

# 用 COTS 多处理机实现红外成像跟踪系统

崔春明 王天冠 周起勃

(中国科学院上海技术物理所, 上海, 200083)

**摘要** 研究了一种基于 COTS 多处理器的实时红外多目标成像跟踪处理系统, 详细描述了在 COTS 多处理器上实现的跟踪处理算法, 同时给出了整个系统的软硬件框架. 这种基于编程的图像处理系统具有高效、灵活的特点, 修改起来非常方便. 该系统自研制成功以来已进行了多次试验, 取得了良好的结果.

**关键词** COTS 多处理机, 图像处理, 多目标跟踪, 并行处理, 软件体系结构.

## IMPLEMENTATION OF THE IR IMAGE TRACK SYSTEM WITH COTS MULTIPROCESSORS

CUI Chun-Ming WANG Tian-Guan ZHOU Qi-Bo

(Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083, China)

**Abstract** This paper describes a real-time IR image track system implemented with COTS multiprocessors. The implementation of a real-time IR image processing algorithm onto the processors is discussed. The architecture of the hardware and software of the system was described, too. The use of programmable image processing of the system provides increased adaptability to higher performance robust algorithms and flexibility for future modification. It is proved by the experiment that the system has good ability of tracking multiple targets.

**Key words** COTS, image processing, multiple-targets tracking, parallel processing, software architecture.

### 引言

过去人们设计了很多种红外成像跟踪系统, 由于技术水平的限制, 硬件功能缺乏灵活性, 因此新的环境下必须重新设计硬件, 然而精确跟踪制导武器使用环境的变化要求不断的研究新型的实时成像跟踪处理系统, 这就要求所设计的系统易于实现、易于移植、便于扩展. 现在大规模集成电路和可编程 DSP 处理芯片应用, 性能优良的算法, 使得现代的图像跟踪处理器在系统结构和图像的跟踪算法上都有了较大的发展. 整个趋势是软件在整个系统中的比例大幅度增加, 使得系统的灵活性, 适应性都相应的得到增加. 目前较为热门的方法是用 COTS (Commercial Off-The-Shelf, 商业产成品) 多处理器产品来实现<sup>[1,2]</sup>.

COTS 多处理器实现主要有以下好处:

易于实现, 对于一个复杂庞大的图像跟踪处理系统, 已经不太可能由一个公司或研究机构来研制. 而 COTS 多处理器已经提供了大部分处理硬件, 系

统的研制时间也大大减少.

具有可变性. COTS 多处理器产品比较成熟, 具有一定的通用性. 对于不同的应用, 其硬件结构不需要作太大的改动.

易于升级. 由于 COTS 多处理器产品具有前后兼容性, 所以只要软件对作少部分改动即可用于新的产品上去.

由于主要算法都由软件实现, 因此更改算法法则非常简便. 并且如编制软件时注意软件体系结构, 使其在一定程度上脱离对硬件体系结构的依赖, 这样对于异类处理板, 其软件移植也非常方便, 编程量也大大减少.

本文将介绍如何在 COTS 多处理器上实现图像跟踪处理器.

### 1 红外搜索与跟踪

红外跟踪处理算法构成如图 1 所示, 主要包括空间滤波、阈值比较、目标提取、跟踪分析.

