



**The Research of Moving Object Detection and Tracking System and
its Implementation on DSP**

by

LIANG Laipeng

B.E. (Hunan University)2006

A thesis submitted in partial satisfaction of the

Requirements for the degree of

Master of Engineering

in

Electronics Science and Technology

in the

Graduate School

of

Hunan University

Supervisor

Professor LI Fuhai

April,2011

湖南大学

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名：梁来翔

日期：2011年5月27日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权湖南大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

1、保密□，在_____年解密后适用本授权书。

2、不保密□。

(请在以上相应方框内打“√”)

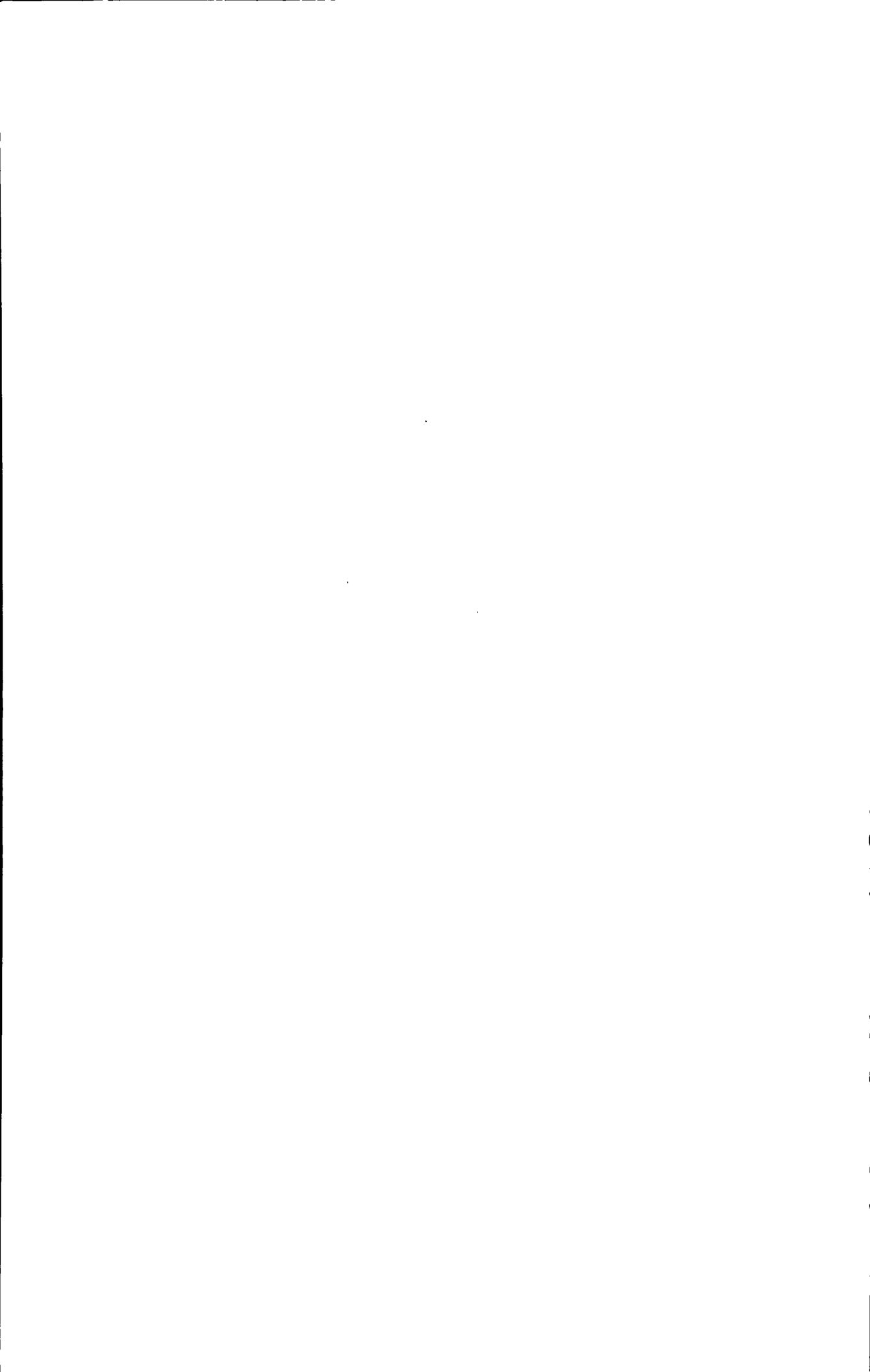
作者签名：梁来翔

日期：2011年5月27日

导师签名：

梁来翔

日期：2011年5月27日



摘 要

目前, 视频图像监控系统已经在各个方面的得到广泛应用, 对运动目标的检测与跟踪是视频监控系统中重要组成部分。运动目标检测可以判断是否存在运动目标, 并确定其位置, 运动目标跟踪可以监控运动目标的时空变化。构建一个实时、稳定和有效地运动目标检测与跟踪系统, 具有重要的理论价值和实际工程意义。

对运动目标的检测与跟踪是智能视频监控系统的的首要步骤, 在此基础上, 可进一步实现智能识别、行为理解等更高层次的视频图像处理。针对视频图像处理工作实时性较强和数据运算繁琐的特点, 本文对视频监控系统中目标检测与跟踪系统进行了深入研究和分析, 并设计了一个基于 Blackfin DSP 的运动目标检测与跟踪系统。

首先对当前常用的运动目标检测与跟踪算法进行了分析比较, 重点探讨了帧间差分法检测算法和均值偏移跟踪算法的处理流程。从总体功能和结构出发, 设计了一个基于 Blackfin DSP 的目标检测与跟踪系统硬件平台, 包括通讯控制模块、视频输入、输出模块和外部存储等模块的设计; 接着对系统进行了硬件调试, 包括时钟配置、DMA 传输配置及 I²C 总线配置, 重点介绍了利用 Blackfin533 的 PPI 接口, 实现硬件板与调试 PC 以及 DSP 与 FPGA 子系统的通讯控制模块配置, 给出调试结果。然后利用 Visual DSP++ 开发环境, 对基于帧间差分法运动目标检测算法与 Meanshift 运动目标跟踪算法进行建模与测试, 给出程序优化的思想和方法; 最后根据测试实验结果, 说明各个模块能够实现相应功能, 系统达到了设计要求。

关键词: BLACKFIN; 目标检测; 目标跟踪; 帧间差分法; MEANSHIFT

Abstract

At present, the video surveillance system has been widely used in various aspects, and moving target detection and tracking is an important part of it. Object detection involves verifying the presence of an object and possibly locating it precisely. Object tracking is to monitor an object's spatial and temporal changes. It is of great theory value and project meaning to realize a real-time, stable and effective Object detection and tracking system.

Object detection and tracking is the first step of the intelligent video surveillance system, which is the basis of Intelligent Recognition and other high-level behavior understanding of motion objects. To meet the real-time requirement and the large volume of data processing in monitoring tasks, we make a in-depth research and discussion in object detection and tracing of video surveillance system, and design a Blackfin DSP-based moving object detection and tracing system.

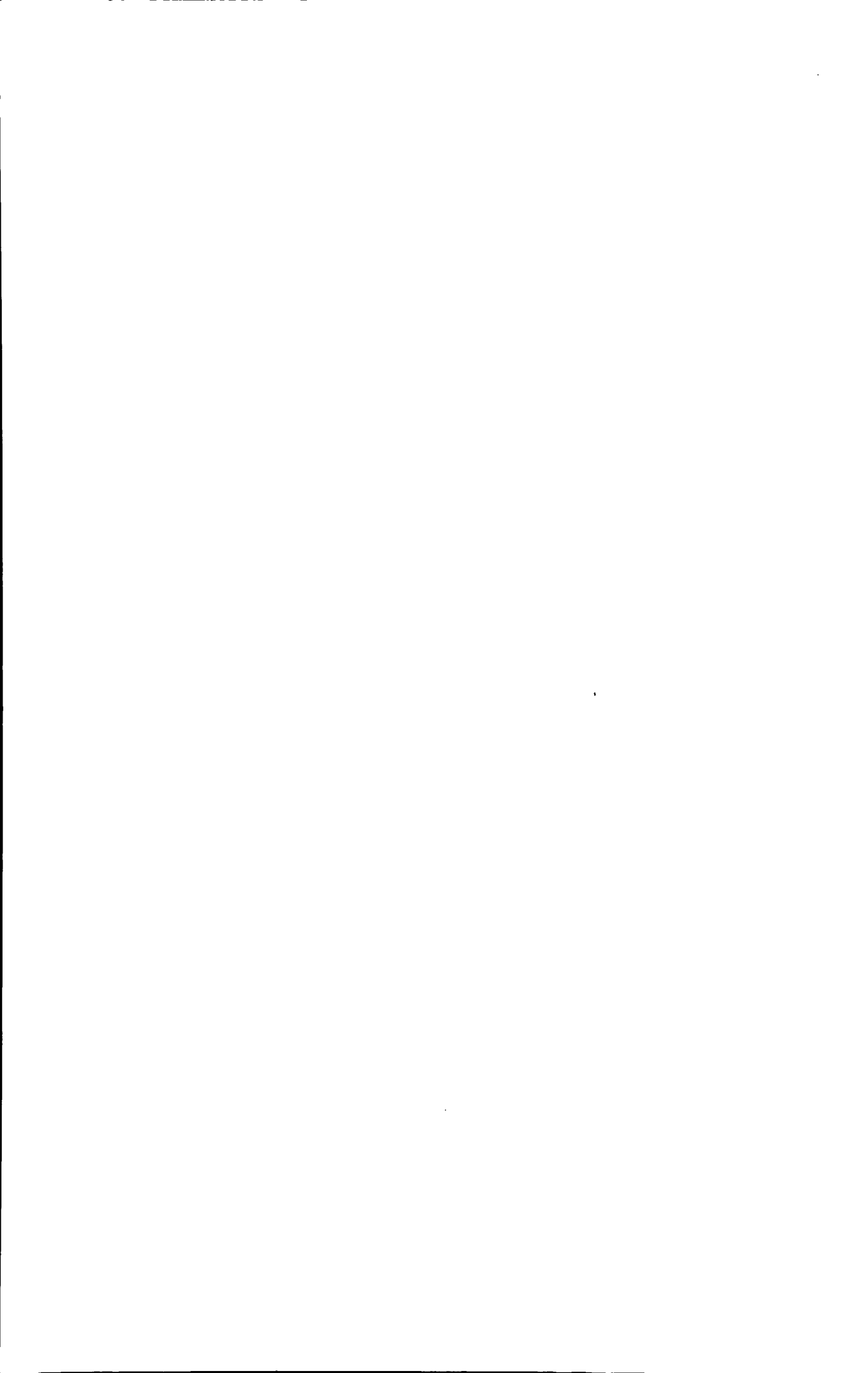
First of all, several commonly used moving object detection and tracking algorithms are discussed and compared, focusing on the frame difference method and the Meanshift method. Secondly, a complete hardware platform for the Blackfin DSP-based video processing system is designed, based on the overall hardware structure and the function of the system, including the communication and control module, video input and output module and external memory module, etc. Thirdly, the hardware platform is configured and tested, including the the clock circuit, the DMA transmission and the I²C bus, focusing on how to use the PPI port to communicate the DSP board with PC and the FPGA subsystem, the test results are given. Then, using the tool of Visual DSP++, a model of frame difference method and Meanshift method is built and tested, including how to optimize the program. The results of the test show that the corresponding functions can be achieved, and the design requirements can be met.

KeyWords: BLACKFIN; Moving Object Detection; Moving Object Tracing;
Frame Difference Method; Meanshift

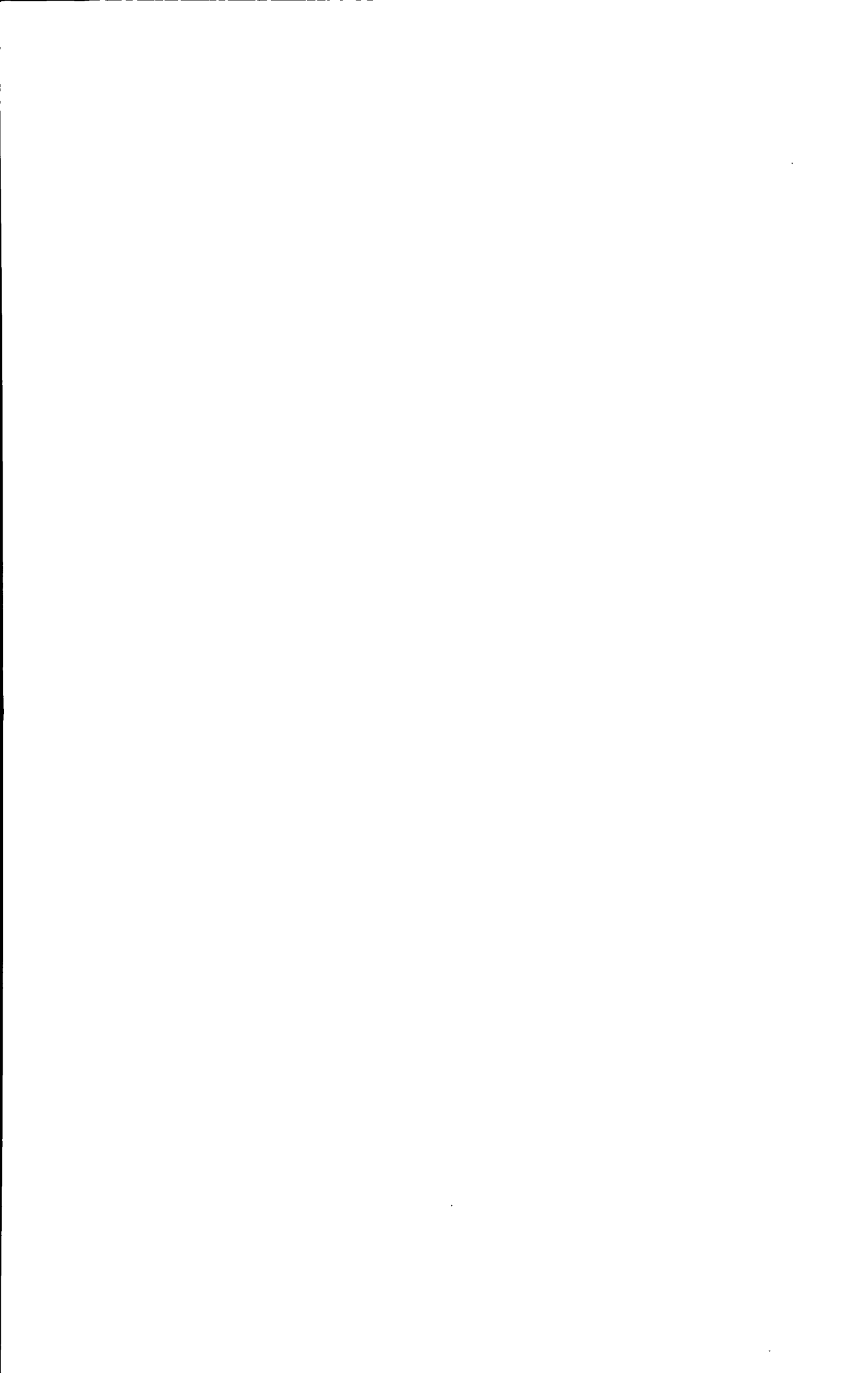


目 录

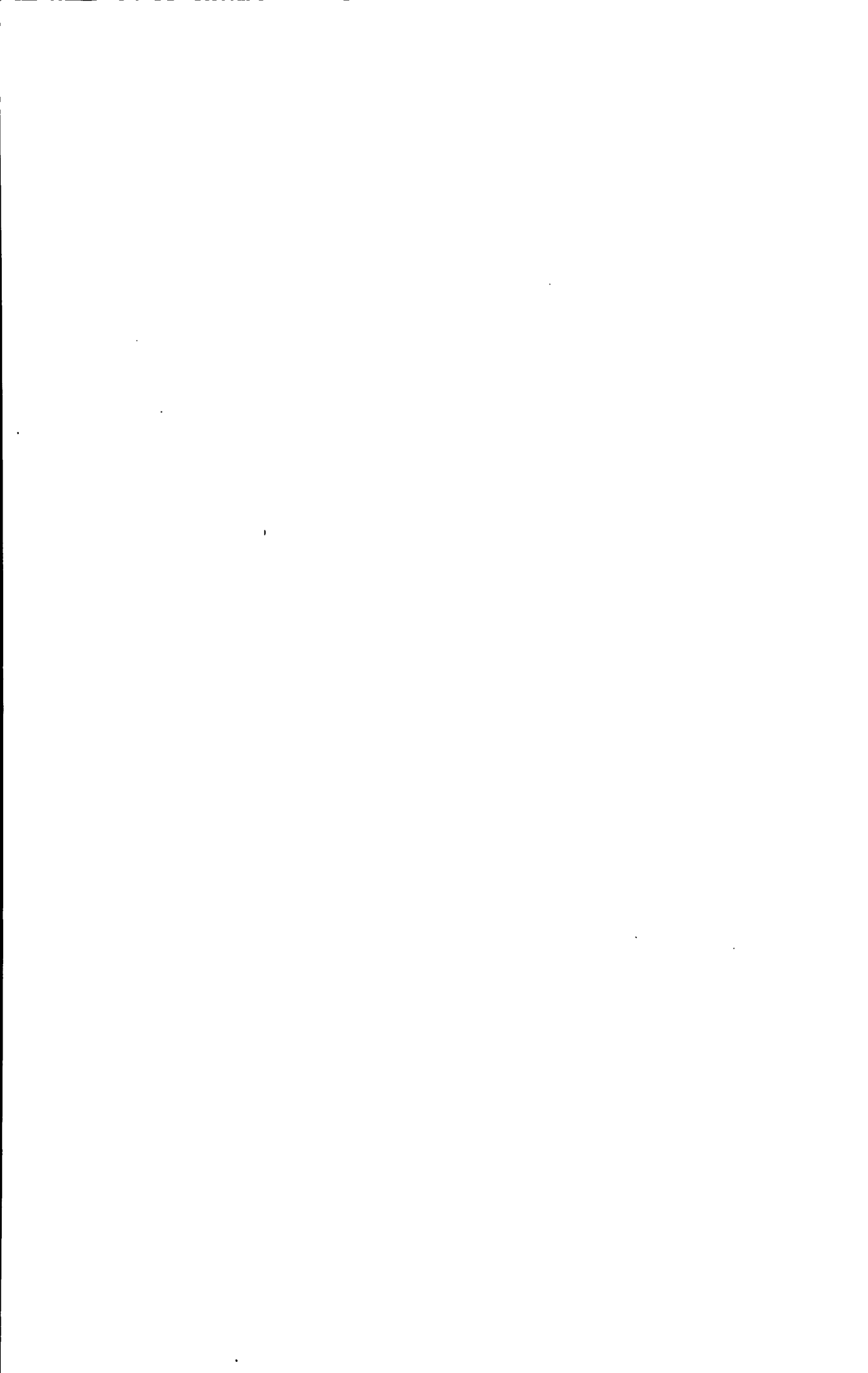
学位论文原创性声明和学位论文版权使用授权书.....	I
摘要	II
Abstract.....	III
第 1 章 绪论	1
1.1 课题背景及意义	1
1.2 国内外研究现状	1
1.2.1 多媒体处理器技术现状	1
1.2.2 运动目标检测、跟踪算法研究现状	3
1.3 本文主要内容	5
第 2 章 运动目标检测与跟踪算法研究与设计	6
2.1 引言	6
2.1.1 颜色空间概述	6
2.1.2 图像处理的研究	7
2.2 运动目标检测算法研究	8
2.2.1 目标检测算法分类	8
2.2.2 目标检测算法分析与比较	10
2.2.3 帧间差分法设计	11
2.3 运动目标跟踪算法研究	12
2.3.1 运动目标跟踪基本过程	12
2.3.2 Meanshift 目标跟踪算法设计	14
2.4 本章小结	21
第 3 章 运动目标检测与跟踪系统硬件设计	22
3.1 系统整体设计	22
3.2 系统处理模块设计	23
3.2.1 Blackfin533 DSP	23
3.2.2 FPGA 特性及其基本结构	25
3.3 通信控制模块设计	25
3.3.1 PC 机与板卡连接设计	25
3.3.2 BF533 与 FPGA 连接设计	26



3.4 视频输入、输出模块设计	26
3.4.1 视频输入模块	26
3.4.2 视频输出模块	27
3.5 外部存储模块设计	28
3.5.1 SDRAM 模块设计	29
3.5.2 FLASH 模块设计	30
3.6 调试接口模块设计	31
3.7 电源电路设计	32
3.8 本章小结	32
第 4 章 运动目标检测与跟踪系统硬件调试	33
4.1 初始化与时钟配置	33
4.1.1 硬件系统初始化	33
4.1.2 时钟频率配置	33
4.2 通信与控制	35
4.2.1 PPI 接口配置	35
4.2.2 DMA 传输配置	37
4.3 视频采集、存储与输出	38
4.3.1 I ² C 总线配置	38
4.3.2 视频采集电路	40
4.3.3 视频数据存储	42
4.3.4 视频输出电路	42
4.4 本章小结	44
第 5 章 运动目标检测与跟踪系统软件设计	45
5.1 软件开发环境与流程	45
5.1.1 软件开发环境	45
5.1.2 软件开发流程	45
5.2 算法软件设计	46
5.2.1 系统初始化	46
5.2.2 图像平滑	47
5.2.3 目标检测	48
5.2.3 目标跟踪	50
5.2.4 优化原则与方法	53
5.3 本章小结	55
总结和展望	56



参考文献.....	58
致谢	61
附录 A 攻读学位期间所发表的学术论文	62
附录 B 电路原理图	63



第1章 绪论

1.1 课题背景及意义

随着计算机技术与现代通信技术的迅速发展,视频监控在社会生活和军事各方面得到了非常广泛的应用,如银行、商场、车站和交通路口等公共场所的监控,但实际的监控任务仍需较多的人力配合完成,且现有视频监控系统所提供的视频或图像文件,通常是没有经过处理和识别的,只能用作事后取证,无法充分发挥监控的实时性和主动性^[1]。为了能实时分析、跟踪、判别监控对象,智能视频监控的重要性就凸现出来。智能视频监控的本质是采用机器视觉处理技术对视频图像信号进行处理和高层次分析,无需人为干涉,即可对视频帧中的目标进行检测、定位与跟踪,并进一步对目标进行行为分析和行为判断,以及自动报警和智能存储。

