

布匹疵点检测多机任务并行处理平台的构建

易燕¹, 郑力新², 周凯汀¹, 林似水³

- (1. 华侨大学 信息科学与工程学院, 福建 厦门 361021;
2. 华侨大学 工学院, 福建 泉州 362021;
3. 厦门科华恒盛股份有限公司, 福建 厦门 361008)

摘要: 针对布匹疵点检测算法复杂度高、实时性差等问题,提出一种以多台计算机并行处理为基础的布匹疵点检测平台的构建方案.首先,改进多处理器硬件体系为松耦合方式;然后,根据图像处理算法一般流程,抽象出图像处理模型,并依据模型将处理算法以任务的形式进行分解、封装;最后,通过构建简单实用的用户数据报协议,将任务传输给相应的客户端进行解析处理.结果表明:平台切实可行,检测算法得到显著的加速.

关键词: 布匹疵点; 图像采集; 并行处理; 动态任务; 松耦合; 用户数据报协议

中图分类号: TP 302.1 **文献标志码:** A

目前,纺织企业的布匹疵点检测基本上是由人工完成的,检测速度慢、漏检率高,亟需一种能够快速检测疵点的自动检测系统.织物疵点检测的机器视觉实现是近年来国内外学者在图像领域的热门研究课题之一^[1].龚艳军^[2]提出了相机+数字信号处理器(DSP)+计算机的硬件结构,并通过自适应小波分解的疵点检测算法找出最适合布匹纹理的小波,但该方法实现复杂,实时性得不到保障.文献[3]提出了利用 PC 机构建双缓冲并行系统代替专业的 DSP 开发板,但其采用的双阈值结合形态学的算法过于简单,检测率并不理想.赵振宏等^[4]构建了相机+多 DSP+FPGA 的硬件结构,但对于如何使用多协处理器来提高检测系统的处理性能,并未给出方法.本文构建一种多机并行处理平台,将不同的疵点检测算法运行在平台之上以提高算法的处理性能.

1 多机平台的整体构建与实现

1.1 硬件系统设计

采用多处理器方式可获得较高的处理性能,但传统结构中服务器大部分时间都处于空闲状态,且一般采用紧耦合的方式,即各个处理器的处理任务事先分配好,运行时不能调整.这样可使系统获取一个较好的性能,但各部分容易相互影响,可能致使整个系统崩溃.为克服传统结构的不足,考虑容错机制^[5-6],整个系统采用松耦合方式,即在运行过程中动态分配各个处理器的处理任务,且服务器也参与处理.图 1 为多机平台的硬件系统框图.

1.2 软件架构系统

根据疵点检测图像处理一般过程,抽象出图像处理算法的一般模型,如图 2 所示.图像处理算法由一系列相互递进的过程组成,每个过程又由多个相互独立的处理函数组成.将图像处理算法看成是可以在多台计算机之间并行处理而互不影响、相互独立的任务,即引出多机平台的软件架构(图 3).

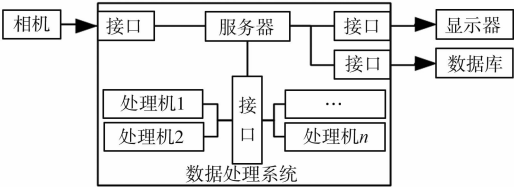


图 1 多机平台硬件系统框图

Fig. 1 Hardware system block diagram of multiprocessor platform

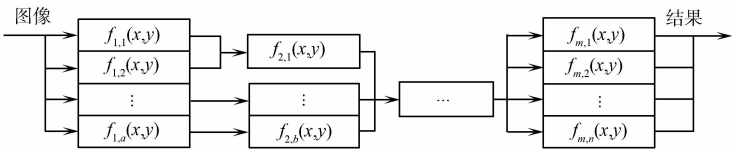


图 2 图像处理算法模型

Fig. 2 Model of image processing algorithms

1.3 多机数据的高速可靠传输

硬件系统是由多台计算机通过千兆以太网组成的小型局域网,具有网络传输速度快、环境简单、干扰因素少的特点. 依据具体情况,数据的可靠传输至关重要而数据包的顺序则无关紧要,因此协议的关键是解决数据发送过程中的丢包问题. 参照 TCP(transmission control protocol)协议的可靠性机制^[7],以及 UDT(UDP-based data transfer protocol)等实现过程,引入了简单的握手机制及数据重发机制,其框架如图 4 所示. 图 4 中:实线表示数据流;虚线表示控制流.

