

多级多维离散小波变换的快速提升计算

钟广军¹, 成礼智², 陈火旺¹

(1. 国防科技大学计算机学院, 湖南长沙 410073; 2. 国防科技大学理学院, 湖南长沙 410073)

摘要: 提升方法是计算离散小波变换的有效手段, 它由一系列的提升步和拉伸变换组成. 在计算多级和多维离散小波变换时, 现有方法在每一次小波分解的过程中都做完整的提升步计算和拉伸变换计算. 我们发现该方法存在运算过程的冗余, 为此本文提出了一种称之为后拉伸变换的提升方法, 基本思想是计算完所有的提升步后, 再统一进行拉伸变换. 它能减少离散小波变换的乘法运算量. 例如, 对图像与视频压缩中应用广泛的 Daubechies 9/7 小波, 做一维 5 级分解时与现有方法相比, 乘法运算减少 20%, 而二维 5 级分解时, 乘法运算减少 28%.

关键词: 离散小波变换; 提升; 图像压缩; 快速计算

中图分类号: TN917 **文献标识码:** A **文章编号:** 0372-2112 (2001) 11-1475-03

Fast Lifting Algorithm for Multi-Level and Multidimensional Discrete Wavelet Transform

ZHONG Guang-jun¹, CHENG Li-zhi², CHEN Huo-wang¹

(1. School of Computer, National University of Defense Technology, Changsha 410073, China;

2. School of Science, National University of Defense Technology, Changsha 410073, China)

Abstract: Lifting Scheme can be used to calculate Discrete Wavelet Transform (DWT) efficiently. It is composed of a number of lifting steps and the scaling transform. When it is used to calculate multi-level and multidimensional DWT, there exist many redundant multiplications. In this paper we propose, a new technique, called Post Scaling Lifting Algorithm (PSLift), which needs fewer multiplications. The basic idea is first to calculate all lifting steps of each level and each dimensional DWT, and then calculate the scaling transform. For example, when five level decompositions are implemented for Daubechies 9/7 biorthogonal wavelet, compared to the known methods, the number of multiplication needed is reduced by 20% and 28% compared to one dimensional DWT and two dimensional DWT.

Key words: discrete wavelet transform; lifting; image compression; fast computation

1 引言

小波变换在信号与图像处理中应用广泛, 其离散小波变换(DWT)的高效计算一直是人们感兴趣的研究课题. Sweldens^[1] 引进的提升方法(Lifting Scheme) 由于其计算复杂性只是原有的滤波器卷积方法的一半左右^[2], 因而成为计算 DWT 的主流方法^[3,4]. 对宽度较小的小波滤波器, 减少其 DWT 的运算量是非常困难的^[5], 提升方法由于使 Daubechies 9/7 小波^[6] (以下简称为 D97 小波) 的加法运算量降低了 43%, 乘法运算量降低了 33%^[2], 因而它被推荐为 JPEG2000^[7] 计算 DWT 的方法.

用提升方法计算 DWT 由两个过程组成: 首先对输入信号做一系列的提升步 (lifting steps), 得到中间系数, 然后对中间系数进行拉伸变换 (scaling transform). 用它计算多级 DWT 或多维 DWT 时, 现有的方法是依次按级或按维做 DWT. 研究发现, 该方法存在重复运算, 重复运算是由在每次小波分解的过

程中都做一次拉伸变换引起的. 因此对算法结构进行调整: 在计算完多级和多维的所有的提升步后, 再对中间系数进行统一的拉伸变换处理, 这样减少了多次拉伸变换的冗余运算, 因而也减少了计算 DWT 的运算总量.

文章第 2 节分析了用现有提升方法计算多级一维 DWT 的冗余乘法运算, 并提出能够减少乘法运算的后拉伸变换的提升方法. 第 3 节提出计算二维 DWT 的后拉伸变换方法, 在第 4 节推广到多维 DWT, 最后是实验结果分析和结论.

