

双线性插值图像放大并行算法的设计与实现

苗 莎, 郑晓薇

(辽宁师范大学 计算机与信息技术学院, 辽宁 大连 116081)

摘 要: 图像处理中对于图像放大常使用双线性插值算法, 其运行的快慢直接关系到实时图像处理的速度. 文中在多核计算机上设计了双线性插值并程序, 实现了图像放大的快速处理. 充分利用多核技术提升多核处理器的资源利用率, 缩短执行时间, 发挥多核系统的优异性能. 实验结果可看出, 文中算法对于大规模图像放大呈现出良好的加速处理能力, 表明了双线性插值多核并行算法在实时放大图像的应用中是有效及可行的.

关键词: 多核; 双线性插值; 并行算法; 图像放大

中图分类号: TP316

文献标识码: A

文章编号: 1000-7180(2011)11-0182-03

Design and Implementation of Image Zooming Parallel Algorithm Using Bilinear Interpolation

MIAO Sha, ZHENG Xiao-wei

(Department of Computer & Information Technology, Liaoning Normal University, Dalian 116081, China)

Abstract: Image processing often used bilinear interpolation algorithm to zoom images, its speed directly related to the real-time image processing speed. In multicore computer designed a bilinear interpolation parallel program, achieved to high speed process of image zooming. Full use of multicore technology to enhance the resource utilization of multicore processors, shorten the execution time and played superior performance of multicore system. Through experiment results, it can be seen that this algorithm showed a good ability in accelerate large image processing, indicate the bilinear interpolation multicore parallel algorithm in application of real-time image zooming is effective and feasible.

Key words: multicore; bilinear interpolation; parallel algorithm; image zooming

1 引言

在多核环境下, 如果程序仍然是串行执行, 多核 CPU 将无法发挥作用, 所以开发程序就必须考虑多核并行处理, 以有效利用多个处理核. 利用多核并行的数据分配是隐式的, 不用考虑数据在内存中的位置. 进程管理、消息传递、处理器间通信及同步操作由系统完成; 消息传递通信开销小, 不需要显式通信, 其通信时间要少于多机之间的通信. 因此多核并行的工作量较小, 使用简单. Intel 和 AMD 双核乃至四核处理器的推出使得许多应用领域利用多核来

提高问题的求解速度; 同时计算机体系结构的发展变化也使得利用 OpenMP 并行编程工具实现并行化的方法更加简便有效^[1].

图像放大作为图像处理中的基本操作, 在医学、军事、遥感、数字电视、航天系统等领域以及手机、相机等诸多数码设备上都有普遍的应用. 由于图像包含的数据量很大, 目前在个人电脑上进行图像放大仍是一个非常耗时的计算过程, 不能满足实际应用中的实时性要求. 单核时代, 在提高图像放大的速度上, 已有许多好的研究成果^[2-3]. 多核处理器的出现, 为其提供了另一个有效的途径. 采用多核并行算

法实现图像放大可以在更短的时间内生成图像,可以突破物理极限的约束,以较低的投入完成大量计算的任务。图像放大中常使用双线性插值算法,其求解速度很大程度上决定了图像放大的处理速度。文中设计了双线性插值多核并行算法,实验表明,该算法充分利用了多核处理器带来的性能提升,可有效提高图像放大的处理速度。

2 图像放大原理

图像放大通过将原图像的坐标 (x, y) 乘以放大系数 r_x 和 r_y 产生新的坐标 (x', y') , 变换表达式为 $x' = x \times r_x$, $y' = y \times r_y$ 。以矩阵的形式表示为:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} r_x & 0 & 0 \\ 0 & r_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

其中, $r_x > 1$, $r_y > 1$ 分别为 x, y 方向的放大比例。

当对一个图像进行放大时(如放大4倍),首先通过放大的倍数得到新图像的高度和宽度,此时新图像是一个待填充的空白图像网格。然后依据原图像中已知像素点的值,按照放大倍数找到它们在新图像中的位置并直接进行填充,即像素 $(4i, 4j)$ 的值直接取自原图像。但是所产生的空白像素点,其值是未知的(如点 p),这时可根据已知像素点的值采用插值算法进行估算,逐步填充空白网格直至完整,从而完成图像放大的处理。如图1所示。