空域数字滤波的目的是为了提高目标相对于背

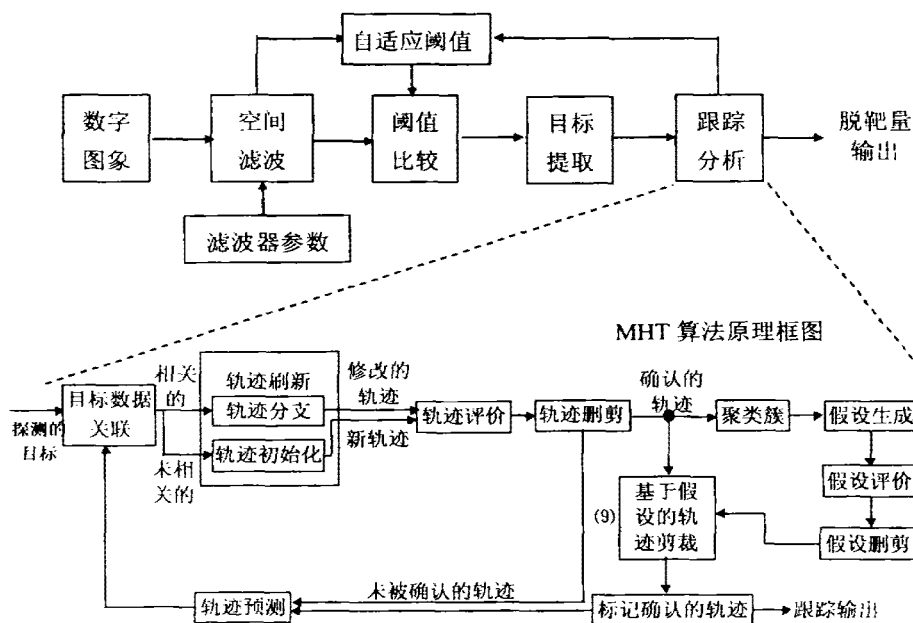


图1 红外目标跟踪处理算法  
Fig.1 IR target tracking algorithm

景和噪声的信噪比,以利于目标的检测.滤波的结构通常根据目标、背景和噪声的特性而设计.不同情况下用于背景杂波抑制的滤波算法也不同.成点象时,相当于在图像背景中迭加一个脉冲,因此要突出图像中的高频部分,而面目标则须突出图像中的低频部分.对于面目标可采用中值滤波,其有良好的抑制冲击噪声能力.对于点目标,采用 $5 \times 5$ 的具有保护带的Robinson滤波器,其突出小目标的能力比其它的滤波器的效果好<sup>[3]</sup>.

目标的热辐射通常高于自然背景,利用阈值比较可将目标有效的提取出来.自适应阈值随着目标和背景的变化而变化,提高了目标抽取的准确性.一种分区图像门限计算方法是它将全视场图像分成 $8 \times 8$ 个子区.实时统计各子区的灰度值的均值 $\mu_{i,j}$ 和方差 $\delta_{i,j}^2$ , $ij = 1, 2, 3$ .则各子区门限为 $T_{i,j} = \alpha\mu_{i,j} + \beta\delta_{i,j}^2$ ,其中 $\alpha, \beta$ 为加权系数.在将图像分区时,为了避免目标在分区的边缘出现不平稳的状况乃至丢失,各分区图像之间有一定的重叠.

目标提取将检测出图像中的潜在目标的位置并把它准确的从背景中提取出来,目标提取还要将属同一目标的点合并到一起,形成目标块,并标以序号<sup>[4]</sup>.每个目标都可表示为一系列所谓的目标特征量.它为以后的跟踪分析作准备.

跟踪分析采用“先探测后跟踪”的算法将空间处理和寻迹处理相结合,既利用目标的二维空间特性,

又利用其第三维时间特性来探测目标,准确地分析确认目标的轨迹,去除虚假目标,并可对目标湮没或部分湮没于背景,多目标间的“碰撞”等情况作出预测和判断,最后将实时跟踪目标.

