

基于 OpenMP 的压缩感知并行处理算法

巫小婷, 邓家先*

(海南大学 信息科学技术学院, 海口 570228)

(*通信作者电子邮箱 jxiandeng@126.com)

摘要: 针对压缩感知重建算法复杂度高、运行时间长等缺点, 提出一种应用于多核处理器的压缩感知并行算法。在认真分析压缩感知算法的基础上, 利用 OpenMP 对压缩感知的编码测量和正交匹配追踪 (OMP) 算法进行并行处理, 提升程序的性能。实验结果表明, 随着线程数的增加, 程序的执行效率显著提高, 加速比呈线性增长; 并且重构过程越复杂, 其性能优化越明显。

关键词: 压缩感知; OpenMP; 并行; 正交匹配追踪; 加速比

中图分类号: TP301.6; TP391.4; TN911.73 **文献标志码:** A

Compressed sensing parallel processing algorithm based on OpenMP

WU Xiao-ting, DENG Jia-xian*

(College of Information Science and Technology, Hainan University, Haikou Hainan 570228, China)

Abstract: Concerning the high complexity and long-time running of the compressed sensing reconstructed algorithm, a compressed sensing parallel algorithm based on multi-core processors was proposed. On the basis of a careful analysis of the compressed sensing algorithm, OpenMP was used for compressed sensing measurement and Orthogonal Matching Pursuit (OMP) algorithm for parallel processing to improve program performance. The experimental results show that the speedup is in linear growth with the increasing threads. The execution of the procedure is more effective. Moreover, the more complex the reconstruction process is, the more obvious the performance optimization will be.

Key words: compressed sensing; OpenMP; parallel; Orthogonal Matching Pursuit (OMP); speedup

0 引言

随着多核处理器的诞生, 人们逐渐将目光放在多线程并行处理^[1]上。OpenMP^[2] (Open Multi-Processing) 作为并行程序的应用编程接口, 利用其编写压缩感知并行程序, 不仅开发简单, 而且通用性好。

压缩感知^[3-4] (Compressed Sensing, CS) 理论指出: 在某一已知变换域中稀疏表示的高维数据可以通过不相干的观测矩阵投影到低维空间中, 这些低维数据通过一定的优化算法可以重构出原始高维数据。目前, 压缩感知的研究主要集中在重建算法的优化和观测矩阵的选取等方面。近年来, 国内外学者相继提出了贪婪算法、凸优化算法、统计优化算法等重建算法, 大都在讨论怎样从少量数据中最大限度地重构原始数据。然而, 这些优化算法都需经过反复迭代才能获得精确的数据, 导致重构过程繁琐、运行时间长等问题。针对这一不足, Gan^[5] 提出了分块压缩感知算法, 先对图像分块, 再进行压缩感知。虽然此算法的复杂度有所降低, 但重构过程的执行效率仍然不高。

针对上述问题, 本文提出基于 OpenMP 的压缩感知并行处理算法。该算法将压缩感知的编码测量和重建算法这两部分分别划分为多个独立任务, 利用 OpenMP 将各任务分配到各线程中并行执行, 充分利用了多核处理器资源, 降低了程序的运行时间, 提高了程序的运行效率。

收稿日期: 2011-09-29; 修回日期: 2011-11-17。

作者简介: 巫小婷 (1987-), 女, 福建南靖人, 硕士研究生, 主要研究方向: 数字图像处理; 邓家先 (1964-), 男, 湖北钟祥人, 教授, 主要研究方向: 数字图像处理、数字滤波器设计、自适应信号处理。

1 OpenMP 简介

OpenMP 是一个为在共享存储的多处理器上编写并行程序而设计的应用编程接口, 由环境变量、编译指令及用来支持它的函数库组成。OpenMP 采用 Fork-Join 的并行执行模型 (见图 1): 当程序开始时, 主线程串行执行; 遇到派生点后, 主线程创建出一队并行线程, 执行并行区代码; 并行区的代码执行完成后, 派生的并行线程挂起或退出, 由主线程继续执行程序, 直到碰到下一个派生点。

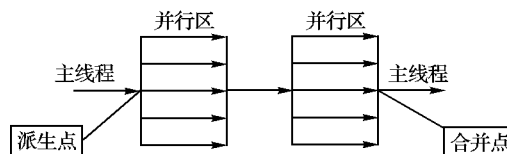


图 1 OpenMP 并行执行模型 (Fork-Join)