智能视频监控中运动目标的检测与跟踪涉及信号传输、视频图像处理、计算机视觉等多个研究领域。目标检测是指从视频流中提取出目标,如确定目标的颜色或形状、区域等信息,它必须从连续的视频流或图像序列中提取目标^[2]。运动目标跟踪则是对目标进行连续的跟踪处理以确定其运动轨迹,其关键在于得到图像检测的静态目标与被跟踪的运动目标的对应关系。该课题的研究成果在交通监视、商场银行监控、超市客流统计等领域都有重要的实用价值。随着图像处理识别技术的进步和软硬件性能的提高,智能化的目标检测与跟踪系统必将在更多领域得到广泛的应用。

1.2 国内外研究现状

1.2.1 多媒体处理器技术现状

随着数字视频图像处理技术及数据通信技术的飞速发展,视频图像处理系统的应用越来越广泛^[3]。目前视频图像处理系统从处理平台上大致可分为两类:PC方式及嵌入式方式。前者是后者的基础,主要通过PC机上运行一定的软件来实现,相对来说简单易行、成本低廉,但实时性较差;后者则是在嵌入式系统中运行视频图像处理程序及通信协议,实现实时视频处理。

嵌入式系统发展了半个多世纪,现在已经应用于不同领域的各种系统。嵌入式系统以是PC技术为基础,以应用为中心,且软硬件可裁减,能较好满足应用系统对功能性、可靠性以及成本、体积、功耗等指标要求的专用计算机系统。嵌入式系统可依据处理器的位数、主频和应用领域等标准分类,位数和主频反映其计算能力^[4]。根据嵌入式系统的应用领域,通常分类如下:

(1) 基于单片机(如MCS-51、96等)的简单控制系统。该类嵌入式系统集成众多外围接口,可以方便地进行系统设计,适合于仪表和控制领域。不足之处是计算能力不强,难以适用于复杂的信号处理领域。

(2) 基于微处理器的嵌入式系统。包括ARM、POWERPC、MIPS和68K系列等,此类嵌入式系统外围接口丰富,且能运行微型操作系统。因可移植操作系统于其上,故有丰富的软件资源可利用,适用网络通信和人机交互等软件复杂的领域,优点是可开发出具有良好人机交互界面的复杂系统,同时可以很好的利用已有的软件资源,不足之处是信号处理能力不如DSP。

(3) 基于DSP的数字信号处理系统。与单片机系统相比,DSP系统具有许多更加适于数字信号处理的优点:普遍采用改进的哈佛总线结构,内部有硬件乘法器、累加器,使用多级流水线结构,采用并行处理,并有专门设计的适于数字信号处理的指令系统,可进行多片DSP并行处理。此类系统主要用于移动基站等数据量大且实时性要求较高的领域。

数字视频图像处理系统需要对图像进行实时分析、压缩编码、解码等大量的运算处理,利用DSP作为视频处理的嵌入式平台,可以发挥其性能优势,在一定程度上满足图像实时处理的要求^[5]。首先,用户开发自由度更大,支持多种个性化开发;其次,DSP处理能力强,可在一片DSP上同时运行多路音视频信号的压缩处理;第三,开发周期短,可快速实现技术更新;第四,芯片功耗低,对提高产品的稳定性和续航能力提供可靠保障,可满足便携式产品的要求。

此外,基于FFGA的图像处理系统,近年来也有较大发展^[6],其处理速度较快,灵活性较强,因设计较复杂,成本较高,在高端的先进视频编码网络摄像机有采用。媒体处理器既具备通用DSP的优点,提供了一系列多媒体设备实现工具,又针对多媒体处理增加了各种接口功能,优化了CPU。

正是出于这种考虑,一批新型的DSP芯片,即媒体处理器(Media Processor)应运而生,并很快得到业界的广泛关注。飞利浦、ADI、TI等半导体供应商都相继推出了颇具竞争力的产品。例如TI的TS320DM642、飞利浦的PNx1500和ADI的Blackfin系列等,它们集成了丰富的多媒体信号接口,针对多媒体信号的特点优化了处理器结构,甚至内嵌各种特殊协处理器用于音视频的专门处理,大大缓解了通用处理器的压力。

ADI的Blackfin系列芯片的基础是ADI(Analog Deivices)和Intel公司联合开发的微信号架构(MSA),它将一个32位的RISC型指令集和两个16位乘法累加(MAC)信号处理功能与易用性结合起来,诸如监督和用户模式、存储保护、8/16/32位算术及多媒体处理扩展。Blackfin DSP既能用作音视频处理器,还能用作图形用户界面(GUI)和网络处理的处理器。Blackfin DSP架构经过优化,能够在信号处理和控制处理应用中都很好发挥作作用,可以同时用作主处理器和音视频处理器。Blackfin

是完全可编程的, 无需改变硬件平台就可以很容易地支持新标准。如ADI可视电话平台中的Blackfin DSP通过软件升级就能够支持H.264。开发人员只需要掌握同一种编程方法、指令系统、和开发环境, 这样就可大大缩短产品面市时间, 削减开发成本, 降低开发风险, 提高IP复用性等^[7]。

综上所述, 通用DSP运算速度快, 处理能力强, 能满足一般性实时图像处理的需要, 而在视频处理方面, 通常要求处理器具有针对多媒体应用的、优化的内核结构和丰富的媒体接口, 通用DSP在这方面难以充分发挥性能。故本系统采用ADI公司的Blackfin系列DSP作为核心来设计实现运动目标检测与跟踪系统。

1.2.2 运动目标检测、跟踪算法研究现状

智能视频监控的关键部分是对特定目标的自动检测与跟踪。图1.1为运动目标检测与跟踪系统原理框图, 目标检测与跟踪大致可分为背景处理、运动检测、目标匹配和轨迹跟踪等步骤。运动检测是运动目标检测与跟踪的第一步, 从序列图像中将变化的区域提取出来, 技术难点为背景图像的不稳定性(遮挡、光线等)、慢移动和静移动等^[8]。目标跟踪是在目标检测的基础上, 在连续的图像帧间创建基于位置、形状、纹理、色彩等有关特征的对应匹配关系, 使用适当的匹配算法, 在序列图像中寻找与目标模板最相似的图像的位置, 将目标定位。

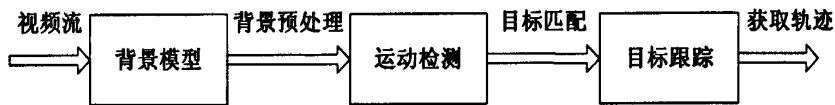


图1.1 运动目标检测与跟踪原理框图

1 运动目标检测算法

运动目标检测算法大致分为三种:

(1) 光流法(Optical Flow)。光流的概念是由 Gibson 于 1950 年首先提出来的, 其基本原理是给图像中的每一个像素分配一个速度矢量, 通过计算速度向量的光流场来初始化基于轮廓的检测算法, 从而有效地提取和跟踪运动目标^[9]。先给出图像的亮度模式, 再利用时间变化的特征点匹配度量视差, 以图像中的角点为特征点, 在图像序列中检测并跟踪角点, 记下角点在图像序列中的位置, 再通过相邻图像帧之间角点的位移场来计算光流场^[10]。该方法的优点是摄像机运动的情况下也能检测出独立的运动目标^[11]。然而大多数光流计算方法相当复杂, 抗噪性能差, 不适合应用于实时运动目标检测系统。国内不少学者对光流法进行了研究, 大都在理论阶段, 涉及实际应用的较少。

(2) 背景差分法。该方法是当前目标检测方法中最直接、最简单的一种, 通过对序列图像建立一个背景模型, 估算出当前最优背景图像, 利用当前图像与背景图像的差分来检测出运动区域, 利用运动目标和背景在灰度或色彩上的差别, 通

过将背景图像和当前图像做差分运算,相减的结果中每一像素的值和一个预先设定的阈值相比较,如果大于阈值,则认为该点是前景点,否则认为是背景点。背景差法在背景变化缓慢时能够检测出完整的运动像素,但对外部环境变化敏感,当光线突变或背景扰动时,检测效果明显下降,此时则需建立新的背景模型,因此背景模型的选取和更新对于背景差法来说十分重要^[12,13]。背景差分法需要时常更换背景,使得该算法在具体实现时需要消耗更多的资源^[14]。

(3) 帧间差分法。对于背景静止的运动目标检测,随着时间的变化运动目标发生位移而背景不变。位移体在视频帧中的体现就是发生运动的区域的灰度值产生了较大的变化,对这种变化进行检测就可以分析出目标的运动情况,通过对相邻帧进行差分就可以检测到这种变化。帧间差分法就是利用相邻两帧或多帧图像之间运动目标的差异来进行运动目标的检测和提取,每处理一帧图像就用处理的图像对上一帧图像进行更新,将一定时间内没有变化的像素置零,通常选取相邻的两帧或三帧作为参与运算的图像,其主要优点是计算简单、能够迅速检测出场景中的运动像素,不足之处是通常适用于静止背景,在动态背景下需要对摄像机进行运动补偿后再进行差分运算。

上述三种方法中,基于帧间差分法和背景差分法的运动目标检测算法研究最多,其中帧间差分法较之于背景差分法,计算更简单、处理速度更快,也更易于硬件实现。

2 运动目标跟踪算法

目前,运动目标跟踪算法主要有^[15]:

(1) 基于区域的跟踪,先初始化跟踪目标,再运用相关算法,在后续图像中跟踪目标。该算法的优点是性能稳定,目标未遮挡时跟踪精度高,缺点是目标变形或遮挡时跟踪精度下降。

(2) 基于特征的跟踪,与前者相比,该算法使用目标整体作为相关对象,主要优点是能解决目标部分遮挡的问题,特征容易捕捉,不足之处是处理重叠和非相关结构的干扰时能力较差。

(3) 基于活动轮廓的跟踪,该算法利用前几帧的先验形状信息建立下一帧的目标形状模型,可以从整体上识别物体,在物体变形、局部被遮挡的情况下,仍具有较强的鲁棒性,但对跟踪的初始值相当敏感,要求被跟踪物体具有清晰轮廓,自动跟踪初始化比较困难。

(4) 基于模板的跟踪,通过目标模型与图像数据间的匹配来跟踪目标,这种方法需要先验知识来描述目标模型。与其他跟踪方法相比,该算法在遮挡或其他运动目标干扰时鲁棒性较好,跟踪时只需建立模型,其结构、运动约束等先验知识都会自动融合,目标运动方向突变时,跟踪仍然有效。缺点是需要构造模型,使计算代价提高,且模型质量高低直接影响跟踪的精度和稳定性。

此外还有其它一些常用的目标跟踪方法，如基于粒子滤波的目标跟踪，基于卡尔曼滤波的目标跟踪，基于 Meanshift 的目标跟踪等。基于 Meanshift 的目标跟踪法是一种基于模板匹配的跟踪方法，因其简单、快速、无需参数和快速模式匹配的特性而得到广泛的应用^[16]。

1.3 本文主要内容

研究内容主要包括运动目标检测与跟踪算法的研究与设计，以及DSP硬件系统设计两部分。本文所设计的基于Blackfin533的视频处理系统，具有数据并行处理、后续升级方便等优势。算法设计方面，综合考虑目标检测与跟踪算法的运行速度、鲁棒性和准确性等因素，选择一种检测结果准确、消耗资源较少、运行快速的检测算法，以及一个简单、快速、无需参数和快速模式匹配的基于模板匹配的跟踪方法，并在DSP上实现。

本文共5章，每章内容如下：

第1章 绪论。简要介绍课题的选题背景和意义、研究现状以及本文主要内容。

第2章 运动目标检测与跟踪算法研究。对几种常用的运动目标检测算法进行比较分析，经比较后选择帧间差分法作为本系统的检测算法。对几种常用的运动目标跟踪算法进行研究分析，经比较分析选择基于Meanshift的运动目标跟踪算法作为本系统的跟踪算法。

第3章 运动目标检测与跟踪系统硬件设计。设计了基于Blackfin533的硬件整体框架，根据系统需求详细阐述系统各模块的器件选择与硬件构成，给出各个模块的硬件电路图。

第4章 运动目标检测与跟踪系统硬件调试。对各个模块进行调试，对各模块寄存器进行设置，并给出仿真结果。

第5章 运动目标检测与跟踪系统软件设计。介绍系统软件设计环境，提出程序优化的方法，给出软件设计框架和流程，通过分析各个步骤的功能完成了各部分软件实现，并给出处理效果图。

最后对本文所作研究工作进行总结，并指出进一步研究工作与展望。

第 2 章 运动目标检测与跟踪算法研究与设计

2.1 引言

计算机视觉是在数字图像处理的基础上发展起来的新兴学科，它从信息处理的层次研究视觉认知过程，以及视觉信息处理的计算理论与计算方法，包括视频图像特征提取、运动视觉、物体建模和图像识别等方面。

运动目标检测是视频监控系统自动化的第一步，作用是将变化区域从图像数据序列中提取出来，得到所需运动矢量，进而识别并跟踪运动目标。目标检测的准确性对视频图像各种后续处理诸如目标跟踪、目标分割和行为理解等的影响非常大，因此是视频监控系统研究中的一重要课题。然而由于天气、光线、影子等干扰的存在，使得目标检测变得困难。提高检测算法的实时性和鲁棒性是提高系统跟踪准确度和稳定性的关键，这就要求检测算法简单有效并适于硬件实现。

运动目标跟踪是计算机视觉研究的核心方向之一。与目标跟踪相关的应用领域有智能视频监控、人机交互、自动驾驶、视觉控制、农业自动化、医学图像和视觉重构等^[17]。简言之，运动目标跟踪就是从序列图像中找出感兴趣区域或运动目标在每帧图像中的位置序列。运动目标跟踪的目的在于通过对序列图像进行分析研究，计算出运动目标在连续帧图像中的位移、运动速度、运动目标的数量及运动目标的运动轨迹等运动参数，为更高层次的行为理解、运动分析提供依据。

2.1.1 颜色空间概述

目前常用的颜色空间主要包括：RGB, CMY, CMYK, HSL, HSV, CIEXYZ, CIE Lab和LCH, YUV, YIQ, RGB, Xerox Corporation YES, Kodak Photo CD YCC 等。

1 颜色空间的分类

按技术角度，颜色空间分为下三类：

(1) RGB/计算机图形颜色空间：主要用于计算机和电视机颜色显示系统，如 HSL, RGB, HIS和HSV等；

(2) XYZ/CIE颜色空间：国际照明委员会定义的颜色空间，通常用作颜色的基础度量方法，该颜色空间与设备无关，在科学计算中广泛应用，对不能直接相互转化的两个颜色空间，可利用该颜色空间作为过度，如CIE1931XYZ, LCH等颜色空间；

(3) YUV/电视系统颜色空间：因广播电视的需求而开发的颜色空间，通过压缩色彩信息来有效传输彩色电视图像，如YUV, YIQ等。

2 颜色空间的转换

颜色空间转换是根据颜色在不同颜色空间之间的映射关系，将一种颜色空间中的颜色转化到另一个已知条件下的颜色空间中去，目的是简化彩色规范。颜色的转换总是同颜色空间联系在一起的，颜色空间的转换需要一定的转换模型。

当前视频采集设备大多是基于RGB颜色模型的，而RGB颜色模型并不能非常精确地描述像素点的颜色信息，当图像是希望对比度的时候，RGB空间显然不能携带足够的颜色信息。此时就需要将RGB颜色与其他颜色空间进行转换。较常用的是与HSV颜色空间转换，转换公式为^[18]：

$$H_1 = \arccos \left\{ \frac{[(R-G)+(R-B)]/2}{[(R-B)^2+(R-B)(G-B)]^{1/2}} \right\} \quad (2.1)$$

当 $B \leq G$ 时， $H=H_1$ 。当 $B > G$ 时， $H=360^\circ-H_1$ 。

$$S = \frac{\text{Max}(R,G,B) - \text{Min}(R,G,B)}{\text{Max}(R,G,B)} \quad (2.2)$$

$$V = \frac{\text{Max}(R,G,B)}{255} \quad (2.3)$$

2.1.2 图像处理的研究

运动目标检测与跟踪的前期工作是图像的预处理，目的是最大限度地减小背景噪声及其它干扰对系统的影响，尽量消除无关信息，恢复有用信息，增强有用信息的可检测性以及尽量简化数据，从而增强图像增强，特征抽取，图像分割、匹配和识别的可靠性，提高目标检测与跟踪的精确性。对于视频图像处理系统来说，预处理应在尽量少增加系统负担的前提下提高视频图像的质量，抑制各种干扰和噪声。

1 图像增强

对比度定义为亮度最大值和最小值的比值，一般的图像采集都有一定的对比度范围，当对比度较小时，会出现细节分辨不清的情况，这种情况下，通过图像增强可改进视觉效果。

目前，常用的图像增强方法有直方图变换增强和灰度变换增强两种。灰度变换增强是把一个灰度级映射到另一个灰度级上，分为线性变换、分段线性变换及非线性变换三种。直方图变换为把灰度分布较窄的图像调整为灰度分布较宽或均匀的图像，从而增大视觉反差，使图像增强。

2 图像去噪

噪声过滤是图像预处理的一个重要的方法。产生噪声的原因很多，主要产生于图像的采集和传输过程中，如电子元器件的内部噪声、感光材料的热噪声、器件机械运动产生的抖动噪声等。常见的噪声包括椒盐噪声、脉冲噪声和高斯噪声

等。椒盐噪声主要是随机出现的黑白亮度值。脉冲噪声主要是随机出现的正脉冲噪声和负脉冲噪声。高斯噪声主要是由传感器引起的亮度分布服从高斯分布的噪声。这些噪声影响了图像质量，甚至淹没了图像特征，势必影响运动目标的检测。因此，在运动目标检测前，有必要对视频图像消息进行去噪预处理，这是图像处理中一个必不可少的环节。

图像去噪的工作也称图像平滑或滤波，目的在于消除图像中的干扰、提高图像质量、强化图像特征。一般的图像去噪是对噪声图像使用局部算子，即当要对某个图像像素进行平滑处理时，仅对它附近小邻域内像素进行处理。当前的图像去噪方法主要有空域的方法和频域的方法两类，空域的方法是用图像平滑模板对目标图像卷积处理，以减小噪声；频域方法是先图像变换，再用合适滤波器滤波，之后反变换得到去噪图像。

空域滤波主要包括平滑线性滤波、统计排序滤波及维纳滤波等。它们针对不同噪声有不同的去噪特性，如平滑线性滤波和维纳滤波可有效滤除高斯噪声，而统计排序滤波能有效滤除脉冲噪声和椒盐噪声。

2.2 运动目标检测算法研究

现有的运动目标检测方法通常先进行目标识别特征向量的提取，然后对目标运动进行预测^[19]，主要包括背景差分法、帧间差分法和光流法等。这些算法的实现方式和效果都有区别，下面对它们进行分析比较。

2.2.1 目标检测算法分类

1 基于运动场估计法

该类方法通过图像序列的时空相关性估计运动场，建立相邻帧的对应关系，再利用运动目标与背景运动形式的不同进行检测，主要是光流法等。光流(Optical Flow)，是由Gibson于1950年首先提出来的，光流法首先假设图像灰度的变化完全是由运动目标或背景的运动引起的，而目标和背景的灰度不随时间变化。当运动目标发生变化时，图像表面的灰度模式运动称为光流场。光流场计算基本公式最初是由美国学者Horn和Schunch于1980年提出的，它是图像亮度基本不变，且灰度变化也很小的假设下，进行图像检测的有效方法。

光流场计算基本方法：设 t 时刻点 (x, y) 的灰度值为 $P(x, y, t)$ ，在 $t + \Delta t$ 时刻，该点位置为 $(x + \Delta x, y + \Delta y)$ ，记该点的灰度值为 $P(x + \Delta x, y + \Delta y, t + \Delta t)$ ，假设

$$P(x + \Delta x, y + \Delta y, t + \Delta t) = P(x, y, t) \quad (2.4)$$

将上式左侧在点 (x, y, t) 用泰勒公式展开，得到

$$P(x, y, t) + \frac{\partial P}{\partial x} g \frac{\Delta x}{\Delta t} + \frac{\partial P}{\partial y} g \frac{\Delta y}{\Delta t} + \frac{\partial P}{\partial t} + O(dt^2) = P(x, y, t) \quad (2.5)$$

记
$$u(x, y, t) = \frac{\Delta x}{\Delta t} = \frac{dx}{dt} \quad (2.6)$$

$$v(x, y, t) = \frac{\Delta y}{\Delta t} = \frac{dy}{dt} \quad (2.7)$$

(2.5) 式中, $O(dt^2)$ 代表高阶项, 因高阶项可忽略, 即得到光流约束方程:

$$\frac{\partial P}{\partial x} gu + \frac{\partial P}{\partial y} gy + \frac{\partial P}{\partial t} = 0 \quad (2.8)$$

2 背景差分法

背景差分法是是目前较常应用的一种运动目标检测方法, 其基本原理是先构建一个完整的背景图像, 然后用当前帧图像与背景帧图像相减, 通过阈值分化差分结果来检测出运动目标^[20]。该方法适用于静止背景下的图像分割, 基于两个假设: 首先, 在静止背景下, 若外界光照条件不变, 序列图像中的背景保持不变。其次, 目标可视且灰度与背景灰度存在一定的对比度。基本的背景差分法流程图见图2.1。

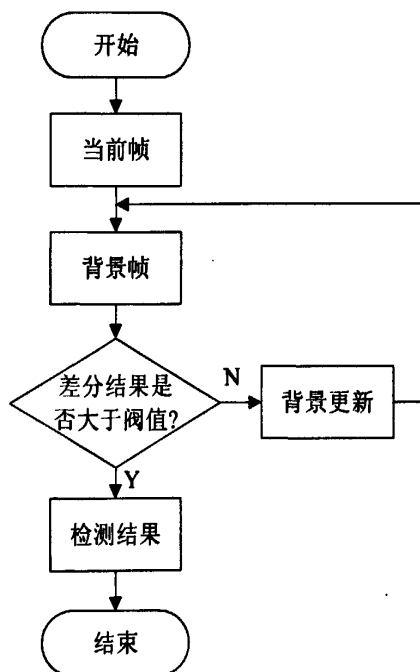


图2.1 基本背景差分法流程图

简单描述如下, 记 $B(x,y)$ 为背景图像, 记当前帧图像为 $F(x,y,t)$, 用当前图像与背景图像做差分运算, 得到掩膜图像为:

$$Fd(x,y,t) = |F(x,y,t) - B(x,y)| \quad (2.9)$$

设定一个阈值 T , 若 $Fd(x,y,t) \geq T$, 则 (x,y) 点属于运动目标, 反之则为背景。为实现长时间的视频监控, 背景差分法需要不断更新背景图像以减少动态场景变化

