离散小波变换的最优Lifting分解方法

吴江华,张田文
(哈尔滨工业大学计算机科学与技术学院,黑龙江哈尔滨 150001)

摘 要: 现有Lifting分解(Lifting Factorization,LF)算法不能在较短时间内求解出较长离散小波变换(Discrete Wavelet Transform,DWT)滤波器组的所有数值稳定LF,更无法计算最优LF.在LF的形式化分析基础上,提出了DWT多相矩阵的常数最大公因子分解方法,极大缩小了分解搜索空间;进一步,从数值稳定性及计算复杂性两方面研究了LF评价问题;最后,提出了基于联合优化的最优LF算法.实验结果表明,本文算法在解的质量、求解时间及适用范围上都有较大改进.

关键词: DWT; 最优Lifting分解; Euclidean算法; 分解稳定性
中图分类号: TN911.72 **文献标识码**: A **文章编号**: 0372-2112 (2004) 11-1910-05

An Approach to the Optimal Lifting Factorization of DWT

WU Jiang-hua, ZHANG Tian-wen
(*School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang* 150001, *China*)

Abstract: Lifting factorization (LF) was one of the latest advancements of study on fast DWT implementations. Because of exponential size of the factorization set, known algorithms could not find in reasonable time all stable (to say nothing of optimal) LFs of a given DWT with long filters. In this paper, a so-called constant GCD (Greatest Common Divisor) factorization approach to FIR polyphase matrix was given, which contracted considerably the factorization space. Furthermore, the problem of how to evaluate a certain LF was investigated in terms of both numerical stability and computation complexity. Consequently, an algorithm called OLFA (Optimal LF Algorithm) was readily available. All of the theoretical results were constructively proven. Experimental data show that OLFA obtains considerable improvements in solution quality, computation time and application range over the existing algorithms, thus makes LF a tool of great generality and practicability for fast DWT implementations.

Key words: DWT; optimal lifting factorization; Euclidean algorithm; stability of factorization

1 引言

Sweldens提出的Lifting Scheme旨在构造第二代小波[1]及经典小波[2].但Daubechies和Sweldens进一步研究发现[3],任何双带FIR完全重建系统的多相矩阵都可分解为有限个Lifting步和对偶Lifting步的交替级联.此结论从DWT的实现角度指出,Lifting实现较标准多相实现更有效,且可用于双正交情形.因此,LF迅速受到广泛关注[4~7].基本原理如下:

设全体实Laurent多项式记为 $\mathbf{R}[z,z^{-1}]$;任意非零Laurent多项式 $p(z)=\sum_{n=t}^{s}p_n z^{-n}$ 的次数定义为 $|p(z)|\triangleq s-t$, $|0|$ 定义为 $-\infty$;当两个Laurent多项式的 GCD 为零次时称它们互质.

给定 $a(z), b(z)\in\mathbf{R}[z,z^{-1}]$,若 $|a(z)|\geq|b(z)|>-\infty$,则令 $a_0(z)=a(z), b_0(z)=b(z), i=0$,由Euclidean迭代 $a_{i+1}(z)=b_i(z), b_{i+1}(z)=a_i(z)\%b_i(z), i=i+1$ 可求得 $GCD(a(z),b(z))=a_n(z)$,其中 $n=\min\{j|\ b_j(z)=0\}$.该过程的逆可表示为: $\begin{bmatrix} a(z)\\ b(z)\end{bmatrix}=\prod_{i=1}^{n}\begin{bmatrix} q_i(z) & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix} a_n(z)\\ 0\end{bmatrix}$,其中 $q_{i+1}(z)=a_i(z)/b_i(z)$

对于给定的DWT多相矩阵 $P(z)$,总可通过列变换使得 $\det P(z)=1$,故其多相分量 $h_e(z), h_o(z)$ 互质,进而有它们的 GCD 必为单项式.特别地,当该 GCD 为常数 K 时,由上式有

$$\begin{bmatrix} h_e(z)\\ h_o(z)\end{bmatrix}=\prod_{i=1}^{n}\begin{bmatrix} q_i(z) & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix} K\\ 0\end{bmatrix}\tag{1}$$

当迭代步数 n 为偶数时,可如下构造行列式为1的多相矩阵

$$P^0(z)=\begin{bmatrix} h_e(z) & g_e^0(z)\\ h_o(z) & g_o^0(z)\end{bmatrix}=\prod_{i=1}^{n}\begin{bmatrix} q_i(z) & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix} K & 0\\ 0 & 1/K\end{bmatrix}\tag{2}$$

