

基于逆向投影的全景泊车系统设计与实现

赵三峰, 谢 明, 陈玉明

ZHAO Sanfeng, XIE Ming, CHEN Yuming

南京工业大学 电气工程与控制科学学院, 南京 211816

College of Electrical Engineering and Control Science, Nanjing Technology University, Nanjing 211816, China

ZHAO Sanfeng, XIE Ming, CHEN Yuming. Design and implementation of around view parking system based on reverse projection algorithm. Computer Engineering and Applications, 2017, 53(23): 267-270.

Abstract: The around view parking system uses 4 wide angle cameras installed in the car's front, back, left and right to collect the images in real time, which are transmitted to the image processing equipment for correction, projection transformation and image composition on the screen to form a 360 degree panoramic view. The system uses processor HI3520D and video decoder NVP1918C as the core of hardware, and uses the fish-eye correction algorithm, projection transformation and reverse projection algorithm as the software core. The experimental results show that the around view parking system can achieve seamless and clear images in real-time display.

Key words: around view parking; HI3520D; NVP1918C; reverse projection

摘 要: 全景泊车系统利用安装在车身前后左右的4个广角摄像头实时采集周围的图像,并传送给图像处理设备进行校正、俯视变换和图像合成,最后在显示屏上形成一幅360°的全景俯视图。系统以处理器HI3520D和视频解码器NVP1918C为硬件核心,以鱼眼校正、投影变换和逆向投影算法为软件核心,最终实现实时显示无缝和清晰画面的360°全景泊车影像系统。

关键词: 全景泊车; HI3520D; NVP1918C; 逆向投影

文献标志码: A **中图分类号:** TP391.4 **doi:** 10.3778/j.issn.1002-8331.1605-0373

泊车辅助系统主要分为两大类:一类是基于传感器,另一类是基于视觉。目前使用比较广泛的是基于视觉的泊车辅助系统,而360°全景视觉泊车辅助系统是目前最先进的辅助系统。全景视觉泊车辅助系统从根本上消除了车辆周围的视觉盲点,帮助驾驶员更为直观地停泊汽车。本系统的原理为:通过在车身前、后、左、右安装四个广角摄像头,由于相机成像过程中存在非线性畸变^[1-2],因此需要进行畸变矫正^[3-5],对畸变矫正后的图像进行标定得到投影矩阵,转换图像视角为俯视图,最后利用逆向投影算法合成一幅360°全景图像。

逆向投影算法是全景泊车系统的核心,通过逆向投影合成一幅无缝的、高清晰的图像。目前大多算法是基于图像拼接技术^[6-12],这种算法的计算过程过于复杂,拼接过程也十分困难,远没有逆向投影简单有效。其原理如图1所示。

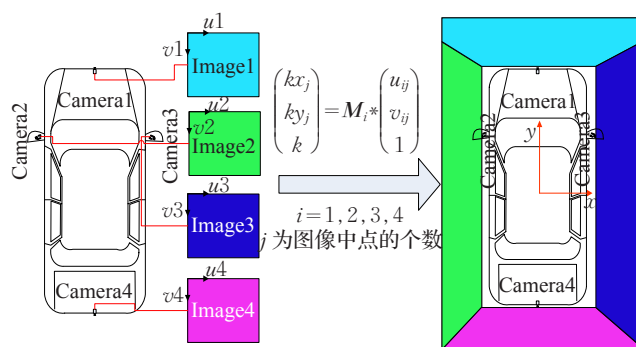


图1 全景泊车系统基本原理

1 系统方案与指标

系统主要由三部分组成:4个水平视角为175°的鱼眼摄像头,一块图像采集卡,一个7英寸的TFT显示屏。系统信息处理流程如图2所示。

基金项目: 2015年江苏省六大人才高峰项目(No.XXRJ-012)。

作者简介: 赵三峰,男,硕士,研究方向:机器视觉;谢明,男,博士生导师,研究方向:机器视觉;陈玉明,男,讲师,研究方向:机器视觉。

收稿日期: 2016-05-26 **修回日期:** 2016-11-10 **文章编号:** 1002-8331(2017)23-0267-04

CNKI网络优先出版: 2016-11-21, <http://www.cnki.net/kcms/detail/11.2127.TP.20161121.2040.054.html>