对目标检测的影响, 实际场景中往往伴随着光线的变化以及外界扰动, 一次背景差分法的难点不在与当前图像与背景图像的差分, 而在于背景模型的维持与更新 [21]。

3 帧间差分法

对于背景静止的运动目标检测来说, 随着时间的变化其背景图像不变只发生运动目标位移。位移反映视频帧中就是运动区域的灰度值发生了较大的变化, 检测到这种变化就可以分析目标的运动情况, 通过对相邻两帧或多帧做差分就可以检测到这种变化。帧间差分法也叫时间差分法就是利用图像序列相邻两帧或多帧, 做基于像素的时间差分, 通过阈值化提取图像中的运动区域。每处理一帧图像就用该帧图像对上一帧图像进行更新, 对更新的图像添加时间标志, 将一定时间内没有变化的像素置零。再对原有图像进行分割、边缘提取等处理, 以提取出运动目标区域, 完成目标检测。帧间差分法是实际运动目标检测应用最多的算法, 图 3.2 为帧间差分法的基本原理框图。

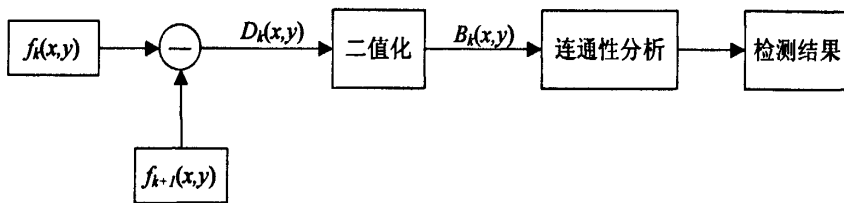


图 2.2 帧间差分法基本原理框图

2.2.2 目标检测算法分析与比较

光流法的优点是对目标的帧间运动限制较少, 可处理较大的帧间位移, 不仅携带了目标的运动信息, 还携带了景物三维结构的信息, 能够检测独立运动的目标, 在处理背景运动和遮挡问题上有很大的优势。但是该方法对光线十分敏感, 易受环境影响, 噪声、光线、天气等因素影; 计算必须利用图像序列的相邻两帧来实现, 如果某些目标暂时停止, 就很难把运动目标分离出来; 而且光流计算需要多次的迭代运算, 导致时间消耗较大, 在没有特殊硬件的支持下, 运算速度较慢, 且抗干扰能力较差, 当只需从静止背景下提取运动目标时性价比不高, 故而实际目标检测中光流法运用较少。

背景差分法的计算速度很快, 得到的差分结果可直接提取出较完整的目标位置、大小、形状等信息, 但也存在一些不足: 对光线很敏感, 背景噪声很多, 对背景图像的维持与更新有很高的要求; 当前景中的运动目标与背景图像的灰度值相近时, 检测出的信息不够完整; 检测出的部分区域易产生重影, 对后续的分析、识别等工作构成影响。故该方法只适用于外界条件较好的场合, 目前大多数监控系统越来越少应用这种非自适应的背景图像估计方法。

帧间差分法主要优点为：

- (1) 原理简单，较易于实现；
- (2) 主要涉及加减运算，运算量小，处理速度较快，实时性较好；
- (3) 因相邻帧图像之间时间间隔一般较小，故对运动目标敏感而对光线变化不太敏感，抗干扰能力较好。

主要缺点有：

- (1) 图像检测、分割精度不高，难以对目标区域进行精确描述；
- (2) 若目标运动速度过快或帧间的时间间隔过长，可造成两帧间无覆盖区域，而无法分割出运动目标。同理，若目标运动速度过小且帧间的时间间隔太小，可能形成两帧之间的过覆盖，甚至运动目标完全重合，而无法检测出运动目标；
- (3) 对内部灰度较均匀的目标，相邻帧的差分可能会在目标内部形成较大的空洞，而把一个目标分割成多个目标。

在实际中，由于帧间差分只使用了之前的一帧，可能无法识别一个颜色统一的较大的目标的内部像素，不能很好达到目标的完整性，这就为后续目标跟踪、分类造成影响，然而作为一种快速检测运动目标出现及运动目标的初步定位方法，帧间差分法在视频监控方面应用非常广泛。总的来说，帧间差分法较比光流法计算量小、更易于硬件实现，较之背景差法不需要时常更换背景、环境适应能力更好。因此，从工程应用的角度出发，本设计将采用帧间差分法来实现对运动目标的检测。

2.2.3 帧间差分法设计

首先，在假设外界光照未变化的条件下，将视频序列图像第k帧和第k+1帧分别表示为

$$f_k(x, y) = M_k(x, y) + B_k(x, y) + n_k(x, y) \quad (2.10)$$

$$f_{k+1}(x, y) = M_{k+1}(x + \Delta x, y + \Delta y) + B_{k+1}(x, y) + n_{k+1}(x, y) \quad (2.11)$$

将k+1时刻的图像与k时刻图像的对应像素作差分运算，

$$D_k(x, y) = f_{k+1}(x, y) - f_k(x, y) = [M_{k+1}(x + \Delta x, y + \Delta y) - M_k(x, y)] + [B_{k+1}(x, y) - B_k(x, y)] + [n_{k+1}(x, y) - n_k(x, y)] \quad (2.12)$$

其中 $M_{k+1}(x + \Delta x, y + \Delta y) - M_k(x, y)$ 为运动目标引起的图像变化， $B_{k+1}(x, y) - B_k(x, y)$ 是相邻帧间背景差， $n_{k+1}(x, y) - n_k(x, y)$ 为相邻帧间残留噪声。

用阈值T来根据公式(2.12)，二值化差分图像 $D_k(x, y)$ ，将像素值为255的点认为是前景点，即运动目标上的点；将像素值为0的点则认为是背景图像点，由此得到二值图像 $B_k(x, y)$ 。

$$B_k(x, y) = \begin{cases} 255 & \text{前景点, } |f_k(x, y) - f_{k+1}(x, y)| > T \\ 0 & \text{背景点, 其他} \end{cases} \quad (2.13)$$

其中, 阈值 T 的大小决定着整个系统检测的灵敏度, 若取值太大会抑制图像中的有用变化, 破坏运动目标的连续性; 若取值过小则不能有效减小噪声。

在帧间差分法中, 因阈值 T 无自适应性, 而难适应环境的变化, 下面介绍一种改进的算法。判定条件式 (2.12) 中增加整体光照敏感表征项, 其判定条件如下:

$$\sum_{(x,y) \in A} U(|f_{n+m}(x,y) - f_n(x,y)| - (T + \lambda \frac{1}{N_A} \sum_{(x,y) \in A} |f_{n+m}(x,y) - f_n(x,y)|)) > N_T \quad (2.14)$$

其中, λ 为抑制系数, N_A 为总区域像素数, 可以为整个图像。和判别式 (2.12) 一样, 如果统计像素点差分的绝对值 $|f_{n+m}(x,y) - f_n(x,y)|$ 超过 T 的像素点数大于一个设定值 N_T , 则判定该区域存在运动的目标。

整体光照敏感表征项

$$\lambda \frac{1}{N_A} \sum_{(x,y) \in A} |f_{n+m}(x,y) - f_n(x,y)| \quad (2.15)$$

表征光线整体变化。若光线变化比较小, 则表征项的值趋近于 0; 若图像光线变化较大, 则表征项有明显增大。可避免因光线变化产生的误判, 可有效抑制光线变化产生的影响。

式 (2.12) 中利用帧间差分法提取运动前景掩膜 $D_k(x,y)$ 时, 也要考虑背景图像的变化。帧差分法进行改进, 体现在选择最佳差分对象, 加入光照补偿, 或者如上节介绍的阈值 T 自适应。

2.3 运动目标跟踪算法研究

由于含有运动目标的序列图像比静止目标的一帧图像提供了更多的有用信息, 所以从动态序列图像中可以获得在单帧静态图像下很难得到的目标信息。

2.3.1 运动目标跟踪基本过程

2.3.1.1 目标跟踪特征描述

目标跟踪的原理就是在下一帧图像中找出目标确切的位置。一般的跟踪方法是首先提取被跟踪目标的图像, 建立一个模板, 然后在下一帧图像中进行匹配搜索, 直到找到匹配的位置。运动目标特征的选择对于跟踪的准确性非常关键。与静态图像不同, 运动图像是一类特殊的图像序列, 其含有大量的运动信息、前景目标的梯度信息、深度信息、颜色纹理特征、图像的直方图以及运动序列图像间的时间相关等。这些信息都可以作为跟踪目标的特征。

在跟踪系统中, 目标特征是指运动目标具有的一个或多个可度量的函数, 计算特征是对物体的一些重要特征进行定量估计的度量函数^[22], 特征提取是提取出一组特征, 组合形成特征向量, 这些被削减的信息包含了后续处理所需的全部知

识，特征选择和特征测量是特征提取的两个关键问题。

在具体的算法中，一般通过提取目标的形状特征来表示：

(1) 点：目标被定义为一个点，可以是目标的质心，也可以是一个点集。常用于目标在图像中所占区域较小时的情况；

(2) 几何形状：目标被描述为矩形或者椭圆等几何形状。适用于目标运动模型有平动、仿射或投影变换等情况；

(3) 轮廓：比较适合复杂的非刚体目标；

(4) 骨架模型：适用于有关节的形体或刚体目标。

在实际应用中，将目标的形状特征与外观特征一起应用一般能够提高的跟踪效果^[23,24]。主要外观描述特征有：

(1) 目标外观概率密度：通过计算图像模型区域内像素的参数获得，在实际应用中，目标外观概率密度估计可为有参的，如高斯型、混合高斯型等，也可为无参的，如Parzen窗、颜色直方图等；

(2) 模板：模板可以定义为目标的简单几何形状或者轮廓，同时包含了目标的空间和外观特征信息，但只适用于目标姿态在整个跟踪过程中保持不变的情况；

(3) 自适应外观模型：对目标的空间形状和外观特征同时建模，目标的形状被定义为一组标记，必须对先前的采样进行学习，故次描述特征不能被广泛应用。

对于图像帧中的运动目标，选取合适的目标特征对于跟踪算法的性能好坏至关重要。良好的目标特征应具有以下四个特点：

(1) 可区分性：不同目标之间，目标和背景之间，特征应有较明显的差异；

(2) 可靠性：同类的目标之间，特征值应比较接近；

(3) 独立性：同一目标的各特征应互不相关，相关性高的特征可以合并；

(4) 数量少：系统的计算复杂度随特征量的增加而迅速增长。高维空间中的原始特征的，可以映射到低维空间来表示。

2.3.1.2 目标跟踪基本过程

图2.3是运动目标跟踪流程图，基本步骤为：

(1) 在序列图像中检测运动目标和运动区域；

(2) 对检测到的运动目标进行图像分割；

(3) 对运动目标进行特征提取，并建立目标匹配模板；

(4) 用预测模型估计目标下一帧可能出现的位置，以确定下一帧的搜索范围。

(5) 用前一帧的目标模板在估计的搜索范围进行匹配搜索，寻找最佳匹配对象，若在预测范围内未找到目标，则进行特殊情况处理，如遮挡、暂停等；

(6) 不断利用匹配到的新目标图像，去修正被跟踪目标的模板数据。

在实际中，运动目标跟踪系统的难点是目标运动的复杂性以及视频数据的特殊性，如目标在图像中的大小、运动状态、摄像机的运动与否、运动目标与背景

的对比度，背景的稳定性等，这些因素会不同程度地影响运动目标跟踪的准确性和稳定性，主要体现在：光线变化、背景景物的变化、目标阴影与遮挡、前景目标在灰度、颜色及形状的相似、非静态背景的影响、运动目标高速运动、前景目标的高速运动等。

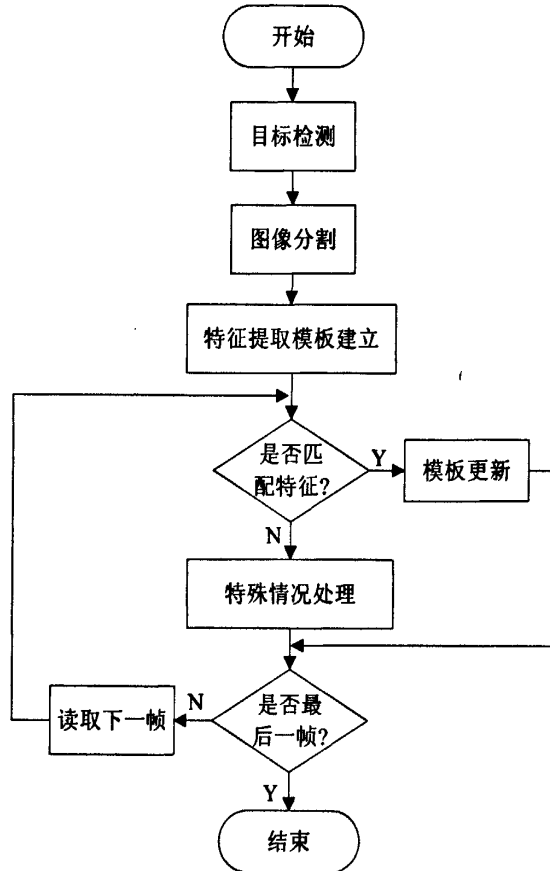


图2.3 运动目标跟踪流程图

2.3.2 Meanshift 目标跟踪算法设计

根据第1.2.2节的介绍，基于Meanshift的目标跟踪法基于模板匹配，且简单、快速、无需参数和快速模式。均值偏移（Meanshift）算法，最早是由Fukunaga等人的一篇关于概率密度梯度函数估计的文章中提出来的，最初的含义就是偏移的均值向量^[25]。Meanshift算法就是数据点到样本均值的重复移动，Meanshift算法属于聚类分析方法，密度估计器梯度递增，收敛点就是密度梯度局部极大值。Meanshift算法原理简单，运算较快，一般在分割后形成多个模态区域，便将问题分析从像素域转换到特征域，对后续的处理有利。

2.3.2.1 无参数密度估计

Meanshift算法是从密度函数梯度的无参估计中推导获得，无参估计是基于样本集来估计密度函数，最常用的是核函数密度估计，即根据核函数来计算样本集以得到密度函数。下面首先介绍核函数：记 R^d 为一个 d 维的欧式空间， X 点属于 R^d ，

用列向量表示, $\|x\|^2 = x^T x$, R 表示实数域。若一个函数 $K: R^d \rightarrow R$, 存在一个轮廓函数 $k: [0, \infty] \rightarrow R$, 即

$$K(x) = c_{k,d} k(\|x\|^2) \quad (2.16)$$

且符合以下条件:

- (1) k 是非负的;
- (2) k 是非增的, 即有: 若 $a < b$, 则 $k(a) \geq k(b)$;
- (3) k 是分段连续的, 且 $\int_0^{\infty} k(r) dr < \infty$;

则函数 $K(x)$ 称为核函数, 其中, $c_{k,d}$ 是一个使 $K(x)$ 等于1的归一化常数。在Meanshift算法中, 两类核函数较常用, 一种单位高斯核函数

$$K_N(x) = (2\pi)^{-\frac{d}{2}} \exp(-\frac{1}{2}\|x\|^2) \quad (2.17)$$

另一种是Epanechnikov核函数

$$K_E(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2) (1 - \|x\|^2), & \|x\| \leq 1 \\ 0, & \text{其他} \end{cases} \quad (2.18)$$

其中, C_d 为 d 维的单位球体的体积。

给定 R^d 空间中的 n 个数据点集 $\{x_i\}_{i=1, \dots, n}$, $K_H(x)$ 为空间的核函数, 则 R^d 空间 x 处核密度估计量

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n K_H(x - x_i) \quad (2.19)$$

其中, $K_H(x) = |H|^{-\frac{1}{2}} K(|H|^{\frac{1}{2}} x)$, H 为 $d \times d$ 维的正定对称矩阵, 称为带宽矩阵。使用带宽矩阵可增加估计灵活性, 在不同的子空间可采用不同的带宽, 同时也会增加计算量。实际应用中, 带宽矩阵常采取对角阵形式($H = \text{diag}[h_1^2, \dots, h_d^2]$)或比例单位矩阵形式($H = h^2 I$)。比例单位矩阵形式的优点是在带宽参数 $h > 0$ 时, 只需确定一个参数, 但首先要确保用于特征空间的欧几里德矩阵的有效性。采用比例单位矩阵形式的带宽矩阵, 把写成基于核函数的轮廓函数的密度估计表达式可写为

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n k\left(\frac{x - x_i}{h}\right) \quad (2.20)$$

将(2.17)式代入上式, 可得一个通用的核密度估计式

$$\hat{f}_{h,K}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \quad (2.21)$$

2.3.2.2 均值偏移向量

用Meanshift算法分析基本密度函数 $f(x)$ 的一个特征空间,首先是找到这个密度函数的模式,再对这个模式进行相关聚类。此模式应该在核密度梯度 $\nabla f_{h,K}(x)$ 的零点当中,而Meanshift算法直接对密度的梯度进行估计就能定位这些零点。密度梯度估计量 $\hat{\nabla}f_{h,K}(x)$ 是利用(2.21)式的线性性质来估计的

$$\hat{\nabla}f_{h,K}(x) \equiv \nabla \hat{f}_{h,K}(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^n (x-x_i) k' \left(\left\| \frac{x-x_i}{h} \right\|^2 \right) \quad (2.22)$$

若 $k(x)$ 在 $[0, \infty]$ 上除了有限点外可导,可以定义轮廓函数 $g(x)=-k'(x)$, $g(x)$ 导出的新核函数 $G(x)=Cg(\|x\|^2)$, C 是归一化因子。将 $g(x)$ 代入(2.23)式,得到

$$\begin{aligned} \hat{\nabla}f_{h,K}(x) &= \frac{2C}{nh^{d+2}} \sum_{i=1}^n (x-x_i) g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right) \\ &= \frac{2C}{nh^{d+2}} \left[\sum_{i=1}^n g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^n x_i g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)}{\sum_{i=1}^n g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)} - x \right] \end{aligned} \quad (2.23)$$

其中, $\sum_{i=1}^n g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)$ 为基于核函数 $G(x)$ 的无参密度估计,记为 $\hat{f}_{h,G}(x)$,

$$\hat{f}_{h,G}(x) = \frac{C}{nh^d} \sum_{i=1}^n g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right) \quad (2.24)$$

定义均值偏移向量 $m_{h,G}(x)$ 为:

$$m_{h,G}(x) = \frac{\sum_{i=1}^n x_i g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)}{\sum_{i=1}^n g \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)} - x \quad (2.25)$$

它是使用核函数 $G(x)$ 作为权值的加权平均值与 x 的差。因此,式(2.24)可记为

$$\hat{\nabla}f_{h,K}(x) = \frac{2}{h^2 C} \hat{f}_{h,G}(x) m_{h,G}(x) \quad (2.26)$$

从而得到

$$m_{h,G}(x) = \frac{h^2 C \hat{\nabla} f_{h,K}(x)}{2 \hat{f}_{h,G}(x)} \quad (2.27)$$

式(2.26)表明，在点 x 处，基于核函数 $G(x)$ 的均值偏移性向量与基于核函数 $K(x)$ 的归一化密度梯度估计成正比，归一化因子为基于核函数 $G(x)$ 的密度估计，因此，均值偏移向量总是指向密度增大最大的方向。

2.3.2.3 均值偏移算法

由(2.26)式，记 Meanshift 向量 $m(x)$ 为

$$m(x) = \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} \quad (2.28)$$

由第二章相关结论可知，Meanshift 算法流程图见图 2.4。

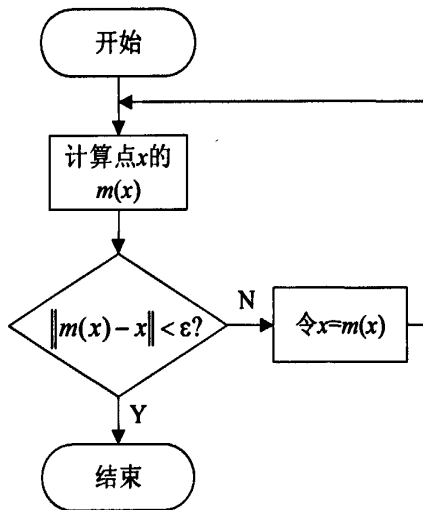


图 2.4 均值偏移算法计算流程图

给定初始点 x 、核函数 $G(x)$ 、容许误差 ε ，则 Meanshift 算法循环执行下面步骤，直到满足结束条件：

- (1) 计算 Meanshift 向量 $m(x)$;
- (2) 若 $\|m(x) - x\| < \varepsilon$ ，则结束循环；
- (3) 否则，把 $m(x)$ 赋值给 x ，继续执行(1)。

由式 (2.26) 得 $m(x) = x + m_{h,G}(x)$ ，因此 Meanshift 算法就是不断沿密度梯度方向移动，移动步长既与梯度大小有关，也与该点的密度有关，在密度大的地方步长小，在密度小的地方步长大些。这说明 Meanshift 算法是一种变步长的梯度上升算法，在满足一定的条件下，最终会收敛到该点附近的峰值。

2.3.2.4 Meanshit 的目标跟踪算法

Meanshit 的目标跟踪算法中被跟踪目标的目标区域就是核函数的作用区域, 区域大小等于核函数的带宽。基于 Meanshit 的目标跟踪算法通常采用颜色特征, 用 Bhattacharyya 系数衡量目标模型直方图和候选目标区域直方图的相似性, 这样目标跟踪问题就转化为 Meanshit 模式匹配问题。

1 目标模型描述

目标模型描述过程为对初始帧图像中目标区域内每一个像素点计算在特征空间中特征值的概率。将 R、G 和 B 的每个子空间分成 m 个相等的空间, 即特征值的个数为 m , 用 $\{x_i\}_{i=1, \dots, n}$ 表示目标模板区域的 n 个像素值, 目标模板区域的中心坐标为 x_0 , 则目标模型的特征值 $u=1, \dots, m$ 的概率密度就是目标模板区域对应的核函数直方图, 为

$$\hat{q}_u = C_1 \sum_{i=1}^n k \left(\left\| \frac{x_0 - x_i}{h} \right\|^2 \right) \delta [b(x_i) - u] \quad (2.29)$$

上式中:

