

连通域标记并行算法在多核处理器上的设计和实现

张 健 徐茂兴 (浙江工业大学 计算机学院 浙江 杭州 310023)

摘 要: 为满足全方位视觉运动目标检测跟踪系统中检测和跟踪实时性的要求,采用多核程序设计和并行处理技术,对系统中连通域标记算法,重新进行了设计和实现。为解决由图像分割造成的连通域的分离,应用边界处的游程码,将两个不同任务块中分割处的连通域合并成一个连通域。最后通过实验证明该并行算法比传统算法在多核处理器上运行更快,更好地解决了多核处理器饥饿问题,让多核处理器发挥了更高的效率。

关键词: 连通域标记;多核处理器;图像分割;并行;游程码

Design and Implementation of Connected Component Labeling Parallel Algorithm with Multi-Core Processor

ZHANG Jian, XU Mao-Xing

(College of Computer, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: To meet the real-time requirement of moving objects detection and tracing based on omnidirectional vision, multi-core programming and parallel processing technology are applied to the redesign and realization of the connected component labeling algorithm. The paper introduces run-length encoding into algorithm to solve separation of connected component by image segmentation. Two segmented connected components for different tasks' block are merged into one connected component. Experiments show that the parallel algorithm runs faster than the traditional algorithm in multi-core processor. It has better solved the problem of hunger in multi-core processors and made the multi-core processor more efficient.

Keywords: connected component labeling; multi-core processor; image segmentation; parallel; run-length encoding

全方位视觉运动目标的检测与跟踪中的一个关键技术是对全方位图像的实时检测,因此,在全方位视觉运动目标检测跟踪系统中,通过对全方位图像的实时采集和处理,得到二值化的前景图像,再用连通域标记算法分割前景图像;最后采用形态学算子加以过滤,得到运动目标;最后对该目标进行跟踪。显然检测得到的运动目标的准确性,直接影响到对该目标的跟踪。连通域标记算法是目标检测部分的核心。

1 引言

目前全方位视觉系统的研究都是基于单核处理器

平台的,为了满足不断提高的实时性要求,现阶段的单核处理器平台已经无法满足未来全方位视觉检测和跟踪系统对计算能力的需求。因此,对目标检测和跟踪系统进行硬件加速以提高计算效率已成为了一个重要的研究方向。

多核处理器的出现为目标检测和跟踪提供了很好的平台。传统的程序是按照单线程的方法来编程的,不能享受和利用多处理器带来的性能的提升。免费大餐不久就将结束^[1],多核将引领软件开发将发生基础性变化,特别对接下来几年里那些面向一般应用、运行在PC和低端服务器上的应用软件将不得不通过多

基金项目:浙江省科技厅计划基金(2004C31103)

收稿时间:2009-07-20;收到修改稿时间:2009-09-19

核并行编程的思想来重新设计和利用。

基于这些研究现状,本文对在多核处理器上如何加速连通域标记算法并提高各个核的利用率进行了探索和研究,设计并实现了基于多核处理器的连通域标记并行算法。

2 多核并行计算的发展趋势

2.1 单核 CPU 发展的限制

目前,单核 CPU 的速度已超过 3GHz,届时与现处理器相当面积上集成的晶体管数目将超过 100 亿^[2,3],提高主频会增加处理器的功耗和设计的复杂度,其中最大的困难就是提高主频所带来的高发热问题;高发热问题会导致芯片运行的不稳定,因此目前主频的提升空间已不大。在单核时代,提高性能的另一手段是用 superscalar(超标量)处理器的方式,让处理器在一个时钟周期内执行多条指令。单纯提高单核 CPU 速度,还面临着存储器访问速度匹配的问题。

