汽车全景图像主动安全系统的设计与实现

重庆大学硕士学位论文

（专业学位）

学生姓名：杨钰超

指导教师：谢昭莉 副教授

兼职导师：汤朝明 高级工程师

学位类别：工程硕士（控制工程领域）

重庆大学自动化学院

二〇一五年四月

Design and Implementation on Vehicle Active Safety Systems with Panoramic Image

A Thesis Submitted to Chongqing University
in Partial Fulfillment of the Requirement for
Professional Degree

By

Yang Yuchao

Supervisor: Associate Prof. Xie Zhaoli

Pluralistic Supervisor: Senior Eng.Tang Chaoming

Specialty: ME (Control Engineering Field)

College of Automation of Chongqing University,

Chongqing, China

April 2015

摘　　要

汽车存在着视线盲区，这些视线盲区容易使驾驶员无法及时掌握道路情况，从而引发交通事故。论文利用机器视觉构建汽车周边环境的全景图像，解决驾驶员的视线盲区问题，并对进入汽车安全距离范围内的障碍物发出声音报警，从而增强汽车在行驶过程中的主动安全性，对汽车的行驶安全有着重要的实际意义。

论文设计了一种汽车全景图像主动安全系统，该系统能够使用设置在车身前后左右四个方向的高清广角摄像头，采集汽车四周环境的画面，并将图像转换为俯视图，然后采用图像拼接算法将四幅俯视图拼接为全景图像。

论文针对系统采集汽车四周环境画面的需求，确定了以四个广角摄像头安装在汽车前后左右四个方向作为图像采集方式。针对系统需要计算能力强、图像处理速度快、稳定性好、便捷等车载要求，确定了以TI公司的DSP+ARM双核芯片TMS320DM6446作为核心芯片；针对系统需要及时处理4路摄像头图像信号的需求，确定了以TVP5158作为视频解码芯片的ARM＋DSP嵌入式系统方案。

根据系统需求，设计了采用ARM处理器作为程序管理的嵌入式Linux全景图像主动安全系统，包括系统硬件电路设计、软件开发平台搭建。硬件电路设计由电源保护电路、电压转换电路、TVP5158 外围电路等组成。电源保护电路主要解决在车载条件下，为核心板和摄像头提供稳定电源；电压转换电路主要用于为核心板上各个芯片提供不同的内核电压以及 I/O 口电压；TVP5158 外围电路主要包括4路视频信号输入输出电路以及芯片外围引脚电压电路。软件开发以TI公司的达芬奇技术为基础，在虚拟机Linux系统上搭建了DM6446的ARM编译环境，主要包括安装MontaVista Linux内核、安装DVSDK工具链，设置交叉编译环境等。软件开发过程中主要通过NFS实现文件系统的挂靠，从而避免开发过程中频繁的烧写。

结合所设计的嵌入式系统平台，在DM6446的DSP端上开发基于特征的障碍物检测算法和基于坐标系转换的全景图像拼接算法。在软件开发的实验平台上，通过实际的测试实验，验证了所设计的硬件系统能够同时采集和处理 4 路摄像头信号，满足设计需求。同时，通过实际功能测试实验，验证了障碍物检测算法和全景图像拼接算法切实有效。最后，通过系统整体实车实验，验证了论文的研究成果。

关键词：主动安全系统，全景图像，嵌入式Linux，图像拼接

ABSTRACT

There are many blind sight areas around the vehicle, which will cause drivers frequently unable to get the road conditions. As a result, it will lead to traffic accidents. Paper builds panoramic image around the vehicle by machine vision to get rid of blind sight areas. Paper also designs an obstacle detection device system that can warn driver dangerous vehicles which are too close to it by sound alarm. In this way, they will improve active safety of the running vehicle, which is so important to us.

Paper designs a vehicle active safety system with panoramic image. This system consists of four wide-angle cameras set on the vehicle body which are in four directions. They grab image around the vehicle. After that, image is transformed to top view of the vehicle. In the end, the four channels of image are stitched to a panoramic image from top view.

Paper determines the image acquisition mode with four wide-angle cameras set on the vehicle body which are in four directions aiming at getting the surrounding video of the vehicle. Paper also determines the dual-core chip named TMS320DM6446 which is developed by TI company as the core chip. This chip has strong computing capability, high speed of image processing, good stability and convenience in vehicle. Paper determines the video decoding chip named TVP5158 to meet the need for processing 4 cameras image signal in the same time. The ARM and DSP embedded system solutions consist of TVP5158 and TMS320DM6446 .

Paper designs embedded Linux panoramic images of active safety systems by using ARM processor as the program manager to meet the system needs. Paper designs the system hardware circuits and builds the software development platform. The system hardware circuits consist of power supply protection circuits, the voltage conversion circuits, TVP5158 peripheral circuits and so on. Power supply protection circuits aim at providing stable power supply for the core plate and camera. The voltage conversion circuits are mainly used to provide different voltages for each core and I/O port on chip. TVP5158 peripheral circuits include a 4-channel video signal input and output circuits and chip peripheral pins voltage circuits. Software development is based on TI's DaVinci technology, with building compiler environment on Linux virtual machine system for DM6446. It consists of installing the MontaVista kernel and DVSDK tool chains and setting cross-compiler environments. Software development process is

mainly realized through NFS file system, which is anchored to, to avoid the frequently programming.

Paper develops a feature-based obstacle detection algorithm and a panoramic image stitching algorithm based on coordinate transformation on the DM6446's DSP side by using the designed embedded system platform. Paper verifies the hardware system able to collect and process the 4-channel video signal at the same time through actual test experiments. It meets the design requirements. Meanwhile, obstacle detection algorithm and panoramic image stitching algorithm are effective by actual function tests. At last, paper verifies the research results by whole real car experiments.

Keywords: Active safety system, Panoramic image, Embedded Linux, Image stitching

目　录

1 绪　　论

1.1 课题研究的背景及意义

汽车作为现代文明的标志之一，已经融入了千家万户的生活。据《上海市综合交通 2011 年度报告》中显示，2010 年仅上海市机动车的持有量就达到 248.7 万辆[1]。同时，2011 年我国汽车市场呈现平稳增长态势，平均每月产销突破 150 万辆，全年汽车销售超过 1850 万辆，再次刷新全球历史纪录[2]。

汽车在给人们带来便捷的同时，也因为驾驶员操作失误、视野不足等原因夺去了众多人的生命。据世界卫生组织 2011 年报告，现在全球每年约有 130 万人死于交通事故，并有 2000 万人至 5000 万人遭受非致命伤害，平均每一分钟就有两人死于交通事故，十多人在交通事故中受伤。而在中国，每年大约有 21 万起道路交通事故，6 万余人死亡。据 2012 年 12 月中国疾病控制中心的监测数据显示，在各类伤害死因中道路交通伤害已经成为我国人群第一位伤害死因[3]。

汽车厂商为了避免交通事故的发生，提出了主动安全系统这一理念。为了预防汽车事故发生，避免人员受到伤害而采取的安全设计系统被称作主动安全系统。主动安全系统是针对原有的被动安全系统而提出来的，两者的结合能更好地避免交通事故的发生、减少事故的损害。主动安全系统和被动安全系统的划分依据是交通事故发生前后的不同作用。具体来说，主动安全系统是避免交通事故的发生，被动安全系统是避免交通事故中驾驶员和乘客的伤害发生。

从主动安全系统和被动安全系统的定义中可以看出，虽然被动安全系统能够避免交通事故中人员的伤害发生，但是，那是“事后”处理，是交通事故发生后不得已而为之的事情，是把“大事化小”的举措。主动安全系统则不同，它能避免交通事故的发生，将“小事化无”，避免交通事故造成的直接经济损失，此外，它还可以避免交通事故造成的其它间接经济损失，比如交通事故造成的道路堵塞、处理事故带来的误工误事等。鉴于两者在交通事故发生中的作用不同，汽车安全现已经由过去的以碰撞安全为核心的被动安全系统向以预防为核心的主动安全系统发展。

汽车主动安全系统能扩展驾驶员的感知范围，增强驾驶员的认知能力，这都得益于系统使用的传感器。汽车主动安全系统使用各种不同类型的传感器来检测自身及其周围的环境状况变化，传感器主要有摄像头、红外传感器、GPS 定位仪、光电码盘、激光雷达、超声波传感器、无源光束传感器等等。当环境状况变化发生后，如果达到预先设定的报警值，主动安全系统有两种处理方式：一种是会发出报警信号提醒驾驶员，让驾驶员做出修改。比如常见的车速报警装置，当汽车

行驶速度高于设定速度时，就将通过车速表内的报警开关联通蜂鸣器发出报警信号，提醒驾驶员已超速。另外一种是自动调整相关装置，使环境状况变回原来的正常值或者减轻环境状况变化，比如汽车的前大灯自动控制变光装置，当汽车进入隧道或者天变暗时，可通过光传感器检测到车前亮度变暗，从而自动打开车前大灯，保证驾驶员前方道路的明亮。在这两种主动安全系统的处理方式中，发出报警信号提醒驾驶员纠正错误驾驶是十分重要的，因为，虽然现在的汽车智能化程度越来越高，越来越多的设备也能自动执行，但是在目前的汽车控制中，绝大多数控制还是由驾驶员来直接操纵的，其对整个车辆系统的性能有着重要的影响。英国科学家研究就发现，虽然每一起交通事故的发生均在不同程度上受到驾驶员、汽车和道路环境的影响，但是，在所有的交通事故中，直接因为驾驶员操纵不当而引发的交通事故占总共事故的 65%。进一步的研究发现，算上与驾驶员操纵间接相关的交通事故，可占总共事故的 95%。同样的研究也在美国进行，他们得出类似的数据，直接与驾驶员操纵相关的事故占 57%，直接、间接与驾驶员相关的事故占总共事故的 94%[4~5]。中国的交通事故统计也得出了相同的结论，其与驾驶员相关的事故占到总共事故的 90%左右。驾驶员操纵不当和失效成为引发交通事故的主要原因已经被世界各国所公认[6~8]。

在驾驶员实际驾驶汽车过程中，有超过 90%的环境信息是来自于驾驶员的双眼，例如道路状况、道路交通标志、标线和信号、道路障碍物等等[9]。由于人自身双眼视线范围的限制，加上汽车是个半封闭壳体，驾驶员的视线受到了车体的阻碍，即使设计了后视镜来扩展驾驶员的视线范围，驾驶员仍然有大范围的视线盲区。视线盲区的存在和视线角度的影响，往往使得新手驾驶员手忙脚乱，特别是在复杂的路况，如倒车入库、车多路窄的情况时。视线盲区的存在，给汽车碰撞追尾事故的发生带来了一定的可能性。

解决这类问题的一个可行办法就是采用辅助的传感装置来扩大驾驶员的视线范围，减小甚至消除视线盲区。使用摄像头作为传感器，以视觉图像为信息的机器视觉就是一个扩大驾驶员视线范围的好办法。图像信息具有信息量丰富、信息采集非接触、信息采集覆盖面广、多个传感器同时工作、性价比高等优点，是其它传感器信息无法比拟的。

目前，一些汽车制造商已经意识到视线盲区带来的危害，已经开始着手采用多个摄像头捕捉图像信息来弥补驾驶员的视线不足，从而将传统的倒车后视镜系统升级为全景图像监控系统[10]。全景图像具有广义定义和狭义定义，狭义定义中指具有半球状视角的图像，广义定义中泛指利用多幅图像拼接成的大图像，而对于主动安全系统中的全景图像，主要定义为车身前后左右图像拼接而成的图像。全景图像具有信息量大，信息真实、可靠，亲历感强的特点。因此，为驾驶员提

供车辆周边的全景图像，辅助以障碍物检测与报警功能则能够彻底解决驾驶员的视线盲区问题，从而促使驾驶员更快反应，避免和减少车辆发生刮蹭、碰撞等交通事故。

1.2 国内外研究现状

1.2.1 全景图像在主动安全系统中的应用

全景图像是现在泊车辅助中的一个研究热点，很多汽车厂商已经在研发基于机器视觉的全景泊车辅助系统。全景泊车辅助系统的概念是由 K.Kato 等四人于 2006 年在《Image Synthesis Display Method and Apparatus for Vehicle Camera》中首次提出的[11]。

日本汽车厂商率先将全景图像运用在泊车辅助中。日产公司在 2007 年时，研发出一款全景泊车辅助系统，命名为 AVM(Around View Monitor)。该系统能够利用摄像头采集车身周围图像，并在车载显示器上显示拼接好的 360 度全景鸟瞰图。同时，该系统能够通过测距雷达来检测车身与障碍物之间的距离，并发出距离过近警报。2009 年时，本田公司也发布了带有类似系统的汽车。此外，富士通公司在 2013 年时发布了全球首款三维全景成像行车辅助系统。

英菲尼迪公司也独自开发了自己的 AVM 系统，并将其应用在了旗下的 EX35、FX 等车型上。该系统可将四个超广角摄像头所摄图像通过电脑编辑后形成 360 度全景，并显示在中控台显示屏上[12]。

博通、飞思卡尔、OmniVision 三家公司在 2013 年时联合推出了 360 度全景泊车辅助系统，该系统装载在最新型的宝马 X5 车上，这是全球首款基于以太网的停车辅助系统[13]。

德尔福公司研发了 360 度智能全景泊车引导系统，该系统可提供车辆周围 360 度的全景视角，用于协助泊车操作，该系统成品曾在 2013 年的上海车展上首次亮相[14]。

随着国内汽车数目的不断增长，国内的汽车电子商发现了全景泊车系统存在的重大商机，纷纷加入了进来。

深圳汉华安道科技有限责任公司在 2011 年开发了 360 度全景泊车辅助系统，该系统具有全景融合、无缝拼接和自动校准等功能，能解决泊车过程中盲区、死角问题[15]。

除此之外，国内的其它汽车电子厂商也推出了类似的产品，在此不再一一详细叙述。

对于全景图像在汽车主动安全上的研究方向，国内的相关学者也主要是将研究的重点放在了如何将全景图像应用于泊车辅助系统中[16~17]。这些文献设计的全

景图像泊车辅助系统运用的核心芯片各不相同，但主流为 DSP 芯片，比如利用 TMS320DM643 芯片实现了全景泊车辅助系统，该系统能在车辆导航仪上显示汽车周边 360 度的全景鸟瞰图，但是只能实现显示图片功能[16]。部分文献不只设计了全景泊车辅助系统这个单一系统，而是将全景泊车辅助系统与基于视觉的自动泊车系统相结合，能依照不同的泊车任务需求，手动选择两者，但是只显现了将汽车四周的四个摄像头采集的图像拼接为鸟瞰图的功能[17]。

1.2.2 全景图像拼接技术

为了得到汽车周围的全景图像，需要将安装在车上的多个摄像头采集的图像进行拼接。

图像拼接算法根据变化域的差异可以划分为两个大类：一类是基于频域的图像拼接算法，另一类是基于空间域的图像拼接算法。

基于频域的图像拼接算法的原理是将图像变换到频域后再做相关的处理，常见的有基于傅里叶梅林变换（FMT）的图像拼接算法[18]，Kuglin 和 Mines 提出的相位相关法[19]，Castro 和 Morandi 提出的扩展相位相关法[20]。

基于空间域的图像拼接算法目前热门的主要可以分为两类：一是基于区域，一是基于特征[16]。对于这两类方法各自的优缺点，各个文献中均有所谈及。

基于区域的拼接算法拼接成功率较高，但是其运算数据较多，同时占用 DSP 内存较大，不适合对运算速度要求快的系统。基于特征的拼接算法拥有比较高的鲁棒性。基于特征的图像匹配算法热门的主要有以下几种：Harris 等人提出的 Harris 角点检测算子[21]，Richard Szeliski 提出的基于 Levenberg-Marquardt 迭代算法及其改进算法的图像拼接算法[22]，Kyung 等人提出的基于水平移动模型的柱面全景图像拼接算法[23]，Brown 和 Lowe 等人提出的基于不变量技术的尺度不变特征变换算法（简称 SIFT 算法）[24~25]，Bay 等人提出的基于不变量技术的快速鲁棒性特征检测算法（简称 SURF 算法）[26]，以及其它很多以 SIFT 算法为基础的改进算法。

1.2.3 障碍物检测技术

基于单目视觉的障碍物检测算法可以分为两类，一类是基于特征，另一类是基于运动。基于特征的障碍物检测方法，是以障碍物的特征作为检测判断标准，对于道路交通中的障碍物来说，主要有两类特征明显的障碍物，一类是汽车障碍物，另一类是行人障碍物。对于汽车障碍物来说，其特征主要有对称性、车底阴影、垂直/水平边缘等等[27~29]；对于行人障碍物来说，其特征主要有形状、高度、纹理等等[30~33]。基于运动的障碍物检测方法主要可以分为三类，即运动补偿法、光流法和帧间差分法。帧间差分法对于背景图像的要求较高，即背景图像不能差别较大，同时，帧间差分法对检测目标也要求较高，对于运动的检测目标，其速度不能过快，否则，无法检测出来。因此，帧间差分法主要被应用于视频监控中。

运动补偿法可以利用已知或者是测得的摄像机的运动参数来补偿检测，从而达到障碍物检测的功能[34~37]。光流法可以使用多种不同的检测手段，目前文献中谈及的主要有以下几种：一种是可以利用实际图像的光流场同地平面光流场之间的区别来实现对障碍物的检测[38]，或者说可以利用 Canny 边缘算子的融合技术同光流场分割相结合来实现对障碍的检测[39]，也可以利用极线约束、高度约束和正向深度约束来实现对障碍物的检测[40]，也有学者利用基于特征点的道路面投影位移矢量来检测障碍物[41]。

综上所述，在全景图像应用方面，目前，国内外的研究者主要是将全景图像应用于泊车辅助系统中。该系统的首要目的是通过为驾驶员提供汽车四周的全景图像从而方便泊车，同时通过为驾驶员提供图像，一定程度上提高了驾驶员在泊车过程中的安全性，但是这种安全性能是一种被动的，需要驾驶员自我观察发现危险，如果系统能增加障碍物检测与报警功能，则能大大增强系统的实用性，减少驾驶员的操作负担，让驾驶员更专注于驾驶上。

1.3 论文的主要研究内容

论文主要是设计一种汽车全景图像主动安全系统，它能够通过安装在汽车前后左右的四个摄像头，采集汽车四周的画面，对图像进行汽车障碍物检测，当图像中出现汽车障碍物时，会判断该障碍物同本车的距离是否小于预先设定的安全距离，若小于安全距离，则向驾驶员发出声音报警信号；利用图像拼接技术将四幅图像拼接为全景图像，显示汽车四周环境，从而提高汽车的主动安全性能。

论文的主要研究内容如下：

①针对研究方向，做了详细的需求分析、功能分析和器件选型分析，确定了硬件设计方案和硬件器件。

②设计了基于嵌入式 Linux 系统的全景图像主动安全系统，包括硬件电路设计、软件开发平台的搭建。

③在实验平台和实车上研究并实现了基于特征的障碍物检测算法和基于坐标系转换的全景图像拼接算法。

④归纳总结了论文的工作，并对后续研究工作进行了展望。

1.4 论文的主要章节安排

论文第一章为绪论，介绍了全景图像在汽车主动安全领域具有很强的现实意义和经济利益，然后详细介绍了相关产品，并对所涉及的技术研究现状进行了归纳总结，最后，提出了论文的主要研究内容。

论文第二章为相关基础知识简介，主要介绍了摄像头模型、坐标系建立与转换、障碍物检测方法等知识，为后面的研究做理论储备。

论文第三章为硬件系统设计与实现，主要完成了需求分析、功能分析、选型分析，并根据上述分析进行了硬件电路设计，提出了系统的性能指标。

论文第四章为系统软件开发平台的搭建，主要完成了开发平台介绍、Linux 系统安装、嵌入式 Linux 开发环境搭建等。

论文第五章为障碍物检测与全景图像算法研究，主要完成了摄像头失真校正、障碍物检测与报警、逆投影变换获取俯视图、俯视图全景拼接算法的研究。

论文第六章为实验与测试，主要针对第五章的算法做了实际的针对性实验并进行了实车测试，通过实验测试，证明了硬件系统可靠、软件算法正确。

论文第七章为结论与展望，介绍毕业论文所取得的成果与结论，并针对研究过程中存在的不足进行了反思，明确了后续工作开展的方向。

2 图像拼接与障碍物检测理论基础

2.1 摄像头模型

2.1.1 线性摄像头模型

线性摄像头模型又叫做针孔模型，光线通过针孔，被“投影”到成像表面，如图 2.1 所示。焦距是针孔模型中很重要的一个参量，是用来表述投影平面上物体的大小与实际物体大小的关系量，针孔相机的焦距就是针孔到投影平面的距离，即图 2.1 中的 f ， Z 是摄像头到物体的距离， X 是实际物体的长度， x 是投影平面上物体投影的像的长度[42~43]。它们之间的关系如式(2.1)所示。

$$-x = f\frac{x}{z} \tag{2.1}$$

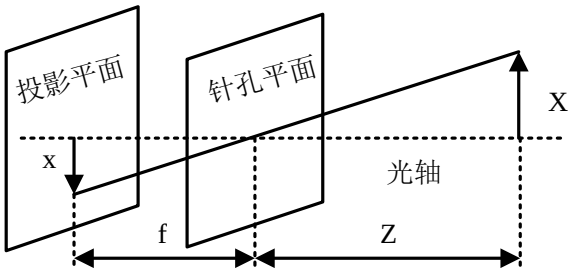

图 2.1 线性摄像头模型

Fig.2.1 Linear camera model

一般情况下图像传感器的中心位置通常不在光轴。在对线性摄像头建模时需要加入两个偏移参量 C_x ， C_y ，假设在世界坐标系中的点 Q ，坐标为 (X, Y, Z) ，其在投影平面上的投影点为 (x, y) ，则它们的关系如式(2.2)所示。

$$\begin{cases} -x = f_x\left(\dfrac{X}{Z}\right) + C_x \\ -y = f_y\left(\dfrac{Y}{Z}\right) + C_y \end{cases} \tag{2.2}$$