(1) $k(x)$ 为核函数的轮廓函数, 处在目标区域外围的像素易受背景或遮挡等影响, 故处在目标区域中心的像素更可靠, 一般对距离目标中心远的像素赋予较小的权值, 对靠近中心的像素赋予较大的权值, 以增强密度估计的鲁棒性;

(2) 轮廓函数 $k(x)$ 中 $\left\| \frac{x_0 - x_i}{h} \right\|^2$ 的作用是归一化目标区域, 消除大小不同目标计算的影响;

(3) δ 为克罗内克 (Kronecker delta) 函数, $\delta [b(x_i) - u]$ 用来判断目标区域各个像素的颜色值是否属于第 u 个特征空间; h 为核函数窗宽, $b(x_i)$ 为 x_i 点像素对应颜色空间中的颜色值, C_1 为归一化系数, 使得条件 $\sum_{u=1}^m \hat{q}_u = 1$, 因此得到

$$C_1 = \frac{1}{\sum_{i=1}^n k \left(\left\| \frac{x_0 - x_i}{h} \right\|^2 \right)} \quad (2.30)$$

2 候选目标模型

初始帧以及之后每帧图像中可能存在目标的区域称为候选区域, 对候选区域的描述称为候选目标模型。在目标可能出现的候选区域, 对特征空间特征值计算即为候选目标模型描述。

设 $\{x'_i\}_{i=1, \dots, n_y}$ 为当前帧候选目标像素, y 是候选区域中心坐标, 那么候选目标区

域核函数直方图，也即候选模型特征值 $u=1, \dots, m$ 的概率密度为

$$\hat{p}_u(y) = C_2 \sum_{i=1}^n k \left(\left\| \frac{y - x'_i}{h} \right\|^2 \right) \delta [b(x'_i) - u], u=1, \dots, m \quad (2.31)$$

其中， C_2 为归一化系数，根据条件 $\sum_{u=1}^m \hat{p}_u(y) = 1$ ，可得

$$C_2 = \frac{1}{\sum_{i=1}^n k \left(\left\| \frac{y - x'_i}{h} \right\|^2 \right)} \quad (2.32)$$

2.3.2.5 加入 Bhattacharyya 系数的 meanshift 跟踪算法

相似性函数可描述目标模型和目标候选区域之间的相似程度，理想情况下两个模型的概率分布应该是完全相同的。在统计检验中目标和候选目标分类的错误概率与两个样本分布的相似性直接相关，故而可用分类的错误概率来衡量两个样本分布的相似性。两个样本分布的分类错误概率越高，说明分布相似度越高^[35]。这就把目标位置估计的问题转化为使目标模型跟候选区域Bayes误差的最大化问题。

下面的研究中，假设目标在估计位置邻域任何位置出现概率相等。实际度量中经常用的Bhattacharyya系数和Bayes误差密切相关，也和样本密度函数相关。如果目标模型直方图估计的离散密度函数为 $\hat{q} = \{\hat{q}_u\}_{u=1, \dots, m}$ ，其中 $\sum \hat{q}_u = 1$ ，给定的位置 y ，记密度函数

$$\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1, \dots, m} \quad (2.33)$$

其中，

$$\sum_{u=1}^m \hat{p}_u(y) = 1 \quad (2.34)$$

则Bhattacharyya系数样本估计

$$\hat{\rho}(y) \equiv \rho(\hat{p}(y), \hat{q}) = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u} \quad (2.35)$$

上式几何含义为 m 维的单位矢量 $(\sqrt{\hat{p}_1}, \dots, \sqrt{\hat{p}_m})^T$ 和 $(\sqrt{\hat{q}_1}, \dots, \sqrt{\hat{q}_m})^T$ 夹角的余弦值，其值在 $0 \sim 1$ 之间， $\hat{\rho}(y)$ 越大，两模型相似度越高。对当前帧不同的候选区域运算所得候选模型中，使 $\hat{\rho}(y)$ 最大的那个候选区域就是当前帧中目标所在位置。

直方图虽然不是最佳非参密度估计方法，但它计算量小可满足实时处理的要求。利用(2.35)式可以将目标和候选目标分类的距离记为

$$dist(y) = \sqrt{1 - \rho[\hat{p}(y), \hat{q}]} \quad (2.36)$$

可以通过最小化式(2.36)的 $dist(y)$ 距离值获得目标在当前帧中最可能的位置 y , 这等价于最大化(2.35)式中的Bhattacharyya系数的 $\hat{p}(y)$ 。

公式(2.36)得到的统计量可用来进行目标定位:

- (1) 它与Bayes误差相近, 接近最优的;
- (2) 运用离散密度函数, 不受目标比例变化的影响;
- (3) 适用于任意分布。

下面介绍目标定位算法的推导过程, 设当前帧中目标的中心位置为 y , 为使 $\hat{p}(y)$ 最大, 先将前一帧目标中心位置 y_0 定为当前帧的目标中心, 从 y_0 开始搜索最优匹配的目标区域。

首先计算当前帧 y_0 处的候选目标模型 $\hat{p}(y_0)$, 把Bhattacharyya系数 $\hat{p}(y)$ 在 $\hat{p}(y_0)$ 处用泰勒公式展开, 略去高阶项:

$$\rho[\hat{p}(y_0), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(y) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \quad (2.37)$$

把(2.33)式代入(2.37)

$$\rho[\hat{p}(y_0), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} + \frac{C_2}{2} \sum_{i=1}^{n_k} s_i k \left(\left\| \frac{y - x'_i}{h} \right\|^2 \right) \quad (2.38)$$

其中, k 为当前帧中核函数的轮廓函数 $k(x)$, 其中

$$s_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \delta[b(x'_i) - u] \quad (2.39)$$

式(2.38)中的第一项与 y 无关, 只有第二项与 y 有关, 故要最小化 $dist(y) = \sqrt{1 - \rho[\hat{p}(y), \hat{q}]}$, 只需对

$$\frac{C_2}{2} \sum_{i=1}^{n_k} s_i k \left(\left\| \frac{y - x'_i}{h} \right\|^2 \right) \quad (2.40)$$

进行最大化设置。第二项是当前帧中利用核函数的轮廓函数 $k(x)$ 和像素的权值 s_i 计算得到的密度估计。通过计算式(2.40)的Meanshift向量, 可得候选区域目标中心位置 y_0 指向目标真实位置 y 的向量

$$m_{h,G}(y) = y - y_0 = \frac{\sum_{i=1}^{n_k} s_i x_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_k} s_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)} - y_0 \quad (2.41)$$

其中， $g(x) = -k'(x)$ ， $m_{h,G}(y)$ 是目标中心从起始点 y_0 向 y 运动的向量，迭代过程就是以 y_0 为起点不断计算 (2.41) 式，以两个模型相比颜色变化最大的方向移动，直至 Bhattacharyya 系数最大时即定位为目标最终中心位置，便停止迭代，因此该算法优于一般的全局搜索算法。

2.4 本章小结

本章主要对运动目标检测与跟踪算法进行了研究。包括图像增强、图像去噪等图像处理的基本方法，对当前流行的运动目标检测与跟踪算法进行了研究和对比，重点研究了帧间差分法和均值偏移算法原理。

第 3 章 运动目标检测与跟踪系统硬件设计

硬件电路是系统的运行平台，是各模块功能实现的载体，本章是运动目标检测与跟踪系统的硬件设计，主要包括硬件电路总体设计、各个功能模块的电路设计以及各电子元件的选择等。

3.1 系统整体设计

首先，CCD摄像头获取模拟视频信号后送往视频解码单元，经解码后得到YUV格式的行场同步信号，输入FPGA图像采集单元，之后分成三路，一路送往调试PC用于调试，另一路经过FPGA暂存起来，第三路由Blackfin533通过按行DMA的方式将图像数据传至DSP内部进行检测与跟踪算法处理，输出的目标轨迹信息送往PC显示，调试PC可通过串口向FPGA和DSP发送控制信息，系统硬件设计整体框图见图3.1。

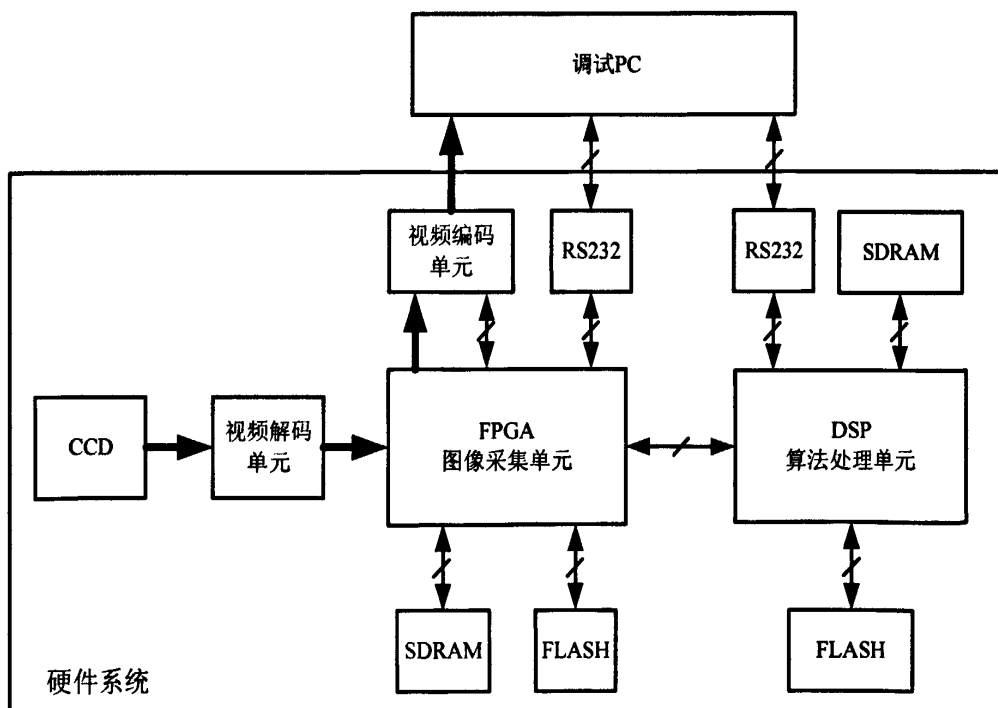


图 3.1 系统整体设计框图

硬件系统由以下功能模块组成：

(1) 系统处理模块：包括FPGA图像采集模块和Blackfin533算法处理模块，用以完成系统的控制、信号的运算处理等功能：

(2) 通信控制模块：通过Blackfin533的串口及串口芯片实现硬件系统与调试

PC的数据通讯，以及BF533与FPGA子系统的通信；

(3) 视频输入、输出模块：主要由外接CCD摄像头、视频解码芯片SAA7113和视频编码芯片SAA7121等组成，完成视频信号的采集、输出显示等任务；

(4) 外部存储模块：包括SDRAM和FLASH等，其中SDRAM用来存储视频图像信号及运算过程中产生的数据等，FLASH主要用于Blackfin533和FPGA的上电加载程序。

3.2 系统处理模块设计

3.2.1 Blackfin533 DSP

美国模拟器件公司(ADI)是全球通用DSP芯片主要供应商，该公司所生产的高性能信号处理器处于世界领先水平。Blackfin系列属于ADI和英特尔公司联合开发的16位DSP的一个大系列，被广泛应用在图像、文本等数据流处理领域。Blackfin系列目前发布的产品包括BF531、BF532、BF533、BF561、BF592等。本系统选用BF533作为核心处理器，因为该型号处理器具备各种接口，性价比高，具有强大的多媒体数据处理能力。BF533处理器的结构适于各种数字信号如音视频和数据通信的处理，并为这些处理提供了各种控制模块。BF533为该系列中公布的第四款产品，图3.2为其结构框图。

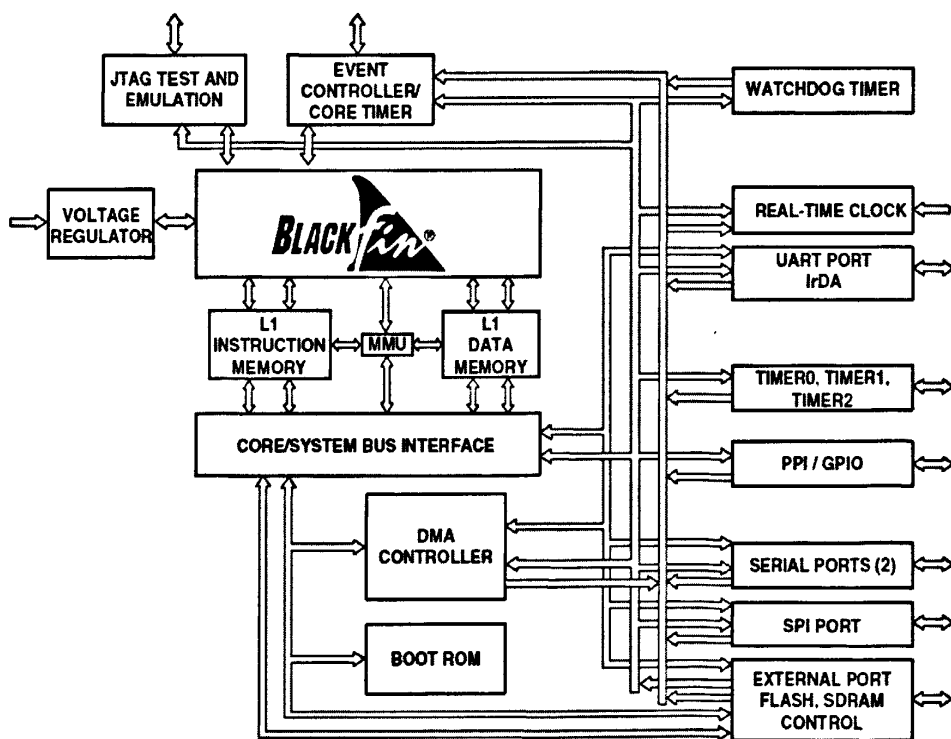


图 3.2 BF533 结构框图

BF533的总体特点为：

(1) 16位的定点DSP内核，可在600MHz下持续运行，灵活的软件控制动态电源管理方案；

(2) 4GB统一寻址空间，L1指令存储器为80KB SRAM,其中16KB可配置为四路组联合Cache；

(3) L1数据存储器为2个32KB SRAM Bank，每个Bank包括2个16KB SRAM，其中一个16KB可以配置为Cache；

(4) 4KB L1 暂存数据 SRAM，访问速度为处理器最快速度，有利于处理器内核压缩视频数据时安排待处理数据，不可配置为 Cache；

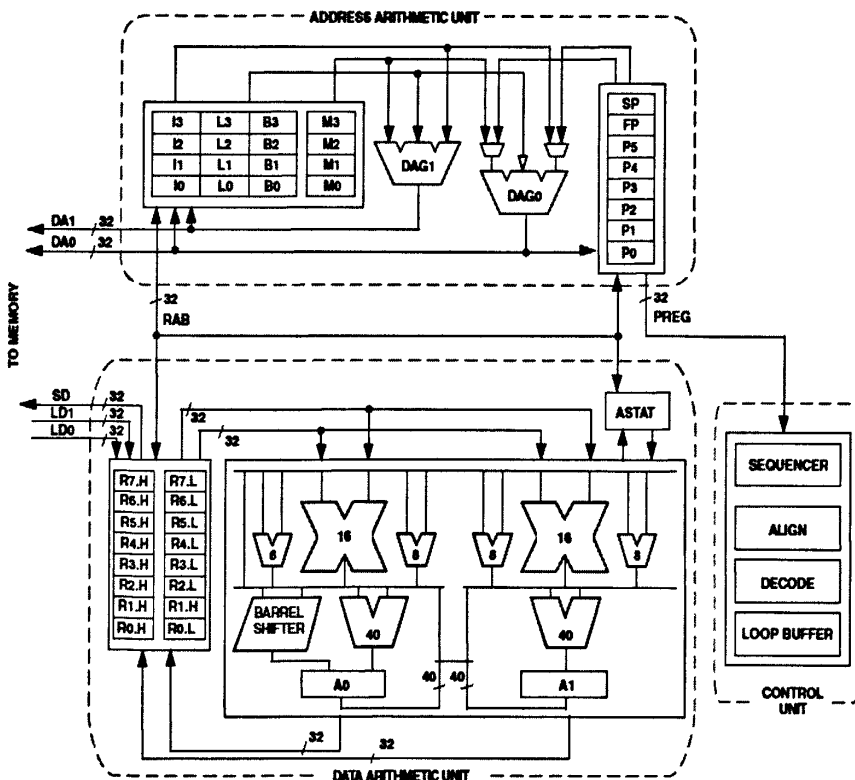
(5) 支持片外同步、异步存储器，灵活引导方式（SPI或外部存储资源）；

(6) 内存的管理单元可提供内存保护，事件处理；

(7) RTC模块，Watch Dog定时器，3个定时/计数器，支持PWM，16个GPIO；一个支持IrDA通用串行接口；

(8) 并行接口支持ITU-656视频数据格式，一个SPI兼容端口，两个双通道全双工同步串行接口，支持8个立体声I2S通道；

(9) 12通道DMA，片内自带1x-63xPLL；0.7~1.2V核心电压，3.3V I/O。



此外，芯片还备有看门狗电路及多种定时器，能充分保证软件设计的稳定性。还可以控制不同电压输入，动态调整运行频率，有效降低功耗，适用于便携式产品设计。设计了很多提高算法效能的技术，如芯片资源采用动态分配方案，90%

以上都能被利用，当用于多媒体解码时，所有的资源都能被调用。二维 DMA 传输可大大降低软件负荷，其动态 Cache 分配可明显提高视频处理效率。通过以上介绍可知 BF533 是非常适于视频处理的 MSA 架构处理器。

3.2.2 FPGA 特性及其基本结构

视频采集芯片采用可升级的 Xilinx 公司生产的 Spartan-3E 系列 FPGA，本课题选用 XC3S500E，后续研发可根据设计需求，无需更改硬件的前提下字节升级为 XC3S1600E。该芯片为 Spartan-3E 系列上的改进产品，提供更多的 I/O 端口和更低的成本，采用 90nm 工艺、最大 I/O 端口数 376 个、18 种 I/O 标准、最大单端口传输速率 622Mb/s、最多 36 个 18×18 的专用乘法器、648KB 的 RAM、时钟频率 5~300MHz、包含多个专用片上数字时钟管理模块及丰富的布线资源^[22]。

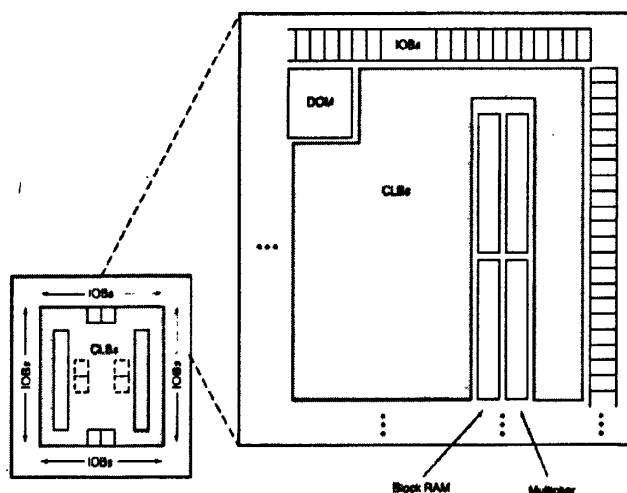


图 3.4 Spartan-3E 系列产品内部结构图

Spartan-3E 内部结构见图 3.4，主要包括可配置逻辑模块 CLB、输入/输出接口模块 I/O 单元、数字时钟管理模块 DCM、数字延迟锁相环、BlockRAM 和硬核乘法器。CLB 是 FPGA 的基本逻辑单元，包括一个配置开关矩阵，以及选择电路和触发器。八个 DCM 用来提供数字时钟管理和相位环路锁定。相位环路锁定可变换时钟频率和调整时钟输出相位。BlockRAM 内嵌于芯片的边缘，可配置存储结构。硬核乘法器相当于 ASIC 电路。此外还含丰富的布线资源以连通内部各单元，布线长度、设计工艺等因素对信号的驱动能力和传输速度有很大影响。

3.3 通信控制模块设计

通信控制模块用来完成 PC 机与板卡之间以及 BF533 与 FPGA 子系统之间的数据传输与控制。

3.3.1 PC 机与板卡连接设计

编码设计中要进行大量的调试工作，从 BF533 的存储空间中通过 JTAG 接口导

出数据往往花费大量的时间,利用UART接口,可大大缩短图像输出的时间,并且在后续的设计中可以利用UART接口作为系统与本地监视器连接的接口,作为本地的数据通信,其连接框图见图3.5。

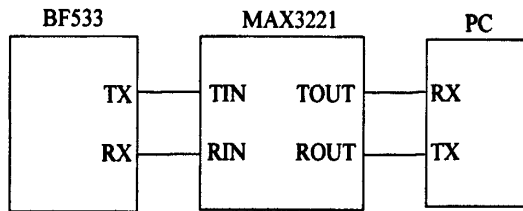


图3.5 BF533与PC机连接框图

3.3.2 BF533 与 FPGA 连接设计

BF533上的PPI(Parallel Peripheral Interface)接口是一个高速并行外设接口,也是视频端口, BF533通过它来连接FPGA子系统,连接原理图见图3.6。

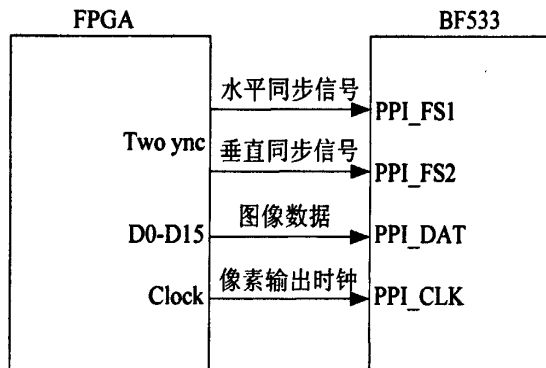


图3.6 BF533与FPGA数据传输原理图

3.4 视频输入、输出模块设计

视频输入模块的主要作用是视频图像采集,实现光电转换,并经A/D转换成数字信号,再存储到SDRAM中。视频输出模块的作用是将视频信号在FPGA的控制下输出到视频编码芯片中,经过视频编码芯片处理后,经D/A转换成标准的模拟视频信号,再送至显示器输出显示^[27]。

3.4.1 视频输入模块

本系统的视频输入模块主要由sony公司的CCD CAMERA(帧频25FPS、分辨率720×576),飞利浦公司的SAA7113芯片和FPGA的I²C接口等组成^[28]。

