

分类号\_\_\_\_\_

密级\_\_\_\_\_

UDC \_\_\_\_\_

编号\_\_\_\_\_

# 中国科学院研究生院

## 硕士学位论文

### 基于 TMS320DM648 的全景视频监控系统

张振凯

指导教师 王文杰 副教授 焦建彬 教授

中国科学院研究生院

申请学位级别 硕士学位 学科专业名称 计算机应用技术

论文提交日期 2010 年 4 月 论文答辩日期 2010 年 5 月

培养单位 中国科学院研究生院

学位授予单位 中国科学院研究生院

答辩委员会主席 \_\_\_\_\_

## 独 创 性 声 明

本人郑重声明：所提交的学位论文，是本人在指导教师的指导下，独立进行研究工作所取得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人或集体已经发表或撰写过的研究成果，也不包含为获得中国科学院研究生院或其它教育机构的学位或证书所使用过的材料。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结果由本人承担。

签名: \_\_\_\_\_ 日期: \_\_\_\_\_

## 关于学位论文使用授权的说明

本人完全了解中国科学院研究生院有关保管、使用学位论文的规定，其中包括：①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的,复制赠送和交换学位论文；⑤学校可以公布学位论文的全部或部分内容。

(涉密的学位论文在解密后应遵守此规定)

签名: \_\_\_\_\_ 导师签名: \_\_\_\_\_ 日期: \_\_\_\_\_

## 摘要

传统的视频监控系统有许多缺点，例如大部分只能在监视器上提供一幅监控场景的画面，有的即使能够把多路视频显示在一个监视器上，也不能为用户在一幅画面上无冗余地提供被监控区域的全景图。为了克服传统监控系统的这些缺点，本文开发了一种基于高性能 DSP 的全景视频监控系统，并在此系统上实现了动态人体目标检测算法。

本系统是在美国德州仪器（Texas Instruments）公司生产的 TMS320DM648 基础上实现的。论文首先对基于 DSP 的嵌入式图像处理系统进行了综述，介绍了 DSP 的选型和软件开发流程，然后详细论述了全景视频监控系统的设计思想和软硬件设计开发。在硬件方面，文中主要设计并实现了系统中的存储系统、视频采集系统、视频显示系统、时钟电路以及 JTAG 电路等。在软件方面，文中主要设计了根据 SIFT 特征点匹配的图像拼接技术以及在 DSP 上的实现和改进。最后实现了可用于智能视频监控的动态人体目标检测算法。

**关键词：**TMS320DM648、智能视频监控系统、SIFT 特征点

## A Panoramic Video Surveillance System Based on TMS320DM648

**Zhang Zhenkai** (Computer Application)

Directed by: **Wang Wenjie, Jiao Jianbin**

Traditional video surveillance systems have a lot of shortcomings. For example, most of them only can display a single channel video on one monitor. Even though some could display multi-channel video on a monitor, there is a lot of redundant information on the surveillance image. In order to overcome the shortcomings traditional systems have, we have developed a panoramic video surveillance system which could also detect the mobile human based on a high performance DSP.

We have implemented our system based on Texas Instruments's TMS320DM648. This paper begins with stating how to develop a DSP embedded system for image processing, including DSP type selection and application development. Then, this paper elaborates on both the designing idea of this surveillance system and how to design and develop its hardware & software. In the terms of hardware design, this paper mainly describes how to implement the system's memory system, video capture system, video display system, clock circuit and JTAG circuit. In the terms of software design, this paper mainly depicts how to implement and improve the SIFT feature matching algorithm, which is the basis of the image stitching. Finally, this paper states how to implement the human detection in the system so as to achieve the intelligent surveillance.

**Keywords:** TMS320DM648, Intelligent Video Surveillance System, SIFT

## 目录

摘要.....	I
目录.....	III
图目录.....	V
表目录.....	VII
第一章 绪论.....	1
1.1 课题来源、背景和意义.....	1
1.1.1 课题来源.....	1
1.1.2 课题背景及意义.....	1
1.2 视频智能监控系统的国内外研究现状.....	3
1.3 本文的研究内容.....	4
1.4 本文的组织结构.....	5
第二章 基于 DSP 的嵌入式图像处理系统.....	6
2.1 DSP 嵌入式系统开发综述 .....	6
2.1.1 DSP 处理器结构特点 .....	6
2.1.2 图像处理系统中的 DSP 选型.....	9
2.2 TMS320DM648 的特点 .....	10
2.3 DSP 上的软件开发 .....	11
2.3.1 CCS 集成开发平台 .....	12
2.3.2 DSP/BIOS 实时操作系统 .....	13
2.3.3 DSP/BIOS 驱动程序开发 .....	15
2.3.4 RF5 参考框架 .....	17
第三章 基于 TMS320DM648 的全景视频监控系统原型 .....	19
3.1 系统结构.....	19
3.2 基于 DDR2 的存储系统的设计 .....	20
3.2 IIC 总线 .....	22
3.2.1 IIC 总线的概念 .....	22
3.2.2 IIC 总线的信号类型 .....	23
3.2.2 IIC 总线的数据传输格式.....	24
3.2.2 系统中 IIC 总线的连接.....	25
3.3 视频采集系统的设计 .....	25
3.4 视频显示系统的设计 .....	27
3.5 JTAG 接口电路的设计 .....	27
3.5 时钟电路的设计 .....	29
第四章 全景图像在 DSP 上的实现 .....	31
4.1 SIFT 特征点的提取.....	31
4.1.2 高斯低通滤波.....	32

4.1.3 DoG 图像金字塔的构建 .....	33
4.1.4 测试候选的特征点的稳定性 .....	35
4.2 改进的 SIFT 特征点的描述子 .....	37
4.2.1 计算特征点的主方向 .....	38
4.2.2 生成以及改进 SIFT 描述子 .....	38
4.3 SIFT 特征点的匹配 .....	42
4.4 全景图像的实现 .....	44
第五章 智能视频监控系统 .....	48
5.1 动态目标检测 .....	48
5.2 人体目标检测 .....	50
5.2.1 MSO 特征 .....	50
5.2.2 Adaboost .....	52
5.2.3 级联的 Adaboost 分类器 .....	53
5.2.3 人体检测在系统中的实现 .....	54
总结与展望 .....	56
参考文献 .....	57
发表文章目录 .....	60
致谢 .....	61

## 图目录

图 2-1 TMS320DM648 功能框图.....	10
图 2-2 DSP 上应用程序的开发.....	11
图 2-3 CCS 下开发程序的步骤.....	12
图 2-4 DSP/BIOS 配置工具.....	14
图 2-5 应用程序结构中的设备驱动组件.....	15
图 2-5 应用程序结构中的设备驱动组件.....	16
图 2-7 微驱动的操作流程.....	17
图 2-8 RF5 参考框架中的各个模块.....	18
图 3-1 全景视频监控系统整体框架.....	19
图 3-2 全景视频监控系统框图.....	20
图 3-3 DDR2 存储控制器的数据通路.....	21
图 3-3 DDR2 存储系统的原理图.....	22
图 3-4 IIC 总线设备互连.....	23
图 3-5 开始信号 (S) 和结束信号 (P) .....	23
图 3-6 响应信号 (ACK) .....	23
图 3-7 IIC 总线上的位传输.....	24
图 3-8 完整的数据传输.....	24
图 3-9 IIC 总线的连接.....	25
图 3-10 Video Port 的结构框图.....	26
图 3-11 TVP5154 在系统中的原理图.....	26
图 3-12 SAA7105H 在系统中的原理图.....	27
图 3-13 JTAG 接口电路的原理图.....	28
图 3-14 时钟电路的原理图.....	29
图 4-1 用 $\sigma=1.6$ 对第一幅 6 次高斯滤波后的结果.....	32
图 4-2 DoG 金字塔的构成.....	33
图 4-3 DoG 金字塔的实验结果.....	34
图 4-4 极值点检测.....	34

图 4-5 稳定的 SIFT 特征点.....	36
图 4-6 SIFT 特征点的主方向.....	37
图 4-7 领域梯度及其特征点的描述子.....	38
图 4-8 不同的算法匹配得到的特征匹配的错误率.....	39
图 4-9 SIFT 特征点描述子的生成.....	40
图 4-10 SIFT 特征点匹配结果.....	43
图 4-11 图像拼接之前的 8 幅视频帧.....	44
图 4-12 拼接之后的全景图.....	44
图 4-13 图像拼接之前的 8 幅视频帧（俯视）.....	45
图 4-14 拼接之后的全景图（俯视）.....	45
图 5-1 帧差法的流程图.....	47
图 5-2 动态目标检测的实验结果.....	47
图 5-3 人体检测的流程图.....	48
图 5-3 单个特征的计算.....	49
图 5-4 矩形区域的灰度值求和.....	49
图 5-5 特征空间的划分.....	50
图 5-6 级联分类器的示意图.....	51
图 5-7 全景图象中人体检测结果.....	53

表目录

表 3-1 PLL1 时钟频率范围.....	29
------------------------	----

## 第一章 绪论

### 1.1 课题来源、背景和意义

#### 1.1.1 课题来源

本课题来源于国家自然科学基金（课题编号：60672147、60872143）以及中国科学院“百人计划”资助课题。

作者在论文工作期间，参与了以下课题：

- 1) 航天科工集团哈尔滨风华机器厂“‘天目’全景摄像预警控制系统开发”；
- 2) TMS320DM642 上的 TCP/IP 协议栈的移植；
- 3) 基于 FPGA 的视频中运动目标的跟踪；
- 4) 爬壁机器人的控制、通讯及视频系统的开发；
- 5) 南京军区徐州某部“弹药库智能监控系统”。

#### 1.1.2 课题背景及意义

随着科学技术的飞速发展及人类安防意识的不断增强，视频监控系统在各个领域得到了越来越广泛的应用，例如，日常生活中的小区监控、超市的安全监控等，银行系统的柜台监控，交通方面的违章监控等。这些监控系统大都是由一个或多个摄像机以及与之相连的电视监视系统组成，它们的用途主要是对固定场景进行监控并保存结果。当异常情况（如盗窃行为等）发生后，保安人员才通过记录的结果察看发生的情况，但往往为时已晚。另外，当对重要部门（如军事禁区等）进行监控时，需要工作人员连续监视屏幕。这种工作非常枯燥，很容易使监控人员疲劳，漏报、误报事件时有发生。可见，传统的视频监控系统不但需要大量的人力，而且有时监控效果并不理想。所以人们对视频监控系统的智能化要求越来越高。鉴于此，目前国内学术界及工业部门开始着眼于研究新一代的具有智能化视觉监视功能的监控系统。这种监控系统具有巨大的实用价值和难以估量的商机。图像处理理论研究的日趋成熟以及当前计算机技术、数字信号处理技术的高速发展也使得这项研究成为可能 [1]。

开发具有智能化视觉监视功能的监控系统的核心是要对进入监控视场的运动目标(诸如人，汽车等)进行实时的检测、跟踪和理解，并由此做出相应的判断

和处理（如控制存储模块的动作、报警等），同时，根据不同场合需求，进一步提供相关信息。

现有的视频监控系统主要分三类：模拟视频监控系统，PC 机监控系统及嵌入式视频监控系统。模拟视频监控系统是将远端摄像头采集的模拟视频信号通过电缆直接发送到监控中心的监视器上。模拟系统设计、制造比较简单，但是传送距离比较短，安装、使用以及数据的保存都比较困难。基于 PC 机卡的远程图像监控系统由机插视频卡构成，在监控现场，由若干个摄像机、各种检测、报警探头及数据设备等构成，通过各自的传输线路连接到监控工控机上。除了处理各种信息和完成本地所要求的各项功能外，系统利用视频压缩卡和通信接口卡，通过网络将信息传输到一个或多个监控中心。这种监控系统虽然实现了视频监控的基本功能，但其可靠性差，视频获取前端比较复杂，需专人管理。嵌入式视频监控系统却具有不可比拟的优势，与模拟视频监控系统相比，具有传送距离远，图像质量好，安装、使用以及数据的保存都很简单等特点：与 PC 机监控系统相比集成度高，可靠性好等特点 [1][2][3]。

目前大部分嵌入式视频监控一般只能监视一个区域，如果需要监视全景区域，不仅需要把摄像头放置于不同的区域，而且还要精心设计这些摄像头之间的关系，才能实现全景监控的目的。

本文提出了一种全新的视频监控系统，这种嵌入式系统使用图像拼接算法把 8 路视频图像拼接在一帧中，只显示非重叠的区域，形成了一幅全景图。与传统的视频监控系统相比，这种视频监控系统能在一帧中提供更多有用的信息。这种系统可以应用在汽车的辅助安全驾驶、安全监控等许多方面。而且，我们把人体检测加入了整个系统，使得其能在发现动态人体目标时报警，从而使用户可以在大多数时间中无需关注被检测区域，应此具有其很好的应用价值。

我们选择使用 DSP 来实现这个系统。DSP 是一种独特的微处理器，专门用来处理大量的数字信号 [2]，其快速的 MAC 指令使得 DSP 非常适用于处理数字信号。而且 DSP 具有稳定性好、可重复性高、可大规模集成、可编程性强和易于实现自适应处理等特点，给数字信号处理的发展带来了巨大机遇，使得信号处理更灵活、功能更复杂，应用遍及国民生活的各个方面，如互联网、高速调制解调器、无线通信、语音识别、图像处理、工业控制、生物医学、航空航天和各类

自动化仪器等等。随着半导体制造工艺的发展和计算机体系结构的改进，DSP 芯片的处理功能越来越强大，运算速度也越来越快，高速实时数字信号处理成为研究的重点。

在我们的系统中，需要实现一种图像拼接技术。图像拼接技术是计算机视觉与图像处理领域的一个重要的研究内容，其在现实生活中也发挥着巨大的作用。由于图像采集设备视野范围的有限性，在某些应用场合需要把多幅具有重叠关系的图像拼接为一幅完整的图像，以扩大视野范围，这便是图像拼接的目的。为了进行图像拼接，相邻图像的边界必须存在重叠区域，该重叠区域代表了相同的场景内容，图像配准的工作就是搜索相邻图像中相同的内容，从而确定它们的相对位置关系。该技术在很多实际领域中均有应用，例如：遥感图像的拼合、全景图像的生成和物体的三维重建等[21][27][29]。

在我们的系统中，还需要实现人体目标检测技术。人体目标检测技术有许多应用价值和研究意义。成熟的人体目标检测系统能自动识别危险目标，可用于重点建筑（核电站、水电站、水库、银行、重要桥梁等）及重要地点（国家重点保护建筑、博物馆、体育场馆、地铁站、监狱等）的安全保卫工作，还可以及时监控交通事故并可以在十字路口、新建高速路端等进行自动行人检测和安全预警，并且随着城市规模的扩大，住宅小区逐渐成为安全防范的重点，结合红外/近红外等视频设备，该技术可增强小区安全监控、减少犯罪，从而增强居民的安全感[41]。

因此，在本文提出的这个全景监控系统中，需要结合很多的研究热点来完成，这些技术不仅涉及基于 DSP 的嵌入式设计的关键技术，也有图像拼接技术和动态目标检测和人体目标检测的技术。这种全景视频监控系统具有很好的科研价值和应用意义。

## 1.2 视频智能监控系统的国内外研究现状

智能视频监控系统的广阔应用前景使得它成为了计算机视觉领域中备受关注的前沿方向[1]。它的研究内容相当丰富，涉及到图像处理、机器视觉、模式识别、人工智能、数字信号处理等学科知识。因此，这个领域不仅具有重大的应用价值，而且还具有很高的理论研究价值，因此，吸引了国内外众多研究人员投入到对它的研究中来，这方面的研究成果也是层出不穷。

国外，尤其在英、美两国对智能视频监控中的运动分析进行了深入的研究[4]。较突出的有英国雷丁大学计算机系的 VIEWS 项目组，该项目组开展了对车辆和行人的跟踪及其交互作用识别的相关研究。这一新技术，主要通过分析反常行为来识别嫌疑犯，并可快速发出警报。该技术有望使未来的闭路电视监视器不仅可自动识别扒手和盗车贼，而且还会预报地铁或机场内可能发生的行凶抢劫或恐怖活动。据研究人员介绍，正在开发的新技术采用了相对原始的手段识别图像，如将低而长的可移动物体识别为“汽车”，将高而瘦的定义为“人”等等。采用新技术的监控系统，主要通过记录不同的物体运动轨迹来得出有关分析结果。

国内在智能视频监控系统技术方面的研究起步比较晚，主要的研究团队有中科院自动化所模式识别国家重点实验室、北京大学视觉与听觉信息处理国家重点实验室及清华大学智能技术与系统国家重点实验室[2]。

总的来说，由于国外在这方面的研究开展的比较早，而且软硬件设备更加齐全，因此国外的智能视频监控系统在应用平台及理论上都比较成熟、完善。相对而言国内则起步较晚，理论于技术都有待进一步提高。我国基于嵌入式技术的网络视频监控系统刚刚起步，所以研究并开发一种基于嵌入式系统的全景视频监控系统具有很大的工程实际意义。基于 DSP 的全景视频监控系统，由于可以灵活的修改其图像处理算法，它的应用主要面向用户的特定需求和对实时性有较高要求的场合。因此，有理由相信在嵌入式系统的基础上构建视频图像采集，处理及传输系统具有广阔的市场前景。

### 1.3 本文的研究内容

本文的主要研究内容如下：

- (1) 选择特定型号的 DSP 来构建此系统中的图像处理硬件系统，本文选择了美国德州仪器公司的 TMS320DM648，并根据它的特性，选择特定的外设。
- (2) 在此系统的硬件系统上，实现并改进 SIFT 特征点的提取和描述，在不失去鲁棒性的同时，加快了其提取和描述的速度。
- (3) 实时的动态目标检测算法和人体目标检测算法在本文的全景视频监控系统上的实现，从而达到了智能视频监控的目的。

#### 1.4 本文的组织结构

第一章，绪论。主要论述了全景视频监控系统的研究背景和意义，分析了国内外的研究现状，列出了本文的主要研究目的和研究内容，总结了本文的主要贡献。

第二章，基于 DSP 的嵌入式图像处理系统。主要综述了基于 DSP 的嵌入式系统的设计方法，同时也对 DSP 开发应用程序进行了简单的综述。

第三章，基于 TMS320DM648 的全景视频监控系统的原型。主要论述了所提出的全景视频监控系统的组成部分和如何与 DM648 相连。

第四章，基于 DSP 平台的全景图像拼接。主要陈述了传统上 SIFT 特征点的理论基础和提取方法，并且给出了根据 DSP 的特性对 SIFT 特征点的提取过程和描述子进行的改进，并且在过程中给出了特征点匹配的实验结果以及最后全景图像的实验结果。