式中 $f_x = Fs_x$ 、 $f_y = Fs_y$ ，单位为像素， F 为透镜焦距的长度，单位是毫米， s_x 、 s_y 的单位是像素/毫米。

将公式矩阵化，可得到图像在线性摄像头下的模型公式(2.3)。

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = M\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} f_x & s & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.3}$$

式中矩阵 M 被称作线性摄像头在理想模型下的内参矩阵。f_x、f_y是以像素为单位的焦距。(C_x, C_y)为投影平面的中心，称为基准点。s 为畸变因子[44~45]。

2.1.2 非线性摄像头模型

因为实际生产过程中镜头的打磨工艺和装配误差，所以摄像头通常都不满足线性模型，需要重新建立摄像头的非线性模型。图像在投影平面上离中心轴越远，则畸变越大，图像失真越严重。非线性模型下的成像可用式(2.4)来描述。

$$\begin{cases} x' = x + \delta_x(x, y) \\ y' = y + \delta_y(x, y) \end{cases} \tag{2.4}$$

式中，(x', y')为线性模型下图像点的理想值坐标；(x, y)是实际图像点坐标，δ_x和δ_y是非线性畸变参数[46]。非线性畸变参数是个变化量，其与像素点在像平面中的具体位置有关，可用式(2.5)表示。

$$\begin{cases} \delta_x(x, y) = k_1 x(x^2 + y^2) + (p_1(3x^2 + y^2) + 2p_2 xy) + s_1(x^2 + y^2) \\ \delta_y(x, y) = k_2 y(x^2 + y^2) + (p_2(3x^2 + y^2) + 2p_1 xy) + s_2(x^2 + y^2) \end{cases} \tag{2.5}$$

式中，$\delta_x(x, y)$和$\delta_y(x, y)$多项式的第一项表示径向畸变，第二项表示离心畸变，第三项表示薄棱镜畸变，图像畸变通常由这三种类型的畸变叠加所致。公式中的k_1，k_2，p_1，p_2，s_1，s_2称为非线性畸变参数[47~48]。

通常情况下，人眼对径向畸变较为敏感，对离心畸变和薄棱镜畸变并不敏感，因此，只需考虑径向畸变即可，公式(2.4)和(2.5)可简化为：

$$\begin{cases} x' = x(1 + k_1 r^2) \\ y' = y(1 + k_2 r^2) \end{cases} \tag{2.6}$$

其中$r^2 = x^2 + y^2$，同时可以得出 x 方向的畸变与 y 方向的畸变相对值为$(\delta_x / x, \delta_y / y)$，图像畸变严重程度与径向半径的平方成正比[46]。

2.2 坐标系

图像的采集与显示过程中有个很重要的因素，那就是坐标系的建立。由于系统需要建立全景图像，就必须使用到投影变换，在使用投影变换时，也必须使用到坐标系。

2.2.1 世界坐标系

因为摄像头的位置将会伴随汽车的行驶而改变，所以，需要建立一个坐标系来方便研究。假设摄像头安装在汽车正前面，如图 2.2 所示的O点，摄像头的前端朝下，光轴为线OG，ABU 所在面即为路面，Y 轴是同地面相平行的车身对称线，摄像头位置在面 ABU 上的投影点即为点I，ABCD 所围成的面积即为摄像头在地面上的摄影范围，点P 为该区域内的任意一点。

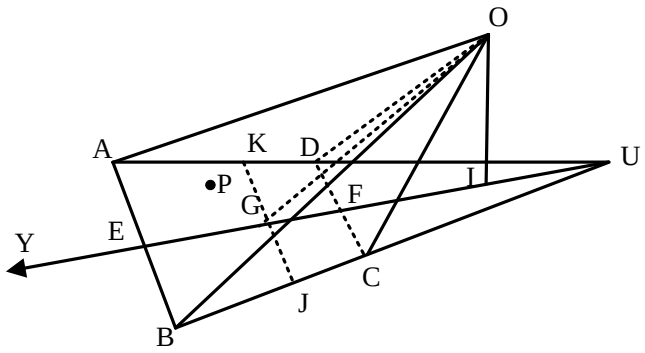

图 2.2 世界坐标系

Fig.2.2 World coordinate

如果将图 2.2 的世界坐标系往竖直方向上做投影，即可得到如图 2.3 所示的竖直方向坐标系。在图 2.3 中γ角即为摄像头光轴 OG 与地面的夹角，2α 为摄像头在竖直方向上的视角大小，点 E 和点 F 分别为摄像头在 Y 轴上采集图像的极限位置，其中 Y 轴即为车辆的行驶方向，h 为摄像头与地面的高度距离。

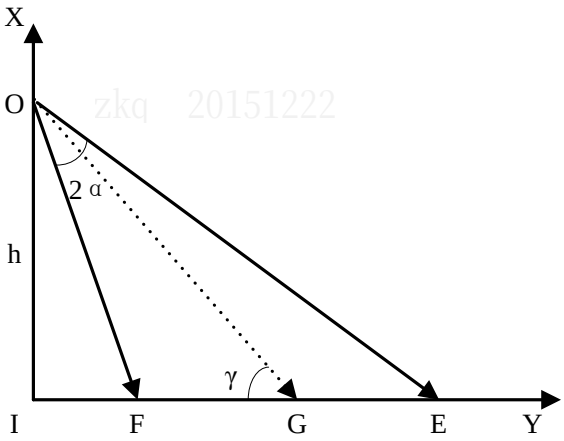

图 2.3 坐标系竖直投影

Fig.2.3 Vertical coordinate projection

如果将图 2.2 的世界坐标系往水平方向上做投影，即可得到如图 2.4 所示的水平方向坐标系，夹角 2β 即为摄像头在水平方向上的视角。

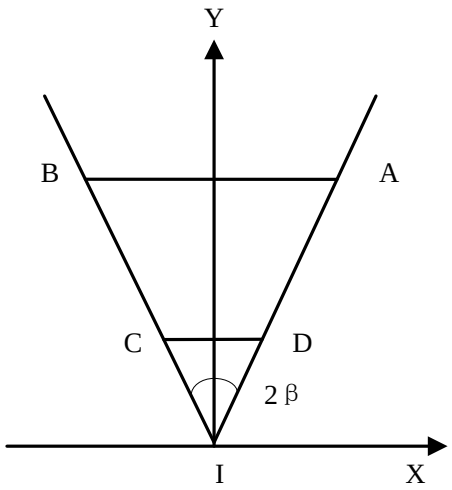

图 2.4 坐标系水平投影

Fig.2.4 Horizontal coordinate projection

2.2.2 图像坐标系

小孔成像图如图 2.5 所示：在世界坐标系中的任意一点 P，在摄像头所获取到的图像上都有点 P'。

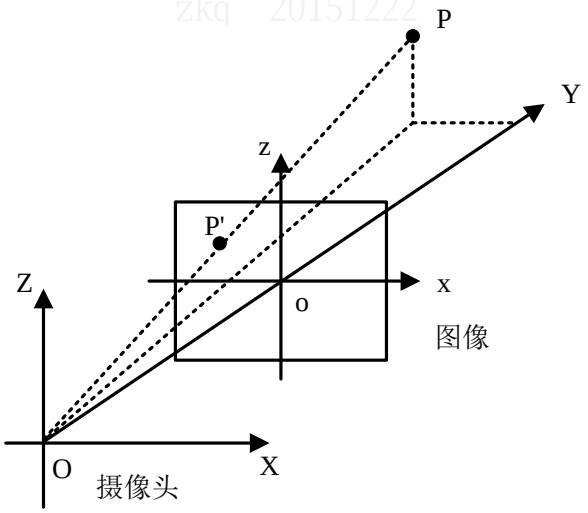

图 2.5 图像成像图

Fig.2.5 Imaging picture figure

故可以用摄像头所获取到的图像中心点，即光轴上的 G 点对应的 G' 点，作为图像坐标系的坐标原点，如图 2.6 所示。

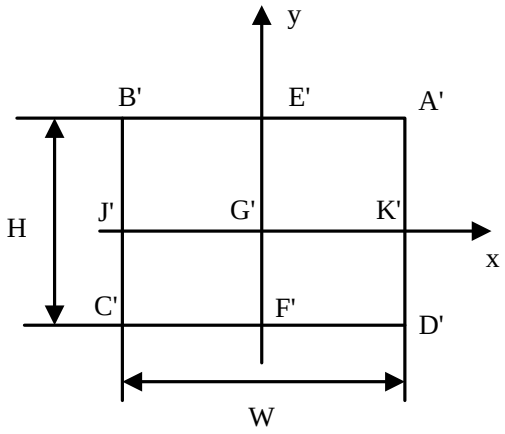

图 2.6 图像坐标系

Fig.2.6 Image coordinate

在图 2.6 中， A'B'C'D' 区域即对应于世界坐标系中摄像头视野范围在水平方向上的投影 ABCD 区域，G' 即对应于点 G ，W 和 H 分别代表图像的横向像素和纵向像素大小。由于图像坐标系和数字图像之间的单位不同，图像坐标系单位为米，数字图像的存储单位为像素，一幅图像的像素往往从图像的左上角开始扫描，所以需要建立两者的转换关系。

2.2.3 逆投影变换坐标系

通过将图像进行逆投影变换后得到的俯视效果图所确定的坐标系可称为逆投影变换坐标系。逆投影变换坐标系同数字图像的扫描位置相同，将左上角作为扫描的原点，并且将像素作为基本单位，如图 2.7 所示。

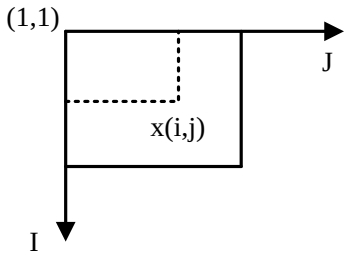

图 2.7 逆投影变换坐标系

Fig.2.7 Inverse perspective transforming coordinate

2.3 坐标系转换

2.3.1 逆投影变换

投影变换是指从图像坐标系到世界坐标系的转换方法，在文献[49]中详细介绍了这种方法，而逆投影变换则是一种将世界坐标系转换到图像坐标系的方法。

取路面上的一点 P，P 点为世界坐标系中的点，其坐标值为(P_x, P_y)，P'为图像坐标系中的点，其坐标值为(P_x', P_y')，两者为对应关系，根据图 2.2，图 2.4，图 2.5 和图 2.6，可以得到逆投影变换关系为：

$$\begin{cases} P_x' = \dfrac{P_x W}{2\tan\beta\sqrt{h^2 + P_y^2}} \\ P_y' = \dfrac{P_y H - Hh\tan\gamma}{2P_y\tan\gamma\tan\alpha + 2h\tan\alpha} \end{cases} \tag{2.7}$$

式中 h 为摄像头距地面的高度，2α、2β 分别为摄像头在竖直、水平方向上的视角，γ 为摄像头光轴 OG 与地面的夹角，H 和 W 为图像长与宽。

2.3.2 坐标映射与转换

俯视图像与原始图像的坐标映射关系 f 如图 2.8 所示。

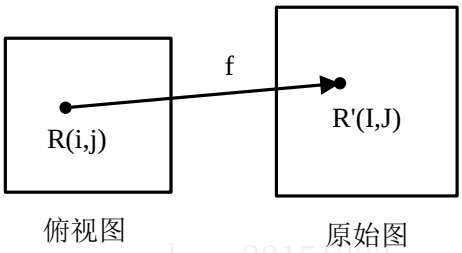

图 2.8 坐标映射

Fig.2.8 Coordinate mapping

由文献资料知，容易得到两者的坐标映射关系 f 为：

$$\begin{cases} I = \dfrac{H}{2} - P_y' \\ J = \dfrac{W}{2} + P_x' \end{cases} \tag{2.8}$$

其中，$R'(I,J)$即为原始图像中的像素点灰度值，即 YUV 色彩空间中的“Y”，(P_x', P_y')为俯视图中的像素点$R(i,j)$在世界坐标系下的坐标，该坐标容易由图像坐标系中的坐标通过式(2.7)以及式(2.9)得出。

$$\begin{cases} P_x = j * \dfrac{n}{N} - \dfrac{n}{2} \\ P_y = \dfrac{m}{2} - i * \dfrac{m}{M} + \dfrac{h}{\tan\gamma} \end{cases} \tag{2.9}$$

其中，m 与 n 为世界坐标系下摄像头的纵向与横向距离，M 与 N 为生成的俯视图的长和宽。

由于图像坐标映射变换后得到的灰度值可能带有小数，而数字图像的像素点只有是整数值才有意义，因此可以利用图像插值来解决上述问题。

2.4 障碍物检测方法

障碍物检测方法往往依据障碍物定义的不同而不同，目前，并没有对汽车行驶过程中的障碍物有着统一的定义。由于在汽车行驶过程中，周边汽车的影响与危害较大，故论文将障碍物定义为：车道上的其它行驶车辆。将障碍物危险定义为：距本车距离过近（小于安全距离）的其它行驶车辆。

论文采用四个安装在车辆的前后左右四面的广角摄像头，每两个广角摄像头均会有部分的视线重合区域，除此之外的其它地方则没有重合，故可将重合区域看作为双目视觉，非重合区域视为单目视觉。然而，由于非重合区域所占面积为绝大部分，重合区域只在汽车的四个角，即左上、右上、左下、右下，而对于汽车行驶过程中较为危险的后方追尾，则不包含在内。因此，如果采用双目立体视觉，则会因为重合区域面积不够，产生检测盲区，而使用单目视觉，就能完整检测到汽车周围的障碍物，故障碍物检测可采用基于单目视觉的障碍物检测方法。各种障碍物检测方法优缺点比较，如表 2.1 所示。

表 2.1 障碍物检测方法优缺点比较

Table 2.3 Advantages and disadvantages among different obstacle detection methods

方法名称	优点	缺点
基于特征的检测方法	速度快、算法相对简单	只适用于特征已知的障碍物
帧间差分法	运算量小、处理速度快	对运动目标速度要求高、检测和分割的精度不高
光流法	携带运动信息和场景信息	对噪声、多光源、阴影和遮挡敏感，计算复杂，虚警率高

由于系统要求障碍物检测快速、准确，同时，系统要求对光照和阴影不敏感，因此，光流法不适用。而汽车在行驶过程中，速度较快，摄像头所摄背景不恒定，容易使得两帧之间图像无相互覆盖的区域，无法分割出运动目标，故而无法采用帧间差分法。作为论文障碍物选择的汽车其特征较为明显，故论文决定选择基于特征的检测方法作为障碍物检测方法。

2.5 本章小结

本章节主要介绍了图像拼接与障碍物检测算法的理论基础，为后面章节的设计与研究做了理论储备。具体来说，首先介绍了摄像头模型以及坐标系相关知识，然后介绍了障碍物检测方法，并针对论文的实际研究情况，选择了合适的算法。

3 硬件系统设计与实现

3.1 系统分析

3.1.1 系统需求分析

论文要设计与实现的汽车全景主动安全系统，主要需满足以下几点需求：

①全景图像完整。需要得到汽车四周的全景图像，图像应该完整，无视线盲区。

②图像清楚。采集到的图像应该清晰，从而能准确的实现对障碍物进行特检测，全景图像系统生成的显示图像应该清晰，能辨认出周边的情景。

③障碍物检测快速准确。需要设计一种快速简单准确可靠的障碍物检测方法，该方法能够检测出距车身距离小于安全范围的汽车障碍物。

④危险报警。检测并识别到安全范围内的障碍物后，能够发出声音报警信号，提醒驾驶员危险存在。

⑤摄像头选择与安装。需要选定合适的图像采集方案，既要能够得到汽车四周的图像信息，防止视线盲区，又要成本低廉，能够实现产品化。同时，摄像头的安装应该尽可能地减少对汽车的损伤，保持汽车的整体美观性。

⑥实时性较好。需要采用高速的视频处理方案，包括硬件芯片、软件算法处理，保证系统的良好实时性。

3.1.2 系统功能分析

论文设计的全景主动安全系统主要功能可以分为以下几点：

①图像采集。利用摄像头采集图像。

②图像处理。快速处理各个摄像头采集到的图像，并将其拼接为全景图像。

③图像显示。显示出拼接出来的全景图像。

④摄像头失真校正。对广角摄像头进行图像失真校正，便于后面逆投影变换形成俯视图。

⑤坐标系的建立。建立图像坐标系与世界坐标系的映射关系，从而确定图像位置，拼接点位置。

⑥障碍物检测与报警。利用摄像头采集到的图像，根据障碍物判断算法，检测障碍物存在，并对安全范围内的障碍物发出报警信号。

⑦图像拼接。将摄像头采集到的图像通过图像拼接算法拼接为全景俯视鸟瞰图像。

3.2 硬件系统总体框图与模块叙述

根据 3.1 节的系统分析结果，对硬件系统进行总体设计，设计的总体框图如图 3.1 所示。

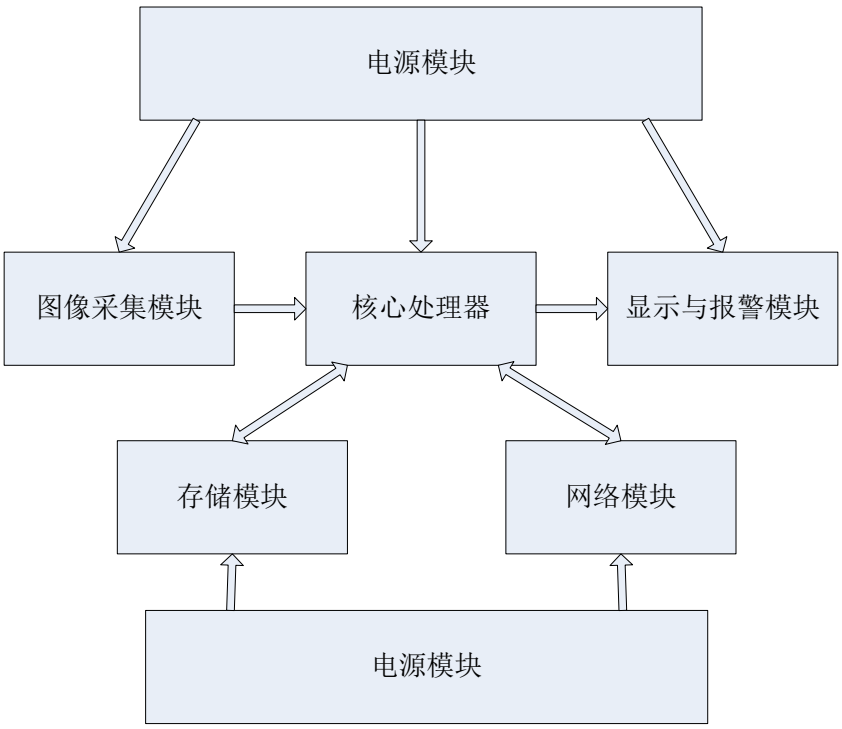

图 3.1 硬件系统总体框图

Fig.3.1 The overall block diagram of the hardware systems

从图 3.1 中可以看出，硬件系统主要分为六个部分，这六个部分的主要功能如下。

3.2.1 图像采集模块

图像采集模块是整个系统的关键部分之一，图像采集方式设计的好，能大大减少核心处理器的计算负担，加快整个系统的反应速度，从而提升整个系统的应用价值。图像采集模块主要可分为两大部分，一部分是视频图像的采集，另一部分是视频图像的解码。视频图像采集主要需完整采集到汽车四周的图像，从而为后续全景图像的拼接和障碍物的检测提供素材。视频图像解码主要是将采集到的视频图像进行压缩处理，减轻核心处理器的负担，能够使核心处理器更专注于图像拼接和障碍物检测上。