从理论上讲,MHT(Multiple Hypothesis Tracking Algorithm)算法<sup>[5,6]</sup>是最有前景的跟踪算法之一.如图1所示,其基本原理是当收到每一个测量数据之后,根据这一次的测量与另一次测量或前一帧跟踪之间的联系对测量的不确定性进行数据关联处理.计算这个数据属于不同假设的几率,它可能属于已知目标的新的测量数据;可能是一个新出现的目标的测量数据;也可能这个测量数据只是一个虚警.然后通过Kalman滤波<sup>[6]</sup>对目标状态(通常包括位置、速度、有时包括加速度)和可能的其它信息提供实时的预测.当收到更多的测量数据之后,利用一些其它的已和的或者计算的信息,如未知目标的密度、虚警的密度、探测几率以及目标位置探测的不确定性等条件,来计算更新假设的概率.通过删剪(pruning)那些不可能的假设和合并那些具有相同目标预测的假设,最后得出最有可能的假设,根据假设找到目标,实时跟踪目标.

## 2 COTS 处理器的选取

红外图像处理由于其算法复杂,整个跟踪系统可能需要上亿次的运算能力,因此,现在大多采用专

用的处理芯片,很多厂商都能提供板级的通用 COTS 多处理器产品,处理器之间通过高速交叉网络连接,可供选择的处理器有 ADSP21060、TMS320C6x 等。

由于红外成像跟踪的复杂性,单靠一个处理器不可能完成所要得功能,因此红外成像跟踪系统一般是一个并行的体系结构,处理器的多少可通过式(1)近似计算

$$\frac{de * oe * r * p * op * me}{ppr * pe} = PN, \quad (1)$$

其中  $de$  为重迭率,定义为所处理的图像大小和实际图像的大小之比<sup>[5]</sup>,  $oe$  为向量计算效率,定义为系统的向量乘积输出每点的时间和 1K 大小向量乘积输出每点时间之比,  $r$  为帧频,  $p$  为一帧图像的点数,  $op$  为每点处理所需要的指令数,  $me$  为存储器带宽因子,即访问外部存储器和访问内部存储器时间之比,  $pe$  为实际处理器的效率,处理器一般不能达到峰值来运行,  $ppr$  为处理器峰值速度。

### 3 基于 COTS 多处理器的系统结构

#### 3.1 硬件框架结构

针对红外图像跟踪处理的特点以及所要达到得易于移植、便于扩展的要求,我们设计的硬件框架采用模块化与开放式的主从分布系统结构,如图 2 所示,系统主机(HOST)采用 PC 机,高速处理单元采用基于 DSP 构成的 COTS 数据处理从机(SLAVE),主从机之间通过 PCI 总线交换数据。

图像输入切换接口接受主机控制程序的命令从 3 个红外波段中选择一个波段图像输入,COTS 板完成主要的跟踪处理,构成图像处理器,各种信号预处理,目标提取和跟踪模式算法等软件都将在此板上运行,最后处理得到的目标位置数据将由主机读入并和从图像输入切换接口读入的图像数据合并(主要是在目标上迭加波门),PC 主机是总控部分,

其上运行的软件完成对各 I/O 接口和 DSP 处理板的控制,向用户提供友好的接口,接受用户的命令给图像处理器以及将图像处理结果反馈给用户并显示迭加目标信息后的图像,同时显示被跟踪目标的一些信息,最后由红外系统接口输出脱靶量。

COTS 多处理器采用 CORECO 公司的具有四个 C6201 处理芯片的处理板 Python/C6<sup>[7]</sup>,采用图 3 所示的体系结构,提供高达 6400MIPS 运算能力,处理器之间的通讯有 3 种方式,第 1 种是通过 HEXGATE 高速交叉开关,4 个处理器,输入总线和输出 PCI 总线都通过它来连接,此高速交叉开关可以动态的配置六个数据输入输出端口之间的数据交换,并且传输可在多对端口之间同时进行,第 2 种是全局内存,每个 C6201 处理器即有局部内存便于独自的数据处理,又有全局内存便于数据交换,第 3 种是通过 C6201 处理器的主机接口通过 PCI 总线和主控计算机交换数据,从软件角度看,处理器之间的通讯分为大批量的端口读写和小批量的消息交换两种方式。