第五章，智能视频监控系统，主要陈述了动态目标检测和人体目标检测在系统中的实现。为了能使系统更好地用于监控领域，系统中实现了实时性的动态目标检测和人体目标检测。

最后是对现有工作的总结和对未来工作的展望，以及对如何进一步完善全景视频监控的各个研究热点和难点的探讨。

## 第二章 基于 DSP 的嵌入式图像处理系统

### 2.1 DSP 嵌入式系统开发综述

数字信号处理器 (DSP, Digital Signal Processor), 是专门用于数字信号处理运算的微处理器, 其主要应用是实时快速地实现各种数字信号处理算法。DSP 的集成度高, 特别是可编程性高和易于实现自适应处理等特点, 作为数字信号处理核心器件, 在图像处理、语音处理等领域得到广泛的应用[17]。

#### 2.1.1 DSP 处理器结构特点

##### (1) 哈佛体系结构

一般的通用处理器使用冯·诺依曼体系结构, 在这种结构中, 在靠近处理器的存储层次上, 其代码段和数据段共同存放在一个高速缓冲存储器内, 数据和程序使用同一地址总线和数据总线, 处理器要执行任何指令时, 都要先从储存器中取出指令解码, 再取操作数执行运算, 即使单条指令也要耗费几个甚至几十个周期, 这种结构不能满足 MAC 指令必须在一个指令周期中对存储器进行四次访问的要求, 所以 DSP 一般使用哈佛体系结构[17]。

哈佛体系结构的主要特点是把靠近处理器端的存储层, 即高速缓冲存储器, 分成了两个独立的高速缓冲存储器, 每个存储器独立编址、独立访问[16]。与两个存储器相对应的是系统中的两套数据总线和地址总线, 分别为程序的数据总线与地址总线, 数据的数据总线与地址总线。这种分离的地址总线和数据总线可允许在一个机器周期内同时获取指令字 (来自程序高速缓冲存储器) 和操作数 (来自数据高速缓冲存储器), 从而提高了执行速度, 使数据的吞吐率提高了 1 倍。又由于程序和数据存储器在两个分开的空间中, 因此, 取指和执行能完全重叠。并且在哈佛体系结构中, 有时会通过增加数据高速缓冲存储器和它的总线来实现更大的存储带宽。现代高性能通用处理器通常也具有两个片上超高速缓冲存储器, 一个存放数据, 一个存放程序。从理论的角度上讲, 这种双片上高速缓冲存储器与总线连接等同于哈佛结构, 但是, 通用处理器使用控制逻辑来确定哪些数据和指令字驻留在片上高速缓存里, 这个过程通常不为程序设计者所见, 而在 DSP 里, 程序设计者能明确的控制哪些数据和指令被存储在片上的存储单元或缓存中。

### (2) 快速的乘加功能部件

数字信号处理中最重要的基本运算是乘法累加运算，它们占用了大量运算时间，是最主要和最耗时的运算。通用处理器内部只能执行加法和逻辑操作，乘法操作则由协处理器完成。对于信号处理的应用，乘法累加运算是紧密相连的，相乘后紧跟着要将乘积结果进行累加，因而乘加应统一考虑，构成以乘法器、加法器流水线为中心的乘法功能部件，另一方面，VLSI 技术使得乘法器、加法器完全能集成在一个芯片中[17]。

DSP 中设置了乘法功能部件用来做乘法累加运算，即用来执行 MAC 指令 [16]。这种操作相比起通用处理器中的同样的操作，需要更少的时钟周期，往往可以在单个时钟周期内完成，大幅提高了 DSP 作乘法累加运算的速度。因此，乘法功能部件和 MAC 指令是 DPS 芯片实现快速运算的保证。现代高性能的 DSP 芯片甚至具有两个以上的乘法功能部件以提高运算速度。数据宽度也从 16 位增加到 32 位。无论定点或浮点 DSP 芯片都有这一功能部件，而且做乘法累加运算是 DSP 芯片的主要速度指标。

### (3) 流水线技术

DSP 中把流水线分成了许多流水段操作，可以减少指令的 CPI，同时就增加了数字信号处理的并行性。采用流水线的处理器是由一系列处理电路组成，这些处理电路称为流水段。操作数流经每个流水段，即每个流水段对操作数进行部分处理，操作数经过所有流水段后才能得到最后结果。流水线操作即把一条指令分成一系列步骤来完成，不同步骤完成不同的任务，一条指令只有经过所有步骤才能得到结果。流水线分为指令流水线和算术流水线。指令流水线是指取指令和执行指令的不同阶段在流水线上进行；算术流水线是指算术逻辑运算的不同阶段在流水线上进行。DSP 一般采用指令流水线方法。处理器可以并行处理 2~6 条指令，每条指令处于流水线的不同流水段[16]。

### (4) 特殊的寻址方式

在数字信号处理中要遇到大量的地址运算，在某些情况下，地址运算量甚至超过了数据的运算量[16][18]。DSP 中设计有一个特殊的硬件算术单元——地址产生器。地址计算由这个专门的硬件来负责，不需要耗费额外的时间。现在的某些通用处理器中也有独立的地址产生单元，对于一些特殊的地址运算不需要耗费

额外的 CPU 时间，但是 DSP 的地址运算单元更强大，支持如比特翻转寻址和循环寻址，可极大加快 FIR 滤波运算和 FFT 变换的速度。

#### (5) 特殊的 DSP 指令集

DSP 处理器的指令集设计有两个目标：处理器底层硬件使用率最大化和存储 DSP 程序所需的存储空间最小化。为实现第一个目标，传统 DSP 处理器允许程序员在一条指令中显式指定多个并发操作，通常包含一个或两个取数操作与一个并发的算术操作。为实现第二个目标，指令长度一般较短，处理器结构中限制了哪些寄存器只能被哪些操作使用，哪些操作才能被合并到一条指令中。为了进一步减少指令编码所需位数，DSP 处理器比其他处理器提供更少的寄存器，并使用模式位来控制处理器操作的特征（如舍入或饱和），而不是将这些信息编码到指令中。所有这些导致传统 DSP 处理器有着高度专用、复杂和不规则的指令集，使得产生高效率的汇编语言代码变得很困难[17][18]。

#### (6) SIMD 结构

大部分 DSP 都采用单指令流多数据流（SIMD）技术提高一些算法的性能。例如，一条 SIMD 乘法指令可以在一个时钟周期内，在不同操作数集合上进行两个或多个乘法操作。对于多媒体和信号处理应用经常出现的一些向量操作，这一技术能够大大提高计算速率。为有效利用 DSP 的 SIMD 功能，需要程序员做大量工作。程序员需要很好的组织存储器内的数据，使 SIMD 处理能够全速运行，有时还需重新组织算法以便最大限度利用处理器资源。SIMD 只有在那些能够并行处理数据的算法中有效率，对那些本质上串行的算法（例如，使用一个操作的结果作为下一个操作的输入），SIMD 作用不明显。

#### (7) 丰富的片内外设

DSP 处理器为了自身工作的需要和与外部环境的协调工作，往往设置了丰富的片内外设。一般来说，DSP 处理器的外设主要包括：时钟发生器（振荡器和锁相环 PLL）、定时器、软件可编程等待状态发生器（用于较快的片内设施与较慢的片外电路及存储器协调）、通用 I/O、同步串口（SSP）和异步串口（ASP）、主机接口（HIP）和 JTAG 边界扫描逻辑电路（用于对 DSP 处理器作片上的在线仿真，以及多 DSP 处理器条件下的调试）[18]。

### 2.1.2 图像处理系统中的 DSP 选型

对图像处理技术而言，由于要处理的数据量大，计算复杂，计算中间结果精度要求高，因此需要选择合适的 DSP 处理器芯片。DSP 的选择根据实际的应用系统的需要而确定[16]。选择 DSP 处理器芯片时应考虑如下诸多因素：

(1) 确定选择定点或浮点 DSP。数字信号处理算法的数据格式有定点和浮点之分，而数字信号处理系统采用的数据格式决定了它所处理信号的处理精度、动态范围和信噪比，且不同数据格式的易用性和开发难度也不一样。选择定点或浮点 DSP，首先要看模数转换时需要的比特数，如果图像的每个像素小于 16 bit，则用 16 bit 定点 DSP 即可；如果大于 16 bit，则需要用浮点 DSP 来捕捉更大的动态范围。其次考虑算法的复杂度和经济问题。一般说来，浮点 DSP 芯片的运算精度高，动态范围大，寻址空间大，指令运算能力较强，但功耗大、成本高、体积较大。定点 DSP 芯片的运算精度与浮点 DSP 芯片相同（数据位数和浮点芯片相同的情况下），而功耗、成本、体积与浮点 DSP 芯片相比较小，且易于实现，稳定性好。

(2) DSP 处理器芯片的运算速度。和通用处理器类似，衡量一款 DSP 芯片性能的一个最重要的指标就是其运算速度，这是 DSP 芯片选型过程中需要考虑的一个重要因素。DSP 芯片运算速度的衡量标准包括：DSP 处理器芯片的主频、MIPS（每秒执行百万条指令）以及单周期的乘加次数。

(3) 其他考虑因素。在硬件方面还应考虑芯片的硬件资源：不同型号 DSP 芯片所提供的硬件资源是不相同的，如片内 Cache 的大小，外部可扩展的程序和数据空间的大小，总线接口，I/O 接口等。在软件方面主要是开发软件的功能性和时间要求等因素：DSP 系统的开发过程中，开发工具是必不可少的。如果没有开发工具的支持，要想开发一个复杂的 DSP 系统几乎是不可能的。如果有功能强大的开发工具的支持，则开发的时间就会大大缩短。所以，在选择 DSP 芯片的同时必须注意其开发工具的支持情况，包括软件和硬件的开发工具。

在图像处理系统中应用最广的为美国德州仪器公司 (Texas Instruments) 生产的 TMS320C6000 系列的 DSP，以及基于此系列的 DSP 而推出的专门用于多媒体处理的 Da Vinci 系列的 SoC。TMS320C6000 系列的 DSP 可专门用来进行图像处理，因为其具有较高的性能以及丰富的片上外设，尤其它的 DM64X 系列。

其 DM64X 系列为定点型 DSP, 运算速度可达 5000MIPS 以上; 采用先进的 VLIW 结构, 有 8 个独立的功能部件, 其中 6 个为 32/40 位的 ALU, 2 个为专用的乘法器, 还有 64 个 32 位的通用目的寄存器; 二级片上存储结构, 其中 L2 Cache 也可配置为片上 RAM 使用; 最高可寻址 1GB 外部存储空间的 64 位 EMIF 总线, 可与 SRAM、E2PROM、SDRAM 和 SBSRAM 等无缝连接, 最新的 Da Vinci 系列中还有 DDR2 接口; 具有 64 路独立通道的 EDMA 控制器; 最为关键的是其片上有多个 Video Port 接口, 可直接输入 BT. 656 格式的视频, 便于 DSP 处理。

## 2.2 TMS320DM648 的特点

本文开发的系统中选择了美国德州仪器公司生产的 TMS320DM648, 作为此系统的多媒体处理器, 用于图像处理。

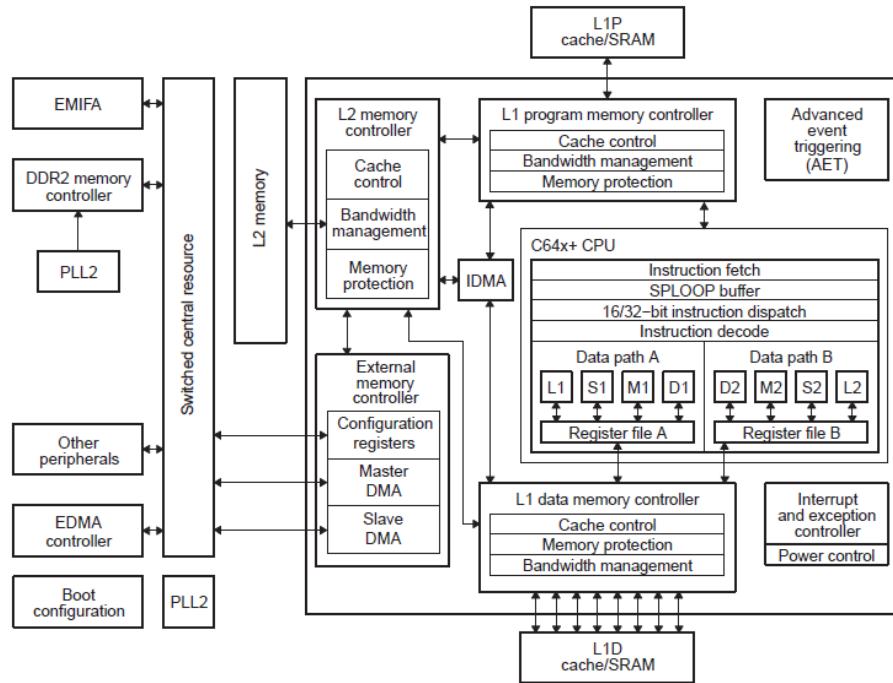


图 2-1 TMS320DM648 功能框图

TMS320DM648 是一款面向多媒体数据处理的高性能、数字视频定点 DSP, 其核心处理器主频最高可达 900MHz[16]。它具有丰富的片上外设。其功能结构图如图 2-1 所示。DM648 与 TI 之前的 DM642 相比, 性能有了很大的提高。DM648 采用的 C64x+ 内核工作周期性能比 DM642 提高了 20%, 代码密度提高了 20-30%, 每周期执行的 16 位 MMAC 翻了一番, EDMA 带宽提高了三倍, 而且代码与 C64x

内核完全兼容。DM648 采用集成的视频与影像协处理器(VICP)，基于指令的引擎 VICP 支持前端处理功能，且能与主 DSP 内核并行工作，实现包括 VICP 功能库与高效色彩空间转换、图形缩放与解隔行扫描等功能。DM648 的主要新特性还包括：

- (1) 5 个 16 bit 的 Video Port 口，支持多种视频标准，每个 VP 口可支持两通道的视频输入和 1 通道的视频输出。
- (2) 1 个 DDR2 存储接口。
- (3) 1 个带 SGMIII 的千兆以太网 MAC，在多处理器系统中，支持将芯片用作 PCI 桥。
- (4) 1 个高速 DMA 引擎，支持处理器管理各式各样高带宽存储器和 I / O 口。除此之外，DM648 还集成了通用定时器、通用输入/输出 (GPIO) 端口等其他外设，利用这些外设接口可以降低视频应用的开发难度和成本。

## 2.3 DSP 上的软件开发

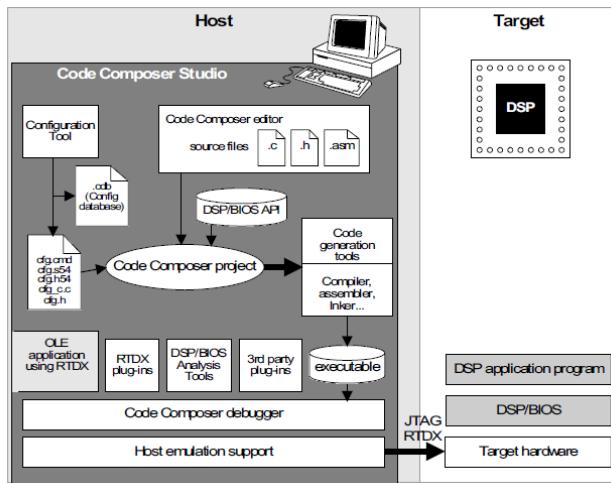


图 2-2 DSP 上应用程序的开发

为了加速 DSP 应用程序的开发，德州仪器公司开发出了 eXpress DSP 技术 [12]。eXpress DSP 是一种实时 DSP 软件技术，它是一种 DSP 编程的标准，利用它可以加快开发 DSP 软件的速度。eXpress DSP 包括：CCS 集成开发平台、eXpress DSP 软件（DSP/BIOS 内核、TMS320 DSP 算法标准、eXpress DSP 参考框架）和第三方支持软件三大部分。利用该技术，软件调试、软件进程管理、软件的互通及算法的获得等方面都变得容易，可加快软件开发进程。eXpress DSP 算法标准是 TI 针对代码的可重复利用提出的，只适用于 TI 的 DSP 器件。目的是按照

这个标准开发出的 DSP 算法无需修改或只需要少量的修改就可以集成在新的 DSP 系统中。

### 2.3.1 CCS 集成开发平台

CCS 是美国德州仪器公司推出的一套集成开发环境，通过 CCS，可以方便的进行应用程序的移植与调试，并可以进行变量实时观察、数据图像化等，这就大大的方便了后期程序的调试并缩短了系统开发周期。CCS 中主要的集成工具包括：高效的编译器、汇编器等，可快速完成程序的全编译并进行下载；CCS 中集成了嵌入式操作系统 DSP/BIOS，可用来进行准确的内存分配及 cmd 文件的生成；实时分析软件，可以在程序调试过程中进行变量值的实时观察；数据可视化软件，可以把内存中图像数据以图像的形式进行显示，对于本系统来说有着重要的作用[12]。

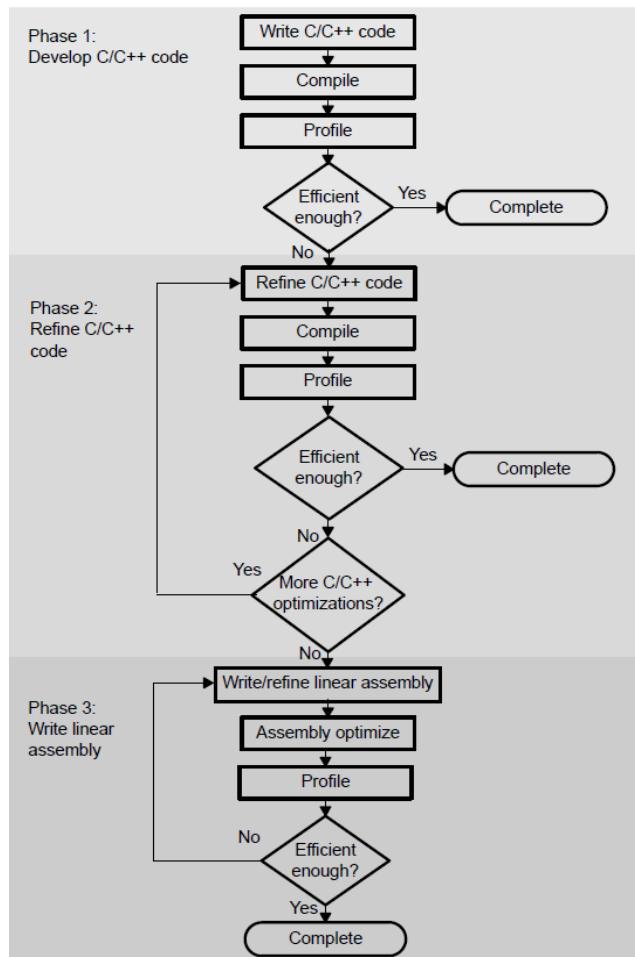


图 2-3 CCS 下开发程序的步骤

在 CCS 下开发程序的步骤可分为 3 个阶段，如图 2-3 所示。第一个阶段为

简单地用 C 语言实现程序的算法，而不需要知道 DSP 的特性，该阶段开发出来的程序往往实时性能很差。第二个阶段为优化上一阶段写出的程序，一般为利用 CCS 可支持的内联函数和优化选项。第三个阶段为利用 CCS 中提供的代码分析工具，从程序中找出对时间要求较高的区域，然后用线性汇编重新编写这部分代码，从而达到实时性要求。