2 多级一维 DWT 的快速提升方法

沿用文[2]的一些基本术语. 记小波滤波器对为 $\{h, g, \tilde{h}, \tilde{g}\}$, $P_S(z)$ 为其综合端多相矩阵, $P_a(z)$ 为其分析端多相矩阵, 文[2]利用 Laurent 多项式的 Euclidean 算法, 将 $P_S(z)$ 转换成下式:

$$P_S(z) = \prod_{i=1}^m \begin{bmatrix} 1 & -s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -t_i(z) & 1 \end{bmatrix} \begin{bmatrix} 1/K & 0 \\ 0 & K \end{bmatrix} \quad (1)$$

其中 $s_i(z), t_i(z)$ 为 Laurent 多项式, K 为非零常量, 称之为拉伸系数. 相应的 $P_a(z)$ 表示为:

$$P_a(z) = P_a^{ST}(z) P_a^{LT}(z) \quad \text{其中 } P_a^{ST} = \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix},$$

$$P_a^{LT}(z) = \prod_{i=m}^1 \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \quad (2)$$

从式(2)可以看出, 小波分解由两部分组成, 第一部分是一系列的提升步, 即 $P_a^{LT}(z)$ (以下简称为提升步), 第二部分是拉伸变换, 即 $P_a^{ST}(z)$. 在计算拉伸变换时, 各个数据之间没有关联, 只是在自身的基础上放大 K 倍或者缩小 $1/K$.

在对信号进行小波分解时, 一般都要做多级分解. 设 $J(J \geq 1)$ 为分解级数, $\{c^0\}$ 为原始信号, $\{c^{j+1}\}, \{d^{j+1}\}$ 是对 $\{c^j\}$ 做一次小波分解得到的低频分解部分和频带细节部分, 现有的多级提升方法^[3]可描述为算法 1:

算法 1. 现有的多级提升算法

for($j=0; j < J; j++$)

按照 $P_a^{LT}(z)$ 对 $\{c^j\}$ 做提升步计算, 得到低频分量 $\{lc^{j+1}\}$ 与高频分量 $\{ld^{j+1}\}$;

$\{lc^{j+1}\}$ 放大 K 倍得到 $\{c^{j+1}\}, \{ld^{j+1}\}$ 乘以 $1/K$ 得到 $\{d^{j+1}\}$;

记 M 为计算 $P_a^{LT}(z)$ 时每个输入信号的平均乘法次数, 于是做 1 级小波分解其平均乘法次数为 $M+1$. 用算法 1 计算 J 级 DWT 需要的平均乘法次数为:

$$(M+1)(1-2^{-1+J} + \dots + 2^{-(J-1)}) = 2(1-2^{-J})(M+1) \quad (3)$$

由于 $P_a^{LT}(z)$ 是一种线性变换, 另外从算法 1 看到 $\{lc^{j+1}\}$ 与 $\{c^{j+1}\}$ 有下述倍数关系: $\{c^{j+1}\} = K\{lc^{j+1}\}$, 因此在多级小波分解过程中, 利用线性变换与倍数关系, 可以先做完所有的提升步, 然后对剩余的低频系数和频带系数进行拉伸变换处理, 具体过程为:

算法 2 多级一维 DWT 的后拉伸变换提升算法

第一步(提升步):

①按照 $P_a^{LT}(z)$ 对 $\{c^0\}$ 做提升步计算, 得到 $\{lc^1\}, \{ld^1\}$;

②for($j=1; j < J; j++$)

按照 $P_a^{LT}(z)$ 对 $\{lc^j\}$ 做提升步计算, 得到 $\{lc^{j+1}\}, \{ld^{j+1}\}$;

第二步(后拉伸变换):

①将 $\{lc^j\}$ 放大 K^j 倍得到 $\{c^j\}$;

②for($j=J; j > 2; j--$)

将 $\{lc^j\}$ 放大 K^{j-1} 倍得到 $\{d^j\}$;

③将 $\{ld^1\}$ 乘以 $1/K$ 得到 $\{d^1\}$;

利用算法 2 计算 J 级分解需要的平均乘法次数为:

$$M(1+2^{-1+J} + \dots + 2^{-(J-1)}) + 1 - 1/4 = 2(1-2^{-J})M + 3/4 \quad (4)$$

由于算法 2 把拉伸变换放在最后执行, 使花在拉伸变换上的总运算量减少, 从而降低了计算 DWT 的运算量. 实际中经常用到的小波, 计算 $P_a^{LT}(z)$ 的平均乘法运算次数 M 一般比较小, 拉伸变换运算量占有较大的相对比重, 因此上述运算量的减少是非常有意义的. 如 D97 小波^[2]其 $M=2$. 若设分解级

数 $J=5$, 根据(3)和(4), 算法 1 与算法 2 在每个信号上的平均乘法运算量分别为 5.81 和 4.63, 算法 2 比算法 1 减少了 20%.