对于图像处理系统来说,I/O 吞吐量对系统的处理能力和效率至关重要,Python/C6 上的 C6201 有 4 个 DMA,可以实现两级 I/O 缓冲,即图像数据从总线到 C6201 的相对较慢的 SBSRAM 存储的交换和 SBSRAM 存储器到 C6201 的片内高速 RAM 的交换,这样,数据处理可以在片内高速 RAM 中进行,数据的吞吐量得到了极大的提高,Python/C6 处理板还配有各种 API 函数完成存储器的管理,资源(如 DMA、中断、时钟)的管理。

#### 3.2 软件框架结构

实时图像跟踪跟踪处理器总体框架由 5 部分组成:用户接口(UI)、主机控制部分(HC)、IM 信息管理接口、子图像处理程序部分和 I/O 子系统,总体框架部分之间的关系如图 3 所示。

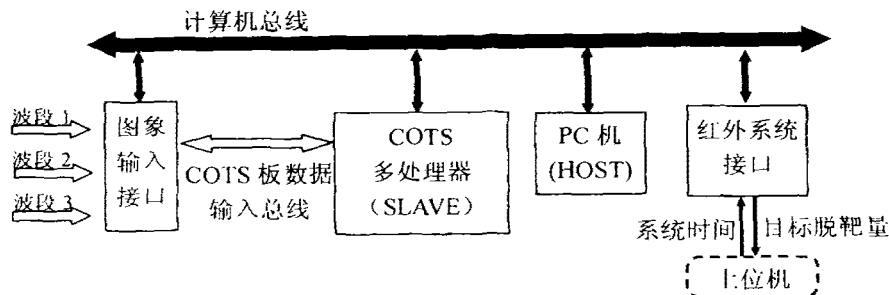


图 2—硬件体系结构

Fig.2 The architecture of hardware

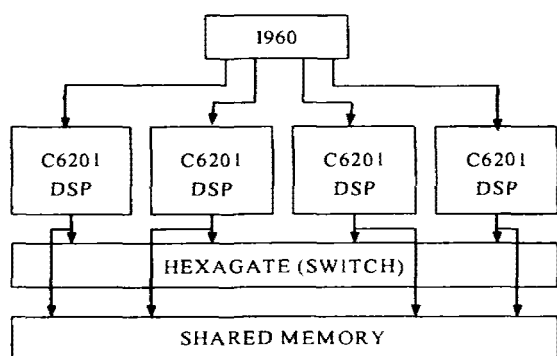


图3 Python/C6 COTS多处理器体系结构  
Fig.3 The architecture of Python/C6 COTS multiprocessors

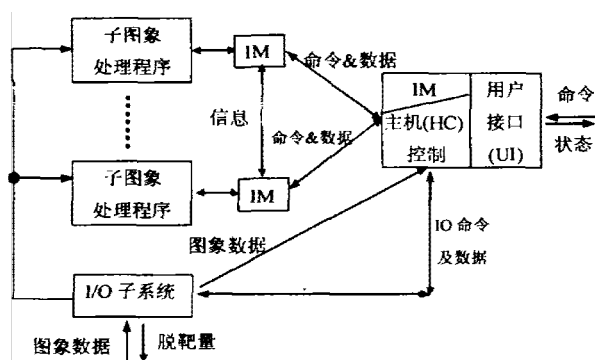


图4 软件体系结构  
Fig.4 The architecture of software

用户接口可以修改包含任何必要的程序或特别的系统接口控制,其主要目的就是接受用户的命令给图像处理器以及将图像处理结果反馈给用户。主机控制部分主要负责分配加载各图像处理子任务给各处理器;综合各子图像处理任务的分析结果;将综合的结果信息迭加到原始图像信息中,送至用户接口。用户接口和主机控制合起来向用户提供方便的操作接口。