图像传感器是图像采集系统中的核心部分,目前常用的主要有CCD和COMS两种。两者都是利用感光二极管进行光电转换,CCD集成在半导体单晶材料上,而CMOS集成在金属氧化物半导体上;CCD将采集到的各个像素的数据汇聚至边缘

再进行放大处理,要外加A/D转换电路,其特殊工艺可保证数据在传送时不会失真,而CMOS每个像素旁边都连着ADC,电荷信号直接放大并转换成数字信号,在传输距离较长时会产生噪声。相对来说CCD成像质量高、噪声小且寿命长,故本系统采用CCD传感器进行图像采集。

SAA7113 是飞利浦公司生产的一款视频解码系列芯片,它支持多种视频信号格式,能将标准视频信号作 A/D 转换,转化为 8 位的数字信号,将转换结果输出给后续单元,同时输出行、场同步信号。

从 CCD 摄像头得到的视频信号经 SAA7113 进行钳位、滤波、A/D 转换等处理以及 YUV 分离电路,得到 ITU655 视频数据流,输出给 FPGA 进行处理^[29]。其电路原理图见图 3.7。

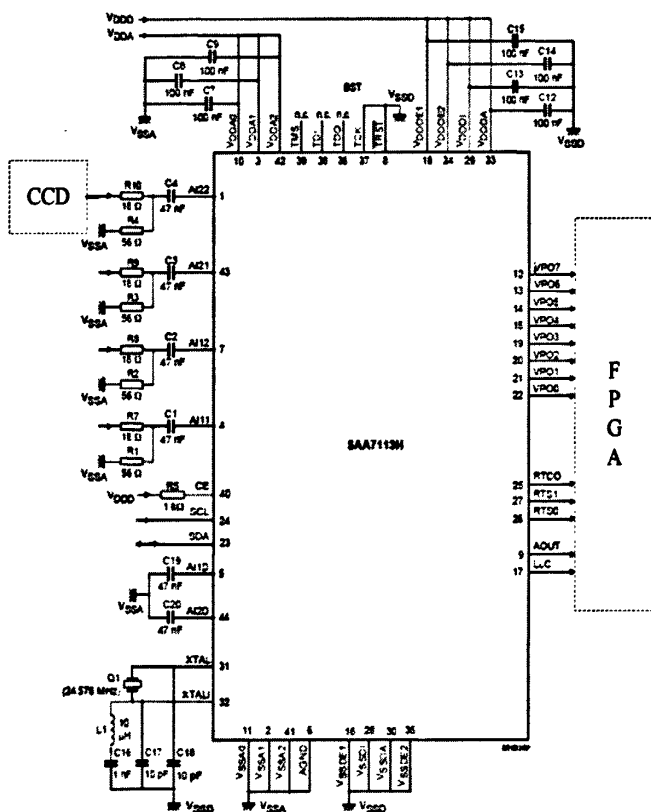


图 3.7 SAA7113 视频采集电路

采用 3.3V 供电,内含锁相环电路,需外接一个 24.576MHz 的晶振,以产生 LLC (line-locked system clock output) 信号及二频信号 LLC2 (频率 13.5MHz)。LLC2 用于同步整个系统,一个 LLC2 周期采集一个像素数据。通过自带的 I²C 接口控制诸如输入信号预处理、色度亮度、输出数据格式和输出同步信号选择等电路。因为 I²C 接口采用漏极开路 (Open Drain) 设计,而元件本身只能输出低电平,故必须通过一个上拉电阻保证两根总线在空闲状态时处于高电平状态。

3.4.2 视频输出模块

经过系统处理后的数字视频信号在 FPGA 的控制下输出给视频编码芯片 SAA7121, 经过 SAA7121 处理后, 转换成标准模拟视频信号, 并送至显示器输出显示。SAA7121 是飞利浦公司的一款高集成度视频编码芯片, 其内包括三个 10 位 D/A 转换器, 支持多种视频格式, 能将数字视频信号编码为 PAL 制、NTSC 制等标准模拟视频信号^[30]。

FPGA 将 SDRAM 中储存的已处理的图像数据输入视频编码芯片 SAA7121 进行编码处理并输出给显示器, 其视频输出原理图见图 3.8。XC3S500E 将 ITU-656 视频数据流输入 SAA7121 芯片, 经过 SAA7121 分离出 Y、U、V 信号, 再交由相应的 D/A 模块转化为模拟信号。

SAA7121 的配置方式与 SAA7113 相同, FPGA 通过 I²C 总线对其内部的寄存器进行配置, 以实现对其 SAA7121 的控制。

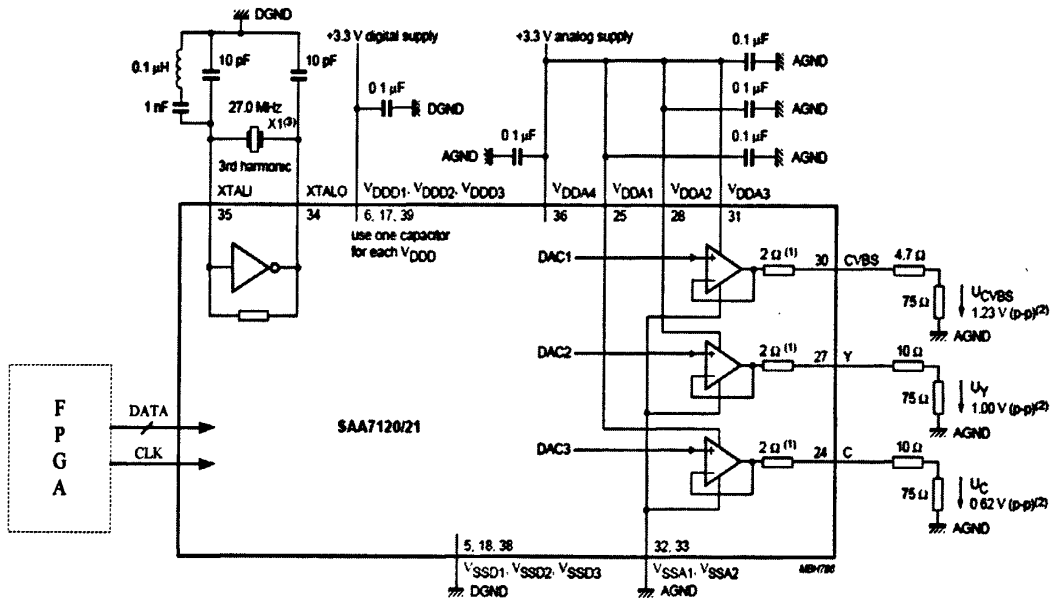


图 3.8 SAA7121 视频输出电路

3.5 外部存储模块设计

虽然 BF533 和 FPGA 内部具有一定的 RAM 资源, 但是数量有限, 如 BF533 包括两个 32KB 的数据存储器和一个 80KB 的程序存储器, XC3S500E 内部共具有 73KB 的分散 RAM 和 360KB 的块状 RAM, 而本系统采集的图像分辨率为 720×576 , 每像素占用一个 Byte, 则一幅图像所占的空间即为 414,720 Byte, 这对于视频信号处理来说是不够的。因此, 本系统需要扩展外加存储器芯片来辅助 FPGA 进行视频数据处理。

外部存储模块主要包括 SDRAM、SRAM、DRAM、VRAM 和 FLASH 等。其中 DRAM、SDRAM 为动态存储器, 需要经常更新, 容量较大、价格相对较低; SRAM 为静态存储器, 不用刷新, 价格相对较高。本系统选用 SDRAM 作为外部动态存储

模块。

3.5.1 SDRAM 模块设计

SDRAM称为同步动态存取存储器，同步表明存储器的存取过程受系统时钟信号的控制。SDRAM用突发模式进行内存读写，系统时钟将所有地址数据和控制信号同步，这大大提高了数据传输能力^[31]，简化了设计；作用与DRAM相似，也定时刷新，通常每15.6us刷新一次。SDRAM可同时对DRAM的两列地址进行内存存取，SDRAM的存取速度可达180MHz，SDRAM的访问一般为四个连续周期，第一个周期需要四个系统时钟周期，后三个周期只需要1个系统时钟周期。在系统应用中，SDRAM一般用来存储采集到的数据和处理过程中产生的中间数据。

本系统采用的 SDRAM 是两片 MICRON 公司生产的 MT48LC8M32 芯片，存储空间为 8M×16bits，访问速度较快。

1 SDRAM与BF533连接

SDRAM地址分配范围：0x00000000-0x07ffffff，图3.9为BF533存储空间分配图，SDRAM与BF533连接电路原理图见图3.10。

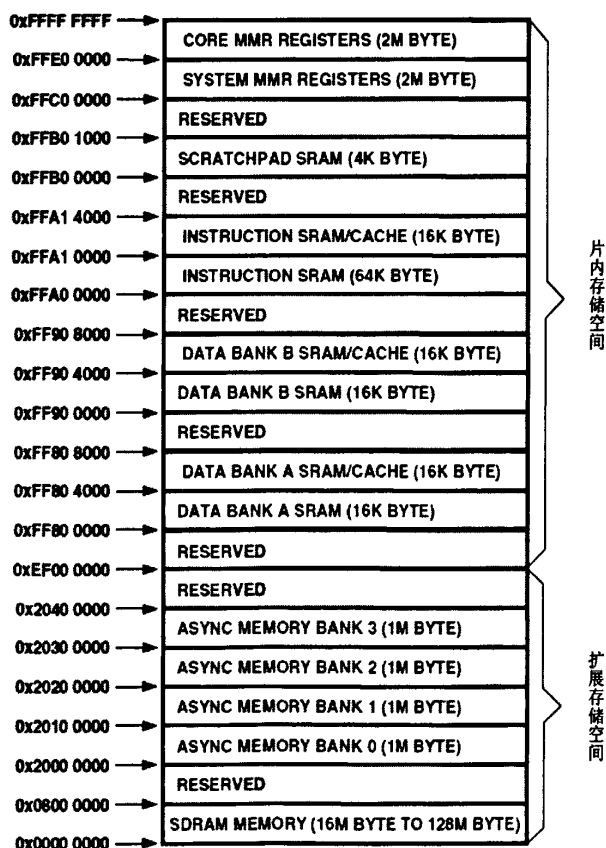


图 3.9 BF533 存储空间分配图

2 SDRAM 与 FPGA 连接

SDRAM 与 FPGA 连接电路原理图见图 3.11。

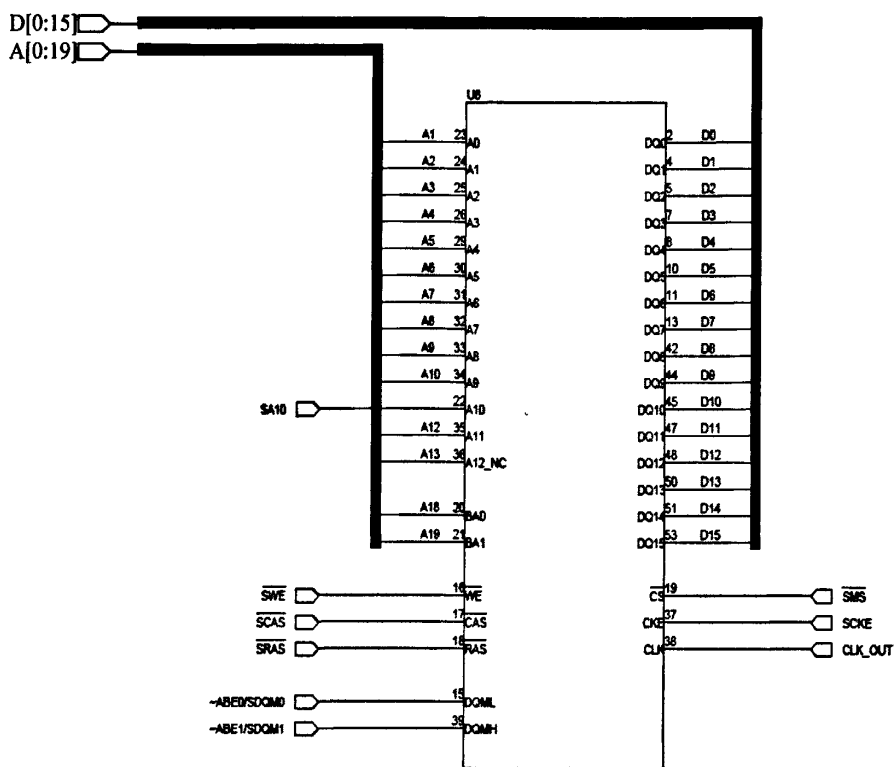


图3.10 BF533与SDRAM连接原理图

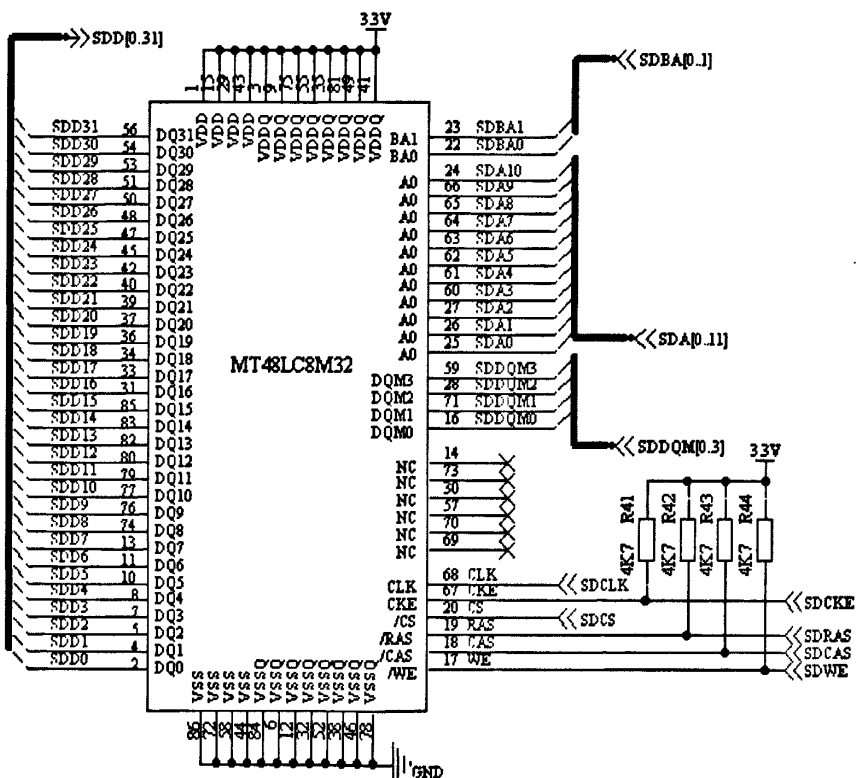


图3.11FPGA与SDRAM连接原理图

3.5.2 FLASH 模块设计

在系统中，FLASH的作用是在DSP和FPGA上电加载时提供程序代码。系统每

调试完一个程序，就要对DSP进行引导，以使DSP系统能够自行脱机运行。引导方法有并行I/O口引导、串行口引导、SPI引导装载和外部并行引导等。

采用FLASH进行BF533引导程序，分配地址：0x20000000-0x200ffff; FLASH与BF533的设计连接框图见图3.12。

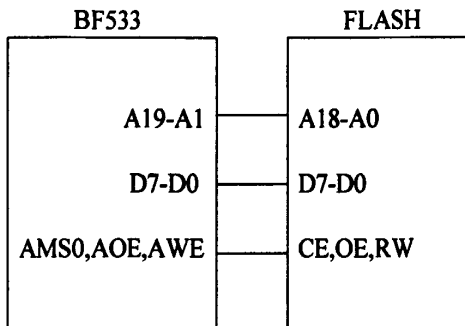


图3.12 与BF533与FLASH连接框图

FPGA也是SDRAM型结构，需要一片FLASH进行上电配置^[32]。本系统采用的XCF04S^[26]具有4Mbit的存储空间，其FLASH配置电路原理图见图3.13。

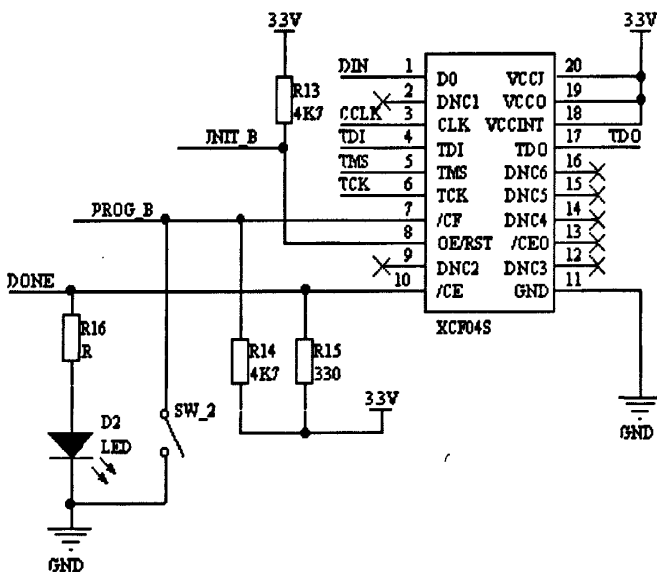


图 3.13 FPGA 的 FLAS 配置电路原理图

按下SW_2时可实现对XCF04S的配置，之后每当系统上电时，XCF04S即可为FPGA提供加载程序。

3.6 调试接口模块设计

调试接口模块用于调试阶段，本系统为标准JTAG接口，接口连接与仿真器引脚一致，ADI仿真器JTAG引脚连接图见图3.14。系统为低噪声系统，故TRST脚悬空；为使信号的上升沿小于10us，EMU0和EMU1脚需接上拉电阻，可选4.7K或10K。

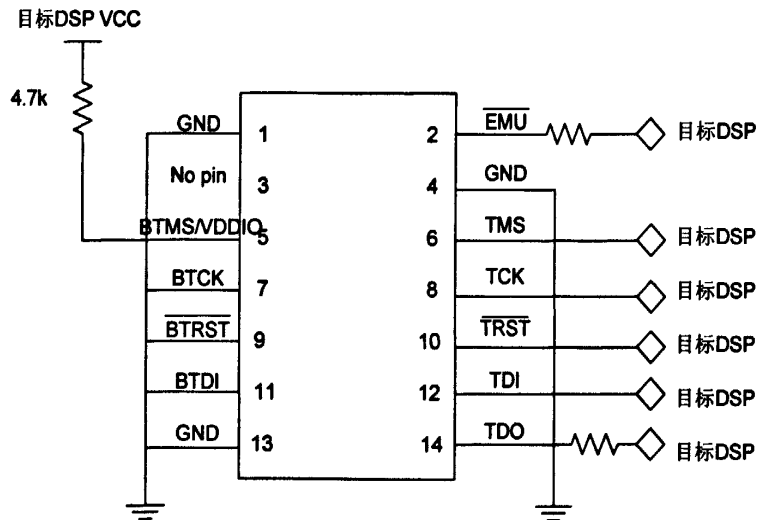


图3.14 JTAG连接图

3.7 电源电路设计

电源电路提供系统各模块所需电源，系统需要双电源供电，外部 I/O 使用 3.3V 电源，内核使用 1.2V 电源。本系统选用设计简单、使用方便、纹波小的线性稳压芯片，通过 LT1086CM-3.3 将外部 5V 电源转换成 3.3V，再通过 LT187CM-1.2 将 3.3V 电压转换成 1.2V，电源电路原理图见图 3.15。

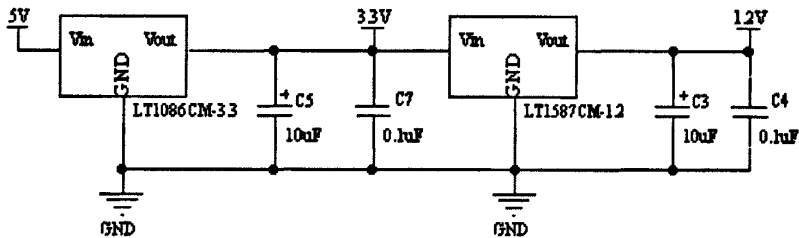


图 3.15 电源电路原理图

3.8 本章小结

本章主要介绍了运动目标检测与跟踪的总体功能设计与硬件设计。包括处理器设计、通信控制模块设计、数据输入输出模块设计、SDRAM 存储器设计、FLASH 存储器设计、调试接口以及电源电路设计，主要介绍了各模块的功能与结构，给出了各模块的设计原理图。

第 4 章 运动目标检测与跟踪系统硬件调试

4.1 初始化与时钟配置

4.1.1 硬件系统初始化

由于驱动程序位于服务程序上方，因此应当首先初始化服务程序，再初始化设备管理器。首先要初始化中断管理器，然后初始化外设总线接口 EBIU(External Bus Interface Unit)，然后依次是电源管理、端口控制、延迟回调、DMA 管理器、标志控制、定时器等服务程序，最后是设备管理器。无需使用的服务程序不用进行初始化。只要按照这个特定顺序正确地完成所有的初始化，就不会发生任何冲突，应不会发生在未完成初始化的情况下试图调用服务程序或设备驱动程序的情况。

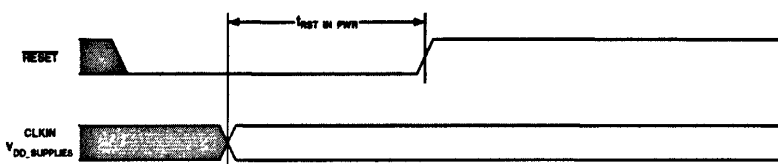


图4.1 上电启动时序图

4.1.2 时钟频率配置

BF533 可以动态改变工作频率和工作电压，在满足应用性能的前提下，使消耗的功耗最小。BF533 的输入时钟 CLKIN 提供所需的时钟频率，经片内锁相环 PLL 精确倍频后得到内核时钟 CCLK，CCLK 分频后，得到系统时钟 SCLK。内核以 CCLK 频率运行，而外设访问总线 PAB、DMA 总线、外部地址总线 EAB、EBIU 等则以 SCLK 运行。为了优化性能和功耗，Blackfin 允许内核时钟和系统时钟动态变化。