### 2.3.2 DSP/BIOS 实时操作系统

稍微复杂点的嵌入式设备上都会有一个嵌入式的实时操作系统，它负责嵌入系统的全部软、硬件资源的分配、任务调度，控制、协调并发活动。常见的嵌入式操作系统有 Linux、VxWorks、Windows CE、Palm OS、Symbian 等。由于 DSP 体系结构和任务的独特性，DSP 一般不直接支持这些操作系统。DSP/BIOS 是以德州仪器的 DSP 器件为特定目标设计的一个实时操作系统，它为多任务协调调度和用于软件开发的底层诊断工具提供了可升级的解决方案。现在 DSP/BIOS 已经成为德州仪器软件开发工具的一部分。DSP/BIOS 主要由 3 部分组成：多线程实时内核和 API 应用程序接口函数、实时分析工具 RTA 及 DSP/BIOS 配置工具[17]。

DSP/BIOS 以模块化方式提供给用户，对线程、中断、定时器、内存资源、所有外设资源的管理能力都可以根据需要剪裁，多线程程序的调度管理、同步和通信、内存管理、实时时钟管理、中断服务管理、外设驱动程序的管理完全由 DSP/BIOS 实时内核来完成，而数字信号算法作为一个线程嵌入到系统即可。DSP/BIOS 提供大量底层的 API 函数接口，可用于支持系统实时分析、线程管理、调度软件中断、周期函数和 Idle 函数（后台运行函数）以及外部硬件中断及各种外设资源的管理；利用图形化的 DSP/BIOS 配置工具可以快速的设置 DSP/BIOS 模块的参数，建立 DSP/BIOS 对象，设置片级支持库（CSL）的参数等；实时分析工具位于主机端用于配置、优化和监视实时应用软件的调试，方便软件执行效率的实时分析与优化。

DSP/BIOS 内核是一个在 200 字节到 4K 字节之间的可裁减的软件框架，这个框架支持应用程序中多线程之间的实时协调调度。该内核对实时 I/O 也提供管理服务，以便使内核和外围设备间的数据流得到有效管理。与基于主机端的一系列分析工具相结合，该内核能够在程序执行过程中实时捕捉程序运行信息。尽管 DSP/BIOS 提供了很好的实时系统的框架，但是典型的嵌入式 DSP 系统仍然需要

程序员在提供核心的信号处理算法以外，再提供额外的控制专用系统硬件的底层软件模块。编译后该内核的实际大小取决于系统设计中采用了哪些部分。DSP/BIOS 内核模块与用 C 或汇编语言编写的程序块基本相同，在代码开发的链接阶段嵌入到应用程序目标代码中。链接器在某个应用中只提取它所用到的内核资源中的函数进行相应的处理，这样可以方便的剪裁内核并确保 DSP/BIOS 消耗的内存最少[34]。

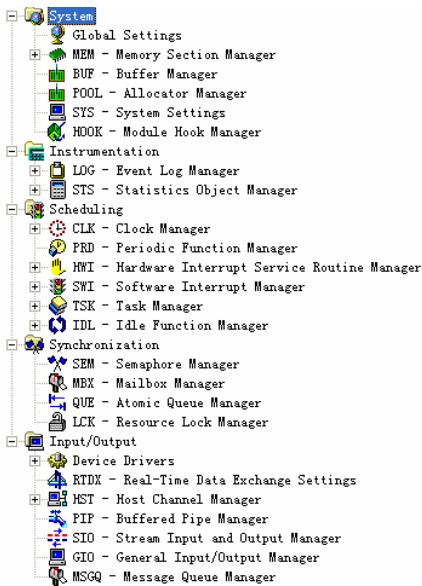


图 2-4 DSP/BIOS 配置工具

DSP/BIOS 配置工具是一个可视化的编辑器，它为用户提供了在程序开发阶段静态的创建和配置 DSP/BIOS 内核对象的功能。使用配置工具可以和调用 DSP/BIOS API 的方式相结合，创建并配置目标应用系统运行时需要的各个内核对象如线程、数据流等。与在程序执行期间调用 DSPBIOS API 动态的创建对象相比，使用配置工具静态的创建和配置对象会大大减小目标代码大小，因为创建对象函数的代码不必链接到最终的目标代码之中了。使用配置工具静态的创建和配置对象的另一个优点是，配置工具可以检查为对象设置的参数是否符合语法，这样在程序开发阶段就可以避免多种错误。而且，配置工具对静态创建的各个对象进行全面检查后，在程序实际执行前就可以精确计算出程序运行需要的数据存储区大小、堆栈大小等信息[34]。

### 2.3.3 DSP/BIOS 驱动程序开发

本文使用德州仪器定义的类驱动模型（class-driver）和微驱动模型（mini-driver）来开发此系统中芯片的驱动。类驱动模型一般提供对多线程 I/O 请求的串行化和同步，另外还负责设备实例的管理。在典型的实时系统中，最多只需要几类的类驱动用于代表不同类型外围设备的驱动，其中包括异步 I/O 类、字符 I/O 类和视频类。微驱动模型能处理具体外围设备特有的所有控制和初始化操作，所有的微驱动模型都会符合一个称为 IOM 的标准接口[38]。

一个有效的驱动模型应该能够采用 I/O 请求包的形式来有效地代表一个发往微驱动的 I/O 请求，即将请求信息包含在驱动开发者定义的一个结构体中。图 2-5 给出了使用双层设备驱动模型的应用程序中各层之间的关系。

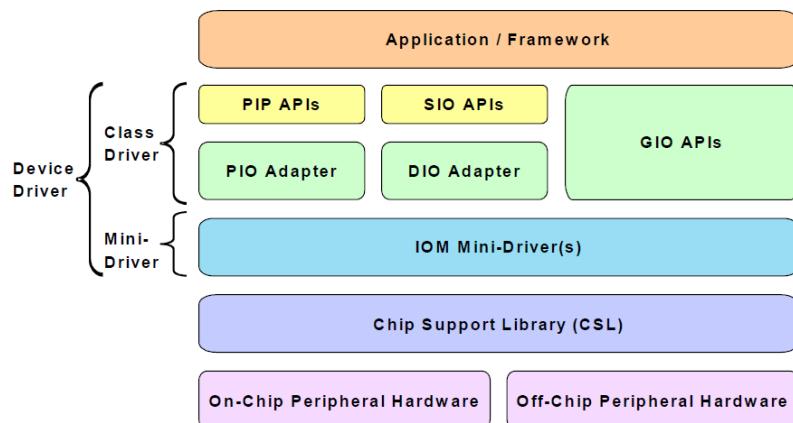


图 2-5 应用程序结构中的设备驱动组件

顶层的应用程序并不直接与微驱动交互，而是通过一个或者多个类驱动与微驱动连接。每一类类驱动向应用程序提供一个 API 接口，并负责与 IOM 微驱动接口进行通信。类驱动通过使用 DSP/BIOS API 来实现操作系统级的服务，并调用标准的微驱动接口来访问外围硬件设备。

DSP/BIOS 目前定义了三种类型的类驱动，分别为 PIP/PIO，SIO/DIO，和 GIO。对于 PIP/PIO 和 SIO/DIO 类驱动，应用程序调用的 API 函数就是 DSP/BIOS 中现有的 PIP 和 SIO 函数，这些 API 与相应的适配器（PIO 和 DIO）进行交互，而适配器则直接与微驱动进行通讯。使用 GIO 类驱动时，应用程序则调用一套新的 GIO API 函数直接与微驱动进行通讯。

在一个 DSP 应用程序中可能同时存在多种类型的类驱动，应用程序开发者

可以选择在一个系统中使用一种或者多种类驱动，而对于微驱动开发人员就不必开发类驱动，因为每个微驱动都提供一套标准的微驱动接口函数，供类驱动使用以访问该外围设备，以及供 DSP/BIOS 使用来管理设备驱动。微驱动利用片级支持库 CSL 来访问外围设备寄存器，存储器和中断源。

DSP 应用程序使用的每个 DSP/BIOS 模块都包含一个在 DSP/BIOS 初始化时由 DSP/BIOS 调用的初始化例程。DSP/BIOS 如同对待其他模块一样来对待微驱动模型。对于每个被注册到 DSP/BIOS 配置中的微驱动，其初始化函数是在 DSP/BIOS DEV 模块的初始化过程中被调用的。

每个微驱动提供的接口函数表中都包括一个绑定函数 (mdBindDev)，用于绑定指定的设备实例到该微驱动。该绑定函数会在所有驱动的初始化函数调用执行完之后被 DSP/BIOS 调用。通过 mdBindDev 函数的参数，类驱动可以传递一些配置参数给微驱动，如 IRQ 优先级、静态声明的全局存储区等信息。另外类驱动也会为被绑定到微驱动的具体设备实例创建一个通道。这样使得一个微驱动只使用一个配置好的特定的存储区域，而各个设备实例则共享其中的全局数据。

类驱动+微驱动这样的双层驱动模型包括两种类型的对象实例：设备实例和通道实例。设备实例是对一类物理设备的抽象，对应于一个实际的物理设备（例如一个 VP 口），然而同类的设备在一个 DSP 中可能不止一个（例如 DM648 中有 VP0、VP1、VP2、VP3 和 VP4），因此需要支持多设备实例化，即注册同类微驱动的多个设备实例，来驱动多个物理设备并确保清晰的驱动状态，从而使多个物理设备使用同样的微驱动代码。设备实例需要被注册到 DSP/BIOS 的设备表中，如图 2-6 所示。DSP 应用程序通过设备实例的逻辑名称来引用它们。如果配置了多个设备实例，那么在设备表中每一个实例都具有一个唯一的逻辑名称。

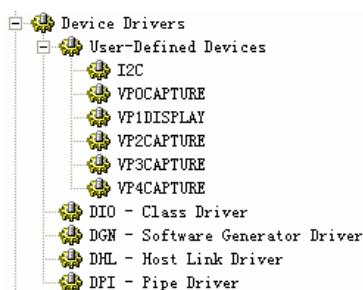


图 2-6 DSP/BIOS 中的设备表

一个设备实例可以支持多种模式的通道实例。通道实例是应用程序和设备实

例之间通讯的路径的抽象。通过调用微驱动模型的 `mdBindDev` 函数可以创建通道实例。类驱动的某些 API 函数（例如 `SIO_create`、`PIO_create` 以及 `GIO_create`）被调用时会运行 `mdCreateChan` 函数来创建一个通道实例。通道实例的一个重要的属性是其操作方向模式，微驱动中通道的模式可以为输入、输出或者输入输出。如果一个应用程序在创建通道时试图指定一个微驱动并不支持的通道模式，微驱动会返回一个错误状态。

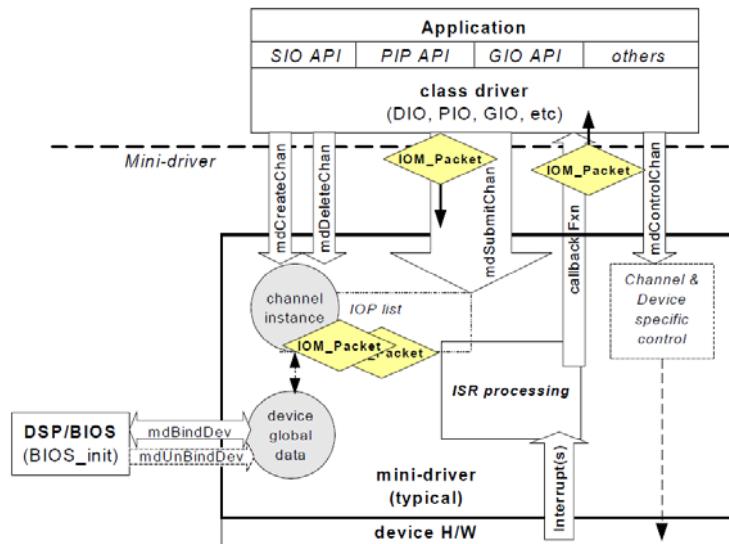


图 2-7 微驱动的操作流程

### 2.3.4 RF5 参考框架

在本系统应用程序设计过程中，采用了 TI 的 RF5 参考框架。存储体管理、设备驱动、多线程创建、通道封装等已在参考框中配置好，应用程序开发者只需集中精力开发应用实现部分即可。RF5 参考框架常用于高集成应用，它支持 100 个通道和众多 eXpress DSP 兼容算法。该结构可用于通讯、音频、视频等[17]。

RF5 参考框架共有 4 个基本的数据处理部件：任务（TSK）、通道（CHAN）、单元（CELL）和标准算法（XDAIS algs），4 部分由很多模块组成，如图 2-8 所示。通常，某个任务中包括一个或多个通道，每个通道中包括一个或多个单元，而每个单元中则封装有一个标准算法。通道可以按序执行多个单元。在典型应用中，多个通道可能包含一套执行功能相同的单元序列，而仅仅参数不同，如视频的 Y, Cb, Cr 三个通道的处理。任务可以同时处理一个或多个通道，其目的在

于组织任务间的数据通讯和设备驱动会话等。与通道不同的是，任务有具体的执行代码，并需要用户自己编写。该部分代码通常是从外界接收数据、控制通道执行等。每个任务总是反复执行自己的代码，完成检查控制信息、获得数据、执行通道、发送数据等操作。

RF5 参考框架中的数据通讯包括任务级通讯和单元级通讯，其通讯机理为使用结构体进行信息传递，而非通过全局变量传输处理数据。任务级通讯主要用 SCOM 消息对列（SCOM queue）和邮箱（MBX）。单元级通讯基于数据缓冲区，且存在一个内部单元通讯对象（ICC 对象），用于对缓冲区的描述。每个单元的输入、输出队列均指向该 ICC 对象。

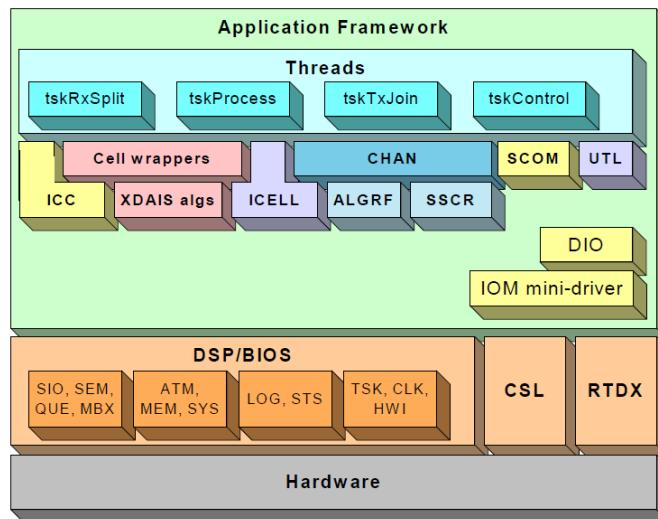


图 2-8 RF5 参考框架中的各个模块

## 2.4 本章小结

本章综述了 DSP 处理器的结构特点，并且给出了在实际的嵌入式图像处理系统中的 DSP 选型。本章还总结了系统所用的 TMS320DM648 的特点，并且给出了 DSP 上软件开发的流程。本章是进行对系统进行软硬件开发的基础。

### 第三章 基于 TMS320DM648 的全景视频监控系统原型

全景视频监控系统的硬件系统的核心为 TMS320DM648 多媒体处理器，采用两片 TVP5154 视频解码芯片把 8 路视频模拟信号转化为数字信号输入，经过图像拼接算法的处理，把 8 路视频图像拼接成全景图，再经过一片 SAA7105H 视频编码芯片把数字信号转化为模拟信号输出，显示在电视机或者其他终端。

#### 3.1 系统结构

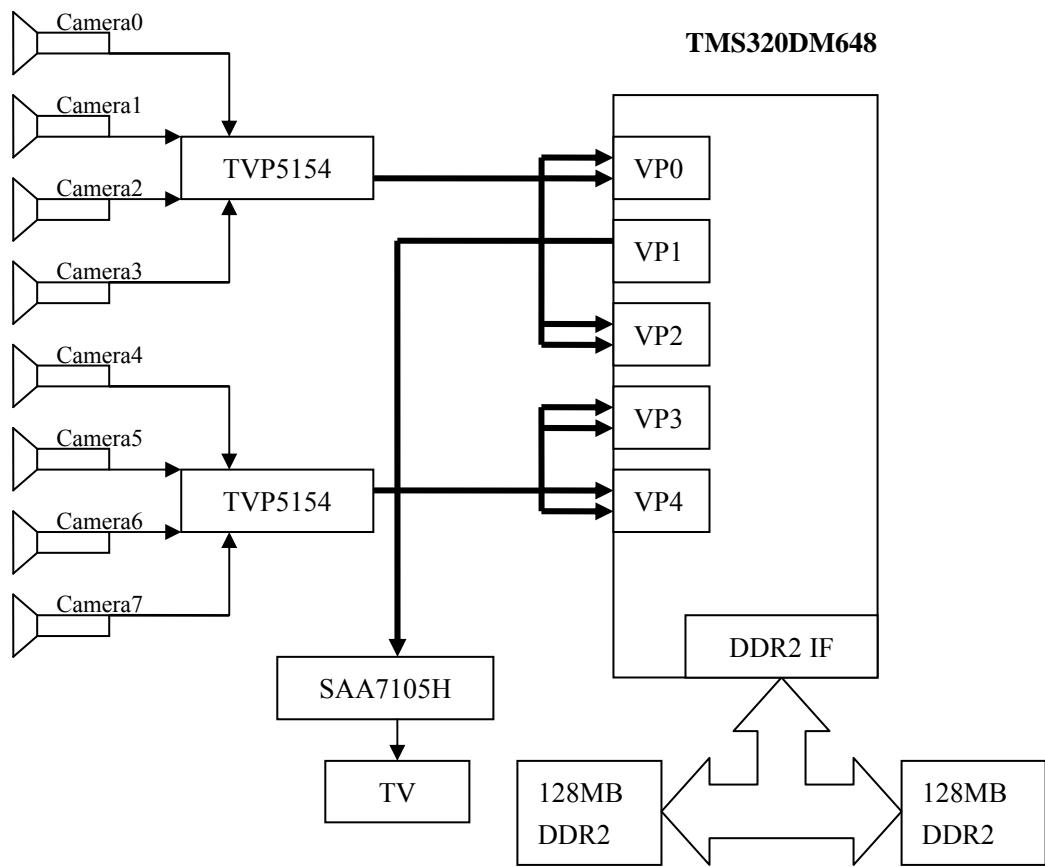


图 3-1 全景视频监控系统整体框架

全景视频监控系统的整体框架如图 3-1 所示。系统其主要功能是完成视频图像的采集、存储、处理和显示，主要由视频输入模块、图像处理模块、存储模块和视频输出模块组成。

