第 38 卷 第 2 期　　电 子 科 技 大 学 学 报　　Vol.38 No.2
2009年3月　　Journal of University of Electronic Science and Technology of China　　Mar. 2009

FFT快速整序算法的对比、改进及实现

贾　渊[1]，王俊波[1]，姬长英[2]

(1. 西南科技大学计算机与技术学院　四川 绵阳　621010；2. 南京农业大学工学院　南京　210031)

【摘要】提出了一种改进的用于基2的FFT整序算法。改进算法对逆序表的生成进行改进，同时给出另一种数据交换的方案。首先，将顺序号分成组号和组员两部分，采用两个数组存储各组号及组员的数，以(0,2,1,3)为初始逆序表，利用已知组员与组号对应的逆序号的大小关系，求出任意更高阶的逆序表。其次，在数据交换时，避免了常规整序中顺序号与逆序号的比较运算。在Windows操作系统下编制了相关算法的C++程序，比较了运行效率，实验表明，改进算法效率最高。

关　键　词　FFT整序；组号；组员；序号分组；逆序

中图分类号　TP31　　　**文献标识码**　A　　　doi: 10.3969/j.issn.1001-0548.2009.02.32

Comparison, Improvement, and Implementation of FFT Permutation Algorithm

JIA Yuan[1], WANG Jun-bo[1], and JI Chang-ying[2]

(1. College of Computer Science and Technology, Southwest University of Science and Technology　Mianyang Sichuan　621010;

2. College of Engineering, Nanjing Agricultural University　Nanjing　210031)

Abstract A new improved permutation algorithm for radix 2 fast Fourier transforms (FFTs) is discribed. The algorithm provides an improved way for obtaining the bit reversal index table and affords an alternative data permutation methods. First, The algorithm divides an index with two parts, one is called group-index, which represents the index belonging to some group, and the other is called group-member which represents the order in the group. And then two arrays are used to store the values of the group-index and group-member power. According to the relationship between the bit reversal values of group-index and group-member, any bit reversal table can be obtained with the known one, which is (0,2,1,3). Second, when the data are needed to permute, the improved algorithm avoids the comparison operations between the index and its corresponding bit reversal value, which are required in the general permutation algorithm. In order to test its performance in the windows operation systems, several algorithms are programmed with C++ language, and their operation times are compared. The experiments show that the performance of the improved algorithm introduced in this paper is superior to that of the relative algorithms.

Key words　FFT permutation;　group-index;　group-member;　indices grouping;　inverse order

整序是FFT算法中的一部分，耗时约占整个算法过程的20%左右[1]。虽然算法简单，但设计不好会使效率下降[2]。常规整序算法除了第一个和最后一个序号，每个序号都要完成求逆序和比较的运算，运算量相当大[3-4]。研究者为提升整序效率进行了多种改进，部分算法专门针对硬件实现或并行处理[5-6]，但对于PC机并不合适。序号分组算法通过已知的低阶逆序表，用较小序号的逆序值求出较大序号的逆序值[7-8]，可减少逆序运算次数。在此基础上，文献[1]和文献[9]提出了任意基的快速整序算法，但该算法把求逆序以及整序结合在一起，理解困难。

无论是序号分组算法还是文献[1]提出的算法，都需要先求得逆序表，然后利用逆序表整序。序号分组算法求取逆序表的速度比文献[1]的算法更快，但算法不需要判断顺序号与逆序号的大小，两者各有优势。如果将两者结合起来，必将提高算法的运行效率。另外，算法采用递归求逆序表，内存消耗相当大。本文结合各算法优点，提出改进算法，并与相关算法比较了运行效率。

1　算法的基本数学原理

利用预先存贮的低阶(指数)逆序，可求出高阶逆序，低阶逆序表和高阶逆序表的关系如定理1所述。

定理 1　2^N 逆序表的前半部分(偶数)由 2^{N-1} 逆

收稿日期：　2007−11−22；修回日期：2008−04−15

基金项目：国家自然科学基金(10776028)；国家863项目(2008AA10Z211)

作者简介：贾　渊(1973−)，男，博士，副研究员，主要从事图形图像处理方面的研究.