做恒等变换[3]后,由Lifting定理[2]即可得 $P(z)$ 的LF:

$$P(z)=\prod_{i=1}^{n/2}\begin{bmatrix} 1 & q_{2i-1}(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix} 1 & 0\\ q_{2i}(z) & 1\end{bmatrix}\begin{bmatrix} 1 & K^2 s(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix} K & 0\\ 0 & 1/K\end{bmatrix}$$
$$=\prod_{i=1}^{m}\begin{bmatrix} 1 & s_i(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix} 1 & 0\\ t_i(z) & 1\end{bmatrix}\begin{bmatrix} K & 0\\ 0 & 1/K\end{bmatrix}\tag{3}$$

在上述求解过程中有以下问题尚未解决:

(1) $P(z)$ 的LF不具有唯一性.DS(Daubechies and Sweldens)没有给出 $s_i(z), t_i(z)$ 全部解的求解算法[3].MA

收稿日期:2003-07-15;修回日期:2004-07-07
基金项目:国家自然科学基金(No.6047511,No.60075010)

(Maslen and Abbott) 虽然讨论了 LF 的自动化及其程序实现[8]，但没有给出形式化分析或证明.

(2) DS 分解方法总假设 $GCD(h_e(z), h_o(z))$ 为常数[3]，但是目前尚无有效方法来保证这一点.

(3) $P(z)$ 滤波器稍长时，现有算法很容易因出现病态方程组而得到误差很大的 LF，无法精确实现 DWT.

(4) 各 LF 之间有“好”、“坏”之别，但文献中还没有关于 LF 评价方面的研究结果.

所以，目前的算法不能在较短时间内求解出较长 DWT 的所有数值稳定的 LF，更无法计算最优 LF，这在很大程度上限制了 LF 的实用性. 为此，本文从原理上改进现有算法.

2　多相矩阵的常数 GCD Lifting 分解

正如 DS 和 MA 一样，为了方便理论分析和应用中算法的硬件实现，在正式讨论之前我们对问题本身及其求解施加两个约束，这不会破坏研究的合理性[3,8]：($R1$) $||a(z)| - |b(z)|| \le 1$；($R2$) $|q(z)| \le 1$ 或 $|q(z)| = -\infty$；$|r(z)| < |b(z)|$

2.1　一般 Lifting 分解

由 Laurent 多项式除法的非唯一性[3]知道，满足 $a(z) = b(z)q(z) + r(z)$ 的 $[q(z), r(z)]$ 可能有多解. 我们分析如何计算 LEA (Euclidean algorithm for Laurent polynomial) 每步迭代中的全部解. 与之对应的分解称为一般 LF.

引理 1　给定 $a(z), b(z) \in \mathbf{R}[z, z^{-1}]$，$0 \le |a(z)| - |b(z)| \le 1$，则 $a(z) = b(z)q(z) + r(z)$ 的全部解 $[q(z), r(z)]$ 如下：

(1) $|a(z)| > |b(z)| > 0$ 时：有三个解；(2) $|a(z)| = |b(z)| > 0$ 时：有四个解；(3) $|b(z)| = 0$ 时：有唯一解.

证明　由 $R1$ 约束知 $a(z)$ 与 $b(z)$ 的单项式项数差异至多为 1. 不失一般性，设 $a(z) = \sum_{i=A}^{A+N} a_i z^{-i}$，$b(z) = \sum_{i=B}^{B+N'} b_i z^{-i}$，$q(z) = z^n(c + d/z)$. 其中，$n \in Z$ 用于 $b(z)q(z)$ 的首(尾)项与 $a(z)$ 的首(尾)项对齐；$c, d \in R$ 用于消项以使 $|r(z)| < |b(z)|$.

① $|a(z)| = |b(z)| + 1 > 1$：若令首项对齐，则 $n = B - A$. 在 $R2$ 约束下，有三种消项可能：最前两项、最后两项及首尾两项. 对应的 $[c, d]$ 分别为 $[a_A/b_B, a_{A+1}/b_B - a_A b_{B+1}/(b_B)^2]$，$[-a_{A+N} b_{B+N-2}/(b_{B+N-1})^2 + a_{A+N-1}/b_{B+N-1}, a_{A+N}/b_{B+N-1}]$ 及 $[a_A/b_B, a_{A+N}/b_{B+N-1}]$；若令尾项对齐，仍有 $n = B - A$. 同样有三种消项，但 $[c, d]$ 的取值完全与首项对齐时相同. 因此尽管有六种对齐与消项组合，却只有三个 $[q(z), r(z)]$ 解.

② $|a(z)| = |b(z)| > 0$：若令首项对齐，则 $n = B - A$. 有三种消项可能，$[c, d]$ 分别为 $[a_A/b_B, a_{A+1}/b_B - a_A b_{B+1}/(b_B)^2]$，$[a_{A+N}/b_{B+N}, 0]$ 及 $[a_A/b_B, 0]$；若令尾项对齐，则 $n = B - A + 1$. 同样有三种消项，$[c, d]$ 分别为 $[0, a_A/b_B]$，$[a_{A+N-1}/b_{B+N} - a_{A+N} b_{B+N-1}/(b_{B+N})^2, a_{A+N}/b_{B+N}]$ 及 $[0, a_{A+N}/b_{B+N}]$. 将 n, c, d 代入 $q(z)$ 可发现共有四个解. $|b(z)| = 0$ 时，易证.

