12)$$

无人车距右分道线距离:

$$D_R = \frac{b_l}{2} - \left(y_0 + \frac{b_c}{2}\right) \quad (13)$$

获取到无人车与左右分道线的偏离距离和偏航角后,就可以设定相应的阈值来判断当前无人车的行驶状况。如图10所示,沿车道方向分割成安全区域和报警区域。对于一般小车宽度为1.5-1.8 m之间,标准高速公路宽度为3.7 m。本文设置车辆与左右分道线0.5 m范围内为报警区,车道中间2.75 m内为安全区。

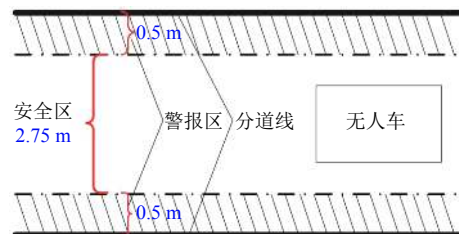


图10 车道安全区域与报警区域的划分

2.4 车道偏离试验

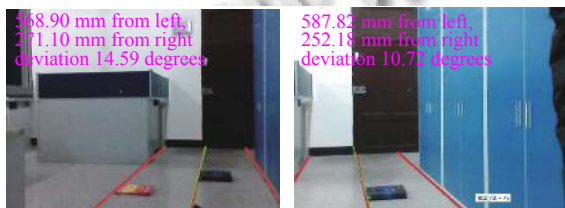
如图11所示为两帧车道偏离检测结果,其中第1行输出为车辆左轮与左分道线的间距和右轮与右分道线的间距;第2行输出为车辆行驶方向与车道中线

的偏航角,当偏航角小于5度时表示行驶方向正常.第3行表示车辆当前的运行状态,只有当偏航角、无人车与左右分道线的间距不超过设定的安全阈值时才会断定为正常行驶.



图 11 车道偏离检测结果

如图 12 所示,本文实验模拟的车道宽度为 120 cm,车辆宽度为 36 cm,相机投影在地面到图像底边的实际地面边界的距离为 205 cm,其高为 70 cm,模拟得到的结果为表 2 和表 3.



(a) 偏航角向左模拟图 (b) 偏航角向右模拟图

图 12 模拟偏航试验图

表 2 无人车相对于车道的偏航角(度)

次数	实际值	测量值	绝对误差	偏向
1	0	2.1	2.1	正常
2	1	3.38	2.38	正常
3	2	4.64	2.64	正常
4	4	2.56	1.44	正常
5	16	14.01	1.99	向左
6	0	0.33	0.33	正常
7	13	10.71	2.29	向右

表 3 无人车与左右分道线距离

次数	左间距	左测量值(cm)	左误差	右测量值(cm)	右间距	右误差
1	60	63.17	3.17	20.8	20	0.8
2	0	1.75	1.7	83.45	80	3.45
3	60	62.05	2.05	21.94	20	1.94
5	40	56.89	0.86	27.11	40	3.14
6	40	40.64	0.64	43.36	40	3.36
7	20	24.57	4.57	59.43	60	0.57

3 算法在 DSP-DM3730 上的移植

TI 公司的 DM3730^[10,11]整合了具有控制优势的 ARM 核和具有运算优势的 DSP 核. DSP 端依靠

DSP/BIOS 操作系统来支持音视频算法, ARM 端依靠 Linux 系统来控制芯片的外部设备,而 ARM 与 DSP 双核之间的数据传输与通信则依赖 Codec Engine 架构来管理^[12]. ARM+DSP 框架如图 13 所示.

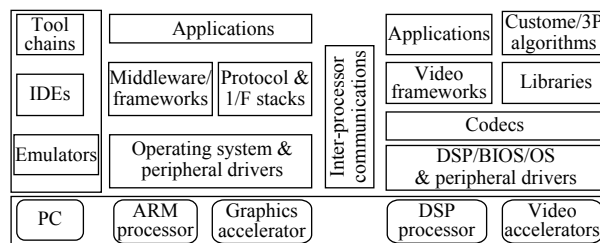


图 13 DSP+ARM 双核框架

3.1 硬件资源

(1) 处理器: DM3730 处理器.

① Pin-to-Pin 兼容 DM3730 系列处理器.

② 1000-MHz 主频 ARM CortexTM-A8 内核.

③ 800-MHz DSP TMS320C64x+TM 内核.