序表对应值乘2组成，后半部分(奇数)由前半部分依次加1组成。

序号分组法在定理1的基础上，将 2^N 逆序表分成 $2^{[(N+1)/2]}$ 组([]表示取整)。每一组的数据有 $2^{N-[(N+1)/2]}$，各组的逆序表呈现如下规律[10]：

推论 1　第0组的值必然由 $2^{N-[(N+1)/2]}$ 逆序表对应值分别乘以 $2^{[(N+1)/2]}$ 组成。

推论 2　第 k ($k=1,2,\cdots,2^{[(N+1)/2]}-1$)组的值与第0组的对应值的差为常数，该常数为每一组的第0个值。

推论 3　第 k ($k=0,1,2,\cdots,2^{[(N+1)/2]}-1$)组的第0个值与 $2^{[(N+1)/2]}$ 逆序表中以 k 为索引下标的值相同。

推论 4　当 N 为偶数时，$[(N+1)/2]$ 等于 $N-[(N+1)/2]$；当 N 为奇数时，$[(N+1)/2]$ 比 $N-[(N+1)/2]$ 大1。

给定某序号 i 的二进制表达 $a_0a_1\cdots a_{N-1}$，可将其分成两部分 $a_0a_1\cdots a_{[(N+1)/2]-1}\big|a_{[(N+1)/2]}a_{[(N+1)/2]+1}\cdots a_{N-1}$，每部分对应的十进制值为 k、m，并分别称为组号和组员(表示第 k 组、第 m 组员)，对应的公式为：

$$i=k\times 2^{N-[(N+1)/2]}+m \tag{1}$$

如果已知 k、m 对应的逆序值为 I_k、I_m，则 i 的逆序值是：

$$I_i=I_m\times 2^{[(N+1)/2]}+I_k=I'_m\times 2^{N-[(N+1)/2]}+I_k \tag{2}$$

式中　I'_m 为 m 在 $2^{[(N+1)/2]}$ 逆序表中对应的值，当 N 为偶数时，$I'_m=I_m$；N 为奇数时，$I'_m=2I_m$，并有如下定理[1,9]：

定理 2　对于 $i=k\times 2^{N-[(N+1)/2]}+m$，如果 m 在 $2^{[(N+1)/2]}$ 逆序表中对应的值 I'_m 大于 k，则逆序值 I_i 必然大于 i，反之也成立。

定理2对快速整序提供了很好的理论支持。如果一个序号 i 由组号 k 以及组员 m 表示，对具有同一组员 m 的不同组号 k 的序号 i，有且只有 $k<I'_m$ 的那些序号 i 才会交换，而相互交换的序号值可通过式(1)和式(2)得到，其中 I'_m 为 m 在 $2^{[(N+1)/2]}$ 逆序表中对应的值。

2　算法设计

序号分组算法通过递归的形式得到 $2^{[(N+1)/2]}$ 逆序表，文献[1]的算法通过序号交换求逆序表，在实际运行过程中，该算法求逆序表的速度不如序号分组法，但该算法不需要进行序号比较就可以直接进行交换。结合两者优点，先采用改进的序号分组法求出 $2^{[(N+1)/2]}$ 的逆序表，然后用文献[1]的算法进行数据交换，整序的效率可得到提升。

2.1　改进的序号分组法求逆序表[11]

由序号分组算法可知，如果要对 2^N 序列进行整序，由长为 $2^{[(N+1)/2]}$ 的逆序表就可以实现。逆序表的求取可通过递归完成。可以证明，对长为 2^N ($N\geqslant 3$)的序列而言，需要求逆序的最小顺序表长为4($N=2$)，相应的逆序表是(0,2,1,3)，从而可用此逆序表生成任何 2^N ($N\geqslant 3$)的逆序表。为了消除递归，改进的逆序表生成算法分两步：