使用 OpenMP 编写并行程序, 开发简单且通用性强。只需在源代码中加入 #pragma 编译指导语句来指明意图, 编译器就可自动对程序进行并行化。而且 OpenMP 程序与平台无关, 当编译器不支持 OpenMP 时, 可自动忽略预处理指令 #pragma, 程序转为普通的串行程序, 代码仍然可以顺利地通过编译并正常运行。此外, OpenMP 对并行算法的高层抽象描述, 降低了并行编程的难度和复杂度, 可以很方便地应用于程序的性能优化。

2 压缩感知

压缩感知主要包括信号的稀疏表示、编码测量和重建算

法这 3 个方面。Candès^[6]指出,信号的稀疏性(或可压缩性)是压缩感知理论的前提条件。通常图像数据在水平和垂直方向相关性较强,经过离散余弦变换(Discrete Cosine Transform, DCT)后,系数的能量主要集中在低频成分。因此,变换后的系数是稀疏的或者近似稀疏的。本文对图像数据进行分块离散余弦变换,假定原始图像数据为 X (大小为 $N \times N$),变换后的系数(稀疏系数)为 Y , φ 为 DCT,则可表示为式(1)。而它的逆变换可以表示为式(2),其中 φ^{-1} 为逆 DCT(Inverse DCT, IDCT)。

$$Y = \varphi \cdot X \quad (1)$$

$$X = \varphi^{-1} \cdot Y \quad (2)$$

而编码测量就是将稀疏系数 Y 投影到与它不相干的观测矩阵 λ (大小为 $M \times N$, 且 $M \ll N$) 中,得到大小为 $M \times N$ 的观测值 Z ,表达见式(3)。因为高斯随机矩阵几乎与任何稀疏信号都不相关,所以一般选用高斯随机矩阵作为观测矩阵。

$$Z = \lambda \cdot Y \quad (3)$$

式(3)是欠定的,无法直接由观测值 Z 来重构稀疏系数 Y 。但当观测矩阵与稀疏系数不相关时,满足了约束等距(Restricted Isometry Property, RIP)条件,理论上稀疏系数 Y 的精确重构是可以通过求解最优 L_0 范数问题来解决^[7]。目前应用比较多的是贪婪追踪系列算法,如正交匹配追踪(Orthogonal Matching Pursuit, OMP)算法^[8-9],稀疏自适应匹配追踪(Sparsity Adaptive Matching Pursuit, SAMP)算法等。本文采用的是 OMP 重建算法,整个压缩感知的过程如图 2 所示。

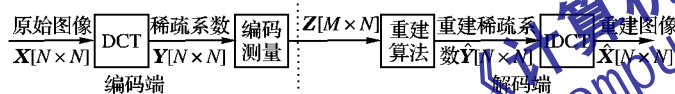


图2 压缩感知编解码过程

3 压缩感知并行算法

在压缩感知算法中,循环占据大部分的计算量,采用细粒度并行化这种针对循环的并行方法,只需在循环外使用 OpenMP 编译制导指令即可简单实现,无需关心其他的并行化细节,而且比粗粒度(函数级别)并行化的工作量要低。

然而,OpenMP 对循环进行并行化的同时,也带来了进程和线程间反复切换调度的开销。这对于运算时间不太长、计算量较小的循环而言,将导致并行执行效率低于串行执行效率。OpenMP 是基于共享存储的程序并行设计,当公用数据较多时,可能会产生数据冲突,影响并行程序执行效率,甚至导致程序执行结果出错。因此,并行优化时还需考虑循环中数据的独立性。基于以上几点,本文选取了编码测量与 OMP 信号重建算法这两部分进行细粒度并行优化。

3.1 编码测量

编码测量是压缩感知的核心,它不同于传统的压缩方法,直接将采样和压缩合二为一。编码测量的实质是稀疏系数矩阵与观测矩阵的相乘。首先将经过变换后的 DCT 系数分成大小相等的 m 块^[10],块编码测量就是各稀疏系数小块与观测矩阵相乘,整个编码测量过程由 m 个块编码测量构成,利用 OpenMP 将这些块编码测量分配到各线程中并行执行(见图 3),以减少程序执行时间。

3.2 重建算法 OMP

压缩感知的重建算法有多种,本文采用 OMP 算法。OMP 算法的实质是以贪婪迭代的方法选择观测矩阵(高斯随机矩

阵)的列,每次迭代中选择的列与当前的冗余向量最大限度地相关,然后去除相关部分,反复迭代,直到迭代次数达到给定迭代次数。OMP 算法保证了每次迭代最优性,减少了迭代次数。但每次迭代仅选取一个原子更新索引集,需付出很大的重建时间代价。此外,OMP 算法的迭代次数与稀疏系数的稀疏程度及采样个数 M 都有关联,为了保证精确重建,随着它们的增大,重建过程所耗费的时间也会增加。