3.2.2 核心处理器

核心处理器是整个系统的大脑，具体对于本系统的设计来说，需要有一款高性能、低功耗、易于开发和扩展的芯片。

3.2.3 电源模块

车载电源一般是直流 12v，因此整个系统设计需要解决车载电源电压稳定性问题，同时为各个芯片提供不同大小的电压。

3.2.4 显示与报警模块

显示与报警模块主要是全景图像显示以及障碍物报警提示，障碍物报警为声音报警。

3.2.5 存储模块

存储模块采用 Nand-flash，该存储方案是目前比较流行的一种内存方式，具有价格低廉、容量大、速度快等多种优点，可以广泛应用于大数据的存储中。

3.2.6 网络模块

网络模块主要是为网络文件系统（简称：NFS）的实现提供了保证，嵌入式 Linux 系统可以通过 NFS 实现文件系统的挂靠，从而避免了在开发过程中频繁的进行烧写，进而大大提高开发效率。

3.3 系统选型分析

3.3.1 图像采集方式选择

①全景图像获取方式

系统利用汽车四周的图像信息拼接为全景图像，这就需要先获取到汽车四周的完整无视线盲区的图像。根据相关文献资料可知，获取汽车四周的图像信息有以下几种方法：

1)一个摄像头+可转动的云台

这种方法主要由一个摄像头，往往是高清摄像头，加上可 360 度转动的云台组成。在摄像头采集图像的同时，云台按照一定的角速度转动，这样就能得到汽车四周的图像信息。这种方法有几个缺点：1、成本较高，尤其是好的转动云台，价格较高。2、采集到的图像有时间间隔，不是同时采集到的。由于该方法是通过云台转动来获得 360 度图像，不同图像获得的时间是不同的，拼接出来的全景图像是不同时刻的图像。这就会造成障碍物检测时的漏检，因为很可能会遇上当障碍物出现时，云台转到了相反方向，没有采集到障碍物信息的危险情况。当然，解决这种问题的方法是有的，那就是加快云台转动的角速度，提高同一位置图像的刷新率，但那往往会加大对云台质量的要求，从而进一步加大成本投入。3、美观要求。为了能够采集到汽车四周的图像，云台往往安装在车顶，而且为了能够不被车顶挡住部分视线，需要将云台架高，这就有点像预警机一样，从而影响车辆的整体美观。更重要的是，对于大型车辆，这样加高后，会使整个车辆的高度增加，无法通过一些限高的隧道、路段等。

2)多个摄像头同时采集图像信息

这种方法是在汽车的每个方向上安装至少一个摄像头，从而消除视线盲区。该方法的优点有：1、能同时采集汽车四周的图像信息，避免拼接出来的全景图像存在明显的图像时刻差。2、成本较低。由于技术的进步，现在高清摄像头的价格已经很低了，能够负担起为一部车配备多个摄像头的成本。3、不影响车辆美观。已商品化的倒车摄像头就已经能够做到小且隐秘，从而不影响车辆的整体美观。

不过，该方法存在数据计算量大，从而导致系统实时性较差的缺点。为了提供汽车四周的完整详细图像，摄像头数量越多越好，然而这将使得图像拼接的计算量骤增，从而降低整个系统的实时性。

3)多个摄像头+转动的云台

这种方法是上述两种方法的综合，在汽车的关键位置安装固定的摄像头，而在车顶云台上安装转动的摄像头，这样就能保证图像不漏采。但是这种方法的成本是最高的，不适合商品化。

基于上述分析，论文将采用方法二来采集汽车四周的完整图像，并使用最少摄像头数目，即四个摄像头[16]。

②图像采集方案

目前，常用的图像采集方案有多种，主要是依据实际使用的背景、环境、任务来选择。常用的图像采集方案如下：

1)PC 计算机+图像采集卡或者 PC 计算机+USB 摄像头

这种方案的优点是图像处理端可以使用计算机 CPU，随着目前计算机的快速发展，CPU 的处理能力也越来越强，图像处理速度也越来越快；同时，计算机上拥有丰富的图像处理相关软件，利于将采集后图像进行后续的处理。但是，这种方案的缺点也很明显，一个主要的缺点就是体积较为庞大，即使是笔记本电脑，体积也较大，尤其不适合车载系统的处理，当然，也许未来能实现计算机与汽车的完美链接。

2)带 USB 口的 ARM+USB 摄像头

这种方案采用的 ARM 芯片，较计算机来说体积大为缩小，适合车载系统，而且 USB 摄像头市面上很好买到，还有不同的外观可选择。但是，这种方案的缺点也很明显，图像传感器采集端使用的 USB 摄像头，一般分辨率较低，图像质量也较差，对于略微高速图像捕捉效果较差，因此，不适合本系统。

3)带 Camera 口的 ARM+CMOS

这种方案具有高集成度的优点，实现难度也较之前的方案要大些，但是由于 CMOS 采集的图像质量没有 CCD 采集的图像好，因此，对于要求图像质量较高的本系统来说，不适合。

4)ARM+ FPGA+CCD+视频处理芯片

这种方案中采用的 CCD 很适合对图像采集具有高质量、快速、体积小、功耗低的系统。采用的 FPGA 具有极高的集成度，能够实现动态系统重构和静态重复编程。采用的 ARM 作为目前流行的处理器，在其平台上能开发出满足设计者要求的应用程序，可实现程序管理功能。这种方案优点突出，能够设计出小巧、高稳定性的图像处理系统，但对设计者的要求较高，具有较高的门槛。

5)ARM+ DSP +CCD+视频处理芯片

这种方案中采用的 DSP 与 ARM 的双核处理器，由于随着技术的发展，DSP 现在的成本已经降低了很多，FPGA 在成本上的优势已经不是很大。同时，DSP 拥有独特的硬件结构体系，很适合用来处理数字信号，正如其名字一样，例如，DSP 采用的哈佛存储结构，专用的硬件乘法器，流水线操作等等，这些都有助于用来实现对各种数字信号的快速处理，因此，DSP 作为一款处理器非常适合具有大量数字信号处理的图像处理系统。

基于上述分析，论文决定采用第五种，即 ARM+DSP+CCD+视频处理芯片的图像采集方案。

3.3.2 摄像头选型分析

全景图像拼接要实现的首要前提就是图像要存在重合区域，没有重合区域无法实现对图像的拼接。而为了使得图像具有重合区域，有两种方法，一种是增加单个摄像头的视角范围，另一种是采用更多的摄像头。由上一小节论文已经按照文献参考确定使用最小摄像头数目，即 4 个摄像头，因此，只能选择宽视角的摄像头作为图像采集传感器。普通的平角摄像头其视线范围在 40~90 度之间，如图 3.2 所示为汽车装上四个普通平角摄像头后的视线范围，图中灰色部分即表示摄像头的视线范围。由图可以看出，使用普通平角摄像头获取的车身周围的实时图像缺少重合区域，并且有很大的视线盲区，因此，采用普通平角摄像头无法满足全景拼接的要求。

广角摄像头作为一种具有短焦距且视线范围宽广的摄像头，其成像的视场角能够达到 180 度，部分甚至能够达到惊人的 270 度[50]。如果采用四个广角摄像头作为图像采集传感器，如图 3.3 所示为采用了广角摄像头后汽车四周的视线范围图，图中灰色部分为单个广角摄像头的视线范围，黑色部分为两个广角摄像头视线范围的重合区域。由图 3.3 可以看出，汽车前后左右安装广角摄像头后，能够几乎完整地采集到汽车四周的环境信息，并且相邻两个广角摄像头存在相当范围的视线重合区域，为全景图像的拼接提供了广泛的素材。因此，论文决定采用四个广角摄像头来采集汽车四周的图像信息，这四个广角摄像头分别安装在汽车的前后左右四面，具体而言，左右广角摄像头安装在汽车左右后视镜处，并保证四个

广角摄像头的安装高度相同，同时，每个广角摄像头前端向下，保证广角摄像头的光轴与地面之间存在夹角，便于后面图像逆投影变换为俯视图和障碍物检测。

图 3.2 普通摄像头视线范围

Fig.3.2 The sight of ordinary cameras

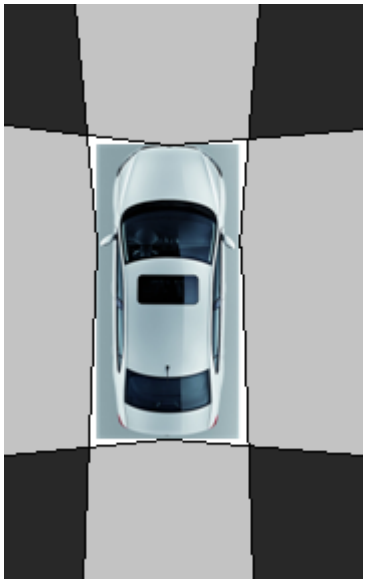

图 3.3 广角摄像头视线范围

Fig.3.3 The sight of wide-angle cameras

3.3.3 核心处理器选型分析

①核心处理器需求分析

根据论文之前的叙述，系统的核心处理器需要有很强的计算能力，能够及时处理 4 路摄像头的图像信号，同时为了节省开发成本，需要具有高性价比，为了能适应车载环境，还必须具有低功耗、高稳定性等嵌入式系统的要求，最好还能够实现系统的升级以及能后续扩展应用的功能。综合目前市场上芯片的售价以及其它文献资料参考，论文决定选用 TI 公司的 TMS320DM6446 芯片（论文以下简称为 DM6446）作为系统核心处理芯片。

②DM6446 的硬件结构

DM6446 作为一款开放架构的双核视频和图像处理专用芯片，具有高性能、低功耗、内存空间较大的特点，其同时提供了专用的视频图像协处理器（简称：VICP）和视频处理子系统（简称：VPSS），并拥有众多的外设扩展[51]。DM6446 的硬件结构和功能框图如图 3.4 所示。

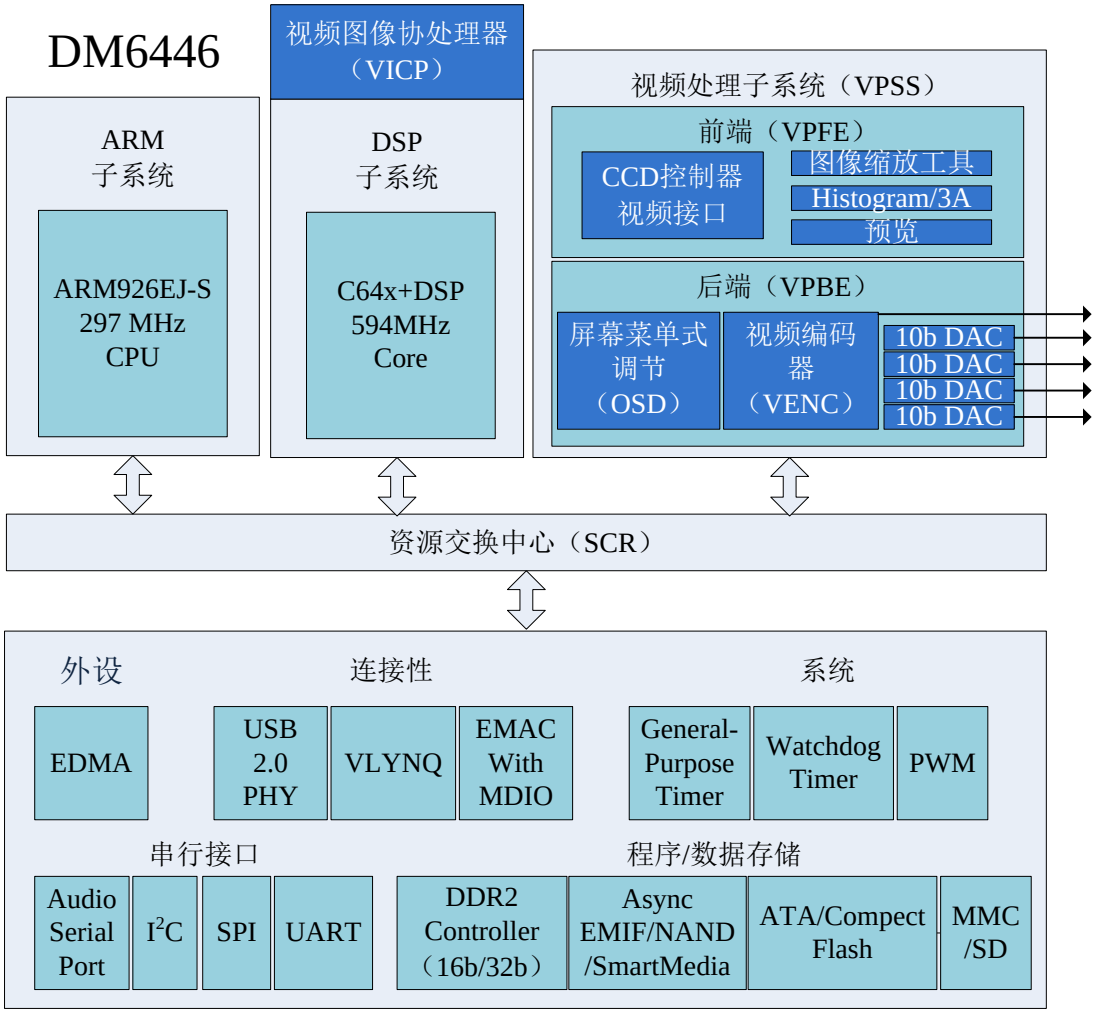

图 3.4　DM6446 硬件结构和功能框图

Fig.3.4 The hardware structure and function block diagram of DM6446

从图中可以看到，DM6446 拥有两个子系统，一个是 297MHz 的 ARM926EJ-S 内核，另一个是 594MHz 的 C64x+DSP 内核。当 DM6446 的 DSP 端频率达到最高 594MHz 时，其每秒可执行最多达 4752 百万个的指令集，具有很高的运算性能，能够充分满足本系统所设计的全景图像处理的计算要求。同时，由于 DM6446 芯片采用多电源的管理模式，使得整个芯片的功耗很低，非常适合车载嵌入式系统开发的环境[52]。

内存方面，DM6446 芯片包含有 16 位 128MB 大小的 FLASH 存储空间，以及 32 位 256MB 大小的 DDR2 SDRAM 内存空间。，ARM9 内核具有 16KB 大小的指令和 8KB 大小的数据 Cache，DSP 内核具有 32KB 大小的程序 RAM/Cache，80KB 大小的数据 RAM/Cache 及 64KB 大小的未定义 RAM/Cache[52]。外设方面，DM6446 具有数量丰富的外部设备，具体可参见图 3.4。

3.3.4 视频解码芯片选型分析

由于论文需要采集四路广角摄像头的图像信号，因此需要一款能够同时解码四个通道广角摄像头视频输入的芯片，而 TI 公司的 TVP5158 就是一款满足上述需求的芯片。

TVP5158 是一款具有独立定标器、噪声抑制、自动对比功能的四通道 NTSC/PAL 视频解码器[53]。该芯片能够同时对多路摄像头的视频输入进行解码，从而降低了解码所需要的成本。由于 TVP5158 芯片与 DM6446 芯片同是 TI 公司为了开发多路图像处理而研发的芯片，所以 TVP5158 在视频输出方面拥有多种适合 DM6446 的方式，从而能够实现与 DM6446 完美配合，而无需其它中间设备转换和外部的 FPGA 的需求。因此，采用 DM6446 作为图像处理芯片，TVP5158 作为视频解码芯片的选择，能够降低系统成本和开发难度。

TVP5158 芯片的具有的具体特性及优势如下：

①能够同时对四个通道输入进来的高质量的 NTSC 制式或者 PAL 制式视频进行解码操作，并可为 DSP 芯片生成单一的数据输出流，从而大大降低每个视频通道解码的成本。

②芯片具有自动对比度控制功能，能够改善在照明条件不好情况下的图像质量，而无需采用 DSP 来处理，从而减少相关软件算法，加快系统响应速度。

③芯片可以使用二维空间滤波技术来实现降低视频噪声的功能，从而提高在不良光照条件下的影像质量与压缩比，进而实现更好的存储压缩。

④能够实现与 DM6446 的音频、视频完美连接，视频输出端支持 8 位的 ITU-R BT.656 协议和 16 位的 YCbCr 4:2:2，并可直接将视频供给 DM6446。模拟/数字核和 I/O 口的电压分别是 1.1v，1.8v 和 3.3v，端口能够直接与 DM6446 相连，而无需进行电平转换。

3.4 硬件电路设计

3.4.1 电源模块电路设计

由于系统芯片对电源的稳定性要求很高，因此效率高、具有良好噪声抑制、输出纹波小的高效电压转换芯片是系统的重要选择。汽车在启动与正常行驶时，由于其它因素的干扰，电瓶输出的电压一般不稳定，会在 10-16v 范围内无规律变化，而这种大幅度的电压波动会对系统芯片造成损害，导致出现严重的事故。因此，需要对从车上引入的 12v 供电电源添加保护电路，从而确保整个系统的电路安全。电源保护电路如图 3.5 所示。

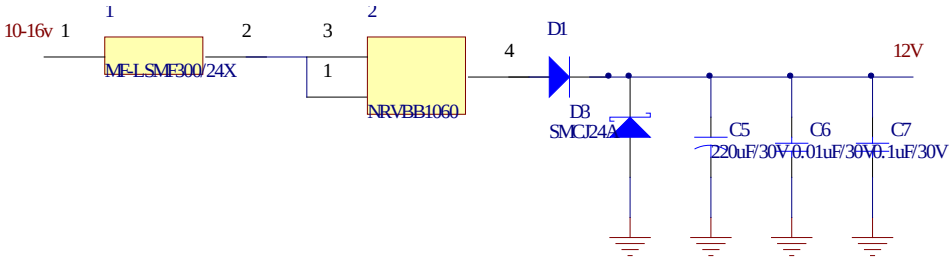

图 3.5 电源保护电路图

Fig.3.5 Power protection circuit diagram

因为系统采用了多种芯片，而芯片各自的内核供电电压以及 I/O 口电压不同，所以需要设计电源电压转换电路来获取各种电压。

为了确保 DM6446 芯片能够稳定工作，需要进行电源管理设置，对芯片的内核以及 I/O 口进行单独供电，以减少电源间的串扰现象。由芯片手册可知，DM6446 的内核供电电压为 1.2v，I/O 口电压可支持 3.3v 和 1.8v[52]。视频编码芯片 TVP5158 的数字核电压为 1.8v，模拟核电压为 1.1v，I/O 口电压为 3.3v[53]。电路上部分其它芯片所需电压为 5v。因此，总的来说，系统需设计 1.1v、1.2v、1.8v、3.3v、5v 的电压。考虑到需要设计诸多电压，为了减少电路设计的复杂性，以及为了方便元器件的购买与更换，后期电路检测的简便，对电源电压转换电路采用统一的电路设计，只是在个别电阻的阻值上有所不同。电源电压 12v 转 1.2v 电路设计如图 3.6 所示。

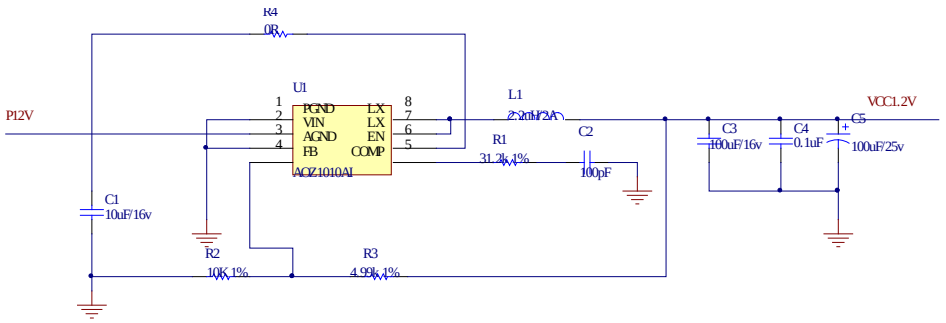

图 3.6 电压 12v 转 1.2v 电路图

Fig.3.6 12v to 1.2v voltage switching circuit diagram

电源电压转换电路采用了芯片 AOZ1016。AOZ1016 是一款简单、高效的 2A 电流降压稳压器[54]。它可以工作在 4.5v 至 16v 的电压输入环境中，满足车载上的电压输入要求，同时 AOZ1016 最低可将电压降低为 0.8v，低于系统所设计的最低电压，因此这款芯片的输入输出电压完全满足系统需求。同时，该芯片还具有短路保护、欠压锁定和输出过电压保护的功能，因而非常适合作为降压芯片使用。

采用该芯片设计的电源电压转换电路非常简单，对不同的输出电压需求，只需替换与芯片 FB 口相连的电阻 R3 的阻值大小、以及与芯片 LX 口相连的电感 L1 的阻值大小即可，其它电路元器件连接方式和阻值大小均不变。具体电压与阻值关系对应如表 3.1 所示。

表 3.1 转换电压与电阻、电感值对应关系

Table 3.1 Relationships among switching voltage, resistance and inductance value

转换电压	R3	L1
1.1v	3.74k	2.2μH
1.2v	4.99k	2.2μH
1.8v	12.7k	2.2μH
3.3v	31.6k	3.3μH
5.0v	52.3k	4.7μH

3.4.2 视频解码芯片 TVP5158 电路设计

TVP5158 具有 128 个引脚，同时多个引脚的电压不同，因此需要尽量减少电源纹波的干扰，应多放置电容，同时电容应该尽量靠近引脚。TVP5158 的引脚电压图如图 3.7 所示。