**视频输入：** 模拟信号经过摄像头采集进来，通过 TVP5154 视频解码芯片转换为 BT.656 数字信号，再经过 DM648 对每一帧进行图像处理后，数据直接通过

视频输出口回放。

视频输出：从已经采集的视频存储器中取得数据或直接从外部采集得到视频数据，经过 SAA7105H 视频编码芯片把数字信号转换成模拟信号，最终在电视等终端输出信号。

存储模块：8 路视频帧存放在基于 DDR2 的存储系统中。此系统中使用了 2 片 128MB 的 DDR2 芯片与 DM648 上提供的 DDR2 接口相连，组成了系统的片外的存储系统。

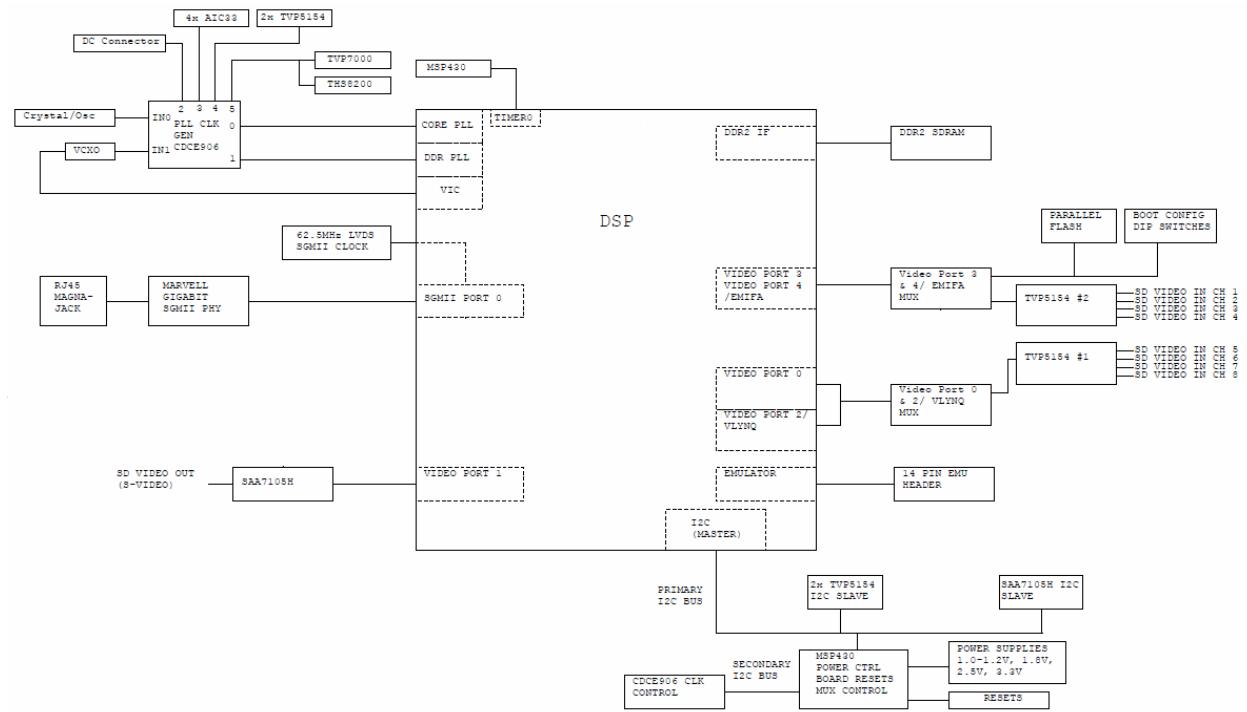


图 3-2 全景视频监控系统框图

图 3-2 为整个系统硬件的系统框图，此系统中只使用了几个必要的接口，分别为 DDR2 IF、Video Port 0-5、Emulator、I2C、SGMII Port 以及 PLL 锁相环等接口。下面对此系统中主要的子系统以及涉及到的芯片进行说明。

### 3.2 基于 DDR2 的存储系统的设计

DM648 片上集成了 DDR2 存储控制器接口，可外接标准的 DDR2 SDRAM。两片 128MB 的 DDR2 SDRAM 主要用来缓存 8 路的视频输入图像，以备后面的图像拼接使用。

在现在的嵌入式系统中，处理器芯片的速度越来越快（例如此系统中使用的 DM648 主频已经达到了 900MHz），相应的也要求存储器的数据吞吐量能够尽可能地缩小处理的瓶颈。DDR2 技术采用了在时钟的上升/下降沿同时进行数据传输的基本方式，但 DDR2 内存却拥有两倍于上一代 DDR 内存预读取能力（即：4bit 数据读预取）。换句话说，DDR2 内存每个时钟能够以 4 倍外部总线的速度读/写数据，并且能够以内部控制总线 4 倍的速度运行。所以，采用 DDR2 SDRAM 作为最主要的外部存储空间，其数据吞吐量已能满足几乎全部嵌入式系统的需求 [10]。

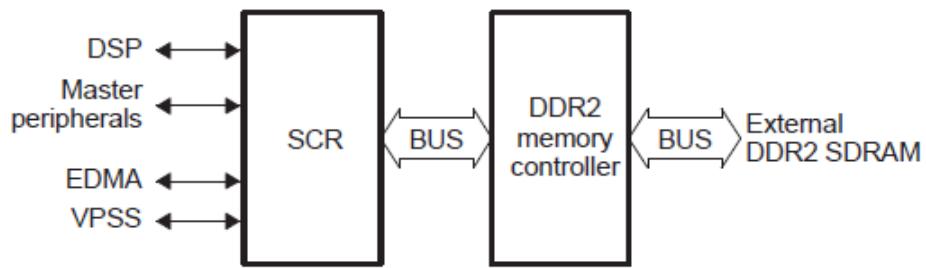


图 3-3 DDR2 存储控制器的数据通路

图 3-3 即为 DM648 上片外设与 DDR2 SDRAM 组成的外部存储系统之间的数据通路，DSP，EDMA，VPSS 以及其他外设可以通过中央交换网络（Switched Central Resource, SCR）存取 DDR2 SDRAM 中的数据。

DM648 中的 DDR2 存储控制器的主要任务是周期性地刷新 DDR2 SDRAM 中存储的数据，使得在需要的时间内保持 DDR2 SDRAM 中数据有效。该控制器最高可支持到 DDR2 400。DDR2 400 的数据吞吐量理论上可达 1.2GB/s 以上，平均的数据吞吐量也在 1GB/s 以上，比传统平台所搭载的 SDRAM 数据吞吐量提高了一倍以上，充分的满足了视频应用高存储器带宽的需求。此系统中共分配给 DDR2 SDRAM 256MB 的地址空间。DDR2 存储控制器上有 32 位数据总线、13 根地址线、3 根 bank 选择信号以及一些控制信号，图 3-3 为全景视频监控系统中的 DDR2 存储系统的原理图。

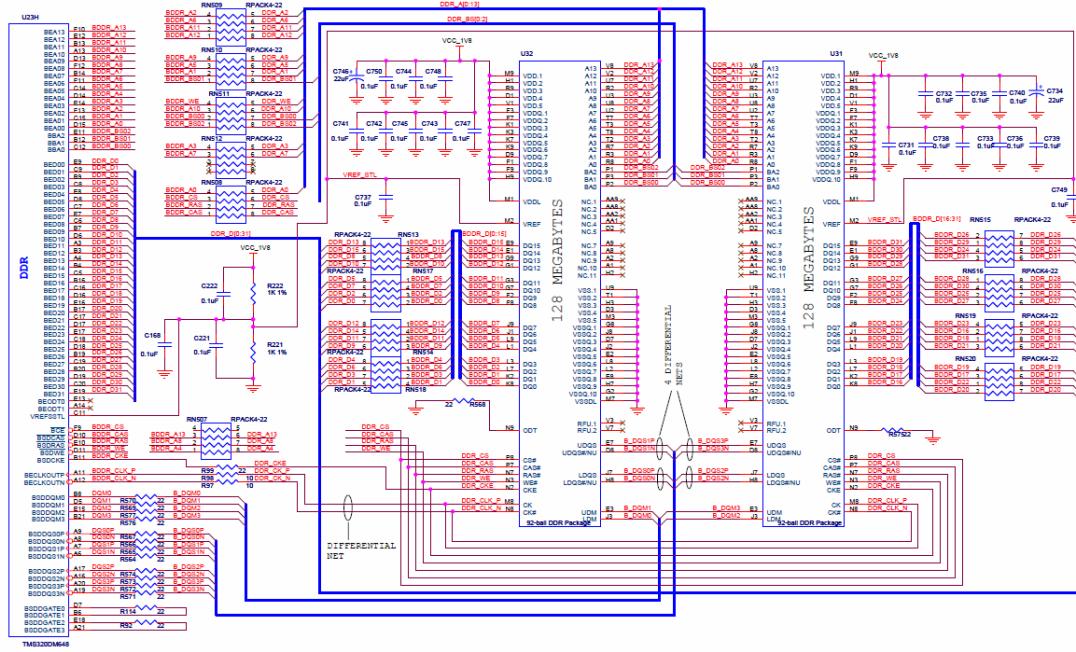


图 3-3 DDR2 存储系统的原理图

### 3.3 IIC 总线

#### 3.3.1 IIC 总线的概念

在此系统中，DM648 通过 IIC 总线来对其片外的芯片进行控制和配置[9]。

IIC (Inter-Integrated Circuit) 总线是一种由 Philips 公司开发的串行总线，用于连接微处理器及其外围设备，它具有以下特点：

- (1) 总线只有两条线：一条为串行数据总线 (SDA)，一条为串行时钟总线 (SCL)。
- (2) 每个连接到总线的器件都可以使用软件根据它的唯一的地址来识别。
- (3) 传输数据的设备间是简单的主/从关系。
- (4) 主机可以用作主机发送器或主机接收器。
- (5) 它是一个真正的多主机总线，两个或多个主机同时发起数据传输时，可以通过冲突检验和仲裁来防止数据被破坏。
- (6) 串行的 8 位双向数据传输，位速率在标准模式下可达 100Kbit/s，在快速模式下可达 400Kbit/s，在高速模式下可达 3.4Mbit/s。
- (7) 连接到同一总线上的 IC 数量只受到总线的最大电容 400pF 的限制。

图 3-4 所示的为多个拥有 IIC 接口的芯片连接到 IIC 的 SDA 和 SCL 线上的示意图。

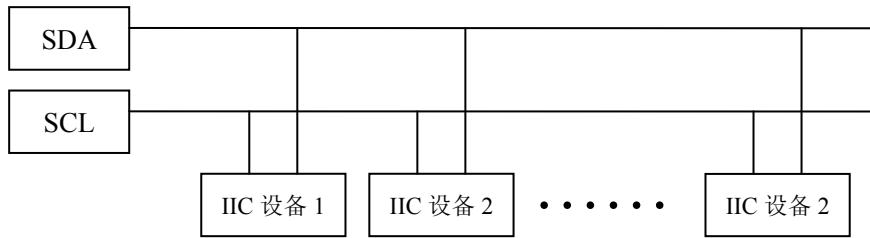


图 3-4 IIC 总线设备互连

### 3.3.2 IIC 总线的信号类型

IIC 总线在传输过程中总共有 3 种类型的信号：开始信号、结束信号和响应信号。

(1) 开始信号 (S): SCL 为高电平时, SDA 由高电平向低电平跳变, 意味着开始传输数据。开始信号的波形如图 3-5 所示。

(2) 结束信号 (P): SCL 为高电平时, SDA 由低电平向高电平跳变, 意味着结束数据传输。结束信号的波形如图 3-5 所示。

(3) 响应信号 (ACK): 接收器在接收到 8 位数据后, 在第 9 个时钟周期, 拉低 SDA 电平。响应信号的波形如图 3-6 所示。

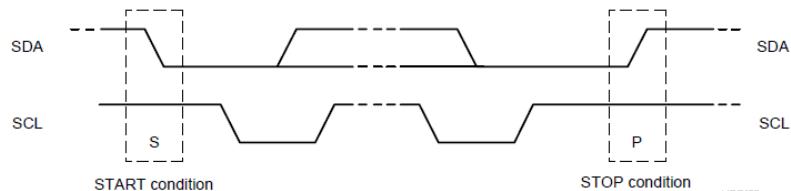


图 3-5 开始信号 (S) 和结束信号 (P)

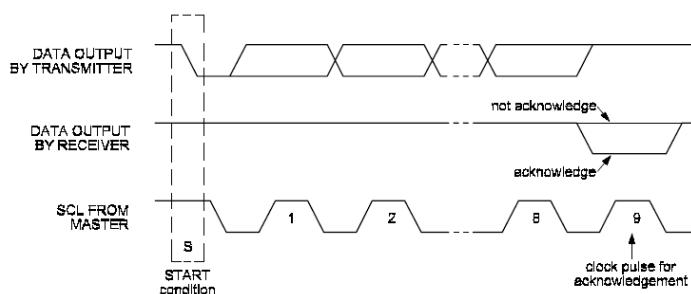


图 3-6 响应信号 (ACK)

SDA 上传输的数据必须在 SCL 为高电平期间保持稳定，SDA 上的数据只能在 SCL 为低电平期间变化，如图 3-7 所示。

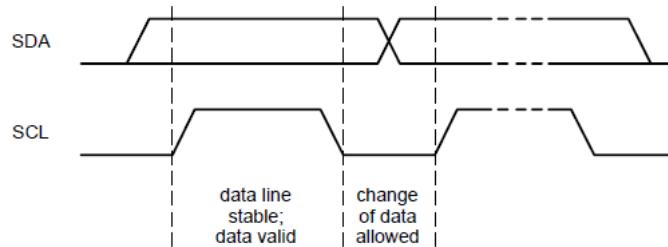


图 3-7 IIC 总线上的位传输

### 3.3.3 IIC 总线的数据传输格式

发送到 SDA 数据总线上的每个字节必须是 8 位的，每次开始位之后可以发送的字节数量不受限制，每个字节发送完后必须跟一个响应位。首先传输的是数据的最高位。如果从机需要完成一些其他功能后（例如响应一个中断）才能继续接收或发送下一个字节，从机可以拉低 SCL 迫使主机进入等待状态。当从机准备好接收下一个字节并释放 SCL 后，数据传输继续。如果主机在传输过程中也需要完成一些其他功能（例如响应一个中断），也可以拉低 SCL 以占住总线。

启动一个传输时，主机先发出 S 信号，然后发出 8 位数据。这 8 位数据中前 7 位为从机的地址，第 8 位表示传输的方向（0 表示写操作，1 表示读操作）。被选中的从机发出 ACK 信号，紧接着传输一系列字节及其响应。最后主机发出 P 信号结束本次传输。

图 3-8 是 IIC 总线上完整的数据传输的格式。

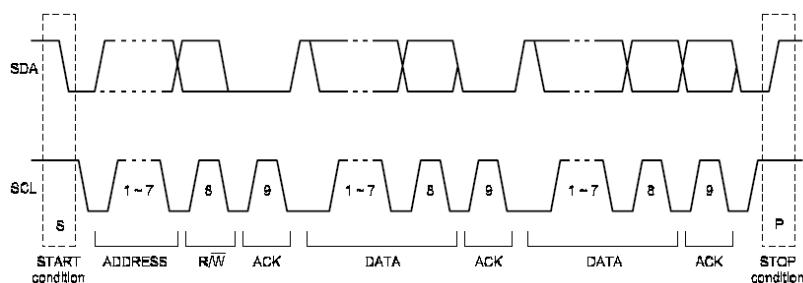


图 3-8 完整的数据传输

### 3.3.4 系统中 IIC 总线的连接

DM648 片上外设中有 IIC 总线的接口，这样方便了实现整个系统。在此系统中，DM648 上的 IIC 总线接口总作为主机，用来配置其他芯片上的寄存器。IIC 与其他芯片器件相连的框图如图 3-9 所示，其中把 2 片 TVP5154、1 片 SAA7105H 以及 MSP430 控制器连接在 IIC 总线上[9]。

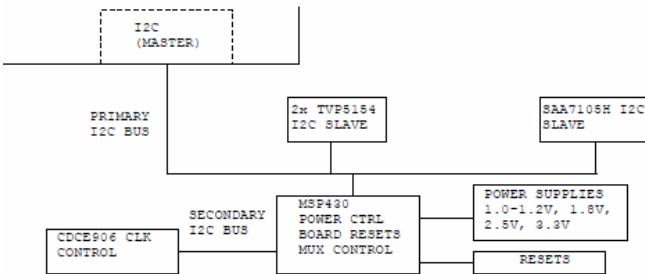


图 3-9 IIC 总线的连接

## 3.4 视频采集系统的设计

TVP5154 单个芯片是一个支持 4 通道的 NTSC/PAL/SECAM 等多种视频格式的视频解码芯片，并且不仅允许影像输出比例的缩小，而且能够提供多种选项，以支持全尺寸或按比例缩小影像的单输出或同时输出。TVP5154 承担了缩放功能，从而为视频处理器节省了宝贵的处理能力。TVP5154 芯片上的每一个通道都是一个可编程的独立的视频解码器，TVP5154 采用 4 个专用 IIC 地址，从而使单个 IIC 总线能够连接 4 个 TVP5154 器件（因此可高达 16 个视频通道）。每个 TVP5154 器件仅需单个 14.1MHz 晶振进行驱动，可额外节省板级空间与成本。TVP5154 支持快速锁定模式，这使解码器能够转换不超过 2.5 场视频输入。在输入转换过程中，当前输入的增益与失调设置被存储起来，从而提高了锁定速度，延长了自动增益控制的收敛时间。稳定的同步输出特性在有源视频的情况下可支持固定的线路数量，从而确保了稳定的输出信号[13]。

DM648 中有 5 个 VP 口，其中每个 VP 口可配置为两通道视频输入，也可配置为一通道视频输出[7]。当 DM648 的 VP 口作为 8 位视频口时，使用 10 位数据总线的高 8 位，即 VDIN[9: 2] 或者 VDIN[19: 12]。VP 口支持多种视频格式。其中本系统中使用 8/10-bit ITU-R BT.656 格式。Video Port 的结构框图如图 3-10

所示。

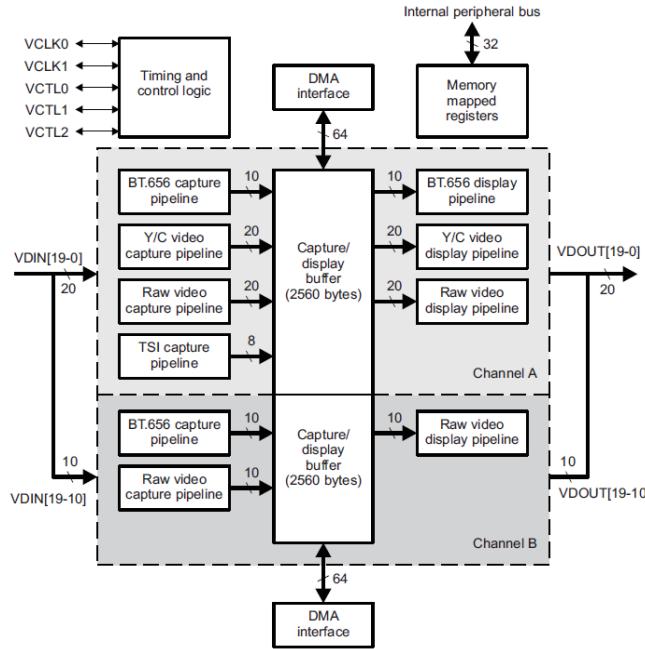


图 3-10 Video Port 的结构框图

在本文的全景视频监控系统中，共使用了 2 片 TVP5154 分别于 DM648 的 VP0 和 VP2 以及 VP3 和 VP4 相连，从而输入 8 路视频信号给 DM648。其电路原理图如图 3-11 所示。