图像处理子程序见图1,它们联合起来共同完成图像处理任务。

各图像处理子系统以及主机中有一个很重要的部分即信息管理(IM)接口。其主要的任务是在各图像处理子任务以及主机间传递相关信息和处理结果数据,信息和数据是根据子任务的名称而不是根据子任务所运行的处理器名称而传递的。这么做的好处是将算法和处理器脱离开来,即在COTS处理板中的处理器多少以及处理器的类型变化以后,只要修改IM接口,而整个算法无须改动。整个系统可扩展性、可移植性得到增强。

I/O子系统接受配置信息负责控制和设置I/O子板,传递图像给图像处理子任务。

为了使系统具有可移植和可扩充性,本框架在以下3方面加以考虑。

1) 图像处理子任务的设计应很好的考虑,使得处理器的数目发生变化时,分配图像处理子任务的分配也较容易。随着处理器的增加,图像处理程序不需做大的改动。

2) 采用API库,COTS处理板的图像处理程序,都尽可能地采用其提供的标准接口库函数(API)以屏蔽具体的硬件操作。

3) 由于帧速,帧的大小可能变化,图像处理算法也可能增加。为了达到可移植目的,必须抽象化并行处理器的体系结构,信息管理(IM)部分就是用于此目的而设计的。不管处理器如何增加或在异类处理器之间也能协调完成任务。

#### 4 图像处理算法到COTS多处理器的映射

使用多处理机则需要图像处理子任务在COTS多处理机之间分配,主要有以下几种:

1) 将整幅图像分为几个子窗口,每个处理单元处理一个子窗口,这种方式使用单程序多数据的方式(SPM, Single Program Multiple Data),各个处理单元中的程序一样,数据却不一样。为了防止在子窗口的边缘丢失目标,一般子窗口之间有重迭。随着窗口的增加,窗口的重迭率越大(见表1),处理效率也越来越低。

2) 各处理器之间按照算法流程,采用流水线的方式,如图1中每个步骤的算法都分给一个处理器完成。任务的分配比较简单,但各处理模块之间任务不平衡,容易造成处理器任务的不平衡。

3) 此种方法介于①和②之间,它将图像按帧为单位,将帧按流水线分给各处理器,如:两个处理器,每秒需处理30帧,则先将第一帧分给处理器1,1/30s后将第二帧分给处理器2。再过1/30s后将

表1 320×240探测器的各种分配表

Table 1 Distribution table for 320×240 pixel sensor

窗口大小	处理器数目	子窗口大小	重迭率
320×240	1×1	320×240	1.00
160×120	2×2	190×140	1.39
80×60	4×4	110×80	1.83
64×48	5×5	94×68	2.08
40×40	8×6	70×60	2.63

表 2 三种方法优缺点的比较

Table 2 Comparison between the three processing distribution methods

	优 点	缺 点
第一种	处理器中的程序一样 (SPMD), 分配任务简单, 处理器之间比较独立, 不需要交换太多的数据, 多处理器之间的任务分配也比较均匀。	子窗口与子窗口之间有重迭, 这样再成处理的图像比实际的图像大, 特别是处理器多时, 这种现象越明显。
第二种	处理器之间交流信息也比较固定。	处理器之间的任务很难均衡。
第三种	分配任务简单, 处理器中的程序一样, 多处理器之间的任务分配也比较均衡。	由于帧与帧之间的相关性, 有可能需要前一帧的信息, 这样流水线会延迟等待。

第三帧分给处理器 1, ..., 如此循环下去。这样每个处理器处理帧的时间间隔为  $1/15s$ , 时间延长一倍。以上 3 种方法的优缺点比较见表 2。

红外图像跟踪处理(见图 1)可以看成数据转换的流水线过程, 这就很容易想到用第二种方式, 但要克服其缺点, 需仔细分割图像处理子任务, 均衡分配处理器之间的任务。从软件体系结构角度来说, 这种数据转换的流水线过程是管道-过滤器结构, 结构中的模块和联系分别是过滤器和管道。图 1 中的每个方框都可以看成过滤器。每个过滤器从输入管道接受数据、处理数据并送到输出管道。管道的控制是分布式的, 每一个过滤器都独立运行。过滤器的共享严格限制在管道上。管道可以是共享存储器或通讯信道。