定理 1　LF 通常是不唯一的. 特别地，对于互质的 $a(z), b(z) \in \mathbf{R}[z, z^{-1}]$，在 $R1$ 和 $R2$ 约束下，LEA 经过 $|b(z)| + 1$ 步迭代可得其全部 LFs. 形如

$$\begin{bmatrix} a(z) \\ b(z) \end{bmatrix} = \prod_{i=1}^{|b(z)|+1} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \beta z^m \\ 0 \end{bmatrix}, m \in Z \tag{4}$$

证明　这里分析的是多步迭代. 由引理 1 知每步迭代中都可能有多解，所以 LF 通常不唯一；若 $a(z), b(z)$ 互质，其 GCD 必定是一个单项式，由引言中的分析知其 LF 如式 (4) 所示；关于迭代步数，可分三种情形讨论：$|a(z)| > |b(z)|$，$|a(z)| = |b(z)|$，$|a(z)| < |b(z)|$. 限于篇幅，具体分析省略.

LEA 迭代得到的 GCD 只是单项式形式 βz^m，不过尺度矩阵 LF 的如下推广形式 (X 为 K 或 $z^{\pm 1}$) 可将式 (4) 继续分解[8]：

$$\begin{bmatrix} X & 0 \\ 0 & 1/X \end{bmatrix} = \begin{bmatrix} 1 & X - X^2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1/X & 1 \end{bmatrix} \begin{bmatrix} 1 & X-1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 - 1/X \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ X & 1 \end{bmatrix} \begin{bmatrix} 1 & 1/X^2 - 1/X \\ 0 & 1 \end{bmatrix} \tag{5}$$

2.2　常数 GCD Lifting 分解

下面进一步研究在什么条件下式 (4) 中 $m = 0$. 我们称这类分解为常数 GCD LF，可表示为

$$\begin{bmatrix} a(z) \\ b(z) \end{bmatrix} = \prod_{i=1}^{|b(z)|+1} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K \\ 0 \end{bmatrix}, K \text{ 为一常数.} \tag{6}$$

定理 2　(常数 GCD LF 存在的充要条件) 给定互质的 $a(z), b(z) \in \mathbf{R}[z, z^{-1}]$，在约束条件 $R1$ 和 $R2$ 下，LEA 通过迭代能够计算得到常数 GCD 的充要条件是：

(1) $|a(z)| \ne |b(z)|$：若 $|b(z)|$ 为偶数，则所有 $a_{2j+1}(z)$，$b_{2j}(z)$ 有常数项；否则 $a_{2j}(z)$，$b_{2j+1}(z)$ 有常数项. 其中 $j \ge 0$.

(2) $|a(z)| = |b(z)|$：若 $|b(z)|$ 为偶数，则所有 $a_{2j-1}(z)$，$b_{2j}(z)$ 和 $b_0(z)$ 有常数项；否则 $a_{2j}(z)$，$b_{2j-1}(z)$ 有常数项，且 $a_0(z)$ 或有常数项或最高次项为 kz^{-1} 或最低次项为 kz. 其中，$j \in N$.

证明　必要性. 由引理 1 的证明易知，$|a(z)| \ne |b(z)|$ 时，LEA 的每步迭代中 $b_i(z)$ 的所有幂项都来源于 $a_{i-1}(z)$. 因此，在式 (6) 的右部，由 $[a_n(z) = K, b_n(z) = 0]$ 我们可逆序地推得：$b_{n-1}(z) = K$，$a_{n-2}(z) = Kq_n q_{n-1} + K$，… 故常数项 K 在 $a_i(z)$，$b_i(z)$ 之间交替出现. 具体地，按 $|b(z)|$ 的奇偶性分别讨论，便可得 (1) 所述结论. $|a(z)| = |b(z)|$ 时，第一步迭代后才有 $|a_1(z)| > |b_1(z)|$. 因此由前述分析知，从第二步迭代开始 K 在 $a_i(z)$ 和 $b_i(z)$ 之间交替出现. 具体地，$|b(z)|$ 为偶时，$|b_1(z)|$ 为奇，$a_{2j-1}(z)$ 和 $b_{2j}(z)$ 都必含有 K，其中 $j \ge 1$. 尤其是 $a_1(z)$ 含有 K，$b_0(z)$ 中必有 K. $|b(z)|$ 为奇时，$|b_1(z)|$ 为偶，$a_{2j}(z)$，$b_{2j-1}(z)$ 都必有 K，其中 $j \ge 1$. $b_1(z)$ 含有 K，虽然 K 未必是从 $a_0(z)$ 中继承的 (第一步迭代时可产生新幂项)，但由引理 1 的证明知，$a_0(z)$ 仅有三种可能：有常数项、最高次项为 kz^{-1} 或最低次项为 kz.