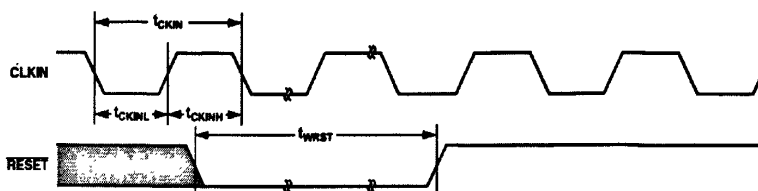


图4.2 时钟、重启时序图

图 4.3 为系统各频率之间的关系图，可知 $SCLK \leq CCLK, SCLK \leq 133MHz$ 。

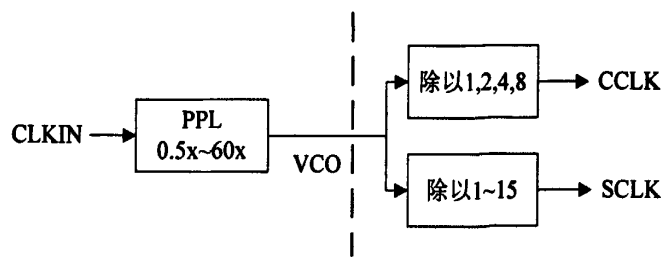


图4.3 各频率之间的关系

表 4.1 为 CCLK 与 VCO 的倍率关系，由 CSEL1-0 决定。

表 4.1 VCO/CCLK 倍率表

CSEL1-0	VCO/CCLK	VCO	CCLK
00	1:1	300	300
01	2:1	300	150
10	4:1	400	100
11	8:1	200	25

表 4.2 为 SCLK 与 VCO 的倍率关系，由 SSEL3-0 决定，其中 VCO/SCLK=SSEL3-0。

表 4.2 VCO/SCLK 倍率表

SSEL3-0	VCO/SCLK	VCO	SCLK
0001	1:1	100	100
0101	5:1	400	80
1010	10:1	500	50

因 BF533 内核时钟最高可达 600MHz，故系统时钟最高可以运行在 133MHz，本系统可忽略 DSP 的功耗问题，为使系统性能最高，同时考虑稳定性，输入时钟 CLKIN 为 20MHz，将内核时钟配置为 320MHz，系统时钟运行在 80MHz。

通过设置图 4.6 所示 PLL 控制寄存器 PPL_CTL 来为 PPL 设置字段从而配置时钟频率。

6	5	4	3	2	1	0		
	PDWN		STOPCK		PPL_PPL	DF		
15	14	13	12	11	10	9	8	7
	MSEL[5:0]						BYPASS	

图4.4 PPL控制寄存器

寄存器各位意义如下：DF 分频位为 0 则通过 CLKIN 至 PLL，为 1 则通过 CLKIN/2 至 PLL；PPL-PPL 为 0 时 PLL 上电，为 1 时 PLL 断电；STOPCK 停止时钟位为 0 时 CLKIN 至 PLL，为 1 时 CLKIN/2 至 PLL；PDWN 能量关闭位为 0 开全部内部时钟，为 1 关全部内部时钟；BYPASS 为 0 未通过 PPL，为 1 通过 PPL；

MSEL[5:0]位为 CLKIN/VCO 乘法单元。

本设计中将 PPL_CTL 设置为 0x2C00，表示开内部时钟，倍频系数为 22、PLL 上电但被旁路，CLKIN 直接到 PLL，使能输入分频器，分频系数为 1。

4.2 通信与控制

4.2.1 PPI 接口配置

BF533 提供的 PPI 数据线宽度可以在 8 位~16 位之间设置，是一个双向的半双工接口，能够与高速 A/D 转换器、D/A 转换器或其它通用外围设备直接并行连接，适合大量数据的高速连续输入与输出，包括三条同步信号线和一个与外部时钟相连的时钟引脚^[36]。本系统中，BF533 通过 PPI 接口连接 FPGA，FPGA 的 I²C 总线与 BF533 的双线接口之间，形成一个控制通道。PPI 时钟由 FPGA 的像素输出时钟 CLK 驱动，PPI 可以在 CLK 高达 65MHz 的情况下接收数据，所以完全能够满足本系统的 FPGA 50MHz 速率的像素数据传输。PPI 接口分 RX 和 TX 两种模式，在这两种模式下与片内时钟同步时序图见图 4.5、图 4.6。

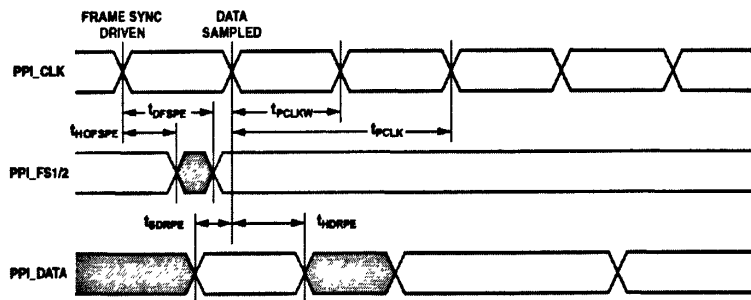


图 4.5 PPI接口RX模式与片内时钟同步时序图

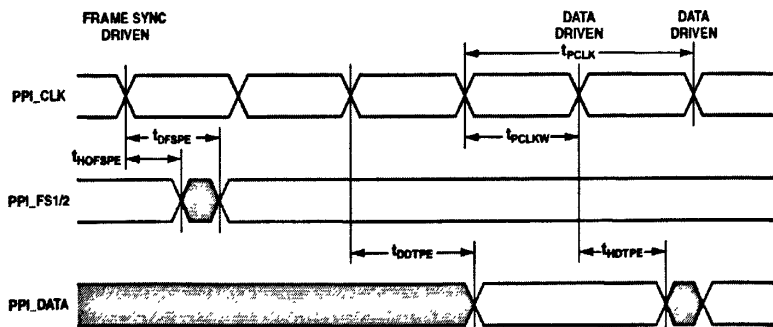


图 4.6 PPI接口TX模式与片内时钟同步时序图

PPI根据经A/D转换后的数据宽度设置PPI的数据线宽度为16位，由FPGA输出的水平同步信号和垂直同步信号分别输入PPI的PPI FSI和PPI FS2同步信号引脚。PPI_FSI置位表示一帧图像开始传输，PPI_FS2置位表示一行图像开始传输。图4.7为寄存器PPI_CONTROL为PPI控制寄存器各位。

本寄存器用来具体设置PPI接口的操作模式、控制信号以及端口带宽。个别位

需组合设置，本系统具体设置如下：

(1) POL[1: 0]用来进行PPI CLK选择，PPI FS1/PPI FS2极性选择，本系统不用设定PLOS、PLOC；

(2) DLEN[2:0]用来设定PPI端口宽度为8到16位。本设计中DLEN不能为大于I/O位的端口宽度，为0008bit，再合成为16bit数据；

6		5		4		3		2		1		0	
FLD SEL		PORT CFG[1:0]				XFR TYPE[1:0]				PORT DIR		PORT EN	
15	14	13	12	11	10	9	8		7				
POLS	POLC	DLEN[1:0]				SKIP EO		SKIP EN				PACK EN	

图4.7 PPI_CONTROL寄存器

(3) SKIP EN为1时，PPI会跳过数据，以节约DMA带宽。本系统将此位置0；

(4) PACK EN位控制PPI数据打包，把两个8位数据合成16位进行传输。本文PACK EN置1；

(5) FLD SEL位在当“active field only”选中的ITU-656模式下有效，决定只传送每帧的Field 1还是Field 1、Field 2都传送。本系统两场数据都要传送，FLD SEL置1；

(6) PORT CFG[1: 0]设置PPI的同步操作模式，ITU-656模式可不设置；

(7) PORT DIR设定数据输入输出方向，数据输入时置0，输出时置1；

(8) XFR_TYPE[1: 0] 设置行操作模式，次位设置成[1, 1]特指GP模式，此处设置为[1, 1]之外的任何值皆可；

(9) PORT EN位为PPI使能位，其它PPI设置要在设置本位之前设置。

在FPGA内做PPI接口驱动IP核，便可完成和DSP的视频流数据交互，设计的VHDL代码行为仿真结果见图4.8、图4.9，结果说明改驱动程序满足PPI接口规范和电气特性。

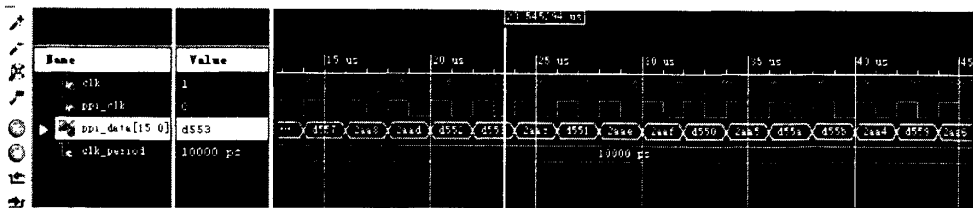


图4.8 PPI接收驱动程序行为仿真结果

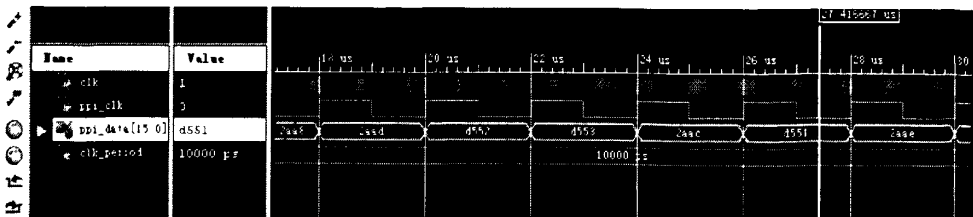


图4.9 PPI发送驱动程序行为仿真结果

4.2.2 DMA 传输配置

DMA控制器是Blackfin DSP架构中的重要组件，它完全独立于内核，无需占用内核周期而直接从外设调用数据。当跟踪图像数据输入DSP进行处理时，如数据量较大，传输时间就比较大，使用DMA传输功能，使数据传输在DMA控制器控制下完成，DSP则进行其它的运算处理，待DMA传输结束，再由DSP对输入的数据进行处理，就把数据传输和数据处理并行进行，减少了处理时间，提高了实时性。本系统中，FPGA将一行数据缓存在存储器中，BF533则采用按行DMA的方式，将数据传输到DSP内部，待DMA完成时产生中断，在中断服务程序中设定DMA的目的地址，指向下一行所在存储区，以完成数据在DSP内部连续存放^[37]。

BF533的DMA控制器可控制六种数据传输：内部存储器与内部存储器、内部存储器与外部存储器间、存储器与SPI接口之间、存储器与SPORT接口之间、存储器与UART接口之间以及存储器与PPI接口之间。

每个DMA通道都有独立外设的一些功能，本项目中所使用的与PPI连接的DMA是默认的DMA0，DMA配置寄存器DMA_CONFIG见图4.10，描述了这一寄存器的具体位设置。

5				4		3		2	1	0
重启 DMA buffer 清除				WDSIZE[1:0]				WNR		DMA EN
15	14	13	12	11	10	9	8	7		6
FLOW[2:0]				NDSIZE[3: 0]				DI_EN		DI_SEL

图4.10 DMA配置寄存器

当数据输入时，各位设置如下：

- (1) 设置DMA0_CONFIG为0xFFC00C08选择DMA 0配置寄存器；
- (2) 设置DMAEN位为0x00000001使通道使能；
- (3) 设置DI_EN位为0x00000080使数据中断使能；
- (4) 设置WNR为0x00000002确定通道方向；
- (5) 设置WDSIZE_16位为0x00000004选择字长为16bit；
- (6) 设置DMA2D位为0x00000010选择2D方式；
- (7) 设置RESTART位为0x00000020选择重新开始；
- (8) 设置DMA_LIST_SMALL位为0x6000选择DMA描述符小工作模式；
- (9) 设置NDSIZE_4位为0x0400设置描述符大小为4。

当数据输出时，配置与输入基本相同，只有DMA方向与DMA使能不同。

4.3 视频采集、存储与输出

4.3.1 I²C 总线配置

I²C(Inter-Integrated Circuit)总线是由飞利浦公司开发一种连接CPU与外围设备的两线式串行总线，由数据线SDA和时钟SCL构成，可发送和接收数据，在CPU与外围IC间、IC与IC间进行双向传送，最高速率可达100kbps。CPU控制信号由地址码和控制量组成，地址码用来接通要控制的电路，控制量确定要控制的量，这样，各器件虽并联于同一条总线上，却互相独立，图4.11为I²C总线配置图。

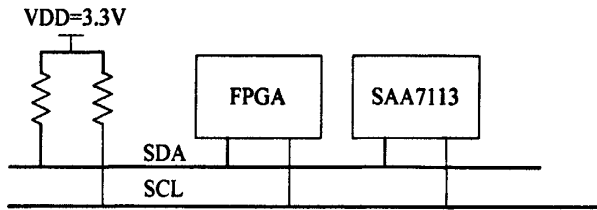


图 4.11 I²C 总线配置图

图 4.12 为 I²C 总线时序图。

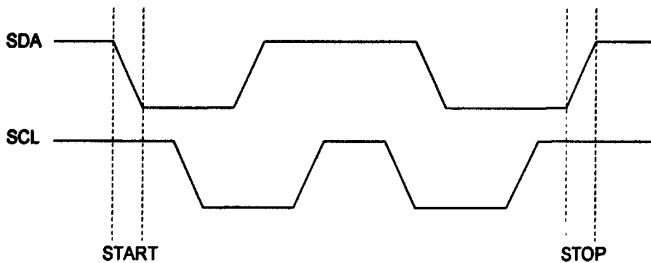


图 4.12 I²C 总线时序图

由图 4.12 可知 I²C 总线共有五种信号：

- (1) 开始信号：SCL 高电平期间，SDA 从高电平向低电平跳变，数据传送开始；
- (2) 结束信号：SCL 低电平期间，SDA 从低电平向高电平跳变，数据传送结束；
- (3) 应答信号：接收数据单元接收到 8bit 数据之后，向发送数据单元发回一个特定低电平脉冲，确认已收到数据。或 CPU 向 IC 发出一个信号，等待受控 IC 发回一个应答信号，CPU 接收到该信号后，作出是否继续发送信号的判断。如未收到应答信号，则判断为受控 IC 发生故障；
- (4) 写信号：开始进行数据写入；
- (5) 暂停信号：数据暂停写入。

由此，SA7113 的 I²C 总线传输数据状态转换过程如图 4.13 所示。

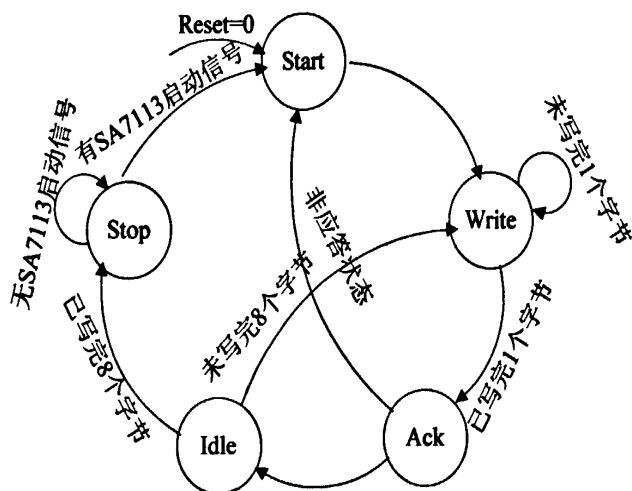


图 4.13 I²C 总线传输数据状态转换图

状态机的编码方式关系逻辑设计性能，因为 FPGA 触发器资源丰富，此处选用 One-Hot 编码方式，通过多用触发器来简化组合电路，提高了器件资源的利用率同时也提高电路的运行速度以及可靠性。部分程序如下，首先状态定义：

```

Parameter [19:0] Start0=20'b0000_0000_0000_0000_0001;
Parameter [19:0] Start1=20'b0000_0000_0000_0000_0010;
Parameter [19:0] Start2=20'b0000_0000_0000_0000_0100;...
    
```

```

接着，发送开始信号： Start0: begin SCLOUT=1'b1;SDAOUT=1'b1;
                        NS=Start1;end...
    
```

```

发写信号： Write0: begin SCLOUT=1'b0;SDAOUT=SDA_TEMP;
                        NS=Write1;end...
    
```

本文在 Xilinx 的 EDK 软件开发平台下，调用 XPS I²C Bus Interface 模块，调用并配置 I²C 协议 IP 核，综合后见图 4.14, xps_iic 作为一个 mb_plb 的 SLAVES-A。

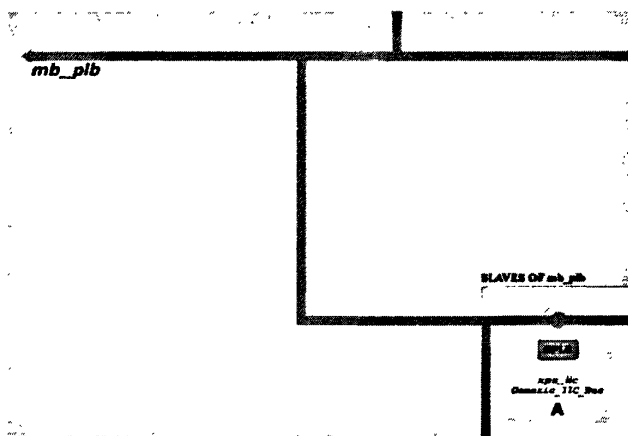


图 4.14 I²C 从器件模块

根据图 3.1 的总体框图，设计的 RS232 口对 SAA7113 等 I²C 从器件进行读写，

结果如图 4.15 所示, 此接口的成功设计, 可方便 FPGA 对视频采集和输出芯片进行寄存器配置。

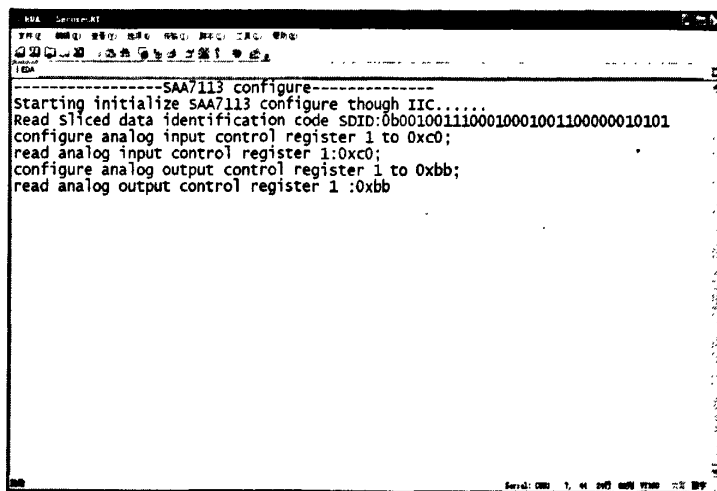


图 4.15 I²C 从器件调试结果图

4.3.2 视频采集电路

SAA7113 芯片对 CCD 的视频/图像数据采集, 转换出的视频数据信号, 按照如图 4.16 所示的时序要求, 在 27MHz 像素时钟 LLC 下, 经过 VPO[0,1...7]并行送出, 其中 RTS1 (DOT) 信号为 SAA7113 内部 RTC 时钟中断等信号的输出, 本课题未使用其 RTC 功能, 故未对其做驱动接口。

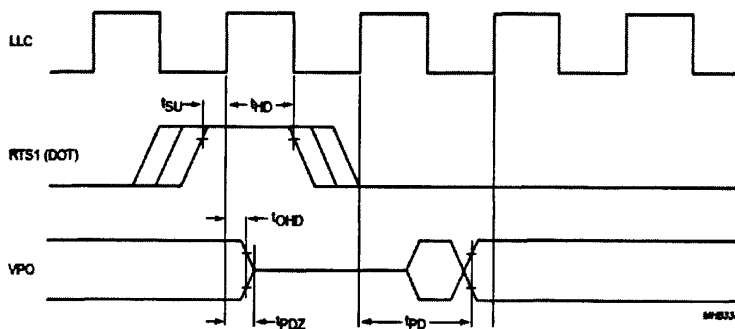


图4.16 SAA7113时序图

在 Xilinx 的 ISE 逻辑开发工具上面, 做出 SAA7113 的视频采集驱动 VHDL 程序, 仿真结果如图 4.17、4.18 所示, 和图 4.16 对比, 说明了时序设计的正确性。

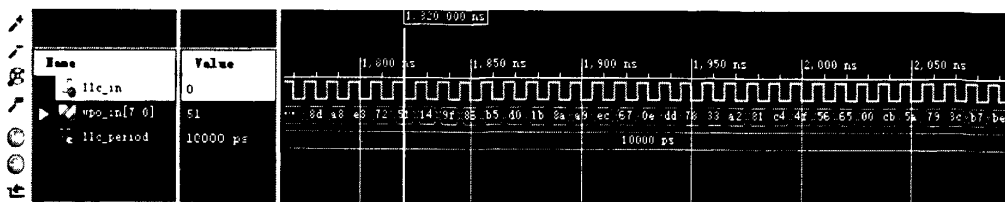


图 4.18 SAA7113 行为仿真总线结果

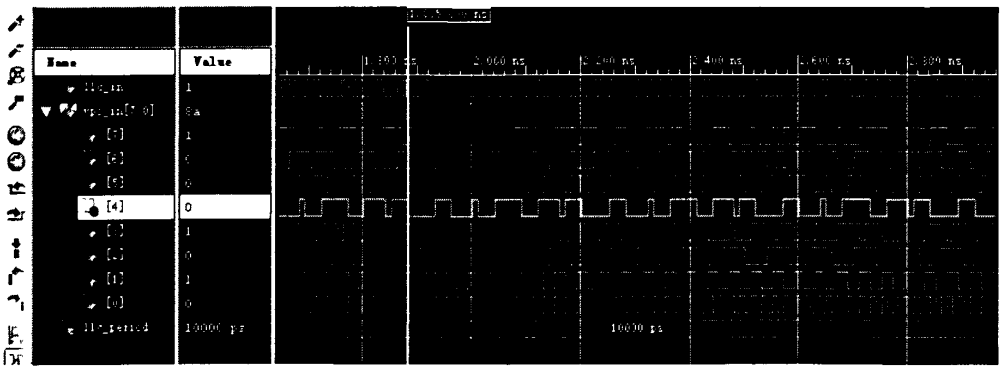


图 4.17 SAA7113 行为仿真分解结果

SAA7113 的寄存器配置如表 4.3 所示。

表 4.3 SAA7113 寄存器配置

地址	功能	值 (hex)	地址	功能	值 (hex)
01	Increment delay	08	0B	亮度对比度	47
02	模拟输入控制 1	C0	0C	色度饱和度	40
03	模拟输入控制 2	33	0E	色度控制	01
06	水平同步开始	E9	0F	色度增益控制	2A
07	水平同步停止	0D	40	Slicer 控制字	02
08	同步控制	98	41~57	行控制寄存器	FF
09	亮度控制	01	59	水平 Slicer 偏置	54
0A	光亮度	80	5A	垂直 Slicer 偏置	07

