

1800101



学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

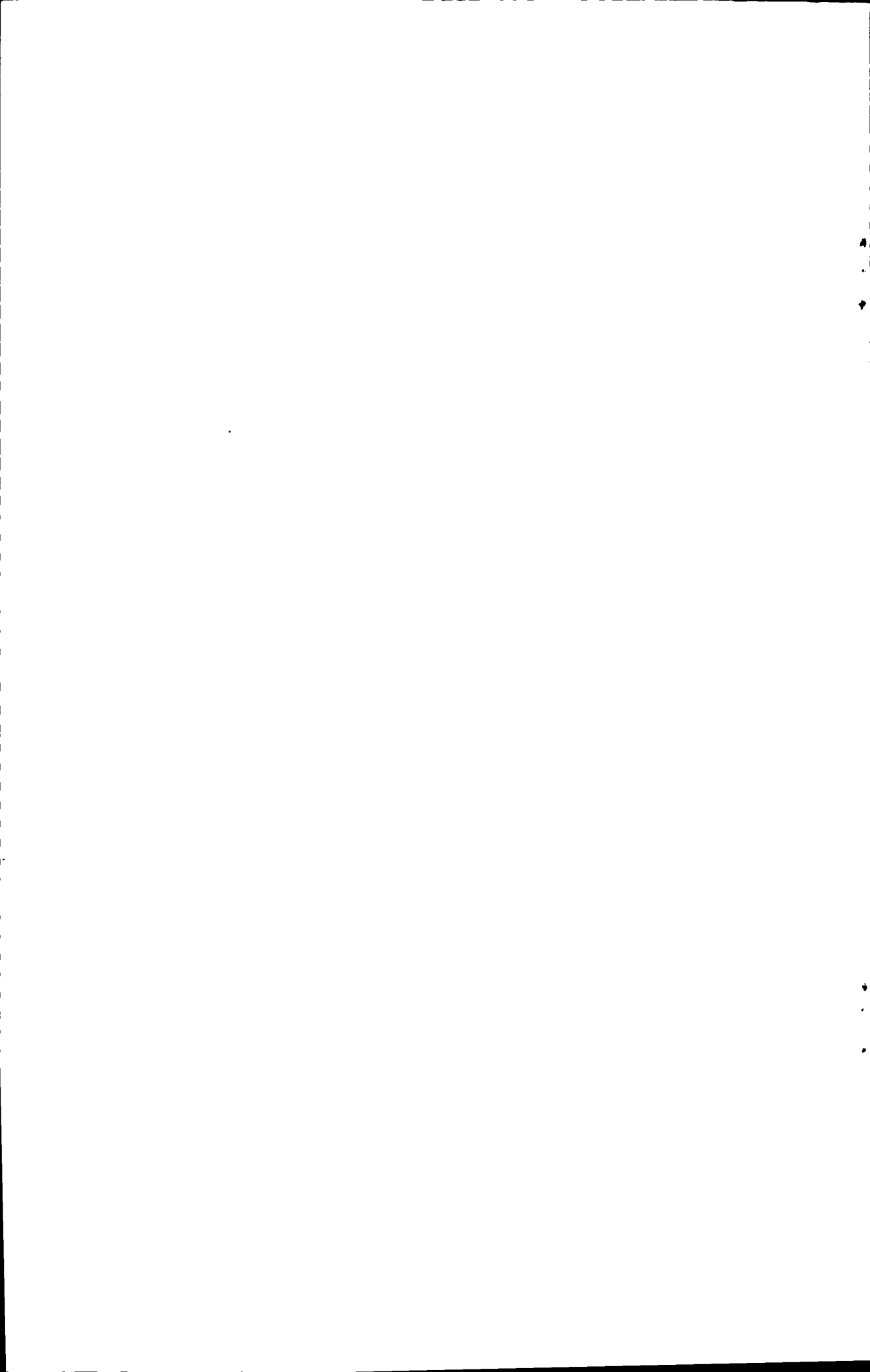
(保密的学位论文在解密后适用本授权说明)

学位论文作者签名: 王亮

导师签名: 余祖俊

签字日期: 2010年6月25日

签字日期: 2010年6月25日



中图分类号: TP391.4, U216.3
UDC: 004.93, 656.2

学校代码: 10004
密级: 公开

北京交通大学

硕士学位论文

基于图像的高速铁路线路环境状态智能识别技术研究

Image Based Intelligent Recognition for Environment
Status of High Speed Railway Lines

作者姓名: 王尧

学 号: 08121938

导师姓名: 余祖俊

职 称: 教授

学位类别: 工学

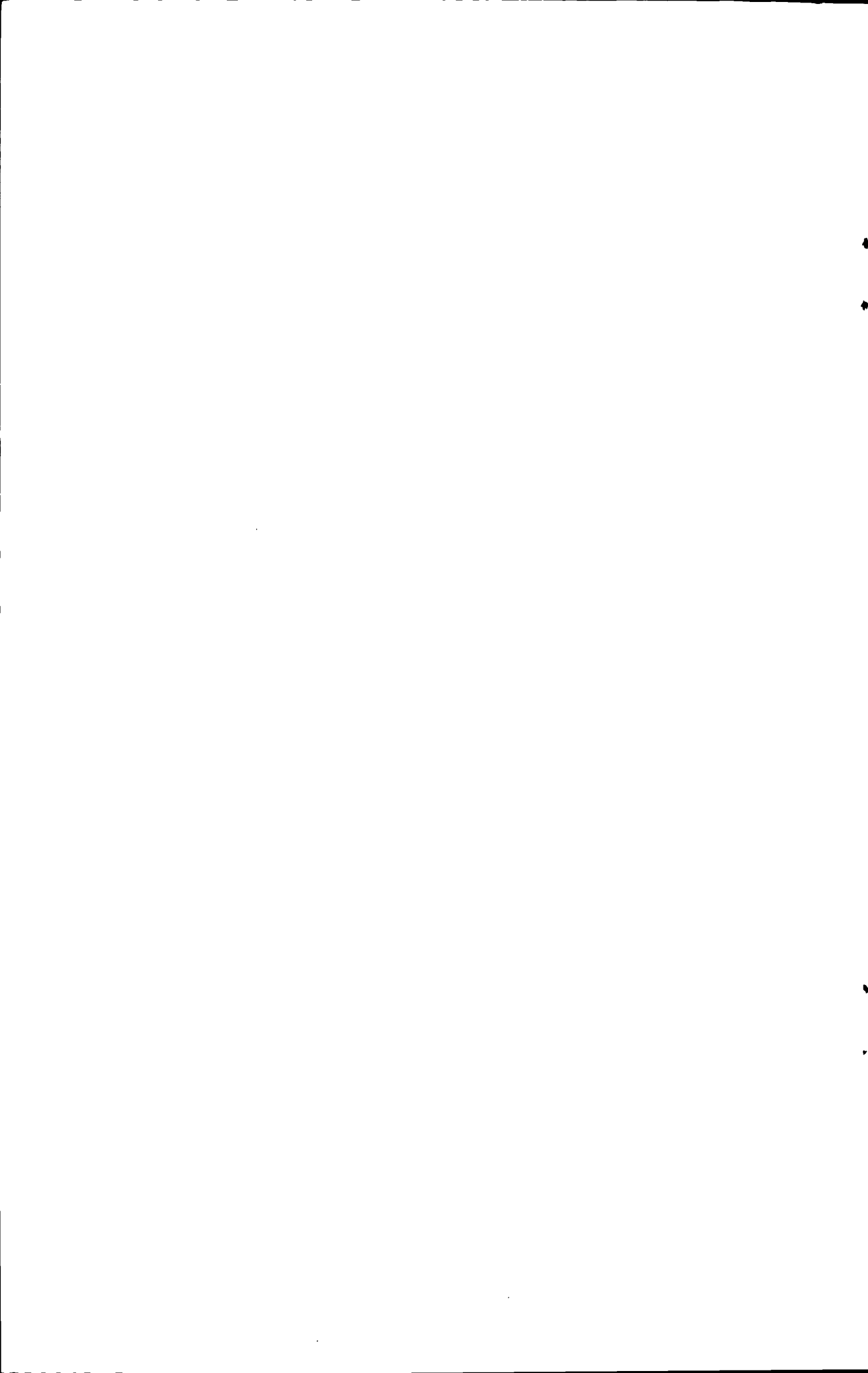
学位级别: 硕士

学科专业: 检测技术与自动化装置

研究方向: 智能化测控技术

北京交通大学

2010年6月



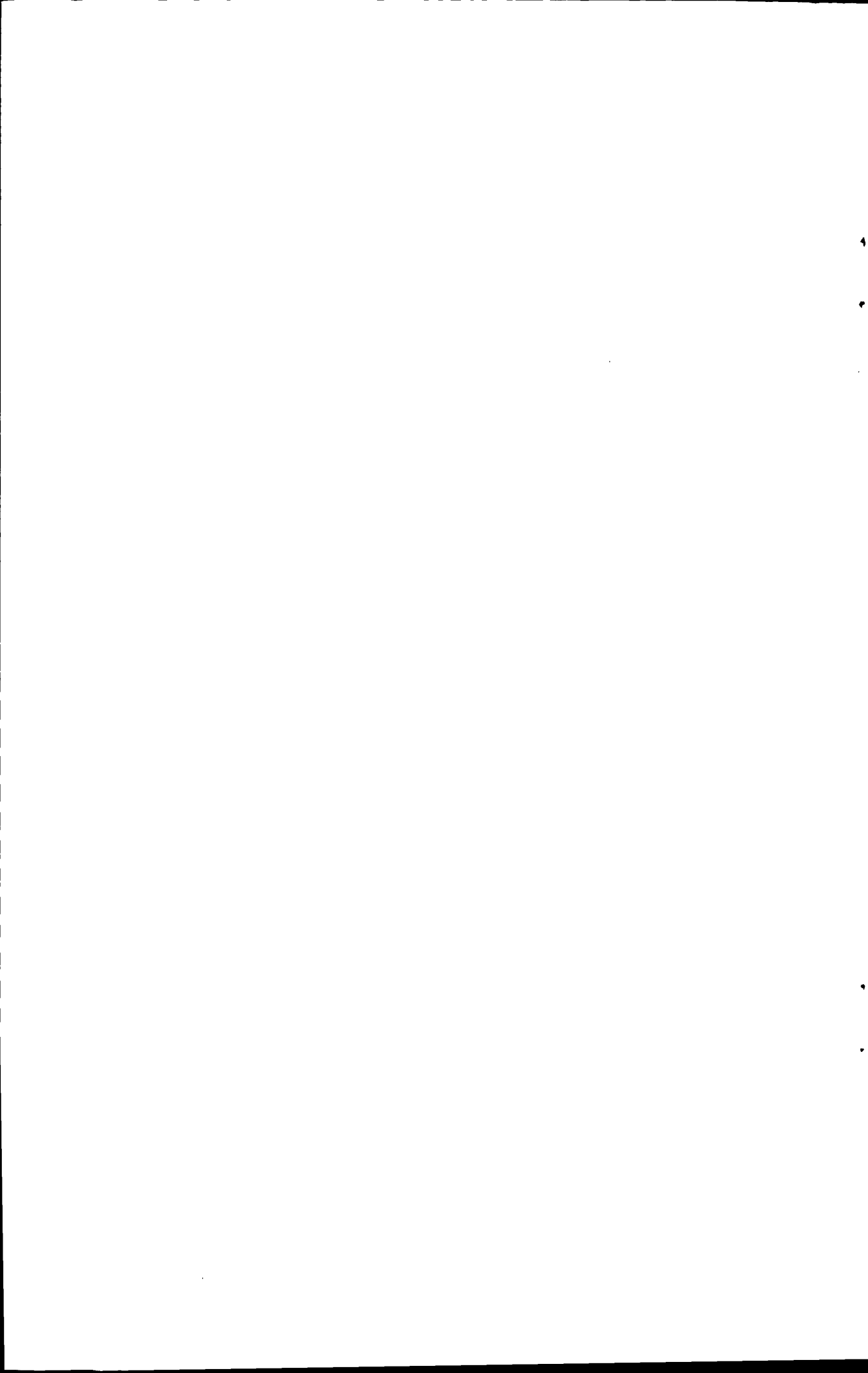
致谢

本文的工作得到了许多师长、同学的指导和帮助，在此我要向他们表示衷心的感谢。首先我要感谢我的导师余祖俊教授，本论文是在余老师的悉心指导下完成的，余老师严谨的治学态度和科学的工作方法给了我极大的帮助。

在实验室工作和论文撰写期间，史红梅老师、朱力强老师、郭保青老师无论是在工作上还生活上都给予了热情的帮助和指导。他们学识渊博，治学严谨，是我学习的榜样。论文进行中，他们帮助解决了许多具体问题，在很多问题上提供了大量富有建设性的意见。此外，计算机学院的李清勇老师在空间异物识别方面提供了很大的帮助和指导。在此向他们表示衷心地感谢。

我还要感谢中国铁道科学院的谭松、李海浪以及中国北车唐山轨道客车有限公司的董焕彬。他们在现场调研、系统方案确定等方面给予了很大的帮助，并提供了大量线路视频资料。

最后我要感谢我的家人和朋友，论文能够顺利完成和他们一如既往的支持和帮助是分不开的。



中文摘要

本文研究了高速铁路线路环境状态智能识别的相关技术, 主要包括实时图像处理 and 基于图像的智能识别两个方面, 并以此为基础构建了基于图像的高速铁路线路环境状态智能识别系统。该系统通过安装在综合检测列车上的摄像机采集线路图像, 利用图像处理、机器视觉等技术对图像进行实时处理和分析, 从而实现对线路环境状态的智能识别。

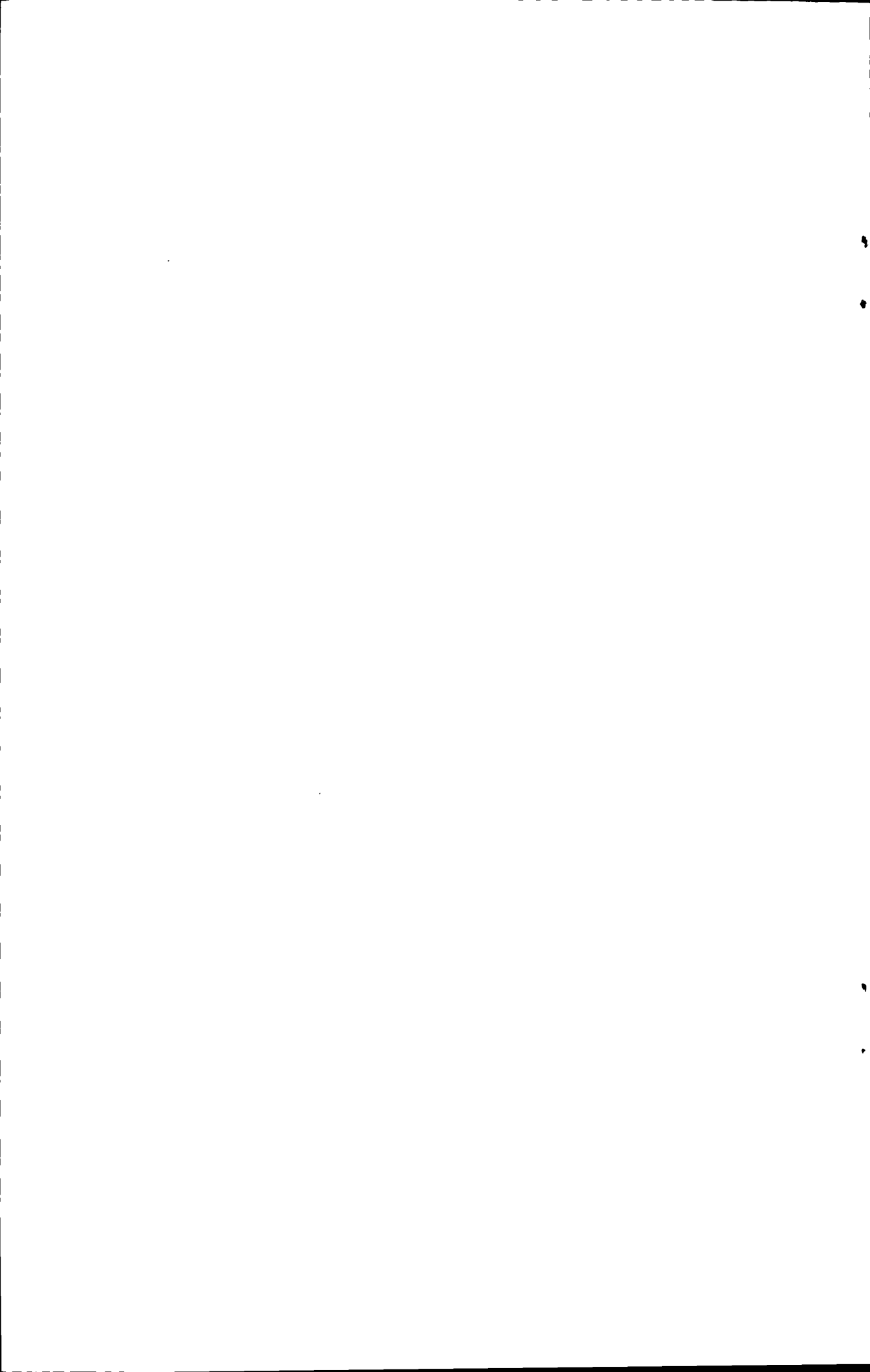
本文首先介绍了高速铁路线路环境状态智能识别系统的总体设计, 包括系统总体方案、系统组成、各部分功能、采用的关键技术等。并重点分析和讨论了成像系统选型和安装的相关问题。随后论文就实时图像处理和基于图像的智能识别两个方面进行了深入地研究。

在实时图像处理方面, 本文采用了基于 FPGA 的硬件处理器来实现图像预处理算法, 以提高系统的处理速度。为了解决当前图像处理硬件设计困难、设计周期长等问题, 本文创新性地提出了基于图像代数的图像处理算法硬件实现技术, 并采用该方法实现了本系统相关的图像处理算法, 如图像滤波、形态学处理、边缘提取等, 的硬件设计, 给出了硬件的仿真结果和图像处理的结果。

在基于图像的智能识别方面, 本文采用了二维图像识别和三维立体视觉识别相结合的方法, 利用二维图像识别技术快速提取线路异常特征, 再利用立体视觉得到异常特征的三维信息, 来确认其是否为线路缺陷。在基于二维图像的线路环境状态的提取和识别方法上, 本文提出了基于灰度统计特征和边缘统计特征的护栏完整性识别算法, 以及基于混合高斯模型的空间异物识别算法。在三维立体视觉识别上, 本文完成了三维立体视觉系统的构建、标定以及三维信息重建等。

关键词: 基础设施检测、机器视觉、图像代数、FPGA。

分类号: TP391.4, U216.3



ABSTRACT

The technologies related to the intelligent recognition of the environment status in high speed railway, mainly including the real-time image processing and the image based intelligent recognition, were reaserched in this thesis. The intelligent recognizing system for high speed railway environment status based on these technologies was also built in this thesis. This system acquires images of railway line by using the camera installed on the comprehensive inspection train, then processes and analyses these images to realize the intelligent recognition of the environment status.

The general design of the intelligent recognition system for the environment status of the high speed railway lines, including the component of the system, the functions of each part, and some key technologies, was firstly introduced in this thesis. The issues about camera selection and installation were also discussed. Then the aspects of real-time image processing and image based intelligent recognition were reasherched in deatail.

In the aspect of real-time image processing, a FPGA based hardware processors, which highly accelerates the processing speed, was used in this system. To simplify the design of image processing hardwire and makes the hardware designed more flexible, a new hardwire implement technique of image processing algorithms based on image algebra was proposed in this thesis. This method was used to implement some algorithms used in system, including 2-D image filter, gray-scale morphology and edge detection, the simulation results and testing results were given in this thesis.

In the aspect of the recognition, a method combines 2-D image recognition and 3-D stereo recognition was proposed in this thesis. Firstly, 2-D image recognition was used to abstract the abnormal features of the railway line rapidly, and then the stereo vision was used to obtain the 3-D information of them, which confirm if these features are defects of the railway line or not. Therefore, this method has both high recognizing speed and high reliability. Fence completeness recognition algorithm and foreign object recognition algorithm were discussed in this thesis, the principles, features and results are given. The architecture and principle of stereo vision, calibration experiment and 3-D information obtain are introduced, and the recognition based on 3-D information is also discussed in this thesis.

KEYWORDS: Inspection of railway infrastructure, Machine vision, Image algebra, FPGA.

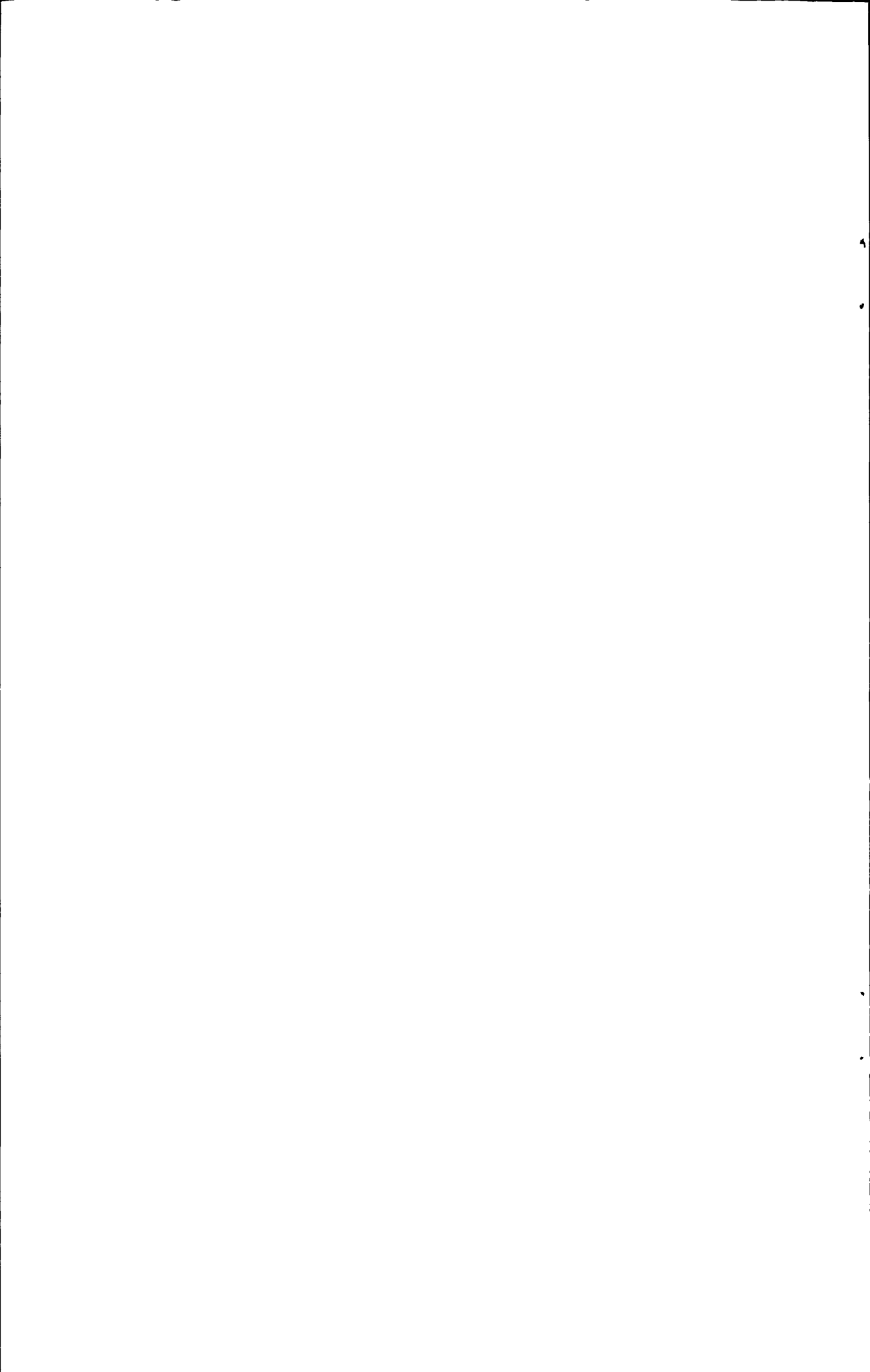
CLASSNO: TP391.4, U216

目录

中文摘要	v
ABSTRACT.....	vii
1 绪论	1
1.1 课题背景	1
1.2 国内外研究现状	2
1.2.1 国内研究现状	2
1.2.2 日本新干线 East-i 高速综合检测列车	2
1.2.3 意大利“阿基米德”号高速综合检测列车	4
1.2.4 美国 Ensco 公司图像检测系统.....	6
1.2.5 其他国外高速综合检测列车	7
1.2.6 小结	7
1.3 本论文的主要内容	8
2 高速铁路线路环境状态智能监测系统总体设计	9
2.1 系统总体方案	9
2.2 成像系统选型与安装	10
2.2.1 相机模型	10
2.2.2 镜头光学特性	14
2.2.3 相机安装位置	14
2.2.4 相机主要参数确定	16
2.3 图像传输与采集	18
2.3.1 图像传输方式选择	18
2.3.2 图像采集	20
2.4 基于 FPGA 的实时图像预处理	21
2.4.1 配置总线接口	22
2.4.2 图像数据接口	23
2.4.3 DDR SDRAM 接口模块.....	26
2.5 系统软件结构	29
3 基于图像代数的图像处理算法硬件实现技术	31
3.1 图像代数	31
3.1.1 点集及其运算	32
3.1.2 图像及其运算	33
3.1.3 模板	34
3.1.4 图像与模板间的操作	35

3.2	图像处理算法的图像代数表示.....	36
3.2.1	空间滤波及其图像代数表示.....	36
3.2.2	数学形态学及其图像代数表示.....	37
3.2.3	边缘提取及其图像代数表示.....	38
3.2.4	图像几何变换及其图像代数表示.....	39
3.3	图像代数基本运算的硬件实现.....	39
3.3.1	模板及模板运算的硬件实现.....	40
3.3.2	图像及其运算的硬件实现.....	43
3.3.3	图像和模板间操作的硬件实现.....	46
3.4	典型图像处理算法的硬件实现.....	48
3.4.1	空间滤波的硬件实现.....	49
3.4.2	灰度形态学的硬件实现.....	50
3.4.3	边缘提取的硬件实现.....	51
4	基于图像的线路环境状态提取与识别.....	55
4.1	护栏完整性识别.....	55
4.1.1	护栏图像几何变换.....	56
4.1.2	基于灰度统计特征的快速识别.....	60
4.1.3	基于边缘统计特征的识别.....	62
4.2	空间异物识别.....	67
4.2.1	识别过程.....	67
4.2.2	基于混合高斯模型的背景估计.....	68
4.2.3	自适应背景更新.....	70
5	基于立体视觉的线路环境状态识别.....	71
5.1	立体视觉几何学.....	71
5.2	摄像机标定.....	72
5.2.1	摄像机标定原理.....	72
5.2.2	双目标定实验.....	73
5.3	线路环境状态异常点的三维信息重建.....	77
5.3.1	立体图像校正.....	77
5.3.2	立体匹配.....	79
5.3.3	三维信息重建.....	81
5.4	基于三维信息的识别方法.....	83
6	总结与展望.....	84
6.1	工作总结.....	84

6.2 展望	84
参考文献	86
作者简介	89
独创性声明	90
学位论文数据集	91



1 绪论

1.1 课题背景

随着高速铁路技术的不断发展,列车的运行速度也不断提高。由于列车不间断地运行,往往使高速铁路基础设施发生各种变形或损坏。为了确保列车能按规定的最高速度安全、平稳、不间断地运行,以及延长线路各部分的使用寿命,必须加强对基础设施的养护维修和监测,保证线路设备经常处于完好状态^[1]。尤其是对高速铁路,基础设施的维护更显得至关重要。

因此,各国对高速铁路基础设施的检测和维护工作都非常重视,研制了大量的基础设施检测和维修设备。其中高速铁路综合检测列车,能够在高速运行条件下对线路各项参数进行综合检测。包括:轨道参数、接触网参数、通信信号状态等,检测结果可作为线路检修的依据。综合检测列车具有自动化程度高、检测效率高、结果准确等特点,大大缩短了检测时间,提高了线路维护的效率和质量。当前,我国正大力发展客运专线,届时基础设施的检修任务将非常繁重。研制高速铁路综合检测列车十分必要。

线路环境状态是基础设施检测的重要组成部分之一,各种人为因素或自然因素都有可能破坏线路环境状态。线路环境的异常,如护栏破损、接触网挂有异物、异物侵入线路区间等,会给安全行车带来隐患。因此需要对线路环境进行监测及早发现隐患、排除隐患。线路环境监测的传统方法通常是人工巡视线路,这样虽然准确率较高,但是工作量大、效率低下、维护成本较高。高速铁路采用的综合检测列车安装有视频监测系统,采用对线路全景进行摄像的方法,可以由人工根据录像实时判断线路环境或存储后离线分析。虽然在一定程度上提高了检测效率且误检率较低,但是工作量仍然较大,并且人工监测受主观因素影响,容易遗漏部分异常点。为了提高检测效率、降低成本、提高精度,有必要研制一套智能识别系统,使其能够对线路环境状态进行自动识别,对线路环境异常进行记录、报警等。

本课题以国家“863”计划重点项目“最高试验速度400km/h高速检测列车关键技术研究及装备研制”为依托。研究高速检测条件下,实现线路环境状态在线自动识别的相关技术,包括基于图像的护栏完整性、空间异物侵入线路区间等的智能识别算法,以及实时图像处理技术等。并以此为基础研制高速铁路线路环境状态的智能识别系统。

1.2 国内外研究现状

目前,在高速综合检测列车和高速铁路基础设施检测领域,日本、意大利、美国、法国等国家有较高的研究水平。研发了大量的成熟产品和先进的检测技术。以往研究热点主要集中在轨道参数检测、接触网参数检测等。其中应用图像技术和机器视觉技术进行检测是近年来出现的较新的研究热点,总体呈现出人工检测自动化的发展方向。国外现有的高速综合检测列车主要有日本的 East-i、意大利的“阿基米德”号、法国的 IRIS320、英国的 NMT 等^[2]。

1.2.1 国内研究现状

我国 1953 年成功研制了第一代轨道检查车,其检测装置与瑞士 Matis 公司生产的 PV-6 轨检车基本相同。随后第二代轨检车于 1969 年试制成功,采用接触式方式,可检测三角坑、左右轨接头高低、水平高低、车体摇晃和列车速度等项目。1986 年,第三代轨检车面世,采用非接触式测量,提高了检测速度。1993 年我们从国外引进了新型轨检车,安装有美国 ENSCO 公司最先进的 T10 轨道检测设备,该系统采用惯性基准测量原理,测量速度可达到 160km/h^[3]。

2007 年我国开始研制高速综合检测列车,由中国铁道科学研究院基础设施检测研究所负责系统集成,动车组由长春轨道客车股份有限公司设计制造。2008 年 6 月“0 号高速综合检测列车”下线交付使用。全车为 8 辆编组,由通信信号检测车、会议车、接触网检测车、数据综合处理车、轨道检测车、餐车、卧铺车和信号检测车等组成。“0 号高速综合检测列车”可在 300 公里时速的速度等级内,实时检测线路轨道、轮轨作用力、牵引供电、通信信号、周边环境是影响列车运行安全的技术指标和相关信息。

1.2.2 日本新干线 East-i 高速综合检测列车

日本先后生产了 6 列新干线电气轨道综合检测列车,以最新的 E926 型 East-i 最具代表性。East-i 是在早期 Dr.Yellow 基础上进一步改进而成的,为 6 辆编组动车组,可以检测轨道几何参数、接触网、通信信号、轮轨作用力环境噪声等内容,最高检测速度 275km/h。各检测系统可独立完成检测工作,整个检测列车在速度、时间和单程位置上保持同步。但该系统数据库和处理软件是由日本铁路研制的专用软件,通用性较差^[4]。

E926 型高速综合检测列车的 1 号车和 6 号车安装有“前方影像录像”系统^[5]。

如图 1-1 所示，整个系统由光源、激光测距仪、全景摄像机以及轨道摄像机组成。如图 1-2 所示，光源和激光测距仪安装在车头前方中部。光源用于给摄像机提供照明。激光测距仪安装在离轨面高 2m 的位置，用于测量检测列车与护栏的距离，当护栏有异物或护栏出现破损时，激光测距仪测得的结果将出现较大的变化。如图 1-3 所示，全景摄像机和轨道摄像机安装在车窗后右侧。全景相机的视野为 100m，用于拍摄前路前方线路环境信息，轨道摄像机视野为 20m，专门拍摄轨道画面。摄像机信号可以传输到整车其他检测部分，也可以进行压缩存储。



图 1-1 East-i 前方影像录像组成示意图

Fig. 1-1 The schematic diagram of front sight video record system on East-i

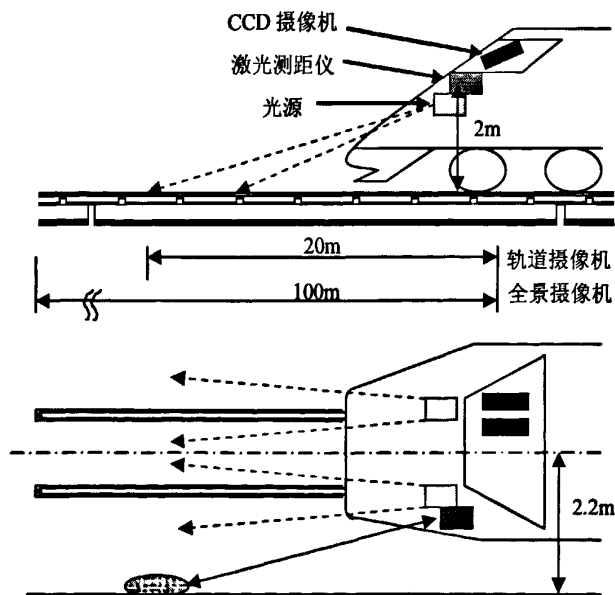


图 1-2 East-i 型综合检测列车光源、激光测距仪安装位置图

Fig. 1-2 Installation positions of light source and laser distance measuring instrument on East-i

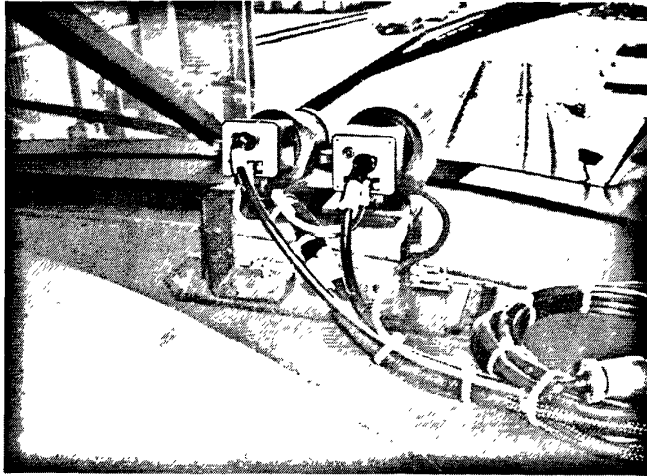


图 1-3 East-i 型综合检测列车全景摄像机与轨道摄像机

Fig.1-3 Cameras for whole scene and track scene on East-i

1.2.3 意大利“阿基米德”号高速综合检测列车

意大利 Tecnogamma 公司和 Mer Mec 公司于 2004 年 4 月 15 日来华进行技术交流,重点介绍其研发并投入运用的“阿基米德”号综合检测列车,如图 1-4 所示。