因此,通过提高单核 CPU 处理能力的方法来提升计算机性能,已经到达瓶颈状态。

2.2 多核并行计算将成为发展趋势

多核并行计算是下一个重大技术进展。利用并行计算技术,计算机可以将不同的任务分配给多个处理器同时处理,而不是由一个处理器每次处理一项任务,从而提升了计算机的运算速度。对多核并行计算的研究将在未来 5 年内迅速发展,这些研究将专注于并行编程技术,包括能够利用多核处理器的各种应用和操作系统^[4]。更多的处理核有助芯片性能的提高,然而软件却在其中扮演关键角色。

总之,多核计算将成为未来的发展趋势,多核编程将成为程序员必须掌握的技术。

3 全方位视觉检测和跟踪系统



图 1 全方位视觉设备

全方位视觉系统由一个 CCD 摄像机和正对着摄像机的反光镜组成,反光镜负责将水平一周的图像反

射给摄像机成像,如图 1 所示。该系统所成的图像称为全方位图像(Omnidirectional Image),也称为基于反射镜面的图像(Catadioptric Image)。这样的一幅图像中包含了水平方向 360°的环境信息。

系统在采集到实时的图像后,并由计算机构建的全方位视觉系统进行图像的处理,得到实时的运动目标信息,对该目标进行跟踪。

3.1 目标检测的方法

实现目标检测,得到运动目标的关键在于通过连通域标记算法分离出运动目标的信息。

由于全方位视觉图像包含的信息量比普通的摄像头的信息量多的多;另一方面目标物体在全方位视觉图像中会扭曲。因此为了获得精确的运动目标,就不得不增加算法的复杂度和数据的计算量。目标检测方法步骤如下:

- 1) 用统计方法建立图像的背景模型;
- 2) 通过减背景法得到前景图像;
- 3) 用连通域标记算法分割前景图像,来获取初步的运动目标信息;
- 4) 最后采用形态学算子加以过滤;
- 5) 得到分离出来的运动目标。

4 基于多核处理器的连通域标记并行算法

多核并行计算已成为当前研究的热点,本文针对全方位视觉在检测和跟踪系统中遇到的问题,利用多核程序设计和并行处理技术,对连通域标记串行算法重新进行了设计和实现。下面给出了连通域标记并行算法在多核处理器上的实现和实验结果。

4.1 连通域标记串行算法描述

在二维图像中,假设目标像素点周围有 m 个($m \geq 8$)相邻的像素,如果该像素灰度与这 m 个像素中某一个像素 A 的相等,那么称该像素与 A 具有连通性。算法首先对二值图像中所有的像素点进行一次完整的扫描,标记所有的目标像素点同时得到等价标记表。等价标记表是记录扫描过程中发现的所有等价对^[5]。等价对的产生是由于扫描次序导致开始时认为是两个不同的连通域,随着扫描的深入发现原本不同的两个连通域属于同一个连通域,把这两个连通域标记号记录在等价标记表中,表明这两个标记号标记的连通域属于同一个连通域。

等价对处理采用递归思想,每递归一次,将属于同一个连通域的所有等价对从等价标记表中找出来,重新归入一类,并把处理后的等价对从等价标记表中删除,直到等价标记表中等价对的个数为零为止。最

后完成运动目标区域的分割,得到相应的连通域个数。

在单核系统上实现的连通域标记串行算法简述如下:

1) 逐行对二值图像中所有的像素点进行连通区域扫描。每一个目标像素点只能根据已经判断连通性的像素点来确定自己的连通性,本文采用的是八连通区域,所以对于普通的一个像素点只需扫描自己和周围已确定连通性的像素点就可确定自己的连通性,即扫描左、左上、上、右上 4 个像素的灰度值就可以了。如图 2 中的 e 像素,只要扫描 a,b,c,d 4 个像素的灰度值。

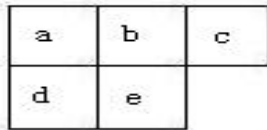


图 2 e 像素连通域扫描

2) 对每一目标像素点进行连通域的判断,如果跟目标像素点连通的像素点有 m 个(1<m)标记过,则按照左、左上、上、右上的优先顺序,确定当前像素的标记值。然后对这 m 个像素所拥有标记值作等价分析,如果 m 个像素标记值有不同的,则分别取不同标记值作为等价对,加入等价标记表中。如图 2 中,a,b,c,d 4 个像素的灰度值都为 255 时,把 d 像素的标记值赋给当前像素 e,然后做出左等价左上、左等价上、左等价右上 3 个等价对,再分别比较 3 个等价对中每个等价对中两个标记值是否相等,如果不相等就把该等价对加入到等价标记表中。