充分性. $|a(z)| > |b(z)|$ 时，若 $n = |b(z)|$ 为偶且 $b_0(z)$ 含有 K，则 $a_1(z)$ 中必有 K. 由引理 1 的证明易知，根据 K 的位置情况，总可以选择适当的对齐与消项组合使得 $b_2(z) = a_1(z) \% b_1(z)$ 含有 K. 依此类推可确保 $a_{2j+1}(z)$，$b_{2j}(z)$ 都含有 K. 由于 LEA 在第 $n+1$ 步时迭代结束，此时 $|a_{n+1}(z)| = 0$，

$a_{n+1}(z)$ 就是所求的常数 GCD. 其他情形的证明完全类似.

对于给定的 DWT ω,我们可以利用定理 2 来判断 ω 的多相分量 $h_e(z)$, $h_o(z)$ 是否存在常数 GCD. 当 DS 的假设(存在常数 GCD)不成立时,我们可以根据 $|h_e(z)| = |h_o(z)|$ 成立与否和 $|h_o(z)|$ 的奇偶性,将 $h_e(z)$, $h_o(z)$ 做适当平移直至满足定理 2 的条件. 此外,由于定理 2 的证明是构造性的,因此我们实际上已经给出了求解任意(可能较长) DWT 常数 GCD LF 的所有步骤. 限于篇幅,具体算法描述省略.

2.3 Lifting 分解数目的讨论

® 一般分解数,记为 $GdN(h(z))$:当 $|h_o(z)|+1$ 为偶数时,应用引理 1 可得:若 $|h_e(z)| > |h_o(z)| > 0$, $GdN(h(z)) = 3^{|h_o(z)|}$;若 $|h_e(z)| = |h_o(z)| > 0$, $GdN(h(z)) = 4 \times 3^{|h_o(z)|-1}$. 该结论与 MA 结论一致[8]. $|h_o(z)|+1$ 为奇数时有同样的结论.

® 常数 GCD Lifting 分解数,记为 $PdN(h(z))$:由定理 2 的充分性证明可知,某些步迭代时 $q_i(z)$ 的选择与 $a_{i-1}(z)$ 中常数项位置相关联. 特别地,当 $|h_o(z)|+1$ 为偶数、$|h_e(z)| > |h_o(z)|$ 且 $h_e(z)$ 的常数项为首项时, $PdN(h(z)) = 3^{(|h_o(z)|-1)/2}$.

显然 $PdN(h(z)) \le GdN(h(z))$. 尤其是由于应用中常数项常靠近首项,故 $PdN(h(z))$ 通常远小于 $GdN(h(z))$. 这个结论十分重要:一般 LF 数为指数型,当 $h(z)$ 较长时,LF 算法不能在合理时间里求出全部解;但我们可在较短时间里求出所有常数 GCD LFs,这为设计最优 LF 算法打下了坚实基础.

3 DWT 的最优 Lifting 分解

3.1 Lifting 分解的计算复杂性分析

多相矩阵 $\mathrm{P}^0(z)$ 的 LF 可统一表示为:

$$\mathrm{P}^0(z) = \begin{bmatrix} h_e(z) & g_e^0(z) \\ h_o(z) & g_o^0(z) \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} z & 0 \\ 0 & 1/z \end{bmatrix}^m \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \tag{7}$$

其中 $m \in Z$. $m \ne 0$ 时上式含有延迟/超前操作,可利用式(5)将这 $|m|$ 个操作进一步分解为 $4|m|$ 个 Lifting 步. 由此可见,一般 LF 比常数 GCD LF 计算复杂性高. 具体说,上式实现 DWT 时每计算一对通道系数所需代价(加/乘法个数)为

$$LiftingCost = 2 + 4(|h_o(z)|+1) + 2(|g_o(z)| - |h_o(z)|+1) + 4 \cdot |m| * 6 = 2|g_o(z)| + 2|h_o(z)| + 24|m| + 8 \tag{8}$$

标准多相实现算法的计算代价为 $StandardCost = 10 + 4|g_o(z)| + 4|h_o(z)|$. 当分解为常数 GCD LF 时,对于长滤波器 $h(z)$, $g(z)$ 有 $LiftingCost \approx \frac{1}{2} StandardCost$. 但是当分解为一般 LF 时,情况则大不相同. 不难证明, $h_e(z)$ 作为关于 z 的多项式,若令其最高及最低幂项指数分别为 H, L,则有 $L \le m \le H$. $|m|$ 完全可能为非常大的数,这时不再有上述结论,甚至会出现 $LiftingCost > StandardCost$ 的情形. 故同一 DWT 的不同 LF 的计算复杂性差异可能是显著的.