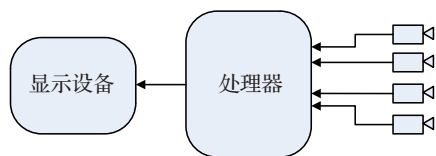


图2 系统信息处理流程

系统的性能指标如表1所示。

表1 系统的性能指标

组成部分	功能说明	技术指标
鱼眼摄像头	实时采集车身周围影像	12 V 供电; CVBS 格式的 NTSC 制式模拟视频信号
图像采集卡	采集图像, 输出合成图像	12 V 供电; 4 路 CVBS 格式的 NTSC 制式模拟视频信号; 视频接口均为标准 AV 接口; 1 路 VGA 接口
TFT 显示屏	显示合成图像	12 V 供电; CVBS 格式的模拟视频信号输入; 标准 AV 端子接口

2 硬件系统的设计与实现

Hi3520D 是一款基于 ARM Cortex A9 处理器内核以及视频硬件加速引擎的高性能通信媒体处理器, 工作频率达到 660 MHz, 具有 32 bit、128 MB×16 的 DDR3 SDRAM 接口, 片上集成了视频处理子系统 VPSS。主板主要由视频采集模块、外围存储模块、视频输出接口、通信接口和电源模块 5 部分。其结构如图 3 所示。

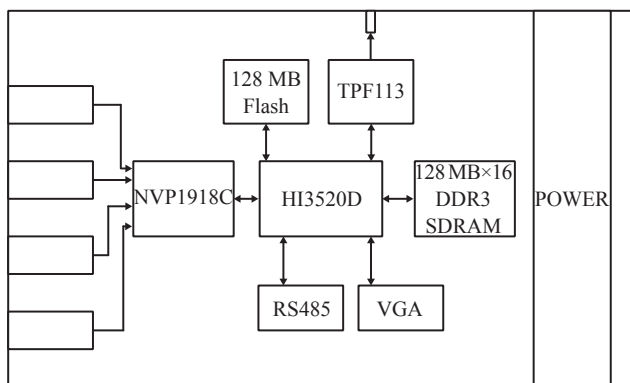


图3 主板结构示意图

(1) 视频采集模块: 视频采集模块将 4 路 175° 鱼眼摄像头采集到的模拟信号通过标准 AV 接口输入到 4 路视频编码芯片 NVP1918C 中, NVP1918C 对模拟信号进行 AD 转换后以 BT.656 格式传输到 HI3520D 中, HI3520D 再对数据进行编码, 编码格式为 H.264。

(2) 外围存储模块: 板上采用三星公司的 1 片 K4B2G1646Q 芯片扩展 2 GB 容量的 DDR3 SDRAM, 用于存放运行过程中的数据和程序, 使用 1 片 Winbond 公司的 25Q128FVTM 扩展 128 MB Flash 用于存放固定程序和数据。

(3) 视频输出接口: HI3520D 通过芯片内部集成的视频处理子系统 VPSS 将处理后的图像经 DA 变换形成 CVBS 的模拟视频信号, 然后通过视频滤波放大器

TPF113 输出到标准 AV 接口。

(4) 通信接口: 可以通过 RS485 和以太网与 PC 通信。

(5) 电源模块: 板子的输入电压是 12 V 电源供电, 使用一片开关稳压芯片 AMS1117, 为各芯片提供所需的 +1.5 V、+1.2 V、+3.3 V, 其最大输出电流为 800 mA。

3 软件构架

本文使用 VS2013 开发环境, MFC 构建可视化界面, opencv2.4.11 和雄迈公司的 NETSDK 作为外部库函数, 建立系统的软件构架。系统软件构成如图 4 所示。