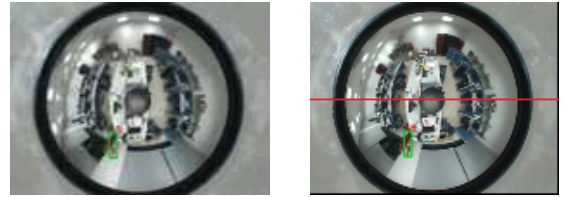
3) 采用基于递归的方式处理等价标记表,将等价标记表中的等价对处理完之后从等价标记表中删除,直到等价标记表为空。

4) 得到最终的运动目标区域的分割和相应的连通域个数。

4.2 连通域标记并行算法设计实现

4.2.1 游程码的设计

由于本文采用了任务分解^[6]的思想将目标区域的二值图像进行了数据的划分,再分别处理划分后各个子块。但是这样处理会将原本是同一区域的目标图像可能分割成几个非连通的连通域。因此需要对这些连通域进行合并,在算法设计过程中本文采用了增加游程码方法,记录分割边界处的连通域。通过游程码的计算来合并被分割开来的目标区域。如图 3 所示:



(a) 全方位视觉图 (b) 全方位视觉图分割处
图 3 全方位图像的分割处理

游程码是通过构造游程结构数组来表示分割后图像边界处的一行。本人在文献[7]的基础上对游程码的标记作了修改,将 L 记为数据块号。游程结构为 struct runcode {short Xstart;short Xend;short L;}, 其中 Xstart 代表连续相邻像素的起始位置,代表连续相邻像素的终止位置,L 代表数据块号。

$$R[j].X_{start} \leq R[i].X_{end} + 1 \quad (1)$$

$$R[j].X_{end} \leq R[i].X_{start} - 1 \quad (2)$$

则认为游程 $R[i]$ 和游程 $R[j]$ 是连通的。

4.2.2 通域标记并行算法的实现

并行算法的设计方法有三种途径: 检测和分析现有串行算法中固有并行性而直接将其并行化。从问题本身的特征出发,设计一个新的并行算法。修改已有的并行算法使其可求解另一类相似问题。

本文根据连通域标记的串行算法,分析其并行化的可行性,推导出基于多核的连通域标记的并行算法。将图像数据按照任务分割原则把数据分块,由主线程将把分割后的数据块交给各个子线程处理,各个子线程分别对各自的数据块进行连通域标记,最后将每个线程处理的结果统一汇总到主线程,得到最终的运动目标区域的分割和相应的连通域个数。

连通域标记算法的并行性分析:

通过对前面串行算法的分析,可以发现在二值图像中进行连通区域的标记过程中,事实上存在并行性:

1) 算法处理的数据是 M*N 的二值图像,图像中各个点的像素数据是无关的。

2) 算法对二值图像进行任务分割后的数据块之间也是无关的。

3) 在同一数据块内部,不论进行行计算还是列计算都是数据无关的。

由于这三种并行性的存在,可以对前面的算法进行并行化的设计,以提高算法的效率和处理器核的利用率。

连通域标记并行算法描述如下:

1) 读取二值图像,获得 M*N 的二值图像的大小,获取当前环境中核的个数;

2) 根据当前的处理器核的个数,确定对任务分解的块数并对二值图像进行划分;

3) 创建子线程,由各个子线程分别对各自的数据块进行连通域标记,并在分割出记录游程码;

4) 对各个线程处理的结果进行汇总计算,对游程码进行计算合并被分割开的连通区域;

