Springer

# Parallel Image Processing

# 并行图像处理

〔德〕托马斯·布劳恩
斯特凡·法伊尔　沃尔夫冈·拉斐　米凯尔·赖因哈特 著
李俊山　李新社　焦　康 译

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

# Parallel Image Processing
# 并行图像处理

T. Bräunl
with S. Feyrer • W. Rapf • M. Reinhardt

〔德〕托马斯•布劳恩
斯特凡•法伊尔　沃尔夫冈•拉斐　米凯尔•赖因哈特　著

李俊山　李新社　焦　康　译

西安交通大学出版社
Xian Jiaotong University Press

## 内 容 简 介

本书介绍并行图像处理的原理、技术和方法，以及在数据并行系统中的实现技术和算法。全书共有 13 章，分别为：绪论、点运算、局部运算、边缘检测、骨架化、形态学算子、图像分割、轮廓线提取、Hough 变换、傅里叶变换、纹理识别、立体图像处理、图像序列分析。内容深入浅出，并有大量的程序实例。书末提供的附录，详细地介绍了本书在算法描述中所采用的并行编程语言 Parallaxis-Ⅲ及其句法、并行处理模型系统和景像库等。

本书可作为计算机、电子信息工程、模式识别与图像处理等专业本科生或研究生的教材或参考书，也可供从事计算机、电子信息工程等相关学科的教学、科研和工程技术人员参考。



**图书在版编目(CIP)数据**

并行图像处理/（德）布劳恩（Bräunl, T.）等著；
李俊山，李新社，焦康译．—西安交通大学出版社，2003.11
书名原文：Parallel Image Processing
ISBN 7-5605-1769-2

Ⅰ.并... Ⅱ.①布... ②李... ③李... ④焦...
Ⅲ.图像处理：并行处理 Ⅳ.TP391.41

中国版本图书馆 CIP 数据核字(2003) 第 096522 号

陕版出图字：25-2003-007 号

| | |
|---|---|
| 书　　名 | 并行图像处理 |
| 著　　者 | 〔德〕布劳恩等 |
| 译　　者 | 李俊山　李新社　焦 康 |
| 出版发行 | 西安交通大学出版社 |
| 地　　址 | 西安市兴庆南路 25 号(邮编：710049) |
| 电　　话 | (029)2668357　2667874(发行部)<br>(029)2668315　2669096(总编办) |
| 印　　刷 | 陕西宝石兰印务有限责任公司 |
| 字　　数 | 236 千字 |
| 开　　本 | 727 mm×960 mm　1/16 |
| 印　　张 | 13.125 |
| 版　　次 | 2003 年 11 月第 1 版　2003 年 11 月第 1 次印刷 |
| 印　　数 | 000 1～3 000 |
| 书　　号 | ISBN 7-5605-1769-2/TP·358 |
| 定　　价 | 19.50 元 |

# 译者序

随着科学技术的迅猛发展，图像处理技术已经成为近代信息处理领域中的一项非常重要的技术，并已在计算机视觉、航空摄影测量、地球资源勘探、气象气候预测、生物医学工程、目标识别与跟踪、飞行器导航、导弹武器精确制导等领域得到了广泛的应用，成为信息化社会和国防建设中的重要支撑技术，具有十分广阔的发展和应用前景。

由于大多数图像处理问题是计算密集型的，所以许多串行处理方法在应用中遇到了难以实时计算的困难。随着并行计算机及其并行处理技术的发展，并行图像处理技术应运而生，而且已成为图像技术领域和计算机科学领域中的一个重要学科方向。

并行图像处理技术，就是根据图像处理技术和并行处理技术的基本原理，研究各类串行图像处理方法的并行实现原理、技术和方法，以及在不同互连结构（如传统环网、折叠环网、超立方体等）的并行计算机（如 SIMD-单指令流多数据流，MIMD-多指令流多数据流）上的并行处理与并行实现。

本书是目前国际上最新出版且十分罕见的并行图像处理专著，主要研究各种串行图像处理方法在 SIMD 计算机上的并行处理技术及并行算法。书中内容是对作者及当前国际并行图像处理领域研究成果的总结。书中首先介绍了并行图像处理与并行数据系统；接着介绍了各种并行图像处理方法，包括：点运算、局部运算、边缘检测、骨架化、形态学算子、图像分割、轮廓线提取、Hough 变换、傅里叶变换、纹理识别、立体图像处理、图像序列分析等，涉及图像处理研究中的绝大多数内容；最后在附录部分较详细地介绍了本书在算法描述中所采用的并行编程语言 Parallaxis-Ⅲ及其句法、并行处理模型系统和景像库等。书中不仅系统地介绍了各种图像处理方法的并行处理方法，而且给出了大量的并行处理算法。特别是作者还通过互联网提供了可以免费下载