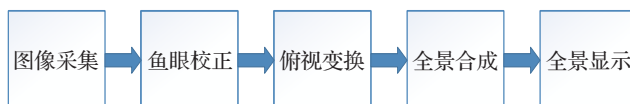


图4 系统软件构成

4 算法流程

4 路摄像头分别采集 4 个方向的 4 张图片, 然后对每个方向的图片进行校正, 俯视变换, 最后将 4 张图片逆向投影到合成区域, 形成全景图像并显示。

4.1 鱼眼校正

通过研究普通摄像头和鱼眼摄像头的成像原理^[13], 推理出鱼眼摄像头的校正公式, 图 5 和图 6 分别是普通摄像头和鱼眼摄像头的投影。

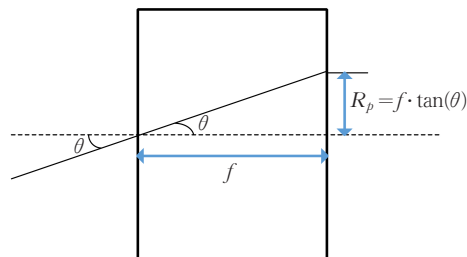


图5 普通摄像头投影

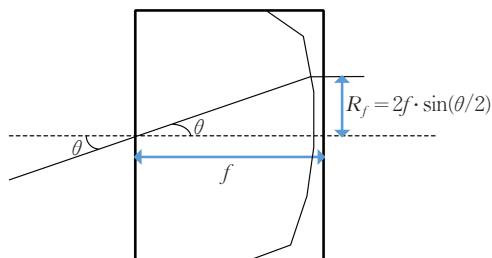


图6 鱼眼摄像头投影

图 5、6 中 f 表示摄像头的焦距, R 表示像素点在成像平面的径向位置。该算法推倒公式来自文献[1], 下面直接写出最终的结果:

$$\begin{cases} x_f = \frac{2x_p \cdot \sin(\arctan(\lambda/2))}{\lambda} \\ y_f = \frac{2y_p \cdot \sin(\arctan(\lambda/2))}{\lambda} \\ \lambda = (\sqrt{x_p^2 + y_p^2})/f \end{cases} \quad (1)$$

其中, (x_p, y_p) 表示校正图像的坐标, (x_f, y_f) 表示鱼眼图像的坐标。

此算法的效果如图 7 所示。

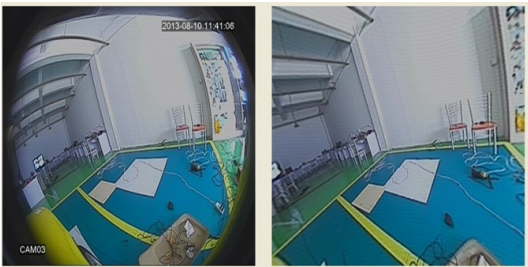


图 7 校正前后图像

4.2 俯视变换

本文采用直接线性变换(DLT)^[14-15]来找到俯视变换的投影矩阵, 这种方法的优点在于不需要知道摄像头视角等参数, 只需要在图像坐标系下标定 4 对特征点就可以计算出 8 个未知的参数, 从而得到单应性矩阵, 并利用单应性矩阵完成俯视变换。

直接线性变换的公式如下:

$$u_i = \frac{c_{00}x_i + c_{01}y_i + c_{02}z_i + c_{03}}{c_{20}x_i + c_{21}y_i + c_{22}z_i + 1} \tag{2}$$

$$v_i = \frac{c_{10}x_i + c_{11}y_i + c_{12}z_i + c_{13}}{c_{20}x_i + c_{21}y_i + c_{22}z_i + 1} \tag{3}$$

其中, (u_i, v_i) 表示图像坐标, (x_i, y_i, z_i) 为物体空间坐标, $c_{00}, c_{01}, \dots, c_{22}$ 为未知参数。但是本文的物体选择的是路面特征点, 因此公式的 $z=0$, 简化后的二维 DLT 公式为:

$$u_i = \frac{c_{00}x_i + c_{01}y_i + c_{03}}{c_{20}x_i + c_{21}y_i + 1} \tag{4}$$

$$v_i = \frac{c_{10}x_i + c_{11}y_i + c_{13}}{c_{20}x_i + c_{21}y_i + 1} \tag{5}$$