3.2 Lifting 分解的稳定性分析

在对 $D10$ 做 LF 时,MA 发现同一 DWT 的不同 LF 可能会表现出不同的稳定性[8]. 但是他们并没有对此做理论分析,也未提出具体的解决办法.

在矩阵论中已知,关于实矩阵的分解稳定性有结论[9]:

$$\begin{gathered} fl(\mathrm{A}+\mathrm{B}) = (\mathrm{A}+\mathrm{B}) + \mathrm{E}_1,\ fl(\mathrm{CD}) = \mathrm{CD} + \mathrm{E}_2, \\ \|\mathrm{E}_1\|_\infty \le u(\|\mathrm{A}\|_\infty + \|\mathrm{B}\|_\infty), \\ \|\mathrm{E}_2\|_\infty \le nu(\|\mathrm{C}\|_\infty \|\mathrm{D}\|_\infty) + O(u^2), \end{gathered} \tag{9}$$

其中, E_1, E_2 为误差矩阵, $fl(\cdot)$ 为浮点运算模型.

然而此处不能直接引用该结果,因为我们讨论的是多项式矩阵. 不过,如果把 $h_e(z)$ 或 $h_o(z)$ 视为一个"数组",则可在稳定性分析中研究各滤波器系数上的"个体"误差. 为此,将 $h_e(z)$, $h_o(z)$ 的分解表示成如下形式:

$$\begin{aligned} \begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} &= \sum_i \left(\begin{bmatrix} h_{e,i} \\ h_{o,i} \end{bmatrix} z^i \right) = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} Kz^m \\ 0 \end{bmatrix} \\ &= \prod_{i=1}^{n} \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} c_i & 0 \\ 0 & 0 \end{bmatrix} z^{n_i} + \begin{bmatrix} d_i & 0 \\ 0 & 0 \end{bmatrix} z^{n_i - 1} \right) \begin{bmatrix} K \\ 0 \end{bmatrix} z^m \\ &= \prod_{i=1}^{n} (\mathrm{A} + \mathrm{C}_i z^{n_i} + \mathrm{D}_i z^{n_i - 1}) \begin{bmatrix} K \\ 0 \end{bmatrix} z^m \end{aligned} \tag{10}$$

由此可见 $\mathrm{h}_i \triangleq \begin{bmatrix} h_{e,i} \\ h_{o,i} \end{bmatrix} = \sum \left(\prod_{i=1}^{n} \mathrm{M}_i \right) \begin{bmatrix} K \\ 0 \end{bmatrix}$. 其中, $\mathrm{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $\mathrm{C}_i = \begin{bmatrix} c_i & 0 \\ 0 & 0 \end{bmatrix}$, $\mathrm{D}_i = \begin{bmatrix} d_i & 0 \\ 0 & 0 \end{bmatrix}$, $\mathrm{M}_i \in \{\mathrm{A}, \mathrm{C}_i, \mathrm{D}_i\}$.

进而从式(9)可知,LF 产生的滤波器系数的 ℓ_∞ 误差上界可由 $\|\mathrm{A}\|_\infty$, $\|\mathrm{C}_i\|_\infty$, $\|\mathrm{D}_i\|_\infty$ 和 $|K|$ 来度量. 特别地,对于给定的 $h_e(z)$ 和 $h_o(z)$,这些范数及 $|K|$ 的大小越"不平衡",分解的稳定性就越差. 由于 $\|\mathrm{A}\|_\infty = 1$,一个直观的稳定性判据为 $\|\mathrm{C}_i\|_\infty$, $\|\mathrm{D}_i\|_\infty$ 和 $|K|$ 的数值接近于 1 的程度. 该结论与 Maslen 和 Abbott 的观察结果完全一致. 通过刻划这种"不平衡"的程度,我们设计出两种实用的稳定性准则:

(1) Rule1:若 $R_\infty < 10^{S_1}$ 则稳定,否则不稳定. 其中, $R_\infty \triangleq \max(\max_{1 \le i \le n}(\max(|c_i|, |d_i|)), |K|)$.

(2) Rule2:若 $R_p < 10^{S_2}$ 则稳定,否则不稳定. 其中, $R_p \triangleq \left(\sum_{i=1}^{n} (w_{i,1} |c_i|^{p \times Sgn(|c_i|-1)} + w_{i,2} |d_i|^{p \times Sgn(|d_i|-1)}) + |K|^{p \times Sgn(|K|-1)} \right)^{1/p}$, $w_{i,1}, w_{i,2} \ge 0$ 且 $w_{i,1} + w_{i,2} = 1$, $p \in N$, $Sgn(x)$ 为符号函数.

S_1, S_2 的取值与具体应用有关,本文均取为 3.