④ 片内集成存储器用于 ARM CPU(16 KB I-Cache, 16 KB D-Cache, 256 KB L2 Cache) 和片上存储(64 KB SHared SDRAM, 112 KB ROM).

(2) 存储器: Micron mDDR 与 NandFlash 集成芯片.

① 256 MByte 32 位 mDDR 133 MHz.

② 512 MByte 16 位 Nand Flash.

(3) 视频接口、USB 接口、音频插座、调试接口、SD/MMC 插座、电源接口、串口插座、扩展接口、网络接口.

3.2 软件资源

搭建软件开发平台的资源有:VirtualBox; uBuntu V10.04; PC 端的超级终端 Tera Term Pro.

安装 TI 提供的支持音频和视频算法的 DVSDK 组件包,该包含有各算法支持库和各支持库的调用管理引擎(Code Engine),以及 XDM 算法的编译工具 xdctools,同时提供了音频、视频算法进行编解码的例程以供参考.利用该包可以很方便地开发 DSP 下的图像语音处理软件,并且实现 ARM 与 DSP 之间的联系.Codec Engine 是 DVSDK 组件包的核心模块,是 ARM 和 DSP 的通信与数据传输桥梁,过渡于 ARM 应用层和 DSP 信号处理层之间.该模块要与 DSP 端建立的 Codec Engine 服务器进行通信需要使用 DSP/BIOS、DSP Link、DSP 算法接口的标准 xDIAS^[13]和 XDM 协

议. 为了发挥 DM3730 的双核作用, Codec Engine、DSP Link、xDIAS 和 XDM 是软件中不可缺少的部分.

DM3730 开发板可以从板载的 NandFlash 中启动, 也可以从 SD 卡的 Fat32 分区中启动. 无论是从板载的 NandFlash 中启动还是从 SD 卡启动, 都需要用 XLoader 对开发板进行一些初始化工作, 接着从 NandFlash/SD 中读取 Uboot 程序到存储器, 然后 Uboot 从 NandFlash/SD 卡读入启动参数, 加载 Linux 内核 uImage 到存储器, 解压缩后运行, 直到 Linux 内核启动后, 重新初始化 DM3730 板, 加载 NandFlash/SD 卡上的 FileSystem, 执行 FileSystem 中的程序并启动控制台. 由此可见, 开发板上的 Xloader、Uboot、uImage 和文件系统在整个启动过程是层层依赖的关系.

3.3 搭建交叉编译环境

安装 DVSDK 组件包后有自带的交叉编译器—arm-arago-linux-gnueabi-gcc. 只需将 DVSDK 组件包中的交叉编译器的安装路径添加在用户目录中的 .bashrc 文件中即可.

3.4 搭建网络文件传输服务 (TFTP)

TFTP 网络能将编译的程序和数据烧写到目标板上运行, 或将目标板上文件或者运行结果上传到 Linux 主机端. tftp 工作目录在 /tftpboot, 在文件传输过程中, 均需要先将文件拷贝到该目录下, 然后才能在超级终端 Tera Term Pro 中使用以下命令实现 uBuntu 和目标板中文件的传输:

- (1) tftp -g -r fileName uBuntu 的 IP 地址
- (2) tftp -p -l fileName uBuntu 的 IP 地址
- (3) g: get, -r: remote, -p: put, -l: local

3.5 移植 OpenCV 和 EMCV 库

OpenCV 开源的视觉库提供了丰富的图像处理算法, 在各大操作系统上均可编译通过生成相应的执行文件, 因此, 这些代码只要经过适当修改和编译便可移植在嵌入式系统中.

EMCV 库基于 C 语言实现, 能运行于 TI DM64x 系列 DSP 中, 移植 EMCV 相当于把 OpenCV 中少部分算法移植到 DSP 中^[14].

在 DM3730 的 ARM 应用程序中调用交叉编译后的 OpenCV 接口函数, 而在 DSP 端调用 EMCV 接口函数. 适用 cmake 交叉编译 OpenCV 库之前, 需要编译安装 ffmpeg 资源包、libx264 和 libxvid 两个库、汇编

编译器 yasm, 以及 libpng 和 libjpeg 库, OpenCV 才能读取和显示 png 和 jpeg 格式图片, 而 libpng 库的安装依赖于 zlib 库.

所有库的安装目录均设置在交叉编译的目录中, 且将编译安装后的生成库通过 tftp 网络传送到目标板的 lib 目录上. 至此, 就可以实现在 DM3730 目标板的 ARM 端调用 OpenCV 库函数了.