如果用矩阵的形式表示, 如下:

$$ROI = \begin{pmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & 1 \end{pmatrix} \tag{6}$$

ROI 表示路面坐标到图像左边的变换矩阵, 则 ROI^{-1} 表示图像到路面的投影矩阵。

4.3 图像合成

本文的创新点就在于跳出了传统图像拼接的思维, 采用一种更加简单有效的算法来实现无缝拼接 360° 全景。

假设路面合成区域的大小为长 R_L , 宽为 R_w , 单位: mm。 R_w 的宽度方向平均分成 M_{out} 等份, R_L 的高度方向平均分成 N_{out} 等份, 其中, 任意一点坐标用 (i_{out}, j_{out}) 表示。如图 8 所示。

图中蓝色区域表示汽车的位置。设汽车中心为路面的原点, 则路面上任意一点 (i_{out}, j_{out}) 的坐标为 (x, y) , 那么它们的关系是:

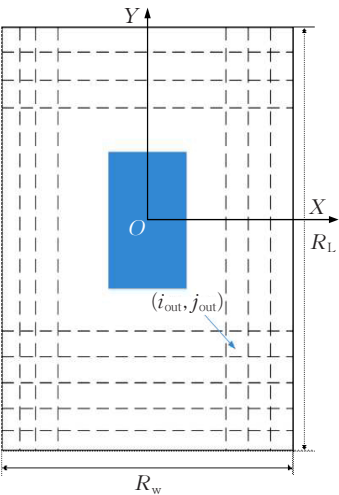


图 8 离散化后的路面

$$\begin{cases} x = \frac{i_{out}}{M_{out}} \cdot R_w - \frac{R_w}{2} \\ y = \frac{j_{out}}{N_{out}} \cdot R_L - \frac{R_L}{2} \end{cases} \tag{7}$$

其中, M_{out} 表示合成图像的宽度, 单位: 像素; N_{out} 表示合成图像的高度, 单位: 像素; R_w 表示合成区域的宽度, 单位: mm; R_L 表示合成区域的长度, 单位: mm; (i_{out}, j_{out}) 表示图像的坐标, 单位: 像素; (x, y) 表示合成区域路面的坐标, 单位: mm。

下面判断路面上的点被哪个摄像头拍到, 因此将路面分成八个区域, 如图 9 所示。

I (左前)	II (前)	III (右前)
IV (左)	汽车	V (右)
VI (左后)	VII (后)	VIII (右后)

图 9 路面的八个区域

图 9 中, II、IV、V 和 VII 四个区域只能被前, 左, 右, 后四个摄像头看到。 I、III、VI 和 VIII 为两个摄像头的交叉区域, 可能被两个摄像头看到, 因此需要判断交叉区域被哪个摄像头采集到, 四个交叉区域的判别方法相同, 因此以区域 I 为例描述如下:

(1) 取区域 I 内任意一点 (i_{out}, j_{out}) , 计算其路面坐标 (x, y) 。

(2) 假定该点可以被前摄像头采集到, 通过与前摄像头的投影矩阵 ROI_{front} 相乘, 便可以得到该点在前摄像头的像素坐标 (u, v) , 如果 $0 \leq u \leq u_{max}$, 且 $0 \leq v \leq v_{max}$, 则该点可以被前摄像头采集到, 令该点标记 ID_{front} 为 1。同理, 如果也可以在左摄像头被采集到, 令该点标记 ID_{left} 为 1。

(3) 如果 $ID_{front} = 1, ID_{left} = 0$, 取该点在前摄像头像

素的色值,并填充到 (i_{out}, j_{out}) 所在的合成区域;如果 $ID_{front} = 0, ID_{left} = 1$, 取该点在左摄像头像素的色值,并填充到 (i_{out}, j_{out}) 所在的合成区域;如果 $ID_{front} = 1, ID_{left} = 1$, 取该点在前和左摄像头的色值的均值,并填充到 (i_{out}, j_{out}) 所在的合成区域。