在对 OMP 算法进行优化时,先对压缩数据进行分块,不仅减小了重建的迭代次数,降低运算复杂度,还有利于进行并行算法设计。在块压缩数据进行 OMP 重构的迭代过程中,当前循环的数据与前一次循环的数据相关性强,若在此处进行并行处理,将导致执行结果出错。由于各压缩块间的数据彼此独立,各块间的 OMP 重构过程也相互独立。因此,将派生点选择在 OMP 重构过程的入口,即将各数据块的重构分配到各线程中并行执行(见图 4)。此时,各个数据块的 OMP 重构运算量相当,OpenMP 编译指导自动将任务分配到各线程中,各线程的负载均衡,并行执行效率高。并且,该派生点位于程序的较高层,线程间切换的开销较小。

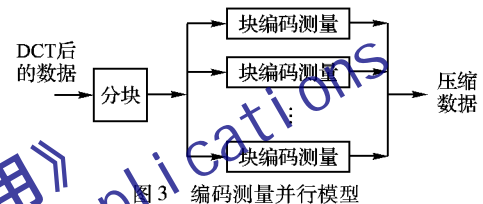


图3 编码测量并行模型

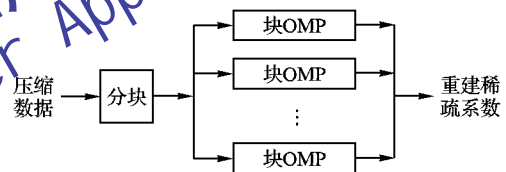


图4 OMP重建算法并行模型

4 性能评测与分析

4.1 性能评测标准

为了方便地描述并行程序的性能,一般采用加速比指标来度量。加速比定义为:

$$S(n) = \frac{\text{最优的串行化算法的运行时间}(T_1)}{n \text{ 线程并行算法的运行时间}(T_n)}$$

固定问题大小的加速比,即 Amdahl 定理。一般加速比越大,程序的性能越好。

4.2 实验结果

本文使用 Microsoft Visual Studio 2008 开发工具,采用 OpenMP 编译制导语句进行并行程序设计。实验平台为 HP XW6600 Workstation (Intel Xeon E5430/2.66 GHz 双 CPU (四核); 内存为 2 GB)。

本文对大小为 512×512 的 Lena 图像进行 16×16 的分块 DCT,将变换后的系数按照频率顺序排放,相同频率的系数放在同一小块中;再进行编码测量和 OMP 重建;然后对重构数据进行重排,系数按照频率的高低放在原始小块内;最后进行 IDCT。

表 1 是采样数 $M = 130$,迭代次数 $T = 60$,对所有图像数据进行压缩感知时,编码测量模块、OMP 重建模块及两个模块所耗费的时间。从表中得知,随着线程数的增加,耗费的时间逐渐减小,速度提升。在编码测量过程中,当线程数超过 6 时,耗时不再随着线程数的增加而减小,这是由于 OpenMP 开辟线程需要耗费一定的时间,线程的开销时间与并行节省的

时间抵消了。而 OMP 重建过程中由于耗时远远大于线程的开销时间,所以没有出现此现象。编码测量和 OMP 重建这两个并行过程在 2~8 线程的加速比分别是 1.94, 2.89, 3.78, 4.65, 5.43, 6.12, 6.92, 随着线程数的增加,加速比增大,程序的性能提升。

表1 压缩感知各并行模块耗费的时间

并行方式	单线程	并行线程数						
		2	3	4	5	6	7	8
编码测量	0.063	0.031	0.021	0.016	0.016	0.008	0.008	0.008
OMP 重建	29.969	15.454	10.344	7.938	6.438	5.516	4.890	4.328
编码测量 + OMP	30.032	15.485	10.375	7.954	6.454	5.531	4.906	4.343

表2 部分压缩感知算法的峰值信噪比

采样次数 M	迭代次数 T	不进行压缩感知的块数 ($n \times n$)	压缩比	峰值信噪比/dB
130	60	0×0	0.508	21.79
		1×1	0.510	26.09
		2×2	0.516	28.26
		3×3	0.525	30.21
192	100	0×0	0.750	30.06
		1×1	0.751	32.14
		2×2	0.754	34.01
		3×3	0.759	35.32

经过 DCT 的系数按照频率顺序排放后,低频系数分布在图像的左上方,这部分系数稀疏度较低,不适合 OMP 重建信号^[11]。因此,本文对图像左上角的 $n \times n$ 个小块(小块大小 16×16)不进行编码测量,即只对中高频的系数进行压缩感知,牺牲了少量的压缩比,提升图像的重建质量(见表2)。