3.3 DWT 的最优 Lifting 分解

给定 DWT ω,设其全部分解为 $L_D(\omega)$,则对 $\forall l_D \in L_D(\omega)$,我们记 $R_c(l_D)$ 为由式(8)得到的 l_D 所需计算量; $R_s(l_D)$ 为由 R_∞ 或 R_p 计算得到的 l_D 稳定性度量;应用中对 l_D 的其他性能度量统称为 $R_a(l_D)$. 现可构造 LF 评价函数为

$$\Phi(l_D) = w_1 R_c(l_D) + w_2 R_s(l_D) + w_3 R_a(l_D),\ w_1 + w_2 + w_3 = 1$$

我们的目标是求最优分解 $l_{optimal}(\omega) = \arg\min_{l_D \in L_D(\omega)} \Phi(l_D)$.

但是,一方面由 2.3 小节的分析已知"求解任意 DWT 的全部 LF"问题具有指数复杂性,是 NP-难的,不存在一个对任意 DWT 均适用的 LF 算法. 即使在实际应用中给定的 ω 总是有限长度的,但当长度稍大时我们便很难求解其全部 LF. 另一方面,由上一小节分析易知随着滤波器长度的增加极易出现不稳定 LF,因而也就没必要求解出全部 LF. 有效的解决办法是:首先求解出性能较好的分解 $L'_D(\omega)$,然后在 $L'_D(\omega)$ 中极小化 Φ 以求解最优分解 $l_{optimal}$. 为此,通过如下两条途径将 LF 搜索空间 $L_D(\omega)$ 缩小为 $L'_D(\omega)$:(1) 只求解常数 GCD LF.

我们在 2.3 小节已知通常 $PdN(h(z)) \ll GdN(h(z))$;(2) 在 LEA 的每步迭代中预检 LF 的稳定性(使用 Rule1 准则),这样可尽早筛掉稳定性差的 LF.

至此,我们可以给出 DWT 的最优 LF 算法 OLFA (Optimal lifting factorization algorithm):

Step 1:求双带多相矩阵 $P(z)$; $i = 0$; $a_0(z) = P[0][0]$; $b_0(z) = P[1][0]$;用 $a_0(z)$ 和 $b_0(z)$ 初始化解树 STree 的根节点;

Step 2:while $(i < |b_0(z)| + 1)$ do {

对 STree 中第 i 层的各节点调用 PSLEA$(a(z), b(z))$ 并将输出集合中的各元素作为子节点存储;对 STree 的第 $i+1$ 层各节点逐一判断是否稳定,若不稳定则删除之; $i = i + 1$;}

遍历 STree 以获得所有较稳定的常数 GCD LF $L_D'(\omega)$;

Step 3:在集合 $L_D'(\omega)$ 中计算各元素 l_D 的评价值 $\phi(l_D)$ (不须计算 R_c) 并找到评价值最小的元素,记为 $d_{\min}$;调用 EMake$(d_{\min})$ 并将结果记为 $l_{\min}$;计算 Lifting 滤波器,记为 $s(z)$;由 $l_{\min}$ 及 $s(z)$ 生成 ω 多相矩阵的最优 LF,记为 l_{optimal};

对 OLFA 做两点说明:(1) 求解过程用树结构存储,每条由根至叶节点路径上的所有 $q_i(z)$ 和 GCD 构成一个 LF. (2) 须使用子算法:PSLEA 执行常数 GCD LF 中的单步迭代; EMake 确保迭代步数为偶数. 根据 2.2 小节分析我们可设计出 PSLEA; EMake 的设计则由 MA 对奇迭代步数的处理[8]很容易完成.

4　实验结果及分析

本文在 Sony GRX560 Notebook 上实现了 DWT 的最优 LF,设计语言为 Mathematica4.0. 表 1 和表 2 分别给出了双正交 9/7 DWT 及正交 D12 DWT 的最优分解测试过程;此外为说明 OLFA 与现有算法的性能差异,而现有算法又都只是求解一般 LF 集合,本文采取的策略是在相同环境下运行 MA 的 LF 代码(当前最有效的分解算法之一),并与常数 GCD LF 模块作比较,其结果见表 3 的 2～5 列(I和 II分别表示两算法). 最后为方便应用,在表 3 最后一列给出了典型 DWT 的 $P^0(z)$ 最优 LF 结果以供查阅(稳定性度量选为 R_2).

由表 1 知,我们能够求解出 DWT 的全部常数 GCD LF;由表 2 可见,本文算法在滤波器较长的情形下仍能正常工作. 两表中都求解出了相应的最优 LF. 这些测试结果证明了本文算法是正确的.