的 Parallaxis-Ⅲ软件包、语言描述、文档资料、程序实例和并行图像处理软件库。其内容对于促进并行图像处理技术的发展和嵌入式并行图像处理和理解技术及其系统的研究具有十分重要的意义。

译者近几年来在从事并行图像处理技术的研究中，深切地感到目前有关并行图像处理方面的文献偏少，特别是如此专业性地全面介绍各类并行图像处理技术和并行算法，并有大量并行图像处理程序实例和软件的专著仅见此一部。所以立即进行了翻译，希望能对并行图像处理技术在我国的研究与应用有所裨益。

本书在翻译过程中，参阅了国内有关图像处理专著中对有些重要专业术语的译法。文中改正了原著的某些笔误和印刷错误。这些错误基本上都是非实质性的，所以没有逐一注明。译者虽力求翻译准确，但水平有限，误译之处在所难免，敬请读者不吝赐教。

本书的翻译得到了沈绪榜院士的关心和指导。李俊山教授的研究生胡双演、曹素平和封富君参与了部分译稿的整理工作。在此对他(她)们表示真诚地感谢。特别感谢西安交通大学出版社国际合作部赵丽萍主任为本书出版的支持和关心，感谢邹林和贺峰涛编辑对本书付出的辛劳。

译　者

2003 年元月于西安

# 序 言

本书是从图像处理领域介绍大规模并行算法的一系列出版物中发展而来的。图像处理学科在按照 SIMD(单指令流多数据流)原理工作的大规模同步并行或数据并行计算机系统中的应用是一个特别有前景的领域。当大型 SIMD 超级计算机的时代结束时,SIMD 系统作为专用的视觉子系统再度流行,并很快出现在嵌入式系统中。

与基本图像运算的传统顺序执行方式相比较,本书阐述了经常在图像处理中出现的内在并行性。通过并行算法,用一种比顺序运算更简单易懂的方式说明图像运算是可能的。

本书所选择的表示方法认为,程序代码的简明扼要的引用非常有益于加强对内容的理解,比如图像运算,而过长的程序清单很可能分散对主题的注意力。由于这个原因,每一章不但借助于例子解释和定义了主要的图像处理算法,而且还提供一个大规模并行程序的选录。对于图像处理,这意味着每个像素事实上至少有一个处理器可以利用。显然,到少量实际处理器的映射是由编译器实现的,而诸如此类的问题在这里无需关心。

本书所提供的算法都用数据并行程序语言 Parallaxis 书写。附录中给出了 Parallaxis-Ⅲ的介绍及其完整的语言定义。Parallaxis 中的程序可以移植到其他任何数据并行系统,也可在工作站或个人计算机上进行模拟和调试。完整的 Parallaxis 软件包,即程序设计环境、语言描述、文档资料、程序实例和并行图像处理库是不受版权限制的软件,可以在因特网上免费下载,网址是:

http://www.ee.uwa.edu.au/~braunl/parallaxis

Parallaxis-Ⅲ系统是由爱德华·卡普尔(Eduard Kappel)、哈特穆特·凯勒(Hartmut Keller)、哈拉尔德·兰普克(Harald Lampke)、约尔格·施蒂帕(Jørg Stippa)和于尔根·瓦肯达(Jürgen Wakunda)在托

马斯·布劳恩(Thomas Bräunl)的负责下完成的。我们衷心地感谢为Parallaxis-Ⅲ的实现做出过贡献的所有学生和将德文稿翻译成英文稿的米凯尔·尤西克(Michael Juschke)先生。

本书第1～4章、第6～7章、第12章和附录是由托马斯·布劳恩(Thomas Bräunl)撰写的;第5章关于骨架化的内容是由米凯尔·赖因哈特(Michael Reinhardt)撰写的;第8章和第13章关于角检测和图像序列的内容是由沃尔夫冈·拉斐(Wolfgang Rapf)撰写的;第9～11章有关变换和纹理结构的内容是由斯特凡·法伊尔(Stefan Feyrer)撰写的。