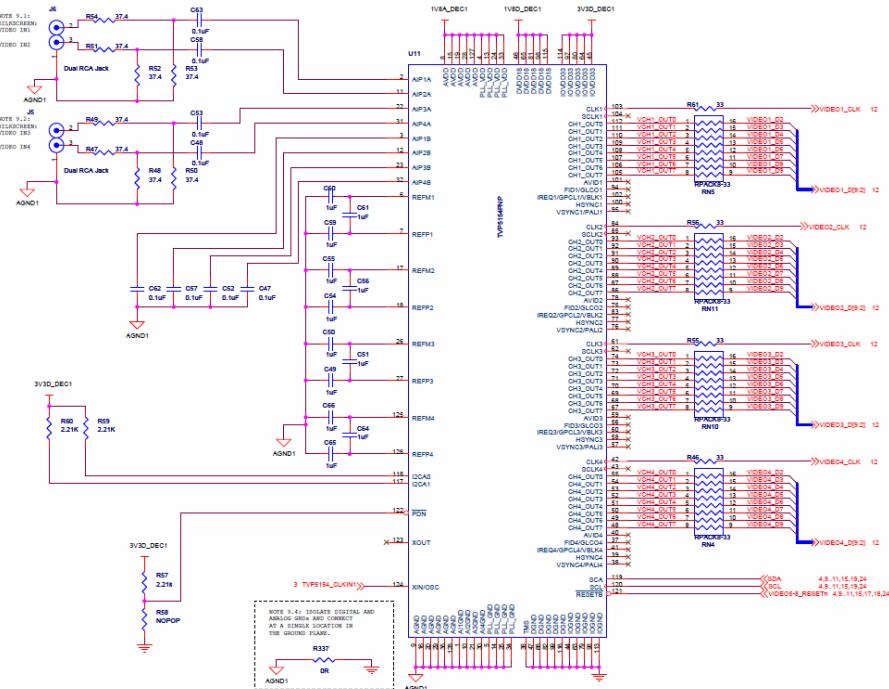


图 3-11 TVP5154 在系统中的原理图

3.5 视频显示系统的设计

本文的系统中选择了使用 SAA7105H 作为处理完视频帧的视频编码芯片，它的输出连接到 S 端子，输出到电视等终端。SAA7105H 是 Philips 公司推出的新一代高级视频编码芯片，现在广泛的应用至图像处理、图像加速卡、图文电视及高清晰度电视等方面。SAA7105H 的输入信号可以是符合 BT. 656 标准的视频格式和数字色差信号(Y-Cb-Cr) 或者是 MPEG 解码的数据；其内置三个 10 位的数模转换器，支持三个模拟通道的 CVBS 信号输出、一路 S 端子信号输出或者一路 VGA 视频输出，支持所有标准的隔行扫描和逐行扫描操作。SAA7105H 的 IIC 总线接口，可以按照主模式或从模式工作。外部器件可以通过 IIC 总线读写其内部寄存器，改变和获取其工作状态[14]。

在此系统中 SAA7105H 芯片需要 27MHz 的晶振来对视频进行编码，其数字视频输入接口与 DM648 的 VP1 相连，然后其输出接口与 S 端子口相连，其电路的原理图如图 3-12 所示。

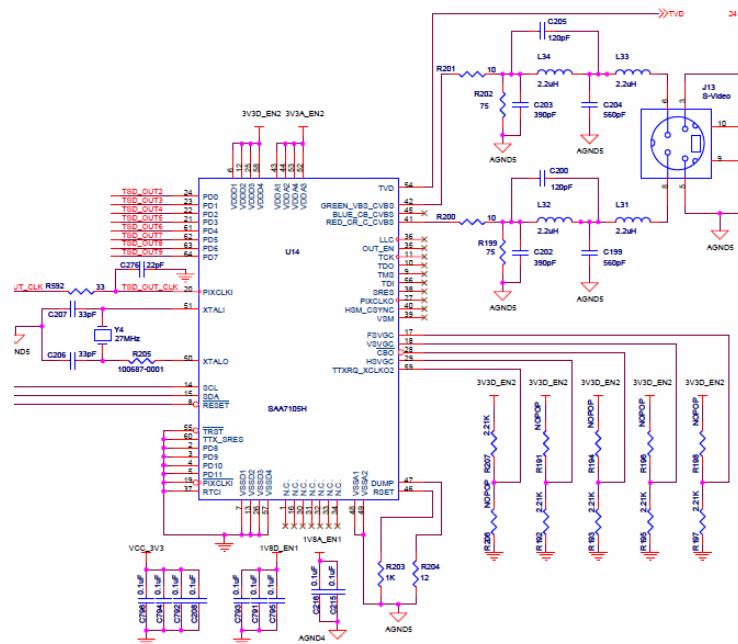


图 3-12 SAA7105H 在系统中的原理图

### 3.6 JTAG 接口电路的设计

JTAG 是基于 IEEE 1149.1 标准的一种边界扫描测试方式。结合配套的仿真

调试软件，可以访问 DSP 的所有资源，包括片内寄存器以及所有的存储器。JTAG 是一种国际标准测试协议，此标准是用来测试端口和边界扫描的，主要用于芯片内部测试。边界扫描接口技术起始于 1980 年，用于解决物理存取问题。这个技术在芯片中封装了测试电路，形成板级测试协议。JTAG 接口标准将极其复杂的电路板测试转换成具有良好结构性的测试，可以通过软件简单而灵活的处理。这个标准定义了可用于完成的功能和互连测试以及内建自测过程的各种指令。

JTAG 仿真器是开发调试环境与目标板之间的接口。JTAG 仿真器的一端通过一个 14 针的接插件与 DM648 芯片的 JTAG 端口进行通信。14 针的 JTAG 接口为：TMS、TCK、TDI、TDO，分别为模式选择、时钟、数据输入和数据输出线。JTAG 接口主要用于和上位机的通信，通过 JTAG 仿真器和 JTAG 接口就可以下载程序到片内地址为 0000 0000 开始的 RAM 中。图 3-13 表示 DSP 的 JTAG 接口电路图的原理图，JTAG 的接口分别与 DM648 的 TMS、TCK、TDI、TDO、EMU1、EMU0 相连接，其余引脚根据 IEEE 1149.1 标准上拉或者下拉。通常情况下 EMU1 和 EMU0 信号必须通过上拉电阻与电源 VCC 相连，上拉电阻的推荐阻值为 4.7KΩ[7]。

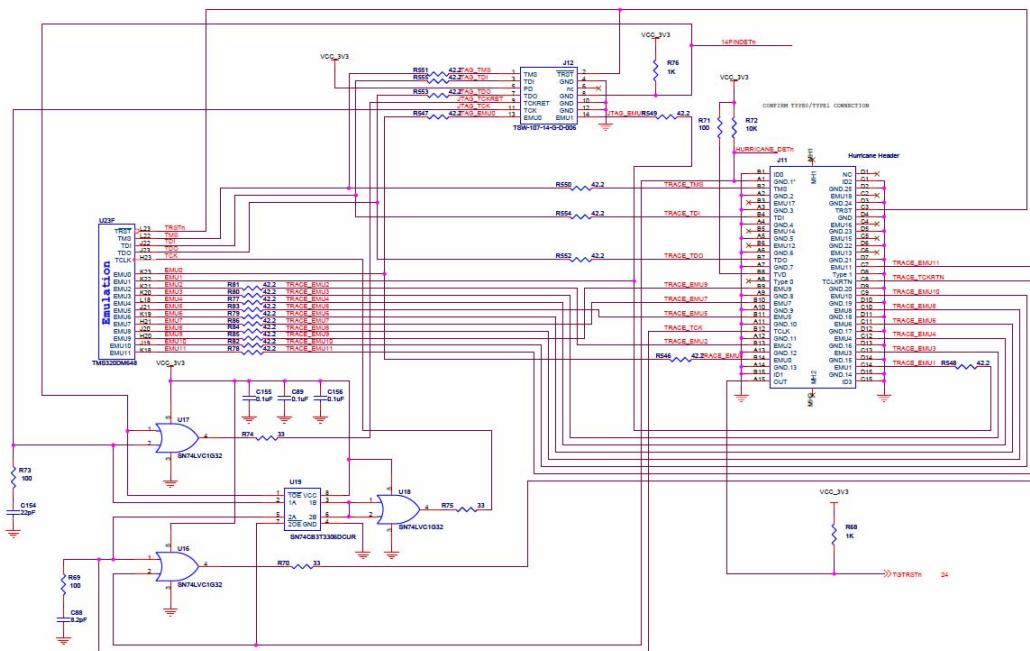


图 3-13 JTAG 接口电路的原理图

### 3.7 时钟电路的设计

DM648 上有两个 PLL 锁相环控制器，分别为 PLL1 和 PLL2。PLL1 控制器是 DM648 中的主要的锁相环控制器，其为 DSP 和系统中其他外设产生时钟输入（例如 McASP 和 EMIF）。PLL2 控制器主要为 DDR2 产生时钟[7]。

PLL1 控制器主要由可编程的乘法器控制器, 分频器 (PREDIV, D1, D2, D3, D4, D5, D6) 构成。PLL1 控制器从 CLKIN1 管脚接收一个输入时钟, 由 CLKMODE[1:0]引脚的逻辑状态来决定倍频、分频模式, 产生一个系统的参考时钟(SYSREFCLK)和六个系统时钟(SYSCLK1, SYSCLK2, SYSCLK3, SYSCLK4, SYSCLK5, SYSCLK6)。PLL1 的电源由 PLLV1 电源管脚从外部引入。表 3-1 为 PLL1 产生时钟频率的范围。

表 3-1 PLL1 时钟频率范围

CLOCK SIGNAL	MIN	MAX	UNIT
CLKIN1	25	66.6	MHz
PLLREF (PLLLEN = 1) <sup>(1)</sup>	25	66.6	MHz
PLLOUT <sup>(1)</sup>	400	720	MHz
	400	800 <sup>(2)</sup>	MHz
	400	900	MHz
SYSCLK4	1/16P <sup>(3)</sup>	PLLOUT/(2*(PLLDIV4.RATIO+1)) <sup>(4)</sup>	MHz

图 3-14 为本系统的时钟电路的原理图，采用 27MHz 有源晶振输入，将 CLKMODE[1:0]配置为 0x11，即进行 33 倍频，使得 CPU 时钟频率为 900MHz。

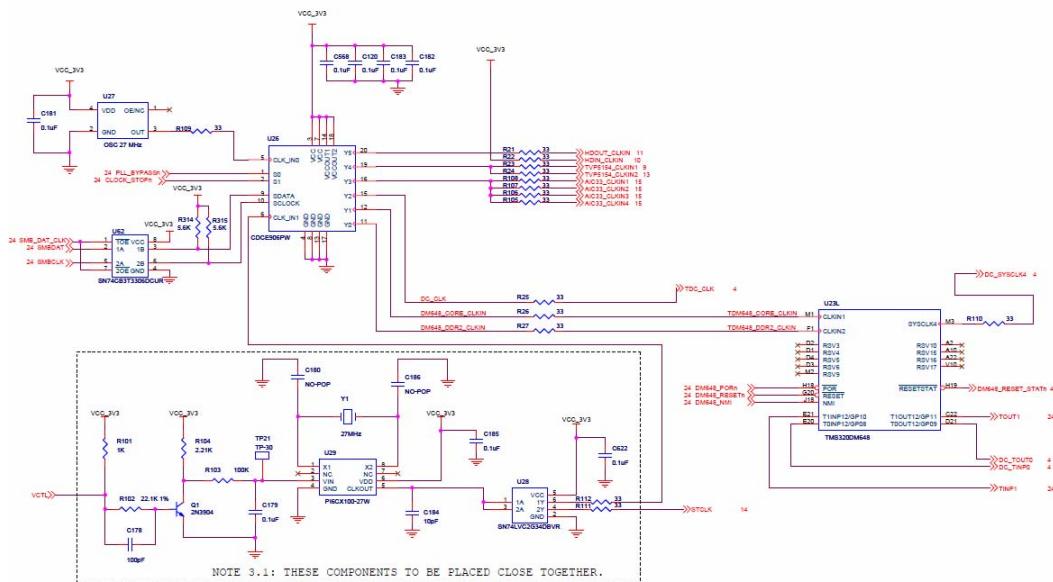


图 3-14 时钟电路的原理图

### 3.8 本章小结

本章主要讲述了整个基于 TMS320DM648 的全景视频监控系统的硬件平台的开发。根据 TMS320DM648 多媒体处理器的外设接口的性质，选择能够达到系统所需功能的芯片，并且根据这些芯片的数据手册，画出了系统中的原理图，从而构建了整个系统的硬件平台的原型。

## 第四章 基于 DSP 平台的全景图像拼接

在全景视频监控系统中，全景图像是通过采集到的 8 路视频图像帧拼接得到的，因此图像拼接算法为此系统软件部分的关键。图像拼接算法可大体分为两种：一种是基于特征匹配的图像拼接算法，另外一种是无特征匹配的图像拼接算法。在系统的实现中，我们采用了基于 SIFT 特征匹配的算法来实现全景图像的拼接。

SIFT (Scale Invariant Feature Transform)，即尺度不变特征变换，是 David G.Lowe 在 2004 年总结了现有的基于不变量技术的特征检测方法，并正式提出的一种基于尺度空间的、对图像缩放、旋转甚至仿射变换保持不变性的图像局部特征点[19]。

SIFT 特征点是目前比较流行的特征描述算法，它能够在发生仿射变换和局部畸变的两幅图像上确定可靠的特征匹配点对。它如此受到大家推崇的原因主要是因为在相同类型的特征点中，它往往能得到最好的效果[31][32]。

SIFT 算法提取的 SIFT 特征向量具有如下特性：

- (1) SIFT 特征描述图像的局部特征，其对图像旋转、尺度缩放、亮度变化具有不变性，对视角变化、仿射变换、噪声也具有一定程度的稳定性；
- (2) 区分性、辨别性(Distinctiveness)好，信息量丰富，适用于在海量特征数据库中进行快速、准确的匹配；
- (3) 多量性，即使少数的几个物体也可以产生大量 SIFT 特征向量；
- (4) 高速性，经优化的 SIFT 匹配算法甚至可以达到实时的要求；
- (5) 可扩展性强，可以很方便的与其他形式的特征向量进行联合。

### 4.1 SIFT 特征点的提取

David G. Lowe 采用高斯差分 DoG(Difference -of-Gaussian)算子，即一种归一化的 LoG (Laplacian-of-Gaussian)算子的近似，对图像进行边缘检测，在边缘图像上提取候选的特征点。

LoG 算子是一种经典的边缘提取算法，不同于目前比较流行的 Roberts、Sobel、Prewitt、Canny 等基于一阶导数算子的边缘检测算子，LoG 算子是二阶导数算子，该算子是一个与方向无关的各向同性的边缘检测算子，它利用二阶导数零交叉所提取的边缘宽度为一个像素单位，所得到的边缘结果无需细化处理，

从而保证了边缘定位精度，其理论基础是 Marr 与 Hildreth 在 20 世纪 70 年代末提出的一种近似人类的视觉系统的“边缘检测理论”。但由于图像边缘和噪声同是频域中的高频分量，二阶导数算子比一阶导数算子更容易受到噪声的影响，因此需要在微分运算之前往往采取适当的平滑滤波。

**DoG 边缘提取算法:** David G. Lowe 基于图像二维平面空间和 DoG 尺度空间，同时检测局部极值以作为候选的特征点，以使特征具备良好的独特性和稳定性。DoG 算子定义为两个不同尺度的高斯核的差分，其具有计算简单的特点，是归一化 LoG (Laplacian-of-Gaussian) 算子的近似[19]。DoG 算子如式(4-1)所示：

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (4-1)$$

#### 4.1.1 高斯低通滤波

LoG 算子在边缘检测之前，利用高斯函数进行低通滤波，然后再利用拉普拉斯算子进行高通滤波并提取零交叉点，构成 LoG (Laplacian-of-Gaussian) 算子。在系统中高斯滤波函数定义为二维高斯函数，如下所示：

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (4-2)$$

其中， $\sigma$  为高斯滤波器标准方差，它决定着对图像的平滑程度，在 SIFT 算子的 DoG 边缘检测中，通过改变  $\sigma$  值的大小决定图像中边缘和其它尖锐不连续的部分的模糊化程度。采用不同方差的 LoG 算子处理图像，可以得到图像在不同分辨率上的描述。较小  $\sigma$  的 LoG 算子可以得到较准确的边缘定位，同时能获得较多的图像细节，但对噪声较为敏感；相反，较大  $\sigma$  的 LoG 算子对噪声滤波效果较好，但容易丢失重要的边缘信息，对相互临近的边缘容易将它们连接在一起，从而影响了边缘检测的效果[19]。

对图像进行高斯低通滤波，如下所示：

$$L(x, y, \sigma) = f(x, y) * g(x, y, \sigma) \quad (4-3)$$

其中，\* 表示在 x, y 方向上对图像进行卷积。当对原始图像采用不同的  $\sigma$  不断地进行高斯滤波后，就形成了一组高斯平滑后的图像。

在进行高斯低通滤波平滑时，为了提高算法的实时性，系统中采用了 Q 格

式表示的高斯滤波窗口来对一幅图像进行滤波。在我们的实验中，我们对一个高斯窗口中的元素用 Q16 的格式表示。这样表示后，每一个元素所占用的位数小于一个半字，所以可以用 DSP 中快速的乘加指令对这幅图像进行滤波。

其中 C64x+ 指令集中的 DOTP2 指令可以在 4 个时钟周期内把两个 32 位的操作数的高 16 位相乘，以及低 16 位相乘，再把相乘的结果相加，增加了处理速度。但在系统实现的时候，高级语言和汇编语言的混编是一件非常麻烦的事情，这时可以利用 TI 公司的编译环境 CCS 中提供的 intrinsics 函数。

C6000 编译器可以识别若干 intrinsics 函数。Intrinsics 函数直接调用某些汇编语句，这些汇编语句在 C/C++ 中实现起来是很繁琐或难以表达的[12]。所以在进行高斯滤波的时候，就调用了 \_dotp2 这个 intrinsics 函数来对整个高斯窗口中的元素进行乘积的累加。图 4-1 便为使用  $\sigma=1.6$  对第一幅 6 次高斯滤波后的结果。