本文使用 Qt 开发图形界面对处理后的视频在 LCD 上显示. 先在 Qt 中的项目文件 .pro 中添加 OpenCV 的头文件路径和库文件. 编译后生成的二进制文件传送到目标板的文件系统上就可以在 ARM 端调用 OpenCV 函数接口, 处理后的结果使用 Qt 来显示.

使用 Codec Engine 作为 ARM 和 DSP 的连接桥梁, 模仿 Codec Engine 自带的 video_copy 例程来添加算法. 针对 EMCV 的移植, 将 EMCV 的文件添加到 video_copy 中, 并在配置文件中添加 EMCV 的文件名. 车道识别和车辆检测算法移植后, 进而测试本文的车道偏离检测和车距安全检测算法, 得到的效果良好.

3.6 算法的移植

TI 的 DVSDK 组件包提供了 Codec Engine 软件模块来实现 ARM 和 DSP 的通信和协同工作. Codec Engine 是一组应用程序编程接口, 用于实例化和运行 xDAIS 算法, 还有 VISA (Video, Image, Speech, Audio) 接口, 用于与 xDM 兼容的 xDAIS 算法进行交互^[15,16], 使 ARM 端的 Linux 可以调用 VISA 标准接口来管理 ARM 与 DSP 的软件. 如图 14 所示, Codec Engine 框架中, Server 集成 Codec 端算法, DSPLink 模块隐藏了 APP 调用 Server 端算法的整个过程, 因此在实例应用中只能看到 Server 和 APP 端.

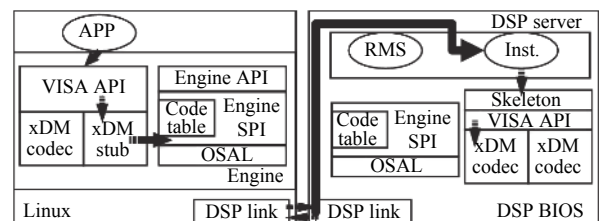


图 14 ARM 与 DSP 的通信框架

如图 15 所示, ARM 应用程序调用 Codec Engine 的 VISA API 接口, VIDENC_process(a, b, c) 传递 a, b, c 参数, Codec Engine 的 stub 会把参数 a, b, c 以及 VISA 接口信息封装处理, 利用消息队列 msgq 传递给

DSP 端. DSP 端的 skeleton 解开这个封装包, 把 ARM 端的参数 a, b, c (虚拟地址) 转换成 DSP 端的参数 $x, y,$

z (物理地址). DSP 端的处理 process 中的请求即能实现 ARM 端 VIDENC_process(a, b, c) 函数的操作^[17,18].

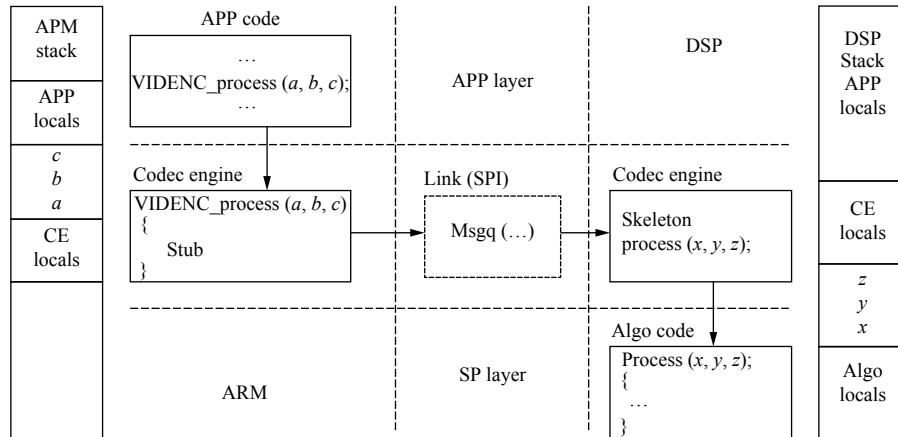


图 15 ARM 和 DSP 通信的函数传递

3.7 实验结果

图 16 为本文算法移植在 DM3730 平台上的运行结果. 已知采集高速车道视频的相机内外参, 使用本文的车距模型和车道偏离模型, 计算的车距和车道偏离角度均显示在图片上. 车辆正常行驶拍摄到的视频得到的检测结果良好.