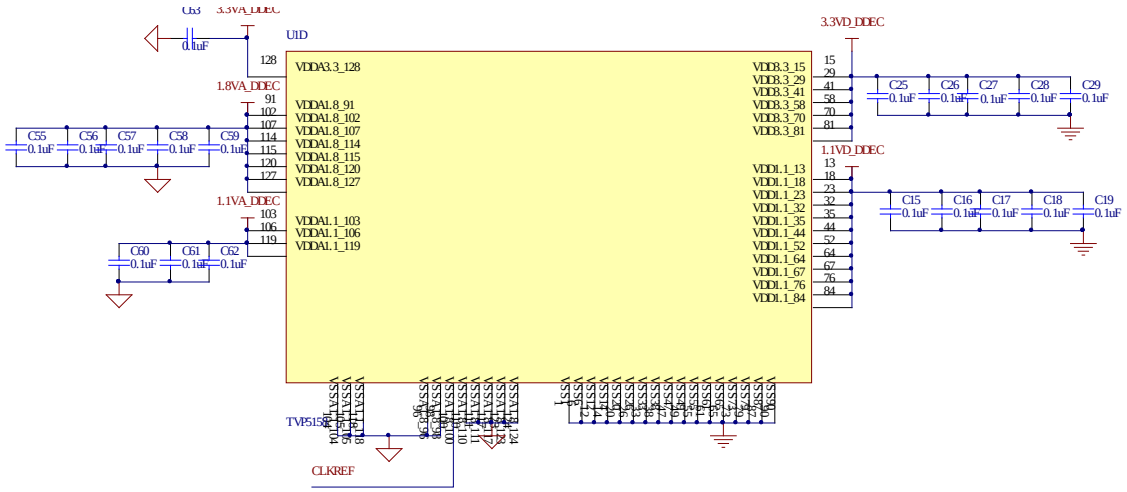

图 3.7 TVP5158 引脚电压图

Fig.3.7 TVP5158’s pins voltage map

TVP5158 解码 4 路视频信号，并输出单一一路数据流，但是，为了增加系统的可靠性，在硬件设计上，设计两路输出，加大故障冗余度。电路图如图 3.8 所示。TVP5158 也能同时采集 4 路音频信号，其电路图设计如图 3.9 所示。

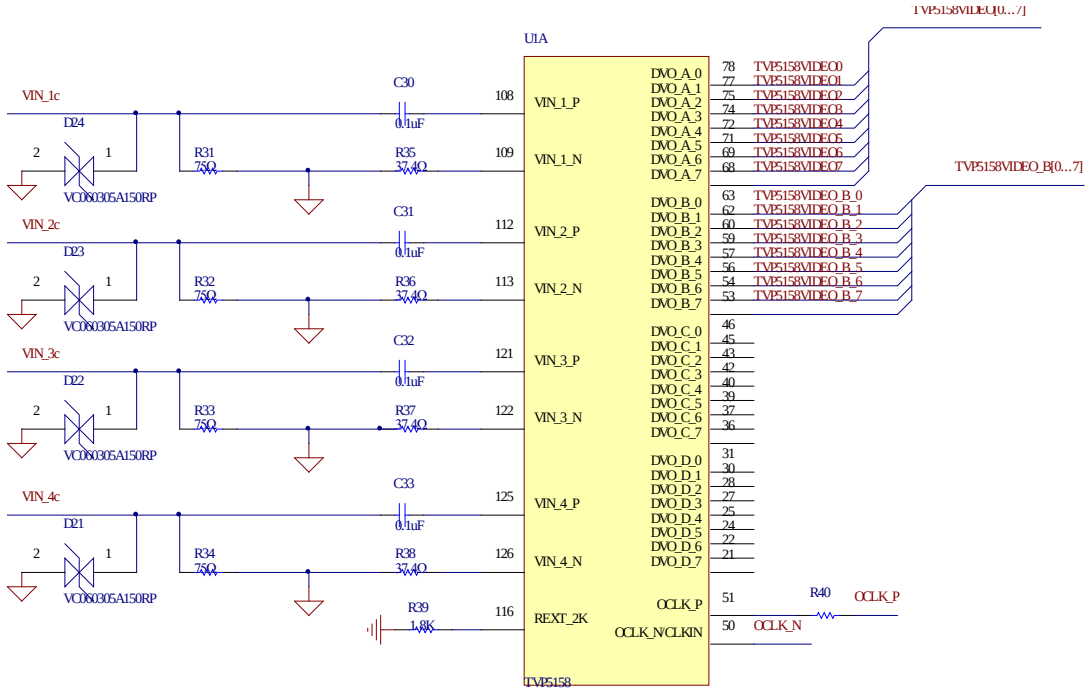

图 3.8 视频输入与输出电路图

Fig.3.8 Video input and output circuit diagram

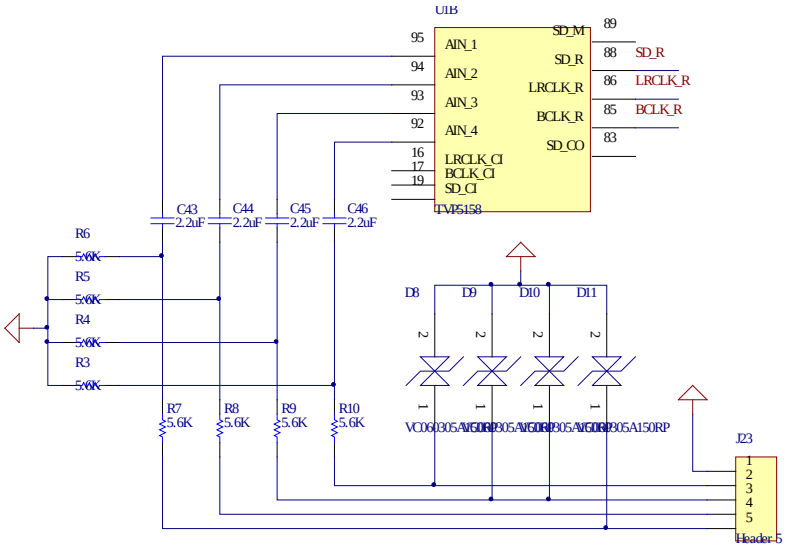

图 3.9 音频输入电路图

Fig3.9 Audio input circuit diagram

3.5 系统性能指标

3.5.1 硬件性能指标

①最小系统：DM6446 处理器，594MHz C64x+DSP 内核，297MHz ARM9 内核，256MB DDR SDRAM，64MB NAND Flash，64KB EEPROM。

②视频输入：CVBS 端子，支持 NTSC/PAL 制视频采集设备，如摄像头，DVD。论文的视频输入为 YUV（即 YCbCr）颜色空间。

③视频输出：CVBS 端子，VGA 输出，支持 NTSC/PAL 制视频显示设备，如液晶屏、显示器；

④音频输入：3.5mm 标准麦克风插槽；

⑤音频输出：3.5mm 标准耳机插槽，双声道立体声输出接口；

⑥外设接口：SATA 硬盘接口，USB2.0 接口；

⑦通讯接口：10/100MBps 以太网 RJ45 接口，RS232 串口，RS485 串口；

⑧摄像头分辨率：320*240；

⑨系统输入电压：DC10-16V。

3.5.2 软件性能指标

根据系统的设计目的，确定系统的软件性能指标如下：

①系统响应时间：从图像采集到全景图像拼接显示整个程序的响应时间小于 0.2s，障碍物检测程序响应时间小于 0.1s；

②检测正确率：障碍物检测正确率高于 90%；

③全景图像拼接完整性：4 个广角摄像头采集到的图像应该完整拼接，完整显示。

3.6 本章小结

本章主要完成了硬件系统的设计与实现，为下一章软件开发平台的搭建提供了核心板硬件支持。具体来说，首先对系统进行了需求分析和功能分析，然后按照分析结果设计了硬件系统的总体框图，接着完成了系统图像采集方式的选择和摄像头、核心芯片的选型分析，在完成选型后，对主要的硬件电路进行了设计，并设计了系统的性能指标。

4 系统软件开发平台搭建

4.1 系统软件开发平台概述

4.1.1 DM6446 的软件结构

DM6446 芯片采用 TI 公司的达芬奇技术，在软件框架在结构上可以分为应用层、信号处理层和 I/O 层三个部分。DM6446 的应用层由 ARM 内核开发，并采用 TI 公司专用的 MontaVista 内核。因此，开发和构建基于 DM6446 芯片的应用程序，必须在 Linux 环境下。MontaVista 内核具有两大优点：实时性高，提供集成开发环境。DM6446 的信号处理层主要是运行在 DSP 一侧，主要负责对信号进行处理。DM6446 的 I/O 层主要是针对 DaVinci 的外设模块进行驱动程序编写。

基于达芬奇技术的 DM6446 的软件架构能够协助开发人员在 Linux 平台上，针对 TI 公司的以 DSP 为基础的系统单芯片处理器开发各种应用软件，或者是在嵌入式 Linux 系统或 DSP/BIOS 实时核心上针对 TI 公司采用达芬奇技术的纯 DSP 组件开发软件。

4.1.2 系统软件方案

针对论文的全景图像系统的需求，第三章在核心处理器选型部分已经选用了 TI 公司的 DM6446 芯片。DM6446 芯片是一款双核芯片，分为 ARM 核和 DSP 核两部分，因此，软件设计也需要分成两部分，即 DSP 处理器部分和 ARM 处理器部分。ARM 处理器端主要的工作是实现应用程序的开发，运行在 Linux 系统下；DSP 处理器端主要是算法实现，ARM 处理器端与 DSP 处理器端通过 TI 公司专用的技术实现交互。

根据论文研究的实际情况，论文的系统软件方案为：在 Windows PC 机下安装虚拟机 Linux 系统，并在虚拟机 Linux 系统下安装 MontaVista Linux 内核，通过 NFS 来实现文件系统挂靠在核心板上，同时通过 MontaVista Linux 内核下的 ARM 端来控制 DSP 端的障碍物检测算法和全景图像拼接算法。软件开发平台的建立将在下一小节中详细叙述。

4.2 系统软件开发平台组成

嵌入式系统由于系统的特殊性，其开发往往需要专门的工具与环境。因此，建立好软件开发平台就是嵌入式系统开发的第一步。针对本系统的开发，主要是开发环境、编译环境的搭建，BootLoader 启动设置等等。

论文的系统软件开发平台主要由 PC 机、核心板、广角摄像头、液晶屏组成，具体的连接方式如图 4.1 所示。PC 机中将使用 Red Hat 5.0 版本的虚拟机 Linux 系

统作为 Linux Host。在 PC 机的 Windows 系统下使用超级终端通过串口线与核心板串口相连，利用超级终端实现命令行输入。PC 机与核心板通过网线连接，是为了实现 NFS 文件系统的挂载，便于利用 NFS 服务器来提升开发的效率。核心板上电启动 Bootloader 后，系统将会自动运行 FLASH 指定位置的 MontaVista Linux 内核镜像文件 uImage，并通过 NFS 来实现对已经在 Linux Host 上安装好了的 MontaVista Linux 目标文件系统的启动。四个广角摄像头通过视频线将图像信号输入核心板，视频线采用 CVBS，核心板再将处理后的图像通过视频线传输到液晶屏，让液晶屏显示图像。

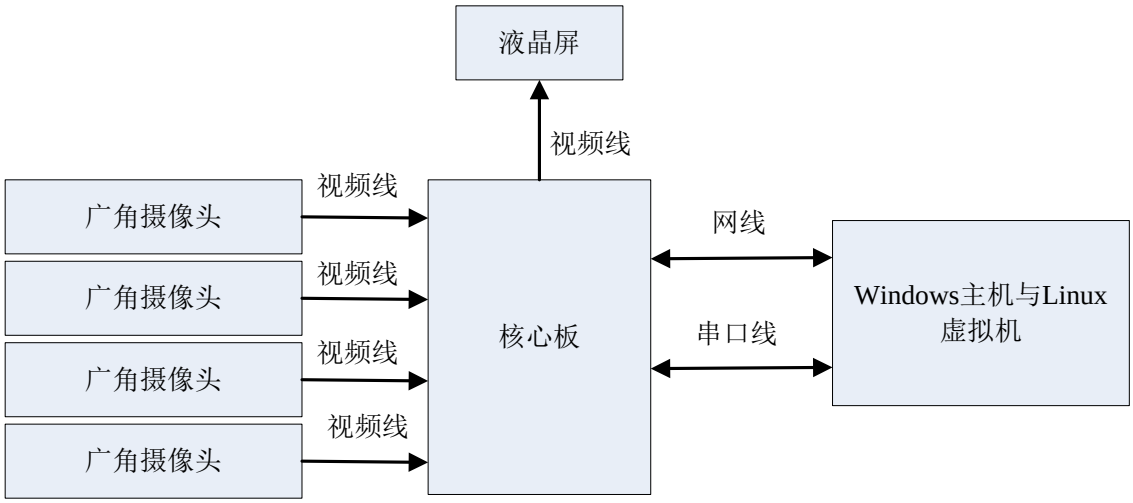

图 4.1　软件平台组成图

Fig.4.1 Software platform elements diagram

4.2.1 虚拟 Linux 操作系统安装

需要在 PC 机上的 Windows 系统下运行 Linux 系统，通常有两种方法：一种是在系统上安装双系统，即同时安装 Windows 系统与 Linux 系统，另一种就是在 Windows 系统下安装虚拟机 Linux 系统。

采用虚拟机系统比双系统具有以下优势：

①能同时运行两个系统。可以同时运行 Windows 系统和 Linux 系统，这就使得程序的编写可以在熟悉的 Windows 系统下完成，然后复制到 Linux 系统下即可，而无需在不方便的 Linux 系统下进行。

②调试方便。避免了双系统下来回重启电脑的麻烦；

③资源占用少。虚拟机只是在其它分区中划分出一部分虚拟成它的分区，而无需像双系统那样占用整个分区。

因此，论文选择了安装 Linux 虚拟系统的方法来实现双系统的使用。

VMware Player 是一款广泛使用好评率很高的的虚拟机软件。该款软件相对于价格不菲和体积庞大的 VMware Workstation 来说，价格免费并且体积小巧，非常利于实际应用。VMware Player 之所以小巧是因为它舍弃了 VMware Workstation 的很多扩展功能，但能够创建和运行虚拟机系统，能支持光驱、移动硬盘、闪存盘等设备，能支持用户网络和多种虚拟机格式。对于论文设计的系统来说这就足够了，因此使用 VMware Player 作为虚拟机软件即可。

安装好的虚拟机系统如图 4.2 所示。

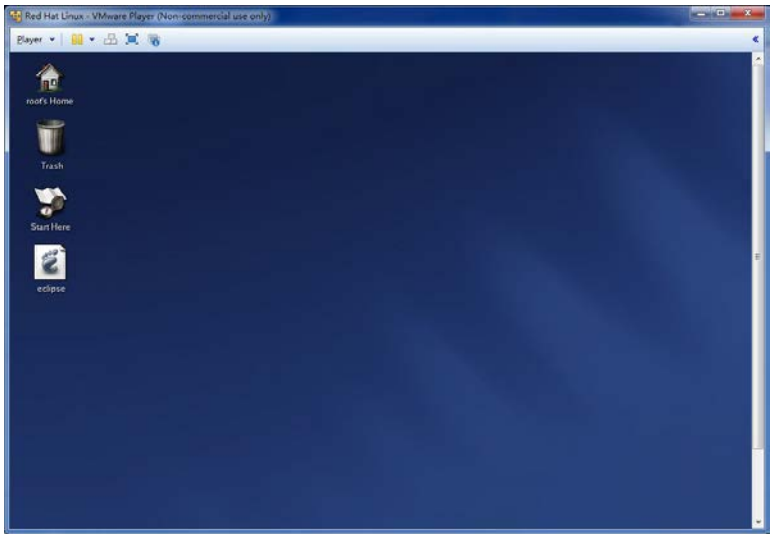

图 4.2 虚拟机系统桌面图

Fig.4.2 Virtual machine desktop map

4.2.2 搭建 ARM 编译环境

搭建 ARM 编译环境分为三步骤，首先是安装 MontaVista Linux 环境，其次是安装 DVSDK 工具链，最后是安装 Codec Servers 。 MontaVista Linux 环境是 TI 公司专门用于开发 DM6446 芯片的内核，该环境主要包含目标文件系统和开发工具包。搭建 ATM 编译环境具体安装步骤参见附录 B。

4.2.3 设置 NFS 服务器

嵌入式 Linux 核心板可以与电脑主机通过网线构成网络，并通过 NFS 实现两者之间文件的共享，这就使得开发嵌入式系统不必反复进行烧写工作，从而提高开发应用软件的效率。对于本系统来说，通过使用 NFS 将 Linux Host 上的 MontaVista Linux 目标文件系统映射到开发板上，从而使得开发板在自身无文件系统的情况下可以执行应用程序。

在虚拟机系统下完成 NFS 的安装后，需要在其上建立电脑主机与核心板文件系统的映射目录。首先建立地址为 /home/davinci/workdir/filesys 的目录，并将其作为系统的映射目录，然后将系统启动所需要的目标文件系统复制到这个目录下，目标文件系统存放于虚拟机系统下的这个目录下：
/opt/mvl_pro_5.0.0/montavista/pro/dekit/arm/v5t_le/target/ 。接下来，再切换进入到 Linux 虚拟机系统的命令行界面，并输入下列命令。

$ cd /

$ mkdir /home/davinci/workdir/filesys

$ cd /home/davinci/workdir/filesys

$ cp /opt/mvl_pro_5.0.0/montavista/pro/dekit/arm/v5t_le/target/* ./

这样就将 NFS 的共享目录建立好了，最后通过编译 /etc/exports 文件来实现目录间映射关系的建立，并且在 /etc/exports 文件中添加如下的命令行：

/home/davinci/workdir/filesys *(rw,sync,no_root_squash)

4.2.4 设置 GCC 的交叉编译环境

在虚拟机系统中找到 ~/.bashsrc 文件，并点击进入后，使用 vi 来编辑，在该文件中添加下面内容：

PATH="/opt/mvl_pro5.0.0/montavista/pro/devkit/arm/v5t_le/bin:

/opt/mvl_pro5.0.0/montavista/pro/bin:

/opt/mvl_pro5.0.0/montavista/common/bin:$PATH"

按“ ESC ”键，并输入“：”字符，然后输入“wq”，再输入“ ENTER ”键，保存文件并退出。切换到 Linux 系统的命令行界面，然后输入下面命令使之生效：

$ source ~/.bashrc

到这一步后，就完成了 GCC 的交叉编译环境的设置。

4.3 uImage 和 Bootloader 的烧写

4.3.1 Bootloader 简介

Bootloaer 是系统上电后运行的第一段程序，具有极其重要的作用，其在系统中的位置如图 4.3 所示，能为后续系统的启动提供一个良好的环境。在嵌入式系统开发中，常见的 Bootloaer 有多种，论文使用最常见的 U-Boot 。

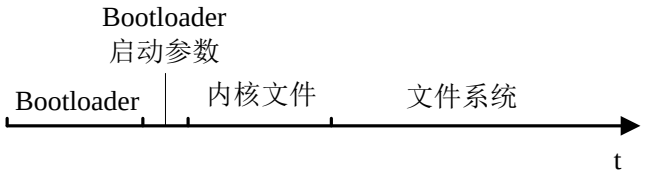

图 4.3 Bootloader 最先启动

Fig.4.3 Bootloader first starts

4.3.2 NAND FLASH 烧写

具体步骤如下：

①创建文件夹，并把这三个文件复制进该文件夹内，这三个文件的名称为：sfh_DM644x.exe 、UBL_DM6446_NAND.bin 、u_boot_1.2.0_DM6446_NAND.bin 。

②需要改变核心板上面的拨码开关的位置，使其能进入烧写状态，首先将 s2 的 COUT0 口，即第 0 端以及 COUT1口，即第 1 端设置为开，并将 COUT2 口，即第 2 端以及 COUT3 口，即第 3 端设置为关，其余剩下的端口都设置为开。

③需要把第六个双针插座的第 3 端与第 4 端进行连接。

④点击程序 cmd.exe ，并利用其找到 sfh_DM6446x.exe 的地址，输入指令进入其中，接通核心板的电源，并且输入命令如下：

sfh_DM644x.exe -flash UBL_DM6446_NAND.bin u-boot-1.2.0_DM6446_NAND.bin

⑤保持状态直到烧写结束，然后断掉核心板的电源，并重置核心板上的拨码开关，将 s2 的 COUT0 口，即第 0 端以及 COUT1口，即第 1 端设置为关，剩下的端口保持原状，接着将核心板的电源接上，这样就完成了 uImage 和 U-Boot 的烧写工作。

4.4 设置 U-Boot 环境变量

在完成 U-Boot 和 uImage 的烧写工作后，需要设置 U-Boot 的启动参数才行，否则将影响软件开发平台的工作。 U-Boot 的设置应系统方案的不同而不同，具体按照需求而定。