“阿基米德”号综合检测列车由 6 节车组成,检测速度 220km/h。针对中国高速综合检测列车,意大利公司在“阿基米德”号的基础上提出“马可波罗”号综合检测列车方案,该方案由 6 节辆组成,检测速度可达 300km/h。

“阿基米德”号由意大利 MER MEC 公司开发,2001 年 9 月交付意大利铁路基础设施检测中心(RFI)使用。阿基米德号综合检测列车为 6 辆编组,最高检测速度 220km/h,具有检测轨道几何参数、钢轨断面、钢轨波浪磨耗、接触网及受流状态、通信信号、车体和轴箱加速度轮轨作用力等功能^[4,5]。

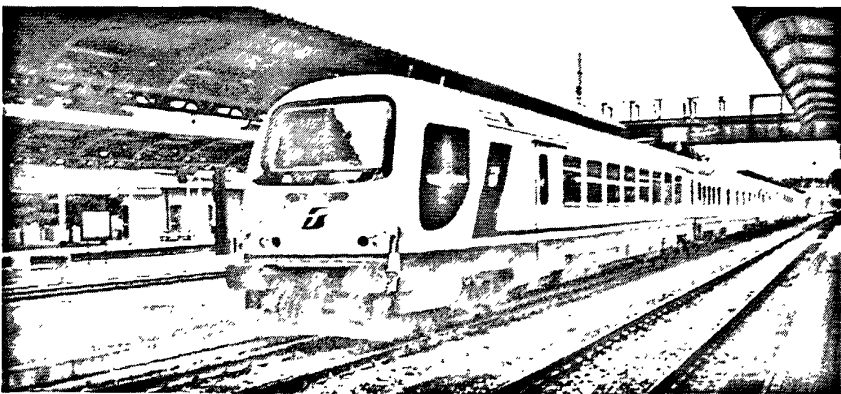


图 1-4 意大利“阿基米德”号高速综合检测列车

Fig.1-4 “Archimedes” high speed comprehensive measuring train

“阿基米德”号综合检测列车中，在多处安装了摄像检查传感器，可进行线路环境视频监测、接触网视频监测分析、轨道及转向架视频监测、司机视线视频监控等。其视频监测系统可实现摄像/图片记录，关键图片可通过 GPRS 网络发送至地面，如图 1-5 所示。



图 1-5 “阿基米德”号综合检测列车线路图像

Fig. 1-5 A railway line image recorded by “Archimedes”

“阿基米德”号综合检测列车中，可利用图像信号进行轨道缺陷检测。轨道缺陷检测系统是基于无接触高速可视技术，由两台线性数字摄像机获得图像信号，进行相应的图像处理从而提取钢轨缺陷特征。其主要功能包括：在高速线路上获得两条钢轨的图像数据；以数字格式记录这些数据；使用专用软件自动分析确定轨道缺陷；该系统设计用来安装在高速车辆上；其配套的软件可以实现：车上系统可以获取、处理、显示和存储两条钢轨的图像；办公室系统可以重放运行中存储的数据，进行分析、处理并得出轨道缺陷检测结果；缺陷按照范围可分为 4 级，如图 1-6 所示。可检测的主要轨道缺陷有：轨枕紧固件的完整情况、钢轨波浪磨耗、钢轨上的石碴坑、紧固件丢失情况、钢轨裂纹、打滑造成的磨耗、脱皮、裂纹、开焊、碎石印、轨枕裂纹、烧熔、断口等。

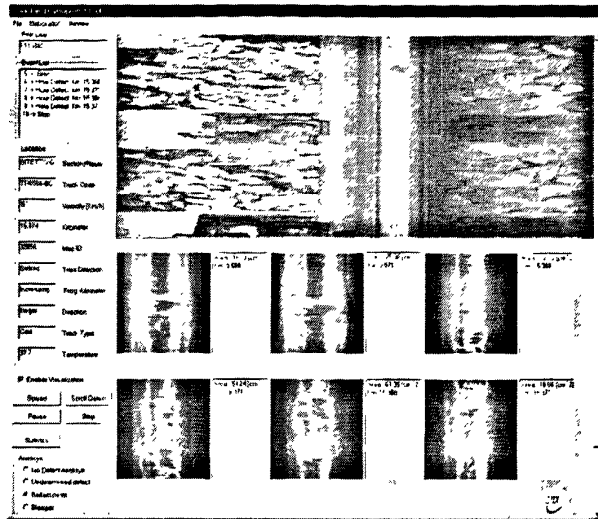


图 1-6 钢轨缺陷检测程序界面

Fig. 1-6 The interface of track lesion detection software

1.2.4 美国 Ensco 公司图像检测系统

美国的 Ensco 公司开发出图像检测系统 (Video Inspection System 简称 VIS)。该系统组成如图 1-7 所示, 系统使用四个高速线扫描摄像机, 采用图像处理技术, 对轨道结构和道床进行监视。可以在全速运行的情况下收集线路存在危险的轨道几何信息, 检测由计算机自动完成, 消除人工轨道监视的主观性, 有助于减少线路维护的成本, 提高安全性和效率^[6]。

系统还能提供车辆运行前方的环境、铁轨以及道床的高清晰度图像, 并可实现连续录像。轨道几何测量、乘车舒适度测量及全球定位系统可以与该系统集成。车载计算机软件可以实时显示图像和分析结果, 如图 1-8 所示。

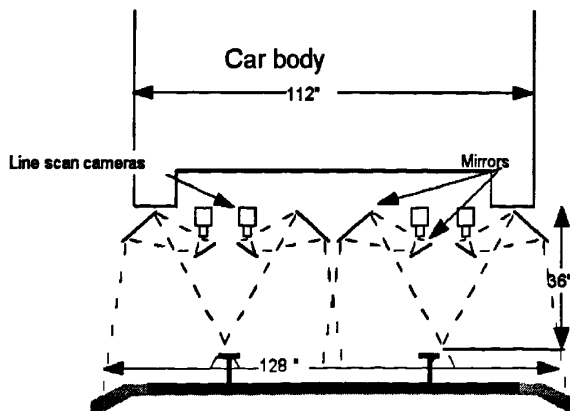


图 1-7 Ensco 图像检测系统示意图

Fig.1-7 The schematic diagram of image measuring system of Ensco

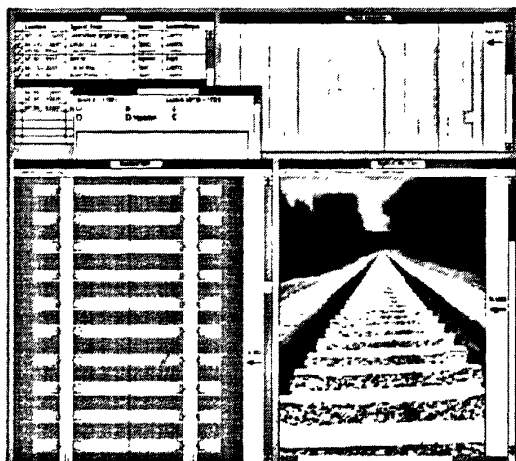


图 1-8 Ensco 图像检测系统软件示意

Fig.1-8 The schematic diagram of image measuring software of Ensco

1.2.5 其他国外高速综合检测列车

IRIS320 是法国国铁(SNCF)研制的高速综合检测列车,最高检测速度 320km/h,目前大部分检测项目还在调试验收中。IRIS320 由一列已使用 7 年的 10 辆编组的 TGV 动车组改造而成,检测系统包括轨道几何参数、钢轨断面、钢轨波浪磨耗、接触网及受流状态、通信信号、车体和轴箱加速度、轮轨作用力、列车交会压力波、定位和环境监视等^[4]。

英国的 NMT 综合检测列车车体采用 1977 年制造的 7 辆编组的一列内燃动力车组,于 2002 年开始,历经 4 年多改造。将轨道几何、接触网、车辆动态响应、视频监控、钢轨表面伤损、轨枕和扣件状态等检测系统安装到一列车上,通过建立定位系统、同步网络、数据网络和综合分析系统将所有参数进行同步检测和综合数据处理。目前,检测系统已集成完毕,投入运营检测,该车最高运行速度为 200 km/h^[4]。

1.2.6 小结

从综合检测列车线路环境检测技术现状的介绍中可以看到,目前的综合检测列车中,均安装有视频检测系统,主要用于线路环境监测和轨道状态监测。其中轨道状态监测的研究水平较高,已经可以实现轨道缺陷的自动监测。目前的线路环境监测利用车头摄像机拍摄列车运行前方视景,可实时提供给各节车厢,同时

进行压缩存储可进行离线分析。但是对线路环境信息的分析仍然是人工完成的，无法对线路环境进行自动监测，监测效率较低。

1.3 本论文的主要内容

本论文完成了高速铁路线路环境状态智能识别相关核心技术的研究，包括图像处理算法的硬件实现技术，基于二维和三维视觉技术的线路环境异常特征提取和识别算法等，并以此为基础完成检测系统原型机的设计研制。系统通过二维快速判断、三维精确确认的方法，使得系统在具有较高检速度的同时，提高了系统的可靠性。论文内容的具体安排如下：

第 1 章介绍了课题的背景与来源，重点分析了高速铁路环境状态检测的国内外研究现状。

第 2 章主要介绍了本文研制的高速铁路环境状态智能检测系统的总体设计。在本章中，首先介绍了系统的组成、各部分的功能等；然后根据系统要求从理论上分析了成像系统，即相机、镜头等设备的选型以及安装等的相关问题；接着讨论了基于 FPGA 的图像采集与预处理技术，分析了本系统采用 FPGA 进行图像预处理的必要性，并重点介绍了 FPGA 图像预处理涉及的一些基础性硬件的设计；最后介绍了系统的软件结构。

第 3 章提出并初步实现了基于图像代数的图像处理算法硬件实现技术，该技术是一种以图像代数理论为基础的进行图像处理硬件设计的通用方法，是为解决图像处理算法硬件设计难度大、通用性差等问题进行的有益的探索和尝试。本章首先简单介绍了图像代数的基础知识，以及一些典型图像处理算法的图像代数表示。随后介绍了图像代数基本运算的硬件实现，给出了仿真结果，并在此基础上介绍了和本系统有关的图像处理算法的硬件实现，给出了仿真结果以及在 FPGA 中实现后的处理结果。

第 4 章分析了基于二维图像识别技术的线路环境状态异常特征提取和识别算法。针对没中算法详细讨论和分析了算法的原理，识别效果以及优缺点等。

第 5 章讨论了基于三维立体视觉的识别方法，介绍了双目视觉的基本原理，双目立体视觉系统的标定方法等。并就三维信息的重建方法，包括立体图像校正、图像匹配、距离信息恢复等问题进行了讨论。

第 6 章对本文的内容进行了系统地总结，并就论文的后续工作进行了展望。

2 高速铁路线路环境状态智能监测系统总体设计

本系统的主要功能是利用车载相机采集的线路图像信息，对线路环境状态进行分析。系统涉及图像采集、传输、处理以及分析等技术。本章将介绍系统的总体设计以及和系统设计相关的基础性问题，包括系统的组成，各组成部分的功能，各器件、设备选型的相关问题等。特别是针对本系统所特有的问题进行了详细地讨论。

2.1 系统总体方案

系统总体方案的示意图如图 2-1 所示。系统由两个车头摄像机和 5 台工业 PC 组成，另由综合检测列车上的其它系统提供接触网摄像机信号和距离同步信息。5 台 PC 中，一台为主控 PC 负责整个系统的控制、报警信息的存储以及与综合检测列车其它系统的通信等，其余 4 台分别完成护栏完整性判断；高清图像压缩存储、线路三维信息离线生成、线路空间异物判断、接触网异物判断等四项功能。距离同步信号提供给每台计算机用于图像信息与定位信息的同步。所有计算机都通过以太网接入综合检测列车的车载高速通信网络。

两台高分辨率摄像机安装在综合检测列车车头，从左右两个不同的角度拍摄线路图像，以实现双目立体视觉。拍摄的图像通过 CameraLink 接口输出。若处理计算机安装在综合检测列车的其它车厢，可使用 CameraLink 与光纤的相互转换装置，利用整车的光纤网络来传输图像数据。

在护栏完整性识别计算机和空间异物识别计算机中，根据需要安装了一块 FPGA 同步图像采集、图像预处理板卡。板卡采用 PCI 接口与 PC 进行接口。摄像机输出的图像由 FPGA 进行采集，并可在 FPGA 内部通过硬件实现图像预处理，可以大大加快图像采集与处理的速度。使得 PC 图像处理的负担得到减轻，减少处理时间，从而保证整个识别系统处理的实时性。PC 读取预处理后的图像，再对图像进行高层次的处理。PC 首先采用二维图像处理技术对线路图像进行初步判断，再利用三维视觉技术对可疑特征进行确认。可在保证处理速度情况下，提供系统判别的可靠性。接触网异物识别计算机由于处理功能较简单，而图像存储计算机不要求实时处理，因此都只安装了图像采集卡。

距离同步信息是由综合检测列车上的其它系统提供的，每台 PC 在发现线路环境异常时，根据该信息记录异常发生的地点，以便维修人员可以快速地寻找故障。

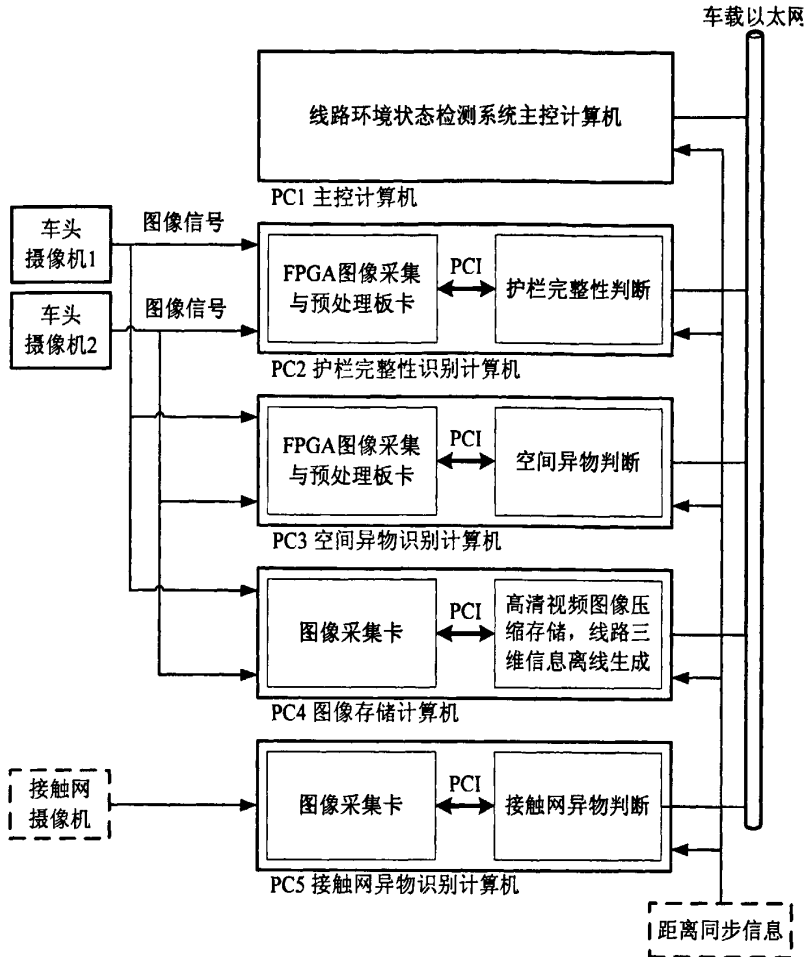


图 2-1 系统组成示意图

Fig.2-1 The composition of the system

2.2 成像系统选型与安装

成像系统是图像的获取设备，主要包括相机和镜头。图像的获取是机器视觉系统中最基本的，而又是常常被忽视的问题^[7]。图像获取的质量直接影响到整个机器视觉系统通的性能，而保证图像获取质量的关键在于合理地选择成像系统。在本系统中，成像系统需要在高速移动的条件下对线路场景进行实时采集。因此，本系统对成像系统的动态性能要求较高。另一方面，相机的安装也将影响图像的获取和相机参数的选择。本节将就本系统中成像系统选型和安装方案设计中的关键问题进行讨论。

2.2.1 相机模型

相机模型是本章分析成像系统选型和安装的基础，因此有必要首先对其进行介绍。相机模型是描述相机和镜头成像的数学模型，它建立了空间中的点与其在图像上所成像点的对应关系。相机模型有很多种，各种模型具有不同的精度和复杂程度。由于本章只利用相机模型进行选型时的参数估算对精度要求不高，因此本文采用了最常用的小孔成像相机模型。

相机的小孔成像模型如图 2-2 所示。在小孔成像模型中，相机的镜头被近似为一个小孔，外界的光线通过这个小孔投射到 CCD 芯片上形成一个外界场景的倒像。相机的处理电路将这个倒像变换为正图像输出。为了方便分析通常将 CCD 平面以投影中心作中心对称得到一个虚拟的平面，这个平面被称为虚拟图像平面。这样相机输出的图像就可以看作是虚拟图像平面上的所成的像。小孔成像中，物点和像点的变换过程可以看作是以小孔为投影中心的中心投影，在数学上可以用仿射变换来描述。为了描述这个变换过程首先需要建立四个坐标系^[8]：

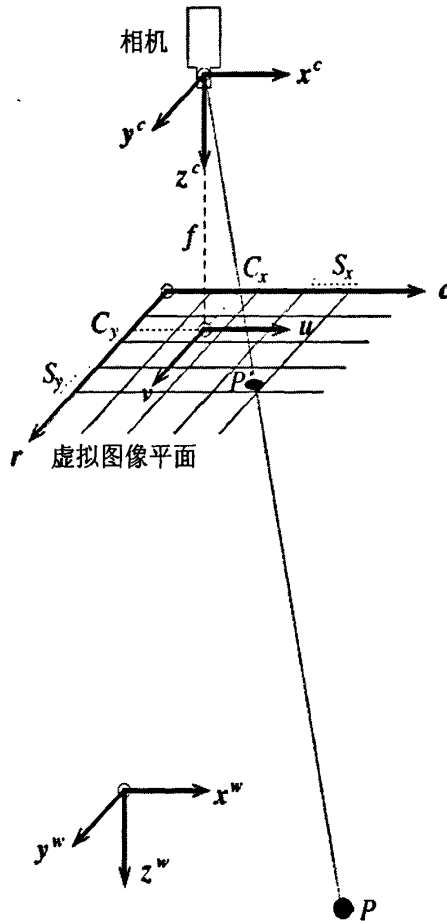


图 2-2 相机小孔成像模型

Fig.2-2 The pinhole model of cameras

(1) 相机坐标系 $X^c Y^c Z^c$ ：为固定在相机上的坐标系，其原点为投影中心， Z

轴垂直于图像平面。

(2) 图像平面坐标系 UV ：为图像平面上的二维坐标系，其原点为相机坐标系 Z 轴与图像平面的交点。其 U 轴与相机坐标系的 X 轴平行。

(3) 图像坐标系 RC ：与图像平面坐标系一样，该坐标系为图像平面的二维坐标系。但是该坐标系的原点在图像的左上角，且单位为像素。该坐标系的坐标即位像素点在图像中的行列值。

(4) 世界坐标系 $X^*Y^*Z^*$ ：根据实际应用选定的场景中的坐标系。

设场景中一点 P ，其在相机坐标系中的坐标为 $P^c = (x^c, y^c, z^c)$ ，并设其在图像平面上的像点为 $P' = (u, v)$ ，则根据中心投影关系可得物点与像点的投影关系可由如下方程表示^[9,10]：

$$u = \frac{f \cdot x^c}{z^c} \quad (2-1)$$

$$v = \frac{f \cdot y^c}{z^c} \quad (2-2)$$

其中， f 为相机的焦距，即投影中心和图像平面的垂直距离。

若使用齐次坐标系，设点 P 的齐次坐标为 $(x^c, y^c, z^c, 1)$ ， P' 点的齐次坐标为 (U, V, W) 。由仿射几何可知齐次坐标与空间坐标满足如下关系：

$$u = \frac{U}{W}, \quad v = \frac{V}{W} \quad (2-3)$$

则方程 (2-1) (2-2) 在齐次坐标系下，可写成如下矩阵形式：

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix} \quad (2-4)$$

这样我们就建立了场景空间点和其像点的对应关系。此时的像点是在图像平面坐标系 UV ，以齐次坐标的形式表示的。为了建立其与图像像素的关系，我们还需要建立图像坐标系 RC 和 UV 的关系。如图 2-2 所示，这两个坐标系是共面的，其主要区别在于其原点位置和度量单位不同。若设 UV 坐标系的原点在坐标系 RC 中的坐标是 $[C_x, C_y]^T$ ，相机 CCD 单个像素的尺寸为 $S_x \times S_y$ ，则像点在坐标系 RC 中的坐标 $[r, c]^T$ 可表示为：

$$c = \frac{u}{S_x} + C_x \quad (2-5)$$

$$r = \frac{v}{S_y} + C_y \quad (2-6)$$

若使用齐次坐标系，则式 (2-5) 和 (2-6) 可以表示成如下矩阵形式：

$$\begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = \begin{bmatrix} 1/S_x & 0 & C_x \\ 0 & 1/S_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2-7)$$

这样, 由 (2-3), (2-4) 以及 (2-7) 即可得到相机坐标系中空间点和图像坐标系中像点的变换关系:

$$z^c \begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = \begin{bmatrix} f/S_x & 0 & C_x & 0 \\ 0 & f/S_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix} \quad (2-8)$$

至此我们已经完整地建立了空间上一点与其在图像上对应的像点的变换关系, 从 (2-8) 中我们可以看出这个对应关系主要和相机的本身的参数有关, 我们通常将这些参数称为相机的内参数。通常我们已经可以利用 (2-8) 来解决许多实际问题, 但是在 (2-8) 中空间点是在相机坐标系下表示的, 而实际应用中被拍摄的场景通常有自己的坐标系, 称为世界坐标系。这样我们还需要建立世界坐标系和相机坐标系的变换关系, 通常称为外参数。

设空间点在世界坐标系 $X^w Y^w Z^w$ 中的坐标为 $\mathbf{P}^w = (x^w, y^w, z^w)$, 则根据坐标变换理论该点在相机坐标系中的坐标可表示为:

$$\mathbf{P}^c = \mathbf{R}(\mathbf{P}^w + \mathbf{t}) \quad (2-9)$$

其中 \mathbf{R} 和 \mathbf{t} 分别为世界坐标系到相机坐标系的旋转矩阵和平移向量。

用齐次坐标表示:

$$\begin{bmatrix} \mathbf{P}^c \\ 1 \end{bmatrix} = {}^c\mathbf{T}_w \begin{bmatrix} \mathbf{P}^w \\ 1 \end{bmatrix} \quad (2-10)$$

其中 ${}^c\mathbf{T}_w$ 为变换矩阵。这样根据 (2-8) 和 (2-10) 可以得到最终的相机模型表达式:

$$z^c \begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = \begin{bmatrix} f/S_x & 0 & C_x & 0 \\ 0 & f/S_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} {}^c\mathbf{T}_w \begin{bmatrix} \mathbf{P}^w \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{P}^w \\ 1 \end{bmatrix} \quad (2-11)$$

其中 \mathbf{M} 称为仿射投影矩阵或相机矩阵。

通常相机精确的内外参数需要标定得到, 但是本章只是利用相机模型进行相机选型的分析和估算, 并不需要非常精确的参数。因此本章只需要根据相机厂商提供的大致参数进行分析即可, 关于相机标定的内容将在本文第 5 章进行介绍。

2.2.2 镜头光学特性

镜头的作用是把外界景物成像在感光器件上,使感光器件上产生景物的像。在本系统中,镜头的特性直接决定了相机的安装和对相机参数的要求。因此有必要对镜头的光学特性进行研究。镜头的光学特性一般可用焦距 f 、相对孔径 D/f 、视场角 2ω 表示。

一般来说镜头的焦距决定了镜头的放大率 β ,当物距 l 较大时,放大率可表示为 $\beta \approx f/l$ 。由此可见,焦距的大小,决定了感光器件上像的大小和实际被拍摄物体大小的比例关系。

镜头的相对孔径即为镜头入瞳直径和焦距之比 D/f ,其倒数成为光圈。镜头像平面的光照度和相对孔径的平方成正比,因此相对孔径主要决定了成像的亮度。通常镜头的相对孔径是可以调节的,在本系统中由于检测列车行驶过程中外界光照变化较大,因此需要自动调节光圈的镜头;另一方面由于本系统需要在高速运动的条件下拍摄线路图像,为了保证成像清晰需要相机的曝光时间较短,这样就间接要求镜头具有较大的相对孔径。所以本系统需要的镜头是能够进行自动光圈调节并且最大相对孔径较大的镜头。

另一个镜头的主要特性是视场角 2ω ,它决定了可被拍摄物体的范围。相机的视场角处理和镜头有关还和感光器件的幅面大小有关。根据图 2-2 和无限远物体理想像高公式:

$$y = f \tan \omega \quad (2-12)$$

这样在感光器件幅面大小一定,即 y 一定时,镜头的视场角 2ω 就由镜头的焦距决定。从式 (2-12) 可以看出,当感光器件幅面大小一定时,焦距越小,则视场角越大。因此短焦距镜头,也就是大视角镜头^[11]。在本系统中,镜头的视场角和相机的安装位置以及角度有着密切的关系。

2.2.3 相机安装位置

在本系统中,相机的安装是一个比较重要的问题。一方面相机的安装必须要满足后续处理的成像要求,比如,为了保证所有线路都能被检测就要,拍摄的视场范围就需要足够大,为了提高立体视觉测量精度,两相机的距离就需要足够大;另一方面相机的安装位置又受到检测列车司机室结构和空间的限制。同时安装位置又直接关系到相机和镜头参数的确定。下面我们就车头相机安装位置的确定进行讨论。由于接触网图像是由其他系统提供的,本文将不讨论接触网相机的安装问题。

为了避免破坏检测列车的空气动力学性能，同时为了保障检测列车的行车安全。车头相机只能安装在检测列车司机室内。本系统将安装在 CRH3 型动车组上，其司机室如图 2-3 所示。由于受空间限制，为保证两相机有足够的距离以提高立体视觉测量的精度，相机只能吊装在司机室上部的横梁上。此处相机距轨面的高度为 3000mm，两相机的安装距离为 800mm。

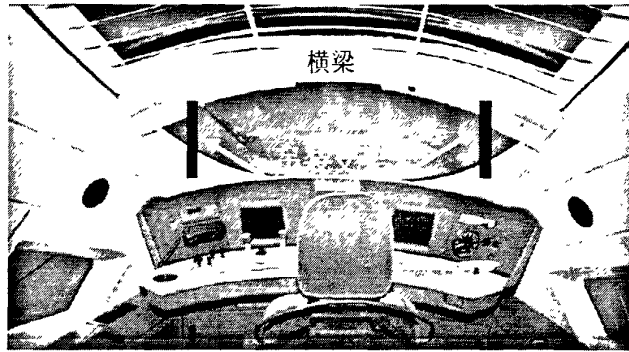


图 2-3 CRH3 型动车组司机室

Fig.2-3 The driving room of CRH3

相机的安装角度必须能够保证相机能够对所需处理的物体清晰的成像。车头相机拍摄的主要是护栏和线路。为了保证处理精度并兼顾处理速度的限制，图像处理系统一帧处理 10m 左右的线路。这样就要求相机的视场至少要覆盖 10m 以上的线路，为了保证可靠性我们将相机的视场确定为覆盖 20m 护栏和线路。覆盖护栏和线路的大小，除了和相机的视场有关还和护栏及线路和列车的相对位置有关。我国新建时速 300~350 公里客运专线标准线路断面布置图如图 2-4 所示，其中规定了线路和护栏（声屏障）的位置。同时标准还规定声屏障不高于轨面 2050mm^[12]。

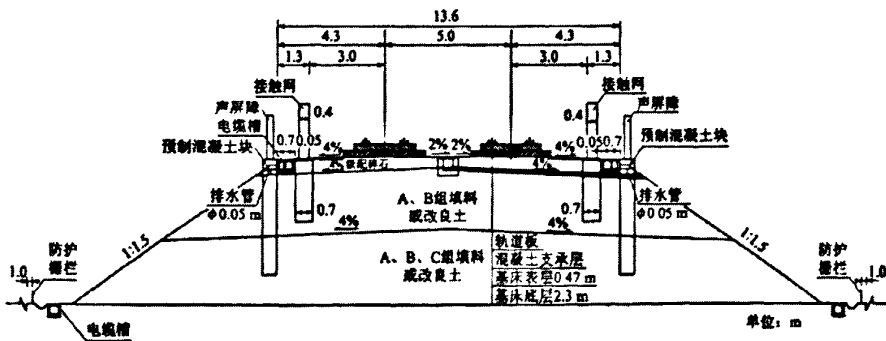


图 2-4 双线无砟轨道标准断面图

Fig.2-4 The standard profile diagram of ballastless double track embankment

相机视场覆盖的最远位置主要受其垂直安装角度的影响，如图 2-5 所示。为了保证能够完整拍摄 20m 长的完整护栏，则图中 L 必须大于 20m。根据图 2-5 所示的几何关系，我们可以得到相机俯角 α 、相机视场角 2ω 于 L 的关系由式 (2-13)

表示。

$$L = \begin{cases} \cot(\alpha - \omega) - \cot(\alpha + \omega), & \alpha > \omega \\ \infty, & \alpha \leq \omega \end{cases} \quad (2-13)$$

可以看出当视场角确定时 α 越小 L 就越大,但是相机俯角还需要考虑视场不被司机室内设备遮挡等问题,同时增大相机的俯角也将减小相机的水平视场范围。另一方面增大相机的俯角可以使得拍摄的线路图像更清晰有利于图像处理。因此相机俯角需要考虑各方面因素综合确定。

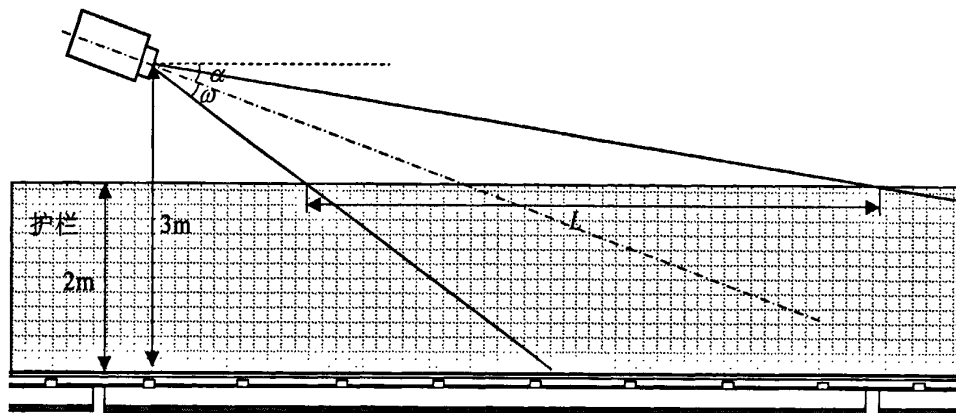


图 2-5 相机垂直安装和视场示意图

Fig.2-5 The schematic of the installation and scene range of the camera in vertical

在水平方向,相机中心线和沿线路中心没有偏转角度。需要考虑的问题是当相机俯角增大时水平视场的缩小后还能否拍摄到护栏。另外,两个相机在水平方向上的视场范围是不同的,而为了能够进行立体视觉测量被测点必须同时位于两个相机的视场内。这样也限制了一次可处理的线路图像的长度。

2.2.4 相机主要参数确定

相机参数和安装位置存在相互影响和相互制约的关系,一方面相机的参数,如焦距、视场角等将影响相机的安装位置,这从式(2-13)就可以看出;而另一方面相机的安装位置又限制相机参数的选择,特别是曝光时间,灵敏度等。所以必须综合考虑这两个问题,同时还要考虑市场上相机的供应情况。下面就相机几个主要参数的计算和选择问题进行讨论。

(1) 曝光方式与时间

由于本系统的成像系统在高速移动的情况下进行图像获取,相机的曝光将在很大程度上影响获图像质量。首先,相机应采用全局曝光技术。由于传统的卷帘式曝光每一行像素实际上是依次曝光的,因此在拍摄高速移动的物体时,将出现

物体的变形。而全局曝光使得整个感光器件的所有像素都同时曝光，这样就避免了图像出现变形。其次，相机要采用较短的曝光时间，以避免图像出现模糊。显然，相机最大允许的曝光时间和物体像点在图像平面中的移动速度有关。为了简化计算，我们假设相机安装的俯角为 0 度即如图 2-6 所示。这样计算会有一些的误差，我们可以在最终确定参数时加入适当的余量。根据相机的安装位置，利用相机模型我们可以计算像点的移动速度。

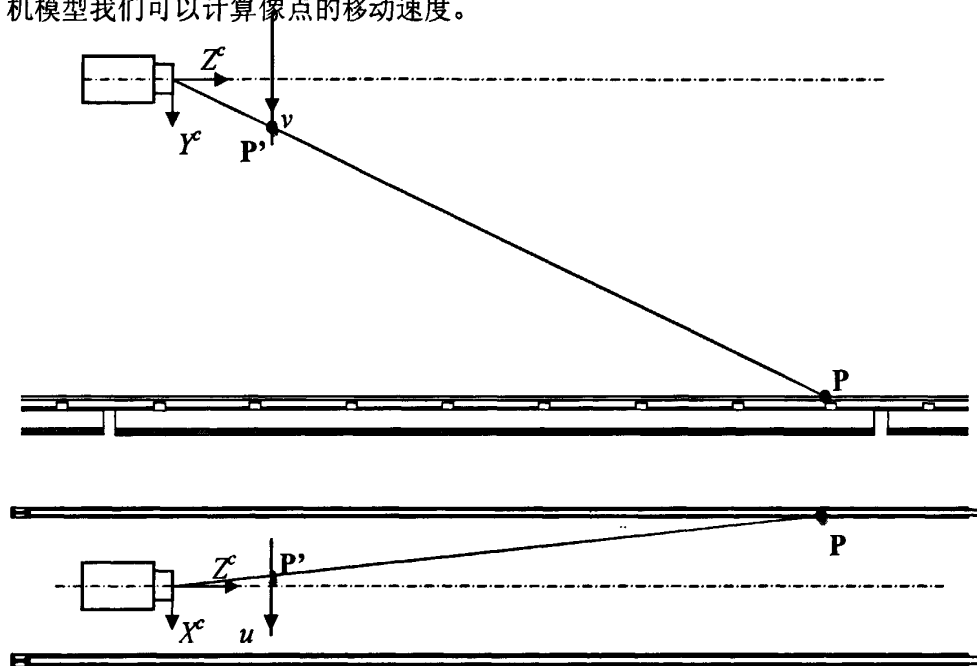


图 2-6 像点移动速度计算简图

Fig.2-6 The simplified diagram for computing of the speed of the image point

相对于列车的行使速度，列车在其他自由度上振动的速度可以忽略不计。因此可以认为对于某个选定的点 P，方程 (2-1) 和 (2-2) 右侧的变量中只有 z^c 是关于时间的函数。则对方程 (2-1) 和 (2-2) 的两边对时间 t 求导可得到 P' 点在两个坐标轴方向上运动速度的分量：

$$\frac{du}{dt} = \frac{d}{dt} \left(\frac{f \cdot x^c}{z^c} \right) = -\frac{f \cdot x^c}{(z^c)^2} \cdot \frac{dz^c}{dt} \quad (2-14)$$

$$\frac{dv}{dt} = \frac{d}{dt} \left(\frac{f \cdot y^c}{z^c} \right) = -\frac{f \cdot y^c}{(z^c)^2} \cdot \frac{dz^c}{dt} \quad (2-15)$$