托马斯·布劳恩
斯特凡·法伊尔
沃尔夫冈·拉斐
米凯尔·赖因哈特
2000年11月　于珀思

# 目　录

# 1 绪论

## 1.1 图像处理

数字图像处理分为两个层次:低层处理和高层处理。低层处理也称为图像预处理;高层处理则称为图像识别、图像理解或计算机视觉,属于人工智能(AI)领域。图像预处理和图像识别之间的界限是不明显的,然而,组合的数据结构可作为一个标准对它们进行区分。低层图像处理通常或者把图像数据转换成图像数据,如对比度增强、噪声缩减,或者计算输入图像的简单特征,如轮廓、直方图和变换等。在低层处理中,图像内容的理解是不相关的。仅仅在图像识别中,才试图将预处理图像的数据解释为目标识别。

因为在所有层次的图像处理中都有大量的数据要处理,所以对计算能力和处理时间要求很高。一幅具有 1024×1024 个像素且每种颜色(即红、绿、蓝)用 8 位颜色分辨率表示的彩色图像需要 3MB 存储空间。然而,特别当很多处理器(processor)能够有效利用时,图像数据的并行处理能够大大减少处理时间,对于低层图像处理更是如此。例如,当处理是基于像素级时。理想的情况是每一个像素与某一个处理器相对应。简单的图像运算一般是高度重复且仅局部相关。例如,滤波运算常常对所有像素进行相同运算,而且处理的顺序是任意的。这样的处理实际上可以通过对每个像素执行相同的指令并行实现,即通过每个处理器执行相同的指令并行实现,这称为同步并行。由于每一个元素运算在每个处理器上并行执行,不需要对处理器间的任务进行同步,所以称这种处理为同步并行。

本书只限于描述图像处理的低层处理方法。这种处理方法可以用同步并行算法简单而有效的实现。这意味着从现在开始像素可以被看作有源处理元素(active processing elements),管理它们的像素值的 PEs 可以与相邻单元交换数据,最终重新计算它们的像素值。这是一个局部图像运算的典型示例。

在后面的各章中介绍的方法是按照它们的复杂程度分类的。局部图像运算一般只需从有限的邻近范围内获得图像数据,而像素运算完全不需要其他像素的数

据。在这种情况下，每个单独的像素的计算要求不高。对于全局图像运算，因为每个单独的像素的值与其他所有像素相关，所以对计算能力要求很高。这种相关可以基于简单的统计分析，例如直方图，或者可由一个综合的变换来定义。

在从原始图像数据到目标识别的过程中常常伴随着数据缩减。例如，当惟一要做的事情是从一个 3MB 的图像中得出结果“OK”或“Fault”时，数据减少就非常明显。因此，对于图像识别来说，大量的图像运算可归为预处理(pre-processors)，它们可以使图像的数据输入量显著减少。例如，边缘检测能够减少输入图像一到两个数量级，只有那些相对于相邻像素有明显的色彩或灰度差的像素才被保留。另一种减少一个数量级的方法是通过把边缘变细来获得的，也就是说只有边缘中心线上的像素被保留。还有一种减少一个或两个数量级的方法出现在直线识别(把特定的边分成直线段)和特征检测时出现。对于角检测，只有那些相邻边缘像素呈现出明显弯曲的像素才保留着。

立体图像分析或图像序列分析需要很多更复杂的算法和启发式方法的运用。那些最适合于某个任务的滤波值和参数通常只能通过实验的方法(反复实验)来确定。非常有趣的是，这里介绍的低层处理方法只在灰度有变化处(灰度梯度)起作用，而高层处理方法以特征分类为基础。

关于数字图像处理的介绍的和最新的研究文献可参考[Ballard, Brown 1982]，[Gonzalez, Woods 1992]，[Jähne 1997]，[Jain 1989]，[Bässmann, Besslich 1993]和[Nalwa 1993]。

## 1.2 并行处理

我们把并行图像处理分成三种模型：流水线处理、异步并行处理和同步或数据并行处理。异步并行处理计算机称为 MIMD(multiple instruction, multiple date——多指令流，多数据流系统)，而同步的并行计算机被认为是 SIMD(single instruction, multiple data——单指令流，多数据流)系统或数据并行系统。在异步并行系统中，每个处理器有它自己的控制流，执行它自己的程序。在数据并行系统中，所有处理器或 PEs(processing elements——处理元)都从中央控制处理机接收它们的命令，所有处理器在相同时间对它们的可能是不同的局部数据执行相同的命令，或处于停止状态(矢量或数据并行处理)。因此，在数据并行系统中，只有一个连续的控制流而没有独立的异步处理，这就使编程简化了许多。因为所有 PEs 是同步运行的，也就是说所有的 PEs 在每一步或多或少地被同步，所以对于像信号监视器或消息传送，无需进行大量的、易出错的同步处理。