1.4 多机任务管理层的实现

管理层负责任务的分配、跟踪及对其他层的管理,具体实现了对客户端的管理和对任务结构体的定义. 把任务封装成一个一维数组,通过传输协议发送给客户端,而客户端接收到任务结构体后,将任务进行解析,重新分配内存,获得可用的内存指针.

1.4.1 任务结构体定义 根据传输数据特点及传输协议定义了任务结构体(图 5). 其中,客户端索引指处理该任务的客户端的标识码,该字段值由任务分配算法根据客户的处理情况进行指定,实现任务的动态分配. 函数索引指处理该任务所需函数的标识码,通过该字段值可指定任务的处理函数,实现不同任务对应不同的处理算法.

1.4.2 任务封装与解析 任务封装的意义是将任务结构体中对应的数据全部放到一个数组里面,以便对该任务进行发送. 任务解析则相反,是将数组中的数据解释成任务的结构,以便应用程序进行处理. 在任务传递时,如果只是将普通的结构体转换成数组的形式,是很容易实现的,有所不同的是结构体里面包含图像指针. 图像指针所指向的内容里面又包含图像数据指针. 因此需要对任务结构体做专门的处理,将需要的数据封装成一个数组,以便程序进行传递. 在图 5 所示的任务结构体中,图像指针为 OpenCV 里使用的 IplImage 结构. 该结构体中定义了图像的宽、高、位深、通道数等信息,而图像的数据则是以指针的方式给出. 因此,可将任务结构体封装成一个数组.

任务解析是将数组解析成图 5 所示的任务结构体形式,但这里并不能简单地将指针进行赋值. 原因在于两台电脑的内存分配不一样,例如在电脑 A 中,分配的图像结构体指针为 0x0012c288,而在电脑 B 中,该指针所指向的内存有可能已被占用. 因此,在任务的解析时,需要对各个数据空间进行重新申请、分配,然后赋值相应的指针.

2 多机平台的疵点检测算法的实现

2.1 并行处理的实现

将图 2 算法模型实例化,假设其实例化算法任务递进模型如图 6 所示. 图 6 中算法的任务递进分为 4 个过程,每个过程中包含 1 个或多个相互独立的处理函数,将它们定义为任务,需将其分配到客户端进行并行处理.

对于任务分配的客户端,先定义了分配算法函数. 然后,通过调用分配算法函数,得到处理任务的

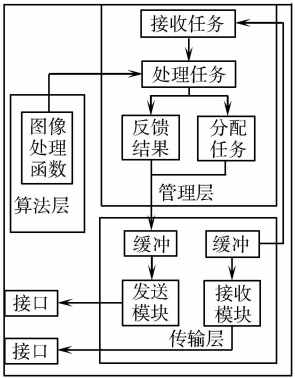


图 3 多机平台软件整体框图

Fig. 3 Software block diagram of multiprocessor platform

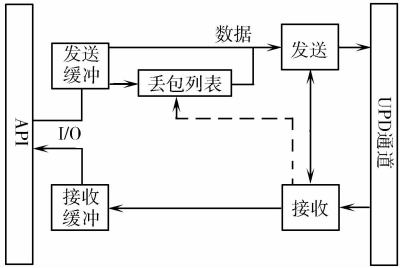


图 4 可靠传输协议

Fig. 4 Reliable transport protocol

0 1 2 3		31
任务ID		
任务类型	任务状态 (F)	
处理任务的客户端索引		
处理任务的函数索引		
图像数量		
图像指针1		
图像指针2		
图像指针3		

图 5 任务结构体

Fig. 5 Structure of task

客户端信息. 最后,按照图 5 所示结构体填充过程对应的任务并标记为 T_{ij} (i 表示第几个过程, j 表示该过程中第几个处理函数), 例如: 过程 1 的处理函数 $f_{1,1}(x,y), f_{1,2}(x,y), f_{1,3}(x,y)$ 对应的任务分别为 T_{11}, T_{12}, T_{13} .