4.4.1 U-Boot 命令介绍

开发人员可以设置、修改、添加或者删除 U-Boot 中的环境变量，环境变量有很多，比如网卡的 MAC 地址、系统启动参数等等，具体设置需要根据开发人员所设计的操作系统方案来确定。论文所设计的软件方案里，将 U-Boot 的环境变量平时存放于 NAND FLASH 里面，在 U-Boot 上电启动后，再将这些环境变量复制到 RAM 中，从而防止这些环境变量掉电消失。

论文将使用 setenv 命令来对系统的环境变量进行设置，当然该命令也可以用于对已经存在的环境变量进行删除。具体的区分是看 setenv 后面携带有几个参数，如

果只有一个，则表示命令为删除，如果携带多个，则除了第二个是名称外，其余都是用来补充说明。setenv 无法用来存储环境变量，因此需要使用 saveenv 来存储已经设置好的环境变量，使得系统重启后已设置的环境变量仍然有效。

4.4.2 U-Boot 环境变量介绍

论文系统中的 U-Boot 需要设置的环境变量主要有以下这几个： baudrate 、ethaddr 、 bootcmd 、 ipaddr 、 serverip 、 bootfile 和 bootargs 。

①baudrate：该变量主要用于设置串口控制台的波特率大小，在论文中，将其设置为115200 。

②ethaddr 和 ipaddr：两者分别用来设置核心板的 MAC 地址和 IP 地址，论文中，将核心板 IP 地址为 192.168.136.229。

③bootcmd：该变量用来设置自动启动时执行的命令，论文中，设置了启动的地址。

④ serverip ： 该变量用来设置虚拟机的 IP 地址，论文中，将其设置为192.168.136.151。

⑤bootfile： 该变量用来设置缺省的下载文件。

⑥bootargs： 该变量用来设置传递给内核 MontaVista Linux 的启动参数，设置好后会送入 Linux 内核中。

论文具体的 U-Boot 环境变量的设置参见附录 C。

到这里为止，就实现了平台的成功建立，系统启动图见图 4.4。在第五章的障碍物检测以及全景图像处理等算法的程序开发过程中， Red Hat Linux 虚拟机将用来做具体程序的开发、调试以及编译工作，而核心板主要用来做最终程序的运行测试。

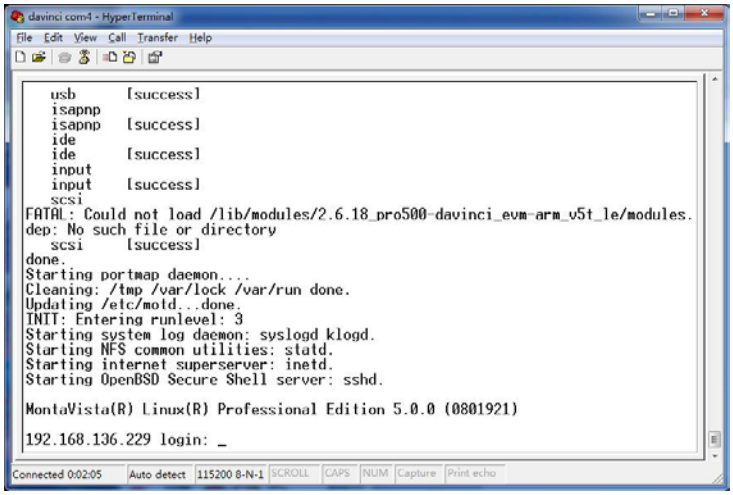

图 4.4 系统启动图

Fig.4.4 System starts figure

4.5 本章小结

本章节主要完成了系统的软件开发平台的搭建，为下一章障碍物检测和全景图像拼接算法的研究提供了开发平台。具体来说，首先介绍了系统的软件开发平台的组成，然后搭建了开发 DM6446 芯片应用程序所需的环境支持，包括虚拟机 Linux 操作系统的安装，并在该系统下安装了 MontaVista Linux 内核、DVSDK 工具链，同时，设置了 NFS 服务器和 GCC 交叉编译环境，接着详细叙述了烧写 Bootloader 和 uImage 的步骤，最后设置了 U-Boot 的环境变量。

5 障碍物检测和全景图像拼接算法研究

5.1 软件算法结构框图

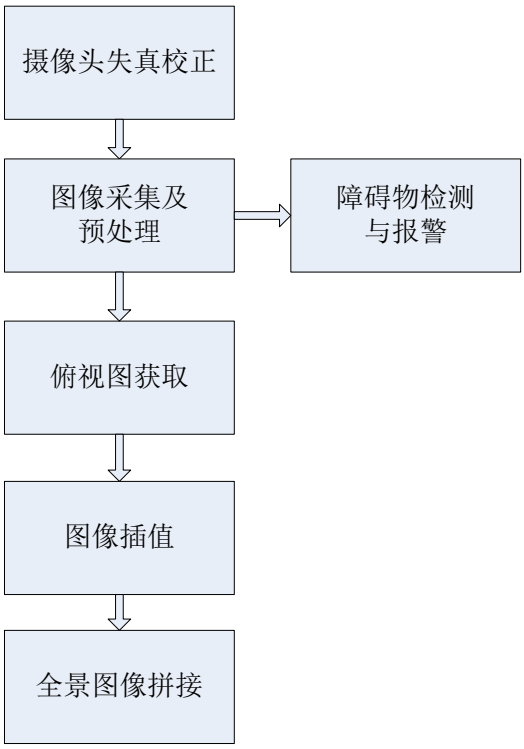

图 5.1 软件框图

Fig.5.1 Software block diagram

系统的软件算法结构框图如图 5.1 所示，主要分为六个部分，现分别简要叙述下这六个部分。

5.1.1 摄像头失真校正

摄像头失真校正主要是解决广角摄像头图像畸变问题，使用相关文献方法来解决问题，利用张正友法来求解摄像头的焦距和偏移参数[42~43]，利用 Browm 的方法来求解畸变参数[55]。具体过程与结果参见 5.6.2 小节。

5.1.2 图像采集及预处理

图像采集主要是完成对四个广角摄像头的图像信号的采集与解码，图像预处理主要是解决采集过程中的噪声问题、白平衡问题等。图像采集及预处理的结果可通过液晶显示屏显示也可通过电脑显示屏显示，在前期的算法研究和实验过程中，主要将显示结果输出到电脑显示屏，更利于发现问题的所在，在后期的系统

总体测试时，将系统的显示结果输出到液晶显示屏上显示。当电脑显示屏能显示出图像时，既可验证硬件系统设计方案的可行性与系统的可靠性，又可验证预处理算法的可靠性，具体的实验过程与结果参见 5.6.1 小节。

5.1.3 障碍物检测与报警

根据汽车障碍物具有独特的特征，这些特征包括：左右两边车体对称，车体上存在诸多的垂直与水平框架、线等，以及汽车底部由于车体遮挡而存在的阴影，这些特征正是论文所设计的障碍物检测方法的检测关键点。通过障碍物检测方法能快速检测出安全范围内的汽车障碍物的存在，并发出报警信号。算法的具体实验过程和结果参见 5.6.3 小节。

5.1.4 俯视图获取

由于摄像头的光轴不与地面垂直，因此，在获取俯视图时需要进行逆投影变换，将汽车前后左右的图像通过逆投影变换转变为俯视图。俯视图获取的具体实验结果参见 5.6.4 小节。

5.1.5 图像插值

图像插值主要解决通过逆投影变换后图像的像素点的灰度值（YUV 色彩空间中的“Y”值）不是整数的问题，论文中主要使用双线性插值法，该方法已是成熟算法，因此，其具体的算法分析论文不在阐述。

5.1.6 全景图像拼接

全景图像拼接是将已得到的汽车前后左右的四幅俯视图进行整体拼接，得到汽车的俯视全景图像，其具体的实验过程与结果参见 5.6.4 小节。

5.2 障碍物检测与报警

5.2.1 障碍物特征描述

由于论文在之前的章节中已经将障碍物定义为车辆，由于汽车障碍物具有独特的特征，因此，以汽车特征作为论文的障碍物检测方法的判断标准。正如上一小节所介绍的，汽车具有诸多特殊的特征，现分别总结汽车所具有的这些特征。

①左右对称性

汽车是具有良好左右对称性的交通工具，目前，道路上几乎不存在非对称性的汽车，因此，对称性是一个可以利用的检测特征。具体到本系统上，在构成的全景图像中，汽车的左右对称性是相对的，只有从本车前方或者后方出现的车辆才满足左右对称性，而从本车左右两端驶近的车辆，由于摄像头只能采集到该车的侧面信息，因此不满足左右对称性，换句话说，也就是只有车前端与后端的摄像头才能检测到具有左右对称性的汽车，而车身左右两端的摄像头就无法检测出

具有左右对称性的障碍物汽车。因此，将左右对称性特征作为前、后两个摄像头检测汽车障碍物的判断标准之一，而车身左右两个摄像头不采用此判断标准。

在使用左右对称性作为检测判断标准时，容易遇上如下问题：首先，由于噪声的存在，会干扰对称性的判断，从而出现误判或者是漏判；其次，对称性容易受到其它具有对称性事物的干扰，比如具有对称性的背景；再次，障碍物汽车本身的对称性也容易被其它事物干扰而遭到破坏，比如一部分车身被其它事物遮挡。因此，左右对称性无法作为特征判断的唯一标准。

②车底阴影

由于车体的遮挡，光线无法照射到汽车底部，因此汽车底部会存在阴影，而在通常的柏油马路上，其车底阴影较突出，很容易识别观察到。根据 Mori 的证明，车底的阴影不会被阳光照射到，其亮度值会在一天里基本上没变化[56]。因此，可以将车底阴影作为检测判断标准之一，并使用固定阈值来检测。

③车体垂直与水平边缘

汽车本身的构造让它包含有很多垂直和水平的结构，尤其是从车辆的前方或者是后方看过去，比如车架、前后视窗等等。因此，可以将车体的垂直与水平边缘视作汽车障碍物检测的判断标准之一。

基于上述分析，论文将使用车体左右对称性、车底阴影以及车体垂直与水平边缘三种特征相结合的检测方法进行汽车障碍物的检测。

5.2.2 障碍物检测算法流程图

由于论文系统的设计要求，使得障碍物检测算法必须要具有快速性和高可靠性，这就要求算法必须简单、易实现。论文所设计的障碍物检测算法的基本步骤主要如下：

①针对汽车的四个摄像头，采用单位像素灰度（YUV 色彩空间的“Y”明亮度）等级来检测路面，确定有无障碍物车底阴影，如果存在阴影，对阴影进行初步定位，并进行下一步。

②对障碍物阴影进行再检测，此时，会在下一帧图像中寻找障碍物阴影。而为了排除噪声造成的阴影干扰，下一帧图像会主要检测上一帧图像“阴影”像素点附近的像素点的灰度等级，即阴影邻域检测。因为，如果是噪声干扰，下一帧图像阴影邻域内一般不会再出现阴影，而如果确实是障碍物，则由于汽车的移动，图像阴影会移动到上一帧阴影像素点的邻域[57]。

③对障碍物进行边缘检测，主要检测障碍物是否存在水平或者垂直的边缘，即图像中的障碍物边缘像素点是否水平、垂直排布。

④由于只有前后摄像头采集到的汽车障碍物才可能具有左右对称性特征，因此，对图像来源进行判断，判断是否是来自前后摄像头采集的图像，如果是前后

摄像头采集的图像，则对障碍物进行左右对称性检测，如果是左右摄像头采集到的图像，则不进行障碍物的左右对称性检测。之所以在设计上对前后摄像头的障碍物多一道检测过程，是为了增加检测的可靠性，因为在道路上行驶的汽车，绝大多数来自于本车的前方或者后方，直接从左右两方出现汽车的几率不大，左右两方的汽车一般是先驶入前后摄像头的视线范围再进入左右两方的视线范围。

⑤对前后摄像头采集到的图像进行障碍物左右对称性检测，如果满足对称性要求，则判断为汽车障碍物。

⑥根据前面障碍物检测中车底阴影定位的结果确定汽车障碍物的位置，将其与预先设置的安全距离线进行比较，如果距离小于或等于安全距离，则向驾驶员发出声音报警信号，如果距离大于安全距离，则继续检测，不发出报警信号。

论文所设计的障碍物检测算法流程图具体内容参见图 5.2。

5.2.3 安全距离线的确定

由于安装在汽车上的四个广角摄像头的安装高度、位置以及摄像头光轴与地面的夹角等数据是已知的，按照论文第二章世界坐标系与图像坐标系之间的转换关系，在已知摄像头距地面高度 h，摄像头在竖直方向和水平方向上的视角大小 2α 、 2β，摄像头光轴 OG 与地面的夹角 γ，所得图像的长与宽为 H 和 W 的情况下，按照式(2.7)，能很容易地确定摄像头视线范围内某一点在图像中的具体位置。因此，可以在世界坐标系下（即真实的车子四周环境下）设定距本车的安全距离（安全距离的数值可由实际情况确定），并将该距离变换到图像坐标系下，设定图像中的安全距离。当真实情况下的汽车障碍物越过安全距离线后，其图像也在图像坐标系中越过了安全距离线。因此，只要确定了汽车障碍物在图像中的位置，即能判断该车是否对本车有危险，以及是否需向驾驶员报警。

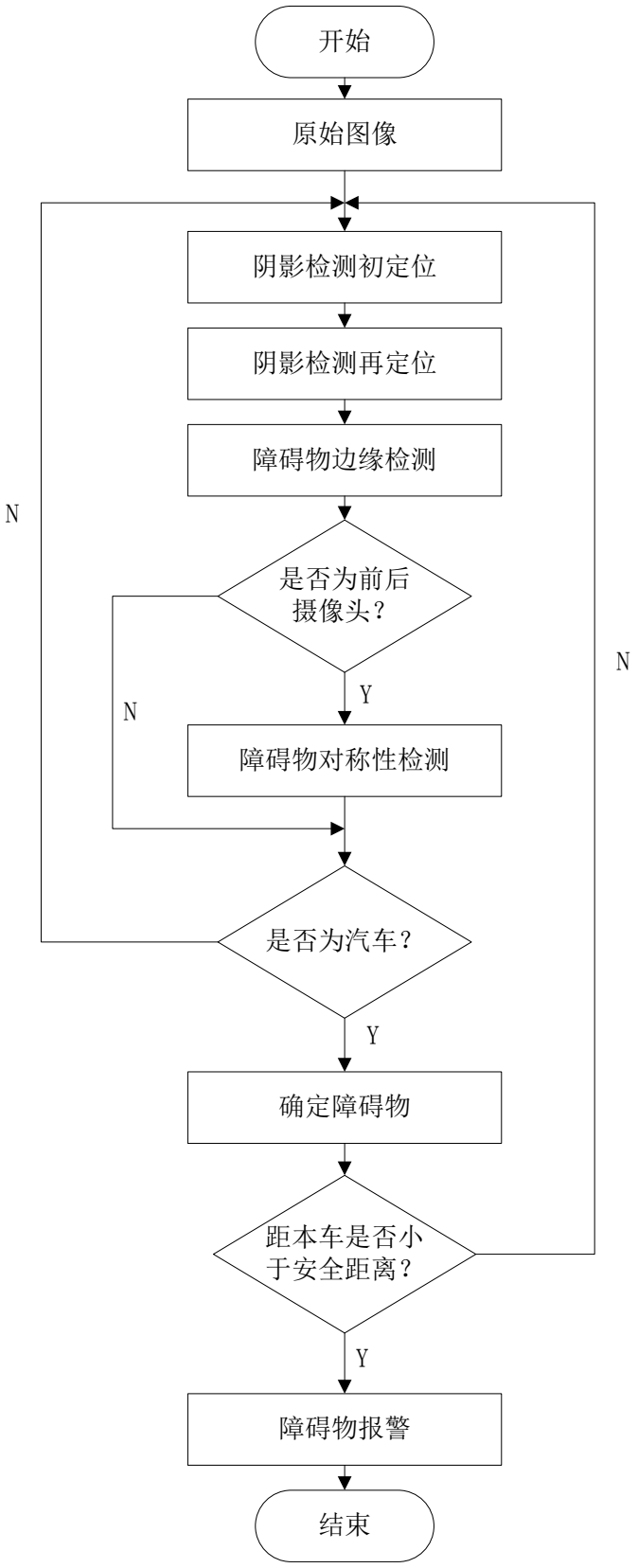

图 5.2 障碍物检测算法流程图

Fig.5.2 Obstacle detection algorithm flowchart

5.3 俯视图的获取

由于受限于摄像头的安装位置，同时，既要使摄像头能看得远，又要使摄像头采集到的图像能变换为俯视图，这就需要摄像头在安装时保持一定的向下夹角，既不能垂直于地面，也不能水平于地面。这样就会使采集到的视频图像具有透视效果，而为了消除图像的透视效果，需要对摄像头采集到的图像做逆投影变换。

将具有透视效果的图像进行变换的办法常用的有三种：第一种是采用空间射影变换的算法[58]、第二种是采用图像单应性矩阵的 DLT 算法[59]，第三种是采用摄像头视角参数的算法[60]。由于要求算法简单有效，而第一种方法与第二种方法的步骤较麻烦，算法较复杂，不适合本系统。第三种方法其视角变换过程较简单，同时，又由于已知系统所使用的摄像头的参数。因此，论文决定采用基于摄像头参数的视角变换算法，其算法流程图如图 5.3 所示。

5.4 全景图像拼接

得到汽车前后左右四个摄像头的俯视图后，现在，需要将这些俯视图全部拼接起来，生成汽车的全景俯视图。对于全景俯视图的拼接算法，论文决定采用基于坐标系转换的方法。具体的算法思想是：首先建立全景地面坐标系与汽车四个方向的图像坐标系，因为摄像头的视角参数是已知且固定的，所以很容易建立全景地面坐标系与汽车四个方向的图像坐标系之间坐标点的相互对应关系，通过该对应关系，能够确定某一全景地面坐标系下的点在相邻两个图像坐标系下的坐标值，然后选择合适的这些点作为图像拼接点，并通过多个拼接点明确图像拼接缝的位置，图像拼接缝在相邻两个待拼接图像中是相同的，因而能用拼接缝实现图像的拼接。

由图 3.3 可知，选用的四个广角摄像头会有视线的重合区域，拼接将会主要利用到这部分重合区域，因此需要建立重合区域在各自摄像头图像坐标系下的位置信息。现在建立全景地面坐标系与汽车四个方向的图像坐标系，并建立地面坐标系与图像坐标系之间的转换关系。

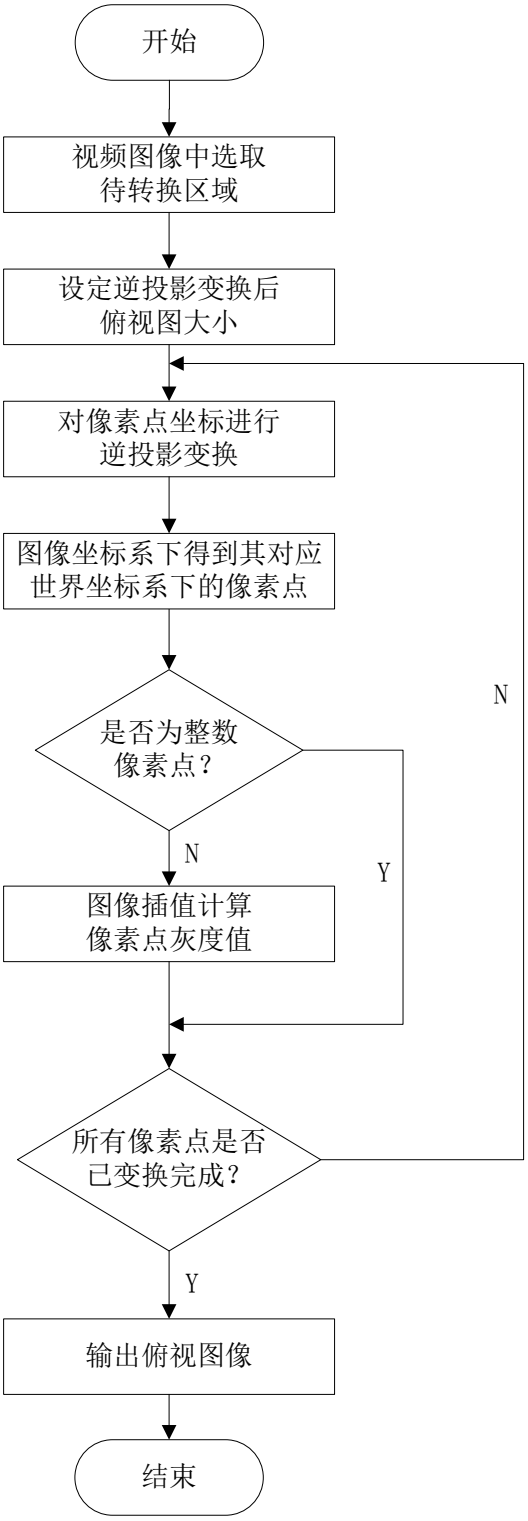

图 5.3 俯视图获取算法流程图

Fig.5.3 The algorithm flowchart of getting a top view

5.4.1 坐标系的建立与转换关系

①全景地面坐标系

全景地面坐标系是整个全景图像拼接的关系，其它图像坐标系都是与之相关联的。坐标系的建立如下：首先令全景图像的几何中心点O点作为整个坐标系的原点（该点大致位于汽车的几何中心位置），然后将汽车行驶方向设为坐标系的Y轴，汽车车头方向设为Y轴的正方向，接着将水平垂直于Y轴的方向设立为坐标系的X轴，并且将车身右侧设立为X轴的正方向，坐标系详细参见图5.4。该坐标系为真实坐标系，故其基本单位为米。