表3 为不同线程下不同压缩比的压缩感知算法执行时间。从表3可看出:随着线程数量的增加,加速比逐渐增大,程序的耗时减小,程序的性能也越好。在相同的线程数下,压缩比值越高,程序的运行时间越长,加速比越大。也就是说,耗时长程序比耗时短的程序性能优化更明显。为了便于观察,选取了压缩比 0.508 和 0.750 的这两组数据进行折线图对比(见图5)。随着线程的增加,加速比几乎是呈线性增长的。但是,在双线程和4线程中,加速比与线程数较接近,而6和8线程中加速比明显低于线程数,这是因为 OpenMP 是基于共享存储的并行编程模型,由于多核计算机的内存访问带宽共享,线程过多将影响内存访问,增大开销。

表3 不同线程下不同压缩比的压缩感知算法执行时间

压缩比	单线程		双线程		4线程		6线程		8线程	
	执行时间/s	加速比	执行时间/s	加速比	执行时间/s	加速比	执行时间/s	加速比	执行时间/s	加速比
0.508	30.032	1.94	15.485	1.94	7.954	3.78	5.531	5.43	4.343	6.92
0.510	29.891	1.94	15.422	1.94	7.890	3.79	5.469	5.47	4.406	6.78
0.516	29.547	1.94	15.250	1.94	7.860	3.76	5.422	5.45	4.313	6.85
0.525	29.141	1.96	14.860	1.96	7.641	3.81	5.313	5.48	4.265	6.83
0.750	177.297	1.98	89.485	1.98	45.422	3.90	30.656	5.78	23.406	7.57
0.751	176.422	1.98	89.203	1.98	45.297	3.89	30.609	5.76	23.390	7.54
0.754	174.360	1.98	88.281	1.98	44.719	3.90	30.219	5.77	23.015	7.58
0.759	171.000	1.99	86.031	1.99	43.766	3.91	29.750	5.75	22.641	7.55

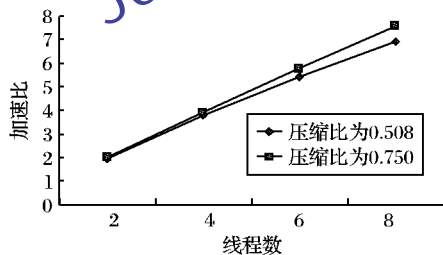


图5 不同压缩比压缩感知算法多线程加速比比较

5 结语

利用 OpenMP 在多核计算机上对压缩感知算法进行多线程并行。从减小程序运行时间和保证重建图像质量着手,根据共享存储编程的特点,对原有的压缩感知的串行算法进行多线程并行化处理,程序的性能较原有串行程序有很大的提升。

参考文献:

- [1] 周伟明. 多核计算与程序设计[M]. 武汉: 华中科技大学出版社, 2009.
- [2] OpenMP application program interface version 3.0 [EB/OL]. [2010-10-10]. <http://www.openmp.org/mp-documents/spec30.pdf>.
- [3] DONOHO D L. Compressed sensing[J]. IEEE Transactions on In-

formation Theory, 2006, 52(4): 1289–1306.

- [4] TSAIG Y, DONOHO D L. Extensions of compressed sensing [J]. Signal Processing, 2006, 86(3): 549–571.
- [5] GAN L. Block compressed sensing of natural images [C]// International Conference on Digital Signal Processing (DSP). Cardiff, UK: IEEE, 2007: 403–406.
- [6] CANDÈS E J. Compressive sampling [C]// Proceedings of International Congress of Mathematicians. Zürich, Switzerland: European Mathematical Society Publishing House, 2006: 1433–1452.
- [7] 刘亚新, 赵瑞珍, 胡绍海. 用于压缩感知信号重建的正则化自适应匹配追踪算法[J]. 电子与信息学报, 2010, 32(11): 2713–2717.
- [8] TROPP J A. Greed is good: Algorithmic results for sparse approximation[J]. IEEE Transactions on Information Theory, 2004, 50(10): 2231–2242.
- [9] TROPP J A, GILBERT A C. Signal recovery from random measurements via orthogonal matching pursuit [J]. IEEE Transactions on Information Theory, 2007, 53(12): 4655–4666.
- [10] 范晓维, 刘哲, 刘灿. 分块可压缩传感的图像重构模型[J]. 计算机工程与应用, 2009, 45(29): 153–155.
- [11] 杨成, 冯巍, 冯辉, 等. 一种压缩采样中的稀疏度自适应子空间追踪算法[J]. 电子学报, 2010, 38(8): 1914–1917.