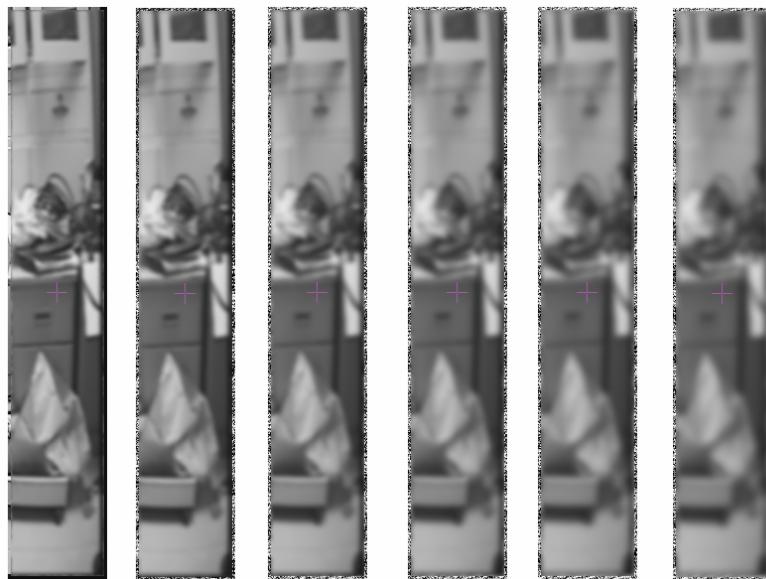


图 4-1 用  $\sigma=1.6$  对第一幅图像进行 6 次高斯滤波后的结果

#### 4.1.2 DoG 图像金字塔的构建

得到一系列用不同  $\sigma$  的高斯核进行高斯滤波平滑后的图像后，就可以构建高斯差分图 DoG(Difference of Gaussian)。1994 年 Lindeberg 研究发现高斯差分函数 DoG 与尺度标准化的高斯拉普拉斯函数非常相近，2002 年 Mikolajczyk 通过实验发现相对于其它的特征提取函数，如 Harris 检测子、Hessian 检测子等，通过求  $\sigma^2 \nabla G$  的极值也能得到稳定的图像特征点，所以利用高斯差分函数 DoG 也可

以得到稳定的图像特征点[19]。

根据 David G. Lowe 的推导,  $\sigma^2 \nabla G$  和 DoG 的关系为:

$$\sigma \nabla G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (4-4)$$

因此可以推得

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1) \sigma^2 \nabla G \quad (4-5)$$

其中  $k$  是一个常数, 它的取值并不影响极值点的检测。当  $k$  趋向于 1 时, 约等式的误差越小, 但在实验中发现, 约等式对稳定的极值点的检测并没有很大的影响, 所以在实验中, 本系统设定了  $k=1.414$  [19]。

David G. Lowe 提出了构建一个 DoG 的金字塔来进行极值点的检测的方法:

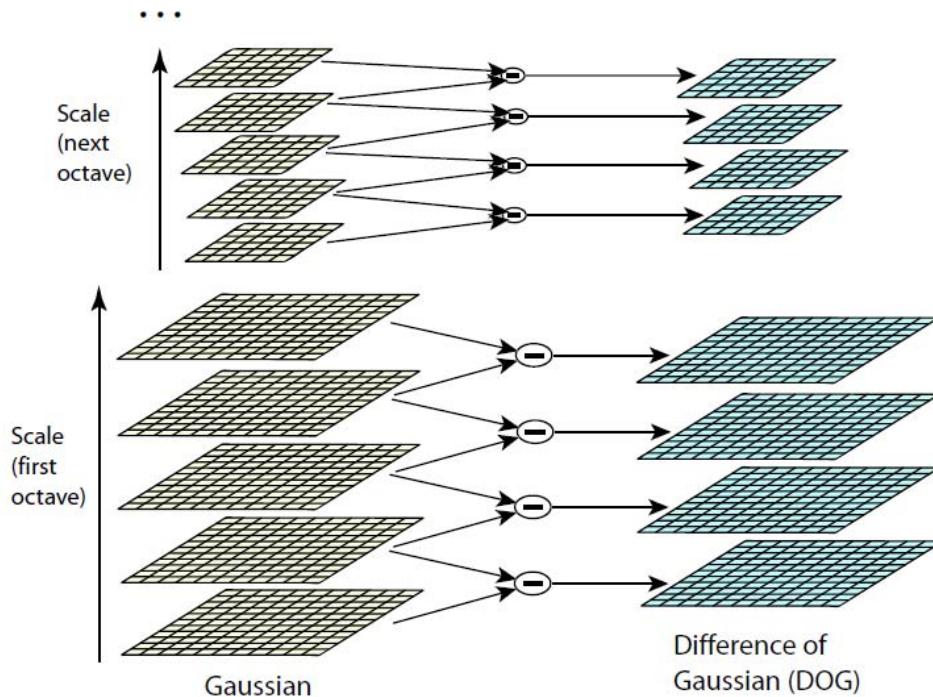


图 4-2 DoG 金字塔的构成

其中 Lowe 建议在 DoG 金字塔中, 每 5 幅 DoG 图像形成一层, 然后再把中间的一幅高斯滤波平滑后的图像进行下采样, 然后对下采样后的图像再次形成一层 DoG 图像, 如此类推, 共形成 5 层的 DoG 金字塔。而在此系统中, 因为摄像头在初始化后就被固定, 因此, 所涉及到的旋转以及拉伸变换相对较少, 所以为了解决匹配初始化的速度, 系统中只建立了一层 DoG, 即平滑过 6 次的图像依次相减产生 5 幅 DoG 图, 如下图所示:



图 4-3 DoG 金字塔的实验结果

#### 4.1.3 测试候选的特征点的稳定性

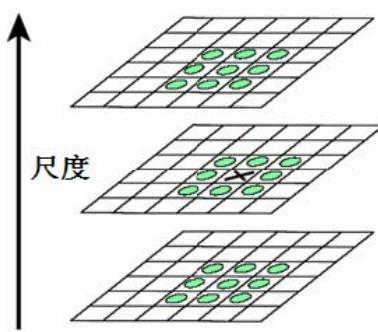


图 4-4 极值点检测

在 DoG 金字塔的同一级上，对不同尺度的三个相邻 DoG 图像，在边缘检测图像上寻找极值，其中的每一幅图像就是图 4-3 中的 DoG 图像，即高斯差分形

成的一幅边缘检测图像。

在检测尺度空间极值时, 图 4-4 中标记为叉号的像素需要跟包括同一尺度的周围邻域 8 个像素和相邻尺度对应位置的周围邻域  $9 \times 2$  个像素总共 26 个像素进行比较(如图绿点所示), 以确保在尺度空间和二维图像空间都检测到局部极值。这样的一个极值点即标记为特征点的一个候选点。

在得到候选的特征点后, 要对得到的候选的特征点进行稳定性测试, 目的是去除一些不稳定的特征点, 把剩下的稳定的特征点描述为 SIFT 特征点, 这样可以增强特征点匹配的稳定性和提高抗噪声的能力[19]。系统中通过三个步骤对一个特征点的稳定性进行测试, 分别为: 对一个特征点进行精确定位, 检测这个点的对比度和边缘响应。

$$\omega = -\left(\frac{\partial^2 D}{\partial l^2}\right)^{-1} \frac{\partial D}{\partial l} \quad (4-6)$$

系统中需要通过拟合公式(4-6)中的三维二次函数以精确定位特征点的位置和尺度。其中  $\omega$  表示当前特征点离其正确位置在  $x$  和  $y$  方向上的偏移量, 以及尺度空间上的偏移量。 $D$  表示此特征点所在的 DoG 图,  $l$  表示了这个特征点的  $x, y, \sigma$  的位置向量。这个公式对特征点及其周围 26 个像素点中每一个点进行一个插值运算, 插值结果会加到这个特征点的  $x, y$  和  $\sigma$  的值上以便用来产生一个新的位置。在此系统中, 当在  $x, y$  和  $\sigma$  方向上的  $\omega$  值小于 0.5 时, 便认为这个位置即是此特征点的精确定位, 如果连续的三个位置都不能满足要求, 就认为这个特征点是不稳定的, 即舍弃这个特征点。

上一步中的偏移量  $\omega$  同样用于计算这个特征点的对比度计算:

$$\xi = D + \frac{1}{2} \left(\frac{\partial D}{\partial l}\right) \omega \quad (4-7)$$

通过式(4-7)计算出结果, 如果计算出的对比度小于一个自定义的阈值, 那么表明这个点的对比度太低, 从而舍弃这个特征点。

对一个特征点进行稳定性测试的最后一步为测试这个点的边缘响应。因为高斯差分运算在边缘上会有很大的响应, 而这些边缘上的特征点往往是不稳定的。在边缘上的特征点的一般表现为: 在垂直方向上的主曲率很小, 而在穿越边缘的主曲率会很大。一个点的主曲率可以通过  $2 \times 2$  的海森矩阵计算得到。

$$H = \begin{bmatrix} \frac{\partial D}{\partial x^2} & \frac{\partial D}{\partial x \partial y} \\ \frac{\partial D}{\partial x \partial y} & \frac{\partial D}{\partial y^2} \end{bmatrix} \quad (4-8)$$

其中 DoG 图上点(x, y)处的一阶导数和二阶导数在图像上可以用一阶差分近似表示，那么二阶导数同样可以用二阶差分代替。通过式(4-8)计算出来的结果，可以判断这个特征点是否有强的边缘响应，上式结果越大，其边缘响应就越大。因此，如果上式结果大于某一个阈值，就舍弃这个特征点，如下式所示：

$$\frac{(\frac{\partial D}{\partial x^2} + \frac{\partial D}{\partial y^2})^2}{\text{Det}(H)} < \text{Threshold} \quad (4-9)$$

通过上面三项检测后，剩下的特征点即为稳定的特征点，便可以对这些特征点进行描述从而形成 SIFT 特征点描述子，如下图所示：

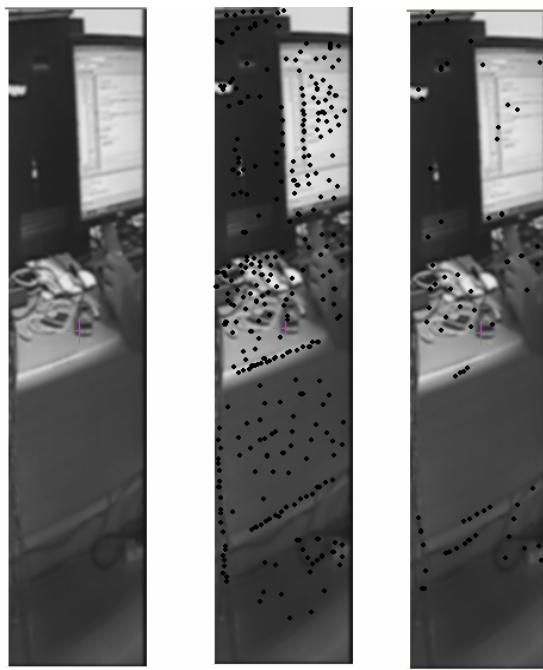


图 4-5 稳定的 SIFT 特征点

图 4-5 显示了稳定的 SIFT 特征点的提取，其中最左边的图为摄像头得到的原始图像，中间一幅图为求 DoG 金字塔中一层的极大极小值得到的极值点（候选点），最右边的图为稳定性测试后得到的稳定的特征点。

## 4.2 改进的 SIFT 特征点的描述子

对于求得的稳定的特征点，系统中采用根据其周围像素的梯度方向对其梯度幅值进行统计的方法对这个特征点进行描述，使其成为一个特征点。这样的梯度

方向统计使对这个特征点的描述更加稳定。

#### 4.2.1 计算特征点的主方向

在对其周围像素进行梯度方向统计之前，要利用特征点邻域像素的梯度方向分布特性为每个特征点指定一个主方向，使描述子具备旋转不变性。

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (4-10)$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))) \quad (4-11)$$

式(4-10)和式(4-11)分别对得到的稳定的特征点( $x, y$ )求出其梯度的幅值和方向。其中  $L(x, y)$  所用的尺度为每个特征点各自所在的尺度[19]。

对于一个特征点的主方向的选取，David G. Lowe 采用了以下方法：在以特征点为中心的邻域窗口内采样，邻域的大小由实验确定，用直方图统计邻域像素的梯度方向。其中每个像素点的梯度方向被分到 36 份中，从 0 度到 360 度，每 10 度一份。梯度方向直方图的峰值则代表了该特征点处邻域梯度的主方向，即作为该特征点的主方向。在梯度方向直方图中，当存在另一个相当于主峰值 80% 幅值的峰值时，则将这个方向认为是该特征点的辅方向。一个特征点可能会被指定具有多个方向（一个主方向，一个以上辅方向），这可以增强匹配的鲁棒性。

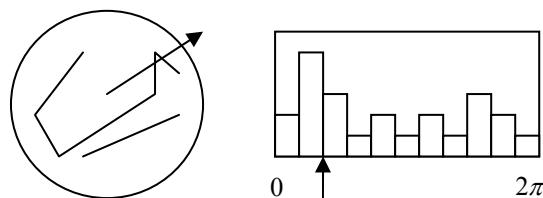


图 4-6 SIFT 特征点的主方向

图 4-6 表示，根据特征点周围像素的梯度方向，对这些像素的梯度幅值进行统计后的结果，通过结果得到 SIFT 特征点的主方向。为了表示方便，其中梯度方向直方图中只显示了 11 个方向的统计，但是发现第二个方向上的梯度幅值累加最大，这个方向即为此特征点的主方向。

与 David G. Lowe 的方法不同，在实验中，我们根据自己的描述子的方向选择，把主方向的选择范围减小到 24 份，即从 0 度到 360 度，每份 15 度。同时，

在此系统中，为了避免求梯度方向和梯度幅值的浮点运算，我们建立了一个查找表，根据每一个像素的  $dx, dy$  进行查找，从而大幅度的加快了运行速度。

#### 4.2.2 生成 SIFT 特征点描述子及其改进

为了使得 SIFT 特征点的描述子具有旋转不变性，首先要将坐标轴旋转为特征点的方向。然后以特征点为中心取  $8 \times 8$  的窗口。在图 4-7 的左图中，中心点为当前特征点的位置，每个小格代表特征点邻域所在尺度空间的一个像素，箭头方向代表该像素的梯度方向，箭头长度代表梯度幅值，图中圈内代表高斯加权的范围（越靠近特征点的像素，梯度方向信息贡献值越大）。然后在每  $4 \times 4$  的图像小块上计算 8 个方向的梯度方向直方图，形成一个种子点，如图 4-7 右图所示。此图中一个特征点由  $2 \times 2$  共 4 个种子点组成，每个种子点有 8 个方向向量信息，可产生  $2 \times 2 \times 8$  共 32 个数据，形成 32 维的 SIFT 特征向量，即特征点描述子。这种邻域方向性信息联合的思想增强了特征点描述子抗噪声的能力，同时对于含有定位误差的特征匹配具有较好的容错性。

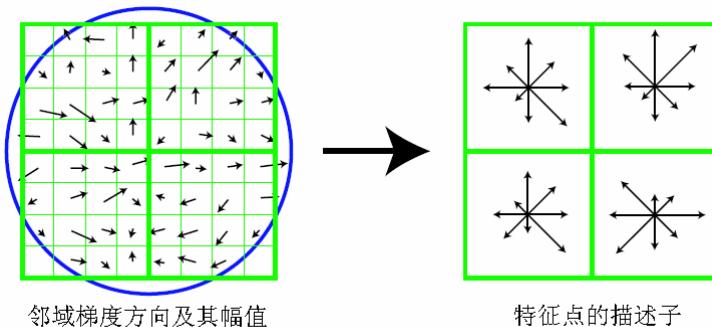


图 4-7 领域梯度及其特征点的描述子

David G. Lowe 提出 SIFT 描述子时，为了增强匹配的稳健性，他建议对每个特征点点使用  $4 \times 4$  共 16 个种子点来描述，一个种子点有 8 个方向向量，即一个特征点就可以产生 128 个数据，最终形成 128 维的 SIFT 特征向量。

Vanderlei Bonato 等人在他们的论文中提出了在硬件资源有限的情况下（例如在 FPGA 上）实现 SIFT 特征点的方法[24]。其中他们分析了，为了达到很高的特征点匹配的正确率同时兼顾了适用于硬件实现这个算法时，FPGA 系统的配置。这些配置对于特征点匹配的错误率由图 4-8 给出。总共有 16 种不同的配置，每一种配置是对 4 种参数的不同选择。第一个参数为在对特征点周围的像素进行

梯度方向直方图统计的时候，是否要根据特征点所在的尺度选择其所在的高斯滤波平滑后的图像。第二个参数为在形成 DoG 金字塔的时候，是否要对原始图像进行 2 倍的放大。第三个参数为是否要对得到的极值点进行精确定位。第四个参数为是否要检验一个特征点的低对比度还有其边缘响应。如图所示，从第 12 种到第 16 种配置所产生的错误率非常的相似（都低于 5%），所以在保证错误率低于 5%的前提下，考虑速度和占用资源的问题，最后选择了第 12 种配置去实现系统中的 SIFT 特征点的提取和描述，即只根据一幅高斯滤波平滑后的图像计算周围像素点的梯度方向和幅值，不需要对原始图像进行 2 倍的放大，要对候选的特征点进行精确定位，以及需要检测特征点的对比度和边缘响应。

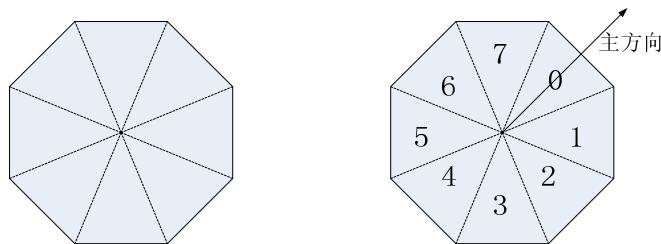


图 4-8 不同的算法匹配得到的特征匹配的错误率

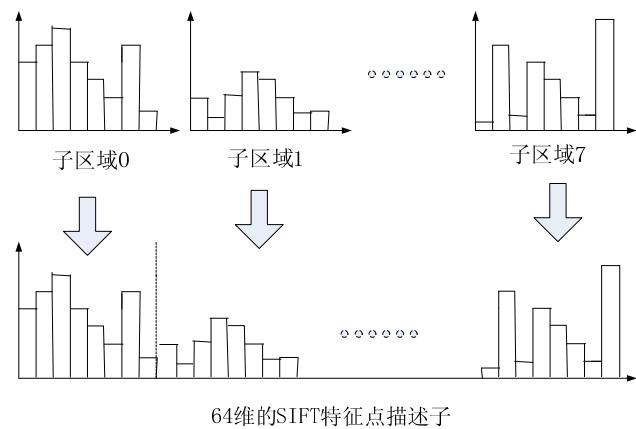
因为在对特征点的周围像素点进行梯度方向直方图统计的时候，需要根据特征点的主方向进行旋转，这样的做法会引入大量的浮点运算。Simon Winder 等人在 07 年总结了 polar samples 的方法，会以较少的计算复杂度和描述子的维数达到很好的效果[25]。这种方法的好处在于描述子是具有各个方向同性现象的，它提高了在空间旋转上的计算速度，使得我们能以很少的计算量达到一个很好的精度。这样使得每个描述子的计算方法一致，只是最后根据主方向对各个统计的子区域进行重新排序，而不是旋转子区域中的各个像素。