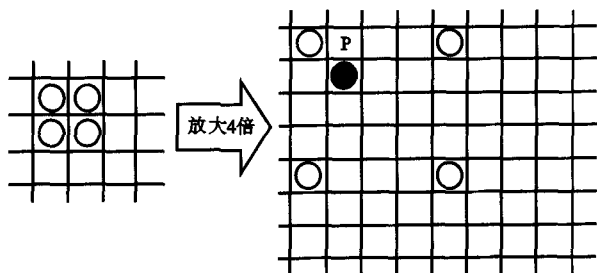


图1 图像放大4倍示意图

3 双线性插值图像放大算法

常用的图像放大算法中最邻近插值法简单直观,处理速度快,但它只是把原始像素简单复制到了其邻域内,放大后的图像会出现方块效应或锯齿现象;使用双线性插值法能够消除锯齿,放大后的图像较平滑,放大效果要比最邻近插值好很多,但其计算量较大,图像生成速度慢。文中将对双线性插值算法并行化以满足一些特殊场合需要快速放大图像的实时性要求。

双线性插值^[4]是利用需要处理的原图像像素

点周围的4个邻点的像素值在 x, y 两个方向上作线性内插,即待插点处的数值用离待插点最近的四个点的值加权求得。对于一个目的像素,设置坐标通过向后映射法得到浮点坐标 $(i+u, j+v)$ 。其中 i, j 是非负整数, u, v 是取值 $[0, 1)$ 区间的浮点数,随放大后像素行列坐标的不同而改变。则这个像素的值 $g(i+u, j+v)$ 可由原图像中坐标为 (i, j) 、 $(i+1, j)$ 、 $(i, j+1)$ 、 $(i+1, j+1)$ 所对应的周围四个像素的值决定,公式为:

$$g(i+u, j+v) = (1-u)(1-v)g(i, j) + (1-u)vg(i, j+1) + u(1-v)g(i+1, j) + uv g(i+1, j+1) \quad (2)$$

其中, $g(i, j)$ 表示源图像 (i, j) 处的像素值,以此类推。双线性插值法原理如图2所示。

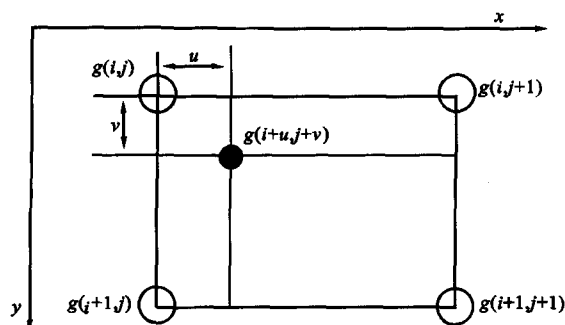


图2 双线性插值法示意图

4 并行算法设计与实现

双线性插值是在水平和垂直两个方向上均进行插值,分别沿着 X 轴和 Y 轴完成两次计算,计算过程中循环对图像每行和每列的数据进行计算,不同行列间的计算过程彼此独立不依赖于执行顺序。

双线性插值图像放大算法的实现步骤如下:

- (1) 读入原图像,得到图像的高度和宽度;
- (2) 根据放大倍数得到新图像的高度和宽度,并创建空白新图像;
- (3) 利用双线性插值得到像素点 (x', y') 的值并写回新图像;
- (4) 得到放大图像。

其中第3步,即双线性插值部分,需要对输出图像中的每个像素进行操作,运算量大,其快慢直接关系到图像放大的速度,是设计图像放大过程最为关键的环节。

使用双线性插值算法进行图像放大时,输出像素的值只由输入像素的邻域像素值确定,后一像素输出结果不依赖于前面像素处理的结果。且输入和输出图像分别利用两个数组来进行存储,处理过程

中不改写输入图像的数据. 因此插值过程以及所利用的像素资源之间均不存在相关性, 不同的输出值可以独立进行计算互不干涉, 具有很好的并行性.