P' 点在图像平面上的移动速度 v_p 为这两个方向上速度的合成：

$$v_p = \sqrt{\left(\frac{du}{dt} \right)^2 + \left(\frac{dv}{dt} \right)^2} = \frac{f \cdot \sqrt{(x^c)^2 + (y^c)^2}}{(z^c)^2} \cdot \left| \frac{dz^c}{dt} \right| \quad (2-16)$$

从式中可以看出 v_p 和相机焦距、物点的空间位置以及列车的移动速度有关。

为了确定最大允许的曝光时间，我们要求出在感兴趣的场景中像点的最大移动速度。根据式(2-16)可以看出，这点是感兴趣场景中离相机纵向距离最近，而横向偏离相机中心线最远的点。

(2) 帧频

帧频是指相机每秒钟能够输出的图像数目。如前所述，本系统设计每帧图像处理 10m 左右的线路图像。这样当检测速度到达 400km/h 时，每秒钟需要处理 12 帧图像。所以相机的帧频只要大于 12fps 即可。一般的相机都可以满足这个要求，因此本系统中相机的帧频并不是主要考虑的因素。

(3) 分辨率

分辨率是相机最为重要的性能参数之一，这里指的相机分辨率主要是位于 CCD 芯片上的像素数。通常分辨率越高，其分辨物体细节的能力就越强有利于图像处理 and 识别。但是分辨率越高图像传输和处理的速度就会降低，因此在本系统中我们选择能够满足处理速度要求情况下最大可能的分辨率。

(4) 灵敏度

相机的灵敏度反映了相机采用的 CCD 芯片的光电转换能力，灵敏度越高相机在弱光条件下拍摄的能力就越强。目前有多种提高相机灵敏的技术，如像增强、背照 CCD、微透镜、片内倍增等技术。前面已经指出本系统需要较小的曝光时间，因此在选择相机时我们选择了具有较高灵敏度的相机。

2.3 图像传输与采集

2.3.1 图像传输方式选择

图像传输就是将相机获取的图像通过一定的方式传输到图像采集卡。由于图像信传输的数据量通常很大，并且随着相机的分辨率和帧频不断提高，因此需要图像传输通道具有较高的带宽和传输可靠性。目前、图像传输技术种类较多各有特点，可以分为模拟和数字两大类。其中常用的主要有^[7]：

(1) 模拟视频信号

模拟视频标准制定于 20 世纪 40 年代。由于该标准应用时间较长，制作模拟摄像机的器件相对便宜。因此许多摄像机还是采用模拟信号进行传输。常用的模拟视频信号标准有 NTSC、PAL 等。模拟相机通常分辨率和帧频都较低，因此限制了它在高端领域的应用。

(2) CameraLink 接口

CameraLink 是目前机器视觉领域通用的数字摄像机和图像采集卡之间的接口。

它不仅定义了数字信号的物理传输方式还定义了接口的机械尺寸等。CameraLink 采用的基本技术是低电压差分信号技术 LVDS, 其最大的优点是传输速度非常高。CameraLink 由驱动器和接收器对组成, 一个驱动器和接收器将 28 个信号按 7: 1 的比例串行传输, 即转换为 4 对信号线按 LVDS 传输, 时钟通过第 5 对信号传输。28 个信号中包括可帧有效、行有效、数据有效等控制信号和 24 个数据信号。根据采用的驱动器和接收器对的个数不同 CameraLink 可以分为 Base、Medium、Full 三种不同的配置, 它们分别采用了 1 至 3 个驱动器和接收器对。CameraLink 最大时钟频率可以达 85MHz, 当采用 Full 配置时最大的传输速率可达 680MB/s。

(3) 千兆以太网

以太网是被广泛采用的局域网物理层标准。目前以太网标准 IEEE802.3 定义了速率为 10Mb/s、100Mb/s、1Gb/s 以及 10Gb/s。由于千兆以太网的广泛使用和高速率, 在传输高速图像方面有很大的优势。2006 年机器视觉摄像机应用层协议得以标准化, 并称为 GigE Vision。GigE Vision 是基于 UDP 协议的, 以便使可传输的数据量最大化。再网络上没有其他负载的情况下, 对于千兆以太网 UDP 的速率可以达到 600~840Mb/s。但是由于 UDP 协议不可靠可能需要重发等, 且以太网采用 CSMA/CD 结构, 实际的带宽通常无法保证。

(4) USB 2.0

最初 USB 接口的速率比较低, 只能连接较低速的设备。在图像设备方面也仅能作为网络摄像机这类分辨率较低的设备接口方式。目前 USB2.0 支持的最大传输速率达到了 480Mb/s, 为在机器视觉中应用提供了可能。虽然 USB 中有视频类协议, 但是机器视觉的 USB 摄像机厂商通常使用自己的传输协议和设备驱动来传输图像。

(5) IEEE 1394

IEEE 1394 又称作火线, 是高速串行总线标准。使用 6 针接插件, 数据传输时使用 2 对双绞线传输信号、另外一根电源和一根地线。对于低功耗的产品可以直接使用电缆提供的电源。最新版本的 IEEE 1394b 定义了 800Mb/s 和 1600Mb/s 速率以及 3200Mb/s 的结构。在机器视觉领域通常采用 IIDC 标准, 它是对 IEEE 1394 针对工业摄像机进行修改后的协议。IIDC 定义了多种视频输出格式, 包括分辨率、帧频及传输的像素数据格式等。另外 IIDC 还定义了标准的方法来控制摄像机的设置, 比如快门、光圈、增益等。IIDC 使用 IEEE 1394 的异步模式来传输控制信号, 而使用等时传输模式传输图像信号。

其中 CameraLink 接口是通用的工业标准, 具有传输速度快, 可靠性高等特点。并且市场上 CameraLink 接口的高性能相机和采集卡较多, 与光纤的转换也较容易, 因此本系统采用了 CameraLink 接口来进行图像传输。

2.3.2 图像采集

图像采集就是把相机输出的图像信号传输到计算机内存中的过程。这个过程需要使用专门的图像采集卡与计算机进行数据通信，本系统采用了加拿大 DALSA 公司生产的 Anaconda 图像采集与处理板卡，其功能结构图如图 2-7 所示^[13]。该卡采用 PCI-X 借口与 PC 相连；图像采集控制器 (ACU) 能够采集一个 Full、Medium 或者 Base 的 CameraLink 借口的相机，最大时钟频率为 85MHz；带有一块基于 FPGA 的图像处理单元 (IPU)，用户可以自行设计并配置其内部逻辑，以实现对图像的实时硬件处理，从而提高图像处理速度。数据传输控制器 (DTE) 实现了各个单元之间的高速数据交换。板卡还提供了大量的高速存储器来存储图像数据和处理的中间结果，其中和 IPU 相连的主要包括 2 块共 128MB 的 DDR SDRAM 作为大块连续图像数据的存储，以及 4 块共 16MB 的 SBSRAM 高速存储器，来存储图像处理中的中间结果；另有高达 2GB 的存储器和 DTE 相连，作为图像采集和传输的缓存。

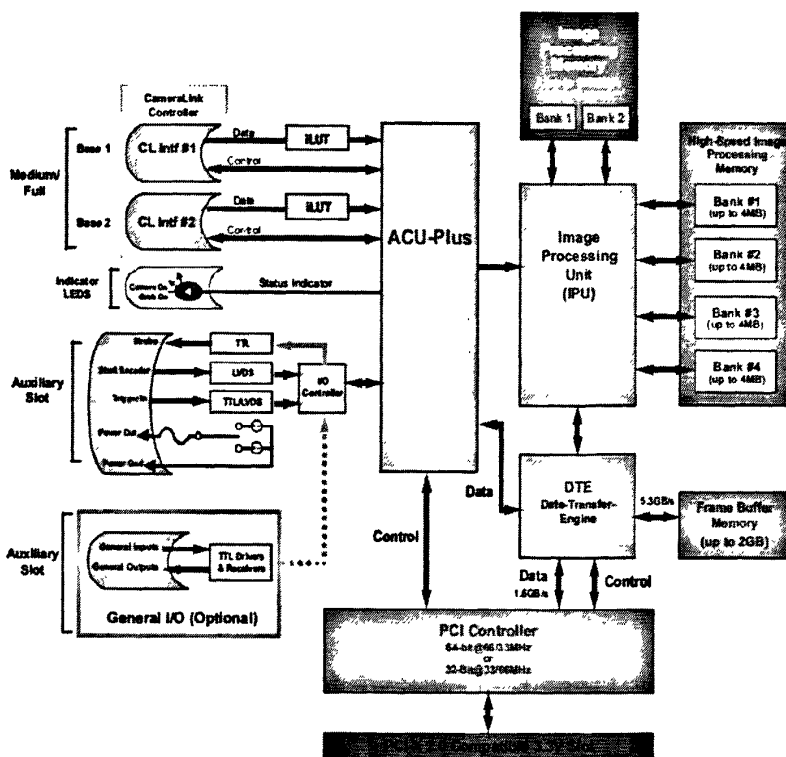


图 2-7 Anaconda 功能结构图

Fig. 2-7 Functional block diagram

在本系统中图像采集主要的问题是如何保证两个相机的同步。因为如果两个相机的曝光过程前后相隔一定时间，则在列车高速行驶的情况下两相机拍摄的将是不同的线路，这样在进行立体视觉测量时将产生错误的结果。如图 2-7 所示，

Anaconda 图像采集和处理卡带有外部触发输入端口，本系统就利用了这些端口来进行两个相机的同步。系统的主控计算机将根据距离同步信号，产生相机的触发信号同时输入给所有的板卡。板卡在收到外部触发信号后将立即触发相机进行曝光，这样就保证了两个相机的同步采集。

2.4 基于 FPGA 的实时图像预处理

很多机器视觉应用对图像处理的速度有较高的要求，如机器人视觉系统，高速生产线检测系统等。事实上，几乎所有的图像处理系统都希望能够尽快给出处理结果。但是由于图像信息的数据量非常的大，提高图像处理系统的处理速度并不容易。传统的处理器通常很难满足高速处理的要求。因此实时图像处理系统或高速图像处理系统通常都包含专门的硬件，如 FPGA、DSP、专用的 VLSI 等；或者是采用了专门的处理器结构，如 SIMD、MIMD 等^[14]。

FPGA 即现场可编程门阵列，是一种可编程逻辑器件。其内部含有大量的逻辑门并可根据需要对其进行配置和连接以实现不同的逻辑功能。经过数十年的不断发展，目前 FPGA 已经可以实现非常复杂的逻辑电路。在图像处理领域，FPGA 可灵活配置实现特殊的处理算法，特别是在在底层处理如像素变换、邻域处理等方面有很大的优势；另外高速输入输出也是 FPGA 的优势，因此在高速图像采集方面 FPGA 已经成为主要的实现手段。

在本系统中图像的分辨率较高，同时需要每秒钟 10 帧以上的处理速度，图像处理的数据量非常的大。因此我们采用了 FPGA 来进行底层的图像处理以提高处理速度，并利用 Anaconda 图像采集和处理板卡为平台来实现基于 FPGA 的图像预处理^[15]。

上一节已经介绍了 Anaconda 板卡的功能结构，可以实现图像采集、传输以及预处理等功能。其中用来实现图像预处理功能的是一块基于 FPGA 的图像处理单元 (IPU)，采用的是 Xilinx Virtex-II Pro 系列 FPGA，型号为 XC2VP-5FF896，其内部包含 20880 个逻辑单元，88 个 18bits×18bits 乘法器以及多达 1847Kb 的内存，还包含 2 个 Power PC 处理器内核。我们使用了 VHDL 语言，利用 Xilinx 公司提供的开发工具如 ISE、Chipscope 等来对其进行开发，并配合使用 Aldec active HDL、Modelsim 等软件来进行仿真等。

IPU 除了实现图像处理外，还需要实现内存读取、数据传输等功能，其内部逻辑结构图如图 2-8 所示。IPU 内部除了包括图像处理硬件逻辑外，还包括了图像总线接口、配置总线接口、DDR SDRAM 接口以及 SBSRAM 接口等。图像处理硬件是其核心，其设计方法将在第 3 章中详细介绍，本节将介绍其他部分的逻辑。

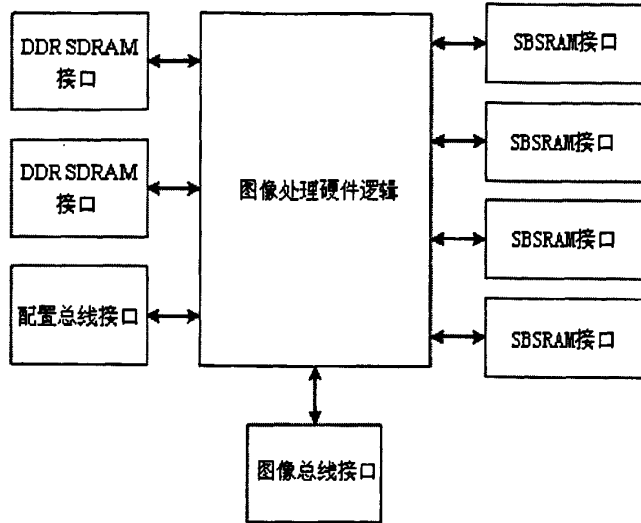


图 2-8 图像处理单元内部逻辑结构

Fig. 2-8 Internal architecture of the image processing unit

2.4.1 配置总线接口

配置总线接口 (PBI) 是 FPGA 中用户逻辑和 PC 软件之间的低速总线接口。PBI 可以实现 PC 与 FPGA 之间低速数据的通讯, 例如 PC 软件可以通过 PBI 读写 FPGA 内部用户逻辑的寄存器, 从而实现软件对用户逻辑的控制。PBI 是一个简单的并行总线接口, 由地址线, 数据线和控制线组成, 表 2-1 列出了 PBI 总线组成信号的名称以及类型, PBI 接口的 Active HDL 仿真时序如图 2-9 所示。

表 2-1 PBI 接口信号

Table 2-1 Signals of PBI

信号名	方向	类型	功能
iPBI_clk	in	std_logic	时钟信号, 频率 33MHz。
iPBI_rst_n	in	std_logic	复位信号, 低电平有效。
ivAddress	in	std_logic_vector(15 downto 0)	地址信号。
ivData	in	std_logic_vector(15 downto 0)	数据输入信号, 由 PC 至 FPGA。
ivBE	in	std_logic_vector(1 downto 0)	字节有效信号, ivBE(0)高电平时表示 ivData(7 downto 0)有效; ivBE(1)高电平时表示 ivData(15 downto 8)有效。
iWE	in	std_logic	写使能信号, 高电平有效。
ovData	out	std_logic_vector(15 downto 0)	数据输出信号, FPGA 至 PC。

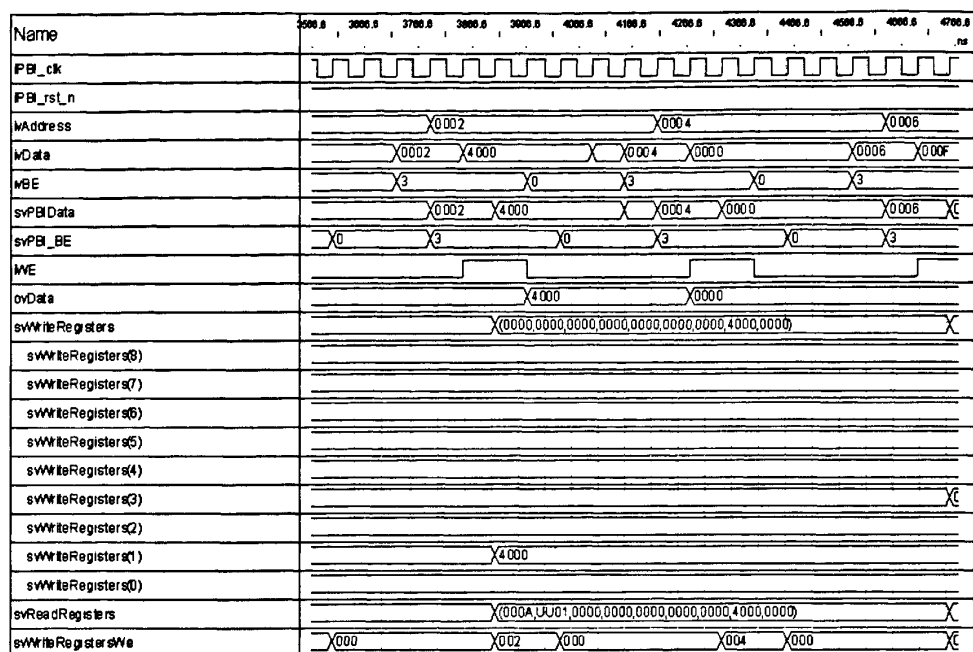


图 2-9 PBI 接口时序图

Fig. 2-9 Waveform of PBI

PBI 总线写时序和一般的并行总线类似，先输出地址信号，随后输出数据信号和写使能行好，通知 FPGA 进行一次写操作。PBI 总线没有读使能信号，这是因为这里的总线实际上是点对点的传输，并是全双工通讯，因此读使能实际上可以认为一直有效。即 ovData 始终输出 ivAddress 指定单元的数据。

2.4.2 图像数据接口

图像数据接口用于高速图像信号的输入和输出。图像数据为 64 位并行总线，总线时钟频率为 133MHz，传输速率可达 1GB/s。完全可以满足高速高清图像传输的传输要求。图像数据接口的信号定义见表 2-2:

表 2-2 图像数据接口信号

Table 2-2 Signals of image bus interface

信号名	方向	类型	功能
iEXP_ddr_clk	in	std_logic	时钟信号，频率 133MHz。
ovLineWidthM1	out	std_logic_vector(11 downto 0)	图像宽度减 1
ovNbLinesM1	out	std_logic_vector(15 downto 0)	图像高度减 1

表 2-2(续) 图像数据接口信号

Table 2-2 (continual) Signals of image bus interface

ivData_B2E	in	std_logic_vector(63 downto 0)	输入图像数据。该总线用来传输采集到的原始图像数据。
ivParity_B2E	in	std_logic_vector(7 downto 0)	输入图像标志。用来指示图像的行、场的开始和结束，具体定义见表 2-3。
iDataValid_B2E	in	std_logic	图像数据有效，该信号高时表示 ivData_B2E 上的数据是有效的图像数据。
iDataReady_B2E	in	std_logic	图像数据传输准备，该信号高时表示图像采集与传输控制器有图像数据可以传输，用户逻辑可以在下一个时钟开始读取图像数据。
oReadEnable_B2E	out	std_logic	图像数据读取使能。
ovData_E2B	out	std_logic_vector(63 downto 0)	输出图像数据，用来传输 FPGA 处理后的图像数据。
oWriteEnable_E2B	out	std_logic	图像输出使能。

表 2-3 图像输入标志定义

Table 2-3 Definition of image input flag

位	定义
ivParity_B2E(7)	帧结束标志，高电平时表示当前数据是该帧的最后一个数据。
ivParity_B2E(6)	保留。
ivParity_B2E(5)	行结束标志，高电平时表示当前数据是该行的最后一个数据。
ivParity_B2E(4)	总是 0。
ivParity_B2E(3)	保留。
ivParity_B2E(2)	帧起始标志，高电平时表示当前数据是该帧的第一个数据。
ivParity_B2E(1)	行起始标志，高电平时表示当前数据是该行的第一个数据。
ivParity_B2E(0)	总是 1。

读取图像数据的 Active HDL 仿真时序如图 2-10 所示，当图像采集单元有新的

图像数据可以传输时，iDataReady_B2E 输出高电平，此时图像处理单元可以使能 oReadEnable_B2E（在图 2-10 中 oReadEnable_B2E 始终是有有效的）。图像采集单元接收使能信号后将 iDataReady_B2E 拉低准备数据传输，一个时钟后在 ivData_B2E 上输出有效的数据并使能 iDataValid_B2E。图像处理单元在检测到 iDataValid_B2E 为高时读取数据，从而完成一次图像数据的传输。

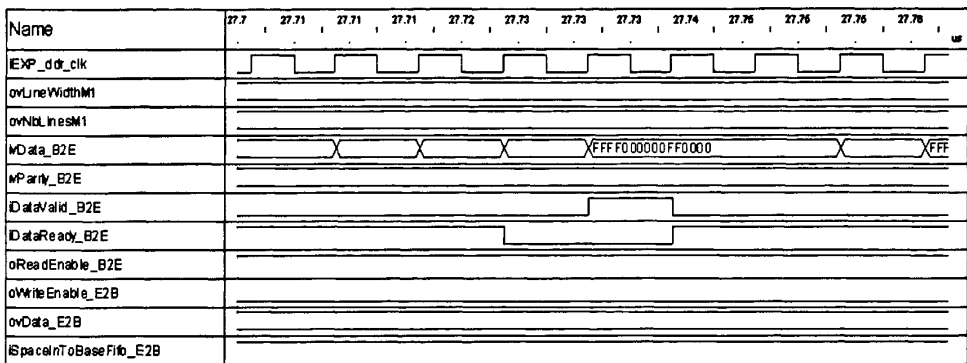


图 2-10 图像数据总线读时序

Fig. 2-10 Read waveform of image bus interface

写图像数据的 Active HDL 仿真时序如图 2-11 所示。写时序相对读时序要简单很多，当有图像数据需要输出时，只要在 ovData_E2B 上输出有效的数据并使能 oWriteEnable_E2B 后保持一个时钟的时间即可。

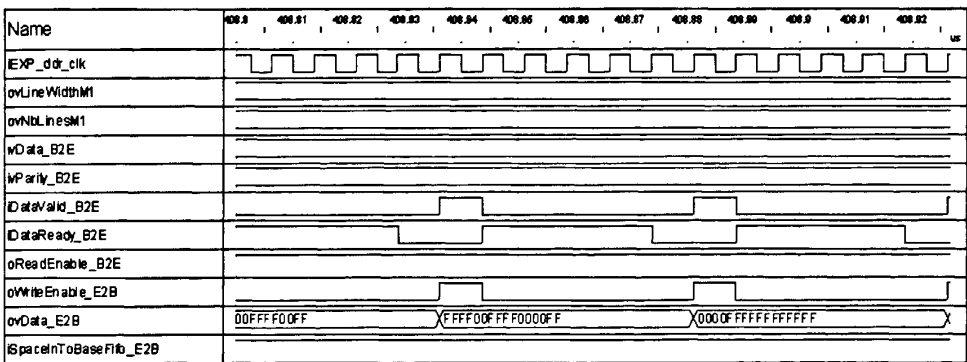


图 2-11 图像数据总线写时序

Fig. 2-11 Write waveform of image bus interface

图 2-12 为利用 Chipscope 软件获得的图像输入输出时序图，它更明确地说明了 ivParity_B2E 的作用。

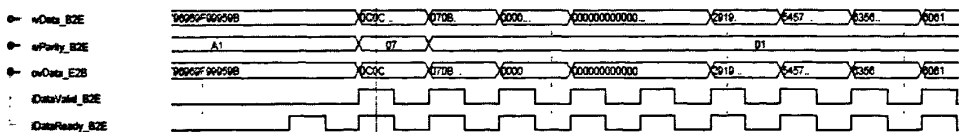


图 2-12 Chipscope 逻辑分析仪获取的波形

Fig. 2-12 Waveform grabbed by Chipscope Logic Analyzer

2.4.3 DDR SDRAM 接口模块

DDR SDRAM 可以快速地进行大量连续数据的存取, 适合存储图像数据和图像处理的中间结果。系统设计了 DDR SDRAM 的控制模块, 实现了对 DDR SDRAM 的控制、读写等, 并为图像处理核心硬件提供了较为简单的接口。这样核心硬件无需直接操纵 DDR SDRAM, 就不需要考虑 DDR SDRAM 物理接口的细节问题, 如配置, 刷新等, 从而简化了核心硬件的开发工作, 只需要了解该控制模块提供的接口的使用方法即可。核心硬件和 DDR SDRAM 控制模块的接口由如下信号组成。

(1) iConfigComplete_DDR1 : in std_logic;

配置完成信号。DDR SDRAM 上电后, 首先需要对其进行一定的配置方可以进行读写等操纵。对 DDR SDRAM 的配置在控制模块中实现, 当配置完成后控制模块在 iConfigComplete_DDR1 上输出高电平, 指示用户可以进行读写操作。

(2) oTransferRequest_p_DDR1 : out std_logic;

传输请求信号。当用户需要进行数据传输时(读或写), 需要通过在此信号上输出高电平, 来请求传输。

(3) iTransferDone_p_DDR1 : in std_logic;

传输完成信号。传输完成信号为高电平时表示本次传输结束。

(4) oTransferMemory2Local_DDR1 : out std_logic;

(5) oTransferLocal2Memory_DDR1 : out std_logic;

oTransferMemory2Local_DDR1 和 oTransferLocal2Memory_DDR1 信号表示本次数据传输的方向。当 oTransferMemory2Local_DDR1 输出高电平时, 表示本次传输为从内存读取数据。当 oTransferLocal2Memory_DDR1 输出高电平时, 表示本次传输为将数据写入内存。用户应避免在这两个信号上同时输出高电平。

(6) oTransferAddressIncrement_DDR1 : out std_logic;

地址变化方向设置信号。本信号用于设置连续传输时地址的变化的方向。该信号为高电平时地址增大, 低电平时地址减小。

(7) ovTransferAddress_DDR1 : out std_logic_vector(31 downto 0);

传输的初始地址。用来设置单次传输的起始地址。本设计不支持字节存取, 地址必须按八字节对齐。因此 ovTransferAddress_DDR1(2 downto 0)是无效的。

(8) ovTransferSize_DDR1 : out std_logic_vector(12 downto 0);

传输数据数量。指定单次传输的数据量, 以字节为单位。最小为 8 字节, 这是因为数据线宽度为 32 位, 且 DDR SDRAM 在每个时钟的上升沿和下降沿个进行一次数据传输, 所以每次至少传输 8 字节的数据。最大为 4096 字节。

- (9) iClkLocal2Memory_r_DDR1 : in std_logic;
 写内存上升沿时钟。该时钟的上升沿和 DDR SDRAM 写时钟的上升沿同步。
 所有写操作的上升沿数据和控制信号都与该时钟同步。
- (10) iClkLocal2Memory_f_DDR1 : in std_logic;
 写内存下降沿时钟。该时钟的上升沿和 DDR SDRAM 写时钟的下降沿同步。
 所有写操作的下降沿数据和控制信号都与该时钟同步。
- (11) iReadLocal2Memory_r_DDR1 : in std_logic;
 上升沿读有效信号。
- (12) iReadLocal2Memory_f_DDR1 : in std_logic;
 下降沿读有效信号。
- (13) ovDataLocal2Memory_r_DDR1 : out std_logic_vector(31 downto 0);
 上升沿读出数据。
- (14) ovDataLocal2Memory_f_DDR1 : out std_logic_vector(31 downto 0);
 下降沿读出数据。
- (15) iClkMemory2Local_r_DDR1 : in std_logic;
 读内存上升沿时钟信号。该时钟的上升沿和 DDR SDRAM 读时钟的上升沿同步。所有读操作的上升沿数据和控制信号都与该时钟同步。
- (16) iClkMemory2Local_f_DDR1 : in std_logic;
 读内存下降沿时钟信号。该时钟的上升沿和 DDR SDRAM 读时钟的下降沿同步。所有读操作的下降沿数据和控制信号都与该时钟同步。
- (17) iWriteMemory2Local_r_DDR1 : in std_logic;
 上升沿写有效信号。
- (18) iWriteMemory2Local_f_DDR1 : in std_logic;
 下降沿写有效信号。
- (19) ivDataMemory2Local_r_DDR1 : in std_logic_vector(31 downto 0);
 上升沿写入数据。
- (20) ivDataMemory2Local_f_DDR1 : in std_logic_vector(31 downto 0);
 下降沿写入数据。

DDR SDRAM 写操作的总体时序如图 2-13 所示。DDR SDRAM 适合快速的存取连续地址的大批数据，对于单个地址的随机存取则相对效率较低。因此 DDR SDRAM 的读写操作通常都是分批进行的，即一次写入一定字节的连续地址的数据。如图 2-13 所示的写操作每次写入了 1024 个字节的数据。

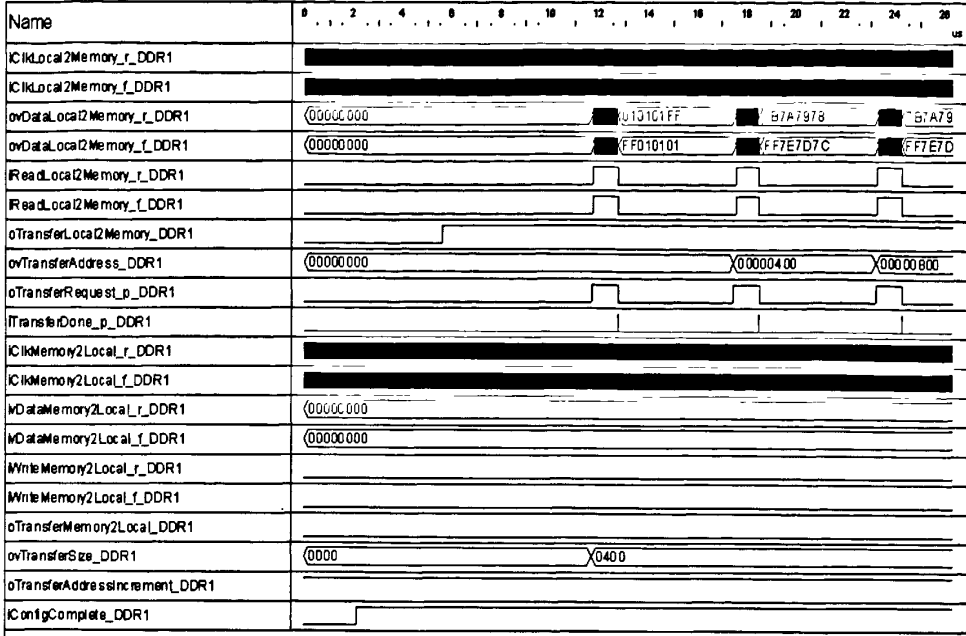


图 2-13 DDR SDRAM 写操作总体时序

Fig. 2-13 Write waveform of DDR SDRAM operation

DDR SDRAM 在进行读写操作之前首先要对其进行配置，在本系统中配置的操在 DDR SDRAM 的控制器中实现，当其配置完成时 iConfigComplete_DDR1 输出高电平。因此用户在对 DDR 进行读写时必须先确认 iConfigComplete_DDR1 为高电平。当要进行一次写操作时首先要通过各个配置信号线设置本次传输的参数，包括传输的首地址、地址增长方向、本次传输数据的字节数以及传输的方向。正确输出配置参数后就可以在 oTransferRequest_p_DDR1 上输出高电平以向 DDR SDRAM 请求传输，当 DDR SDRAM 准备好可以读取数据时，将在 iReadLocal2Memory_r_DDR1 和 iReadLocal2Memory_f_DDR1 输出高电平。此时用户只要根据时钟同步地，依次输出要写入的数据即可，如图 2-14 所示。当指定的传输数据个数达到时，iTransferDone_p_DDR1 将被使能，表示本次传输过程结束。这样就完成了一次写操作。

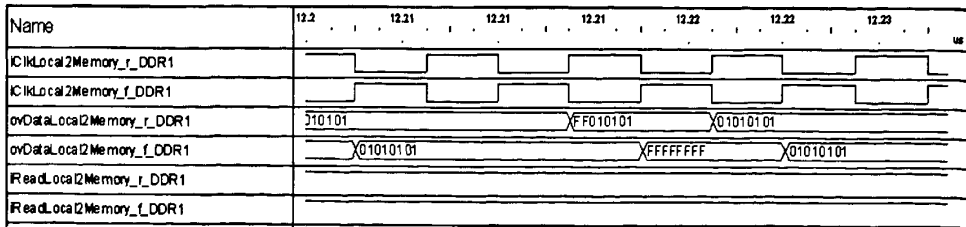


图 2-14 DDR SDRAM 写操作时序细节

Fig. 2-14 Detail waveform of DDR SDRAM write operation

DDR SDRAM 的读操作和写操作非常类似，都首先需要在相应的配置端口

设置传输参数，然后请求传输，其时序如图 2-15 所示。与写时序不同的是在 DDR SDRAM 确认传输请求后将在 iWriteMemory2Local_r_DDR1 以及 iWriteMemory2Local_f_DDR1 输出高电平，图像处理逻辑在检测到高电平后就可以在相应时钟的上边沿读取 ivDataMemory2Local_r_DDR1 和 ivDataMemory2Local_f_DDR1 端口上的数据。同样当指定的传输数据个数达到时，iTransferDone_p_DDR1 将被使能，表示本次传输过程结束。DDR SDRAM 度操作的细节如图 2-16 所示。

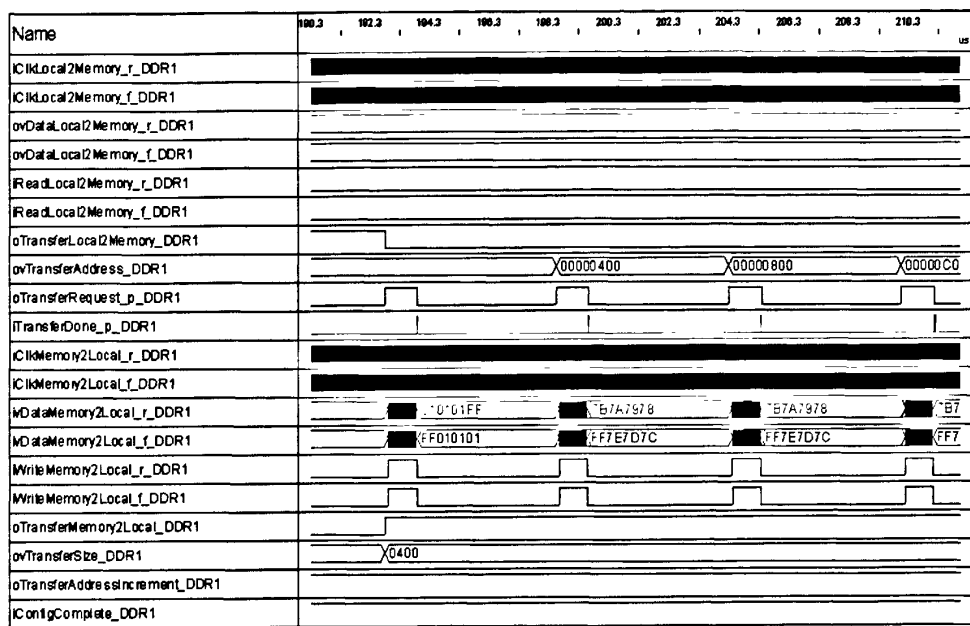


图 2-15 DDR SDRAM 读操作总体时序

Fig. 2-15 General waveform of DDR SDRAM read operation

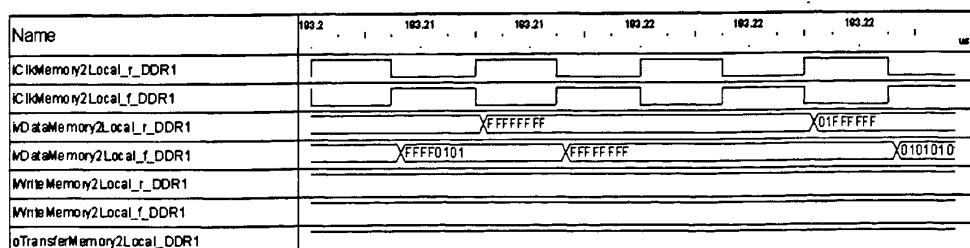


图 2-16 DDR SDRAM 读操作时序细节

Fig. 2-16 Detail waveform of DDR SDRAM read operation

2.5 系统软件结构

本系统的软件主要包括 Anaconda 板卡的驱动程序、FPGA 硬件处理的控制程序、高层图像处理及识别程序以及通信接口程序等四个部分。图 2-17 为本系统软

硬件的结构图，表示了系统硬件和软件的接口，以及它们的层次关系。

在软件方面 Anaconda 板卡的驱动程序位于最底层，它负责建立软件和硬件的接口，实现程序对板卡的控制和从板卡读取图像数据等。驱动程序的上一层是 FPGA 硬件处理的控制程序，控制程序主要通过读写寄存器的方式控制 FPGA 中自行开发的实时图像处理硬件。由于这些硬件的功能是根据不同应用专门设计的，因此控制程序也需要专门进行设计。在 IPU 硬件设计时用户需要设计相应的寄存器来实现软件对硬件的控制，在软件方面 Dalsa 公司提供了称为 SapRTPro 的类，用户可以根据需要创建一个 SapRTPro 派生类来实现对 IPU 内部寄存器的读写。

