

一种不需要乘法的整数小波变换及其 基于行的提升格式

周化雨 关履泰 马建华

(中山大学科学计算与计算机应用系, 广州 510275)

摘要 对提升后的 Donoho 小波进行了进一步研究, 通过正则性的计算和小波采样逼近理论说明 $N = \tilde{N} = 4$ (记为 SWE13/7) 时有较好的逼近性能, 并在采样逼近上优于 D9/7 小波。SWE13/7 的提升系数是简单分数, 从而可以得到在计算过程中仅使用到整数位移和加法的提升格式, 使计算复杂性大幅度低于 D9/7 小波, 而且结果等价于中间过程使用浮点数运算时的结果。基于行的提升格式可以大幅度减少内存使用量, 并在并行计算时提高计算速度, 本文实现了 SWE13/7 的基于行的提升格式, 它的计算量也明显小于 D9/7。数值实验结果说明使用零树编码时, SWE13/7 的图像压缩效果整体上优于 D9/7 效果, 并且计算量小很多。

关键词 提升格式 正则性 整数小波变换

中图分类号: TN919.81 **文献标识码**: A **文章编号**: 1006-8961(2005)08-0977-07

An Integer Wavelets Transform with Multiplierless Operation and Its Line-Based Lifting Scheme

ZHOU Hua-yu, GUAN Lu-tai, MA Jian-hua

(Department of Scientific Computing and Computer Applications, ZhongShan University, Guangzhou 510275)

Abstract This paper does further research for the lifted Donoho wavelets. Through the computing of regularity and the wavelets sampling approximation theory, we explain it has better approximation capability when $N = \tilde{N} = 4$ (denoted as SWE13/7), and it precedes D9/7 wavelets in sampling approximation. The lifting coefficients of SWE13/7 are all simple fractions, so wavelets transform only with integer shift and adder operations can be obtained in computing process, the computing complexity is efficiently lower than D9/7, and the result is equivalent to those using floating number in computing process. The line-based lifting scheme can efficiently decrease the memory usage, and increase the speed at parallel computing case. This paper has realized the line-based lifting scheme of SWE13/7, it has lower complexity than D9/7 in evidence. Numerical experiments indicate that the image compression performance of SWE13/7 precedes that of D9/7 when using EZW coding, and its computing complexity is much less.

Keywords lifting scheme, regularity, integer wavelets transform

1 引言

提升格式既可以用来构造小波, 又可以作为离散小波变换的快速算法, 比 Mallat 算法复杂度小很多, JPEG2000 已经将其纳入规范之中。小波变换应用于图像压缩公认是有很有效的, 但计算量和内存使用量一直是一个需要考虑的问题。降低小波变换的计算量

和内存使用量, 同时提高小波的压缩质量, 在医学图像、遥感图像、视频会议、高清晰电视等领域都是很有意义的。

在文献[1]中 Deslauriers-Dubuc 引入了一种子分划插值格式, 其具体形式详见文献[1]、[2]。它的含义是 $n/2$ 处的值由邻近 N 个点进行拉格朗日 (Lagrange) 插值并重采样得到。

在文献[3]中 Shensa 证明了 Daubechies 的正交

收稿日期: 2004-07-21; 改回日期: 2005-01-05

第一作者简介: 周化雨(1979 ~), 男, 2001年于中山大学科学计算与计算机应用系获学士学位, 现为中山大学科学计算与计算机应用系博士研究生。主要研究方向为小波分析、图像和视频编码。E-mail: zhou_huayu@163.com

小波滤波器与 Deslauriers-Dubuc (简称 DD) 插值小波滤波器之间有关系: $H = h_D * \bar{h}_D$, 其中, h_D 是带 $N/2$ 阶消失矩的 Daubechies 正交小波低通滤波器, $h_D[n] = h_D[-n]$, H 是 N 阶的 DD 插值滤波器。

Donoho 小波滤波器组的定义: $H(Z)$ 为 N 阶的 DD 插值滤波器的 Fourier 变换, 相应的 $\tilde{G}(Z) = Z \overline{H(-Z)}$, $G(Z) = Z$, $\tilde{H}(Z) = 1$ 。 $\tilde{H}(Z)$ 在 $Z = -1$ 处无根, 故对应小波无消失矩, 性质不好, 需要做改进。在文献[4]中 Sweldens 对 Donoho 小波进行了提升, 使提升后的 $G(Z)$ 在 $Z = 1$ 处有 \tilde{N} 重根。