(1) 指数分解。将顺序号逐次分解为组号和组员两部分，并用两个数组Key₁[5]、Key₂[5]存贮组号与组员的指数(二进制位数)，直到组号对应的指数为2时为止，并记录下分解轮数 s。

(2) 逆序表生成。用一个长为 $2^{[(N+1)/2]}$ 的数组Data[$2^{[(N+1)/2]}$]保存逆序表，初始化时设Data[0]=0、Data[1]=2、Data[2]=1、Data[3]=3，根据指数分解的结果生成更高阶的逆序表。整个逆序表生成包括3层循环，最外层是分解轮数，根据轮数序号决定该轮需要生成的逆序表，逆序表的生成根据组号和组员实现，相应的公式为：

$$\left\{\begin{array}{l} s=T,T-1,\cdots,1\\ \left\{\begin{array}{l} \text{Length}=2^{\text{Key}_2[s]}\\ \left\{\begin{array}{l} k=2^{\text{Key}_1[s]}-1,2^{\text{Key}_1[s]}-2,\cdots,1,0\\ \left\{\begin{array}{l} m=0,1,2,\cdots,2^{\text{Key}_2[s]}-1\\ \text{Data}[k\text{Length}+m]=\text{Data}[k]+\text{Data}[m]\text{Length} \end{array}\right. \end{array}\right. \end{array}\right. \end{array}\right. \tag{3}$$

式中　k 为组号；m 为组员；Length 为组长度；s 为分解时的运算轮数。运算时从高到低对 s 取值，对应的求逆序过程就是从低阶到高阶。由于是在初始逆序表的基础上生成，为了防止运算时对初始数据的破坏，逆序表的生成从最后一组开始(逆序数据的存贮由后向前)。

2.2　整序

整序的方法采用文献[1]的算法提出的方案(即定理2)用两层循环完成，外层循环给定一个组员 m ($1\leqslant m<2^{N-[(N+1)/2]}$)，内层循环则从0组开始，只要该组号 k 小于组员号 m 的逆序，就直接交换(即 $0\leqslant k<\text{Data}[m]$)。由于每一组的第0个数据不交换，所以组员 m 从1开始。

3　算法分析

为了进一步对算法的性能进行分析，选择其中具有代表性的算法，在Windows XP环境下给出几种相关算法的程序实现。实验平台的基本情况为：方正文祥电脑(P4 3.0, 512MDDR)、Windows XP、 VC2003.net、C++。为了准确获得程序的运行时间，采用如下策略[12]：通过调用Query Performance Frequency()函数，获得系统的机器内部定时器的时钟频率，然后在需要严格定时的事件发生之前和发生之后分别调用Query Performance Counter()函数，利用两次获得的计数之差除以时钟频率，计算出事件经历的精确时间。由于Windows本身是分时多任务操作系统，其运行时间仍然存在一定的误差。几种算法求逆序表所需的平均时间如表1所示。表中的Wang算法参见文献[13]。

表1 求逆序运行时间表 单位：秒

指数	标准整序	Wang算法	序号分组	文献[1]算法	改进算法
20	0.09	0.01	0.01	0.10	0.01
21	0.19	0.06	0.03	0.20	0.02
22	0.40	0.15	0.06	0.37	0.05
23	0.81	0.32	0.11	0.74	0.10
24	1.66	0.65	0.21	1.50	0.21
25	3.36	1.31	0.40	3.02	0.38
26	6.79	2.6	0.76	6.04	0.73
27	>1 600	102.95	77.17	461.9	67.40

通过表1，不难得出如下结论：

(1) 在一定的指数范围内，文献[1]的算法与标准整序算法的效率相差不大，但是随着指数的增加，文献[1]的算法比标准整序算法效率更好。

(2) 序号分组算法与标准整序算法相比，其效率差异很大，说明改进是很成功的。而本文提出的改进算法与序号分组算法相比，由于消除了递归，程序稍复杂，但效率比没有消除递归时有明显的改善。