OpenMP^[5]是共享存储系统编程的工业标准, 它具有编程简单、移植性好和可扩展等优点, 而且开发周期较短, 并行效率较高. OpenMP 采用 fork/join 并行执行模式, 其编程模型包括一组与平台无关的编译制导 (pragmas)、制导命令 (directive)、函数调用和环境变量. 双线性插值并行程序首先由主线程执行串行部分, 当执行 parallel for 编译指令时, 根据 `omp_set_num_threads(n)` 函数定义的线程数派生出 n 个子线程. 利用 `# pragma omp parallel` 指定后面的语句块由每个 CPU 并行执行. 插值任务被划分成多个子任务, 线程组中的每个线程分别执行对应的子任务, 各像素值同时插入对应的数据区域, 并行区结束后派生的线程退出, 主线程继续执行. 双线性插值图像放大的并行过程如图 3 所示.

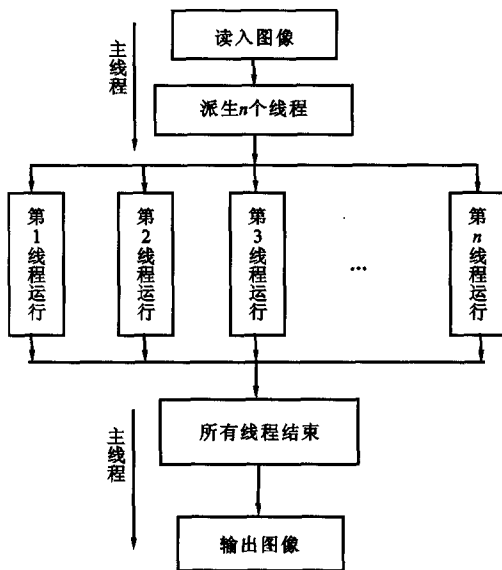


图3 双线性插值图像放大并行过程

并行处理后能够利用多核提供的强大计算能力, 任务线程被分配到各个核上同时执行, 大大缩短了运算时间, 加快图像放大的执行速度.

5 实验结果分析

实验平台是 Intel 双核电脑 (Intel Pentium Dual E2180 @ 2.00 GHz, 1.00 GB 的内存), 开发工具为 Visual C++ 6.0 和 Intel Compiler 9.1, 数据类型为双精度浮点型. 为了验证本研究提出的并行方法的有效性, 在双核平台上对一组不同尺寸的 lena 图像进行放大 2 倍的对比实验, 启动的线程数为 2 个, 分别运行串行程序和并行程序, 并记录处理时

间. 测试时间是算法的计算时间, 不考虑输入输出时间. 实验中以加速比 s_p 作为衡量并行计算性能的指标, 其中 $s_p = T_s/T_p$, T_s 为串行算法的运行时间, T_p 为并行算法的运行时间. 对实验数据进行了 10 次测试以求取时间的平均值, 数据如表 1 所示.

表1 不同尺寸图像串并行算法的运行时间

样本图像 大小(pixel)	平均运行时间(ms)		加速比
	串行算法	并行算法	
240×180	5.94	5.82	1.02
320×240	7.54	6.67	1.13
400×300	12.01	9.04	1.24
800×600	44.38	31.49	1.41

从表 1 可以看出, 通过并行化处理能缩短图像放大时间, 表明双线性插值并行算法的有效性.

双线性插值并行算法的性能与图像大小也有一定的关系, 四种图像规模的实验结果如图 4 所示.

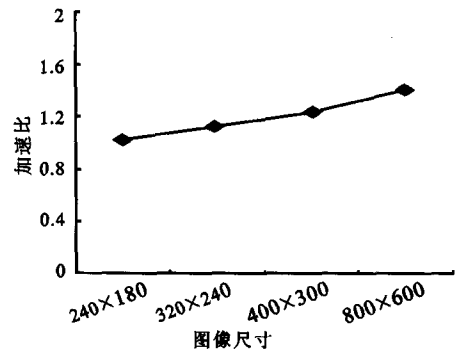


图4 加速比与图像大小的关系

观察图 4 可以看出, 当图像规模较小时, 加速比较小, 随着图像规模的增大, 加速效果更加明显. 这是因为随着图像数据量的增大, 并行开销占程序的总开销的比重越来越小, 并行计算的优势可明显地表现出来.