当合成区域所有像素点的色值都被对应图像色值填充以后,合成图像就完成了。当然如果出现色值为0的点,利用最近邻准则填充。

5 系统测试与效果

以童车作为实验平台,分别在童车的四周安装摄像头,根据实验表明,前后摄像头的安装角度与垂直地面小于 45° 才能达到最好的效果,左右摄像头尽量垂直地面安装。为了避免相机抖动引起合成效果扭曲,安装时应尽量固定好摄像机。童车安装摄像头如图10所示。



图10 摄像头安装

实验效果如图11所示。

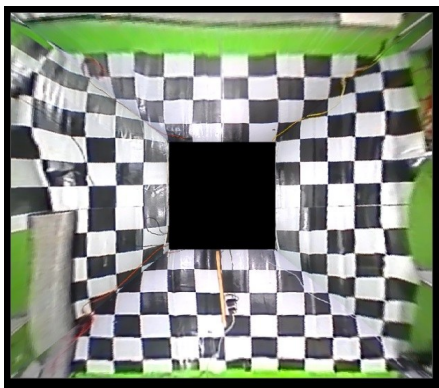


图11 合成效果预览

观察效果图,全景图像基本还原了设定的合成区域图像,这里特意在两个相连的区域放上一根绳子,验证合成区域的交界处是否有缝隙,结果绳子在交界处没有发生断裂,达到了预期的效果。

6 结束语

基于逆向投影的全景合成算法与传统的图像拼接

相比,在算法复杂度上大大减少,更加容易应用到实际当中去。通过观察合成效果,虽然在较远处有些图像畸变,但是近处的效果已经符合设计要求。实验过程中遇到的各种问题,通过调试已得到修正。但是还是有不足之处,在标定的过程中,本实验是通过人工手动标定特征点的,因此会有人为因素的误差,这个会导致合成区域的交界处会有缝隙或者偏差,在接下来的研究中,将改进标定的方法,采用自标定的方式来克服这种人为误差。

参考文献:

- [1] 朱铮涛,黎绍发.镜头畸变及其校正技术[J].光学技术,2005,31(1):136-138.
- [2] 贾洪涛,朱元昌.摄像机图像畸变纠正技术[J].电子测量与仪器学报,2005,19(3):46-49.
- [3] 张伟.鱼眼图像校正算法研究[D].南京:南京邮电大学,2011.
- [4] 邢立新.鱼眼镜头视频图像实时校正技术研究[D].哈尔滨:黑龙江大学,2014.
- [5] Dhane P, Kutty K.A generic non-linear method for fish eye correction[J].International Journal of Computer Applications,2012,51(10):58-65.
- [6] 周猛.基于DaVinciTM的360°全景泊车影像系统的设计与实现[J].电子技术应用,2014,40(10):20-22.
- [7] 张聪.基于鱼眼镜头的车载全景环视系统[D].杭州:浙江大学,2015.
- [8] 廖国品.全景视觉泊车辅助系统算法研究[D].广州:华南理工大学,2013.
- [9] 方财义.全景泊车系统及图像拼接技术的研究[J].汽车零部件,2012(11):103-105.
- [10] 王娟.图像拼接技术综述[J].计算机应用研究,2008(7):26-29.
- [11] 周圣文.基于SIFT算法的全景泊车辅助系统研究[D].安徽芜湖:安徽工程大学,2015.
- [12] Wu Z.Expansion of the visual angle of a car rear-view image via an image mosaic algorithm[J].Optical Engineering,2015,54(5).
- [13] Altera Corporation.A flexible architecture for fisheye correction in automotive rear-view cameras[R].White Paper,2008.
- [14] 王东红,罗均.摄像机标定中的直接线性变换法[J].机械与电子,2006(9):9-11.
- [15] 葛宝臻,李晓洁.基于共面点直接线性变换的摄像机畸变校正[J].中国激光,2010(2):488-494.