3 二维 DWT 的后拉伸变换方法

在计算二维可分离的 DWT 时, 一般采用行列法逐级分解, 如图 1 所示: 先按行对输入数据做小波分解, 再按列对得到的中间数据做小波分解. 记述输入数据为 LL_0 , 它经过 1 级二维 DWT 分解成四个子带: LL_1, HL_1, LH_1, HH_1 .

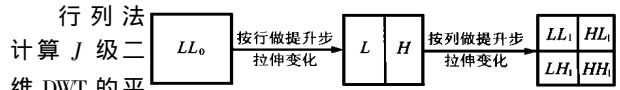


图 1 用行列法计算 2 维 DWT

平均乘法运算量为:

$$2(M+1)(1+4^{-1+J} + \dots + 4^{-(J-1)}) = 8(1-4^{-J})(M+1)/3 \quad (6)$$

类似于多级一维 DWT 后拉伸变换提升方法的讨论, 在按行做完提升步后, 接下来不做拉伸变换, 而接着按列做提升步, 等到每一级和每一维的提升步都做完后, 最后再做拉伸变换. 图 2 给出了 1 级二维 DWT 后拉伸变换方法的流图, 图 3 则给出了 2 级二维 DWT 的后拉伸变换方法的流图. 对于一般的 $J(J \geq 1)$ 级二维 DWT 后拉伸变换方法, 则可总结为算法 3.

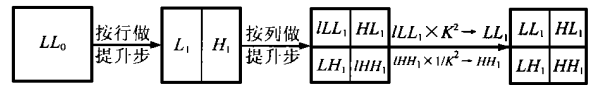


图 2 1 级二维 DWT 的后拉伸变换提升方法

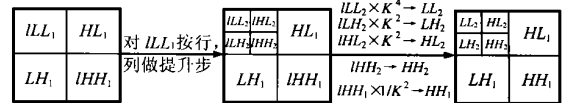


图 3 2 级二维 DWT 的后拉伸变换提升方法

算法 3. 二维 DWT 的后拉伸变换提升算法

第一步(提升步):

①按行对原始二维数据 LL_0 做提升步计算, 得到 L_1, H_1 ;

②按列对 L_1, H_1 做提升步计算, 得到 LL_1, HL_1, LH_1, HH_1 ;

③for($j=1; j < J; j++$)

按行对 LL_j 做提升步计算, 得到 L_{j+1}, H_{j+1} ;

按列对 L_{j+1}, H_{j+1} 做提升步计算, 得到 $LL_{j+1}, LH_{j+1}, HL_{j+1}, HH_{j+1}$;

第二步(后拉伸变换):

① LL_j 放大 K^{2j} 倍得到 LL_j ;

②for($j=J; j > 2; j--$)
 LL_j 放大 $K^{2(j-1)}$ 倍得到 LH_j, HL_j 放大 $K^{2(j-1)}$ 倍得到 HL_j ;

HH_j 放大 $K^{2(j-2)}$ 倍得到 HH_j ;

③if($J > 1$)

LL_2 放大 K^2 倍得到 LH_2, HL_2 放大 K^2 倍得到

HL_2 , 把 HHH_2 直接赋给 HHH_2 ;

④ HHH_1 乘以 $1/K^2$ 得到 HH_1 ;

算法 3 计算 J 级二维 DWT 的平均乘法运算量为

$$\begin{cases} 2M + 1/2, & J = 1 \\ 8M(1 - 4^{-J})/3 + 7/16, & J \geq 2 \end{cases} \quad (6)$$

根据式(5)和(6), 以 D97 小波为例, 设 $J = 5$, 行列法的平均乘法运算量为 7.99, 而算法 3 的平均乘法运算量为 5.77, 算法 3 减少了 28%.