本文采用图 4-9 所示的方法去产生一个特征点的描述子。首先，把一个描述子的区域分成了 8 边形的 8 个子区域，如图 4-9(a)所示。每一个子区域的底边离特征点的距离为 10 个像素。然后，计算每个子区域中的梯度方向直方图，计算的时候根据特征点的主方向旋转相应像素点的方向。正如前面所述，我们把 0 度到 360 度分成了 24 份，即每份 15 度，在计算子区域的梯度方向直方图时，我们定义了 8 个梯度方向，即每个方向占 45 度。假设主方向如图 4-9(a)所示，为箭头指向的方向。主方向指向的那个子区域的编号即编为 0，然后按照顺时针方向，对其他 7 个子区域从 1 到 7 编号。最后，如图 4-9(b)所示，按照编号 0 到 7 把 8 个子区域的梯度方向直方图进行重排列，得到了最后的  $8 \times 8$  总共 64 维的描述子向量。

对于生成的描述子，为了增加其鲁棒性，系统中采用了对其边缘进行平滑的方法，即对向量中的每一维都用其相邻的维来进行平滑，用移位寄存器把每一维右移 1 位，然后把相邻的两维分别右移 2 位，再把结果相加，得到平滑后的描述子向量。



(a) 特征点周围的像素分成区域



(b) 由各个子区域的梯度方向直方图形成描述子

图 4-9 SIFT 特征点描述子的生成

### 4.3 SIFT 特征点的匹配

SIFT 特征点匹配需要选取和确定相应的方法去测试相似性。常用的相似性测试方法共分为三种：距离测度、相似度和概率测度。其中距离测度包括均方根误差、差绝对值和误差、欧氏距离和马氏距离[21]。

#### (1) 均方根误差

两图像匹配均方根误差即描述子向量  $x$  与描述子向量  $y$  之差的模的平方根，即

$$\begin{aligned} S &= \sqrt{|x_1 - y_1|^2 + |x_2 - y_2|^2 + \dots + |x_N - y_N|^2} \\ &= \sqrt{\sum_{i=1}^N |x_i - y_i|^2} \end{aligned} \quad (4-12)$$

$S$  是  $N$  维空间点  $x$  和点  $y$  之间的距离。

#### (2) 差绝对值和误差

两图像匹配的差绝对值和就是指描述子向量  $x$  与描述子向量  $y$  之差各分量的绝对值之和，即

$$\begin{aligned} S &= |x_1 - y_1| + |x_2 - y_2| + \dots + |x_N - y_N| \\ &= \sum_{i=1}^N |x_i - y_i| \end{aligned} \quad (4-13)$$

与差绝对值和类似的还有平均差绝对值和

$$S = \frac{1}{N} \sum_{i=1}^N |x_i - y_i| \quad (4-14)$$

#### (3) 欧氏距离

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (4-15)$$

#### (4) 马氏距离

$$d_M(x, y) = (x - y)^T \sum_y^{-1} (x - y) \quad (4-16)$$

马氏距离测度考虑了基准模板特征的离散程度，其获得的分类能力要优于前三种距离测度。它不受量纲的影响，两点之间的马氏距离与原始数据的测量单位无关；由标准化数据和中心化数据(即原始数据与均值之差)计算出的二点之间的马氏距离相同。它的缺点是夸大了变化微小的变量的作用

#### (5) 相似度

定义相似测度  $S(x, y)$  为

$$S(x, y) = \frac{x^T y}{\|x\| \cdot \|y\|} \quad (4-17)$$

式中,  $\|x\| = (x^T x)^{1/2}$ ,  $\|y\| = (y^T y)^{1/2}$ 。 $S(x, y)$  表示模板与匹配子图像间的相似程度,  $S(x, y)$  越大表示模板与匹配子图像越相似。相似度其实就是两矢量间的归一化相关系数, 而在此基础上, 人们又导出了归一化标准相关系数

$$S(x, y) = \frac{(x - \bar{x})^T (y - \bar{y})}{\|x - \bar{x}\| \cdot \|y - \bar{y}\|} \quad (4-18)$$

式中,  $\bar{x}$  与  $\bar{y}$  分别表示矢量  $x$  与  $y$  的均值。

#### (6) 概率测度

通常采用后验概率评估基准图与对准图之间的相似程度。在此种策略下, 后验概率值最大处确定为匹配位置。如果用  $I_L, I_R$  表示图像对,  $d$  表示视差, 那么根据贝叶斯法则, 后验概率分布:

$$p(d | I_L, I_R) \propto p(d) p(I_L, I_R | d) \quad (4-19)$$

其中,  $p(d)$  往往体现的是视差分布特性, 比如满足局部连续性约束的程度。 $p(I_L, I_R | d)$  则表现为给定视差  $d$  时, 图像对  $I_L, I_R$  之间的相似程度 (可以用相关测度或距离测度表示)。在确立后验概率表达式时, 可以在式中加入一些图像内在的约束条件。在匹配度量的过程中能更大程度地减少匹配的歧义性, 因此概率测度被广泛地应用于现今的各类匹配算法中。

本文选择采用欧氏距离来进行 SIFT 特征点的匹配。这样减少计算的复杂度, 提高程序的实时性。不过分别在两幅图像中取得的 SIFT 特征点描述符向量的数量级可能不同, 这样会引起一定的误差, 为了更加精确的衡量特征点对之间的相似性, 采用了以下公式进行欧氏距离的度量:

$$w(p_i^1, p_j^2) = \frac{\sum_{d=1}^D (x_d - y_d)^2}{\sqrt{\sum_{d=1}^D x_d^2} \sqrt{\sum_{d=1}^D y_d^2}} \quad (4-20)$$

其中,  $p_i^1 = (x_1, x_2, \dots, x_D)$  是  $P^1$  集合中的第  $i$  个特征点描述子,  $p_j^2 = (y_1, y_2, \dots, y_D)$  是  $P^2$  集合中的第  $j$  个特征点描述子,  $P^1 \& P^2$  是两幅待配准图像特征点的集合[21]。

图 4-10 即为两个通道视频帧进行 SIFT 特征点匹配的结果, 其中左边一幅图像为一通道视频帧的右边重叠部分, 右边一幅图像为另一个通道视频帧的左边重叠部分, 图中的黑色圆点为能匹配上的 SIFT 特征点, 黑色连线代表了匹配结果。大部分都能匹配成功。

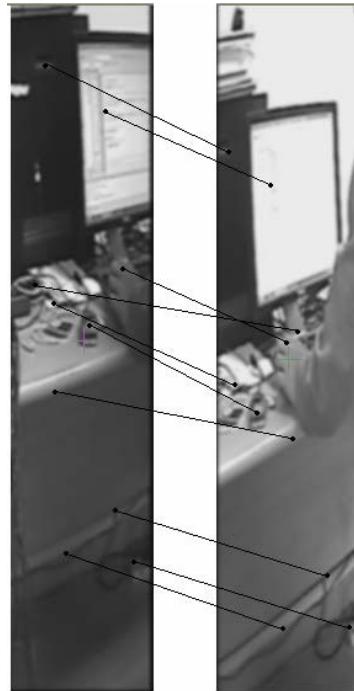


图 4-10 SIFT 特征点匹配结果

#### 4.4 全景图像的实现

根据 SIFT 特征点匹配的结果, 可以算出图像的变化矩阵, 如式(4-21)所示。

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} m_1 & m_2 & t_x \\ m_3 & m_4 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (4-21)$$

每幅图像相对于另一幅图像的变化矩阵都是一个 Affine 变换的矩阵, 其中  $x, y$  代表变换之前的点的坐标,  $u, v$  代表了变换到的图像中此点的坐标。

经过 8 幅视频帧的拼接, 系统实现了全景图像, 如图 4-11 和 4-12 所示。其中图 4-11 为进行图像拼接之前的 8 幅视频帧, 图 4-12 即为拼接之后的全景图。

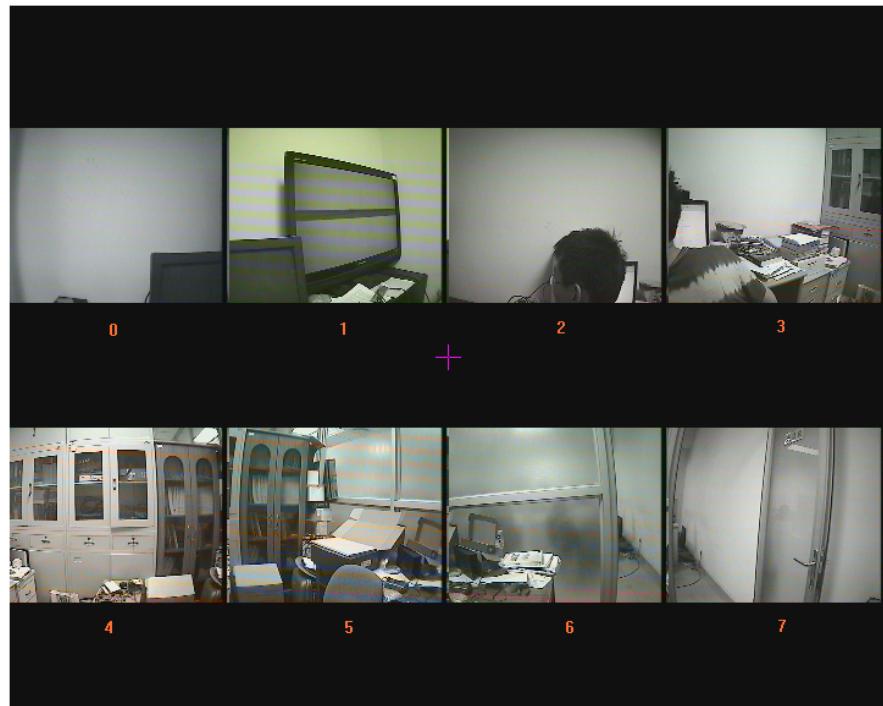


图 4-11 图像拼接之前的 8 幅视频帧



图 4-12 拼接之后的全景图

为了更好地监控整个区域，此系统需要俯视整个监控区域，图 4-13 为 8 幅视频在未进行图像拼接之前的场景图，图 4-14 为俯视整个区域的全景图。



图 4-13 图像拼接之前的 8 帧视频帧（俯视）

在图 4-13 中第 1 个通道的视频帧与第 2 个通道的视频帧的重合区域几乎没有，所以在进行图像拼接的时候，会无法找到正确的 SIFT 特征点匹配，这时本文就采用在全景图像中忽略这两帧的拼接的方法，如图 4-14 所示。



图 4-14 拼接之后的全景图（俯视）

#### 4.5 本章小结

本章是整个系统软件开发的基础，为了在系统中实现全景图像，需要在系统中实现一种图像拼接的技术。在本文的系统中，我们采用了根据 SIFT 特征点匹配的图像拼接算法，达到了很好的鲁棒性。为了克服 SIFT 特征点匹配的不实时问题，本文对 SIFT 提取过程和描述子进行了改进，改进后的实时性有很大的提高。根据 SIFT 特征点匹配得到的变换矩阵，系统能够实时地进行全景图像拼接，实验表明此系统能够完成本文提出的要求。

## 第五章 智能视频监控系统

智能视频监控的目的之一是图像识别，而图像分割是图像识别工作的基础。图像分割是图像识别的关键和首要步骤，分割的效果直接影响到后续处理的效果。图像分割的目的就是把图像中的前景与背景分开，将人们感兴趣的区域（ROI）从图像背景中提取出来，为后续的分类、识别和检索提供依据。图像分割的方法有多种，按算法分类，可分为阈值法、界限检测法、匹配法、跟踪法等。近年来，随着神经生理学、人工智能和模糊理论的发展，人工神经网络、遗传算法、模糊处理方法等等都已经被引入或正将被引入到图像分割与处理这一领域，产生了一些处理效果明显的新算法，如基于模糊处理的分割技术、基于遗传算法的分割技术、基于神经网络的分割技术、基于光流场特性的分割技术、基于图像分形学模型的分割技术等，并取得了明显的效果[2]。

本文的全景视频监控系统采用并实现了动态目标检测进行图像分割和人体目标检测来对监控的区域中的人体进行自动标定，从而达到智能监控的目的。

### 5.1 动态目标检测

动态目标检测是指能从背景图像中检测出运动物体，即指当监控场景背景不变而前景发生变化时，系统能够检测出这一变化，并做出反应，例如：把视频流传到网络上去。由于论文的重点是研究监控系统中的图像处理模块，所以报警这一模块暂时不做处理。从背景中检测、分割和识别出运动的前景物体是很多计算机视觉应用的关键一步，也是实现动态目标分析的前提。动态目标检测的目的是使用基于信号检测的方法，自动分离出运动的像素点和静止的像素点，视频监控系统通过对检测到的运动像素点进行分析来判断运动的类型，最终决定是否有意外情况发生并报警。动态目标检测的原理是按像素点计算特征值绝对差或者按区域计算特征值绝对差的均值。在每个位置上，这个绝对差或者均值与预先设定的或者自适应的阈值进行比较，无论何时超过阈值，该区域或者该点被标注为变化的[35]。

此系统中采用了帧差法来进行动态目标检测。帧差法是在连续的图像序列中两个或三个相邻帧间采用基于像素的时间差分并且阈值化来提取出图像中的运动区域，其流程图如图 5-1 所示。帧差法进行动态目标检测的主要优点是算法简

单，程序设计复杂度低，易于实时处理；对背景或者光线的缓慢变化不太敏感，能较快适应；对目标运动的检测灵敏度高。

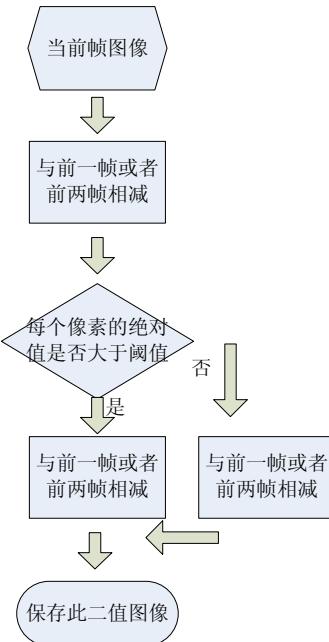


图 5-1 帧差法的流程图

图 5-2 为应用本文的全景视频监控系统，监控某一区域时得到的动态目标，并且用黑色方框标示出来。

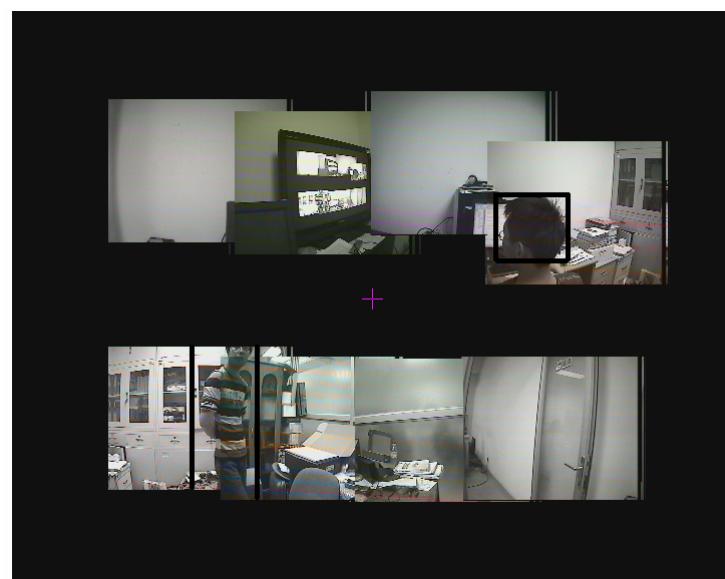


图 5-2 动态目标检测的实验结果

## 5.2 人体目标检测

本文将一种人体检测算法嵌入到此全景视频监控系统中，以达到在动态目标识别的基础上实现人体目标检测。系统中实现的方法为在帧差法得到的运动区域中进行人体目标检测。进行人体目标检测的流程图如图 5-3 所示。

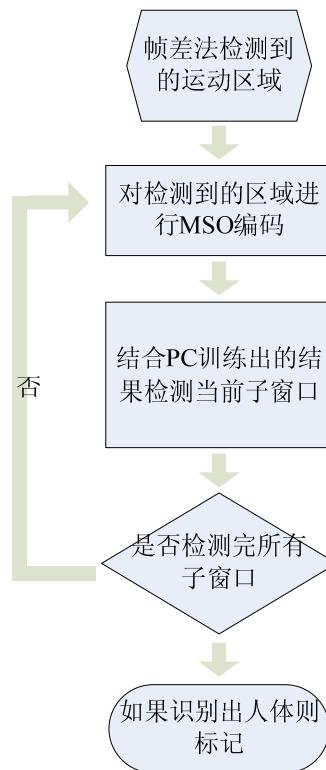


图 5-3 人体检测的流程图

### 5.2.1 MSO 特征

本文的系统中采用了杨志辉等人提出的一种新的 MSO 特征来描述人体[41]。MSO 特征囊括了 Haar-like 和 HOG 的优点，是针对特征区域的形状进行统计的，由全部特征组合起来所形成的特征集反映的是图像在不同尺度上的梯度特征。

在固定尺度上的单个 MSO 特征的计算如图 5-4 所示，其中每个特征区域都是一个  $m \times m$  的正方形，对于这个正方形特征的特征值的计算，可以分为两步：计算正方形区域的毛梯度方向（因为本文这里是在大尺度上求解梯度，相对于原始定义的梯度而言更加粗糙，因此这里将其命名为“毛梯度”）和将计算出来的梯度方向映射成固定的特征值编码。

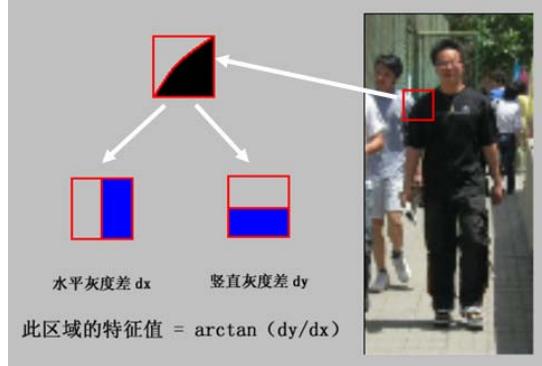


图 5-3 单个特征的计算

首先，将正方形分为左右两个等份，计算左右两个小区域的所有像素的灰度值之和，分别记为  $SUM\_L$  和  $SUM\_R$ ，得出水平灰度差  $dx$ ；再将正方形分为上下两等份，计算出竖直灰度差  $dy$ 。这样这个正方形在形状上便体现了田字形的特征。