图 6 中算法的递进关系可以从过程和任务两个思路来理解. 过程的思路是,图像进入后,由过程 1 进行处理,等待过程 1 全部处理后再转由过程 2 处理,按此顺序递进处理,最后由过程 4 处理输出结果. 为了直观说明任务分配的情况,其具体分配情况如图 7 所示.

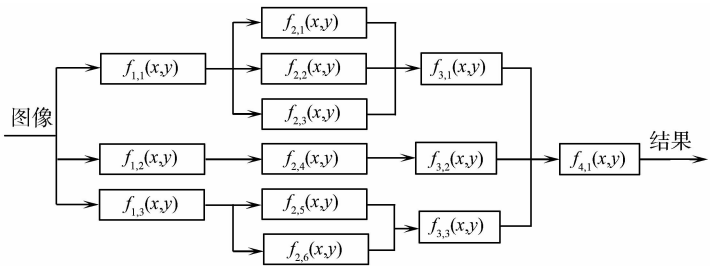


图 6 算法的任务递进
Fig. 6 Steps of the task algorithm

	1	2	3	4	5	6	7	8
客户端1	T11	T13	T21	T23	T25	T31	T33	T41
客户端2	T12	空	T22	T24	T26	T32	空	空

图 7 基本实现的任务分配情况
Fig. 7 Task assignment of basic implementation

通过图 7 可以看出: 如果以“过程”为基元来处理的话, 可能导致某一段客户端的闲置. 通过在客户端实现一个任务列表, 存储 2 个以上的任务来节省任务的分配与接收时间. 在与过程不冲突的前提下, 以“任务”为基元, 根据“谁最闲, 谁处理”的原则, 在分配任务前计算客户端任务的处理量大小(任务的运算量之和), 并将需要分配的任务按照处理量的大小进行排序, 将处理量最大的任务分配给总处理量最小的客户端. 这样不仅克服了任务分配不均的问题, 同时也避免了时间的浪费, 这就是“任务”的思路.

2.2 Gabor 滤波器组的斑点检测算法

Gabor 算法作为一种多尺度空间-空间频域滤波方法, 非常适用于提取多尺度的纹理特性^[8]. 其一般处理过程为预处理、Gabor 滤波、特征提取、图像融合, 然后进行分类等后处理, 最后输出结果. 二维 Gabor 函数空间表达式的一般形式^[9-10]为

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp(-\frac{1}{2}[(\frac{x}{\sigma_x})^2 + (\frac{y}{\sigma_y})^2]) \exp(2\pi jFx). \tag{1}$$

式(1)中: σ_x, σ_y 是尺度因子, 决定函数的空间扩展; F 表示 Gabor 函数的中心频率. 由式(1)变形可得

$$g_{p,q}(x,y) = g(x',y') = \frac{1}{2\pi\sigma^2} \exp(-\frac{1}{2\sigma^2}[(x')^2 + (y')^2]) \exp(2\pi jFx') \tag{2}$$

式(2)中: $x' = x\cos\theta_q + y\sin\theta_q$; $y' = -x\sin\theta_q + y\cos\theta_q$; σ 表示高斯函数的标准差, $\sigma = \sigma_x = \sigma_y$; $\theta_q = \pi \times \frac{q-1}{L}$; p, q 表示尺度膨胀的次数与方向旋转的次数, $p = 1, 2, \dots, S$; $q = 1, 2, \dots, L$.

按式(2)对 Gabor 函数进行尺度膨胀与方向旋转, 可得到一组自相似的 Gabor 滤波器组. 取 $F = \frac{1}{2\sigma}$, 令 $S = 1, L = 2$, 即取 1 个尺度, $0^\circ, 90^\circ$ 两个方向, 然后根据 Gabor 滤波结果求得斑点图像对应的最佳尺度与方向, 再对最佳效果的滤波图像进行融合, 这样可减少冗余图像对结果的干扰. 最后, 经过后期的平滑处理, 得出斑点图像. 斑点检测算法的多机平台实现过程及任务分配情况, 如图 8 所示.