进一步地,从表 3 第 2～4 列的数据比较中可直观地看到,常数 GCD LF 技术及稳定性检测方法能够极大地缩小分解搜索空间,故本文算法是有效的. 尤其是当滤波器较长时(实验中我们发现从 $D12$ 开始),DS 算法或者极易产生病态方程组的求解而无法正常工作(运行时间记为"—"),或者求解出的某些 LF 的稳定性很差. 具体地,通过表 3 的第 2,3 列数据知, $D12$ 的全部 LF 中只有 7 %较稳定的常数 GCD LF,而 $D20$ 则仅有 0.88 %. 可见 DS 算法在求解长滤波器的分解时或者无效或者效率很低,因而不具有通用性和实用性. 可是,一方面应用中又常会使用长滤波器(长度超过 10),如语音处理等;另一方面 DWT 滤波器越长时,Lifting 实现效率才越高[3]. 所以 LF 的稳定性问题将极大限制 DS 算法实际应用价值,本文很好地解决了这些问题. OLFA 能够应用于长滤波器的 DWT(如 $D20$),能够在合理时间里求解出给定 DWT 的最优分

表 1　双正交 9/7 DWT 的最优 Lifting 分解测试过程

(P[z_] = Polyphase[z])//MatrixForm

$$\begin{pmatrix} \frac{0.852699}{z^2} - \frac{0.110624}{z^3} + \frac{0.0378285}{z^4} + 0.0378285 - \frac{0.110624}{z} & \frac{0.418092}{z^2} - \frac{0.0645389}{z^3} - 0.0645389 + \frac{0.418092}{z} \\ \frac{0.377403}{z^2} - \frac{0.0238495}{z^3} - 0.0238495 + \frac{0.377403}{z} & \frac{0.0406894}{z^2} + 0.0406894 - \frac{0.788486}{z} \end{pmatrix}$$

(All ConstantFacts = EuclideanFactorisation[{P[z][[1,1]], P[z][[2,1]]}, EliminationBranch → AllConstant, Chop]//Length)

$$\{\{-20.4611 - \frac{1.58613}{z}, 0.0432874 - \frac{0.00259985}{z}, 173.6 + \frac{189.576}{z}, \frac{0.157443}{z} - 0.0141967\}, 0.307343\}$$

$$\{\{-20.4611 - \frac{1.58613}{z}, 0.0529801 - \frac{0.00259985}{z}, -56.1473z - 103.171, \frac{0.0134881}{z} + \frac{0.0479337}{z^2}\}, -1.85495\}$$

$$\{\{-20.4611 - \frac{1.58613}{z}, 0.0529801 + \frac{0.0529801}{z}, 3.13767z^2 - 17.9921z, -\frac{0.443507}{z^2} - \frac{0.443507}{z^3}\}, 1.1496\}$$

OptimalFactorisation[AllConstantFacts]//ColumnForm

$$\{\{-20.4611 - \frac{1.58613}{z}, 0.0529801 + \frac{0.0529801}{z}, 3.13767z^2 - 17.9921z, -\frac{0.443507}{z^2} - \frac{0.443507}{z^3}\}, 1.1496\}$$

表 2　正交 D12 DWT 的最优 Lifting 分解测试过程

(P[z_] = Polyphase[z])//MatrixForm

$$\begin{pmatrix} -\frac{0.226265}{z^2} + \frac{0.0975016}{z^3} - \frac{0.031582}{z^4} + \frac{0.00477726}{z^5} + 0.111541 + \frac{0.751134}{z} & 0.0010773z^5 - 0.000553842z^4 - 0.0275229z^3 + 0.129767z^2 - 0.31525z - 0.494624 \\ -\frac{0.129767}{z^2} + \frac{0.0275229}{z^3} + \frac{0.000553842}{z^4} - \frac{0.0010773}{z^5} + 0.494624 + \frac{0.31525}{z} & 0.00477726z^5 - 0.031582z^4 + 0.0975016z^3 - 0.226265z^2 + 0.751134z + 0.111541 \end{pmatrix}$$

(AllConstantFacts = EuclideanFactorisation[{P[z][[1,1]], P[z][[2,1]]}, EliminationBranch → AllConstant, Chop])//Length

23

OptimalFactorisation[AllConstantFacts]//ColumnForm

$$\{\{-4.43447, 0.214593 - \frac{0.0633132}{z}, 9.97002z - 4.49311z^2, \frac{0.17965}{z^2} + \frac{0.400695}{z^3}, 0.00550295z - 0.0430454z^2, -\frac{0.122882}{z^2} - \frac{0.428777}{z^3}\}, 3.0899\}$$