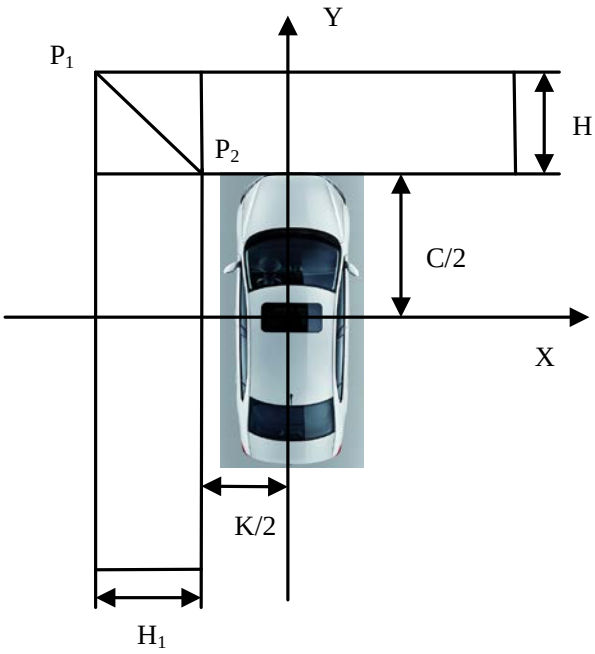

图 5.4 全景地面坐标系

Fig.5.4 Panoramic ground coordinate

假如汽车长度是C米，汽车宽是K米，汽车左右面的摄像头视视线宽度为H_1米，汽车前后面的摄像头视线宽度为H米。点P_1和点P_2是汽车前面摄像头与左面摄像头视线重叠区中的两个点，它们的坐标分别是$P_1(X_1,Y_1)$与$P_2(X_2,Y_2)$，很容易由这两个点得到直线P_1P_2。由线性映射不变性能够得到：点P_1和P_2就能唯一确定一条直线P_1P_2，而这条直线在前面摄像头图像坐标系和左面摄像头坐标系中均是共线的，可以得到，直线P_1P_2就是前面摄像头视线与左面摄像头视线的拼接缝。

由于全景地面坐标系并不是世界坐标系，二者存在一定的不同，因此，需要建立两者的转换关系，才能按照第二章的介绍，将图像坐标系与真实坐标系结合起来。假设全景坐标系存在着一个点$P(X,Y)$，而世界坐标系存在着点P的对应点P'，其坐标为$P'(X',Y')$，很容易得到两者之间的关系如式(5.1)所示。

$$\begin{cases} X' = X \\ Y' = Y - \dfrac{C}{2} \end{cases} \tag{5.1}$$

②前向图像坐标系

汽车前后左右四个方向均要先进行逆投影变换然后得到俯视图，四个方向的坐标系建立方法大致相同，只是与全景地面坐标系的关系不相同。因此，现在以前向坐标系的建立为标准介绍四个方向图像坐标系的创建过程。

由第二章可知，逆投影变换是图像坐标系与世界坐标之间的转换，因此，将公式(5.1)带入第二章的逆投影变换公式，即能够把前向视野变换为俯视图。按照图5.5，设立前向图像坐标系，并规定该坐标系以水平地面为平面，汽车行驶方向为Y轴，车身右侧为X轴，原点取汽车前向摄像头的水平投影。

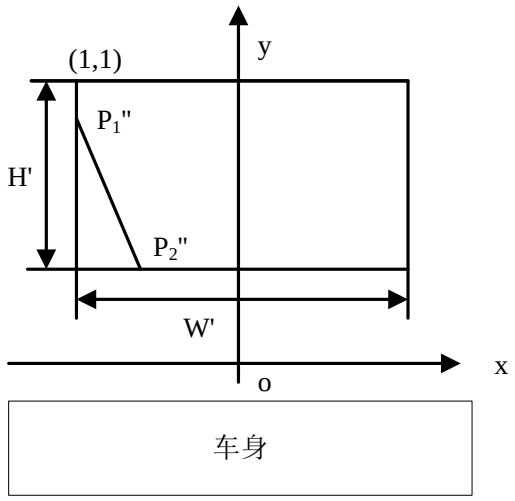

图 5.5 前向图像坐标系

Fig.5.5 Forward image coordinate

很容易得到全景地面坐标系同前向图像坐标系相互之间的转换式，其中，假设全景地面坐标系中有个点为$P(X,Y)$，式(5.2)如下：

$$\begin{cases} X = R_j * \dfrac{W}{W'} - \dfrac{W}{2} \\ Y - \dfrac{C}{2} = \dfrac{H}{2} - R_j * \dfrac{H}{H'} + \dfrac{h}{\tan\gamma} \end{cases} \tag{5.2}$$

式中，H与W分别代表道路的真实长度与宽度，H'为得到的俯视图$R(i,j)$的长，W'为得到的俯视图$R(i,j)$的宽。通过变换，可将式(5.2)转变为式(5.3)：

$$\begin{cases} R_i = \dfrac{H'}{H} * (\dfrac{C+H}{2} + \dfrac{h}{\tan\gamma} - Y) \\ R_j = \dfrac{W'}{W}(X + \dfrac{W}{2}) \end{cases} \tag{5.3}$$

假设全景地面坐标系存在有点P_1与点P_2，可以由式(5.3)得到它们位于正向坐标系下的点$P_1"$与点$P_2"$的像素坐标值$R_1"$与$R_2"$，式如(5.4)所示。

$$\begin{cases} R_1" = (\dfrac{H'}{H} * (\dfrac{C+H}{2} + \dfrac{h}{\tan\gamma} - Y_1), \dfrac{W'}{W} * (X_1 + \dfrac{W}{2})) \\ R_2" = (\dfrac{H'}{H} * (\dfrac{C+H}{2} + \dfrac{h}{\tan\gamma} - Y_2), \dfrac{W'}{W} * (X_2 + \dfrac{W}{2})) \end{cases} \tag{5.4}$$

5.4.2 图像拼接步骤

全景图像是由汽车前后左右四个方向的俯视图像构成的，拼接顺序为前向俯视图像与左侧俯视图像拼接，拼接后得到拼接俯视图 1，再用拼接俯视图 1 与后向俯视图像进行拼接，拼接后得到拼接俯视图 2，再用拼接俯视图 2 与右侧俯视图像进行拼接，得到汽车的全景图像。现在以前向俯视图像和左侧俯视图像拼接过程为例介绍拼接算法流程图，如图 5.6 所示。

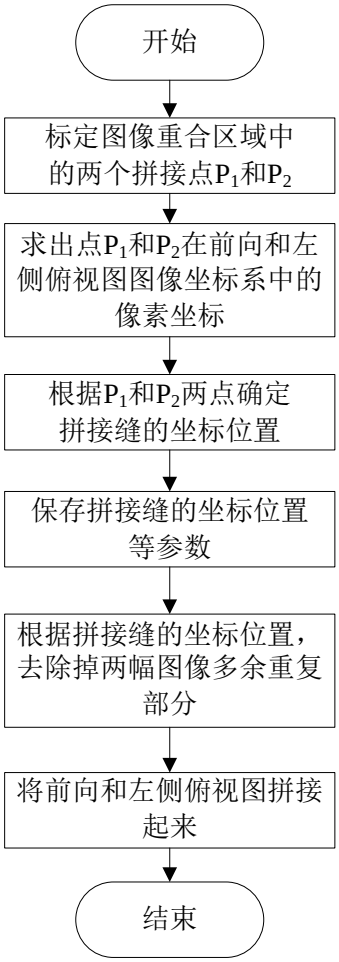

图 5.6 图像拼接流程图

Fig.5.6 Image stitched flowchart

在全景图像拼接过程中，由于只是对图像的像素坐标进行变换，而未针对像素的 YUV 色彩空间值进行改变，并且全景图像完成图的该点像素的色彩空间值就是其对应的原图像的像素色彩空间值，因此，只要建立原图像与全景图像完成图之间像素点的对应关系，即可将原图像的像素点的 YUV 色彩空间值赋值给全景图像。又因为摄像头的位置信息已知，很容易得到原图像与全景图像完成图像素点之间的关系，而为了减少运算，可以在安装完摄像头后，将摄像头位置信息输入，并在算法中形成像素位置映射表，从而在程序执行时，直接通过查找像素位置映射表，获取全景图像完成图相应位置的 YUV 色彩空间值，如图 5.7 所示。

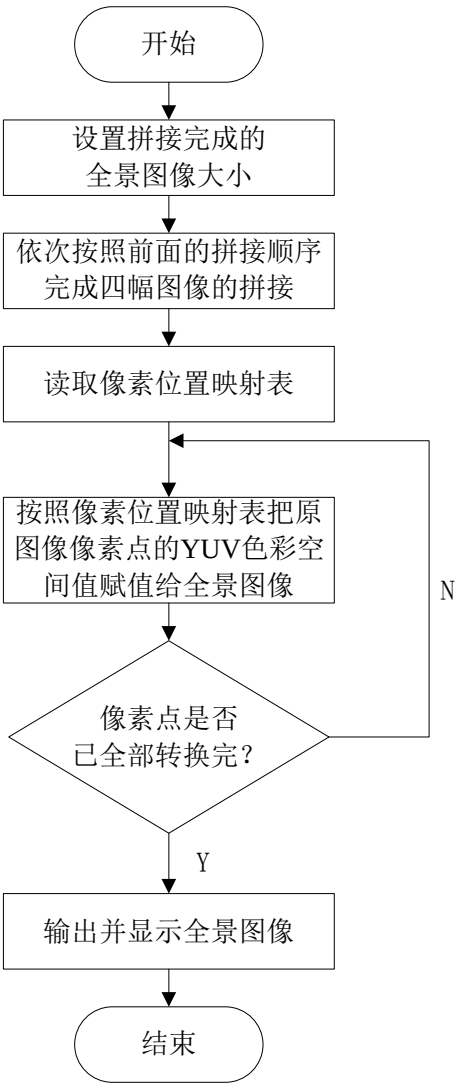

图 5.7 全景图像拼接流程图

Fig.5.7 Panoramic image stitched flowchart

5.5 本章小结

本章节主要完成了障碍物检测和全景图像拼接算法的理论研究以及相关算法分析。具体来说，首先根据系统需求，设计了软件算法的结构框图；然后根据障碍物检测与报警的需求，提出采用基于特征的障碍物方式来检测汽车障碍物；为了得到汽车周围的图像俯视图，提出了俯视图的获取算法；为了将四个广角摄像头拼接为全景图像，提出了基于坐标变换的全景图像拼接算法。本章节为下一章节实验提供了理论基础。

6 实验与测试

6.1 实验测试过程与结果分析

6.1.1 系统平台运行实验

系统平台运行实验主要用来检测硬件系统设计方案的可行性和硬件的可靠性，即验证核心板是不是能够完成图像采集及处理工作。

本实验设备主要由三部分组成，包括四个广角摄像头、核心板、电脑、液晶显示屏。

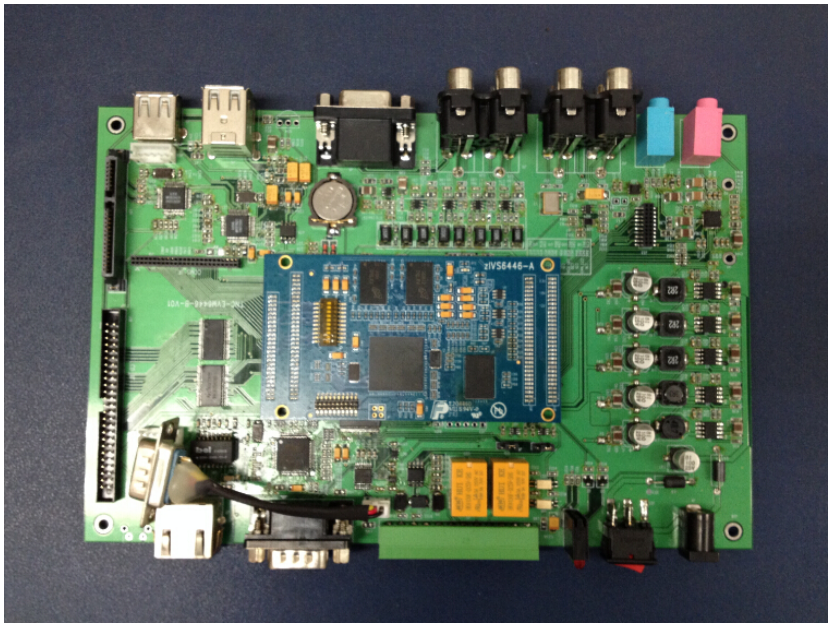

图 6.1 核心板图

Fig.6.1 Core board picture

为了便于对算法进行实时地修改、调试、验证，将采用 NFS 挂载文件系统的形式进行实验。实验测试前需要解决以下两部分内容：

①按照论文第四章系统软件开发平台的叙述内容，完成系统的软件开发环境的组建。

②编写、编译并下载完成图像采集算法与图像显示算法。

图 6.2 实验装置连接图

Fig.6.2 Test equipments connections picture

实验具体步骤如下：

①建立超级终端。点击运行超级终端软件，然后创建新的端口连接，将其名称取为 davinci com4。设置 COM4 口的属性，将波特率设置为 115200，这个要与 U-Boot 环境变量中的 baudrate 变量的设置相同，同时把数据位定成 8 位，并确定不进行奇偶校验，然后把停止位确定为 1，最后将串口确定为不进行数据流控制，如图 6.3 所示。

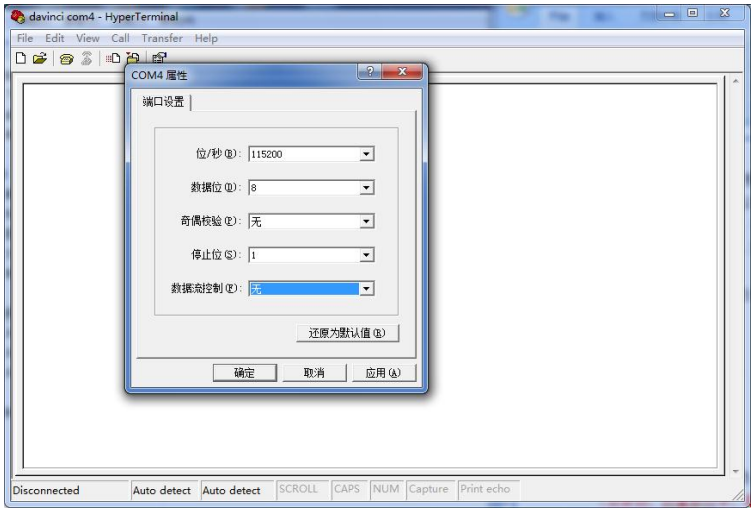

图 6.3 超级终端 COM4 口设置

Fig.6.3 Hyper Terminal COM4 port settings

设置完毕后就能够看见超级终端的界面了，见图 6.4。

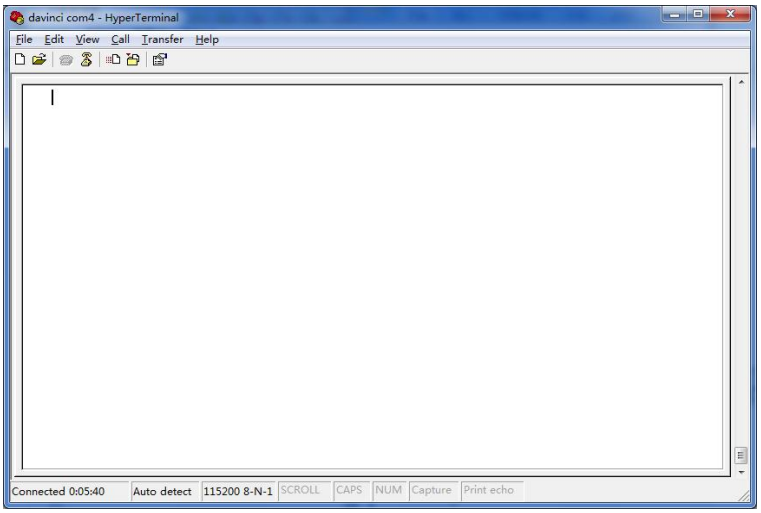

图 6.4 超级终端界面图

Fig.6.4 Hyper Terminal interface map

②启动虚拟机 Linux 系统。点击 VMware Player 软件，并选择 Linux 虚拟机然后运行，采用 root 身份登陆。

③核心板上电并进入系统，当出现提示 192.168.136.229 login：的时候，键入用户名：root，以 root 账号进入系统，进入系统后可看见等待输入命令，见图 6.5。

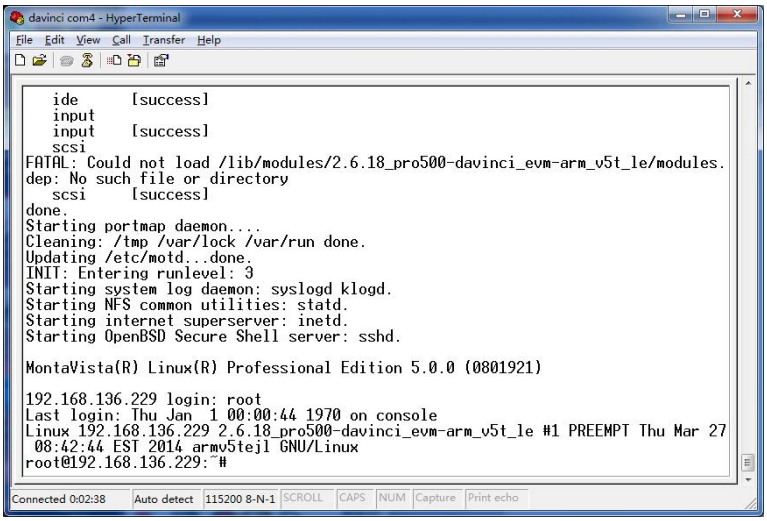

图 6.5 超级终端下命令行输入

Fig.6.5 Hyper Terminal command lines inputs

④向超级终端的命令行界面键入下列命令进入图像处理程序所在目录。

$ cd /root/TNC-EVM6446/tvp5158

显示结果见图 6.6。

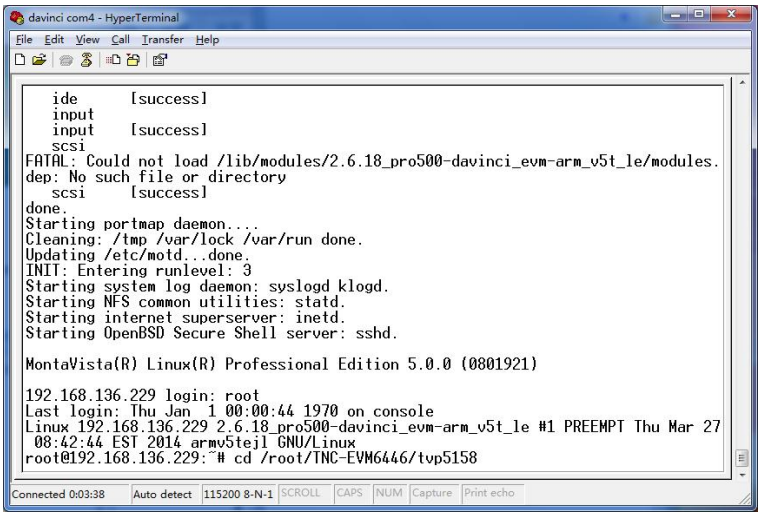

图 6.6 超级终端下进入图像处理程序所在文件夹

Fig.6.6 Enter the folder where the image processing programs exist by Hyper Terminal

⑤超级终端下输入下列命令运行图像处理程序。

$ sh ./mcload.sh

$ ./mc_test.out

超级终端显示结果如图 6.7 所示。

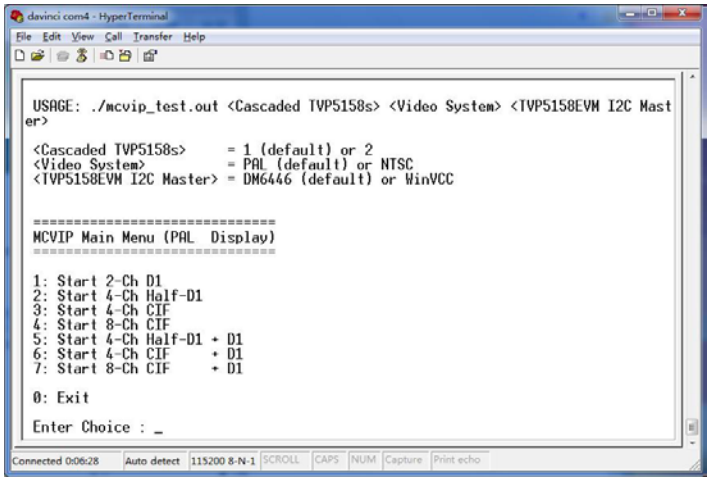

图 6.7 运行图像处理程序后结果

Fig.6.7 Results of running the image processing programs

具体来说，选项 1 是单摄像头输入，只显示一个摄像头的图像信号；选项 2 是单摄像头半屏显示，只显示屏幕一半图像；选项 3 是 4 分频显示，将屏幕分为 4 个部分，用于四个摄像头同时显示；选项 4 是全景图像显示；选项 5-7 是预留功能选项，暂时无设计安排；选项 8 是退出程序。