图 16 本文算法在 DM3730 平台上的运行结果图

4 结论与展望

本文的车距检测模型^[19], 不但能够检测前方车辆与无人车的距离, 还能计算出前方车辆相对于相机光轴的偏转角度. 基于 CCP 车道偏离算法, 建立车道偏离模型, 实现了无人车当前位置和相对于分道线的偏航角的计算, 并设定无人车在车道上的安全区和警报区实现车道偏离的及时报警. 对于车距的建模, 在实验室模拟车道计算路面上的随机点, 得到的误差较小, 稳定性较好. 对于车道偏离检测, 模拟高速路车道搭建车道偏离模型, 计算车辆偏离车道的偏航角且车辆与左右分道线的距离, 检测误差范围小, 准确性较高. 还使用运行状况良好的高速路视频测试本文算法, 偏离反馈均显示正常. 尽管本文采用先进的 ARM+DSP 框架实现车辆动态图像识别的系统设计, 且采用 Codec Engine 框架实现 ARM 和 DSP 的通信, 实验过程稳定性较高, 但是在嵌入式平台内存和运行主频有限且运行算法尚未优化的情况下难以实现算法的实时性处理, 因此有待完善运行平台, 更好地协调车道检测、车辆检测与跟踪、车道偏离检测和车距检测 4 个模块. 有待进一步使用本文的算法在路面上测试.

参考文献

- 1 韦庭. 基于单目视觉的辅助驾驶系统中的图像处理研究 [硕士学位论文]. 成都: 电子科技大学, 2012.
- 2 Risack R, Mohler N, Enkelmann W. A video-based lane keeping assistant. Proceedings of the IEEE Intelligent

- Vehicles Symposium 2000. Dearborn, MI, USA. 2002. 356–361.
- 3 Lee S, Kwon W, Lee JW. A vision based lane departure warning system. Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients. Kyongju, Republic of Korea. 1999. 160–165.
- 4 戴斌, 裘伟. 逆透视投影下车道偏离时间的在线估计. 计算机工程与应用, 2007, 43(21): 235–238. [doi: [10.3321/j.issn:1002-8331.2007.21.071](https://doi.org/10.3321/j.issn:1002-8331.2007.21.071)]
- 5 李欣. 基于单目视觉的车辆前方障碍物测距[硕士学位论文]. 西安: 西安建筑科技大学, 2013.
- 6 佟卓远. 基于机器视觉的前方车辆检测与测距系统设计[硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2015.
- 7 Bao DS, Wang PK. Vehicle distance detection based on monocular vision. 2016 International Conference on Progress in Informatics and Computing. Shanghai, China. 2017.
- 8 Zhang Z. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11): 1330–1334. [doi: [10.1109/34.888718](https://doi.org/10.1109/34.888718)]
- 9 余太武. 基于机器视觉的车道偏离预警研究[硕士学位论文]. 昆明: 昆明理工大学, 2013.
- 10 Texas Instruments: SPRS685D. DM3730, DM3725 Digital Media Processors.
- 11 朱银忠. 基于 DM3730 的人脸识别系统设计[硕士学位论文]. 苏州: 苏州大学, 2012.
- 12 Texas Instruments: SPRUE67D. Codec Engine Application Developer User's Guide.
- 13 Texas Instruments: SPRAAE7B. Using Adapters to Run Existing xDAIS Algorithms with Codec Engine.
- 14 陈存弟, 刘金清, 刘引, 等. 基于 DM642 的纹理检测与 Adaboost 分类器相结合的车牌定位. 计算机系统应用, 2017, 26(7): 56–64. [doi: [10.15888/j.cnki.csa.005853](https://doi.org/10.15888/j.cnki.csa.005853)]
- 15 吴厚源. 基于 DM3730 的疲劳驾驶预警系统[硕士学位论文]. 厦门: 厦门大学, 2014.
- 16 Texas Instruments. TMS320DM3730 Software Developers Guide.
- 17 林上升, 韩润萍. 基于 OMAP3530 硬件平台的 ARM 和 DSP 协同开发方法. 电子技术应用, 2013, 39(2): 6–8, 11. [doi: [10.3969/j.issn.0258-7998.2013.02.001](https://doi.org/10.3969/j.issn.0258-7998.2013.02.001)]
- 18 李阳. 3G 移动定位与视频监控系统的设计与实现[硕士学位论文]. 哈尔滨: 哈尔滨工程大学, 2014.
- 19 陈存弟. 基于 DSP 的智能交通车辆动态图像识别[硕士学位论文]. 福州: 福建师范大学, 2019.