5) 得到最终的运动目标区域的分割和相应的连通域个数。

4.2.3 实验结果

为了验证连通域标记并行算法在多核处理器上的核的利用率和运行速度是否已经提高,本文在双核处理器平台上分别对连通域标记串行算法和并行算法进行了测试实验。

其中:双核处理器: Intel(R)core(TM)2Duo CPU T5450 @ 1.66GHZ 内存 1G 运行操作系统 Vista;

单核处理器: 将双核处理器的核关闭一个。

程序运行的结果如表 1 所示:

表 1 两种算法运行结果的比较

算法	运行时间(单)	运行时间(多)	加速比
串行算法	465ms	423ms	465/294= 1.582
并行算法	487ms	294ms	

表 1 中的运行时间是仅连通域标记算法在系统中运行的时间,运行时间/s(单)是指在单核处理器中算法运行的时间,运行时间/s(多)是指在多核处理器中算法运行的时间。

本文将加速比定义为:

$$S(n) = \frac{\text{单核处理器上最优串行化算法计算时间}}{\text{使用 } n \text{ 个处理器并行化计算时间}};$$

加速比通常小于核的个数,加速比一般要求向 CPU 核数靠近,加速比越大,程序的性能越好。加速比指标是多核编程中有别于单核编程的重要因素,一般来说,要更好地发挥多核 CPU 的性能,加速比需要实现两个目标:必须随 CPU 核数的增加而线性增加;得到的加速比要尽量趋近 CPU 核数。

第一个目标的实现是非常重要的,否则更多的核反而是性能下降了,因此必须实现这个目标。第二个目标属于优化目标,在实际运用中,要想使加速比达到更高,需要花费更多的时间去设计和优化程序;因此,需要在软件开发成本和最终的加速比性能目标之间取得合理的平衡。

表 1 中列出了四个时间,分别是串行算法和并行算法在单核和双核上得到的数据,从数据中可以反映

出几点情况:

1) 串行算法在双核处理器上性能并没有多大提高,因此需要将程序并行化,否则无法提高核的利用率。

2) 并行算法在单核处理器上用了更多的时间,说明程序存在并行计算开销,主要包括算法开销,线程开销,线程切换等系统开销。算法开销主要是指将串行算法并行化后所增加的计算开销。

3) 并行算法在双核处理器上的运行时间 294ms,根据加速比定义算出, $S(2) = 465/294 = 1.582$ (精确到 3 位)。而没有并行化前, $S(2) = 465/423 = 1.099$;串行算法在双核上运行速度只提高了 1.099 倍。从这个实验结果看,算法通过并行化的设计,核的利用率和运行速度确实都提高了。

5 总结

本文针对全方位视觉实时跟踪处理图像的环境,根据多核并行计算的发展趋势,对现有的连通域标记算法在多核处理器上的进行了设计和实现。该算法在运行速度上有了明显的提高,并且充分利用了双核处理器上的两个核。下一步要研究的方向是针对多核处理器,对任务调度策略, CPU Cache 的存取等问题进行研究,可以使加速比进一步的提高。

参考文献

- 1 Sutter H. The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software. Dr.Dobb's Journal, 2005,30(3):1 - 9.
- 2 Fischer S. Technical Overview of the 45nm Next Generation Intel Core™ Microarchitecture (Penryn). IDF 2007, Beijing.
- 3 Khaira MS. Micro-2010: lead performance microprocessor of the year 2010-myth or reality. Proc. of Twelfth International Conference on VLSI Design. 1999. 157 - 163.
- 4 周伟明.多核计算与程序设计.武汉:华中科技大学出版社, 2009. 2 - 38.
- 5 徐正光,鲍东来,张利欣.基于递归的二值图像连通域像素标记算法.计算机工程, 2006,32(24):186 - 188.
- 6 Shameem A. Jason R. 多核程序设计——通过软件多线程提升性能.北京:电子工业出版社, 2007.
- 7 蔡世界,于强.基于游程编码的连通区域标记算法优化及应用.计算机应用, 2008,28(12):3150 - 3153.