⑥选择选项 1，观察单摄像头图像。图像如图 6.8 所示。

图 6.8 单摄像头图像

Fig.6.8 Single camera image

⑦选择选项 3，观察 4 个摄像头图像。图像如图 6.9 所示。

图 6.9 4 路摄像头图像

Fig.6.9 Four cameras images

由单摄像头采集的实验结果可以看出，系统能够实现对单摄像头的图像进行采集，并完成摄像头失真校正及图像预处理工作。为了验证 4 路摄像头信号能够同时被处理器处理，同时为了检测系统的响应时间是否满足系统性能指标规定，

在每路摄像头图像的右下方显示时间，这样就能看出每路摄像头采集图像的时间，从而查看出有无延迟。由 4 路摄像头同时采集的实验结果可以看出，4 路摄像头的显示时间是一致的，没有明显的延迟，图像显示也正常，无问题。从而验证了硬件系统设计方案正确，硬件电路焊接无误，图像采集和显示算法没有什么大问题。

6.1.2 摄像头失真校正实验

摄像头失真校正实验主要是解决广角摄像头图像失真问题，由于已知摄像头的参数，故采用参数法来校正摄像头的畸变。具体来说，使用张正友法来求解焦距和偏移参数，使用 Browm 方法来进行畸变参数的求解。由于相关文献中已经详细介绍了具体的数学推导过程，因此，论文省略这部分内容。

具体的失真校正实验步骤如下：

①将黑白棋盘格标定纸平铺于地面上，利用广角摄像头采集标定在不同角度的图像约 25 张；

②对采集到的图像进行处理，将相邻黑格的公共点作为角点，然后提取角点。

③将已得到的摄像头参数带入编好的 Matlab 软件校正算法程序中，部分算法见图 6.10。

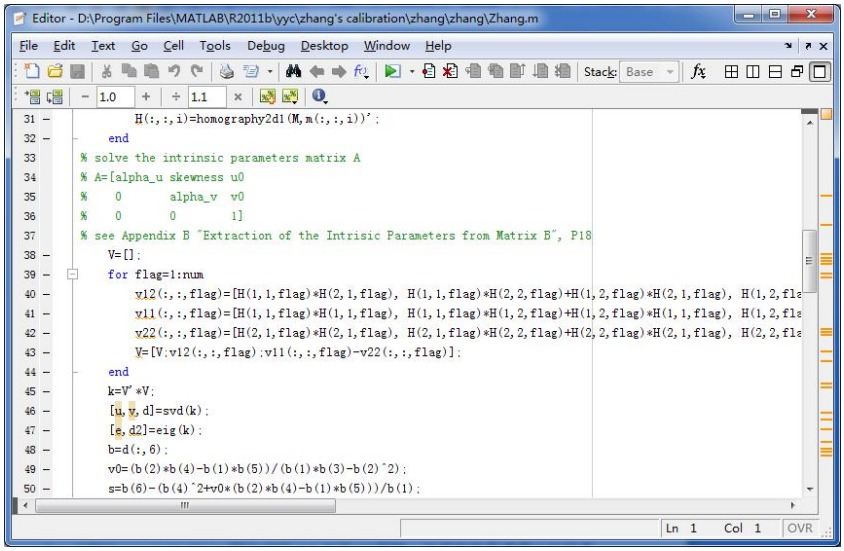

图 6.10 张正友校正法

Fig.6.10 Zhang's calibration

④求解得出校正后的摄像头内部参数和摄像头外部参数。

⑤将校正数据写入图像处理程序，用于解决摄像头失真问题。

运用参数法校正摄像头图像后，得到处理后的畸变图像，如图 6.12 所示，将其与处理前的畸变图像（如图 6.11 所示）进行对比。可以看出算法对于广角摄像头的失真校正效果还是挺不错的。

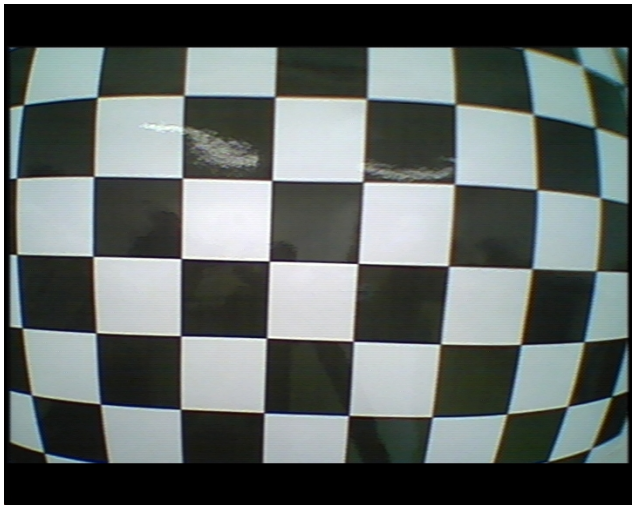

图 6.11 畸变图像处理前

Fig.6.11 Distorted image before processed

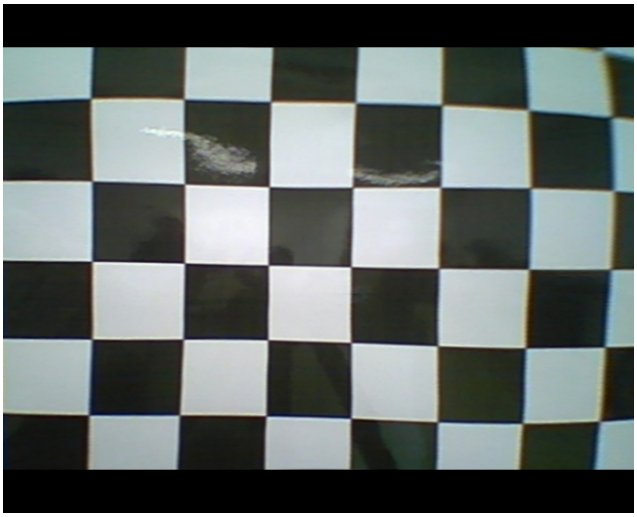

图 6.12 畸变图像处理后

Fig.6.12 Distorted image after processed

6.1.3 障碍物检测与报警实验

障碍物检测程序是与全景图像拼接程序同时并行进行的，并且，障碍物检测是利用原视频图像，即只进行了摄像头失真校正而没有进行过逆投影变换的图像。

由于论文设计的障碍物报警是在汽车障碍物进入安全距离后才发出，因此实验中需要划定安全距离，并将其定为 4m。对于安全距离的测定，采用坐标变换的方法，由于摄像头的安装高度、位置和光轴同地面夹角已知且固定，可以通过建立图像坐标系和世界坐标系相互的变换关系（具体坐标变换关系见论文 2.3 节），从而将真实的 4m 安全距离变换成图像中的安全距离，确定图像中的安全距离线。

当图像中的障碍物位于安全距离线以内时，就发出报警信号，即对应于世界坐标系下障碍物距车身不足 4m。

因为论文系统设计中只需对小于安全距离的障碍物进行声音报警，而无需显示障碍物检测的画面，只在液晶屏上显示全景图像，所以，障碍物检测在设计中无画面显示，也无跟踪框。为了能够测试算法的可靠性，需要进行以下修改：

①只利用单个摄像头对汽车障碍物做检测。

②摄像头采集图像显示在笔记本电脑上，同时在图像上“画出”世界坐标系下的 4m 安全距离线。

③采用固定摄像头检测车道上运动汽车的方式来模拟汽车行驶中车辆的靠近。

具体实验方案如下：

在校园车道旁放置三脚架，按照实验前确定的摄像头位置参数（摄像头高度、摄像头光轴与地面的夹角度数）固定好摄像头，并将摄像头对向车道，确保汽车驶近时距离在 4m 以内。对车道上驶近的汽车进行检测报警，并对检测结果进行统计。实验过程中采集到的图像如图 6.13 所示，统计结果如表 6.1 所示。

图 6.13 障碍物检测实验图 1-4

Fig.6.13 Obstacle detection experiments pictures

表 6.1 障碍物检测实验结果统计表

Table6.1 Obstacle detection results

内容	数目
驶近汽车数（辆）	25
驶离汽车数（辆）	17
报警数（次）	40

通过实验可以看出，进入安全距离的汽车数量总共为 42 辆，而系统报警次数为 40 次，检测正确率为 95.2%，高于系统性能指标设计时的 90%，达到系统设计要求。实验结果的具体分析如下：

由于未对阴影首次出现的位置进行判断，导致将驶离汽车也当成了驶近汽车进行报警，在实际运用环境中这种情况也是有可能出现的，比如，本车被其它汽车超车，汽车超过本车后拐进本车道的距离小于安全距离时。同时在论文设计的系统中，如果是超车，必定被汽车后方或者左侧摄像头先捕捉到画面，并已进行障碍物检测报警，而如果超车汽车再驶入前侧摄像头区域，则会导致二次报警。同时，由表 6.1 可以看出，还是漏检两次，具体来说，应该是两辆车距离过近，或者是错车，导致安全距离内一直有汽车障碍物，如图 6.13 所示，第一张图是对车道的检测；第二张图在安全距离内有一辆驶离的黑色汽车，由于它车体有部分在安全距离内，因而促发了报警；第三张图可以看到，驶离黑色汽车还没完全离开安全距离时，对面的驶近的白色汽车已经进入安全距离，此时报警声音会一直在响，系统会默认为原来的黑色汽车还没离开安全距离；第四张图是安全距离内只有驶近的白色汽车，报警声音继续响。从上述实验结果上来说，论文设计的障碍物检测算法不能具体区分安全距离内的汽车障碍物，只能判断在安全距离内的是否有汽车障碍物，但是对于行驶中的本车来说，区分不是主要的，检测安全距离内是否有危害本车行驶安全的障碍物才是主要的，因此，论文所设计的障碍物检测算法满足了设计方案的初衷的。

6.1.4 全景图像拼接实验

在进行全景图像拼接之前，首先需要得到汽车四个方向图像的俯视图，因此，需要对图像做逆投影变换。逆投影变换的过程及公式参见论文 2.3 节，俯视图的获取算法流程图参见图 5.3，论文使用的逆投影变换实际参数值参见表 6.2。

全景图像拼接实验是在实验室环境下模拟汽车的摄像头视角环境进行的。之所以采用模拟汽车摄像头视角环境，一是为了在实车测试前先做前期研究；二是在模拟环境下发现问题能够实时调整、改正；三是以三脚架模拟汽车四边，能更好地调整摄像头位置（包括摄像头角度、距离、位置等），为算法的研究带来帮助。

表 6.2 逆投影变换参数

Table 6.2 Parameters to inverse projection transformation

参数名（单位）	符号	参数值
摄像头水平视角范围（°）	2β	178
摄像头垂直视角范围（°）	2α	47
摄像头俯角（°）	γ	45
摄像头安装高度（m）	h	0.60
图像水平分辨率（ppi）	W	320
图像垂直分辨率（ppi）	H	240

实验室环境下模拟汽车摄像头视角环境搭建的实施过程为：

①在实验室放置四个三脚架，并在每个三角架上安装一个广角摄像头。如图 6.14 所示。

②将四个三角架以矩形位置放置，模拟汽车的四个边，这样相当于汽车的前后左右均有一个广角摄像头。

③调节三角架上广角摄像头的高度，以实际摄像头安装高度 0.6m 左右为准。

图 6.14 广角摄像头安装示意图

Fig.6.14 Wide-angle camera installation picture

这样就实现了在实验室环境下对汽车摄像头视角环境的模拟。在全景图像拼接实验中，首先采集 4 个三角架上的广角摄像头的图像信息，然后将图像信息转换为各个方向的俯视图，如图 6.15 所示。

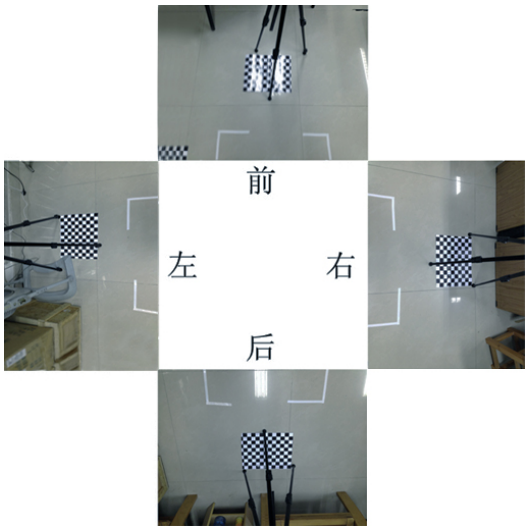

图 6.15 拼接前四面俯视图

Fig.6.15Top view of four directions before stitched

通过图 6.15 可以看出，论文采用的逆投影变换方法还是非常有效地得到了四个摄像头的俯视图信息。

得到四个方向上的俯视图后，再根据全景图像拼接算法，明确前向俯视图像和左侧俯视图像里拼接点的图像坐标，从而通过拼接点得到拼接缝的图像坐标，为了拼接后的图像能够正常显示，需要把待拼接图像的拼接缝重复一侧的图像去掉，然后通过查找像素位置映射表，将原图像的像素点 YUV 色彩空间值赋值给拼接图像对应的像素点，从而实现了前向俯视图像和左侧俯视图像的拼接。再按照全景图像拼接的顺序，依次按照逆时针方向进行拼接，并最终得到全景图像，见图 6.16。

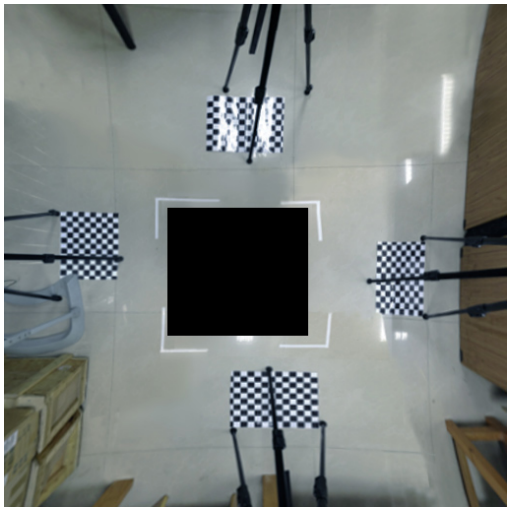

图 6.16 拼接完成的全景图像

Fig.6.16 Panoramic image after stitched

通过图 6.16 可以看出，论文采用的全景图像拼接方案能够实现图像的拼接，并能够生成全景鸟瞰图。生成的全景鸟瞰图包含全部四个摄像头采集到的图像，无缺失，满足系统设计中对全景图像拼接完整性的性能指标要求。同时，可以看出，图像清晰完整，全景图像边缘虽然有一点失真现象，但是失真现象不影响观察，故满足设计要求。

6.2 系统整体实车实验

在进行系统的整体实车实验之前，需要完成几项准备工作，包括文件系统固化与程序烧写、摄像头安装与标定等，这些工作是将系统由实验室环境转换到实车测试环境所必须的。

6.2.1 文件系统固化与程序烧写

由于在实验室环境下整个系统采用的是 NFS 挂靠模式，也就是说需要使用 PC 机提供 Linux 文件系统，而在系统的整体实车实验中，将无法使用到 PC 机，故需要将文件系统以及程序烧写并固化到系统核心板上。论文所使用的文件系统固化将采用虚拟内存盘（ramdisk）来实现，具体步骤参见附录 D。

6.2.2 摄像头安装、标定

四个广角摄像头的安装位置在论文第三章的 3.3.2 小节摄像头的选型中已经谈及，具体到本实验中，前后摄像头均安装在车前后牌照上方，左右摄像头安装在车左右后视镜的下方。摄像头的安装角度需要通过黑白标定纸的标定，确定摄像头所能看到的左右最大角度，以及最远距离。同时，需要确保相邻摄像头存在重合的视野区域，

6.2.3 实验结果分析

得到实车周围的全景图像如图 6.17 所示。

图 6.17 实车全景图像

Fig.6.17 Panoramic image of the actual vehicle

图 6.17 是拍摄的液晶屏上显示的汽车全景图像，中间的汽车图像是程序后期添加，主要用于显示汽车的位置。根据四个摄像头的安装位置，是光轴朝外，无法得到汽车自身图像，采用图像拼接将四幅图像拼接为全景图像后，将出现中间黑色无图像区域，给人的视觉效果很差，因此，添加上汽车图片，将很直观的展示出全景俯视图的视觉效果，从而能更好地看出汽车四周的路面情况。从图 6.17 可以看出，全景图像能够较为明显的显示出汽车周围的路面情况，但是由于不同摄像头采集到的图像光照不同，会出现图像亮度的不同，从而使得拼接后的图像出现亮度不同区域，这是系统的不足之处。

现有的全景泊车辅助系统，比如英菲尼迪的 AVM 系统，该系统可将四个广角摄像头所摄图像通过电脑编辑后形成 360 度全景，并显示在汽车的中控台显示屏上，如图 6.18 所示。

图 6.18 英菲尼迪 AVM 系统显示图

Fig.6.18 The figure of Infiniti AVM system

可以看出，论文设计的全景图像主动安全系统与英菲尼迪的 AVM 系统相比，优势主要有：论文设计的系统具有障碍物检测与报警功能，能够对小于安全距离的汽车障碍物进行报警；图像拼接平滑，无明显的拼接缝。

论文设计的全景图像主动安全系统与英菲尼迪的 AVM 系统相比，不足之处主要有：图像畸变比英菲尼迪的 AVM 系统大；其次，论文设计的系统没有在液晶屏上显示单个摄像头的图像，只显示了拼接完成的全景图像；英菲尼迪图像的亮度处理好于论文设计的系统。

6.3 本章小结

本章节主要完成了系统各部分单独实验测试以及系统整体实车实验测试。具体来说，在系统各部分单独实验测试中，通过系统平台运行实验检测验证了硬件

系统设计方案的可行性和硬件的可靠性，核心板能够实现同时处理四路摄像头采集到的图像并显示出来；通过摄像头失真校正实验解决了广角摄像头图像失真问题，从而验证了论文采用的失真校正方法的可行性；通过障碍物检测与报警实验验证了论文采用的基于特征的障碍物检测算法的可行性，能够检测出图像中小于安全距离的汽车障碍物并发出声音报警；通过全景图像拼接实验验证了论文采用的全景图像拼接算法的可行性，能够将四个广角摄像头采集到的图像拼接为全景图像，并清晰显示。通过系统整体实车实验，验证了整个系统连调的可靠性，实现了文件系统和程序的固化和烧写，达到了论文的研究目的，增强了汽车的主动安全性能；最后，通过与现有系统进行比较，得出论文所设计的主动安全系统的优势与不足。

7 结论与展望

7.1 结论

论文以汽车主动安全作为研究背景，主要设计了一种汽车全景图像主动安全系统，该系统能够使用设置在汽车前后左右 4 个方向的广角摄像头，采集汽车四周的画面，对图像进行汽车障碍物检测，当汽车障碍物距本车的距离小于安全距离时，系统会发出声音报警，并利用图像拼接技术将四幅图像拼接为全景图像，显示汽车四周环境。所获得的主要结论和成果如下：

①针对系统需求，确定了 4 路摄像头图像信号采集与处理的硬件设计方案。选用 TI 公司的基于达芬奇技术的TMS320DM6446作为核心芯片，TVP5158作为视频解码芯片，并以 4 个广角摄像头作为图像采集传感器。

②完成汽车全景主动安全系统的实验平台的设计与搭建，并实现了该系统在实车上的整体实验。

③采用基于特征的障碍物检测方法来检测汽车障碍物，通过提取车底阴影、左右对称性、车体垂直与水平边缘作为检测特征，实现对小于安全距离的汽车障碍物的声音报警。采用基于坐标系转换的算法来获取汽车四周的俯视图像并将其拼接为全景图像。

④通过实验平台与实车的测试，验证了论文采用的障碍物检测方法与全景图像拼接方法的可行性。

7.2 展望

论文取得了一定的研究成果，但是在研究过程中也发现了许多问题和不足，这些都需要在后面的研究过程中进行改正和完善，总的来讲存在以下这几点。

①论文的前期研究工作是在自己搭建的实验平台上完成，后期进行了实车整体实验，实车整体实验效果还不错，但是由于时间有限，并没有进行工艺化设计，没有做成产品。在之后的研究中可以增加工艺化设计，将系统产品化。

②系统目前是对障碍物进行检测，当障碍物小于安全距离后进行声音报警，这只是一种检测，缺乏控制。后续研究可以增加车速控制，当检测到障碍物小于安全距离后可以控制油门和刹车，使驾驶员无法踩下油门或者自动轻带刹车，降低车速。

致　　谢

嘉陵江水涨又落，歌乐山花开又谢。七年重大生，一生重大人，母校培育了我，母校教育了我，母校养育了我，感谢母校，感恩母校！

初入2004实验室是在我大四的时候，报名参加了飞思卡尔比赛。那一年，速度与激情、知识与实践相互碰撞；那一年，认识与熟悉了实验室的师兄与师姐；那一年，被老师的睿智与平和所吸引，决定研究生就在2004实验室读了！

三年研究生学习与生活，如弹指一瞬，送走了师兄与师姐，迎来了师弟与师妹。每个人都那么具有个性，每个人又都那么具有共性，个性是与生俱来的，共性是老师培养的。严谨的学风、端正的作风、严密的逻辑思维、和善的与人相处，这些都是老师给我们树立并教导我们的。老师不仅授人以鱼而且授人以渔，将学与用相结合，向我们展示了实践才是知识最好的应用之处。感谢谢昭莉老师！感谢盛朝强老师！

煤科院的实习生活是我人生中难得的一笔财富，教会了我许多在学校无法习得的东西，感谢汤朝明老师在我实习过程中以及后面学习过程中对我的指导！

实验室的个性犹如一个万花筒，让实验室充满了乐趣；也犹如一座书库，让我们能相互学习。周志忠师兄的大器、颜昱师兄的机敏、张德全师兄的沉稳、张宇师兄的萌、李城武师兄的孤傲、廖萍师姐的随和、杨茜师姐的热情一一在脑海中闪现。实验室时常进行激烈的辩论，让人不得不感慨何龙思维的敏捷，张宗楠知识涉猎的广博，陈超谈吐的犀利。

感谢我的父亲与母亲，是你们从小养育了我，把我培养成人，感谢你们！

感谢评阅论文和参加答辩的各位专家、教授！

杨钰超

二〇一五年四月 于重庆

参考文献

[1] 上海市城市综合交通规划研究所.上海综合交通 2011 年度报告(摘要上)[J].交通与运输.2011(05):11-13.