从几种算法求逆序表的效率看，改进的算法效率最高。为了比较传统整序算法、无须逆序数的FFT整序和改进算法的运行效率，本文设计了一组实验，对一给定的数组(已知逆序表)进行整序，耗时如表2所示。表中的第4列即为无须逆序数的FFT整序运行时间。

表2 3种FFT整序算法的运行效率对比

指数	标准	改进	无逆
20	0.10	0.09	0.05
21	0.20	0.19	0.11
22	0.41	0.37	0.24
23	0.88	0.74	0.50
24	1.80	1.54	1.10
25	3.64	3.12	2.22
26	7.36	6.27	4.60
27	>1 600	590.71	>1 600

通过表2，有如下结论：

(1) 改进算法的运算效率优于传统整序算法，平均节约时间在10%左右，当指数超过27后，本文算法的优越性更加明显。

(2) 改进算法在整序方面的效率提升并不如求逆序表那样明显。表2的算法对比是试验将逆序表转换成顺序表，因此标准整序算法的效率与表2中的结果较接近。而改进算法求逆序表的过程所耗费的时间较少，整序过程也是通过交换完成的，因此整序过程的时间主要耗费在数据交换上。

(3) 无须逆序数的FFT整序算法，通过数据交换的方法进行整序[14]。当$N<27$时，其效率比改进算法要高；当$N\geqslant 27$以后，后者效率远高于前者。这是由于数据交换的效率与计算机的软硬件环境相关，尽管后者的数据交换次数要少于前者，但两者使用存储单元的方式不完全一致，造成了效率差异。

(4) 逆序表的出现为图像的傅里叶变换的快速算法提供了很好的保证。由于图像的每一行(列)在进行傅里叶变换后，都需要整序，而每一行(列)的长度是一致的，因此整序时计算逆序号的过程都要重复，如果利用逆序表(只需生成一次)，就可对所有行(列)进行整序，节约的时间是可观的。

5 结 论

采用改进的序号分组算法求出逆序表，消除了递归；同时借鉴文献[1]的算法的优点，整序时不需要判断顺序号与逆序号的大小，直接进行数据交换，节省了判断所需的时间。当序列较短时，基于逆序表的算法，其效率并不一定优于传统整序算法，但用于图像傅立叶变换或硬件实现时，改进算法具有较高的效率。

本文研究工作得到西南科技大学校博士基金项目(08zx7101)和校“十一五”重点项目(062x113)的资助，在此表示感谢。

参 考 文 献

[1] 林水生，黄顺吉．一种实现任意基的快速整序算法[J]．电子科技大学学报, 1998, 27(4): 343-346.
LIN Shui-sheng, HUANG Shun-ji. A fast digit-reversal permutation algorithm for radix-B FFT[J]. Journal of University of Electronic Science and Technology of China, 1998, 27(4): 343-346.

[2] ALAN H K. Bit reversal on uniprocessor[J]. SIAM Review, 1996, 38: 289-307.

[3] 高 丽，刘卫新，张学智．FFT标准移序算法的优化[J]．探测与控制学报, 2004, 26(2): 62-64.
GAO Li, Liu Wei-xin, ZHANG Xue-zhi. Optimization of arranging order standard arithmetic for FFT[J]. Journal of Detection & Control, 2004, 26(2): 62-64.