4 多维 DWT 的后拉伸变换方法

算法 3 很容易推广到 n 维 ($n > 2$) DWT, 只不过把其中的行列二维改为多维. 在这里仅分析其乘法运算量. 容易推出计算 J 级 n 维 DWT 的平均乘法运算量为:

$$\begin{cases} nM + 1, & J = 1, n \text{ 为奇数} \\ nM + 1 - P_n^{n/2}/2^n, & J = 1, n \text{ 为偶数} \\ nM(1 - 2^{-nJ})/(1 - 2^{-n}) + 1 - 2^{-2n}, & J \geq 2, n \text{ 为奇数} \\ nM(1 - 2^{-nJ})/(1 - 2^{-n}) + 1 - 2^{-2n} - P_n^{n/2}/2^n, & J \geq 2, n \text{ 为偶数} \end{cases} \quad (7)$$

其中 $P_n^{n/2}$ 表示从 n 个元素中选择 $n/2$ 个元素的排列运算. 而行列法的平均乘法运算量为:

$$n(M + 1)(1 - 2^{-nJ})/(1 - 2^{-n}) \quad (8)$$

对 D97 小波, 假设分解级数 $J = 5$, 计算 3 维 DWT^[4, 8] 用推广的行列法平均乘法运算量为 10.29, 而用后拉伸变换方法仅为 7.84, 减少了 24%.

5 实验结果与分析

为了检验本文提出的后拉伸变换提升方法的有效性, 以 Daubechies 9/7 小波为具体的小波滤波器, 在 Pentium II 400 微机, Windows98 平台上测试了它与行列法进行 5 级小波分解的实际运行时间.

表 1 二维后拉伸变换方法与行列法的实际运行时间比较(单位: 秒)

输入数据的尺寸	行列法运行时间	后拉伸变换方法的运行时间
512×256	0.11	0.05
512×512	0.22	0.16
1024×512	0.44	0.38
1024×1024	0.93	0.83
2048×1024	1.81	1.64
2048×2048	3.68	3.35

从表中可以看出: 本文的后拉伸变换方法相对于行列法, 运行时间要短一些. 当输入数据的尺寸较小时, 两者的运行时间差距比较明显; 当输入数据尺寸增大后, 差距维持在 10% 左右. 考虑到操作系统进行系统调用的一些不确定因素, 以上波动应属正常.

6 结论

采用提升方法能有效计算 DWT, 但是用现有的提升方法计算多级和多维 DWT 时, 仍有冗余的乘法运算. 本文提出后拉伸变换的提升方法, 它能减少多级和多维 DWT 中的乘法运算, 并且该方法具备提升方法的优点: 例如它不需要占用额外的内存空间, 仍然可改造成整数到整数的变换. 对压缩中常用的 Daubechies 9/7 小波, 乘法运算的减少是非常明显的, 对图像做二维 5 级分解时能减少 28%. 因此, 在图像压缩和视频压缩中采用本文提出的后拉伸变换提升方法, 势必能加速编码和解码过程.

参考文献:

- [1] W Sweldens. The lifting scheme: A construction of second generation wavelets [J]. SIAM J. Math. Anal., 1997, 29(2): 511-546.
- [2] I Daubechies, W Sweldens. Factoring wavelet transforms into lifting steps [J]. J. Fourier Anal. Appl., 1998, 4(3): 245-267.
- [3] W Jiang, A Ortega. Parallel architecture for the discrete wavelet transform based on the lifting factorization [A]. Proc of SPIE in Parallel and Distributed Methods for Image Processing III, July 1999. Available from <http://biron.usc.edu/~wjiang/research.html>.
- [4] J Xu, S Li, Z Xiong, Y Q Zhang. Memory-constrained 3D wavelet transforms for video coding without boundary effects [A]. Proc. ISPACS'00 [C], Honolulu, HI, November 2000. Available from <http://www.microsoft.com/china/research/downloads/gimedia/spli/IMG-37.pdf>.
- [5] O Rioul, P Duhamel. Fast algorithms for discrete and continuous wavelet transforms [J]. IEEE Trans. on Information Theory, 1992, 38(3): 569-586.
- [6] M Antonini, et al. Image coding using wavelet transform [J]. IEEE Trans. on Image Processing, 1992, 1(2): 205-220.
- [7] <http://www.jpeg.org/fcd15444.1.htm>. JPEG2000 Final CD.
- [8] B K Kim, Z Xiong, W A Pearlman. Very low bitrate embedded video coding with 3D set partitioning in hierarchical trees (3D SPIHT) [J]. IEEE Trans. Circuits and Systems for Video Technology, Special Issue, September 1998.

作者简介:



钟广军 男, 1974 年 1 月出生于湖南澧县. 现在国防科技大学计算机学院攻读博士学位. 目前感兴趣的研究领域有图像处理, 视频压缩.