SAA7113 初始化完成后,便可采集 CCD 的模拟视频信号并输出解码后的 YUV 格式数据,数据格式如图 4.19 所示。

消隐期		时序同步码				720 个像素 YUV 4:2:2 数据						时序同步码				消隐期		
80	10	FF	00	00	SAV	U0	Y0	V0	Y1	...	V18	Y19	FF	00	00	EAV	80	10

图 4.19 SAA7113 输出的 YUV 数据格式

上表中,“80, 10, ...”是消隐期信号,“FF, 00, 00, SAV(EAV)”为时序同步码,代表有效数据的开始和结束。每行 720 个像素数据中,处于两个时序同步码之间的数据,才是有效数据。

通常 I²C 总线传输的数据由外部存储器存储起来,然而数据相对固定且数据量不太大时,可将初始化的数据写在程序中,可减少频繁数据交换从而使操作简化。SAA7113 初始化可采用这种方式,通过检测应答信号来改变输入值:

```
Always @(posedge CLKIN)begin//Input Data for Initialization.
```

```
Case (ACK_CNT)//ACK_CNT is a counter for acknowlege.
```

```
8'h00:DATA_REG<=8'h48;//-Slave Address+Wt
```

```
8'h01:DATA_REG<=8'h00;...
```

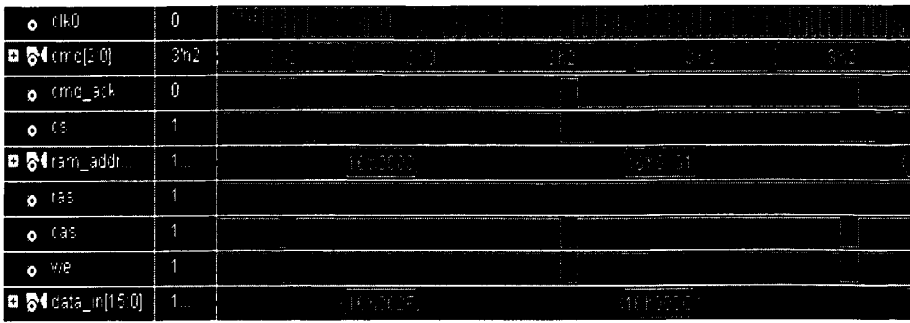



图 4.22 FPGA 的 SDRAM 写数据时序图

只写的^[40]。系统上电后，通过 I²C 总线对 SAA7121 进行初始化，以接收 YUV 格式数据。SAA7121 将接收的数据进行编码处理，组合在一路模拟信号并输出。初始化的过程与 SAA7113 类似，这次不再赘述。SAA7121 的功能时序图如图 4.23 所示。

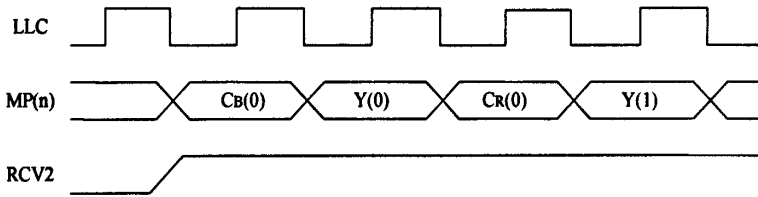


图 4.23 SAA7121 功能时序图

在 Xilinx 的 ISE 逻辑开发工具上面，做出其 VHDL 程序，仿真结果如图 4.24、4.25 所示，和图 4.23 对比，说明了时序设计的正确性。

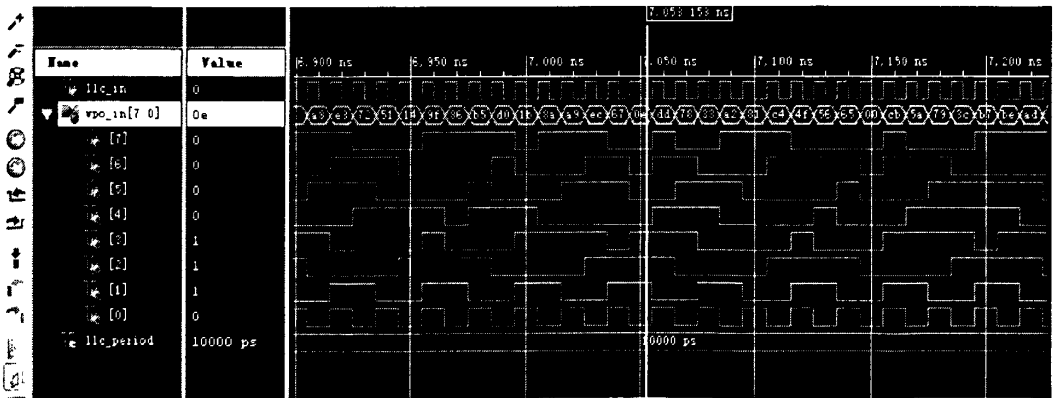


图 4.24 SAA7121 的行为仿真分解结果

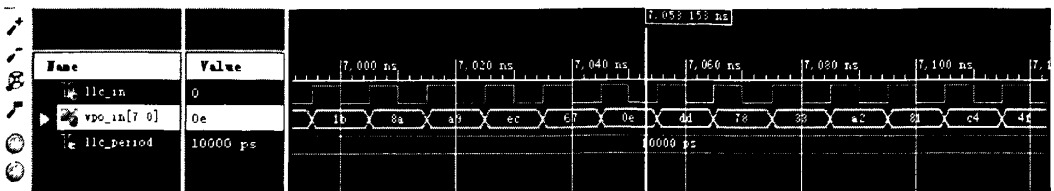


图 4.25 SAA7121 的行为仿真总线结果

4.4 本章小结

本章主要介绍了运动目标检测与跟踪系统的硬件调试。包括硬件系统的初始化、系统时钟配置、通信控制寄存器的设置、视频采集存储输出相关寄存器的设置，给出了仿真结果。在中断服务程序和跟踪窗口处理函数中设置计数器，证明系统可以达到25帧/s的处理速度，能满足实时性要求。

第 5 章 运动目标检测与跟踪系统软件设计

一个完整的视频图像处理系统包括硬件系统和软件设计两部分，前者为系统提供运行平台，后者在前者的基础上完成系统具体功能的实现。本文首先介绍了软件平台和总体开发思路，接着介绍了算法的软件设计思路，重点研究了目标检测与跟踪算法的设计。

5.1 软件开发环境与流程

5.1.1 软件开发环境

ADI 公司为 BF533 提供了方便的 Visual DSP++ 开发环境和完备的仿真器。用户可通过 Visual DSP++ 项目管理环境来开发、调试程序，这个环境提供了代数语法编译器、建库归档器，链接器、C/C++ 编译器和包括 DSP 和数学函数的 C/C++ 运行库。其中编译器可以高效率地把 C/C++ 代码编译为 Blackfin DSP 汇编代码^[33]。

图 5.1 为 Visual DSP++ 的开发环境界面。

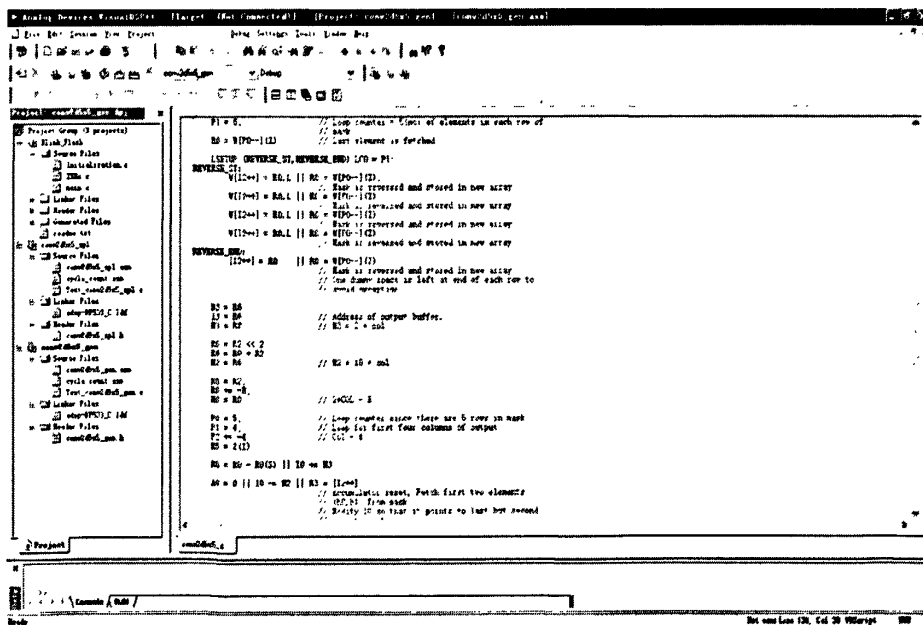


图 5.1 Visual DSP++ 开发环境界面

5.1.2 软件开发流程

将汇编、C/C++ 等格式的源代码用编译器、编译器加以处理，可得到 ELF 格式的目标代码，再经连接器得到可执行代码，过 JTAG 接口调试通过后，使用 loader 工具把可执行代码转化为可引导的加载文件，用 FALSH 加载工具把写可引导加载文件写入 FALSH 芯片，当 DSP 启动时即可进行 FALSH 引导，系统便可脱离 PC

独立运行。BF533 软件开发流程框图见图 5.2。

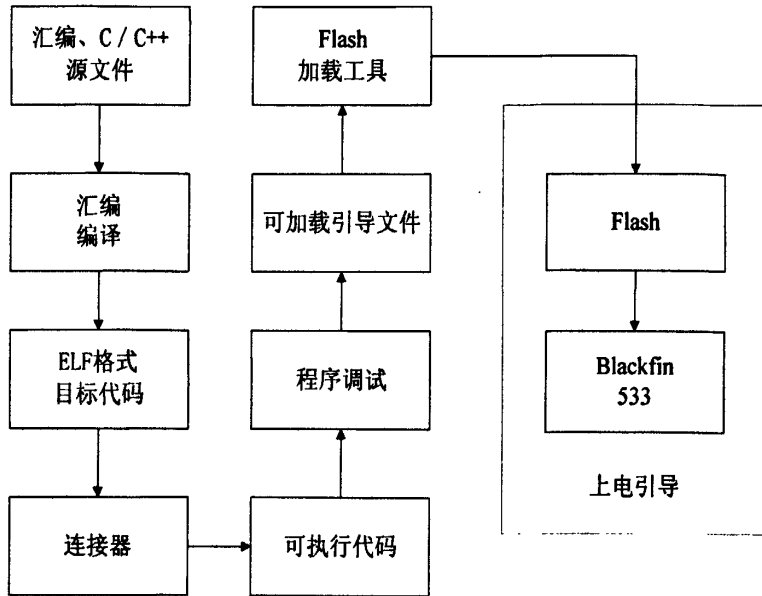


图 5.2 BF533 软件开发流程框图

5.2 算法软件设计

5.2.1 系统初始化

图 5.3 为软件流程图，首先进行 DSP 初始化设置即配置 DSP 时钟和中断，使

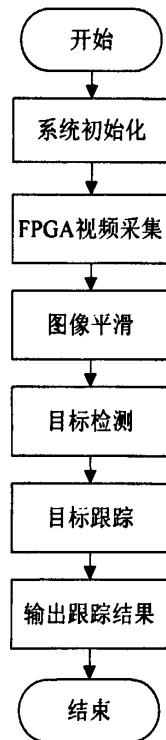


图 5.3 软件流程图

DSP 一直运行在监控模式，接着配置 DMA 传输和 EBIU 接口、PF 端口；然后向 FPGA 发送视频数据，FPGA 进行数据采集后将数据流传输至 BF533 进行检测与跟踪算法处理，最后输出跟踪结果。

5.2.2 图像平滑

实际获得的图像数据在形成、传输和处理的过程中，存在着各种干扰，如光电转换时元件灵敏度不均匀性、数字化时的量化噪声、传输误差以及人为因素等。图像平滑即图像去噪，主要目的是去除实际成像过程中，设备和环境因素造成的图像失真。图像平滑的本质是低通滤波，让低频的信号部分通过，而阻截高频的噪声部分。

目标检测一般要进行参考图像或背景图像的建模，即将一帧已知图像作为参考图像或背景图像，背景提取一般是利用背景图像的不变性或缓变性^[34]。背景模型是基于像素的，即假设每个背景像素的值都是一个独立的随机过程，那么在较长的时间内，背景中每一个像素的颜色值稳定分布在某一固定邻域内，故其灰度值的分布可用高斯分布来表示，即背景中每个像素 P_{ij} 的灰度值 x 服从均值为 μ_{ij} ，白方差为 σ_{ij}^2 的高斯分布，即

$$f(x) = \frac{1}{(2\pi)^{\frac{1}{2}} \sigma_{ij}} e^{-\frac{(x-\mu_{ij})^2}{2\sigma_{ij}^2}} \quad (5.1)$$

式(5.1)中，均值和方差已知，则高斯分布已定，此时构造背景模型的任务就是估计出背景中每个象素的均值和方差。可以通过背景的样本序列来估计背景模型中每个像素的均值和方差。采用背景图像序列估计背景模型的参数，样本均值估计和方差估计为

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (5.2)$$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2} \quad (5.3)$$

其中， x_i 为第 i 幅图像的灰度值。在假定加性噪声是随机独立分布的前提下，采用邻域的平均或加权平均可有效抑制噪声干扰，平滑模板的思想是将一点和周围八个点做平均来去除发生突变的点，从而滤掉噪声，其代价是图像会有一定程度的模糊。可认为距离某点越近的点对该点的影响越大，引入了加权系数的高斯

模板，为 $\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ ，可以看出，距设图像位置较近的点被赋予了较大的权

值，靠近边缘的点被赋予了较小的权值，这样就在一定程度上体现了位置信息。图5.4为高斯上述高斯模板在 $\sigma=1$ 时的频域特性。

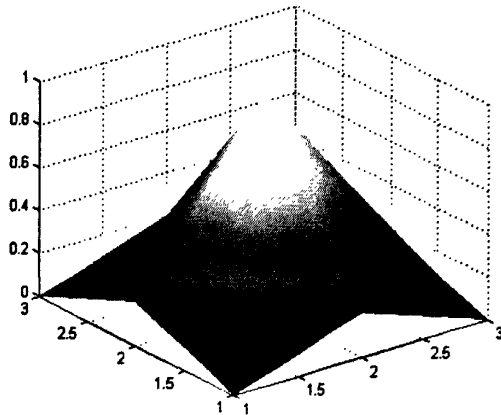


图 5.4 $\sigma=1$ 时的高斯滤波器频域特性

用上述模板对图像进行平滑处理，图5.5为高斯模板平滑处理前后的效果对比图。



图5.5 高斯模板平滑效果图

5.2.3 目标检测

在视频监控系统中，运动目标检测处于最底层，是各种后续高级处理的基础。对固定摄像头，背景建模是实时区分运动目标和图像背景的有效方法。本文选择帧间差分法作为系统的处理算法。根据第 2.2.1 节基本原理的介绍，帧间差分法处理的基本流程如图 5.6 所示。

处理过程为，先读入一帧图像作为初始图像并转化为一组数据，并创建RAM空间，将初始帧图像数据存储起来。再依次读入相邻帧图像的像素数据，每读入一个像素数据就与前一帧图像中相应像素数据进行差分运算、阈值判断和二值化处理。同时，用新一帧图像的像素数据更新RAM中存储的前一帧数据，保证RAM中的图像数据始终是当前帧图像的前一帧的数据。一帧数据处理完成后输出一维

数据处理经过，并同时处理下一帧图像。

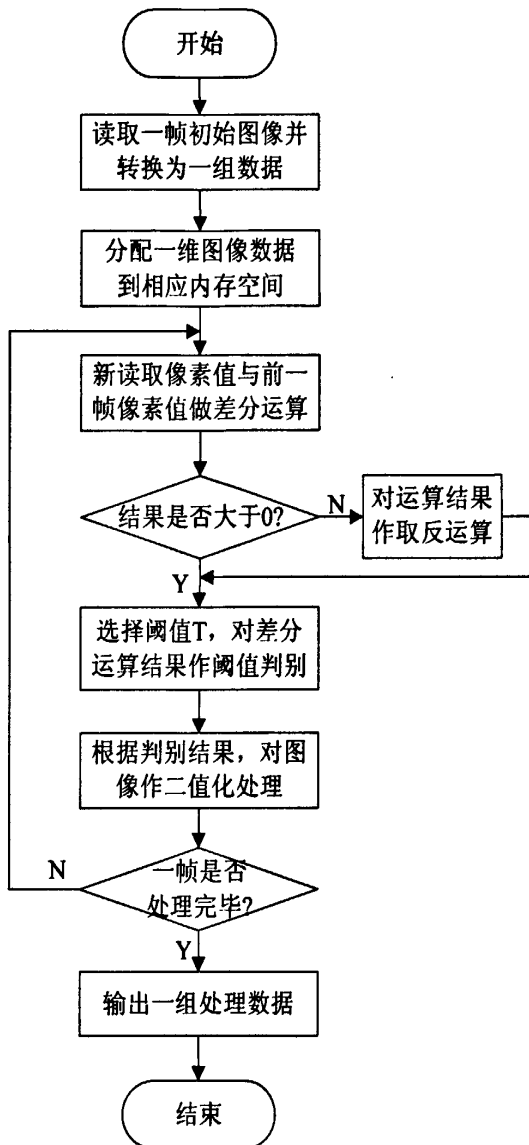


图 5.6 帧间差分法处理流程图

根据2.2.3节，将第k帧图像与第k+1帧图像进行比较并差分运算，对运算结果进行二值化得到如图所示的二值图。二值图像中对所有对象描述的像素值相同，要对不同的目标进行区分，只能通过像素间的连通关系。连接关系主要分为：

- (1) 四连接：当前像素为黑，且其四个相邻像素中至少有一为黑；
- (2) 八连接：当前像素为黑，且其八个邻近像素中至少有一为黑。

连接在一起的黑色像素的集合称为连通域，对一个如图5.7所示的二值化图，四连接关系下共六个连通域，八连接关系下共二个连通域。因不同连通域代表不同目标，故统计连通域个数，即可得目标物个数。为了加以区别，需对不同的连通域设一个二值矩阵表示一个黑白图像，为讨论方便，令“黑=1”、“白=0”则图5.7中的二值化图可表示为矩阵f。

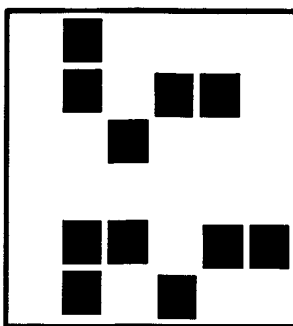


图5.7 二值化图

$$f = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.4)$$

之后对检测到的帧进行连通性分析，再进行判别，若某一区域中满足条件的像素点数超过一定值，则检测到运动目标，且认为该连通区域是运动目标所在的区域。加入 2.2.3 节所述自适应的模板后，检测实验结果图见图 5.8，可知当视频帧中出现运动目标时，能较好地检测到运动目标。

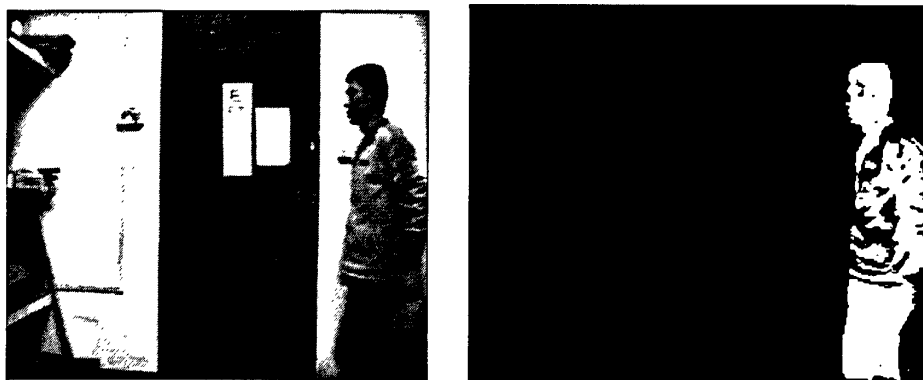


图5.8 检测实验结果图

5.2.3 目标跟踪

5.2.3.1 目标跟踪整体软件设计

Meanshift 目标跟踪算法流程图见图 5.9，跟踪算法软件设计如下：

- (1) 建立目标模型：计算初始帧中目标区域像素点特征值在特征空间的概率；
- (2) 对当前帧建候选模型：计算当前帧中候选区域内的所有像素点特征值在特征空间的概率；

(3) 用相似性度量函数计算初始帧与当前帧的相似性，通过计算相似性度量函数极大值即得到目标相关的均值偏移向量，根据 Meanshit 算法的收敛性，迭代计

算均值偏移向量，最后会收敛到目标在当前帧的位置，达到跟踪的目的。

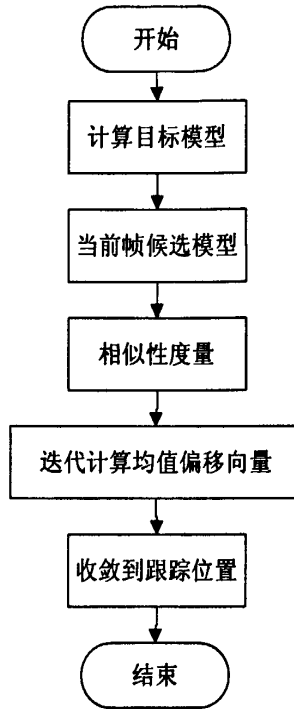


图 5.9 Meanshit 目标跟踪算法流程图

5.2.3.2 相似性度量与窗口跟踪

用 MeanShift 目标跟踪算法对模型帧与待检测帧进行概率密度的相似性度量，实验为对运动的行人进行了跟踪，图 5.10 为跟踪实验结果。图中的行人从右向左移动，其帧频为 25PFS。图中共四幅图像，相邻的两幅图像相差 10 帧，用框标记出运动目标。

从图 5.10 可以看出利用 MeanShift 目标跟踪算法虽然能跟踪到运动的人体，但是第 21 帧存在较明显的偏差。因运动目标的角度、光线发生较大变化时，模板更新并没有能够跟上这些变化，导致匹配效果变差。