表 3　OLFA 与现有分解算法的比较

DWT	分解数目		运行时间(秒)		最优 Lifting 分解
	I	II	I	II	
$D4$	4	1	1.22	0.24	$\{\{1+\frac{2}{1-\sqrt{3}},\frac{\sqrt{3}}{4}+\frac{-2+\sqrt{3}}{4z}\},\frac{\sqrt{2}}{\sqrt{3}-1}\}$
$D6$	12	4	1.47	0.33	$\{\{2.4255,0.0793395z-0.352388,\frac{0.561415}{z^2}-\frac{2.89535}{z}\},0.43188\}$
$D8$	36	6	1.78	0.48	$\{\{8.61873z-3.10293,\frac{0.479039}{z}-\frac{0.0380879}{z^2},-1.68049z-0.179379,4.94964-\frac{2.09409}{z}\},0.740708\}$
$D10$	108	12	6.86	0.95	$\{\{3.77152,0.0698843z-0.247729,\frac{3.03369}{z^2}-\frac{7.59758}{z},0.0157993z^3-0.0503964z^2,\frac{0.172573}{z^4}-\frac{1.10315}{z^3}\},0.347389\}$
$D12$	324	23	—	1.91	$\{\{-4.43447,0.214593-\frac{0.0633132}{z},9.97002z-4.49311z^2,\frac{0.17965}{z^2}+\frac{0.400695}{z^3},0.00550295z-0.0430454z^2,-\frac{0.122882}{z^2}-\frac{0.428777}{z^3}\},3.0899\}$
$D20$	26244	230	—	23.1	$\{\{-7.05573,0.138938-\frac{0.0440724}{z},18.9141z-10.1944z^2,\frac{0.0581401}{z^2}+\frac{0.0671946}{z^3},0.47338z^3+16.9368z^2,\frac{1.26509}{z^4}-\frac{0.0970869}{z^3},1.23607z^3-0.53554z^4,\frac{5.91717}{z^5}-\frac{2.54919}{z^4},-0.0728087z^6-0.169z^5,-\frac{23.5911}{z^6}-\frac{121.167}{z^7}\},0.542262\}$
(4,2)	9	3	1.62	0.40	$\{\{0,-\frac{z}{4}-\frac{5}{4},1-\frac{3z}{13},\frac{13}{16z}-\frac{169}{48}\},-\frac{6\sqrt{2}}{13}\}$
(9-7)	27	3	1.89	0.54	$\{\{-20.4611-\frac{1.58613}{z},0.0529801+\frac{0.529801}{z},3.13767z^2-17.9921z,-\frac{0.443507}{z^2}-\frac{0.443507}{z^3}\},1.1496\}$

解,而不是全部分解或一个非最优分解.综上所述,本文算法在解的质量、求解时间及适用范围上都有较大改进.

5 结论

DWT 的 Lifting 分解不唯一.现有算法只能求解长度较短 DWT 的一个或全部分解.当求解一个分解时,通常不能保证该分解是某种意义下最优的;而求解全部分解不仅极耗时,且解集合中只有少量分解性能较好.为此本文提出 DWT 的常数 *GCD* Lifting 分解以及分解的性能评价模型,并在此基础上设计了实用的最优分解算法.实验结果表明,本文算法为 DWT 的快速计算提供了一个实用工具.

参考文献:

[1] W Sweldens. The lifting scheme: A construction of second generation wavelets[J]. SIAM Journal on Mathematical Analysis, 1997, 29(2): 511 - 546.

[2] W Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets[J]. Applied and Computational Harmonic Analysis, 1996, 3(2): 186 - 200.

[3] I Daubechies, W Sweldens. Factoring wavelet transforms into lifting steps[J]. Journal of Fourier Analysis and Applications, 1998, 4(3): 247 - 269.

[4] A R Calderbank, I Daubechies, W Sweldens, B L Yeo. Wavelet transforms that map integers to integers[J]. Applied and Computational Harmonic Analysis, 1998, 5: 332 - 369.

[5] G Piella, H J A M Heijmans. Adaptive lifting schemes with perfect reconstruction[J]. IEEE Transactions on Signal Processing, 2002, 50(7): 1620 - 1630.

[6] S Movva, S Srinivasan. A novel architecture for lifting-based discrete wavelet transform for JPEG2000 standard suitable for VLSI implementation[A]. IEEE Proceedings of the 16th International Conference on VLSI Design[C]. New Delhi, India: IEEE Computer Society Press, 2003. 202 - 207.

[7] 钟广军,成礼智,陈火旺.多级多维离散小波变换的快速提升计算[J].电子学报,2001,29(11):1475 - 1477.

[8] M Maslen, P Abbott. Automation of the lifting factorisation of wavelet transforms[J]. Computer Physics Communications, 2000, 127: 309 - 326.

[9] G H Golub, C F Van Loan. Matrix Computations[M]. Third edition, Baltimore, Maryland: Johns Hopkins University Press, 1996.

作者简介:

吴江华　男,1975 年 9 月生于湖北黄冈,1999 年在哈尔滨工业大学获得硕士学位,现在哈尔滨工业大学计算机应用技术专业攻读博士学位,主要研究方向包括小波分析、数据压缩、信号处理和计算机视觉等.

张田文　男,1940 年 3 月生于辽宁大连,教授,博士生导师,主要研究领域为人工智能、图像处理和模式识别等.