在程序结构的最上层是高层图像处理和识别程序以及通信接口程序。在高层图像处理和识别程序用来完成最终线路状态的识别，结果的显示存储等。在这部分我们使用了 HALCON 图像处理软件库来加快软件的开发，以及提供软件的效率。HALCON 是一个专业的图像处理和机器视觉工具，它提供了大量的图像处理操作的库函数和一些接口程序来加快机器视觉应用的开发速度。

通信接口程序负责各个子系统和主机的通信，同时也负责和综合检测列车其它系统的通信。

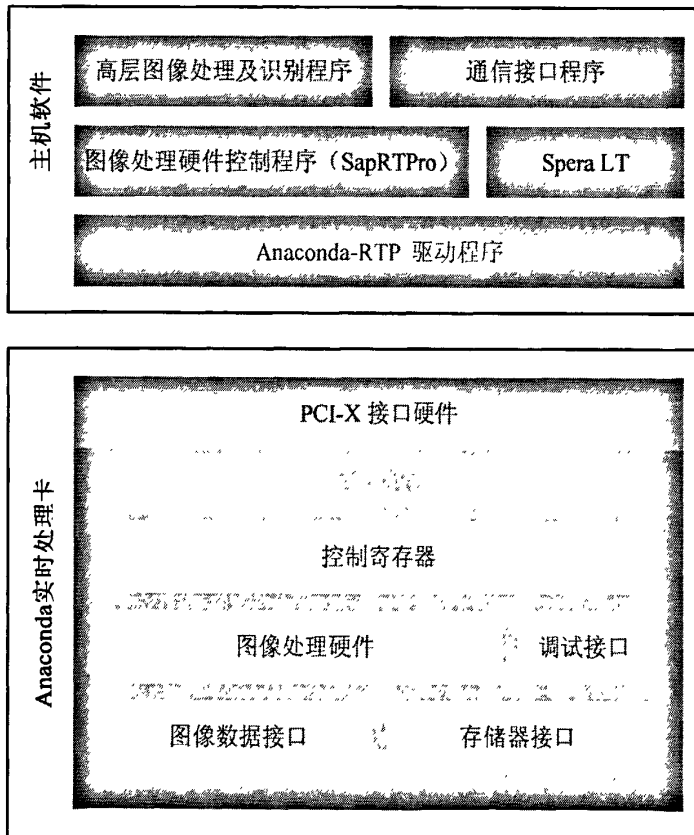


图 2-17 系统软硬件结构图

Fig. 2-17 Architecture of software and hardware

3 基于图像代数的图像处理算法硬件实现技术

随着机器视觉技术的不断发展,其应用也领域不断扩大,许多机器视觉应用对图像处理的实时性要求较高。本文研究的高速铁路线路环境智能识别系统正是如此,当综合检测列车需要以 350~400km/h 的速度对线路进行检测,系统每秒需要实时处理 100m 左右线路的图像。若完全采用软件实现图像处理算法,由于受到现有计算机体系结构的限制,通常只能采用串行实现,实时性较差不能满足系统的要求。

近年来,特别是在 FPGA、可编程 ASIC 等大规模可编程逻辑器件得到广泛应用后,图像处理算法的硬件实现技术成为比较热门的研究领域,许多研究者利用 FPGA 实现了一些典型算法的硬件设计^[16-18]。由于硬件设计可以灵活的使用并行处理结构,从而大大提高了处理的实时性。但是,目前硬件实现技术还只能针对具体的一种或者一类图像处理算法单独设计相应的硬件实现,还没有形成一套通用的方法。因此设计和调试需要花费较长的时间,并且对于较复杂的算法实现难度很大。

图像代数(Image Algebra)理论是最近发展起来的关于图像变换和分析的数学理论,其研究目标是为图像变换、图像分析、图像理解等是建立一套广泛而又统一的数学理论^[19-21]。利用图像代数描述图像处理算法可以使得程序编写简化,并且图像代数在本质上是并行的,已经有很多基于图像代数的并行图像处理环境问世^[22,23]。本章提出了一种基于图像代数的图像处理算法的硬件实现技术,其基本思想是将图像代数中的基本元素和基本运算利用硬件实现,形成基础硬件库。在实现图像处理算法时,就可以根据该算法的图像代数表示,调用相应的基础硬件即可构成该算法的硬件处理系统。充分利用图像代数的并行处理特性,并使设计的硬件具有更大的通用性。

3.1 图像代数

图像代数可以看作是一种为表示图像处理算法而设计的数学语言,图像代数的用处是为建立这些算法提供一个通用的数学环境。图像代数本质上是并行的,可以方便地在并行机上实现。已经证明用图像代数描述任何图像处理算法在理论上都是可行的。本节将介绍与本章内容密切相关的图像代数的一些基本概念和理论基础。本节只介绍了图像代数中最基础和与本文内容相关的内容,关于图像代数的详细理论,有兴趣的读者可以阅读相关专门文献^[19,20]。

3.1.1 点集及其运算

点集 (Point Sets) 即点组成的集合, 点集是一个拓扑空间。一个点集由两部分组成, 即点和拓扑关系。点集通常由粗体大写字母标示, 例如 $\mathbf{W}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}$ 。

点通常由粗体的小写字母标示如 $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{X}$, 如果点 $\mathbf{x} \in \mathbb{R}^n$, \mathbf{x} 可以标示为 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 。

在图像处理中通常采用的点集都为 n 维欧拉空间 \mathbb{R}^n 。但在计算机视觉算法中也常常使用其它拓扑空间, 如冯·诺伊曼拓扑(von Neumann topology)、奇偶积拓扑(odd-even product topology)等。图像代数对点集的形状没有任何限制, 可以是矩形、圆形等。在数字图像处理应用中较常用的点集有整数点集合 $\mathbb{Z} \subset \mathbb{R}$ 以及 n 维整数点集 $\mathbb{Z}^n = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} = (x_1, \dots, x_n), x_i \in \mathbb{Z}, i = 1, \dots, n\}$ 等。其中两种最常遇到的点集为:

$$\mathbf{X} = \mathbb{Z}_m \times \mathbb{Z}_n = \{(x_1, x_2) \in \mathbb{Z}^2 : 0 \leq x_1 \leq m-1, 0 \leq x_2 \leq n-1\} \quad (3-1)$$

以及

$$\mathbf{X} = \mathbb{Z}_m^+ \times \mathbb{Z}_n^+ = \{(x_1, x_2) \in \mathbb{Z}^2 : 1 \leq x_1 \leq m, 1 \leq x_2 \leq n\} \quad (3-2)$$

点集可以用图示的方法表示, 图 3-1 给出了式 (3-2) 所表示的点集的示意图。

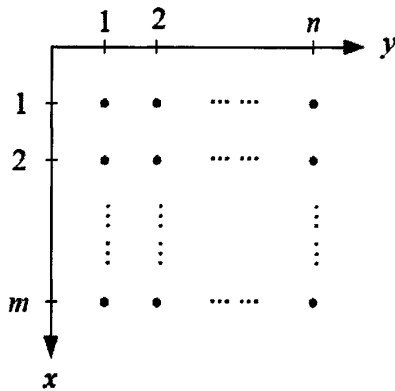


图 3-1 点集示意图

Fig. 3-1 Schematic diagram of a point set

常用的点集都是向量空间 \mathbb{R}^n 的子集, 因此点集继承了所有向量空间的基本运算。例如, 设 $\mathbf{X} \subset \mathbb{R}^n, \mathbf{x} = (x_1, \dots, x_n), \mathbf{y} = (y_1, \dots, y_n) \in \mathbf{X}, k \in \mathbb{R}$, 则点以及点与标量的各种运算可以按如下定义。

点加:

$$\mathbf{x} + \mathbf{y} = (x_1 + y_1, \dots, x_n + y_n) \quad (3-3)$$

标量乘:

$$k \cdot \mathbf{x} = (k \cdot x_1, \dots, k \cdot x_n) \quad (3-4)$$

标量加:

$$k + \mathbf{x} = (k + x_1, \dots, k + x_n) \quad (3-5)$$

除了向量空间的基本运行外, 图像代数还引入了三种点的乘法运算

$$\mathbf{x} \cdot \mathbf{y} = (x_1 \cdot y_1, \dots, x_n \cdot y_n) \quad (3-6)$$

$$\mathbf{x} \times \mathbf{y} = (x_2 \cdot y_3 - x_3 \cdot y_2, x_3 \cdot y_1 - x_1 \cdot y_3, x_1 \cdot y_2 - x_2 \cdot y_1) \quad (3-7)$$

$$\mathbf{x} * \mathbf{y} = x_1 y_1 + x_2 y_2 + \dots + x_n y_n \quad (3-8)$$

3.1.2 图像及其运算

在图像代数中, 主要的操作对象是图像、模板以及领域等。其中图像是最基础的类型, 因为在图像代数中模板和领域都可以看作是特殊的图像。设 \mathbf{X} 为一个点集, 图像 \mathbf{a} 的定义域为 \mathbf{X} , 则在图像代数中图像 \mathbf{a} 可以表示为:

$$\mathbf{a} = \{(\mathbf{x}, \mathbf{a}(\mathbf{x})) : \mathbf{x} \in \mathbf{X}\} \quad (3-9)$$

其中 $(\mathbf{x}, \mathbf{a}(\mathbf{x}))$ 是图像的一个元素即像素, \mathbf{x} 为像素位置, $\mathbf{a}(\mathbf{x})$ 为图像 \mathbf{a} 在 \mathbf{x} 上的像素值。设 $\mathbf{a}(\mathbf{x}) \in \mathbb{F}$, 则图像 \mathbf{a} 可以看作是 \mathbf{X} 到数域 \mathbb{F} 的映射即 $\mathbf{a} : \mathbf{X} \rightarrow \mathbb{F}$, 记为 $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ 。

本文仅对实图像进行讨论即设 $\mathbf{a}(\mathbf{x}) \in \mathbb{R}$, 下面引入图像的基本运算。

设 $\mathbf{a}, \mathbf{b} : \mathbf{X} \rightarrow \mathbb{R}$, $k \in \mathbb{R}$, 则图像的几个基本运算可以定义为:

$$\mathbf{a} + \mathbf{b} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) + \mathbf{b}(\mathbf{x}), \mathbf{x} \in \mathbf{X}\} \quad (3-10)$$

$$\mathbf{a} \cdot \mathbf{b} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) \cdot \mathbf{b}(\mathbf{x}), \mathbf{x} \in \mathbf{X}\} \quad (3-11)$$

$$\mathbf{a} \vee \mathbf{b} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) \vee \mathbf{b}(\mathbf{x}), \mathbf{x} \in \mathbf{X}\} \quad (3-12)$$

$$\mathbf{a} \wedge \mathbf{b} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) \wedge \mathbf{b}(\mathbf{x}), \mathbf{x} \in \mathbf{X}\} \quad (3-13)$$

$$k \cdot \mathbf{a} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = k \cdot \mathbf{a}(\mathbf{x}), \mathbf{x} \in \mathbf{X}\} \quad (3-14)$$

$$k + \mathbf{a} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = k + \mathbf{a}(\mathbf{x}), \mathbf{x} \in \mathbf{X}\} \quad (3-15)$$

此外, 实数域中的函数 $f : \mathbb{R} \rightarrow \mathbb{R}$, 均可以扩展到图像域上:

$$f(\mathbf{a}) = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = f(\mathbf{a}(\mathbf{x})), \mathbf{x} \in \mathbf{X}\} \quad (3-16)$$

例如:

$$\sin(\mathbf{a}) = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = \sin(\mathbf{a}(\mathbf{x})), \mathbf{x} \in \mathbf{X}\} \quad (3-17)$$

我们还可以定义位置加法运算, 即图像的位移运算。

设 $\mathbf{a} : \mathbf{X} \rightarrow \mathbb{R}, \mathbf{X} \subset \mathbb{Z}^2, \mathbf{y} \in \mathbb{Z}^2$

$$\mathbf{a} + \mathbf{y} = \{(\mathbf{z}, \mathbf{b}(\mathbf{z})) : \mathbf{b}(\mathbf{z}) = \mathbf{a}(\mathbf{z} - \mathbf{y}), \mathbf{z} - \mathbf{y} \in \mathbf{X}\} \quad (3-18)$$

对于图像还可以定义图像的特征函数 (characteristic functions), 这些函数的结果是一幅二值图像。

$$\chi_{\leq \mathbf{b}} \mathbf{a} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = 1 \text{ if } \mathbf{a}(\mathbf{x}) \leq \mathbf{b}(\mathbf{x}), \text{ otherwise } \mathbf{c}(\mathbf{x}) = 0\} \quad (3-19)$$

$$\chi_{< \mathbf{b}} \mathbf{a} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = 1 \text{ if } \mathbf{a}(\mathbf{x}) < \mathbf{b}(\mathbf{x}), \text{ otherwise } \mathbf{c}(\mathbf{x}) = 0\} \quad (3-20)$$

$$\chi_{= \mathbf{b}} \mathbf{a} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = 1 \text{ if } \mathbf{a}(\mathbf{x}) = \mathbf{b}(\mathbf{x}), \text{ otherwise } \mathbf{c}(\mathbf{x}) = 0\} \quad (3-21)$$

$$\chi_{\geq \mathbf{b}} \mathbf{a} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = 1 \text{ if } \mathbf{a}(\mathbf{x}) \geq \mathbf{b}(\mathbf{x}), \text{ otherwise } \mathbf{c}(\mathbf{x}) = 0\} \quad (3-22)$$

$$\chi_{>b} \mathbf{a} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = 1 \text{ if } \mathbf{a}(\mathbf{x}) > \mathbf{b}(\mathbf{x}), \text{ otherwise } \mathbf{c}(\mathbf{x}) = 0\} \quad (3-23)$$

$$\chi_{\neq b} \mathbf{a} = \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) : \mathbf{c}(\mathbf{x}) = 1 \text{ if } \mathbf{a}(\mathbf{x}) \neq \mathbf{b}(\mathbf{x}), \text{ otherwise } \mathbf{c}(\mathbf{x}) = 0\} \quad (3-24)$$

图像代数中还的定义了一些图像特有的函数

定义域限制函数:

设 $\mathbf{a} : \mathbf{X} \rightarrow \mathbb{R}, \mathbf{Z} \subset \mathbf{X}$, 则将 \mathbf{a} 限制在 \mathbf{X} 的子集 \mathbf{Z} 中的限制函数可以定义为:

$$\mathbf{a}|_{\mathbf{Z}} = \{(\mathbf{x}, \mathbf{a}(\mathbf{x})) : \mathbf{x} \in \mathbf{Z}\} \quad (3-25)$$

值域限制函数:

设 $\mathbf{a} : \mathbf{X} \rightarrow \mathbb{R}, S \subset \mathbb{R}$, 则将 \mathbf{a} 限制到 S 中的限制函数表示为

$$\mathbf{a}|_S = \{(\mathbf{x}, \mathbf{a}(\mathbf{x})) : \mathbf{a}(\mathbf{x}) \in S\} \quad (3-26)$$

图像值域限制概念在计算机视觉中应用非常广泛, 因为很多图像处理任务都限制在图像值满足一定属性的图像区域中进行。图像限制为表达许多图像处理算法提供了有用的工具, 比如图像阈值操作可以表示为: $\mathbf{a}|_{\geq k} \equiv \mathbf{a}|_{[k, \infty)}$, 即取图像值大于 k 的图像区域。将两种限制操作组合在一起可用如下式子表示:

$$\mathbf{a}|_{(\mathbf{Z}, S)} = \{(\mathbf{x}, \mathbf{a}(\mathbf{x})) : \mathbf{x} \in \mathbf{Z} \text{ and } \mathbf{a}(\mathbf{x}) \in S\} \quad (3-27)$$

图像扩展运算是另一种重要的运算, 设有两幅图像 $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$, $\mathbf{b} \in \mathbb{F}^{\mathbf{Y}}$, 将图像 \mathbf{a} 扩展到图像 \mathbf{b} 中记为 $\mathbf{a}|^{\mathbf{b}}$:

$$\mathbf{a}|^{\mathbf{b}}(\mathbf{x}) = \begin{cases} \mathbf{a}(\mathbf{x}) & \text{if } \mathbf{x} \in \mathbf{X} \\ \mathbf{b}(\mathbf{x}) & \text{if } \mathbf{x} \in \mathbf{Y} \setminus \mathbf{X} \end{cases} \quad (3-28)$$

在图像扩展的基础上还可以定义图像拼接运算。设图像 $\mathbf{a} \in \mathbb{F}^{\mathbf{Z}_1 \times \mathbf{Z}_2}$, $\mathbf{b} \in \mathbb{F}^{\mathbf{Z}_1 \times \mathbf{Z}_3}$, 则行拼接运算可以定义为:

$$(\mathbf{a} | \mathbf{b}) \equiv \mathbf{a}|^{\mathbf{b}+(0, k)} \quad (3-29)$$

其中 $(\mathbf{a} | \mathbf{b}) \in \mathbb{F}^{\mathbf{Z}_1 \times \mathbf{Z}_2 \cup \mathbf{Z}_3}$

多幅图像拼接可表示为 $(\mathbf{a}_1 | \mathbf{a}_2 | \dots | \mathbf{a}_l) = ((\mathbf{a}_1 | \mathbf{a}_2 | \dots | \mathbf{a}_{l-1}) | \mathbf{a}_l)$

列拼接可以使用类似的方法定义也可以简单地使用行拼接的转置来定义。

3.1.3 模板

模板是图像处理中最常用的工具之一, 大量的图像处理算法, 如空间滤波、边缘提取、数学形态学处理等, 都是利用模板对图像操作实现的。在图像代数中模板和模板操作同样是最有力的工具。

在图像代数中模板被定义为像素值为图像的特殊图像。设 \mathbf{X} 为原始图像的定义域, \mathbf{Y} 为结果图像的定义域, 则模板 \mathbf{t} 是定义域为 \mathbf{Y} 值域为 $\mathbb{F}^{\mathbf{X}}$ 的图像, 即 $\mathbf{t} : \mathbf{Y} \rightarrow \mathbb{F}^{\mathbf{X}}$ 或 $\mathbf{t} \in (\mathbb{F}^{\mathbf{X}})^{\mathbf{Y}}$ 。为了方便符号表示, 令 \mathbf{t} 在 \mathbf{y} 点的值 $\mathbf{t}_y \triangleq \mathbf{t}(\mathbf{y}), \forall \mathbf{y} \in \mathbf{Y}$ 。则图像 \mathbf{t}_y 可以表示为:

$$t_y = \{(x, t_y(x)) : x \in X\} \quad (3-30)$$

像素值 $t_y(x)$ 称为模板在点 y 处的权。

若 t 为实模板，则 t_y 的支撑定义为：

$$S(t_y) = \{x \in X : t_y(x) \neq 0\} \quad (3-31)$$

而 t_y 的无穷支撑定义为：

$$S_\infty(t_y) = \{x \in X : t_y(x) \neq -\infty\} \quad (3-32)$$

以及：

$$S_{-\infty}(t_y) = \{x \in X : t_y(x) \neq -\infty\} \quad (3-33)$$

在图像代数中模板可分为移动不变 (translation invariant) 和移动可变 (translation variant) 模板。称一个模板 $t \in (\mathbb{F}^X)^X$ 为移动不变，当且仅当对任意 $x, y, z \in X$ 满足 $t_y(x) = t_{y+z}(x+z)$ 。不满足该条件则称为移动可变。

在图像处理中使用的模板大都为移动不变模板，有限支撑的移动不变模板可用图形来表示。例如，令 $X \in \mathbb{Z}^2$ ， $y = (x, y)$ 为 X 内的任意一点，并设 $x_1 = (x, y-1)$ ， $x_2 = (x+1, y)$ ， $x_3 = (x+1, y-1)$ 。若模板 $t \in (\mathbb{R}^X)^X$ 在任意一点 y 处的权为， $t_y(y) = 1$ ， $t_y(x_1) = 3$ ， $t_y(x_2) = 2$ ， $t_y(x_3) = 4$ ， $t_y(x) = 0$ ，其中 x 为其他点。显然该模板的支撑为， $S(t_y) = \{y, x_1, x_2, x_3\}$ 。该模板可用如图 3-2 所示的图形表示，其中用多边形表示的单元指示了 y 的位置。

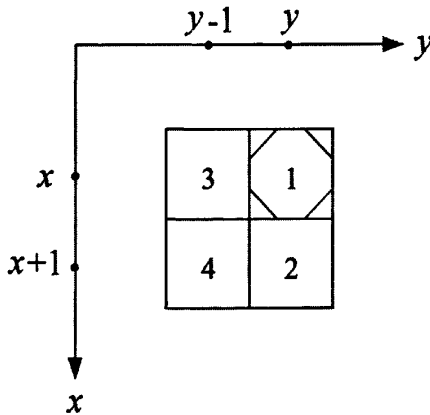


图 3-2 有限支持移不变模板的图像表示

Fig. 3-2 Pictorial representation of a translation invariant template

3.1.4 图像与模板间的操作

定义图像和模板间的操作可以将图像和模板结合起来实现图像处理算法。图像和模板间的操作包括了图像处理算法中的相关、卷积等，为表述图像空间滤波，数学形态学等操作提供了统一的数学工具。利用抽象代数可定义多种图像与模板

间的操作, 但图像代数中常用的操作主要有广义卷积、加性极大和乘性极大。

用模板 t 对图像 a 进行广义后向卷积定义为:

$$\mathbf{a} \oplus \mathbf{t} = \{(\mathbf{y}, \mathbf{b}(\mathbf{y})) : \mathbf{b}(\mathbf{y}) = \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) \times \mathbf{t}_y(\mathbf{x}), \mathbf{y} \in \mathbf{Y}\} \quad (3-34)$$

式中, \mathbf{b} 是结果图像, \mathbf{Y} 为其定义域。注意 \mathbf{t}_y 在其支撑之外为零, 所以用模板 t 对图像 a 进行广义后向卷积也可以定义为:

$$\mathbf{a} \oplus \mathbf{t} = \{(\mathbf{y}, \mathbf{b}(\mathbf{y})) : \mathbf{b}(\mathbf{y}) = \sum_{\mathbf{x} \in S(\mathbf{t}_y)} \mathbf{a}(\mathbf{x}) \times \mathbf{t}_y(\mathbf{x}), \mathbf{y} \in \mathbf{Y}\} \quad (3-35)$$

后向加性极大定义为:

$$\mathbf{a} \uparrow \mathbf{t} = \{(\mathbf{y}, \mathbf{b}(\mathbf{y})) : \mathbf{b}(\mathbf{y}) = \max_{\mathbf{x} \in S(\mathbf{t}_y)} [(\mathbf{a}(\mathbf{x}) + \mathbf{t}_y(\mathbf{x})) : \mathbf{x} \in \mathbf{X}], \mathbf{y} \in \mathbf{Y}\} \quad (3-36)$$

后向乘性极大定义为:

$$\mathbf{a} \uparrow \mathbf{t} = \{(\mathbf{y}, \mathbf{b}(\mathbf{y})) : \mathbf{b}(\mathbf{y}) = \max_{\mathbf{x} \in S(\mathbf{t}_y)} [(\mathbf{a}(\mathbf{x}) \times \mathbf{t}_y(\mathbf{x})) : \mathbf{x} \in \mathbf{X}], \mathbf{y} \in \mathbf{Y}\} \quad (3-37)$$

利用加性和乘性极大可以定义他们的对偶加性极小和乘性极小:

$$\mathbf{a} \downarrow \mathbf{t} = (\mathbf{t} \uparrow \mathbf{a}^*)^* \quad (3-38)$$

$$\mathbf{a} \downarrow \mathbf{t} = (\mathbf{t} \uparrow \mathbf{a}^*)^* \quad (3-39)$$

3.2 图像处理算法的图像代数表示

上一节介绍了图像代数的一些基本概念, 理论已经证明任何图像处理算法均可用图像代数来描述。本节将在上一节的基础上介绍一些典型图像处理算法的图像代数描述。

3.2.1 空间滤波及其图像代数表示

空间滤波是最常用的图像处理算法之一, 它是一种基于领域的图像处理算法, 利用被处理的像素及其领域中像素的灰度值确定输出。空间滤波通过一个模板和图像进行卷积运算来实现。在实际应用中常采用将模板在图像上逐点移动的方法, 对每一个点按照事先定义的关系计算输出。

一般地, 利用一个 $m \times n$ 的模板对一幅图像进行卷积操作可用如下公式表示:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t) \quad (3-40)$$

其中 $a = (m-1)/2$, $b = (n-1)/2$ 。

在图像代数中, 利用模板以及图像和模板的操作可以方便的表示卷积运算。如上节所述, 广义后向卷积定义为:

$$\mathbf{a} \oplus \mathbf{t} = \{(\mathbf{y}, \mathbf{b}(\mathbf{y})) : \mathbf{b}(\mathbf{y}) = \sum_{\mathbf{x} \in S(\mathbf{t}, \mathbf{y})} \mathbf{a}(\mathbf{x}) \times \mathbf{t}_{\mathbf{y}}(\mathbf{x}), \mathbf{y} \in \mathbf{Y}\} \quad (3-41)$$

因此，只要适当的定义模板，空间滤波在图像代数中可以表示为：

$$\mathbf{b} = \mathbf{a} \oplus \mathbf{s} \quad (3-42)$$

其中 $\mathbf{a} \in \mathbb{R}^X$ 为待处理图像， $\mathbf{s} \in (\mathbb{R}^X)^X$ 为使用的模板。

不同的空间滤波算法可通过选择不同的模板来实现，例如最常用的高斯滤波算法可定义如下模板。

$$s_{(x,y)}(v,w) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-v)^2 - (y-w)^2}{2\sigma^2}} \quad (3-43)$$

3.2.2 数学形态学及其图像代数表示

数学形态学是以形态为基础对图像进行分析的数学工具，它的基本思想是用具有一定形态的结构元素，去量度和提取图像中的对应形状，以达到对图像分析和识别的目的。数学形态学的应用可以简化图像数据，保持它们基本的形状特性，并除去不相干的结构。最初数学形态学方法只能处理二值图像，所以必须将灰度图像先进行二值化。20世纪70年代末80年代初人们开始研究灰度形态学，使得其可以应用于灰度图像和彩色图像，特别是后者近年来在图像理解和计算机视觉的研究中得到广泛重视。

数学形态学的基本运算包括：膨胀、腐蚀、开启和闭合。基于这些基本运算可推导出各种数学形态学的组合运算，进一步还可以构成各种进行图像处理和分析的实用算法。数学形态学的两个基本运算可以定义为：

膨胀运算：

$$\mathbf{b}(\mathbf{x}) = \max\{\mathbf{a}(\mathbf{x} + \mathbf{y}) : \mathbf{y} \in \mathbf{B}^*\} \quad (3-44)$$

其中 $\mathbf{B}^* = \{-\mathbf{b} : \mathbf{b} \in \mathbf{B}\}$ ，称为 \mathbf{B} 的共轭模板。

腐蚀运算：

$$\mathbf{b}(\mathbf{x}) = \min\{\mathbf{a}(\mathbf{x} + \mathbf{y}) : \mathbf{y} \in \mathbf{B}^*\} \quad (3-45)$$

开启和闭合时这两个基本运算的组合运算，其中开启操作为先对图像进行腐蚀操作然后进行膨胀操作，而闭合操作则刚好相反。

参考式 (3-36) 至 (3-39)，可以得到数学形态学运算的图像代数表示：

膨胀运算：

$$\mathbf{a} \uparrow \mathbf{t} \quad (3-46)$$

腐蚀运算：

$$\mathbf{a} \downarrow \mathbf{t} \quad (3-47)$$

开启运算：

$$(\mathbf{a} \downarrow \mathbf{t}') \uparrow \mathbf{t} \quad (3-48)$$

闭合运算:

$$(\mathbf{a} \uparrow \mathbf{t}) \downarrow \mathbf{t}' \quad (3-49)$$

3.2.3 边缘提取及其图像代数表示

边缘提取算法是一种重要的图像处理算法,应用十分广泛。它是许多高层次处理算法如图像分割、特征提取、场景分析等的基础。图像中物体的边缘通常是灰度值变化较大的位置,因此边缘提取算法就是利用这个特性,通过求取图像的梯度来提取物体的边缘。边缘通常包含两个属性即幅值和角度,分别表示边缘的强度和方向。目前常用的边缘提取算法大多是基于梯度算子与图像进行卷积运算的方法,典型的算子有 Laplace 算子、Prewitt 算子、Sobel 算子等。

下面以 Sobel 算子为例说明边缘提取算法的图像代数表示,其它算子的表示与此类似。

设 $\mathbf{a} \in \mathbb{R}^x$ 为待处理图像, a_0, a_1, \dots, a_7 为像素 (i, j) 的 8-领域内的像素的灰度值,这些像素按逆时针排列,如图 3-3 所示。

a_3	a_2	a_1
a_4	(i, j)	a_0
a_5	a_6	a_7

图 3-3 8-领域像素排列

Fig. 3-3 Pixels arrangement of a 8-neighborhood

则 sobel 边缘的幅值图像 $\mathbf{m} \in \mathbb{R}^x$ 定义为:

$$\mathbf{m}(i, j) = (u^2 + v^2)^{1/2} \quad (3-50)$$

其中

$$u = (a_5 + 2a_6 + a_7) - (a_1 + 2a_2 + a_3) \quad (3-51)$$

$$v = (2a_0 + a_1 + a_7) - (a_3 + 2a_4 + a_5) \quad (3-52)$$

梯度方向图像 \mathbf{d} 可定义为:

$$\mathbf{d}(i, j) = \arctan\left(\frac{u}{v}\right) \quad (3-53)$$

利用图像代数,则式 (3-50) 和 (3-53) 可表示为:

$$\mathbf{m} := \left| (\mathbf{a} \oplus \mathbf{s})^2 + (\mathbf{a} \oplus \mathbf{t})^2 \right|^{1/2} \quad (3-54)$$

$$\mathbf{d} := \arctan2\left(\left. (\mathbf{a} \oplus \mathbf{s}) \right|_{\mathbf{m}>0}, \left. (\mathbf{a} \oplus \mathbf{t}) \right|_{\mathbf{m}>0}\right) \quad (3-55)$$

其中模板 \mathbf{s} 和 \mathbf{t} 分别定义为:

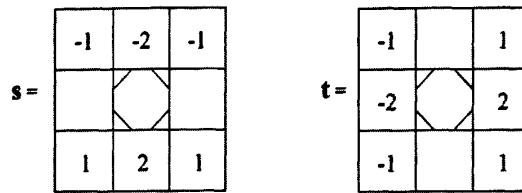


图 3-4 Sobel 边缘提取模板

Fig. 3-4 Templates of Sobel edge detection

除了和图像进行卷积的方法，另一种常用的边缘提取算法是形态学边缘检测，它首先利用适当的模板对图像进行膨胀操作，然后从结果上再减去原图像，从而得到图像边缘。因此其图像代数表示为：

$$a \uparrow t - a \tag{3-56}$$

3.2.4 图像几何变换及其图像代数表示

图像几何变换是图像分析以及计算机视觉中非常常用的方法。它是将图像中像素的几何关系进行变换，从而改变图像的几何形状，以便进行分析。比如在本系统中，我们就对护栏图像的区域进行了几何变换，从而得到了矩形的护栏图像。关于几何变换的详细介绍可以参照 4.1 节的讨论。本节我们着重讨论其图像代数表示，几何变换实质上是对像素坐标的变换，即将图像中一点的坐标按一定的关系变换为另一个坐标。在图像代数中的变换关系 f 可以表示为点集之间的映射：

$$f: Y \rightarrow X \tag{3-57}$$

设变换前的图像为 $a \in F^X$ ，变换后的图像 $a \circ f \in F^Y$ ，则几何变换在图像代数中可以用如下方程表示：

$$a \circ f = \{(y, a(f(y))) : y \in Y\} \tag{3-58}$$

实际上式 (3-58) 使用的是反变换方程，这里的 $f(y)$ 可以是任何的线性或非线性变换。若此处为线性变换，且定义域为二维空间时， $f(y)$ 即为式 (4-9)。

3.3 图像代数基本运算的硬件实现

本章提出的基于图像代数的图像处理算法硬件实现技术，就是将图像代数中的基本元素和基本运算利用硬件实现，形成基础硬件库。在实现图像处理算法时，就可以根据该算法的图像代数表示，调用相应的基础硬件即可构成该算法的硬件处理系统。上面我们已经就图像代数的基本理论和一些典型图像处理算法的图像代数表示方法进行了讨论，这一节我们将讨论常用的图像代数基本元素和运算的硬件实现。

3.3.1 模板及模板运算的硬件实现

模板及模板运算是图像代数中最有力的工具。大量的图像处理算法都是以模板和图像的运算为基础的。在图像代数中模板被定义为像素值为图像的特殊图像。对于一个模板 $t \in (\mathbb{F}^X)^Y$ ，其在 y 点的值可以表示为：

$$t_y = \{(x, t_y(x)) : x \in X\} \quad (3-59)$$

实际中，通常取 $X = Y = Z_m^+ \times Z_n^+$ ，则模板 t 是一幅 $m \times n$ 的图像，该图像的每一个像素 t_y 也是一幅大小为 $m \times n$ 的图像，这样在进行模板的硬件实现时需要对每个像素设计其相应的硬件模块，不但消耗大量的硬件资源也需要花费大量的时间。但是实际上模板和图像进行运算时只有模板的支撑区内的数据是有用的，这样只需对模板的支撑区进行硬件实现即可，通常一个模板支撑区相对于图像大小要小得多。并且，如前所述对于一个有限支撑移不变模板 $t \in (\mathbb{F}^X)^X$ 对任意 $x, y, z \in X$ 满足 $t_y(x) = t_{y+z}(x+z)$ ，并可用图形来表示。这表明有限支撑移不变模板对于每个 y 其相对结构是相同的，因此只需要利用硬件实现一个 t_y 。因此，对于有限支撑移不变模板，其硬件实现所需资源以及需花费的时间将比图像代数中一般概念的模板少得多。而实际算法使用的模板通常都是有限支撑移不变模板，所以本章我们只针对有限支撑移不变模板的硬件实现进行讨论。如没有特殊说明本文以下所称的模板都是指有限支撑移不变模板。