[2] 中华人民共和国工业和信息化部装备工业司.2011 年汽车工业经济运行情况[EB/OL]http://zbs.miit.gov.cn/n11293472/n11295142/n11299183/14442503.html.2012-01-19.

[3] 重庆市西南机动车驾驶员培训中心.汽车学习驾驶员培训教材[M].成都：电子科技大学出版社，2013：8-9.

[4] Parent M, Gallais G. Intelligent transportation in cities with CTS[A].IEEE 5th International Conference on Intelligent Transportation Systems[C]. Piscataway,IEEE Operation Center .2002:826 - 830.

[5] 杨明.无人自动驾驶车辆研究综述与展望[J].哈尔滨工业大学学报，2006，38：1259-1262.

[6] 重庆市西南机动车驾驶员培训中心.汽车学习驾驶员培训教材[M].成都：电子科技大学出版社，2013：8-9.

[7] 黄席樾，唐高友.基于机器视觉的道路识别与障碍物检测技术研究[J].重庆大学学报.2005(05).

[8] 李玲玲，余文勇，陈幼平，周祖德.基于数学形态学的灰度线形态识别研究与开发[J].计算机工程与应用. 2002(11).

[9] 谢虎.基于 CCD 图像处理的汽车主动安全系统的研究[D].辽宁：东北大学，2008.

[10] 公安部交通管理局.中华人民共和国道路交通事故统计年报(2011 年度)[Z].无锡市：公安部交通管理科学研究所，2011.

[11] Kato K, Suzuki M, Fujita Y, et al. Image synthesis display method and apparatus for vehicle camera[P]. US: US7139412 B2, 2002.

[12] 南方英菲迪斯. 体验全球首创的 AVM 系统 [EB/01]. http://www.bitauto.com/html/news/2009720/20097209544678137.shtml.

[13] 朱敏慧.博通以太网技术推动新一代车内网络发展[N].汽车与配件，2012(1):32-33.

[14]彭利平.德尔福力推汽车电控新技术[N].汽车电器，2013(5):66-68.

[15] 德州仪器 TVP5158 视频解码器与 DM6437 数字媒体处理器被汉华安道用于开发全景泊车辅助系统[N].电子设计工程，2011(23)：6.

[16] 方财义，邓军，檀银学.全景泊车系统及图像拼接技术的研究[J].研究与开发，2012(11)：103-108.

[17] 王旭东.基于环视的自动泊车方法研究与系统设计[D].上海：上海交通大学，2013.

[18] Mann S, Picard R W. Virtual bellows: constructing high quality stills from video[A].The First

IEEE International Conference on Image Processing[C].IEEE, 1994:363 - 367.

[19] Zitová B, Flusser J. Image registration methods: a survey[J]. Image and Vision Computing, 2003: 977-1000.

[20] B. Triggs et al. International workshop on vision algorithms in bundle adjustment- a modern synthesis[J].1999,9:298-372.

[21] Davis J. Mosaics of Scenes with Moving Objects[A].IEEE Computer Society Conference on Computer Vision and Pattern Recognition [C]. Santa Barbara, 1998:354--360.

[22] Milgram D L. Computer methods for creating photo mosaics[J]. Computers IEEE Transactions on, 1975, 24(11):1113 - 1119.

[23] Peleg S, S P, Peleg S, et al. Elimination of seams from photo mosaics[J]. Computer Graphics and Image Processing, 1981, (16):90–94.

[24] Hansen M, Anandan P, Dana K, et al. Real-time scene stabilization and mosaic construction[A]. The Second IEEE Workshop on Applications of Computer Vision[C].IEEE, 1994:54 - 62.

[25] Schmid C, Mohr R, Bauckhage C. Comparing and evaluating interest points[A].The Sixth International Conference on Computer Vision[C].IEEE, 1998:230 - 235.

[26] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics[A]. ACM Transactions on Graphics[C]. 1983,2(4):217–236.

[27] Bogdan,Fijalkowski. All-weather & All-terrain Intelligent Vehicle for Law Enforcement Applications[A].IEEE Intelligent Vehicles'94 Symposium[C].1994,(2):509-514.

[28] Haselhoff A,Kummert A,Davis. A vehicle detection system based on Haar and triangle feature[A].IEEE Intelligent Vehicle Symposium[C].Xi'an:IEEE.2009:261-266.

[29] Margrit B,Esin H,Lary S.Real-time multiple vehicle detection and tracking from a moving vehicle[J].Machine Vision and Applications,2000,12(2):69-83.

[30] Sun Z,Bebis G,Ronald M. Monocular pre-crash vehicle detection features[J].IEEE Transcations on Image Processing,2006,15(7):2019-2033.

[31] 许言午，曹先彬，乔红.行人检测系统研究新进展及关键技术展望[J].电子学报，2008,36(5)：962-968.

[32] Ludwig O,Delgado D,Goncalves V,Nunes U. Trainable classifier-fusion schemes:An application to pedestrian detection[A].IEEE Conference on Intelligent Transportion Systems[C]. St.Louis, MO:IEEE, 2009,1-6.

[33] Lin Z,Davis L. Shape-based human detection and segmentation via hierarchical part-template matching[J].Pattern Analysis and Machine Intelligence,2010,32(4):604-618.

[34] 田广，戚飞虎.移动摄像机环境下基于特征变换和 SVM 的分级行人检测算法[J].电子学报，2008,36(5)：1024-1028.

[35] Qifa Ke,Takeo Kanade. Transforming camera geometry to a virtual downward-looking camera robust ego-motion estimation and ground-layer detection[J].IEEE Computer Society on Computer Vision and Pattern Recognition,2003,1(1):390-397.

[36] Boyoon Jung,Gaurav S. Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment[A].The 8th Conference on Intelligent Autonomous Systems[C]. Amsterdam, Netherlands:IEEE,2004,980-987.

[37] Wybo S,Bendahan R,Bougnoux S,Vestri C,Abad F,Kakinami T. Movement detection for safer backward maneuver[A].The 13th World Congress on Intelligent Transport Systems[C]. Tokey:IEEE,2006,453-459.

[38] Xuebing Wang,Kenji Ban,Kazuo Ishii. Estimation of mobile robot ego-motion and obstacle depth detection by using optical flow[A].IEEE International Conference on International Conference on Systems, Man and Cybernetics[C].San Antonio,TX:IEEE,2009.1770-1775.

[39] 张泽旭，李金宗，李宁宁.基于光流场分割和 Canny 边缘提取融合算法的运动目标检测[J]. 电子学报.2003，(09)：1299-1302.

[40] Klappstein J,Stein F,Franke U. Monocular motion detection using spatial constraints in a unified manner[A].IEEE Intelligent Vehicles Symposium[C].Tokyo:IEEE,2006.261-267.

[41] 刘威，于红绯，杨恒，段勃勃.一种新的基于单目视觉的广义障碍物检测方法[J].电子学报.2011.39(8)：1793-1800.

[42] Gary Bradski, Adrian Kaehler. Learning open CV[M].US:Stanford University,2009.

[43] 于仕琪，刘瑞祯.学习 Open CV[M].北京：清华大学出版社，2009.

[44] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations[J].IEEE，1999.

[45] B. Triggs. Auto calibration from planar scenes[A].5th European Conference on Computer Vision[C].1998:89-105.

[46] 张正友.计算机视觉-计算机理论与算法基础[M].北京：科学出版社，1998.

[47] 吴福朝，阮宗才，胡占义.非线性模型下的摄像机自标定[J].计算机学报，2002，(3)：276-283.

[48] 徐静.基于非线性模型的摄像机标定技术研究[J].现代电子技术，2008，(12)：276-283.

[49] 徐友春.基于机器视觉的汽车主动安全技术的研究[R].北京：清华大学汽车工程系，2003.

[50] 王永仲.鱼眼镜头光学[M].北京：科学出版社，2006：35-47.

[51] 董晨.基于 DaVinci 平台的数字视频技术研究[D].北京：北京交通大学，2007.

[52] TMS320DM6446 Digital Media System-on-Chip Datasheet.2007,3.

[53] TVP5158,TVP5157,TVP5156 Four-Channel NTSC/PAL Video Decoders Datasheet.2010,10.

[54] AOZ1016 EZBuck 2A Simple Buck Regulator Datasheet.2007,9.

[55] Brown D C. Close-range camera calibration[J]. Photogrammetric Engineering,1971,

37(8):855-866.

[56] Charkari N M, Mori H. A new approach for real time moving vehicle detection[J]. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 1993, 1:273 - 278.

[57] 柴毅.智能化汽车主动安全系统研究[D].重庆：重庆大学，2001：72-75.

[58] 马程.车辆辅助驾驶全视角成像关键技术的研究[D].安徽：合肥工业大学，2009：1-4.

[59] 丁鑫.全景视觉泊车辅助系统研究[D].浙江：浙江大学，2010：20-25.

[60] 赵凯.全景可视化辅助泊车系统研究[D].安徽：合肥工业大学，2011：32-37.

附　录

A. 作者在攻读硕士学位期间获得的奖励

[1] 2012 年获重庆大学“研究生优秀共青团干部”荣誉称号

[2] 2013 年获重庆大学“优秀共青团干部”荣誉称号

[3] 2014 年获重庆大学“优秀研究生干部”荣誉称号

B. 搭建 ARM 编译环境

具体安装步骤如下：

①安装 MontaVista Linux 环境

1)从网上下载 MontaVista 环境的目标文件系统和开发工具包，名称为：mvl_5_0_0801921_demo_sys_setuplinux.bin

mvl_5_0_0_demo_lsp_setuplinux_02_00_00_140.bin 。

2)接下来可以选择安装 VMware Tools 软件，该软件安装好后，能够实现将文件从主机 Windows 中直接复制拖进虚拟机 Linux 下的功能，当然，也可以采用共享文件夹方式，即在主机 Windows 下创建共享文件夹，名为 biyesheji_share，将文件放进共享文件夹 biyesheji_share 中并设置，使得虚拟机 Linux 能使用。

3)MontaVista 软件的安装需要在 Red Hat Linux 虚拟机上进行，因此使用管理员账号登陆虚拟机 Linux。

4)虚拟机的共享文件夹存放在/mnt 目录下，因此需要进入该文件夹，找到需要的文件，将其复制到目录 /usr/tmp 下面，以便后面的减压缩等操作，当然也可以直接在共享文件夹里面操作，只是后面不好整理。

5)使用 CTRL+ALT+F1 切换到 Linux 系统的命令行界面，创建文件夹 mvl_pro_5.0.0 来管理相关文件，指令为：

$ mkdir /opt/mvl_pro_5.0.0

6)进入存放下载下来文件的目录，地址为 /usr/tmp，并执行这两个 bin 文件。命令如下：

$ cd /usr/tmp

$ ./mvl_5_0_0801921_demo_sys_setuplinux.bin

$ ./mvl_5_0_0_demo_lsp_setuplinux_02_00_00_140.bin

文件将会被安装在默认的安装目录之下，其具体的地址为 /opt/mvl_pro_5.0.0 。

7)安装完后，会在地址 /opt/mvl_pro_5.0.0 下产生名称如下的两个压缩包文件：

mvltools5_0_0801921_update.tar.gz

DaVinciLSP_02_00_00_140.tar.gz

在压缩文件包 mvltools5_0_0801921_update.tar.gz 中主要存放有开发 DM6446 芯片所需的根文件系统以及进行 ARM 编译所使用的 GCC 的交叉编译环境，在压缩文件包 DaVinciLSP_02_00_00_140.tar.gz 中则存放着 MontaVista Linux-2.6.18 的内核源代码。

8)进入地址 /opt/mvl_pro_5.0.0，并在该目录下解压这两个压缩包文件，具体输入以下命令：

$ cd /opt.mvl_pro_5.0.0

$ tar zxf mvltools5_0_0801921_update.tar.gz

$ tar zxf DaVinciLSP_02_00_00_140.tar.gz

解压完以上安装包后，将会在上面那个地址下产生 MontaVista 文件夹，然后把冗余的文件删除掉后。

②安装 DVSDK 工具链

在完成上面的安装步骤之后，就完成了 MontaVista Linux 环境的安装，接下来需要安装 TI 公司的 DVSDK 工具链。

具体步骤如下：

1)为了便于安装，首先需要创建一个普通用户账号，可以使用下面的命令。

$ useradd -d /usr/yyc -m yyc

$ passwd yyc 012345

2)由于 TI 公司建议客户使用普通用户账号安装 DVSDK，因此，需要建立普通用户文件夹来存放 DVSDK。命令如下：

$ mkdir /usr/yyc

3)将从网上下载下来的 dvsdk_setuplinux_2_00_00_22.bin 文件复制进 Linux 系统中，并利用普通用户身份进入 Linux 系统，然后进入该文件夹，安装 DVSDK。

./dvsdk_setuplinux_2_00_00_22.bin

4)在 dvsdk 安装完成之后，需要将 xdctools 与 bios 工具包一起安装到目录为 /usr/yyc/dvsdk_2_00_00_22/ 下。

./xdctools_setuplinux_3_10_05_61.bin

./bios_setuplinux_5_33_03.bin

5)进入文件夹 /usr/yyc/dvsdk_2_00_00_22/ ，并在其下建立文件。

mkdir cg6x_6_0_23

6)在完成文件夹 cg6x_6_0_23 的创建后，可以通过运行下面的命令来安装 ti_cgt_c6000_6.0.23 到 /usr/yyc/dvsdk_2_00_00_22/cg6x_6_0_23/ 目录下：

./TI-C6X-CGT-V6.0.21.1.bin

此时，就完成了 DVSDK 的安装了。

7)设置环境变量。通过按下 CRTL+ALT+F7 键切换到 Linux 系统图形界面。

找到文件 dvsdk_2_00_00_22/Rules.make，点击进入编辑，修改里面的宏设置，主要是修改 XDC_INSTALL_DIR 宏和 BIOS_INSTALL_DIR 宏。具体的对应修改如下：

XDC_INSTALL_DIR=$ (DVSDK_INSTALL_DIR)/xdctools_3_10_05_61

BIOS_INSTALL_DIR=$ (DVSDK_INSTALL_DIR)/bios_5_33_03

同时，针对需要使用到的 DSP 程序编译，应该修改 CODEGEN_INSTALL_DIR 宏，具体的对应修改如下：

CODEGEN_INSTALL_GIR=$ (DVSDK_INSTALL_DIR)/cg6x_6_0_23

最后将 DVSDK_INSTALL_DIR 宏进行如下修改：

DVSDK_INSTALL_DIR=$ /usr/yyc/dvsdk_2_00_00_22

这就完成了环境变量的设置。

③安装 Codec Servers

Codec Servers 需要安装到 /usr/yyc/dvsdk_2_00_00_22 文件夹之下，首先切换到管理员账号登陆 Linux 系统，然后使用下列命令来进行安装：

$ ./dm6446_codecs_setuplinux_2_00_00_22.bin

等待安装完毕，这样就完成了 Codec Servers 的安装了。

C. NFS 模式下 U-Boot 具体设置

设置的启动参数如下：

DaVinci EVM # setenv baudrate 115200

DaVinci EVM # setenv ethaddr 00:0E:99:EF:EF:22

DaVinci EVM # setenv bootcmd 'tftp 0x80700000 uImage;bootm 0x80700000'

DaVinci EVM # setenv serverip 192.168.136.151

DaVinci EVM # setenv bootfile uImage

DaVinci EVM # setenv ipaddr 192.168.136.229

DaVinci EVM # setenv bootargs console=ttyS0,115200n8
video=davincifb:vid0=0,2500K:vid1=0,2500K:osd0=720x576x16,2025K
davinci_enc_mngr.ch0_output=COMPOSITE　davinci_enc_mngr.ch0_mode=pal
davinci_enc_mngr.ch1_output=COMPOSITE　davinci_enc_mngr.ch1_mode=pal
noinitrd rw root=/dev/nfs nfsroot=192.168.136.151:/root/armfs_dvsdk2.0,nolock
mem=112M　eth=00:0E:FF:FF:FF:80　ip=192.168.136.229::192.168.136.151
DaVinci EVM # saveenv

D. 文件系统固化与程序烧写

具体步骤如下：

①制作 ramdisk

在 Linux 虚拟机系统的 /tftpboot 的目录下建立一个新的大小为 20M 的 ramdisk 镜像文件，并使用 /dev/zero 设备来初始化，其具体的指令如下：

$ dd if=/dev/zero of=/tftpboot/ramdisk bs=1k count=20480

为建立的镜像文件建立专门的文件系统，命令如下：

$ mke2fs -F -v -m0 /tftpboot/ramdisk

接着需要完成 ramdisk 的挂载，挂载地址为 /tftp/ram ，具体命令如下：

$ mkdir -p /tftpboot/ram

$ mount -o loop /tftpboot/ramdisk /tftpboot/ram

将 ramdisk_5.0 挂载到地址为 /tffp/ram0 下，具体命令为

$ mkdir /tftpboot/ram0

$ gzip -d /tftpboot/ramdisk_5.0.gz

$ mount -o loop /tftpboot/ramdisk_5.0 /tftp/ram0

接下来需要复制系统文件，命令如下：

$ cp -av /tftpboot/ram0/* /tftpboot/ram

然后将之前编写好的全景图像拼接与障碍物检测等程序复制到根文件系统之下，并修改设置文件为开机自动启动。卸载 ramdisk，具体命令如下：

$ umount /tftpboot/ram

$ umount /tftpboot/ram0

然后将 ramdisk 进行压缩，具体命令如下：

$ gzip /tftpboot/ramdisk

到这一步，就完成了 ramdisk 的制作了。

②将 ramdisk 烧写进 nand flash

首先烧写 Linux 内核，如下所示：

DaVinci EVM # tftp 0x80700000 uImage

DaVinci EVM # nand erase 0x140000 0x240000

DaVinci EVM # nand write.jffs2 0x80700000 0x140000 0x240000

然后烧写根文件系统，如下所示：

DaVinci EVM # tftp 0x85000000 ramdisk.gz

DaVinci EVM # nand erase 0x540000 0x1000000

DaVinci EVM # nand write.jffs2 0x85000000 0x540000 0x1000000

③设置 U-Boot 启动参数

setenv baudrate 115200

setenv ethaddr 00:0E:99:EF:EF:22

setenv a1 i2c mw 0x20 0x3 0x0

setenv a2 i2c mw 0x20 0x1 0x90

setenv bootcmd'run al;run a2;nand read.jffs2 0x80700000 0x140000 0x240000;

nand read.jffs2 0x85000000 0x540000 0x1000000;bootm 0x80700000

setenv serverip 192.168.136.151

setenv bootfile uImage

setenv ipaddr 192.168.136.229

setenv bootarge console =ttys0,115200n8 ip=dhcp root=/dev/ram0

rw initrd=0x85000000,14M video=dm64xxfb:interface=prgb:mode=240p

eth=00:0E:FF:FF:FF:80 ip=192.168.136.229:192.168.136.1:255.255.255.0