此外，一个 SIMD 系统的 PEs 比一台 MIMD 计算机的处理器设计要简单。

MIMD 计算机的每个处理器需要完整的命令-译码逻辑电路，这就意味着一个 SIMD 系统的处理元(SIMD-PEs)比 MIMD 处理器占用的芯片空间少得多，而且能以非常高的集成度集成。同样地，设计 SIMD 系统可能要用大量的(低档)处理器，比 MIMD 系统要用的处理器多得多。

大规模并行性的表示与用于一台并行计算机中的处理器的数量有关，我们把一个大规模并行系统理解为一个具有 1 000 或更多处理器的并行系统。目前这样的集成化程度仅能在数据并行系统中实现。与经典的异步并行处理相比，这样的大规模并行性需要新的程序设计技术和不同的算法。

大规模并行性在图像处理中具有特别的优势——一个处理器可被每个像素所用。如果像素的数目超过了实际现有的处理器数目，就可以用虚拟处理元。虚拟处理元的概念与虚拟存储器的概念类似。这种新观点开辟了许多新的可能性，并在许多情况下简化了图像处理算法。一个(局部的)图像运算可以通过每个处理元的并行计算而实现，且每一个处理元独立于其他处理元，所需的数据来自于它自身的像素值和与它相邻的那些像素(也即相邻 PEs)的值。所有 PEs 执行相同的运算，由程序员直接使各处理器同步是不必要的。这种方法显然比用异步并行计算机系统的类似方法容易。在异步并行计算机系统中，图像必须被分成几个块(sections，或称为数据片 tiles)由各个单独的异步处理器处理，接着每一个处理器对由它处理的那个数据片中的每一个像素按序循环处理。然而，问题出现在所处理的每个数据片的边缘处，相邻数据片的图像数据要求存储在局部邻域内不同的处理器上。为了解决这个问题，或者必须适当地重叠相邻的数据片，或者必须通过处理器之间的数据交换来实现复杂的同步过程。当所有被处理的图像块必须重新组合成一个新图像时，在图像运算的最后阶段显然需要这样的同步过程。异步并行系统数据交换比算术运算(如加法运算)费时数万倍，因此，应当在程序中尽可能少使用。在数据并行系统中情况就不同了，一次局部数据交换运算(在一个快速网格网络上)与一个单个的算术运算需求的时间相同，因此这样的运算可以频繁使用。

应用大规模并行系统是图像处理的理想方式。下面各节将详细叙述大规模并行系统的一些基本原理。并行图像处理领域的介绍可参考[Bräunl，1993]。

## 1.3 数据并行系统

数据并行系统与同步并行模型相对应，如图 1.1 所示。中央控制器是一个连接有外部设备的标准的顺序型计算机(SISD：single instruction，single data——单指令流，单数据流)。PEs 不执行它自己的程序，而是接受来自控制器的指令。由于 PEs 自身不包含指令译码器，所以它们不是完整的处理器。它们依靠带有局部

存储器和通信通道的 ALUs(算术逻辑单元)。数据并行模型的限制带来了这种简化。在任何时候,PEs 只能执行相同的命令,它们要么根据由控制器发来的命令处理它们的局部数据,要么处于暂停状态。因此,每一个并行选择(If-语句,If-statement)必须分成两步,首先是在所有选择条件为真的 PEs 上执行 then 部分(then-part),而其他的 PEs 处于停止状态。接着是在前一步一直处于被动的 PEs 组执行 else 子句(else-clause),而第一步工作的 PEs 处于停止状态。当然,选择语句的这种顺序执行效率是非常低的。因为一个数据并行系统的简单 PEs 比一台异步并行计算机的集成度更高,所以如果应用适当,大量的 PEs(大规模并行)将使这种低效率得到补偿。