图像和模板的卷积运算的表达式为：

$$a \oplus t = \{(y, b(y)) : b(y) = \sum_{x \in X} a(x) \times t_y(x), y \in Y\} \quad (3-60)$$

式中的加法和乘法可以被替换为其他的运算，如将式 (3-60) 中的加法替换为取最大值，就得到了乘性极大运算即 (3-37)；若继续将乘法替换为加法就成为了加性极大，即式 (3-36)。

从上面的分析中可以看出图像和模板的运算分两步进行。对于结果图像中的一点 y ，首先将输入图像 x 每个像素的值和相应模板中的对应权值进行一次运算，再将所有一次运算的结果进行二次运算，从而获得了结果图像中该像素的值。选择不同的一次运算和二次运算就可以得到各种不同的图像和模板运算，例如选择乘法为一次运算，加法为二次运算就得到了前面介绍的卷积运算，而选择加法为一次运算，取最大值为二次运算就得到了后向加性极大运算。实际上模板也是一种特殊的图像，因此所有定义在图像上的运算在这里都可以被选择为一次运算和二次运算，这样就可以构建出大量的图像和模板运算。

可以看出模板以及图像与模板运算紧密联系的，所以我们在进行硬件实现时将这两部分作为一个整体进行相应的硬件电路设计。其硬件设计示意如图 3-5 所示，其主要包括三部分，支撑区图像缓冲寄存器、权值寄存器以及可编程运算器。

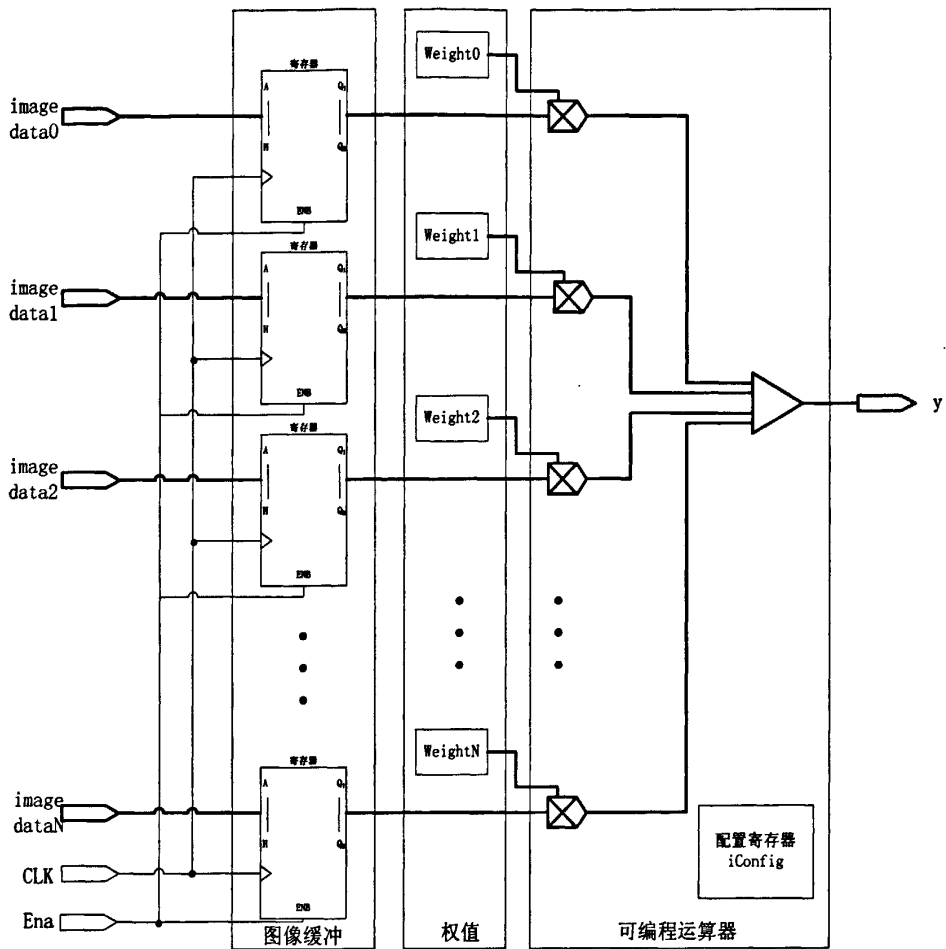


图 3-5 模板及模板运算硬件实现示意图

Fig. 3-5 Hardware implement for template and its operation

图像缓冲寄存器保存当前处理的模板支撑区中的图像数据。输入端口 $x_1 \sim x_N$ 为模板支撑区的图像数据来自图像位移寄存器。端口的具体个数由模板支撑区的大小决定，如使用矩形 3×3 支撑区模板时将有 9 个端口。Ena 为使能信号，Clk 为时钟信号。在 Clk 上升沿时，若 Ena 有效时（高电平），则寄存器装入图像数据；否则寄存器保存原数据。在模板中加入图像缓冲后可以方便地实现流水线处理，当图像位移寄存器更新数据时，模板中的图像数据并不改变。权值寄存器保存模板支撑区中各个点的权值，可以是固定的也可以通过外界进行配置，比如由 PC 通过 PCI 总线写入权值寄存器等。

可编程运算器用来实现式 (3-60) 中的计算，它由两级运算组成，我们将它们由输入向输出方向分别称为一级和二级运算。为了使模板具有更大的通用性，我们这里采用了可编程形式的运算器，即一级和二级运算分别进行什么运算可以通过配置寄存器来进行配置。在设计中我们采用了多个二级运算单元，缓冲区中每

一个像素都对应一个相应的二级运算单元，这样充分利用了并行处理的优点，但是将消耗大量的硬件资源。若在硬件资源紧张而对速度要求略低时，可采用多个像素复用一個运算单元的方法。

根据上述思想，我们利用了 VHDL 实现了支撑区为矩形 3×3 大小的模板的设计。并利用 Xilinx ISE 和 ModelSim 进行了仿真，最后在 Xilinx Veritx II Pro 中实现。

在利用 VHDL 设计时，我们将图像输入端口 (ivImageData) 以及权值输入端口 (weight) 定义为端口数组，可以使用索引值对端口进行选取。这样可以当需要更改模板大小时，方便地更改端口的数目，使代码具有更高的通用性。权值可以由 PC 通过 PBI 总线进行配置，和 PBI 总线数据宽度一样，每一个权值寄存器的宽度是 16 位。

同样可编程运算器的运算模式可以一通过 PBI 总线进行配置，PC 通过写入 iConfig 配置寄存器来控制运算的模式，iConfig 同样为 16 位，其中的低 8 为控制一级运算器的模式，高 8 位控制二级运算器的模式，其各位的具体定义见表 3-1。我们只要合理地配置该寄存器，就可以实现具体的图像处理功能，如当进行空间滤波时设置 iConfig 为 0x0102，灰度膨胀时为 0x0201，灰度腐蚀时为 0x0401。

表 3-1 iConfig 配置寄存器定义

Table 3-1 definition of iConfig register

位	值	功能
iConfig[1,0]	01	一级运算器进行加法运算
	10	一级运算器进行乘法运算
	其它	一级运算器结果始终为 0
iConfig[7:2]	-	保留
iConfig[10:8]	001	二级运算器进行加法运算
	010	二级运算器进行取最大值运算
	100	二级运算器进行取最小值运算
	其它	二级运算器结果始终为 0
iConfig[15:11]	-	保留

模板运算模块在 ModelSim 下的仿真波形如图 3-6 所示。iConfig 分别进行了空间滤波、灰度形态学膨胀以及灰度形态学腐蚀三种运算，得到了正确的结果。

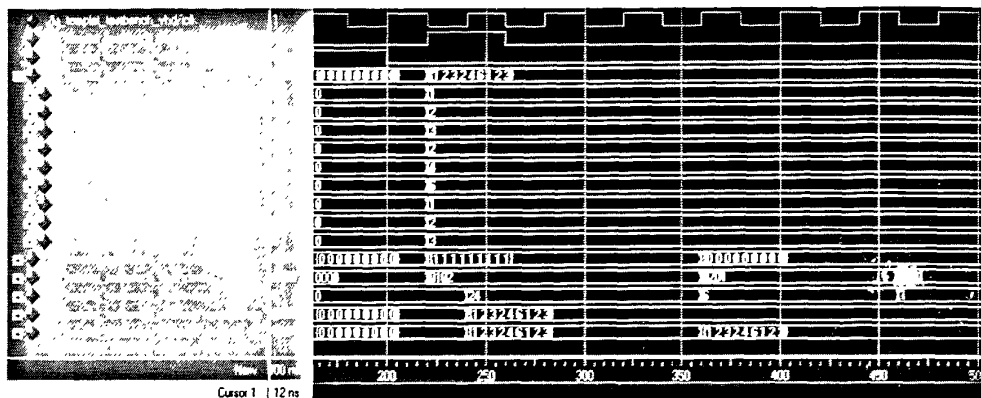


图 3-6 模板运算仿真波形

Fig. 3-6 Behavior simulation waveform of template module

3.3.2 图像及其运算的硬件实现

在图像代数中，图像包含了两个部分，即定义域和像素值。在进行硬件实现时，我们也需要对这两个部分进行建模。图像的定义域是一个点集，在前面我们提到过点集是满足一定拓扑关系的点的集合，因此我们在进行硬件实现时除了要表达点外，还要表达这种拓扑关系。对于一般的图像，像素值是属于一个数域 \mathbb{F} 的，对于通常处理的灰度图像就是正整数域 \mathbb{Z}^+ ，整数就可以方便地在数字电路中实现。

因此，图像可以使用一组寄存器来实现。每一个寄存器保存一个点的像素值，并使寄存器之间满足一定的拓扑关系。实际上，这种拓扑关系并不一定要求寄存器之间在空间上严格地满足这种关系，而只要求其在逻辑上具有相应的关系就可以了。这样图像硬件实现的示意图可以由图 3-7 表示。图像数据通常来自图像缓冲区，由于图像通常是串行输入的，因此图像硬件采用移位的方法将图像数据逐个移入其对应的位置，如图 3-7 所示，当 Shift 信号有效时（高电平），在时钟（CLK）的上升沿图像数据将进行移位操作。

然而通常图像的定义域都较大，即使是一般电视标准清晰度的图像，即分辨率为 720×576 ，也有 414720 个像素。而目前的工业相机分辨率就更高。如果按上述方式实现整个定义域中的图像，将消耗大量的硬件资源。因此我们只能实现图像的一部分，即实现定义域中的一个子集，来降低对硬件资源的需求。

这样的思路实际上是对图像进行了定义域限制运算，得到的是原图像的子图像，若设被实现的图像为 $\mathbf{a} \in (\mathbb{Z}^+)^{\mathbf{X}}$ ，实际实现的图像为 $\mathbf{b} \in (\mathbb{Z}^+)^{\mathbf{Z}}$ ，其中 $\mathbf{Z} \subset \mathbf{X}$ 。则更具式 (3-25) 实现的图像可用下式表示：

$$\mathbf{b} = \mathbf{a}|_{\mathbf{Z}} \quad (3-61)$$

而对于整幅图像进行的操作时，我们可以通过移位操作来改变图像数据的位置。

置,这样就相当于移动了定义域记为 $Z_n, n=1,2,3\dots$, 则此时实际实现的图像为:

$$b_n = a|_{Z_n} \quad (3-62)$$

若不断移动子图像的定义域以覆盖整个图像的定义域,即满足:

$$X \subset Z_1 \cup Z_2 \cup \dots \cup Z_n \quad (3-63)$$

则整幅图像都就被实现了。这样实际上是采用了分时表示的方法,一幅图像被分割成若千的部分,顺序地依次进行表示。在图像运算和处理时就可以依次对这些子图像进行处理,最后将分别得到的结果在拼接成完整的结果就可以了。而实现时具体分块的大小,即子图像定义域的大小是可以根据实际情况来进行选择,当子图像选用的定义域较大时,占用的硬件资源就多,但是其处理速度将较快,反之亦然。

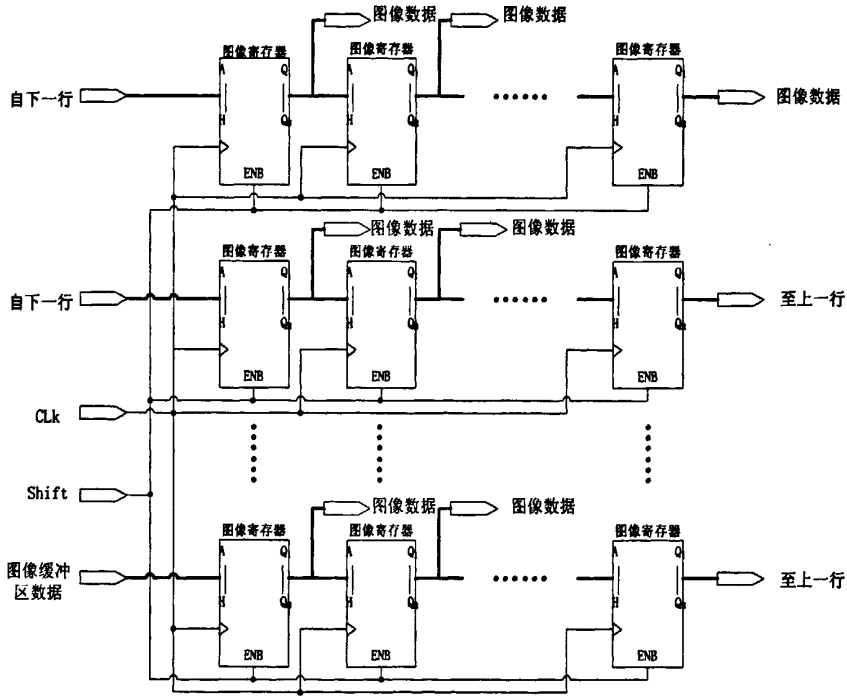


图 3-7 图像硬件实现示意图

Fig. 3-7 Hardware implement schematic of image

但是当图 3-7 中的寄存器不能覆盖到所有的图像时,未被寄存的图像数据需要在存储器中保存。这些数据主要包括从下一行移出并还未能进入上一行寄存器的数据,因此在实际的图像硬件实现时,我们加入了行缓存存储器来存储这部分数据,加入行缓冲后硬件实现示意图如图 3-8 所示。行缓冲的长度正好的一行图像的长度,需要的行缓冲存储器的个数由实现的子图像的定义域的大小有关,当定义域为 3×3 时需要缓存两行的数据,而 5×5 时就需要缓冲 4 行的数据。

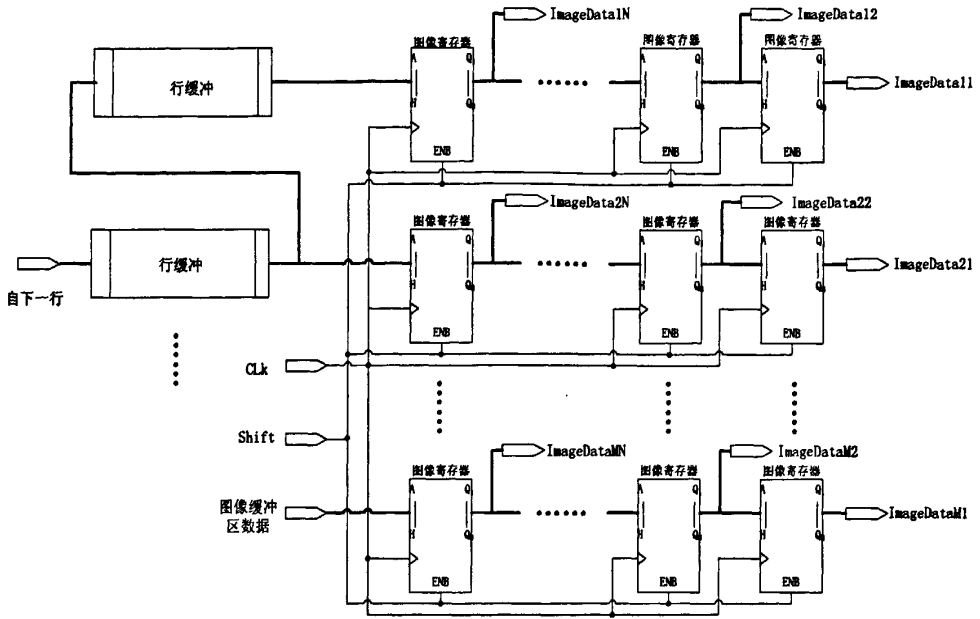


图 3-8 带行缓冲的图像硬件实现示意图

Fig. 3-8 Hardware implement schematic with line buffer of image

行缓冲存储器采用了环形存储结构，即当新的数据写入时，最久的数据被读出^[24,25]。这种结构和常用的 FIFO 比较类似。所不同的是，FIFO 的写入指针和读出指针之间的距离是和 FIFO 中所存储的数据个数有关。而行缓冲存储器中，读出指针和写入指针之间的距离是固定的，可以认为是一个一直存满数的 FIFO。我们利用双口 RAM 实现了这样的环形结构存储器，其示意图如图 3-9 所示。

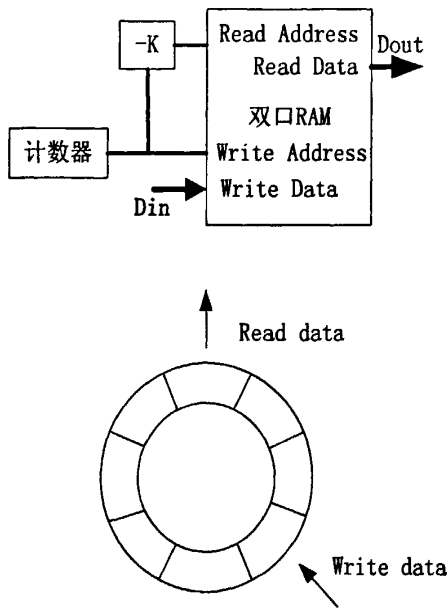


图 3-9 行缓冲存储器结构示意图

Fig. 3-9 Ring structure of line buffer memory

如图 3-9 所示，写地址由一个计数器产生，每当行缓冲中有写入一个数据后，计数器加一。而读地址始终和写地址相差 K 个地址的距离，这个距离就是图像一行的长度，即一行图像所包含的图像数据个数。这样行缓冲的输出始终比输入滞后了 K 个数据，其仿真的波形图如图 3-10 所示，仿真时 $K=8$ 。

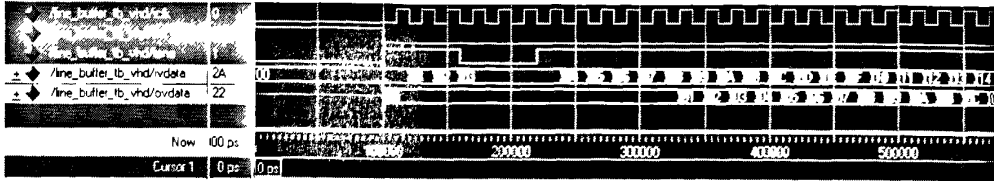


图 3-10 行缓冲仿真波形图

Fig. 3-10 Simulation waveform of line buffer

我们利用 VHDL 实现了定义域为 3×3 的，包含两个行缓冲的子图像硬件。其在 Modelsim 环境下的仿真果然如图 3-11 所示，我们使用了一幅完整大小为 8×8 图像对其进行仿真，可以看到 linebuffer1 和 linebuffer2 分别比图像数据滞后了 2 行和 1 行。

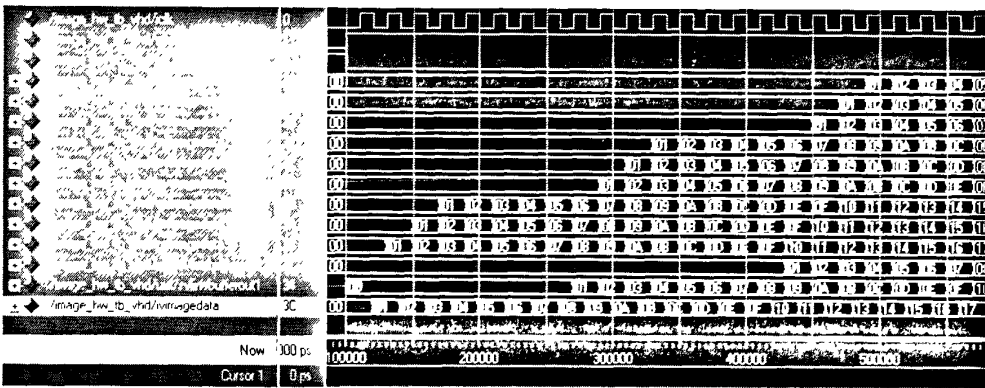


图 3-11 行缓冲和图像寄存器仿真波形图

Fig. 3-11 Simulation waveform of line buffer and image register

3.3.3 图像和模板间操作的硬件实现

前面我们介绍了图像以及模板的硬件实现，本节我们将介绍图像和模板间操作的硬件实现。图像和模板间的操作是图像代数中最有力的工具，大部分常用的图像处理都是利用图像和模板的操作来实现的。前面我们已经利用硬件实现了图像和模板，和前面的问题一样这里需要考虑的还是硬件资源利用的问题。在图像代数中模板是一幅特殊的图像，结果图像的定义域中的每一个像素都有对应着一个模板，若对整幅图像进行实现会消耗大量的硬件资源。并且，前面实现图像是我们只实现了图像的一个子图像。

因此在进行图像和模板的操作时我们也只能对该子图像进行操作，得到结果图像的子图像，然后在将各这些子图像拼接产生结果完整的图像。这在理论上是可以证明的：

设待处理的图像为 $\mathbf{a} \in \mathbb{R}^X$ ，使用的模板为 $\mathbf{t} \in (\mathbb{R}^X)^X$ ，结果图像为 $\mathbf{b} \in \mathbb{R}^X$ 则图像和模板的操作可以表示为：

$$\mathbf{b} = \mathbf{a} \oplus \mathbf{t} = \{(\mathbf{y}, \mathbf{b}(\mathbf{y})) : \mathbf{b}(\mathbf{y}) = \sum_{\mathbf{x} \in X} \mathbf{a}(\mathbf{x}) \cdot \mathbf{t}_{\mathbf{y}}(\mathbf{x})\} \quad (3-64)$$

若将图像分为 n 块子图像，可表示为：

$$\mathbf{a} = (\mathbf{a}_1 | \mathbf{a}_2 | \cdots | \mathbf{a}_n)' = \begin{cases} \mathbf{a}_1(\mathbf{x}), & \mathbf{x} \in X_1 \\ \mathbf{a}_2(\mathbf{x}), & \mathbf{x} \in X_2 \\ \vdots \\ \mathbf{a}_n(\mathbf{x}), & \mathbf{x} \in X_n \end{cases} \quad (3-65)$$

根据式 (3-64) 和 (3-65) 可得：

$$\mathbf{b} = \mathbf{a} \oplus \mathbf{t} = \begin{cases} \sum_{\mathbf{x} \in X_1} \mathbf{a}_1(\mathbf{x}) \mathbf{t}_{\mathbf{y}}(\mathbf{x}), & \mathbf{x} \in X_1 \\ \sum_{\mathbf{x} \in X_2} \mathbf{a}_2(\mathbf{x}) \mathbf{t}_{\mathbf{y}}(\mathbf{x}), & \mathbf{x} \in X_2 \\ \vdots \\ \sum_{\mathbf{x} \in X_n} \mathbf{a}_n(\mathbf{x}) \mathbf{t}_{\mathbf{y}}(\mathbf{x}), & \mathbf{x} \in X_n \end{cases} \quad (3-66)$$

而利用模板对式(3-65)中的子图像分别进行操作可得到结果图像的子图像为：

$$\begin{aligned} \mathbf{b}_1 &= \mathbf{a}_1 \oplus \mathbf{t} = \{(\mathbf{y}, \mathbf{b}_1(\mathbf{y})) : \mathbf{b}_1(\mathbf{y}) = \sum_{\mathbf{x} \in X_1} \mathbf{a}_1(\mathbf{x}) \cdot \mathbf{t}_{\mathbf{y}}(\mathbf{x})\} \\ \mathbf{b}_2 &= \mathbf{a}_2 \oplus \mathbf{t} = \{(\mathbf{y}, \mathbf{b}_2(\mathbf{y})) : \mathbf{b}_2(\mathbf{y}) = \sum_{\mathbf{x} \in X_2} \mathbf{a}_2(\mathbf{x}) \cdot \mathbf{t}_{\mathbf{y}}(\mathbf{x})\} \\ &\vdots \\ \mathbf{b}_n &= \mathbf{a}_n \oplus \mathbf{t} = \{(\mathbf{y}, \mathbf{b}_n(\mathbf{y})) : \mathbf{b}_n(\mathbf{y}) = \sum_{\mathbf{x} \in X_n} \mathbf{a}_n(\mathbf{x}) \cdot \mathbf{t}_{\mathbf{y}}(\mathbf{x})\} \end{aligned} \quad (3-67)$$

则将这些子图像拼接为完整的结果图像可得：

$$\mathbf{b} = (\mathbf{b}_1 | \mathbf{b}_2 | \cdots | \mathbf{b}_n)' = \begin{cases} \sum_{\mathbf{x} \in X_1} \mathbf{a}_1(\mathbf{x}) \mathbf{t}_{\mathbf{y}}(\mathbf{x}), & \mathbf{x} \in X_1 \\ \sum_{\mathbf{x} \in X_2} \mathbf{a}_2(\mathbf{x}) \mathbf{t}_{\mathbf{y}}(\mathbf{x}), & \mathbf{x} \in X_2 \\ \vdots \\ \sum_{\mathbf{x} \in X_n} \mathbf{a}_n(\mathbf{x}) \mathbf{t}_{\mathbf{y}}(\mathbf{x}), & \mathbf{x} \in X_n \end{cases} \quad (3-68)$$

比较式 (3-66) 和 (3-68) 可以看到拼接后得到的结果图像和原始处理的结果图像是一致的。

这里应该指出，子图像的定义域 Z 和模板的支撑区 $S(\mathbf{t}_{\mathbf{y}})$ 必须满足如下的关系，即设实现的 n 个模板分别为 $\mathbf{t}_{\mathbf{y}_1}, \mathbf{t}_{\mathbf{y}_2}, \cdots, \mathbf{t}_{\mathbf{y}_n}$ ，则

$$S(t_{y_1})US(t_{y_2})U\cdots US(t_{y_n}) \subset Z \quad (3-69)$$

即子图像的定义域必须能够包含所有的模板的支撑区。在满足上面的条件下，图像和模板间操作的硬件实现就只要将图像硬件的输出和模板硬件的相应输入连接起来就可以了。我们利用前面已经实现的定义域为的 3×3 的图像硬件，和支撑区同样为 3×3 的模板硬件，构建了图像和模板间操作的硬件，其硬件实现示意图如图 3-12 所示。为了方便画图，图 3-3 将图像寄存器（IR）排列成了一行。

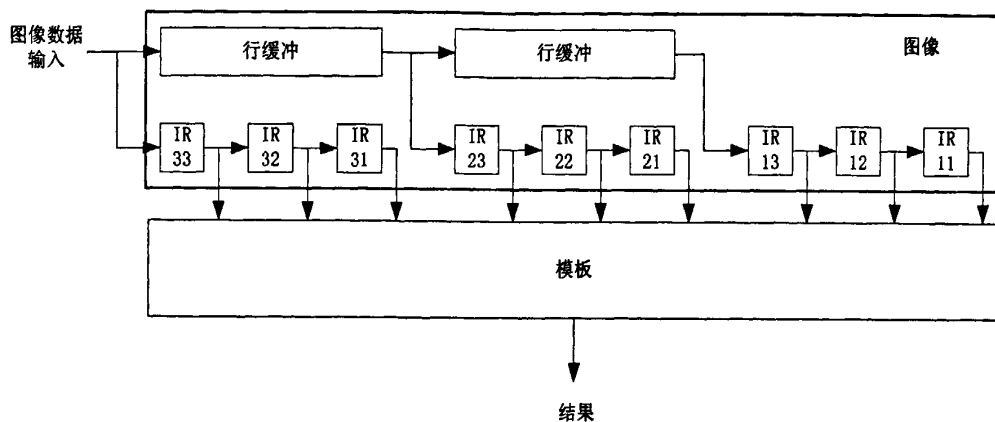


图 3-12 图像和模板间操作硬件实现示意图

Fig. 3-12 Schematic of hardware implement of image-template product

图 3-13 给出了该硬件在 Modelsim 中仿真的波形如所示。仿真时模板被设置为卷积模式，需要指出的是由于模板内还有一级图像缓存，因此输出结果滞后了一个时钟。

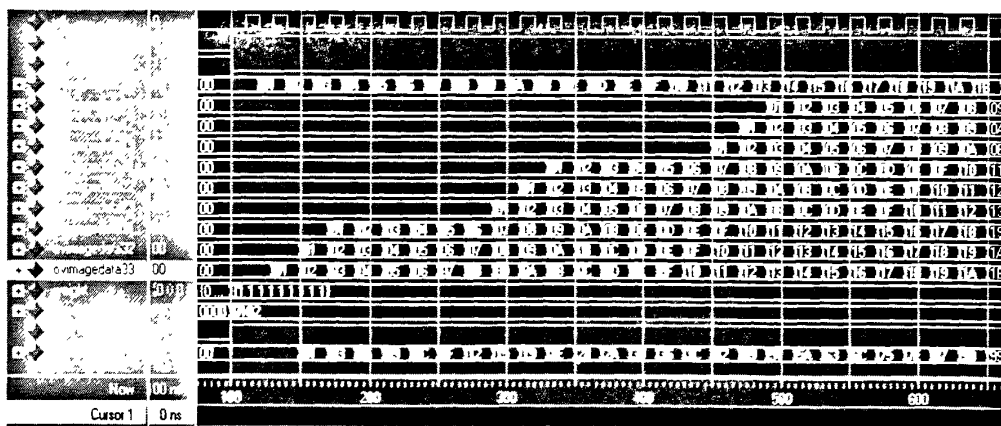


图 3-13 图像和模板间操作硬件仿真波形图（卷积模式）

Fig. 3-13 Simulation waveform of image-template product hardware (convolution mode)

3.4 典型图像处理算法的硬件实现

上一节我们已经用硬件实现了图像代数中的图像、模板以及图像和模板操作

等三个基本元素和运算。初步建立了基于图像代数的基础硬件库。本节将以此为基础来构建一些典型图像处理算法的硬件实现。

3.4.1 空间滤波的硬件实现

在 3.2 节中，我们已经讨论了空间滤波算法的在图像代数中的表示，其可以表示为图像和模板的卷积：

$$\mathbf{b} = \mathbf{a} \oplus \mathbf{s} \quad (3-70)$$

因此空间滤波算法可以利用 3.3.3 节介绍的图像和模板间操作的硬件来实现，我们只要按表 3-1 将模板运算配置寄存器设为卷积模式即可。我们利用 Anaconda 板卡上的 Xilinx Virtex-II Pro 系列 FPGA 实现了模板支持区为 3×3 矩形的模板，子图像的支撑区同样为 3×3 的矩形区域。

我们利用实现的硬件对采集的图像进行了实时处理。处理分两步进行，首先采集的图像先被存储到 DDR SDRAM 中，与此同时保存在另一块 DDR SDRAM 中的上一幅图像的处理结果，被输出到 PC 上进行显示；然后图像处理单元从 DDR SDRAM 中读出图像数据，进行处理并将结果存储到另一块 DDR SDRAM 中。关于图像采集和 DDR SDRAM 读取的电路设计在第二章中已经进行了讨论，这里不再赘述。

空间滤波硬件实现仿真波形如图 3-13 所示。图 3-14 为利用 3 阶高斯滤波算子处理的结果，其中 (a) 为原始图像，(b) 为处理后的图像。可以看到结果图像出现了模糊，图像噪声减小。由于 3 阶高斯算子元素的和是 16，因此只要将卷积的输出结果左移 4 位就可以实现归一化。

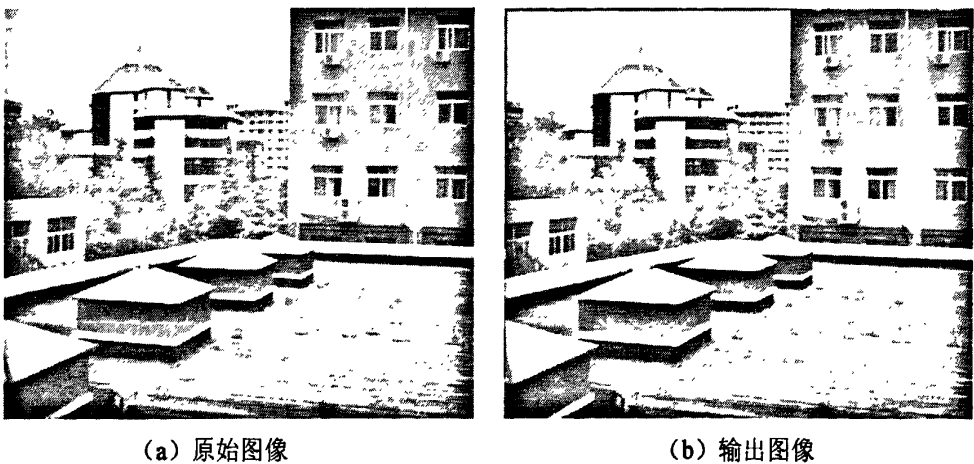


图 3-14 高斯滤波处理结果

Fig. 3-14 Result of Gaussian filter

3.4.2 灰度形态学的硬件实现

灰度形态学的基本运算有两种即腐蚀和膨胀，前面已经讨论这两种运算的图像代数表示，它们可以分别表示为：

$$a \downarrow t, a \uparrow t \quad (3-71)$$

因此其与图像的空间滤波类似，也可以使用图像和模板间操作的硬件来实现，所不同的是，我们要将模板的运算配置寄存器设置分别设置为腐蚀模式和膨胀模式。膨胀运算硬件实现的仿真波形图如图 3-15 所示。

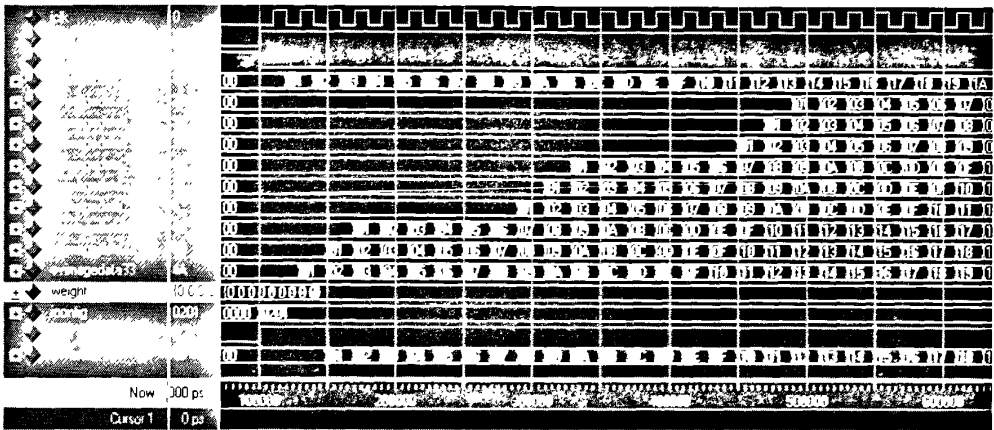


图 3-15 膨胀运算仿真波形图

Fig. 3-15 Simulation waveform of dilation operating hardware

灰度形态学处理一般使用权值全部是 0 的模板，所以在进行膨胀运算时模板实现的其实是求 9 个输入中的最大值。而腐蚀运算实际上是求输入的最小值，其硬件实现的仿真波形如图 3-16 所示。