如图 5-4 所示，区域 A 的灰度值累加和  $SUM\_A$  的计算公式如(5-1):

$$SUM\_A = \sum_{x1 < x < x2, y1 < y < y2} I(x, y) \quad (5-1)$$

其中， $I(x, y)$  表示坐标为  $(x, y)$  像素点的灰度值，这是一种比较直观的求法，然而本文可以参考积分图的求解过程，这会在很大程度上降低整幅图像的运算复杂度。这样本文便可以求出正方形特征的水平灰度差  $dx$  和竖直灰度差  $dy$ 。

$$\begin{aligned} dx &= SUM\_R - SUM\_L \\ dy &= SUM\_D - SUM\_U \end{aligned} \quad (5-2)$$

其中， $SUM\_R$ 、 $SUM\_L$ 、 $SUM\_D$  和  $SUM\_U$  分别表示为正方形特征区域的右半部、左半部、下半部和上半部的灰度值累加和。

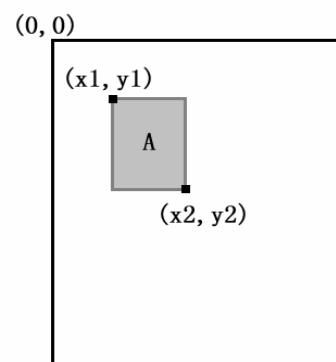


图 5-4 矩形区域的灰度值求和

接着，如图 5-5 所示，我们将 0 到 360 度等分为 18 个区间（并保证对顶角

区域属于同一区间。这样本文就可以根据所求出来的  $dx$  和  $dy$  计算出这个矩形区域内的“大致”梯度方向  $Ori\_Rect$ ,

$$Ori\_Rect = F(\arctan(dy / dx)) \quad (5-3)$$

其中,  $F$  为将角度转换为数值 0, 1, 2, ..., 8 的函数, 如: 当  $\arctan(dy / dx)$  的值在 1 到 20 之间的时候, 则  $Ori\_Rect$  的值为 0; 当  $\arctan(dy / dx)$  的值在 21 到 40 之间的时候, 则  $Ori\_Rect$  的值为 1; 依此类推。

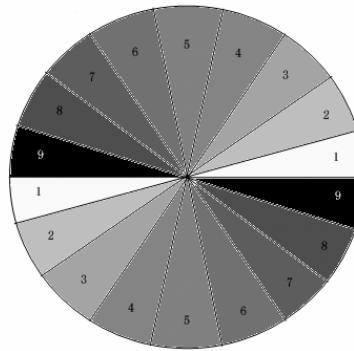


图 5-5 特征空间的划分

### 5.2.2 Adaboost

Adaboost 是 Boosting 算法的一种, 其借鉴分组 (bagging) 核心思想是采用委员会的加权投票机制, 每个委员其实对应一个分类器, 称为弱分类器, 然后把这些弱分类器加权组合起来, 构成一个更强的最终分类器 (强分类器)。Adaboost 算法采用重采样原则和贪婪算法, 每次迭代选择错分率最低的弱分类器组合构成强分类器。Adaboost 通过重采样原则可以改变数据分布, 它根据每次训练集之中每个样本的分类是否正确, 以及上次的总体分类的准确率, 来确定每个样本的权值。将修改过权值的新数据集送给下次分类器进行训练, 最后将每次训练得到的弱分类器最后融合起来, 作为最后的强分类器[41]。

针对每一个 MSO 特征块设计一个弱分类器, 每一个弱分类器的形式如式 (5-4):

$$h_j(x) = \begin{cases} 1 & \text{if } (p_j f_j(x) < p_j \theta_j) \\ 0 & \text{otherwise} \end{cases} \quad (5-4)$$

其中  $h_j(x)$  表示一个弱分类器函数，每个弱分类器函数由一个特征  $f_j$ ，一个阈值  $\theta_j$ ，还有一个用来指示不等式符号的方向的符号函数  $p_j$  组成。在训练的过程中，给定样本标号的前提下，对于每一个弱分类器的阈值，都要决定出一个最佳的阈值，使得训练样本的错误分类数达到最小。默认的 Adaboost 的阈值是用来降低在训练数据时产生的错误率的。一般而言，较低的阈值可以导致一个较高的检测率和一个较高的误警率。

### 5.2.3 级联的 Adaboost 分类器

级联算法的思想是通过使用由粗到精的分级方法，在第一级，选出一些正例和反例的样本构成训练集，使用 Adaboost 算法训练这些样本，通过第一级的所有正例样本被保留进入下一级，未通过的第一级的正例样本被丢弃，通过第一级的反例样本是错分样本，也是和正例样本最相似的样本，从而进入到第二级的训练阶段，继续作为反例样本，因为第二级的反例样本的缺乏，需要从其余的大训练图片中补充反例样本，补充一定量的反例样本和被第一级错分的反例样本最后作为第二级的反例样本，这样使用 Adaboost 进行第二级的训练，以此类推进行多级别的训练，直到满足特定的训练误差要求为止[42]。

从整体的结构来看，整个检测过程实质上就是一棵退化了的决策树，称之为“级联”，如图 5-6 所示。

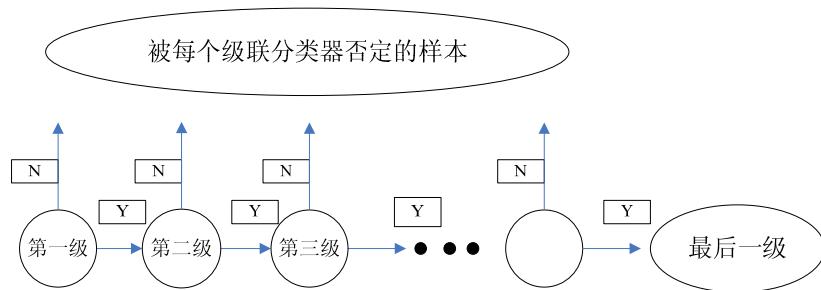


图 5-6 级联分类器的示意图

一些人体检测算法采用级联分类器的形式，因为级联分类器可以先通过前几级排除一些不必要的训练数据特征，它可以正确地检测窗口的同时还可以排除掉大量的无关子窗口，提高检测的性能，降低检测所需要花费的时间。简单的分类器通常被用来排除大量的数据，在这之后利用复杂的分类器来完成降低误正率的错误。

这一系列的分类器将用于所有子窗口的检测。最初的那个分类器可以通过少量的处理来拒绝大量的反例。随后的那些将会继续剔除反例，但相对需要额外的计算。经过几级的处理，子窗口的数量将会明显的减少。从第一个分类器出来的被肯定了的结果将会触发第二个分类器，并通过此分类器来实现更高的检测率。而从第二个分类器出来的被肯定了的结果又会触发第三个分类器，以此类推。相反，从任何一个结点输出的被否定了的结果都会导致对这个子窗口的就不需进入到下一个级联。

### 5.2.3 人体检测在系统中的实现

我们把在 PC 机上训练出来的 MSO 人体模型加载到 DSP 的 DDR2 存储空间中，然后采用级联的 Adaboost 分类器对一块区域的 MSO 编码进行检测。

系统中选定的区域为前一步的动态目标被检测到的区域，系统对非动态的人体不做分析。对选定的区域，先在最大尺度上对动态目标出现的区域进行 MSO 编码，即窗口大小为区域的大小，因为一般帧差法得到的区域大致为目标大小的区域。在确定窗口大小的时候，要保证窗口的宽为 64 的整数倍，长为 128 的同一整数倍，因为本文在 PC 机上进行人体训练的样本大小都是  $64 \times 128$  的，如果不按整数倍进行缩放的话，整个人体的 MSO 编码就会变形，从而无法检测到人体。

对此窗口的 MSO 编码进行检测，如果为人体，则用框标出，否则将缩减此窗口的大小，缩减大小后，系统选定区域中就能包含很多小的窗口，然后对这些小的窗口都进行 MSO 编码，并都通过级联的 Adaboost 分类器进行检测，如果为人体，则标出此区域，否则的话就重复缩减窗口的大小，直到  $64 \times 128$  大小的区域，因为此大小的区域为本文中训练样本的大小，本文假设它是最小的能检测的区域。在每次进行缩小窗口大小时，系统中采用移位寄存器对前一个窗口的长宽往右，平移 1 位，然后在整个动态区域中都所有窗口进行检测。

图 5-7 为全景图象中人体检测的结果，用黑色方框标示出来。



图 5-7 全景图象中人体检测结果

### 5.3 本章小结

本章为在全景视频监控系统的基础上开发出智能视频监控系统，以达到动态人体目标检测的目的。系统中采用了实时性很高的帧差法去进行动态目标检测。在检测到的动态目标区域，系统采用了 MSO 编码和 Adaboost 级联分类器的方法进行了人体目标检测，实验结果表明本文的全景视频监控系统能够达到动态人体目标检测的目的。

## 总结与展望

本文提出了一种基于 TMS320DM648 高性能 DSP 的全景视频监控系统。该系统结合了当前计算机科学及电子工程研究领域的很多热点，包括：高性能 DSP 的应用、多媒体处理以及人体目标检测。这些热点问题都在本文中有所涉及以及解决。

为了实现这种全景视频监控系统的原型，我们使用了美国德州仪器公司生产的多媒体数字信号处理器 TMS320DM648 作为核心，并根据其相应的外设接口进行了硬件设计，从而得到了一个稳定的全景视频处理的平台。

为了实现全景视频监控的目的，我们在自己的硬件平台上实现了基于特征匹配的图像拼接算法。为了使系统更加可靠，我们选用 SIFT 特征点。同时，为了在不失去准确率的情况下提高速度，我们对 SIFT 特征点的提取和描述过程进行了改进。

为了实现智能监控的目的，我们在系统中还实现了动态目标检测，并在有动态目标的区域中进行了人体目标检测。动态目标检测使用帧差法提高了检测效率。对于人体目标检测，我们采用了本实验室提出的基于 MSO 特征的人体检测方法。

大量实验表明我们的系统能够在保证实时性的情况下保证鲁棒性：能提供给用户一个全景视频，并且在有动态人体目标出现的情况下报警。存在的问题为：在进行人体目标检测的时候监视的俯视角不能大于 10 度，否则就需要重新训练不同视角下的样本集，这是我们将来需要改进的地方。

## 参考文献

- [1] 周鹏, 基于 DSP 的视频监控系统. 陕西: 西北工业大学, 2003.
- [2] 王毅轩, 基于 DSP 的视频监控系统研究. 黑龙江: 哈尔滨工程大学, 2008.
- [3] 范金刚, 智能视频监控系统若干关键技术的研究与实现. 四川: 电子科技大学, 2006.
- [4] 韩云锋, 基于 DSP 的视频监控系统终端的设计与实现. 湖北: 华中科技大学, 2007.
- [5] Texas Instruments, TMS320C6000 CPU and Instruction Set Reference Guide, Texas Instruments, Dallas Texas, 2000.
- [6] Texas Instruments, TMS320C6000 DSP Cache User's Guide, Texas Instruments, Dallas Texas, 2002.
- [7] Texas Instruments, TMS320DM647/TMS320DM648 Digital Media Processors, Texas Instruments, Dallas Texas, 2007.
- [8] Texas Instruments, TMS320DM647/DM648 DSP Enhanced DMA (EDMA3) Controller User's Guide (Rev. B) , Texas Instruments, Dallas Texas, 2007.
- [9] Texas Instruments, TMS320DM647/DM648 DSP Inter-Integrated Circuit (I2C) Module User's Guide (Rev. B) , Texas Instruments, Dallas Texas, 2007.
- [10] Texas Instruments, TMS320DM647/DM648 DSP External Memory Interface (EMIF) User's Guide (Rev. B), Texas Instruments, Dallas Texas, 2007.
- [11] 卞红雨, 纪祥春. TMS320C6000 系列 DSP 的 CPU 与外设. 北京: 清华大学出版社, 2007.
- [12] 田黎育, 何佩琨, 朱梦宇. TMS320C6000 系列 DSP 编程工具与指南. 北京: 清华大学出版社为, 2007
- [13] Texas Instruments, TVP5154 Data Sheet, Texas Instruments, Dallas Texas, 2006.
- [14] Philips, SAA7104H: SAA7105H Data Sheet, Philips, 2004.
- [15] S.P. Ierodiaconou, N. Dahnoun and L.-Q. Xu, "Implementation and Optimisation of A Video Object Segmentation Algorithm on an Embedded DSP Platform," In Proceedings of IETCS, 432-437, 2006.
- [16] 宋戈, 基于 DM648 的视频采集系统的设计与实现. 四川: 电子科技大学. 2009
- [17] 孙兴, DSP 在机器人视觉系统的应用于研究. 贵州: 贵州大学. 2006.
- [18] 涂晓昱, 基于 DSP 的通用实时图像处理系统设计与研究. 浙江: 浙江大学. 2004,
- [19] David G. Lowe, "Distinctive image features from scale-invariant keypoints," In International Journal of Computer vision, 60:91-110, 2004.
- [20] Xinying He, Qixiang Ye, Yanmei Liu, Guohong Zhou and Jianbin Jiao, "Image Feature Matching with Network Flow: A Global Optimization Method," In Proceedings of VCIP, Vol.7257 72570F, 2009.

- [21] 何新英, 基于特征点的图像匹配技术研究. 河北: 河北工业大学, 2009.
- [22] 杨艳伟, 基于 SIFT 特征点的图像拼接技术研究. 陕西: 西安电子科技大学, 2009.
- [23] Jingbang Qiu, Tianci Huang, Ikenaga, T., “A FPGA-based Dual-pixel Processing Pipelined Hardware Accelerator for Feature Point Detection Part in SIFT”, In Proceedings of 2009 Fifth International Joint Conference on INC, IMS and IDC, 1668 – 1674, 2009.
- [24] Bonato V., Marques E., Constantinides G.A., “A Parallel Hardware Architecture for Scale and Rotation Invariant Feature Detection”, In IEEE Transactions on Circuits and Systems for Video Technology, 18: 1703 – 1712, 2008.
- [25] Lifan Yao, Hao Feng, Yiqun Zhu, Zhiguo Jiang, Danpei Zhao, Wenquan Feng, “An architecture of optimised SIFT feature detection for an FPGA implementation of an image matcher”, In Proceedings of FPT 2009 International Conference on Field-Programmable Technology, 30-37, 2009.
- [26] Xuelong Hu, Yingcheng Tang, Zhenghua Zhang, “Video object matching based on SIFT algorithm”, In Proceedings of IEEE 2008 International Conference on Neural Networks and Signal Processing, 412-415, 2008
- [27] 李柏林, 基于特征点图像拼接的配准算法研究. 天津: 天津大学. 2008.
- [28] 严磊, 拼接算法研究基于特征匹配的全自动图像. 安徽: 中国科技大学. 2009.
- [29] 李晓娟, 图像拼接技术研究. 陕西: 西安电子科技大学. 2006.
- [30] Krystian Mikolajczyk, Cordelia Schmid, “A performance evaluation of local descriptors”, In Proceedings of computer vision and pattern recognition, 1615-1630. 2003.
- [31] 王蕾. 图像配准技术及应用研究. 西安: 西安电子科技大学, 2007.
- [32] 赵辉. 基于点特征的图像配准算法研究. 山东: 山东大学, 2006.
- [33]. Cucchiara R, Piccardi M, Prati A. Detecting moving objects, ghosts, and shadows in video streams [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(10): 1337-1342.
- [34] 赵秀春, 基于 DSP 的容器检测系统研究. 浙江: 浙江大学, 2005.
- [35] 崔青松, 目标检测跟踪技术研究及 DSP 实现. 山西: 中北大学, 2006.
- [36] 李元军, 基于 VLIW DSP 的图像处理算法与实现研究. 陕西: 西安电子科技大学, 2003.
- [37] 应俊, 基于 DM642 的视频信号处理平台设计. 上海: 上海交通大学, 2005.
- [38] 赵振, 基于 TMS320DM642 的视频处理系统设计的研究. 山东: 山东大学, 2006.
- [39] 范凝凝, 基于机器视觉的管制瓶图像检测. 安徽: 合肥工业大学, 2006.
- [40] 吴晓岚, 基于多帧差高阶统计特征的视频对象提取与 DSP 实现技术. 吉林: 吉林大学, 2006.

- [41] 杨志辉. 基于多尺度方向特征的行人检测算法. 哈尔滨工程大学硕士学位论文, 2009.
- [42] Xu Ran., Zhang Baochang., Ye Qixiang., Human Detection In Images Via L1-Norm Minimization Learning, In: International Conference on Acoustics, Speech, and Signal Processing, 2010.
- [43] Xu Ran., Zhang Baochang., Ye Qixiang., Cascaded L1-norm Minimization Learning (CLML) Classifier for Human Detection, In: IEEE International Conference on Computer Vision and Pattern Recognition, 2010.

## 发表文章目录

### 已发表文章目录

- **Zhenkai Zhang**, Zhihua liu, Jianbin Jiao. “DSP Implementation of a Multi-channel Video Display System with Image Stitching”. 2009 IEEE Youth Conference on Information, Computing and Telecommunications (YC-ICT 2009)

## 致谢

在攻读硕士学位三年的学习生活中，我经历了诸多坎坷，也付出了艰辛的努力，同时也得到了很大的收获。在毕业论文完成之际，由衷地感谢这三年来曾经给予我无数帮助的老师、同学、朋友和家人。

首先，感谢我的导师王文杰教授，感谢王老师三年多来在学习和生活上给予我无微不至的关怀，给我创造宽松的学习环境，同时在科研上给我悉心的指导，从论文的选题方向、总体结构到理论分析等，都提出了许多指导性的建议。不管是在学习还是在生活上，王老师都以自己孜孜不倦的态度和丰富的生活阅历指导着我的人生方向，使我受益匪浅，终生受益。

其次，本课题的研究工作是在导师焦建彬教授的悉心指导下完成的。焦老师在我的学习中，从理论和实验方面给予了大量的、极其有益的建议和具体的指导，并在论文的撰写和审稿中倾注了大量的心血，他诲人不倦的精神和对我的谆谆教导，使我从恩师身上学到了不少可贵的东西，让我受益匪浅。此外，他还在生活和思想上关心我、照顾我，再次表示万分的感谢！尤其是恩师严谨的治学态度、积极的进取精神、平易近人的工作作风将使我终身受益。

还有，特别要感谢叶齐祥老师在三年的学习与生活中对我的极大帮助和支持，他渊博的专业知识和精益求精的动手能力，在理论学习和项目开发过程中给我提供了耐心的引导和帮助。

再次，感谢同实验室的同学们，他们在我的论文工作和撰写中，给予了我巨大的支持和帮助。

我还要衷心感谢我的父母和我的女朋友 Ruina，他们对我工作和学业的热情鼓励和殷切期望，都给我注入了巨大的动力，他们无私的爱和无条件的支持一直指引我奋发拼搏、积极向上，我将对此终生铭记。