## 5 实验结果

现已按前面所述的软、硬件体系结构实现了多目标成像跟踪器, 它可分别跟踪  $3 \sim 5\mu$  和  $8 \sim 12\mu$  两个波段, 分辨率分别为  $320 \times 240$  元和  $128 \times 128$  元, 帧频分别为 30 帧/s 和 50 帧/s。本系统有点、形心、质心、边缘、相关等多种目标提取方式可选。在形心提取跟踪方式时, 可同时测量 32 个目标, 并具有很强的抗干扰能力。

自跟踪器研制成功以来, 已进行多次实验, 运行一直稳定正常, 软硬件可靠性良好。图 5 是该跟踪器跟踪模拟目标的现场图象。跟踪窗始终套住目标。屏幕右边是目标的特征数据。

本系统模块清晰, 处理都是由软件实现, 它对滤波算法的改变、提取目标的方式的变化, 跟踪算法的改变有很强的适应性, 并可灵活修改。这在本系统流

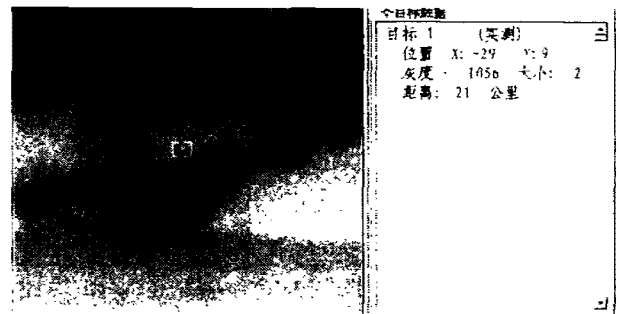


图 5 多目标成像跟踪处理器的跟踪结果

Fig. 5 The experimental result of imaging tracker for multiple-targets

波算法的选取设计中起了很大的作用, 试用了很多滤波算法。

## 6 结语

COTS 多处理器在工程的实践中已经得到了很好的效果。整个系统具有可移植性、可扩展性。本文主要论述了在 COTS 多处理器上实现红外跟踪处理器的技术。该技术都已得到实验论证, 同时其可以用来实现其它基于 COTS 多处理器的图像处理应用。本文所论述的并行技术是相对简单的并行策略, 随着算法的复杂性提高, 算法到多处理器的映像还有待于细化、完善。

## REFERENCES

- [1] Robert Cody, Kevin Ames. Achieving high throughput with COTS multicomputers infrared search and track case study. *SPIE* 1997, **3061**:521—532
- [2] Gillian K Groves, Madhu Chidambara, Robert F Cody. Integrated software architecture for real-time embedded image processing application. *SPIE* 1997, **2086**:198—191.
- [3] David B Reiss. Spatial signal processing for infrared detection. *SPIE* 2235:38—51
- [4] ZHANG Gui-Lin, CHENG Yi-Xin, CAO Wei-Xuan. A connected component labeling algorithm using the run-length code. *Huazhong Univ. of Sci&Tech.* (张桂林, 陈益新, 曹伟焯, 等. 基于跑长码的连通区域标记算法. *华中科技大学学报* 1994, **22**(5):11—14
- [5] Donald B Reid. An algorithm for tracking multiple targets. *IEEE Transaction on Automatic Control* 1979, (24): 843—854.
- [6] Werthmann J R. A step-by-step description of a computationally efficient version of multiple hypothesis tracking. *SPIE* 1992, **1698**:288—300.
- [7] Python/C6 user manual, CORECO company. [http://www.coreco.com/web/home\\_nef/proddsp](http://www.coreco.com/web/home_nef/proddsp)