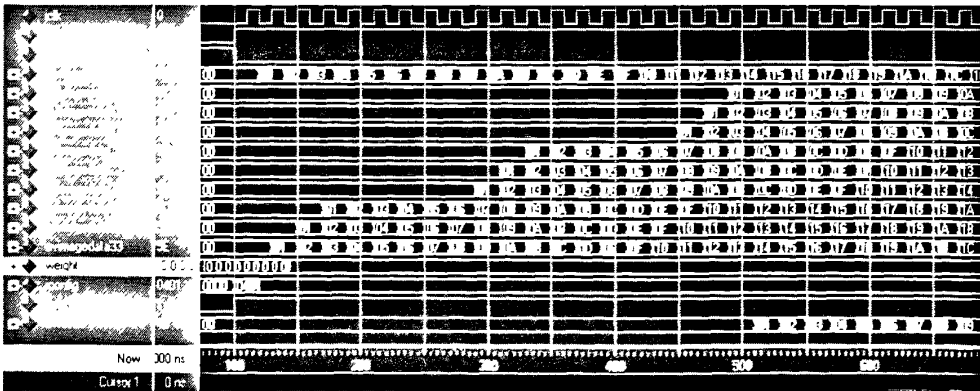


图 3-16 腐蚀运算仿真波形图

Fig. 3-16 Simulation waveform of erosion operating hardware

图 3-17 给出了膨胀和腐蚀的处理结果，其中 (a) 为原始图像，(b) 和 (c) 分别为膨胀和腐蚀的结果。

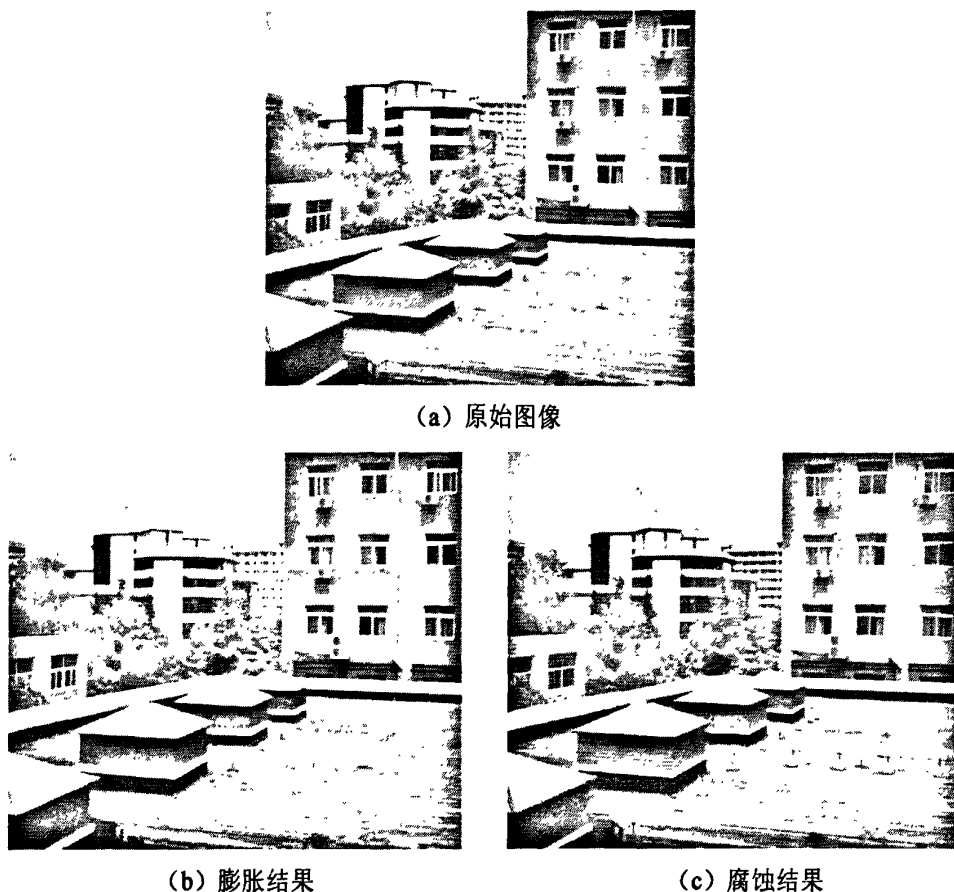


图 3-17 灰度形态学处理结果

Fig. 3-17 Result of gray-scale morphology

对于形态学的开启和闭合运算实际上是膨胀和腐蚀的组合运算，在基于图像代数的硬件实现中，我们可以方便地利用膨胀和腐蚀的硬件来实现。我们只要在两级模板之间在插入一个中间结果图像，将前一级的结果输入到下一级的图像输入端口即可实现组合操作了。这种将两极处理连接起来的方法同样适用于其他处理，比如连续进行两次图像空间滤波处理，或进行滤波后再进行形态学处理等。

3.4.3 边缘提取的硬件实现

前面我们已经介绍了多种图像边缘提取算法及其图像代数表示，其中利用 Laplace 算子进行边缘提取的方法和图像空间滤波是非常相似的，其都是利用模板和图像进行卷积来实现的，可以用式 (3-70) 来表示。因此我们可以使用和 3.4.1 节使用的硬件来实现 Laplace 算子边缘提取，只要改变模板的权值就可以了。有所不同的是，在进行边缘提取时，卷积的结果会出现负数，因此需要在模板的输出结果后加入绝对值运算电路。

在 VHDL 实现时，我们利用了 IEEE 库中 STD_LOGIC_SIGNED 包集来实现有符号数的操作^[26]。在 VHDL 描述文件中声明 STD_LOGIC_SIGNED 包集后就可以对 STD_LOGIC_VECTOR 类型的值按有符号数进行数学运算^[27]，数值采用补码形式表示，因此只要对负数进行“取反加一”就可以得到其绝对值。加入绝对值运算电路后，Laplace 算子边缘提取的仿真结果如图 3-18 所示。其处理结果如图 3-19 所示。

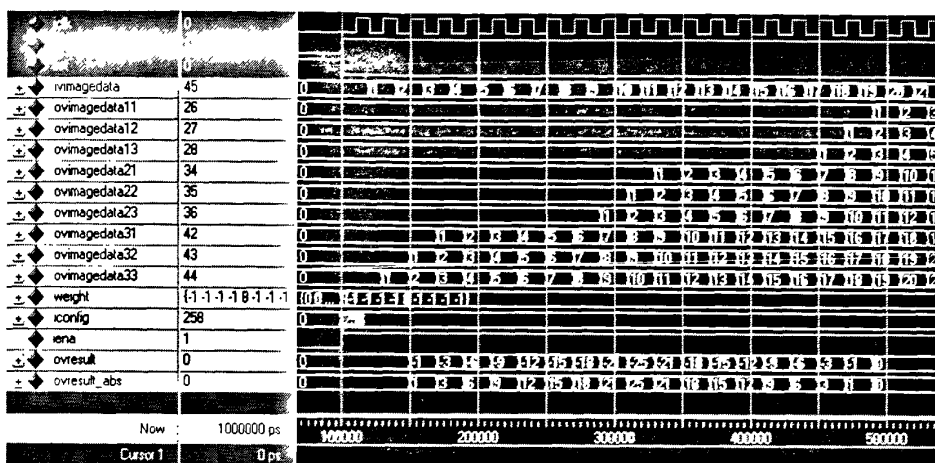
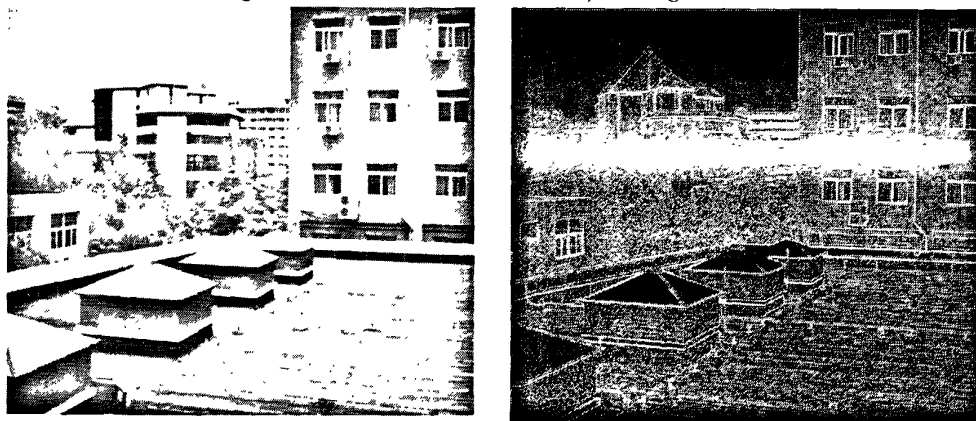


图 3-18 Laplace 算子边缘提取仿真波形图

Fig. 3-18 Simulation waveform of Laplace edge detector



(a) 原始图像

(b) 处理结果

图 3-19 Laplace 算子边缘提取处理结果

Fig. 3-19 Result of Laplace edge detector

Sobel 算子是另一种常用的边缘提取算法，在图像代数中其可以用式 (3-54) 和 (3-55) 来表示。在硬件实现时，由于受到 FPGA 数学运算能力的限制，我们将计算边缘幅值得式 (3-54) 改为：

$$m := |a \oplus s| + |a \oplus t| \quad (3-72)$$

这样在硬件实现时就可以利用将图像同时输入到两个模板，然后再对两个模

板的输出求和的方法来实现 Sobel 边缘提取。其仿真波形和处理结果分别如图 3-20 和图 3-21 所示。

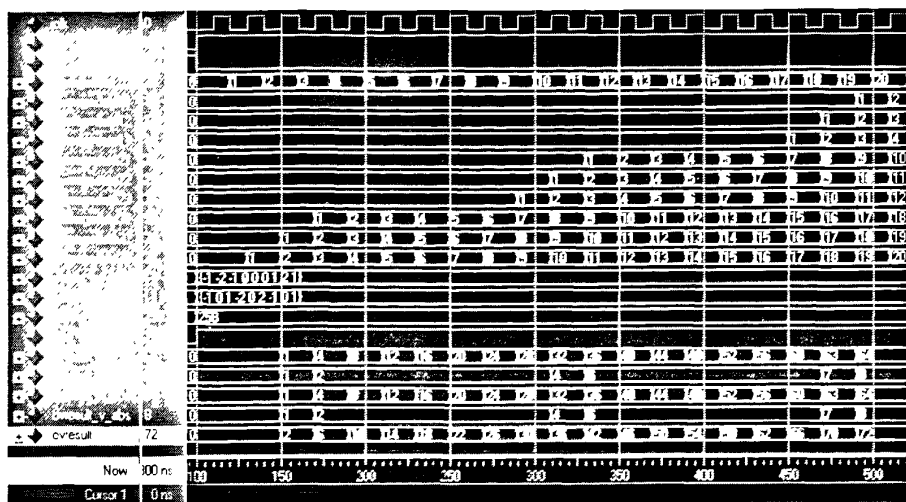
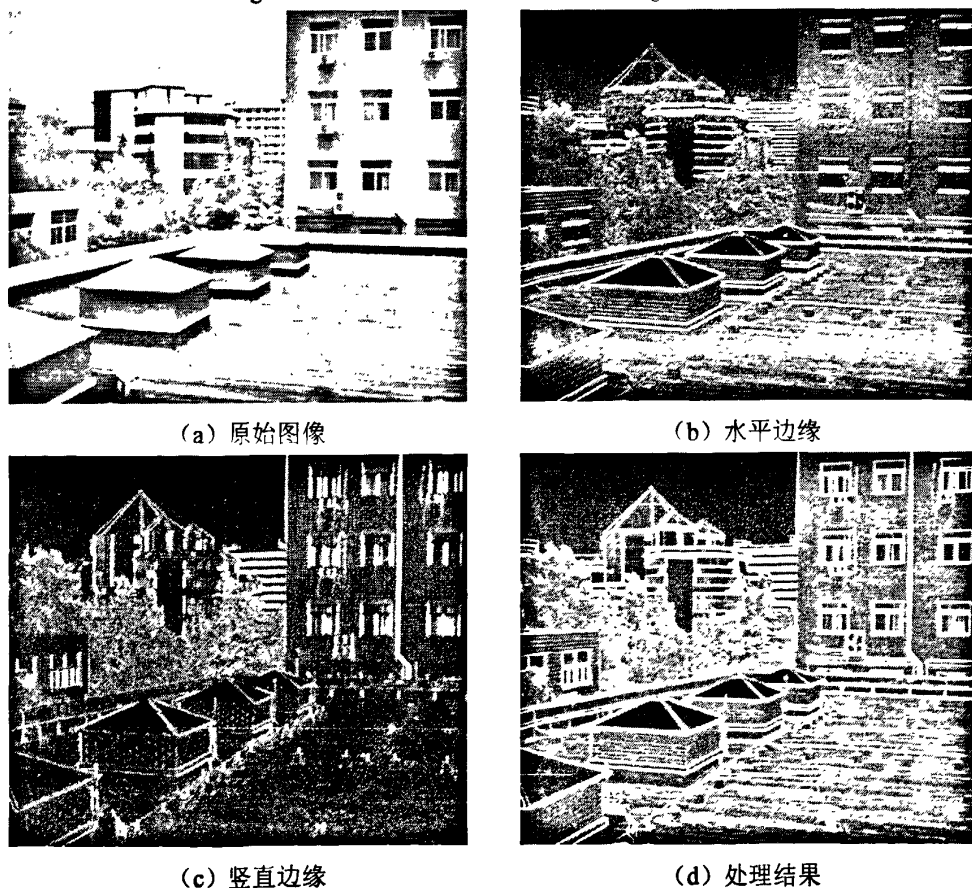


图 3-20 Sobel 算子边缘提取仿真波形图

Fig.3-20 Simulation waveform of Sobel edge detector



(a) 原始图像

(b) 水平边缘

(c) 竖直边缘

(d) 处理结果

图 3-21 Sobel 算子边缘提取处理结果

Fig. 3-21 Result of Sobel edge detector

另一种与上面有本质区别的边缘提取方法是利用形态学实现边缘提取的方法，即先对图像进行膨胀后再减去原图像。在图像代数中可以用式(3-56)表示。在进行硬件实现时，我们可以利用3.4.2节实现的膨胀运算硬件的结果减去图像模板中心元素即可，其仿真的结果如图3-22所示，处理结果如图3-23所示。

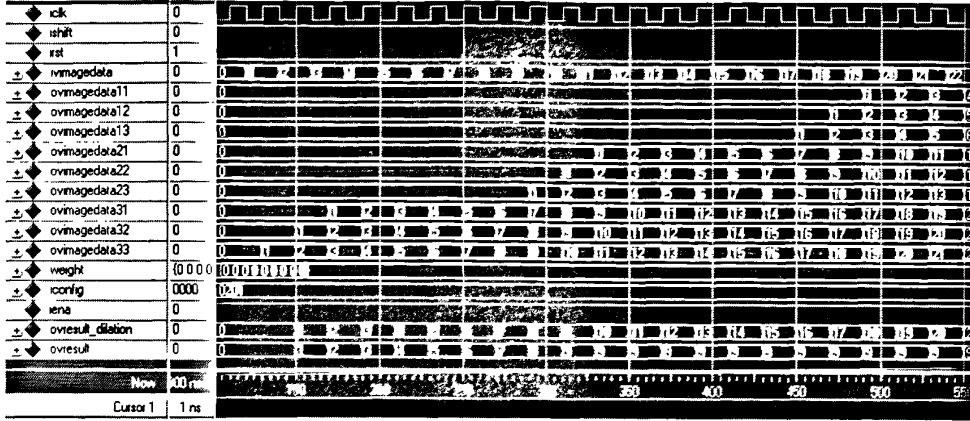


图 3-22 形态学边缘提取仿真波形图

Fig. 3-22 Simulation waveform of morphological edge detector



(a) 原始图像

(b) 膨胀结果



(c) 处理结果

图 3-23 形态学边缘提取处理结果

Fig. 3-23 Result of morphological edge detector

4 基于图像的线路环境状态提取与识别

上一章我们介绍了图像处理算法的硬件实现技术，通过硬件实现部分图像处理算法大大加快了系统的处理速度。经过处理的图像将需要 PC 软件进一步处理，并提取并识别线路环境状态的缺陷。本章我们将介绍基于二维图像的护栏完整性和空间异物的智能识别方法。虽然部分图像处理是在 FPGA 中实现的，但是为了保证方法叙述的完整性，在本章中我们依然介绍完整的识别过程而不只介绍 PC 软件实现的部分。

4.1 护栏完整性识别

为了确保安全，高速铁路实行的是全封闭的管理。护栏是重要的隔离实施，当护栏破损时，人员或者其他动物等将有可能进入线路。因此护栏的破损是较大的隐患，有必要对其进行检测。本节我们将介绍基于二维图像的护栏完整性识别方法。利用该方法可实现在综合检测列车通过线路时，自动对线路两侧的护栏进行检测。图 4-1 为一幅典型的护栏缺损的图像，两侧护栏都有一整列的缺失。

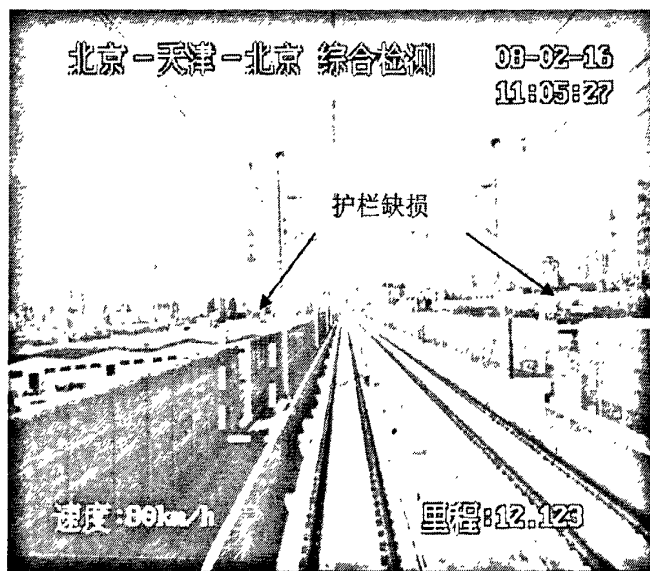


图 4-1 典型护栏缺损图像

Fig. 4-1 Image of typical fence defect

4.1.1 护栏图像几何变换

如图 4-1 所示, 护栏在图像平面中的投影是梯形的, 远处护栏在图像上成像的尺寸较小, 不利于图像的分析。事实上护栏平面和图像平面的投影是一个中心投影的关系, 因此我们可以使用图像几何变化的方法将图像中梯形的护栏区域变换成矩形区域。在变换得到的矩形区域上, 我们就可以使用统一的标准来评测缺陷的大小, 并且护栏的纹理也变成水平和垂直的直线, 从而方便了图像的分析。就来介绍护栏图像的变换方法。

1、变换矩阵

图像几何变换可以表示一个向量函数 \mathbf{T} 将图像区域中一点 (x, y) 变换到一个新的位置 (x', y') 。图 4-2 为在整个区域进行几何变换的示意图。

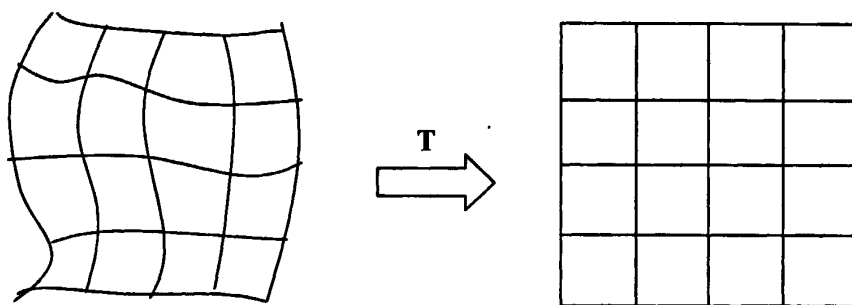


图 4-2 平面图像几何变换

Fig. 4-2 Geometric transform on a plane

\mathbf{T} 可以定义为两个方程组:

$$x' = T_x(x, y) \quad (4-1)$$

$$y' = T_y(x, y) \quad (4-2)$$

若采用向量形式, \mathbf{T} 可以定义为:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4-3)$$

式 (4-3) 中 \mathbf{T} 可以是任意的非线性向量方程。而特别地, 对于射影变换 \mathbf{T} 则可以表示成矩阵形式, 这样 \mathbf{T} 可以看作是变换矩阵:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4-4)$$

其中图像像素采用了齐次坐标形式表示。

从以上的式子可以看出, 图像变换的关键是确定变换矩阵 \mathbf{T} 。以下是几种常用几何变换的变换矩阵, 对于多个变换的组合变换其变换矩阵就是各个变换矩阵的

乘积。

(1) 尺寸缩放

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-5)$$

其中 s_x 和 s_y 分别为 x 轴和 y 轴的缩放因子。

(2) 旋转

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-6)$$

其中 θ 为顺时针旋转的角度。

(3) 平移

$$\begin{bmatrix} 1 & 0 & \delta_x \\ 0 & 1 & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4-7)$$

其中 δ_x 和 δ_y 分别为 x 轴和 y 轴上平移的距离。

(4) 水平和垂直剪切

$$\begin{bmatrix} 1 & \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ \beta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-8)$$

其中 α 和 β 分别为水平和垂直剪切因子。

另外若已知原图像和变换后图像中若干对对应点的位置，也可以求出变化矩阵。本文就采用了这种方法来求解变换矩阵。实际上对于护栏区域图像的几何变换只要已知梯形的 4 个顶点的坐标位置，和其对应的变换后矩形区域的 4 个顶点的坐标位置就可以求出变换矩阵。在求解时我们使用了 Matlab 图像处理工具箱。在 Matlab 中使用 CP2TFORM 函数就可以利用对应点求出变换矩阵。其使用方法如下^[28]：

TFORM = CP2TFORM(INPUT_POINTS, BASE_POINTS, TRANSFORMTYPE)

其中 INPUT_POINTS 为原始图像中对应点的坐标，是一个 $N \times 2$ 的矩阵。每一行为一个点的坐标，第一列为横坐标，第二列为纵坐标。

BASE_POINTS 是变换后图像中的对应点坐标，也是一个 $N \times 2$ 的矩阵。其每一行必须和 INPUT_POINTS 中的点对应。

TRANSFORMTYPE 指定的是变换的类型，如仿射变换 ('affine')、射影变换 ('projective')、多项式变换等 ('polynomial')。前面已经指出护栏平面和图像平面的投影关系是射影变换，因此我们在这项参数应选择使用 'projective'。

2、灰度插值

按式 (4-3) 进行变换时得到的坐标 (x', y') 通常不会是整数, 而变换所得的新图像必须是以整数为坐标的。这就表示原始图像中一个像素点经变换后其往往落在新图像的几个像素之间。这样原始图像中点的灰度值不能直接赋给新图像中的对应点, 因此新图像中像素点的灰度必须通过插值来得到。常用的插值方法有线性插值、最近邻域、双三次插值等。

在进行灰度插值时通常利用式 (4-3) 的反变换来确定原图像中哪些像素点和当前所求的像素点的灰度有关。设现在计算输出图像像素点 (x', y') 的像素值, 其中 $x', y' \in \mathbb{Z}^+$, 则其对应的原图像中的点可由是 (4-3) 的反变换得到:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{T}^{-1} \left(\begin{bmatrix} x' \\ y' \end{bmatrix} \right) \quad (4-9)$$

同样 x 和 y 通常也不是整数, 所以其灰度值是未知的。我们只知道原图像中横纵坐标为整数的点的坐标, 他其实是连续的图像函数 $f(x, y)$ 在整数点的采样后的离散函数 $g_s(l\Delta x, k\Delta y)$ 。为了获得点 (x, y) 上的灰度值必须进行插值。

若用 $f_n(x, y)$ 表示灰度插值后的结果, 其中 n 表示使用的不同插值方法。则其灰度可用如下卷积方程表示:

$$f_n(x, y) = \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} g_s(l\Delta x, k\Delta y) h_n(x - l\Delta x, y - k\Delta y) \quad (4-10)$$

其中函数 h_n 称为插值核。通常插值时只使用有很小的领域内的值, 因此 h_n 中大部分值是 0。

最简单的插值方法是最近邻域插值, 这种方法将离点 (x, y) 最近的离散点 g 的灰度值赋给点 (x, y) 。最近邻域可用图 4-3 (a) 表示, 图中实线表示原图像中的整数网格, 虚线为结果图像的整数网格反变化得到的网格; 图 4-3 (b) 为最近邻域插值方法的插值核 h_1 的一维情况。

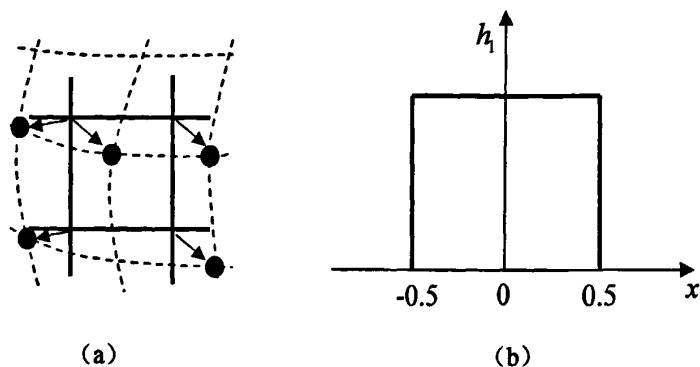


图 4-3 最近邻域插值

Fig. 4-3 Nearest-neighbor interpolation

这样最近邻域插值可以表示为：

$$f_1(x, y) = g_s[\text{round}(x), \text{round}(y)] \quad (4-11)$$

这种方法的位置误差最大为半个像素。这种误差在直线边缘的物体上比较明显，变化后的图像中该物体会出现阶梯状的边缘。

线性插值是另一种更加精确的方法，它使用领域中的四个点来计算点 (x, y) 的灰度值。这种方法架设灰度函数在领域内是线形的。线性插值的示意图如图 4-4 所示。

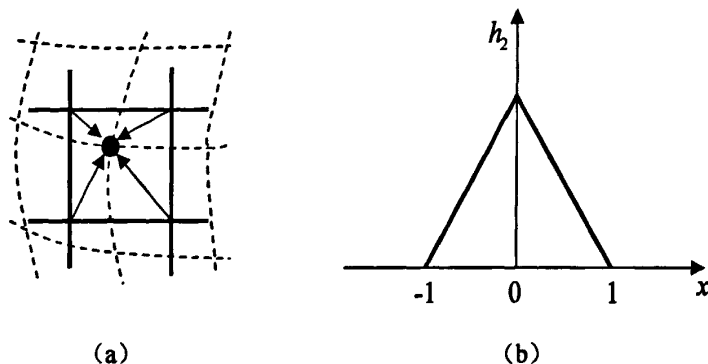


图 4-4 线性插值

Fig. 4-4 Linear interpolation

线性插值可以用如下方程表示：

$$f_2(x, y) = (1-a)(1-b)g_s(l, k) + a(1-b)g_s(l+1, k) + b(1-a)g_s(l, k+1) + abg_s(l+1, k+1) \quad (4-12)$$

其中 $l = \text{round}(x)$ ， $a = x - l$ ， $k = \text{round}(y)$ ， $b = y - k$ 。线性插值会使图像的清晰度有所下降，这是由于实际上对图像求了平均值。其阶梯状边缘的情况较最近邻域插值要小。

双三次插值采用一个双三次多项式来近似表示领域内的灰度函数。领域中的 16 个点被用来进行插值。其插值核的表达式为：

$$h_3 = \begin{cases} 1 - 2|x|^2 + |x|^3 & 0 \leq |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 & 1 \leq |x| < 2 \\ 0 & \text{其他} \end{cases} \quad (4-13)$$

双三次插值不会造成阶梯状边缘的问题，同样也可克服了线性插值中图像模糊的问题，并提供了很好的细节保持能力。

护栏图像几何变化的结果如图 4-5 所示，选用的是射影变换，灰度插值采用了双三次插值。如图 4-5 所示，选取如图 4-5 (a) 所示的护栏区域，其变换结果如图 4-5 (b) 所示。从图中可以看出原来梯形的护栏，经变换后成为矩形这样便简化了后续处理。

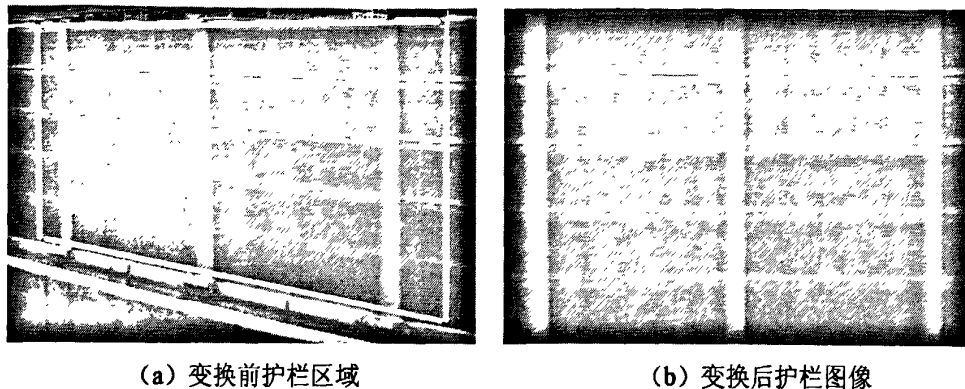


图 4-5 护栏图像几何变换

Fig. 4-5 Geometric transform of fence image

4.1.2 基于灰度统计特征的快速识别

根据护栏图像的特点我们首先提出了基于灰度的快速识别方法，由于这种方法是直接基于灰度统计特征的，因此具有较高的识别速度。特别是在外界光线较强的情况下，使得完好护栏和护栏缺失处的灰度相差较大，该方法将有较好的识别效果。该方法的识别流程如图 4-6 所示。

该方法是在经过几何变换后的图像上进行的。首先对图像进行分割，从图 4-1 可以看出护栏区域的灰度是比较一致的，这可以从护栏区域的灰度直方图中得到证实。图 4-7 为图 4-5 (b) 的灰度直方图，如图所示，大部分护栏区域像素的灰度在 50 至 100 直间，且分布较为集中。进一步可以求得该图像灰度的平均值为 75.2686，标准差为 12.2180。若假设正常护栏区域的灰度呈正态分布则对于灰度值不在区间 $[\mu - 3\sigma, \mu + 3\sigma]$ 内的像素点可以认为是异常的。因此我们可以采用阈值分割的方法对图像进行分割，将异常区域和正常区域分割开来。护栏区域灰度异常考虑到外界光照会有一定程度的变化这里采用了动态阈值分割的方法，即不断根据实际采集的图像更新统计特征从而动态调整分割阈值。

原图像经分割以后成为二值图像，即正常护栏区域像素都为 0，而异常像素为 1。由于图像噪声是不可避免的，因此为了减少图像噪声的影响需要对分割后的二值图像做形态学处理。噪声从两个方面影响分割的结构，一是在正常护栏区域出现零散的异常点，对于这种影响我们采用开启操作将这些点消除；另一方面噪声也可能使异常区域内出现正常灰度值的点，对此我们采用闭合操作。因此在这一步我们采用了一个矩形模板对图像首先进行了开启操作，然后再进行一次闭合操作。

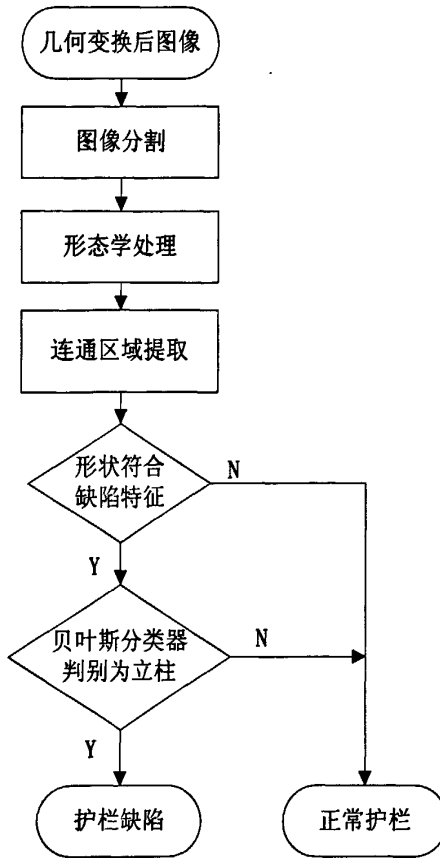


图 4-6 灰度统计特征识别流程

Fig. 4-6 Flow of recognition based on gray-level statistic feature

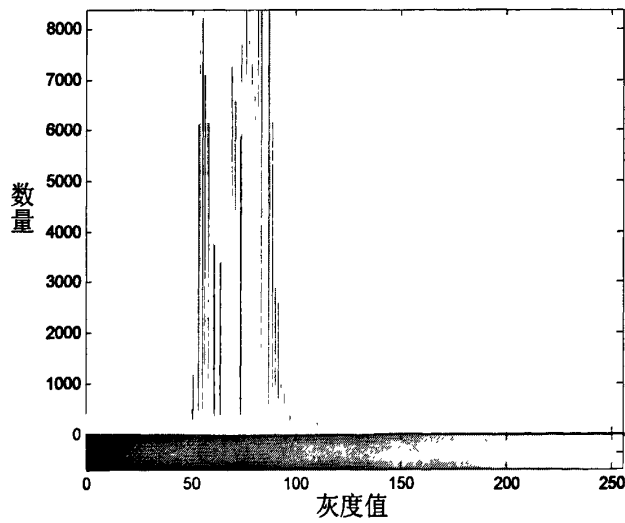


图 4-7 护栏图像直方图

Fig. 4-7 Histogram of fence image

对形态学处理后的图像,就可以进行连通区域提取了,即将二值图像中值为1且相互连通的像素提取出来并做一定的标记。相互连通的区域通常来自于同一个物体在图像中呈的像,而不连通的区域通常是两个不同的物体,这样对每个连通区域进行相应的特征选取才是有意义的。连通区域提取通常分为4连通法和8连通法,在这里两种方法的区别不大。

经过连通性提取后得到的区域,并不一定就是护栏缺陷。这些区域有可能是形态学处理未完全消除的噪声干扰,或是护栏内的一些正常设备,如接触网立柱、信号机以及它们在护栏上的阴影等。因此需要对每一个连通区域进行一定的选取,我们采用的主要选取指标是连通区域的面积,以及区域的长和高等。因为通常干扰造成的异常区域的面积比较小,而本系统设计的技术指标规定只检测缺损大于两个单元的护栏。

我们发现利用上一步的方法对连通区域进行选择后,还是较难区分接触网立柱和真实的护栏缺损区域。因此我们引入了贝叶斯分类器对上一步选取的区域进行分类^[29]。由于接触网立柱有较为集中的灰度分布,而对于护栏缺损区域,其实际上是护栏外的景物,故灰度的分布具有很大的随机性。因此我们可以选用连通区域灰度的平均值和标准差作为识别的特征值,而采用二维贝叶斯分类器对这两种情况进行区分。图4-8给出了基于该方法的识别结果。