[4] 胡广书. 数字信号处理理论、算法与实现[M]. 北京：清华大学出版社, 2003.
HU Guang-shu. Digital signal processing theories, algorithms and implementation[M]. Beijing: Tsinghua University Press, 2003.
[5] CARTER L, Vancouver GATLIN K S. Towards an optimal bit-reversal permutation program[C]//Proceedings of the 39th Annual Symposium on Foundations of Computer Science. Washington, DC, USA: IEEE Computer Society, 1998: 544-553.
[6] YEH C H, PARHAMI B. ART: robustness of meshes and tori for parallel and distributed computation[C]//Parallel Processing. Vancouver British Columbia, Canada: IEEE Computer Society Press, 2002: 463-72.
[7] EVANS D M W. An improved digit-reversal permutation algorithm for the fast Fourier and Hartley transforms[J]. IEEE Trans on ASSP, 1987(35): 1120-1125.
[8] JAMES S W. A new bit reversal algorithm[J]. IEEE Trans Acoustic, Speech and Signal Process, 1990, 38(8): 1472-1483.
[9] 林水生, 黄顺吉. 改进的任意基FFT整序算法[J]. 信号处理, 1999, 15(2): 163-165.
LIN Shui-sheng, HUANG Shun-ji. An improved digit–reversal permutation algorithm for the radix-B FFT[J]. Signal Processing, 1999, 15(2): 163-165.
[10] 张学智, 沈　虹. 实现快速傅里叶变换中逆序的新方法[J]. 西安工业学院学报, 2001, 21(3): 204-206.
ZHANG Xue-zhi, SHEN Hong. A method to inverse order for fast Fourier transformation[J]. Journal of Xi’an Institute of Technology, 2001. 21(3): 204-206.
[11] 贾　渊. 计算机视觉技术在牛肉自动分级中的应用基础研究[D]. 南京：南京农业大学, 2005.
JIA Yuan. Basic research on computer vision techniqu es applying to automatic beef grading[D]. Nanjing: Nanjing Agricultural University, 2005.
[12] 游志宇. VC中基于Windows的精确定时[DB/OL]. [2007-05-28]. http://www.vckbase.com/ document/ viewdoc/?id=1301. 2004.11.24.
YOU Zhi-yu. Precise timing with VC++ in Windows operating system[DB/OL]. [2007-05-28]. http://www. vckbase.com/docu ment/ viewdoc/?id=1301. 2004.11.24.
[13] Wang H H. On vectorizing the fast Fourier transform[J]. BIT Numerical Mathematics, 1980, 20(2): 233-243.
[14] 张学智, 毛　俊. 无须逆序数的FFT整序的一个新方法[J]. 探测与控制学报, 2002, 24(2): 61-63.
ZHANG Xue-zhi, MAO Jun. A new method to inverse order for FFT without inverse order number[J]. Journal of Detection & Control, 2002, 24(2): 61-63.

编　辑　熊思亮

(上接第281页)

[3] 于晓含, 朱哈·伊垒·叶斯基. 基于区域增长及边缘检测的一种图像分割方法[J]. 北方交通大学学报, 1997, 21(1): 47-52.
YU Xiao-han, YLA-JAASKI J. A General Image Segmentation Method Combining Region Growing and Edge Detection[J]. Journal of Northern Jiaotong University, 1997, 21(1): 47-52.
[4] CREMERS D, TISCHHAUSER F, WEICKERT J, et al. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional[J]. International Journal of Computer Vision, 2002, 50(3): 295-313.
[5] ZHANG Hong-ying, WU Bin, PENG Qi-cong. An improved algorithm for image edge detection based on lifting scheme[J]. Journal of Electronic Science and Technology of China, 2005, 3(2): 113-115.
[6] MCINERNEY T, TERZOPOULOS D. Deformable models in medical image analysis: a survey[J]. Medical Image Analysis, 1996, 1(2): 91-108.
[7] JUSTICE R K，STOKELY E M. 3-D segmentation of MR brain images using seeded region growing[C]//18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Amsterdam: IEEE, 1996: 1083-1084.
[8] KASS M, WITKIN A, TERZOPOULOS D. Snake: Active contour models[J]. International Journal of Computer Vision, 1987, (4): 321-331.
[9] TERZOPOULOS D, WITKIN A, KASS M. Constraints on deformable models: Recovering 3d shape and nonrigid motion[J]. Artificial Intelligence, 1988, 36: 91-123.
[10] XU C Y, PRINCE P L. Snakes, shapes, and gradient vector flow[J]. IEEE Trans on Image Processing, 1998, 7(3): 359-369.

编　辑　漆　蓉