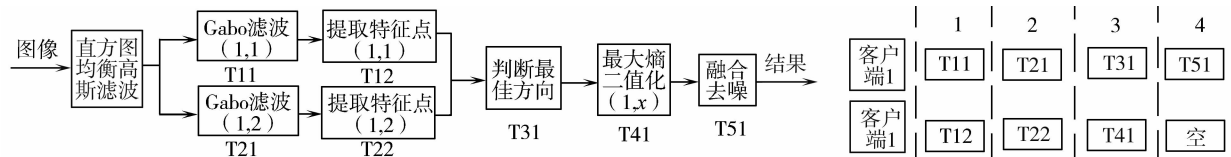


图 8 算法的实现过程及任务分配情况
Fig. 8 Process of algorithm and allocation of task

3 实验结果与分析

测试采用的 PC 机配置均为 Intel(R)Core(TM)2,2.66 GHz,4 G,800 MHz. 将实际拍摄的疵点图像在不同数量 PC 机构建的多机平台下进行实验.

3.1 平台有效性检验

测试图片大小为 512 px×512 px,以断纬疵点图像的检测为例,其处理结果如图 9 所示.



(a) 单台处理结果 (b) 1 服务器+1 客户端处理结果 (c) 1 服务器+2 客户端

图 9 疵点图像处理结果

Fig. 9 Result of processed defect image

分别利用 1 个尺度、2 个方向(0°,90°)和 2 个尺度、4 个方向(0°,45°,90°,135°)的 Gabor 算法,对勾丝、断经、断纬三类疵点多次测试并计算平均时间(t)和加速比(γ),结果如表 1 所示. 其中:加速比 γ 为单台处理机处理所需要的时间与 p 台处理机处理所需的时间的比值.

表 1 三类疵点的处理时间及加速比

Tab. 1 Processing time and speed ratio of three kinds of fabric defect

平台类型	1 个尺度,2 个方向						2 个尺度,4 个方向					
	勾丝		断纬		断经		勾丝		断纬		断经	
	t/s	γ	t/s	γ	t/s	γ	t/s	γ	t/s	γ	t/s	γ
单服务器	0.235	1.00	0.235	1.00	0.234	1.00	0.953	1.00	0.968	1.00	0.969	1.00
1 台服务器+ 1 台客户端	0.140	1.68	0.141	1.67	0.141	1.66	0.546	1.75	0.547	1.77	0.562	1.72
1 台服务器+ 2 台客户端	0.156	1.51	0.157	1.50	0.156	1.50	0.323	2.95	0.341	2.84	0.332	2.92

从表 1 可知:1 个尺度、2 个方向(0°,90°)的 Gabor 算法下,1 台服务器+1 台客户端的加速比均值约为 1.67,较单服务器方式增幅可达到 0.67;对于 1 台服务器+2 台客户端时,加速比不仅没有提高反而降低. 这是由于 Gabor 算法仅有 1 个尺度、2 个方向,按照任务分配准则,在服务器参与处理的情况下,只能将任务分配给 3 台处理器(1 服务器+2 客户端)中的两台,这样其中一台就处于空闲状态,同时服务端要对空闲客户端进行管理,从而导致了上述现象. 此外,从 1 个尺度、2 个方向(0°,90°)和 2 个尺度、4 个方向(0°,45°,90°,135°)的 Gabor 算法计算结果对比可知,算法复杂度越高,处理时间越长,但系统的加速效果越好,为降低漏检率和保证实时性提供了有利的条件.

3.2 快速性对比实验

对于疵点检测可将图片分割为两部分,分别送给两台计算机处理,最后将结果进行拼接. 这种架构用于提高检测速度^[11],这也是许多学者正在研究的一种架构. 通过单机检测算法,分别测试 512 px×512 px,512 px×256 px 两种图片大小的处理时间,并与 1 台服务器+1 台客户端处理 512 px×512 px 大小的图片的处理时间进行对比,结果如表 2 所示.