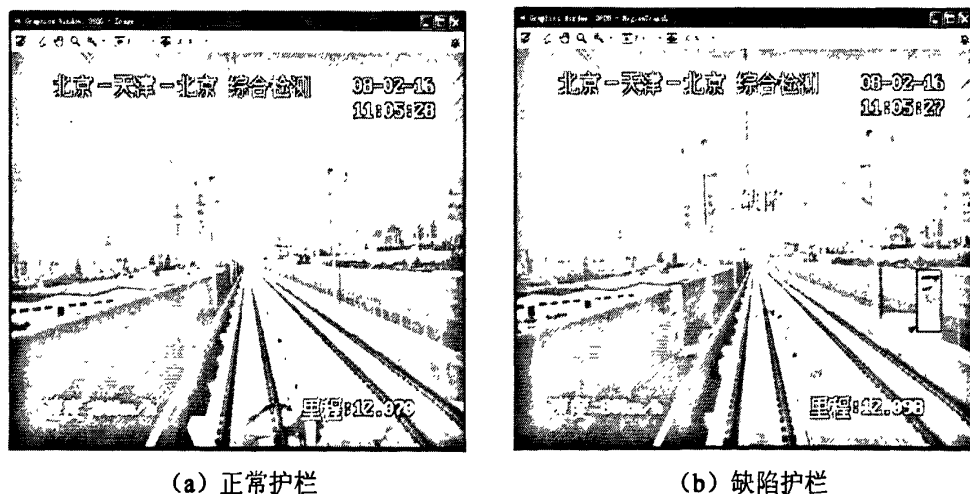


图 4-8 基于灰度统计特征的识别结果

Fig. 4-8 Recognizing result based on gray-level statistic feature

4.1.3 基于边缘统计特征的识别

上一节我们介绍了基于灰度统计特征的识别方法,这种方法识别速度较快,但是对外界光照的影响较敏感,这种方法比较适用于外界光照较强的时候。在光

线较弱时护栏区域和缺损区域的灰度值的差别将减小，因此我们又提出了基于护栏边缘图像统计特征的识别方法。由于边缘图像对外界光照的影响较小，故该方法可以在较大光照强度范围内使用。该方法的处理流程如图 4-9 所示。

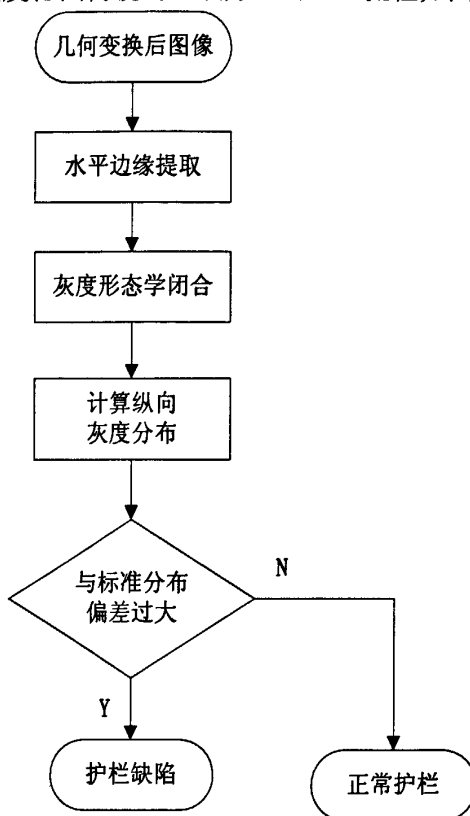


图 4-9 基于边缘特征的识别流程

Fig. 4-9 Flow of recognition based on edge feature

该方法首先对几何变换后的图像进行边缘提取。边缘提取的方法有很多种，主要是提取图像中灰度变换较大的位置，通常有两类方法：一是取一阶或二阶导数值较大的位置，二是提取二阶导数的过零点^[30]。实践中，之前的研究者提出了各种算子和图像进行卷积来实现边缘提取^[31-33]，其中必要常用的有：

(1) Laplace 算子

Laplace 算子是一种非常常用的算子，它是对二阶导数的近似计算。这种算子实际上是对 Laplace 方程 (4-14) 利用离散卷积和进行近似计算。

$$\nabla^2 g(x, y) = \frac{\partial^2 g(x, y)}{\partial x^2} + \frac{\partial^2 g(x, y)}{\partial y^2} \quad (4-14)$$

其近似精确程度和模板的大小有关。最常用的是 3×3 模板，其有两种形式分别采用了 4-领域和 8-领域中的数据。

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4-15)$$

Laplace 算子只能给出边缘的强度，而无法计算边缘方向。另外 Laplace 方法由于采用的是二阶导数，因此容易在某些边缘处提取出两个边缘。

(2) Prewitt 算子

Prewitt 算子和 Laplace 算子不同，它近似计算的是一阶导数，因此它可以计算边缘的方向。这种方法利用 8 个算子（3×3 模板）计算不同方向上的边缘强度，并取结果中强度最大的方向为边缘的方向。当扩大模板时还可以增加计算的方向数目。下面只给出了 3 个 3×3 模板，其他方向上的模板可以对这些模板进行旋转得到。

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (4-16)$$

(3) Sobel 算子

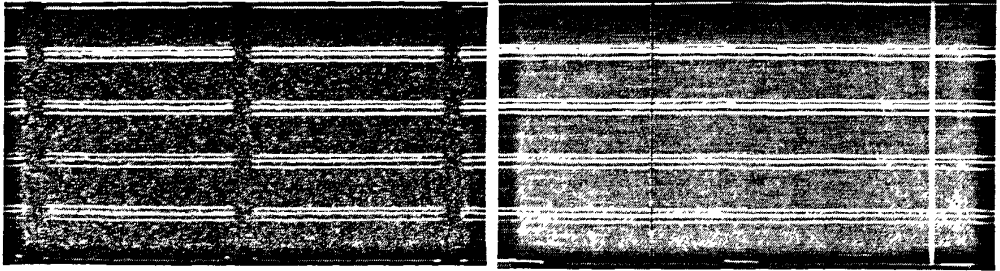
$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}, \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (4-17)$$

Sobel 算子在上一章中已经做过介绍，它通常被用来检测水平和垂直的边缘。它和 Prewitt 算子非常的相似，与 Prewitt 算子相比它给中心位置的导数赋予更高的。

如图 4-5 (b)，在经过几何变换的护栏图像上，有规则的水平 and 竖直的纹理，利用边缘提取我们可以将这些纹理加强，形成水平和竖直的条纹。由于列车运动，图像中竖直的条纹的位置将随机变化，而水平的条纹的位置和强度则将相对保持稳定。而当出现缺损时，将会出现异常的水平条纹或时正常的条纹消失，这样我们就可以利用水平条纹的位置和强度来判断护栏的完整性。

如上所示，我们只关心护栏图像水平方向的边缘，因此我们只需采用 Sobel 水平方向的算子，即式 (4-17) 中的 h_1 ，对图像进行卷积。对图 4-5 (b) 用 Sobel 算子进行卷积的结果如图 4-10 (a) 所示。

图 4-10 (a) 所示，只有水平方向的边缘被提取出来，但是水平方向上的边缘不连续，这是原图像中护栏垂直支撑柱的位置，因为其表面是一致的没有边缘。而支撑柱的宽度比单块护栏要窄得多，因此可以利用灰度形态学的闭运算将这部分不连续的特征消除。我们采用水平宽度大于立柱而小于单个护栏的，垂直宽度是 1 的模板对图 4-10 (a) 进行闭运算，其结果如图 4-10 (b) 所示。



(a) 原始水平边缘

(b) 形态学处理后的边缘

图 4-10 护栏水平边缘图像

Fig. 4-10 Horizon edge images of fence

这样处理以后水平的边缘就连续了，但是当护栏出现缺损时相应消失的水平条纹也将便短。不过由于支撑柱的宽度比单块护栏要窄得多，因此这并不会对识别结果造成影响。这样处理后接触网支柱的干扰也被消除。

为了比较水平护栏条纹的位置和灰度，可以对每一列纵向方向上的灰度分布情况进行比较图像。图 4-11 所示为图 4-10 (b) 中第 250 列（图中蓝色直线所标示的位置）和第 700 列（绿色直线位置）的纵向灰度分布曲线图。

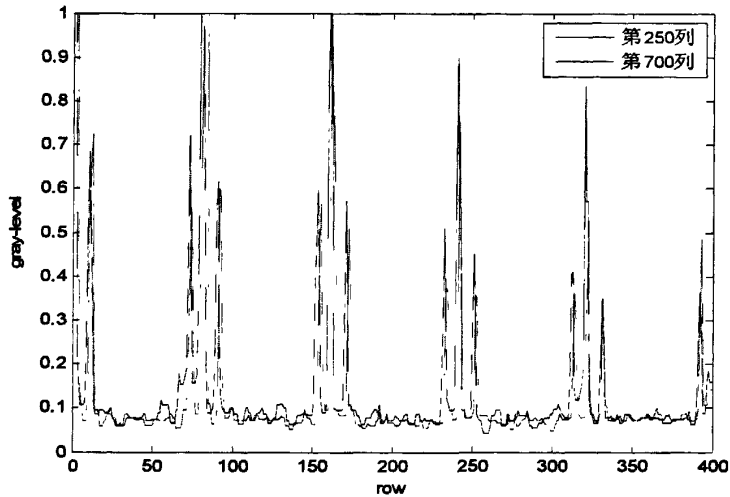


图 4-11 纵向边缘灰度分布图

Fig. 4-11 Vertical distribution of edge gray-level

如图所示，图中蓝色曲线对应图 4-10 (b) 中蓝色直线所标示的位置的纵向灰度分布，绿色曲线对应图 4-10 (b) 中绿色直线所标示的位置的纵向灰度分布。从图中可以看到，这两列都是正常护栏形成的条纹，因此这两条曲线的非常接近，极值位置基本重合。而若护栏有缺损部分，则缺损处纵断面上的灰度分布将有明显的区别。我们将上述的方法应用到图 4-1 的右侧护理区域上，得到如图 4-12 的结果。

图 4-12 (a) 为几何变换后的护栏图像，其边有一处缺损。图 4-12 (b) (c) 分别为边缘提取和闭合操作后的图像，从中可以看出图像在护栏缺损区域出现明显的不规则条纹。图 4-12 (d) 为 3 个典型断面上的灰度分布，曲线的颜色和 (c) 中直线标示的位置相对应，可以看到蓝色曲线所表示的护栏缺损区域灰度分布和正常区域有明显的区别。

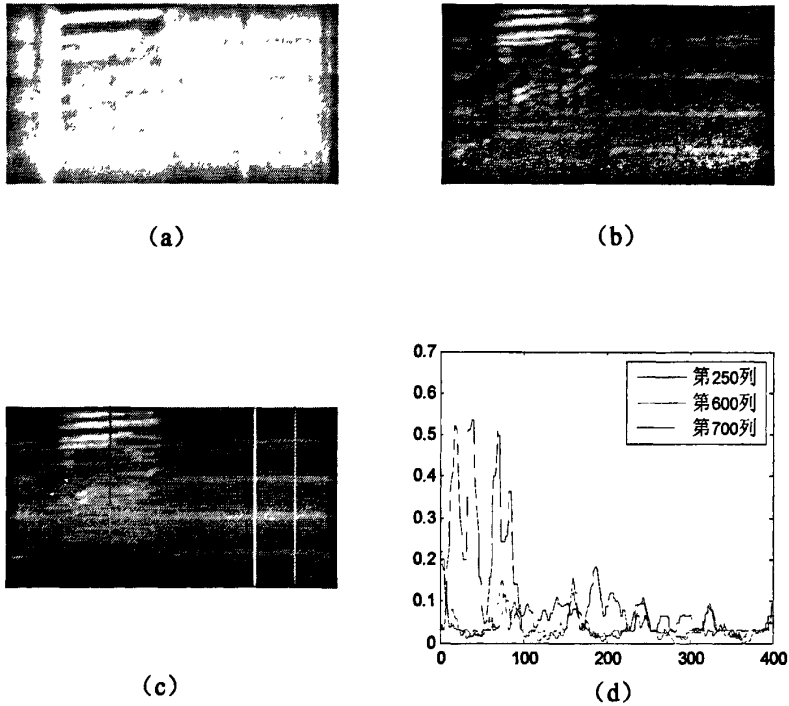


图 4-12 缺陷护栏纵向边缘强度分布图

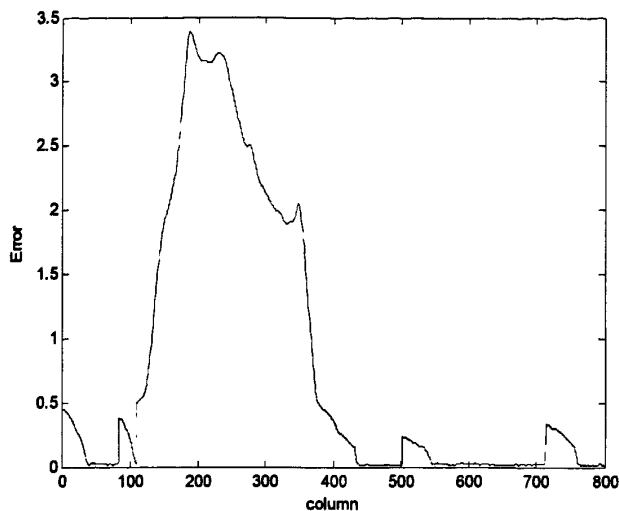
Fig. 4-12 Vertical edge gray-level distribution of broken fence

因此我们可以通过判断图像纵断面上的灰度分布判断护栏完整性。若设 c_i 和 c_{si} 分布表示待比较的断面和标准断面，则可引入断面的异常度 E ，并用如下的方程来表示：

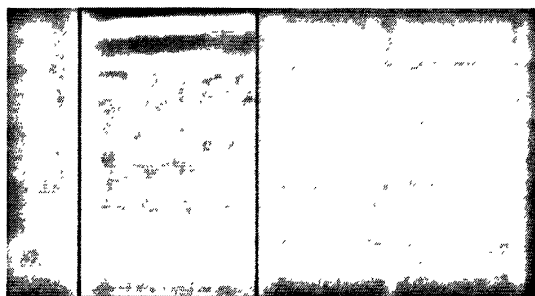
$$E = \sqrt{\sum_{i=1}^N (c_{li} - c_{si})^2} \quad (4-18)$$

其中， c_{li} ， c_{si} 分别表示两个断面的第 i 个数据， N 为断面像素个数，即图像的宽度。该断面的异常度 E 越小则与标准断面的区别就越小，因此当 E 大于一定的阈值时我们就认为其为异常断面。我们采用实时更新的标准断面的方法，如果当前断面被判断为正常断面时则将其设为标准断面；若判断为异常断面则保持原标准断面不变。利用上述方法就可以得到图 4-12(c) 中的每一列的异常度 E 曲线，如图 4-13 (a) 所示。从图中可以看到在正常护栏区域断面的异常度较小，而在缺损处异常度明显增大。若设阈值为 0.5，则可以得到属于异常护栏区域的列为第 111

列至第 377 列，将这些列在护栏上标出后的结果如图 4-13 (b) 所示。



(a)



(b)

图 4-13 边缘统计特征识别结果

Fig. 4-13 Recognition result of edge statistic feature

4.2 空间异物识别

4.2.1 识别过程

在高速铁路线路区间内由于各种原因可能出现异物，这些异物在一定条件下，如大风等，可能侵入限界影响列车运行安全。因此有必要对其进行检测，以便及时清理。由于无法预知会出现怎样的异物，因此异物的特征无法预测，也无法对特征进行建模。而另一方面线路特征比较单一，这样就可以只能通过对线路背景进行建模，从而识别图像中背景以外的元素，作为异常特征。空间异物识别的流程如图 4-14 所示。

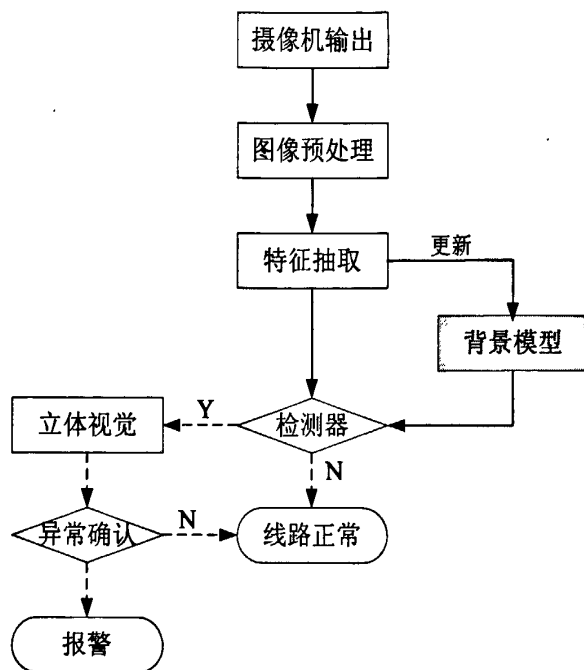


图 4-14 空间异物检测流程图

Fig. 4-14 Flow of special foreign object

首先,对输入的图像进行预处理,主要是滤波处理以减少噪声对识别结果的影响。经预处理的图像将根据线路背景模型,逐点判断像素为背景或异物。通过建立背景模型,将当前图像和背景模型进行差分,并且和阈值比较,确定出差值的绝对值大于阈值的区域为异物区域。为了提高可靠性,我们通过立体视觉方法重建异物区域的三维信息,来确认其是否确实为空间异物。三维重建的具体方法将在下一章进行介绍。

4.2.2 基于混合高斯模型的背景估计

背景模型按照自身的特性来划分可以分为单模态和多模态两种^[34]。单模态背景模型在每个背景像素上的颜色分布比较集中,可以用单分布概率模型来描述;而多模态背景模型在每个背景像素上的颜色分布比较分散,需要用多分布概率模型来描述^[35]。在本系统中,由于存在光照变化,移动的阴影等,因此背景像素值具有多模态的特性。所以本文采用了混合高斯模型来对背景进行建模。

混合高斯模型是有限个高斯模型的线性组合^[36]。已知高斯分布的概率密度函数为:

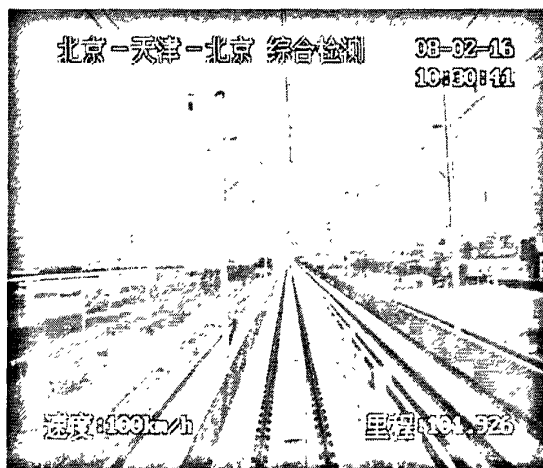
$$\eta(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \quad (4-19)$$

若设背景的混合高斯模型为 K 个高斯分布模型的线性组合，则可定义当前观测点像素值为背景的概率为：

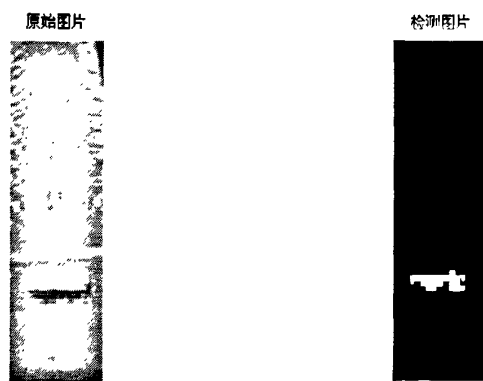
$$P(X_t) = \sum_{i=1}^K \omega_{i,t} \cdot \eta(x_t, \mu_{i,t}, \Sigma_{i,t}) \quad (4-20)$$

其中 K 为分布的个数，通常取 3~5， $\omega_{i,t}$ 是在 t 时刻混合高斯模型中第 i 个高斯模型的权值， $\mu_{i,t}$ 以及 $\Sigma_{i,t}$ 分别为 t 时刻混合高斯模型中第 i 个高斯模型的均值和协方差矩阵。

使用混合高斯模型来表征图像中各个像素点的特征时，用当前图像中的每个像素点与混合高斯模型匹配，如果匹配成功则判定该点为背景点，否则为前景点。由于缺少真实异物的图片，我们只能利用现有正常线路视频，并将查询应答器作为异物对算法进行了测试，结果如图 4-15 所示。我们选取图 4-15 (a) 中红框内为监控区域，(b) 为监控区域内的图像，(c) 为检测结果，白色像素表示异常区域。



(a)



(b)

(c)

图 4-15 基于混合高斯模型的识别结果

Fig. 4-15 Recognition result based on Gaussian mixture model

4.2.3 自适应背景更新

由于背景会随着时间不断地变化,因此需要对背景进行更新。背景进行更新的过程就是背景混合高斯模型的学习过程,其基本方法是通过输入值来更新模型的参数^[37]。这样随着背景逐渐变化,背景模型的参数也不断变化,从而保证背景模型总能够较好地反映背景的实际情况。其具体方法如下

(1) 高斯模型匹配

首先我们把图像中每个像素点 X 的值和已经存在的 K 个高斯模型分布进行匹配,如果当前像素点的观测值与某个分布的均值相近,则认为该像素点和这个分布匹配。与第 i 个分布匹配的条件可以表示为:

$$|x_t - \mu_{i,t-1}| \leq D\sigma_{i,t-1} \quad (4-21)$$

其中 x_t 为当前 t 时刻像素的观测值, $\mu_{i,t-1}$ 为 $t-1$ 时刻,即上一时刻,第 i 个高斯分布模型的均值, $\sigma_{i,t-1}$ 为 $t-1$ 时刻第 i 个高斯分布模型的标准差, D 为匹配阈值,也称置信区间,在本文中设为 2.5。即当输入值与某个高斯分布模型的均值差的绝对值小于标准差的 2.5 倍时,就认为其和该分布匹配。

(2) 权值更新

然后我们根据匹配的结果对各分布的权值按下式进行更新。

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} + \alpha M_{i,t} \quad (4-22)$$

其中 $M_{i,t}$ 的值为

$$M_{i,t} = \begin{cases} 1, & |x_t - \mu_{i,t-1}| \leq D\sigma_{i,t-1} \\ 0, & |x_t - \mu_{i,t-1}| > D\sigma_{i,t-1} \end{cases} \quad (4-23)$$

α 为学习效率,且 $0 \leq \alpha \leq 1$ 。当 α 较大时,适应环境变化能力强,但容易受噪声干扰,前景运动物体对背景模型的影响也较大;反之如果 α 取得比较小,则适应环境变化的能力就低,需要足够长的时间才能适应环境的变化,因此 α 取值的好坏对背景模型的优劣有很大的影响^[38]。

(3) 更新高斯分布模型参数

最后更新各个高斯分布模型的均值和方差。若当前的混合高斯模型中没有高斯分布模型和当前输入值匹配,则新建一个模型来替换当前权值最低的模型,这个新建模型的均值为当前输入值 x_t ,并初始分配一个较大的标准差 σ_0 和一个较小的权值 ω_0 ^[39]。

若存在匹配的高斯模型,则对于匹配成功的模型可按下式更新期均值和方差:

$$\begin{aligned} \mu_{i,t} &= (1-\rho)\mu_{i,t-1} + \rho x_t \\ \sigma_{i,t}^2 &= (1-\rho)\sigma_{i,t-1}^2 + \rho(x_t - \mu_{i,t})^2 \end{aligned} \quad (4-24)$$

其中 ρ 为参数的学习效率,通常取 $\rho \approx \alpha / \omega_{i,t}$ 。

5 基于立体视觉的线路环境状态识别

上一章介绍了基于二维图像信息的线路环境状态的识别。二维识别可以较快地提取可疑的缺陷特征，但是其可靠性较差容易产生误检。为了降低误检率，我们采用了基于立体视觉的三维识别技术。系统对由二维识别检出的可疑特征提取三维信息，从而确认其是否为缺陷。本章将介绍基于立体视觉的三维识别方面的相关问题。

5.1 立体视觉几何学

在第二章中，我们介绍了单个相机的模型，并以此为基础分析了本系统相机选型和安装的问题。本节我们继续讨论双目立体视觉的几何学问题，为后面的讨论奠定基础。从单个相机的模型中我们不难发现，当场景投影到图像平面上后其深度信息便丢失了，因此通常无法从单个相机获得物体的三维信息。而双目立体视觉是从人的双目产生立体感而获得启示，利用两个不同位置的相机同时拍摄同一个物体的图像，这样就可以从中提取该物体的三维信息。

为了便于分析我们假设两个相机平行放置，并具有相同的内参数，且连接两个相机光学中心的直线与相机 1 的 x 轴重合，如图 5-1 所示。连接两相机光线中心的直线通常称为两相机的基线^[40]。

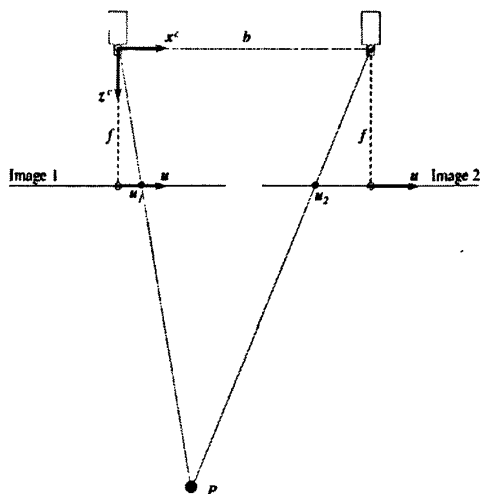


图 5-1 双目立体视觉系统平面图

Fig. 5-1 Vertical section of a binocular stereo camera system

这样根据相机模型场景中一点 $P(x^c, y^c)$ 在两个图像平面上的投影可以表示为：

$$u_1 = f \frac{x^c}{z^c} \quad (5-1)$$

$$u_2 = f \frac{x^c - b}{z^c} \quad (5-2)$$

其中 f 为两相机的焦距, b 为两相机基线的长度。

同一个场景点在两个图像平面形成的像点称为对应点, 而对应点位置的差称为视差 d :

$$d = u_1 - u_2 = \frac{f \cdot b}{z^c} \quad (5-3)$$

这样只要已知相机的参数和两个相机之间的相对位置关系, 根据视差就可以计算出场景点 P 的 z 坐标:

$$z^c = \frac{f \cdot b}{d} \quad (5-4)$$

进而可以求得:

$$x^c = \frac{u_1 \cdot b}{d}, \quad y^c = \frac{v_1 \cdot b}{d} \quad (5-5)$$

这样就确定了场景点的三维信息。从式 (5-4) 可知距离和偏差成反比。因而只有在较近的距离时, 测量才比较精确。

从式 (5-4) 中还可以看出三维信息的确定和相机参数和两相机的相对位置有关, 因此立体视觉任务可以分为两个关键技术^[9]:

- 1、确定相机参数和两相机相对位置、即摄像机标定;
- 2、在两图像中确定对应点、通常被称为图像匹配。

5.2 摄像机标定

5.2.1 摄像机标定原理

对于双目立体视觉, 摄像机标定包括两个部分。首先要对两个摄像机的内部参数分别进行标定, 然后再对两个相机的相对位置进行标定。摄像机标定就是确定其内外参数的过程, 式 (2-11) 给出了空间点和图像点的变换关系, 将其简化后可得:

$$\mathbf{P}' = M\mathbf{P}^w = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \mathbf{P}^w \quad (5-6)$$

其中 \mathbf{P}^w 为空间点在世界坐标系中的齐次坐标, \mathbf{P}' 为该点在图像上的投影点在图像坐标系中的齐次坐标, 矩阵 M 称为相机矩阵。

相机矩阵是一个 3×4 的矩阵表示了相机所有的内参数和外参数。所以相机标定实际上就是求解矩阵 M 的过程。如果我们已知空间一点的坐标和其对应于的图像中点的坐标，那么就可以建立起两个约束方程。 M 中一共有 12 个参数，其中独立的只有 11 个，所以如测得 6 个以上点的世界坐标和相应的像点的图像坐标，就可以得到 12 个以上的线性方程，从而可以用最小二乘法求解出 M ，并得到所需的变换关系^[40]。

从上面的分析可以看出，为了进行摄像机标定必须已知世界坐标系中足够多三维点的坐标，并建立这些点在图像中的投影点的二维坐标的对应关系。通常我们通过在场景中放置一个尺寸已知的标定板来获得精确的空间点位置，在标定板上精确布置圆形或矩形的标志点，这样软件可以方便且精确地提取出这些点在图像中的坐标^[41]。本文使用的标定板如图 5-2 所示，其有一个黑色的矩形边框将标定板的内部区域和背景分开，这样便于寻找标定板的位置。在边框的左上角设置一个标记，这样在进行识别时可以唯一的确定标定板的方向。标定板内部使用均匀布置的黑色圆形标记点，这些标记点的相对位置是已知并可以制作得很精确，同时在图像中可以很方便地提取这种标记点的中心位置。

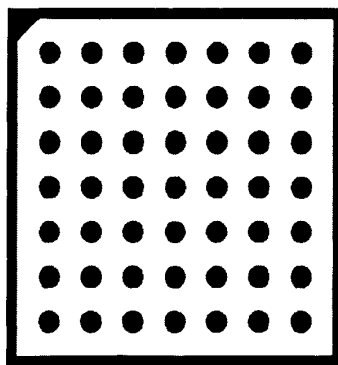


图 5-2 标定板

Fig. 5-2 The calibration plate

5.2.2 双目标定实验

我们在实验室建立了一个双目视觉的实验环境，来进行标定和三维重建的实验，如图 5-3 所示，两个相机的基线长度大约为 400mm。在进行双目相机标定前必须先将两个相机固定，以避免相机的相对位置发生变化。一旦相机的相对位置改变，就必须重新对相机对进行标定。

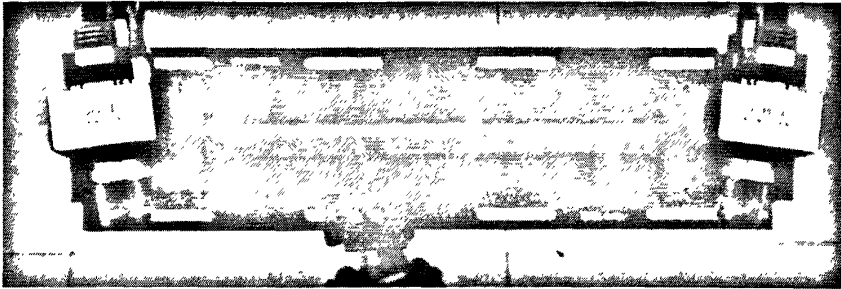


图 5-3 双目相机

Fig. 5-3 Stereo camera system

我们将标定板放入场景中，同时采集左右两个相机的图像。改变标定板在场景中的位置和姿态，再进行采集，为了保证精度至少需要采集 10~15 对的标定图像，并应该使标定板尽量覆盖整个图像区域，且标定板应该具有不同方向的旋转。在拍摄标定图像时，标定板必须被完全包含在两个相机拍摄的图像中。图 5-4 为实验采集的标定图像对的 4 对，上下两行图像一一对应，分别为左相机和右相机采集的图像。

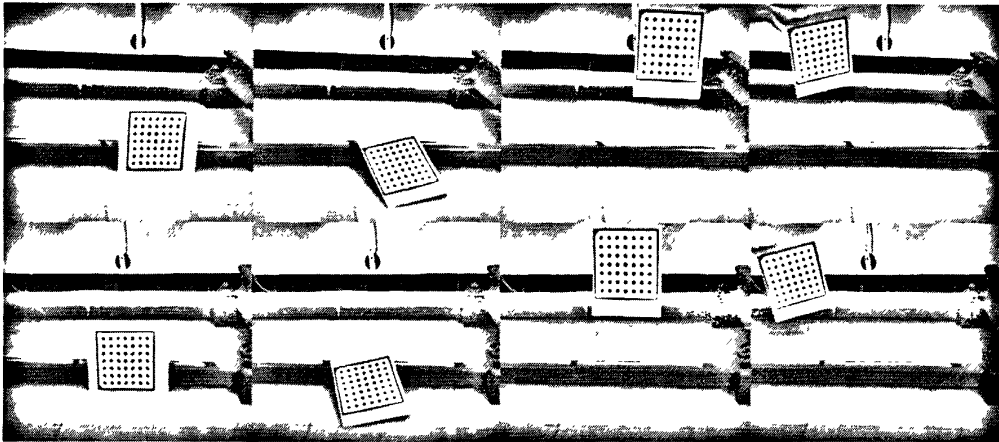


图 5-4 标定图像

Fig. 5-4 Images for calibration

在采集标定图像后，我们需要利用图像处理的方法提取出标定板上的点在图像中的位置，整个提取过程的流程图如图 5-5 所示。首先我们需要在图像中找出标定板区域，这里采用了阈值分割的方法，由于标定板中的白色区域的灰度值将比其他区域的大，因此只要设置一个合适的阈值将比该阈值大的区域提取出来即可。在提取出来的区域上进行连通性提取从而获得多个连通的区域，最后在这些区域中选择区内孔的数目与标定板上黑点的数目相同的区域。为了避免噪声的干扰，我们可以设定一个最小孔径的阈值，将小于该值得孔忽略。为了达到较好的分割效果，在拍摄标定图像时应该使标定板白色区域的灰度值大于 100，并保证其和黑点之间也有 100 的灰度差。

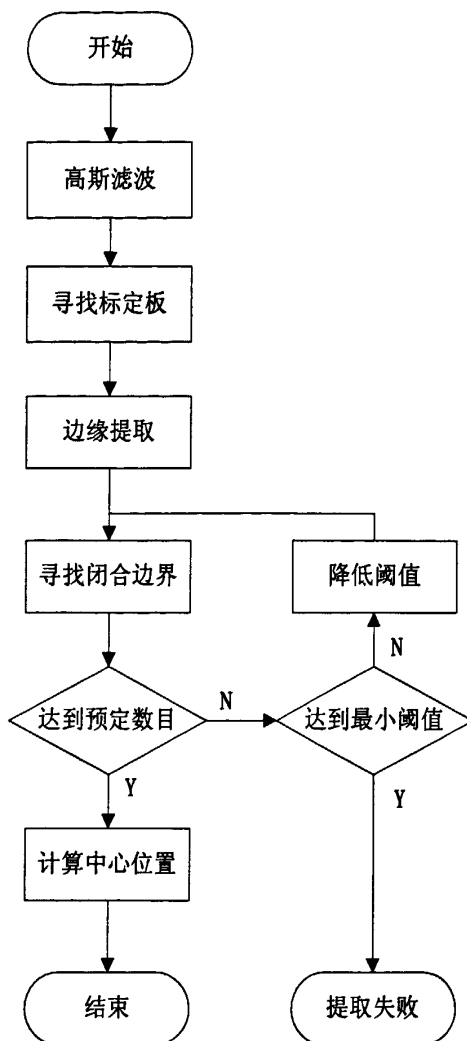


图 5-5 标定板标志点提取流程图

Fig. 5-5 Flow of points abstract of calibration plate

在找到标定板区域后，我们就可以只对标定板内的区域进行处理。我们首先对其进行边缘提取得到边缘灰度图像，边缘图像中灰度较大的区域即为原图像中的轮廓部分。我们首先设置了一个起始阈值对边缘图像进行阈值分割并提取闭合轮廓，若闭合轮廓的数目达到了预计的数目，即标定板上黑色标记点的数目（本文采用的标定板有 49 个标记点）；若未到达预计的数目，则按预先设定的步长降低阈值继续进行上一步，直到找到足够数目的闭合边界。同时我们还设置了最小阈值，当不断降低阈值到达最小阈值后还未找到足够数目的边界则提取失败。在标定板的对比度太小，或者图像中标定板区域太小等情况下容易出现提取失败，因此在拍摄标定图像是应该注意图像的质量。在找到足够的闭合轮廓后，可以认为轮廓包围区域的中心就是标记点的中心。其中心可以采用椭圆型曲线 Hough 变换、重心法等算法进行计算，本文采用了重心法。标定板的提取结果如图 5-6 所示，