PEs 通过网络互相连接,这种连接或者可以被固化在硬件中,或者可以重组,而且允许在 PEs 组中进行快速数据交换。数据交换不会在两个单独的 PEs 之间发生。因此,必须对它们进行同步,而不是让大量的数据交换发生在所有 PEs 之间或发生在一个 PEs 组内。虽然通信可能是异步并行系统的瓶颈,但它在数据并行系统中是高度并行的。数据也可以在控制器和某个单独的 PEs(选择的)之间进行交换,或在所有 PEs 之间进行交换。

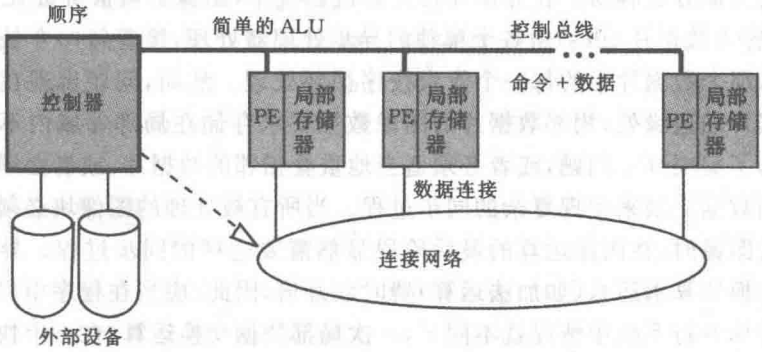


图 1.1 数据并行系统模型

虽然带宽对于 PEs 之间的并行数据交换来说通常是足够的(一次交换的时间=一次算术运算时间),但在同时参与的多个 PEs 和控制器之间的通信却可能出现瓶颈,因为同时参与的多个 PEs 必须顺序与控制器进行通信。由于图像数据总是经过控制器读入(例如从数据单元或硬盘驱动器),然后再分配到 PEs 中,所以控制器和 PEs 之间的总线应该非常快。使用不同的分配策略在某些情况下也是有帮助的。例如,不用控制器对所有需要的 PEs 寻址,就可以把完整的一行图像像素发送到第一列的 PEs,然后利用快速并行网络把已发送到第一列的各 PEs 的数据传送到该 PEs 所在的行中的 PEs。虽然由控制器传送的数据的数量是相同

的，但需要由控制器直接寻址的 PEs 的数量却非常少。

虽然大规模并行计算机含有大量的 PEs，但仍不能满足需求的情况还可能出现。例如，当要处理一幅具有 1024×1024 个像素的图像时，每一个像素应当由一个 PEs 进行处理。在这种情况下，就需要虚拟的 PEs。虽然虚拟 PEs 到实际存在的 PEs 的映射是由应用程序员来实现的，但这种频繁使用的功能最好由程序设计环境实现，或者最好由并行硬件自身完成。

当程序需要的 PEs 数目超过实际的 PEs 数目时，应用程序员就应该采用虚拟 PEs。虚拟 PEs 构成一个抽象级，并可反复执行，系统然后由硬件或软件把虚拟处理器映射到实际的 PEs。同样地，虚拟处理器的概念类似于虚拟存储器的概念。然而，如果应用程序需要的虚拟 PEs 比实际有的 PEs 少，那些不用的实际 PEs 就处于暂停状态。由于数据并行模型的限制，那些不用的处于暂停状态的实际 PEs 不能再用来做其他运算。

具有大量处理元的数据并行系统需要与几十年来程序员已经习惯的冯·诺依曼模型有不同的新的思维方式。下面将说明，许多问题，特别是图像处理方面的问题用数据并行程序语言描述比用顺序语言描述要容易和简单得多。

## 1.4 通信和矢量缩减

如前所述，数据并行交换或者涉及所有 PEs，或者涉及一部分 PEs。因为同步连接可以达到更快的速度，并且这种连接网络通常具有较高的连通程度和带宽，所以数据并行系统中的数据交换要比异步并行系统中的数据交换容易且代价小。图 1.2 表示了一个数据交换的例子，其中每个 PE 把它的局部变量 $x$ 的值平移到与它相邻的右边的 PE。

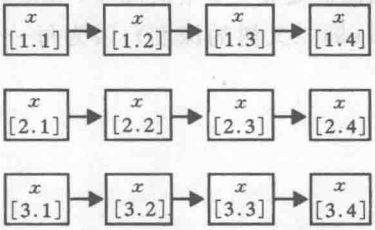


图 1.2 “数据并行”数据交换

许多数据并行系统含有非常快速的网格网络，有时相对慢些，但具有更常用的连接结构。这种网格网络结构主要能把简单常规的结构映射成实际的连接。然