表 2 图片处理时间的对比

Tab. 2 Contrast of image processing time

平台类型	图片大小	t/s			t_{avg}/s
		勾丝	断纬	断经	
单台 PC 机	512 px×512 px	0.235	0.235	0.234	0.235
单台 PC 机	512 px×256 px	0.156	0.157	0.157	0.157
1 台服务器+ 1 台客户端	512 px×512 px	0.140	0.141	0.141	0.141

从表 2 可知:提出的多机任务架构相对于拼接架构单机处理速度提升了约 10.2%,可见该架构具有一定的优势.

4 结 束 语

通过对已有并行处理结构的研究分析,结合疵点检测算法的一般过程,构建了布匹疵点检测多机并行处理平台.实验表明,平台能够获得较为显著的加速效果.然而,对疵点的分类算法尚未进行研究,对于其他检测算法效果仍有待验证.虽将处理过程分解成任务并动态分配给客户端,但对如何结合“数据分解”的方式,进一步提高多机系统的处理性能,仍需进一步研究.

参考文献:

[1] KUMAR A. Computer-vision-based fabric detection: A survey[J]. IEEE Transactions on Industrial Electronics, 2008,55(1):348-363.

[2] 龚艳军. 布匹疵点自动检测系统的研究和设计[D]. 武汉:华中科技大学,2005:8-12,33-46.

[3] CHO C S,CHUUNG B M,PARK M J. Development of real-time vision-based fabric inspection system[J]. IEEE Transactions on Industrial Electronics,2005,52(4):1073-1079.

[4] 赵振华,郑红. 嵌入式可重构的多 DSP 图像并行处理系统[J]. 单片机与嵌入式系统应用,2009(2):12-15.

[5] 陈筠. 嵌入式多处理器实时系统的容错研究与实现[D]. 成都:电子科技大学,2006:4-17.

[6] 张兵. 多 DSP 并行处理模块的实现及任务分配算法[D]. 上海:上海大学,2007:33-44.

[7] 武奇生. 网络与 TCP/IP 协议[M]. 西安:西安电子科技大学出版社,2010:142-151.

[8] UNSER M. Texture classification and segmentation using wavelet frames[J]. IEEE Transactions on Image Processing,1995,4(11):1549-1560(doi:10.1109/83.469936).

[9] MANJUNATH B S,MA W Y. Texture features for browsing and retrieval of image data[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,1996,18(8):837-842(doi:10.1109/34.531803).

[10] BODNAROVA A,BENNAMOUN M,LATHAM S J. Textile flaw detection using optimal Gabor filters[C]//Proceedings 15th International Conference on Pattern Recognition, Barcelona; IEEE Press, 2000: 799 - 802 (doi: 10.1109/ICPR. 2000. 903038).

[11] HSU S,SAWHNEY H S,KUMAR R. Automated mosaics via topology inference[J]. IEEE Computer Graphics and Application,2002,22(2):44-54(doi:10.1109/38.988746).

Construction of a Parallel Processing Platform for
Fabric Defects Detection Based on Multiprocessor Task

YAN Yi¹, ZHENG Li-xin², ZHOU Kai-ting¹, LIN Si-shui³

- (1. College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China;
2. College of Engineering, Huaqiao University, Quanzhou 361021, China;
3. Xiamen Kehua Hengsheng Company Limited, Xiamen 361008, China)

Abstract: According to the problems of high complexity and poor real-time, an approach is proposed to design the platform for fabric defects detection based on multiprocessor parallel processing technology. Firstly, a method is proposed to improve the multiprocessor hardware system as loose coupling. Secondly, a method is proposed to abstract the model of image processing based on the general process of image processing algorithms, and then using the form of task to disintegrate and package the processing algorithm based on the model. Lastly, a method is proposed to transform the task to the corresponding clients to analyze and handle by building a simple and practical user datagram protocol. The experimental results show that the platform is feasible and the detection algorithm has been significantly accelerated.

Keywords: defects detection; image capture; parallel processing; dynamic task; loose coupling; user datagram protocol