由于采用的标定板外框有一个缺口，因此我们可以唯一地确定标定板的方向。

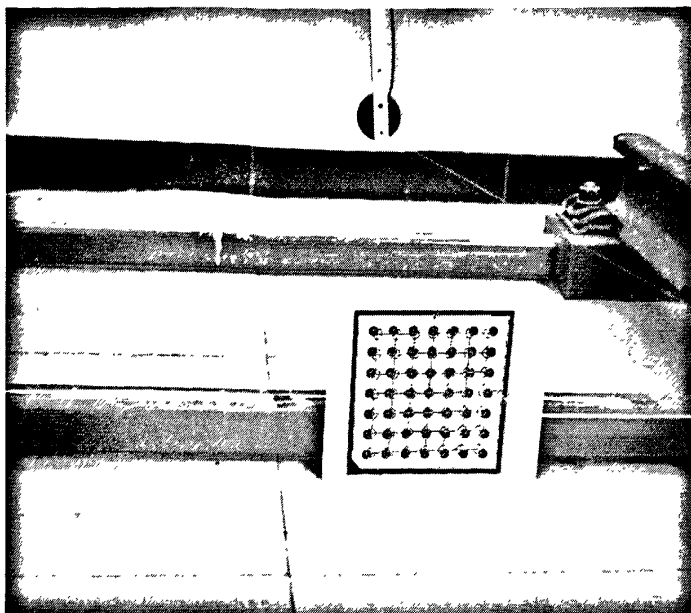


图 5-6 标定板提取结果

Fig. 5-6 Result of calibration plate abstraction

我们已经找出了图像中标定板标记点的位置，而这些点在空间坐标中的相对位置时已知的。这样我们就可以采用 5.2.1 节所介绍的标定原理计算相机的内外参数了。表 5-1 和表 5-2 给出了标定的结果。其中表 5-2 中表示的是右相机的相机坐标系相对于左相机的相机坐标系的变换关系。

表 5-1 相机标定结果（内参数）

Table 5-1 Result of camera calibration (interior parameters)

	左相机	右相机
焦距 f (m)	0.0303849	0.028814
畸变系数 κ	-215.568	-402.108
像素宽 S_x (m)	1.40E-05	1.39E-05
像素高 S_y (m)	1.40E-05	1.40E-05
图像中心横坐标 C_x (像素)	810.338	577.876
图像中心纵坐标 C_y (像素)	641.246	895.451
图像宽 (像素)	1280	1280
图像高 (像素)	1024	1024

表 5-2 相机相对位置

Table 5-2 Relative pose of two cameras

	X 轴	Y 轴	Z 轴
平移 (m)	0.37557	-0.00971	0.148905
旋转 (度)	352.241	342.192	357.411

5.3 线路环境状态异常点的三维信息重建

在进行了相机标定后，我们就得到了相机的内参数和两相机的相对位置。这样我们就可以进行三维信息的重建了。三维信息重建就是根据图像中两个对应点的位置计算出该点在空间中的三维坐标的过程。

5.3.1 立体图像校正

在 5.1 节中我们介绍了双目立体视觉的原理，得到了空间点的 z 坐标与其像点在两幅图像中的视差成正比的结论。在分析时我们假设两个相机是按照理想条件布置的，即两个相机平行放置，并具有相同的内参数，且连接两个相机光学中心的直线与相机 1 的 x 轴重合。但是实际中使用的双目立体视觉系统是不满足这样的条件的，或者为了某种目的必须使两个相机成一定的夹角，如图 5-7 (a)。

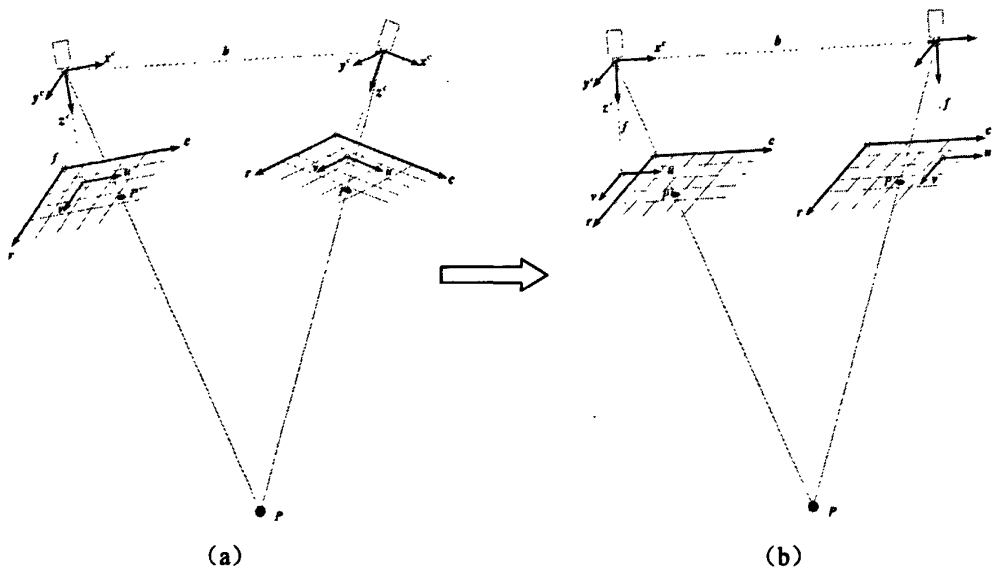


图 5-7 立体图像校正示意图

Fig. 5-7 Rectifying the stereo images

这样尽管两幅图像中的对应点依然满足外极线约束、相容性约束、惟一性约束等基本约束，但是分析起来很不方便且计算较慢。因此有必要采用了立体图像矫正的方法，将不标准的图像对变换为按照标准布置的图像对，即如图 5-7 (b) 所示的情况。这种情况下两幅图像中的对应点总是在同一行，这样不仅缩短了后面立体匹配所需的时间，还简化了计算三维坐标的过程。

可以证明实际中所有的立体视觉几何机构都可以进行这样的变换。为了校正图像我们需要将两个新的成像平面放在同一个平面上。为了保持三维几何结构的一致性，两个投影中心的空间位置不能变，我们只需要旋转相机坐标系使他们的 x 轴与基线方向相同。另外，我们需要创建两个新的中心点，它们之间的连接向量必须与基线平行。这样将留给我们两个自由度。首先我们必须选择一个公共的焦距，然后可以绕基线旋转包含两个成像平面在内的公共平面。这些参数可以通过最小化图像畸变来选择。一般情况下选取图像尺寸都需要使原始图像完全包含在校正后图像中，并消除畸变^[7]。

图 5-8 表示了立体图像校正的结果。图 5-8 (a) 和 (b) 为输入的立体图像对，拍其相机的内参数见表 5-1。图 5-8 (c) 和 (d) 为校正后的图像对，从图中可以看到两幅图像中的对应点都在同一行中，即外极线都是平行的。

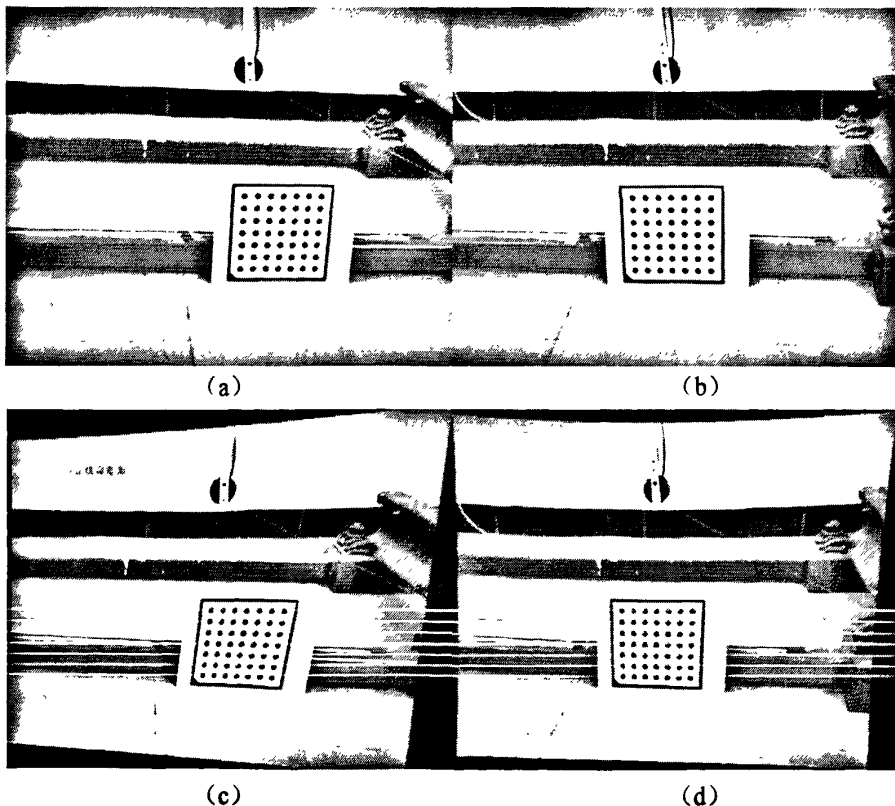


图 5-8 立体图像校正结果

Fig. 5-8 Rectified stereo images

图 5-8 (c) 和 (d) 可以看作是经过变换后的相机所拍摄的图像, 经变换后的相机的内参数见表 5-3, 这两个相机的相对位置见表 5-4。

表 5-3 变换后相机的内参数

Table 5-3 Interior parameters of rectified cameras

	左变换相机	右变换相机
焦距 f (m)	0.0285479	0.0285479
畸变系数 κ	0	0
像素宽 S_x (m)	1.40E-05	1.40E-05
像素高 S_y (m)	1.40E-05	1.40E-05
图像中心横坐标 C_x (像素)	-7.95599	472.029
图像中心纵坐标 C_y (像素)	1043.57	1043.57
图像宽 (像素)	1488	1423
图像高 (像素)	1158	1158

可以看到经过变换后, 两个相机的焦距以及像素大小等参数变为相等, 且畸变被消除; 而图像中心的位置发生了很大的变化, 其中左相机变换后其中心变换到了图形的外部, 并且图像的大小也发生了变化。

表 5-4 变换后相机的相对位置

Table 5-4 Relative pose of rectified cameras

	X 轴	Y 轴	Z 轴
平移 (m)	0.404128	0	0
旋转 (度)	0	0	0

从表中可以看出, 变换后两个相机间的相对位置只剩下 x 轴方向的平移, 这和预期的结果是一致的。这平移的距离就是两个相机的基线长度, 和我们实验环境的设置也是一致的。

5.3.2 立体匹配

通过前面的介绍我们可以看到, 进行空间点三维信息重建的最重要步骤是求图像中对应点的视差, 而要求视差则必须找到立体图像对中的对应点。在立体图像对中寻找对应点的过程就称为立体匹配^[42]。通常在进行三维信息重建时会计算第一幅图像中的每个点的视差, 这种算法被称为密集型重建算法^[43]。显然密集型重建算法会消耗大量的时间, 因此在本系统中我们不采用密集型重建方法。在本

系统中，基于立体视觉的识别只是二维图像识别的补充，即确认二维图像识别提取的异常特征是否为线路环境的异常状态。因此我们只需要对之前已经提取的异常特征进行三维重建就可以了，这样就大大减少了立体匹配的计算量；并且立体图像对已经按上一节的方法进行了校正，只需要在一行图像中寻找对应点就可以了。

典型的立体匹配的算法有 SAD (sum of absolute gray value differences) 和 SSD (sum of squared gray value differences) 两种。这两种算法都是基于模板匹配的原理，只是度量相似度的值不同^[44-46]。其基本思路是以所求点位中心建立一个 $(2n+1) \times (2n+1)$ 的矩形模板，然后在另一幅图像的外极线上寻找与模板最相似的区域。这样主要的问题就变为如何度量两个模板的相似程度。

设在第一幅图像为 $g_1 = g_1(r, c)$ ，第二幅图像为 $g_2 = g_2(r, c)$ ，第一幅欲匹配的点位为 $P(r, c)$ ，这里设 g_1 和 g_2 为校正以后的图像，即两图像中对应点的行坐标相同。这样可以采用如下值来度量模板相似性，它们都是基于灰度值匹配的方法。

$$\text{sad}(r, c, d) = \frac{1}{(2n+1)^2} \sum_{j=-n}^n \sum_{i=-n}^n |g_1(r+i, c+j) - g_2(r+i, c+d+j)| \quad (5-7)$$

$$\text{ssd}(r, c, d) = \frac{1}{(2n+1)^2} \sum_{j=-n}^n \sum_{i=-n}^n (g_1(r+i, c+j) - g_2(r+i, c+d+j))^2 \quad (5-8)$$

其中 d 为视差。

从式 (5-7) 和 (5-8) 中可以看出，这两个值度量的都是两个区域灰度的差别。在允许的视差范围内计算 (5-7) 或 (5-8)，使得 sad 值或 ssd 值最小的视差 d ，即为所求的视差。sad 算法最大的问题是对光照比较敏感，由于两个相机的视角不同，光照对两幅图像的影响也将不同，可能造成两幅图像中的对应点的灰度有所不同。因此我们采用了一种归一化的方法 (NCC) 来计算相似性，其可以表示为：

$$\text{ncc}(r, c, d) = \frac{1}{(2n+1)^2} \frac{\sum_{i=-n}^n \sum_{j=-n}^n \frac{g_1(r+i, c+j) - m_1(r+i, c+j)}{\sigma_1(r+i, c+j)} \cdot \frac{g_2(r+i, c+d+j) - m_1(r+i, c+d+j)}{\sigma_1(r+i, c+d+j)}}{\sigma_1(r+i, c+d+j)} \quad (5-9)$$

其中 m_i 和 σ_i ($i=1, 2$) 分别表示第一幅和第二幅图像中窗口部分的灰度均值和标准差。这种方法的优点是对线形的关照变化影响较小，但是计算量较大。

为了进行匹配我们必须在第二副图像的外极线上搜索最相似的点，实际上我们并不用搜索整条外极线。因为视差和深度有关并随着深度的增加而减小，而一般我们事先知道测量物体的大致位置。这样我们就可以设置一个可能的视差范围，并只在这个范围内进行搜索。这样求视差的过程可以表述为：对于 $d \in [d_{\min}, d_{\max}]$ ，使得 $\text{sad}(r, c, d)$ ， $\text{ssd}(r, c, d)$ 最小，或 $\text{ncc}(r, c, d)$ 最大的 d ，即为点 (r, c) 的视差。

在实际中，由于多种原因如由于透视投影导致当前点在第二幅图像中被遮挡

等, 容易出现错误匹配, 这样有必要对相似度量设置一个阈值, 只有 SAD, SSD 值低于或 NCC 值大于所设的阈值时才认为这个匹配是正确的, 否则认为匹配失败。我们利用 NCC 算法对图 5-8 (c) 和 (d) 计算了视差, 其结果视差图如图 5-9 所示。图中较亮区域表示视差较大, 反之则视差较小。



图 5-9 视差图

Fig. 5-9 Disparity image

5.3.3 三维信息重建

在经过立体图像校正, 并通过立体匹配得到了视差后, 就可以通过式 (5-4) 以及 (5-5) 来计算三维信息了。其中焦距, 基线长度等要使用校正后的相机参数即表 5-3 和表 5-4 所列的数据, 而计算得到的三维坐标也是以校正后的左相机的相机坐标系为参考的。

需要注意的是, 在利用上一节的方法得到的视差是以图像坐标系 RC 为参考的, 而式 (5-4) 中采用的是图像平面坐标系 UV 下的视差值, 因此需要先进行转换。为了使用方便我们可以直接推导基于图像坐标系 RC 的视差和距离的关系。设空间点的投影在两个图像坐标系中的坐标分别为 (r_1, c_1) 和 (r_2, c_2) , 图像坐标系下的视差为 d' , 则根据式 (5-1), (5-2) 以及 (2-5) 可得

$$c_1 = \frac{u_1}{S_x} + C_{x1} = \frac{f \cdot x^c}{S_x \cdot z^c} + C_{x1} \quad (5-10)$$

$$c_2 = \frac{u_2}{S_x} + C_{x2} = \frac{f \cdot (x^c - b)}{S_x \cdot z^c} + C_{x2} \quad (5-11)$$

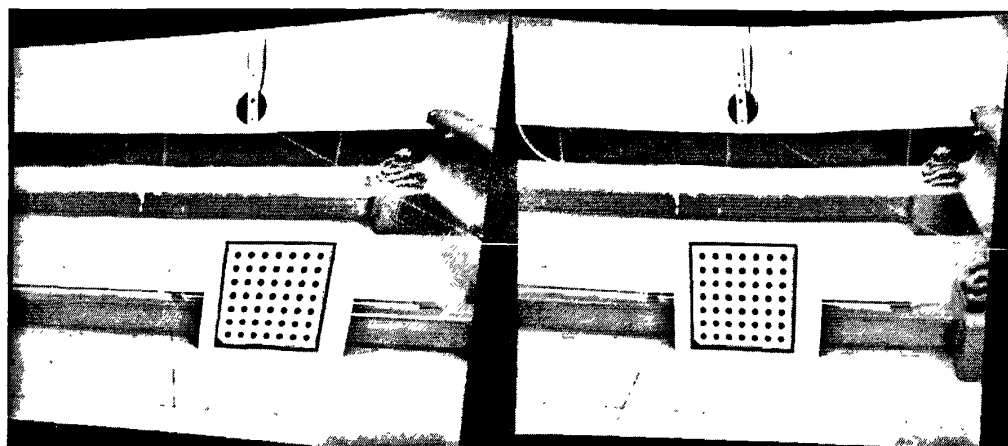
则将两式相减可得:

$$d' = c_1 - c_2 = \frac{f \cdot b}{S_x \cdot z^c} + C_{x1} - C_{x2} \quad (5-12)$$

从而得到:

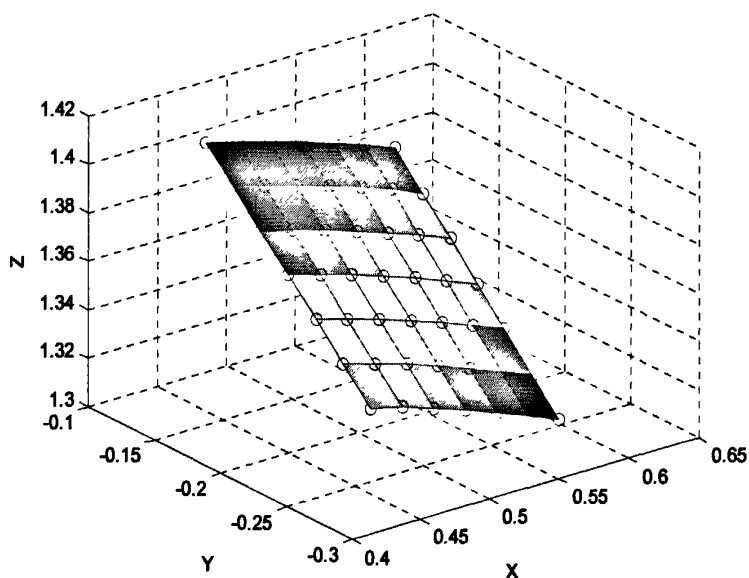
$$z^c = \frac{f \cdot b}{S_x(d' - C_{x1} + C_{x2})} \quad (5-13)$$

这样我们就建立了视差和距离的关系，进而可以求得空间点的三维坐标。为了验证标定以及三维重建的精度，我们依然利用标定板上的点作为基准。我们采用和 5.2 节相同的方法找到图 5-8 (c) 和 (d) 中的标定板并得到标定板上各点的坐标，如图 5-10 (a)，(b) 所示，然后计算各点的视差，最后计算标定板上 49 个点的三维坐标，并利用 Matlab 画出其空间位置，如图 5-10 (c) 所示。



(a)

(b)



(c)

图 5-10 三维重建结果

Fig. 5-10 Result of 3-D information reconstructure

图 5-10 中黑色圆圈表示的就是求得的三维坐标，这些坐标都是以左相机的相机坐标系为基准的。若将这些点连成面后，可以看到这些点在空间上的相对位置与实际标定板上的位置十分吻合。为了进一步验证精度我们计算了相邻两个点的

距离，得到这些距离的平均值为：24.93mm，标准差为：0.0157mm，而实际标定板上点的间距为 25mm。

5.4 基于三维信息的识别方法

前面我们介绍了通过双目立体视觉获得物体空间三维信息的方法，在得到物体的三维空间位置后，我们就可以利用其来进行识别了。本节将介绍本系统中，如何使用三维信息来进行线路环境状态的识别。通常的三维识别方法，是对第一幅图片的每一个点求视差，然后重建每一点的空间位置，这种方法被称为密集型重建算法。这种方法可以得到大量的三维数据，但是需要消耗大量的运算时间，在本系统中这种方法是不可行的。

在前面我们已经提到过我们采用了二维识别和三维信息相结合的方法，其基本思路的示意图如图 5-11 所示。首先左相机的图像被用来进行二维识别，其具体方法我们已经在第 4 章进行了介绍，得到可疑的异常特征，这是得到的可疑特征可能是正常线路设备的干扰；为了减少误报我们由立体视觉的方法得到可疑特征内点的三维信息，根据特征的三维空间位置，就可以准确地判断其是否为线路环境状态的异常。这样我们就可以只对有异常特征的图像，进行三维重建，并且只需重建异常特征内的点就可以了，这样便大大节省了运算时间。

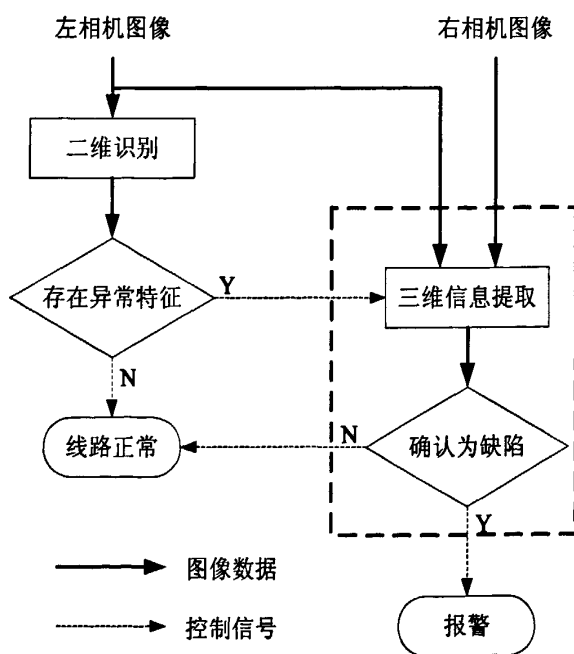


图 5-11 三维识别示意图

Fig. 5-11 Schematic diagram of 3-D recognition

6 总结与展望

前面几章介绍了本文的具体工作，包括高速铁路线路环境状态智能检测系统的总体设计、基于二维和三维视觉的线路环境状态识别以及图像预处理的硬件实现等。本章将对本文完成的主要工作进行一个总结，并对部分研究问题进行展望。

6.1 工作总结

本文主要完成了以下几个方面的工作：

(1) 在充分调研国内外研究现状，以及实地考察的基础上，完成了高速铁路线路环境状态智能检测系统的总体设计。分析研究了系统各器件，特别是成像系统的选型及安装问题。在实验室搭建了系统的原形机，实现了各项功能，处理速度主要性能达到了预期的目标。

(2) 提出了基于二维图像的线路环境识别方法，实现了包括基于灰度特征和边缘特征的护栏完整性识别、基于混合高斯模型的空间异物等。利用 HALCON 以及 Matlab 等环境对算法进行了测试，分析了识别算法的原理，特点和识别效果等问题。

(3) 提出了基于三维视觉的线路环境识别方法，该方法对由二维识别方法提取的异常特征进行三维识别，从而提高了识别的可靠性，降低了误判率。在实验室建立了立体视觉系统的原型机，进行了双目视觉的标定。实现了物体三维信息的重建。

(4) 本文创新性地提出了基于图像代数的图像处理算法硬件实现技术，设计并实现了部分图像代数基本元素和运算的硬件。并以此为基础实现了部分典型图像处理算法。所有的硬件均在 Xilinx ISE 和 Modelsim 环境下进行了仿真，并在 Xilinx Virtex-II Pro 系列 FPGA 上进行了测试。

6.2 展望

本文对高速铁路线路环境状态的智能识别的相关技术进行了研究，在实验室搭建了实验环境，初步具备了应用条件。但是由于时间的关系，某些方面的工作还存在一定的不足，今后的研究工作可重点从这些方面展开。

(1) 基于图像代数的图像处理算法硬件实现技术方面：本文提出该方法的初衷是想建立起一种通用的图像处理算法硬件实现方法，以解决当前图像处理硬件

设计困难、设计周期长等问题。但是由于时间有限，本文只进行了一定的探索性研究，实现了部分图像代数元素和运算的硬件设计，并利用其在 FPGA 上实现了本文所需要的几种图像处理算法。未来可这方面可以进行更深入的研究工作，研究复杂图像代数运算的硬件实现，以便建立和扩充起通用的硬件库供用户调用，或设计专门软件根据图像代数描述自动生成所需硬件等。另外，在硬件实现的资源消耗、处理速度等的优化方面可以进行深入的研究。

(2) 虽然目前的识别算法在原理和测试结果上都是可行的，但是由于图像数据相对缺乏，其可靠性还无法得到完整的测试，算法中部分参数的设定还无法进行最优化选择。今后在现场实验和实际运行阶段，随着大量数据的积累，应根据运行阶段对部分算法进行修改。

参考文献

- [1] 钱仲侯. 高速铁路概论[M]. 北京: 中国铁道出版社, 2006.
- [2] 龚增进, 余祖俊. 铁路综合检测列车采集分析系统总体设计方案[J]. 铁路计算机应用, 2005, 14(12): 27-29.
- [3] 邓学通, 叶一鸣. 准高速轨检车检测原理及应用[M]. 北京: 中国铁道出版社, 2004.
- [4] 许贵阳. 客运专线综合检测列车技术方案的研究和应用[J]. 铁道建筑, 2008, 2: 95-98.
- [5] 黎国清, 许贵阳. 高速综合检测列车技术交流总结[R]. 北京: 中国铁道科学研究院, 2004.
- [6] ENSCO, VisiRail Track Inspection Systems[EB]. Springfield, Illinois: ENSCO Inc, 2005.
- [7] Carsten Steger, Markus Ulrich, Christian Wiedemann. 机器视觉算法与应用[M]. 杨少荣, 吴迪靖, 段德山译. 北京: 清华大学出版社, 2008.
- [8] Milan Sonka, Vaclav Hlavac, Roger Boyle. Image Processing, Analysis, and Machine Vision, second edition[M]. Beijing: Post&Telecommunication Press, 2002.
- [9] MVTec. HALCON Solution Guide II-F: on 3D Machine Vision[ED/CD]. Munchen, Germany: MVTec Software GmbH, 2008.
- [10] Reinhard Klette, Karsten Schluns, Andreas Koschan. Computer Vision: Three-dimensional Data form Images[M]. Singapore: Springer-Verlag Singapore Pte. Ltd, 1998.
- [11] 安连生. 应用光学[M]. 第三版. 北京: 北京理工大学出版社, 2007.
- [12] 中华人民共和国铁道部. 新建时速 300~350 公里客运专线铁路设计暂行规定[S]. 北京: 中国铁道出版社, 2007.
- [13] Coreco Imaging. Anaconda-CL User's Manual[EB/CD]. St-Laurent, Canada: Coreco Inc., 2004.
- [14] E. R. Davies. Machine Vision: Theory, Algorithms, Practicalities[M]. 3th ed. Beijing: Post&Telecommunication Press, 2009.
- [15] Coreco Imaging. Anaconda-CL Developer's Manual[EB/CD]. St-Laurent, Canada: Coreco Inc., 2005.
- [16] Anthony E N. Implementation of image processing algorithms on FPGA hardware[D]. Nashville: Graduate School of Vanderbilt University, 2000.
- [17] 仙云森. 基于 FPGA 的图像处理算法研究及硬件设计[D]. 大连: 大连理工大学, 2007.
- [18] Wang Yao, Yu Zujun, Zhu Liqiang. Position and Attitude Measurement for Railway Vehicles Applied in Railway Infrastructure Inspection[C]. Proceeding of the Third International Symposium on Test Automation & Instrumentation, Vol.4, 2010: 1316-1321.
- [19] G. X. Ritter, J. N. Wilson, J. L. Davidson. Image algebra: An overview[J]. Computer Vision, Graphics, and Image Processing, 1990, 49(3): 297-331.
- [20] Gerhard X. Ritter, Joseph N. Wilson. Handbook of Computer Vision Algorithms in Image Algebra[M]. 2nd ed. New York: CRC Press LLC, 2001.
- [21] 章毓晋. 图像工程(中册): 图像分析[M]. 第 2 版. 北京: 清华大学出版社, 2005.
- [22] G. X. Ritter, P. D. Gader. Image algebra techniques for parallel im-age processing[J]. Journal of Parallel and Distributed Computing, 1987, 4(1): 7-44.
- [23] 董开坤, 胡铭曾, 方滨兴等. 一个基于图像代数的并行图像处理环境[J]. 计算机研究与发
展, 2004, 41(1): 201-206.

- [24] K. Benkrid, A. Benkrid, S. Belkacemi. Efficient FPGA hardware development: A multi-language approach[J]. *Journal of Systems Architecture*, 2007(53): 184-209.
- [25] A. Bouridane, D. Crookes, P. Donachy, et al. A high level FPGA-based abstract machine for image processing[J]. *Journal of Systems Architecture*, 1999(45): 809-824.
- [26] Volnei A. Pedroni. VHDL 数字电路设计教程[M]. 乔庐峰,王志功等译. 北京: 电子工业出版社, 2005.
- [27] Pong P. Chu. RTL hardware design using VHDL: Coding for Efficiency, Portability, and Scalability[M]. Hoboken: John Wiley & Sons, Inc., 2006.
- [28] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins. Digital Image Processing Using MATLAB[M]. Beijing: Publishing House of Electronics Industry, 2005.
- [29] Sergios Theodoridis, Konstantinos Koutroumbas. Pattern Recognition[M]. 3th ed. Beijing: China Machine Press, 2006.
- [30] 阮秋琦. 数字图像处理学[M]. 北京: 电子工业出版社, 2007.
- [31] Rafael C. Conzalez, Richard E. Woods. Digital Image Processing[M]. Beijing: Publishing House of Electronics Industry, 2002.
- [32] Canny J F. A Computational Approach to Edge Detection[J]. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1986, 8(6): 679-698.
- [33] Haralick R M. Digital Step Edges from Zero Crossing of Second Directional Derivatives[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligene*, 1984, 6(1): 58-68.
- [34] 张颖翔. 视频场景中的运动分析与处理[D]. 上海: 上海交通大学, 2009.
- [35] 傅昌宁. 基于高斯混合模型的图像序列运动目标检测技术研究[D], 武汉: 武汉理工大学, 2009.
- [36] D S Lee, J Hull, B Erol. A Bayesian framework for Gaussian mixture background modeling[J]. *IEEE Proc. ICIP. Vol.3*, 2003: 973-976.
- [37] Stauffer C, Crimson W. Adaptive background mixture models for real-time tracking[C]. In: *Proc IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, 1999: 246-252.
- [38] Stauffer C, Grimson W Learning patterns of activity using real-time tracking[C]. In: *Proc IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2000. 22(8): 747-757.
- [39] 许洪伟. 智能视频监控关键技术研究[D], 北京: 北京交通大学, 2009.
- [40] 吴立德. 计算机视觉[M]. 上海: 复旦大学出版社, 1993.
- [41] MVTec. HALCON HDevelop User's Guide[EB/CD]. Munchen, Germany: MVTec Software GmbH, 2008.
- [42] 张宏. 基于双目立体视觉的三维重建技术研究[D]. 武汉: 华中科技大学, 2007.
- [43] 闫龙. 便携型立体视觉测量系统相关技术研究[D]. 济南: 山东大学, 2008.
- [44] D. Conte, P. Foggia, C. Sansone, M. Vento. Thirty Years of Graph Matching in Patten Recognition[J]. *International Journal of Pattern Recognition and Artificial Intelligence*, 2004, 18(3): 265-298.
- [45] 李介谷. 计算机视觉的理论和实践[M]. 上海: 上海交通大学出版社, 1992.
- [46] 章毓晋. 中国图像工程及当前的几个研究热点[J]. *计算机辅助设计与图形学*, 2002, 14(6): 489-500.

- [47] D.Marr. 视觉计算理论[M]. 姚国正, 刘磊, 汪云九译. 北京: 科学出版社, 1988.
- [48] AIA. Specifications of the Camera Link Interface Standard for Digital Cameras and Frame Grabbers Camera Link Specifications Version 1.1[S]. Automated Imaging Association, 2004.
- [49] Casells V, Lisani J L, Morel J M et al . Shape Preserving Local Histogram Modification[J]. IEEE Trans. Image Processing, 1999, 8(2): 220-230.
- [50] DALSA. Anaconda Data Sheet[EB/CD]. Montreal: Dalsa Corporation, 2007.
- [51] Xilinx. Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet. Xilinx, 2007.

作者简历

王尧，男，浙江宁波人，出生于1986年6月。

2004年9月—2008年7月，北京交通大学机械与电子控制工程学院测控技术与仪器专业攻读学士学位；

2008年9月—2010年7月，北京交通大学机械与电子控制工程学院检测技术与自动化装置专业攻读硕士学位；


攻读硕士学位期间参与完成863计划“轨道交通基础设施全断面动态检测技术研究”，“863”计划“最高试验速度400km/h高速检测列车关键技术研究与装备研制”等科研项目。

发表论文：

[1] Wang Yao, Yu Zujun, Zhu Liqiang. Position and Attitude Measurement for Railway Vehicles Applied in Railway Infrastructure Inspection[C]. Proceeding of the Third International Symposium on Test Automation & Instrumentation, Vol.4, 2010, 1316-1321.

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：  签字日期： 2010 年 6 月 25 日

学位论文数据集

表 1.1: 数据集页

关键词*	密级*	中图分类号*	UDC	论文资助
基础设施检测、 机器视觉、图像 代数、FPGA	公开	TP391.4, U216.3	004.93, 656.2	“863”计划
学位授予单位名称*		学位授予单位代码*	学位类别*	学位级别*
北京交通大学		10004	工学	硕士
论文题名*		并列题名		论文语种*
基于图像的高速铁路线路环境状 态智能识别技术研究				中文
作者姓名*	王尧		学号*	08121938
培养单位名称*		培养单位代码*	培养单位地址	邮编
北京交通大学		10004	北京市海淀区西 直门外上园村 3 号	100044
学科专业*		研究方向*	学制*	学位授予年*
检测技术与自动化装置		智能化测控技术	2年	2010
论文提交日期*	2010年6月			
导师姓名*	余祖俊		职称*	教授
评阅人	答辩委员会主席*		答辩委员会成员	
	李长春			
电子版论文提交格式 文本 () 图像 () 视频 () 音频 () 多媒体 () 其他 () 推荐格式: application/msword; application/pdf				
电子版论文出版(发布)者		电子版论文出版(发布)地		权限声明
论文总页数*	91			
共 33 项, 其中带*为必填数据, 为 22 项。				