6 结束语

随着多核计算机的普及, 使得多核并行算法成为一个非常重要的研究方向. OpenMP 是当前比较有代表性的支持多核处理器的并行计算平台. 文中提出了基于 OpenMP 的双线性插值多核并行算法, 并通过对图像的放大处理实验证明了此并行算法的有效性. 在图像放大的计算过程中, 该方法利用了多核的优势, 实现了放大的加速处理, 对于大规模图像放大呈现出良好的实时处理能力. 文中采用的方法带有普遍性, 可推广到其它图像处理的并行化应用中.

(下转第 190 页)

的分组蚁群算法(HCACO),在不同的资源规模下完成资源查找的时间和同样的任务规模下多次执行任务时任务完成的时间。

从仿真结果可以看出,随着类似任务的多次执行,求解路径上的信息素逐渐积累,ACO、CACO 和 HCACO 的求解时间都呈现出一种下降趋势。通过横向比较,HCACO 具有着较为明显的优势。

6 结束语

文中借鉴对网络进行分簇的思想,提出了一种将被动发现与主动查找相结合的网格资源管理模型。与已有的网格资源管理方式相比,本模型通过结合移动 Agent 技术实现利用网络空闲来进行网格资源信息的初步收集,并通过在簇内 LDAP 目录服务器中保存邻簇的总体资源信息的摘要作为 QA 迁移策略的一个启发因子,提高网格资源发现和查找的效率,进而提高整个网格系统的使用性能。

参考文献:

- [1] 刘宴兵,尚明生,肖云鹏. 网格高性能调度及资源管理技术[M]. 北京:科学出版社,2010.5:192—193.

(上接第 184 页)

参考文献:

- [1] 钱晓捷,李秀芳. 基于多核多线程的排序算法优化和实现[J]. 微电子学与计算机,2011,28(1):116—119.
[2] 胡敏,张佑生. Newton Thiele 插值方法在图像放大中的应用研究[J]. 计算机辅助设计与图形学学报,2003,15(8):1004—1007.
[3] 陈建辉,许卫全,王博亮. 一种实时视频分辨率提升算法及其 VLSI 设计与实现[J]. 微电子学与计算机,2005,22(1):148—151.

- [2] Foster I, Kesselman C, Tuecke S. The anatomy of the grid: Enabling scalable virtual organizations [J]. International Journal: Supercomputer Applications, 2001, 15(3):1—4.
[3] 崔亚楠,黄文明,朱英. 基于 Agent 的网格资源共享方法研究[J]. 微电子学与计算机,2009(12):112—116.
[4] Yeong W, Howes T, Kille S. Lightweight directory access protocol. IETF RFC 1777, 03/2895. Draft Standard[s].
[5] 张奇,王汝传. 网格计算环境下基于移动代理的动态资源发现模型的研究[J]. 数据通信,2005(6):27—29.
[6] 党辰,王嘉帧,刘爱珍,等. 一种动态环境下的移动 Agent 智能迁移算法[J]. 计算机工程,2009(5):19—21.
[7] 黄文明,崔亚楠,朱英. 基于 Agent 的网格资源层次发现方法[J]. 计算机系统应用,2009(5):76—80.

作者简介:

王啸森 男,(1988—),硕士研究生. 研究方向为计算机网络、网络计算、移动 Agent 技术.

张新有 男,(1971—),副教授,研究方向为计算机网络、网络体系结构、网络计算、Adhoc 网络.

- [4] 向世明. Visual C++ 数字图像与图形处理[M]. 北京:电子工业出版社,2006.

- [5] Shameem Akhter. 多核程序设计技术—通过软件多线程提升性能[M]. 北京:电子工业出版社,2007.

作者简介:

苗莎 女,(1982—),硕士研究生. 研究方向为并行计算、多核计算机系统.

郑晓薇 女,(1957—),教授,硕士研究生导师,CCF 高级会员. 研究方向为并行计算、多核计算机系统.