5.2.3.3 加入 Bhattacharyya 系数的相似性度量

由 2.3.2 节所述，本系统设计一种加入 Bhattacharyya 系数的目标位置度量算法的 MeanShift 目标跟踪算法，流程图见图 5.11，当 Bhattacharyya 系数最大时即定位为最终中心位置，停止迭代，该算法优于上述改进前的算法，具有稳定、实时性好的特点。

用该方法对同一组视频帧进行跟踪实验，从得到的实验结果如图 5.12 所示，通过对比可以看出，改进的算法跟踪效果明显好于改进之前的效果，随着时间的推移也能较好的对目标轨迹进行跟踪。

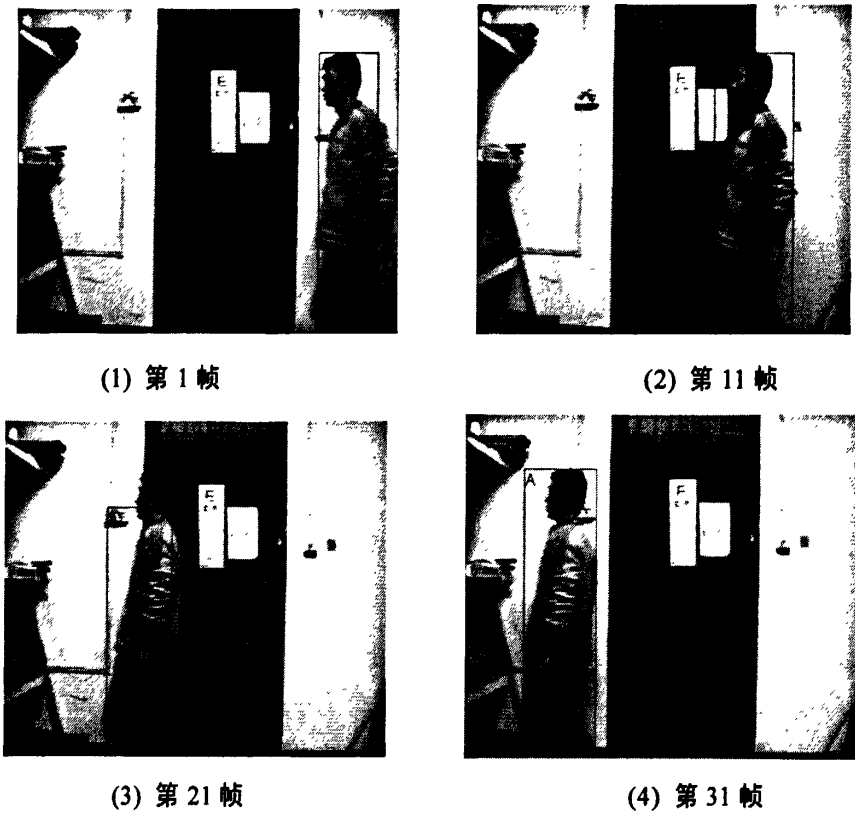


图 5.10 Meanshift 目标跟踪效果图

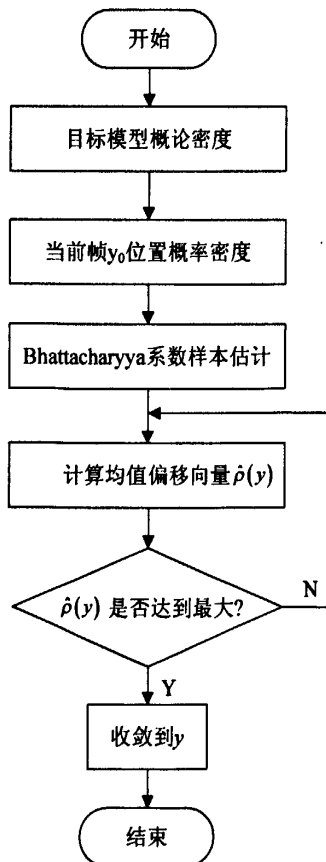


图5.11 加入Bhattacharyya系数的Meanshift跟踪算法流程图



(1) 第 1 帧



(2) 第 11 帧



(3) 第 21 帧



(4) 第 31 帧

图5.12 加入Bhattacharyya系数的Meanshift跟踪算法实验结果图

5.2.4 优化原则与方法

1 C 语言、汇编语言配合使用

基于Visual DSP++开发环境的软件构建中，C语言比其他支持人工组合的编程语言效率高很多，编写好的C代码可直接导入Blackfin DSP，并快速执行运行测试，维护也更简便。为找出耗时较多的代码段，可使用Visual DSP++提供的profiling工具，profiling可以实时统计各个子程序大致占用的指令周期数，并显示出来，此时对占用时间很多的函数进行优化往往可以大大提高运行效率。Visual DSP++可以通过两种方式进行汇编语言与C语言混合编程：

(1) 使用汇编语言设计子函数，其参数和返回值的传递符合Visual DSP++C语言运行环境，在功能较独立且使用汇编时程序较长时使用；

(2) 内联汇编，将汇编语言嵌入C程序内部，在C语言不方便表达的低层操作时使用。

2 程序异常点排除

由于使用汇编语言要自己管理存储器等低层操作，在调试时，常因地址没对齐等原因而使DSP进入异常服务程序。当程序较多时，就很难发现导致异常的指令。

Blackfin内部集成了trace工具,用硬件保存最近16次程序跳转,在异常中断服务程序中设置断点,这样就很容易找到异常发生点。假设第0条指令为进入异常前最后一条指令,第1条指令为异常中断服务程序断点处的指令,则可以断定第0条指令的地址没有对齐。

此外还可以通过 PPI 状态寄存器 PPI_STATUS 提供的当前 PPI 运行状态信息来排除异常点。PPI_STATUS 中 FLD 在 ITU-656 模式下状态为置位或清零,在 PPI 输入模式下才有效。FLD 的状态反映了当前 ITU-656 模式信号的状态。当设置 FLD=0,表示正在接收视频场 1;若设置 FLD=1,表示正在接收视频场 2。PPI-STATUS 设置为 0xFFC0 1004,程序正常时,不必读出,当运行故障时,将 PPI-STATUS 的值读出,可以方便排除 PPI 出错的部分。

3 中断点、监视点配置

嵌入式系统中,中断配置极其重要,为每一个事件,提供一个返回寄存器,对它而言,一个中断也是一个事件,当将返回地址保存到堆栈中时,就会启用优先级更高的中断。DSP响应外部中断和内部中断,进入相应的中断服务程序,即在各个跟踪窗口后沿中断到来时,调用相应图像处理函数,提取出目标所在位置,场中断服务程序中将其显示出来。因此,中断配置保证系统能有条理地处理各个跟踪窗口,是系统正常运行的关键。

BF533采用两级事件控制机制,系统中断控制器SIC和内核事件控制器CEC共同管理所有的系统中断及其优先级。SIC提供多个外设中断与内部中断之间的映射,某个中断可以在SIC中通过编程屏蔽掉。CEC控制九个通用中断(IVG7-IVG15),其中包括专用中断和异常事件中断。仿真、复位、异常事件、硬件错误和内核定时器中断请求不受系统级的中断寄存器组的影响。根据图像处理需要,本系统需要三个外部中断:行中断、跟踪窗口后沿中断和场中断,还要一个DMA结束中断。外部中断可通过PF进入DSP中断控制器。

因PF只能产生FlagA和FlagB两种中断,要识别三个外部中断,就需要把其中两个外部事件映射到同一中断,再判断发生了哪一个中断。本文将场中断映射到FlagA,将行中断和跟踪窗口后沿中断映射到FlagB。在FPGA中设计一个中断寄存器,当外中断发生时,就将相应位置位。DSP则操作两个变量:win_come和win_flag,win_flag是FPGA中断寄存器映像,win_come标志是否有一个跟踪窗口后沿中断到来,在FlagB中断服务程序中,若为跟踪窗口后沿中断,则win_come置位,此时BF533读取FPGA的中断寄存器,并将其清零,读取值与win_flag进行逻辑或操作后存入win_flag,接着DSP判断哪些跟踪窗口后沿到来,并调用相应处理函数,中断响应流程图见图5.13。

监视点和断点相似,监视点可添加一定条件,让程序在满足条件时停止运行,以便观察各寄存器和计算结果。如在没有操作系统的DSP程序中,各个段之间没有

提供保护，某些关键数据可能被其它程序无意中更改，此时，用监视点便可以找出误更改数据的指令。

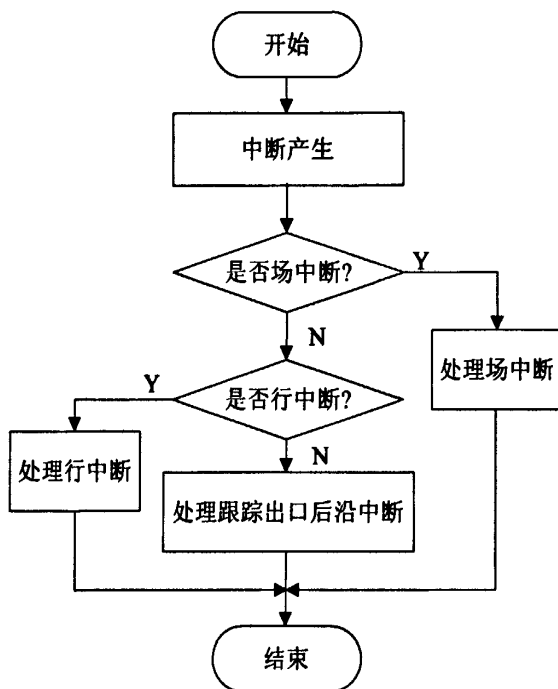


图5.13 中断响应流程图

5.3 本章小结

本章主要介绍了运动目标检测与跟踪系统的软件设计与系统优化。首先介绍了 Blackfin DSP 开发环境 Visual DSP++ 及其配置。然后给出软件整体流程图，进行各功能模块设计包括图像平滑、图像比较、模板更新、目标匹配跟踪等。利用所设计算法对视频数据进行处理，给出处理效果图，通过对运动的手机进行跟踪，表明本系统所采用的算法能够较准确地跟踪其轨迹，最后给出程序的优化原则与方法。

总结和展望

伴随计算机与通讯技术发展,智能视频监控系统的应用越来越普遍。运动目标检测与跟踪作为整个系统的第一步,作用非常重要,可以为更高层的运动目标行为理解和分析提供基础信息。运动目标检测与跟踪处理的实时性和准确性对监控系统功能实现意义重大。

针对视频处理任务数据量大和算法实现的实时性要求,本文利用 Blackfin DSP 并行处理和硬件实现的特点,设计了基于 ADI 公司 BF533 平台的视频处理系统,并在此基础上实现运动目标检测与跟踪算法。

1 主要研究工作

本课题主要完成的工作有:

(1) 研究了运动目标检测与跟踪技术的发展现状,对各算法进行分析比较,并利用 ADI 的 Visual DSP++ 开发环境和完备仿真器工具,在 BF533 上实现了基于帧间差分法的运动目标检测算法与 Meanshift 运动目标跟踪算法;

(2) 设计了基于 BF533 的视频处理系统硬件平台。从系统硬件的总体结构进行了完整的硬件设计,包括通讯管理模块、视频采集输入模块、存储器模块和视频输出模块的电路原理图设计等;

(3) 完成了系统硬件各模块的初始化配置和调试,主要包括 PPI 接口的设置、DMA 传输的配置、I²C 总线的设置以及 YUV 数据格式的处理和 SDRAM 存储模块的调试、视频编解码芯片的配置和控制;

(4) 分析并设计了 BF533 视频处理系统的软件平台,描述了软件的开发平台和总体设计思路,完成了目标检测和跟踪的算法设计,主要包括图像平滑、模板生成、图像比较及模板更新等算法的设计;

(5) 完成了视频处理系统的总体测试,通过测试结果分析表明所选算法和各模块能实现相应的功能,具有一定的实时性和准确性,达到了设计要求。

2 研究工作展望

通过本文的算法研究、软硬件设计和调试,本人对基于嵌入式的智能视频图像处理系统设计和开发的认识和理解更加深刻,科研水平也得到明显提高。由于水平和时间所限,系统尚存在许多不完善的地方,还可以得到完善,主要表现在以下几个方面:

(1) 本系统所设计基于 BF533 的视频图像处理系统作为一个研发平台,虽然各功能模块能够实现,但仍处于研究验证阶段,还无能力形成一个产品投放市场,要开发为实际产品或者嵌入到实际的智能视频监控系统中,还需在功耗、体积、

稳定性、可靠性等方面做进一步改进；

(2) 对基于 DSP 平台的算法，程序代码优化特别重要，进行性能优化时，不仅可以用软件编译器进行自动优化，也可对算法的可优化部分进行具体优化而且效果会更好，本文的研究对这部分工作不足，在提高算法效率方面还可以做进一步的研究；

(3) 针对帧间差分法与 Meanshift 算法存在的不足，本文还可以做深入一步的研究。如克服帧间差分法带来的空洞现象，以及基于基于多特征融合的 Meanshift 算法。已有不少学者对此类问题提出了解决方案，但具体实现的还较少。本文可在实际应用方面做进一步的探索与实践；

(4) 在完成运动目标检测与跟踪的基础上，还可以进行运动目标的行为分析和行为理解等视频图像处理的高层次研究，使系统真正具有自主性和智能性；

(5) 视频图像处理因计算量较大而要求较高的运算速度，为此可考虑选用性能更高的 DSP 处理器，如双核的 BF561 以及最新的 BF592 等；

(6) 用一片 FPGA 配置大容量存储器实现整个系统的设计方案应该可行，而且应该比 DSP+FPGA 更简洁，也应该可以达到更高的运行速度，但对 FPGA 的规格要求较高，课题没有进行实际设计。

参考文献

- [1] 刘建华. 视频监控发展道路之我见. 电视技术, 2009, 33(5): 62-63
- [2] Maria Sagrebin, Josef Pauli. Real-Time Moving Object Detection for Video Surveillance. In: Advanced Video and Signal Based Surveillance, IEEE, NewYork, 2009: 31-36
- [3] 侯志强. 韩崇昭. 视觉跟踪技术综述. 自动化学报. 2006, 32(4):603-617
- [4] Patricia P. Wang, Wei Zhang, Jianguo Li, et al. Realtime Detection of Salient Moving Object: A Multi-Core Solution. In: ICASSP. IEEE, London, 2008: 1481-1484
- [5] GAO Cong. Investigation and Analysis of Arithmetic for Human Motion Detection and Tracking[J]. Industrial Control Computer, 2008, 21(10): 49-50
- [6] Nguyen Q A, Robles-Kelly A, Shen C. Enhanced Kernel-based Tracking For Monochromatic and Thermographic Video. In IEEE International Conference on Advanced Video and Signal Based Surveillance, London, 2006: 51-53
- [7] Chappelle O, Scholkopf B, Zien A. Semi-Supervised Learning. Cambridge, MA: MIT Press, 2006
- [8] 陈峰. 基于 Blackfin DSP 的数字图像处理. 北京: 电子工业出版社, 2009:11-12
- [9] LI Min. JIANG Jian-chun. The Study of Image Edge Detection Based on Erode Algorithm [J]. Computer Applications and Software, 2009, 26(1): 82-84
- [10] LIU Kai-gang, XU Mei-sheng, LI Wei. Algorithm for SAR Image Target Detection Based on Double Threshold Segmentation. Foreign Electronic Measurement Technology, 2008, 27(3): 3-6
- [11] 张琦, 胡昌华, 乔玉坤. 基于权值选择的粒子滤波算法研究. 控制与决策, 2008, 23(1): 894-896
- [12] Liang D, Huang Q, Jiang S et al. Mean-Shift Blob Tracking with Adaptive Feature Selection and Scale Adaptation. IEEE International Conference on Image Processing, Xian, 2007, 3: 369-372
- [13] YIN Yong. WANG Ya-fei. Moving Object Detection Based on Spatial Local Correlation. Opto-Electronic Engineering, 2009, 36 (2): 1-5
- [14] KANG Wen-jin, DING Xue-mei, LIU Gong-lian, Bandwidth-adaptive mean shift tracking algorithm. Journal of Optoelectronics Laser, 2008, 19(1): 135-139
- [15] Chateau T, Belilille V, Chausse F, et al. Real-Time Tracking with Classifiers,

- Lecture Notes in Computer Science, Heidelberg:Springer Berlin, 2007: 218-231
- [16] Dorin Comaniciu, Visvanathan Ramesh, Peter Meer. Kernel-Based Object Tracking[C]. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 25, NO. 5, MAY2003: 564-577
- [17] 陈浩, 彭先蓉. 基于 ADSP BF533 平台的实时多目标跟踪. 成都, 中国科学院光电技术研究所. 2007
- [18] 苏磊, 张登福, 刘涛. 基于 FPGA 的 RGB 到 YUV422 的数字视频转换. 实用技术, 2009, 33(S2): 105-106
- [19] 田启川, 刘正光, 李临生. 基于序列图像的虹膜特征提取与分类. 天津大学学报, 2007, 40(12): 1441-1446
- [20] Wang Jiang-tao, Yang Jing-yu. Object Tracking Based on Kalman-Mean Shift in Occlusions. Journal of System Simulation, 2007, 9: 4216-4220
- [21] Hu Jianhua, Yang Fan, Xu Jianjian. Motion detection based on blocks frame difference and background subtraction. Electronic Measurement Technology, 2007, 30(10): 14-17
- [22] Dedeoglu G Kanade T, S Baker. The Asymmetry of Image Registration and its Application to Face Tracking, IEEE Transactions. on PaRern Analysis and Machine Intelligence, 2007, 29(5): 807-823
- [23] Avidan S. Ensemble Tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(2): 261-271
- [24] Sundaramoorthi GYezzi A, Mennucci A. Coarse-to-Fine Segmentation and Tracking Using. Sobolev Active Contours. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(5): 851-864
- [25] 顾凤墙. 基于 MeanShift 的视频跟踪算法研究[西安电子科技大学博士学位论文]. 西安, 西安电子科技大学, 2009
- [26] Xilinx. Spartan-3E FPGA Family Complete Data Sheet. 2005: 2-66
- [27] Christophe Desmouliers, Erdal Oruklu, Jafar Saniie. FPGA-Based Design of a High Performance and Modular Video Processing Platform. IEEE, 2009: 393-398
- [28] Qian Hui-min. Mao Yao-bin, Wang Zhi-quan. Mean Shift Tracking-9with Self-updating Tracking Window. Journal of Image and Graphic, 2007, 12(2):245-249
- [29] Philips Semiconductors. 9-bit Video Input Processor SAA7113H Data Sheet. 1999: 3-8
- [30] Philips Semiconductors. Digital Video Encoder SAA7121H Data Sheet. 2002: 3-36

- [31] 柳笛, 黎福海, 闫旭. 基于 Blackfin DSP 的图像数据采集设计. 计算机系统应用, 2008, 1: 117-119
- [32] Xilinx. Platform Flash In-System Programmable Configuration PROMS Data Sheet. 2008: 1-24
- [33] W. Hu, T. Tan, L. Wang, and S. Maybank, A Survey on Visual Surveillance of Object Motion and Behaviors, IEEE Trans. SMC, 2004, 34 (3): 334-352
- [34] 钱伟康, 王力, 谢明. 基于 FPGA 的视频图像分割设计与应用. 测控技术, 2009, 28(z1): 101-102
- [35] Lei Y Ding X, Wang S. AdaBoost Tracker Embedded in Adaptive Particle Filtering. In Proc. of ICPR, New York: IEEE Press, 2006, 4: 939-943
- [36] 汪国有, 姜远利, 杨永祥. 基于 Blackfin DSP 的实时视频采集接口设计与实现. 计算机与数字工程, 2006, 34(1): 125-128
- [37] 张洁. Blackfin DSP 在数据高速采集中的应用[J]. 电子测量技术, 2006, 30(2): 133-134
- [38] 覃永新, 陈文辉, 张帆. 实时视频数据采集的 FPGA 实现. 电子技术应用, 2009, 35(9): 57-58
- [39] 程方敏, 黄启俊, 向守坤等. 基于 FPGA 的高速 SDRAM 控制器的视频应用. 电子技术, 2009, 36(7): 22-23
- [40] 鲁一杰, 赵攀, 彭强, 江永全. 基于 DSP 的视频车辆实时检测系统的实现. 计算机应用研究, 2010, 26 (2): 757-759
- [41] Su Xia Wang, Zheng He Song, Zhong Xiang Zhu. Study on Crop Image Feature Extraction of Vehicle-Based Estimmion System on Large Scale Crop Acerage. Proceedings of 2007 Intenational Conference on Mashine Learning and Cybematics. BeiJing, 2007, (1): 377-382

致 谢

在千年学府所特有的浓厚学术氛围中，我终于完成了硕士研究生阶段的学业，在本论文完成之际，我谨向多年来教导、帮助、支持和关心我的所有人表示最衷心的感谢。

首先，我要衷心的感谢我的导师黎福海教授，感谢他对我无私的关怀和谆谆教导。黎老师平易近人的教学态度、严谨的治学风范、渊博的专业知识对我的生活和学习产生了深刻的影响。我从他身上学到的，不仅仅是科学知识，还有更多的做人的道理。在本论文的研究与创作期间，从论文的选题、立论到实验仿真，再到最后的撰写成文无不倾注着老师的心血。在这里我要对恩师黎福海教授表示最诚挚的感谢。

感谢实验室的同窗好友李强、王磊、陈嘉玲、刘建平你们在学习和日常生活等方面给了我很多帮助，让我留下了很多难忘的记忆。感谢肖华、于长江、姚雪芬、欧阳科文、谢琼琳、刘泽等师兄师姐在生活上对我的照顾，在学习上给我指点迷津。感谢我的师弟师妹肖祥慧、陈立、王建财、元芳、彭军、余维克等，你们让我的研究生生活丰富多彩。

感谢电气与信息学院的各位老师，你们的教导会让我终身受益，你们给予我的学识将不断指引我今后的人生道路。

对于我的父母、姐姐和妹妹，我要向你们发出内心最深处的感谢，会用自己的实际行动来回报你们。

最后，感谢母校湖大对我的培养，我将永远铭记自己是千年学府的学子，用自己的努力和成绩来回报母校，回馈社会！

梁来朋

2011年4月15日

附录 A 攻读学位期间所发表的学术论文

- [1] 梁来朋, 黎福海等. 基于均值偏移算法的目标跟踪系统及其在 Blackfin 上实现. 计算机系统应用, 已录用, 文章编号: ad095



附录 B 电路原理图