Donoho 小波可以看作是由 Lazy 小波提升而来的 (这里的 $H(1) = 2$, 即是 DD 插值滤波器的元素都乘以 2)。设 Donoho 小波是由 Lazy 小波提升得到的, 即

$$\begin{cases} H(Z) = 1 + Z\tilde{S}(Z^2) \\ \tilde{G}(Z) = Z - \overline{\tilde{S}(Z^2)} \end{cases}$$

得到 $\tilde{S}(Z^2) = (H(Z) - 1)Z^{-1}$, 接着对 Donoho 小波进行提升

$$\begin{cases} \tilde{H}(Z) = 1 + \tilde{G}(Z)\overline{S(Z^2)} \\ G(Z) = Z - H(Z)S(Z^2) \end{cases}$$

令 $D(Z) = Z^{-1}S(Z^2)$, 则

$$Z^{-1}G(Z) = 1 - H(Z)D(Z)$$

为了使 $G(Z)$ 在 $Z = 1$ 处有 \tilde{N} 重零根, 可以取

$$D(Z) = \frac{1}{2}(H^{(\tilde{N})}(Z) - 1)$$

其中, $H^{(\tilde{N})}$ 表示 \tilde{N} 阶的 DD 插值滤波器, 由此得 $S(Z^2) = \frac{Z}{2}(H^{(\tilde{N})}(Z) - 1)$ 。

当 $\tilde{N} > N$ 时, $\tilde{H}(Z)$ 的长度为 $2\tilde{N} + 2N - 3$, 很大, 不太实用, 事实上, $H(Z)$ 对应的尺度函数 φ 较光滑, 适宜作合成尺度 (量化产生的误差合成时被平滑), 所以分解小波的消失矩一般定为 N 。同时分解小波的消失矩一般比合成小波的大 (产生小的小波系数), 故一般选择 $\tilde{N} \leq N$ 。

$N = 4$ 时, 低通合成滤波器的长度为 7, 对应的分解小波消失矩与 D9/7 分解小波消失矩阶数相同, 所以下面讨论 $N = 4$ 时的情形。此时 $\tilde{N} \leq N$, \tilde{N} 有两种选择 $\tilde{N} = 2$ 和 $\tilde{N} = 4$ 。

通过 $\tilde{\varphi}$ 的正则性和采样逼近理论可以说明选择 $\tilde{N} = 4$ 更佳。

2 正则性和采样逼近误差

文献[5]给出一个滤波器确定尺度函数正则性指数的公式。利用它来计算提升后的 Donoho 小波系统中的尺度函数的正则性。由 Daubechies 正交小波滤波器的自相似^[6,7]和 Donoho 小波的提升过程, 得到提升后的 Donoho 小波的合成低通滤波器和分解低通滤波器分别为

$$H^{(N)}(Z) = \left| \frac{1+Z}{2} \right|^N P_N \left(\frac{2-Z-Z^{-1}}{4} \right)$$

$$\tilde{H}^{(N, \tilde{N})}(Z) = \left| \frac{1+Z}{2} \right|^{\tilde{N}} \left[2P_{\tilde{N}} - 2 \left| \frac{1+Z}{2} \right|^N P_N P_{\tilde{N}} + \left| \frac{1+Z}{2} \right|^{N-\tilde{N}} P_N \right]$$

其中, $P_N(y) = \sum_{k=0}^{N-1} \binom{N-1+k}{k} y^k$ 。

计算 $\tilde{\varphi}$ 的正则性指数, 当 $N = 4, \tilde{N} = 4$ 时, $\sigma = 1.1794$ 。当 $N = 4, \tilde{N} = 2$ 时, $\sigma = 0.5932$ 。可见, $N = 4, \tilde{N} = 4$ 比 $N = 4, \tilde{N} = 2$ 对应的 $\tilde{\varphi}$ 的正则性要好得多。

提升后的 Donoho 小波的特点是合成尺度函数和分解尺度函数分别有 N 和 \tilde{N} 阶消失矩。尺度函数和小波函数的消失矩之间的理想分布是, 它们有相同消失矩, 此时采样逼近误差界估计为 $\|f(x) - S^j(f)\|_{L^2}^2 \leq C2^{-j(N+1)}$ ^[8], 即误差界衰减较快, 采样逼近性能好。

当尺度函数和小波函数有相同阶的消失矩时, 称之为双正交 Coifman 小波系统 (BCWS)^[8], 当 N 为偶数时 BCWS 等同于 $N = \tilde{N}$ 时提升后的 Donoho 小波。这也说明 $N = 4, \tilde{N} = 4$ 比 $N = 4, \tilde{N} = 2$ 时的小波逼近性能要好。记 $N = 4, \tilde{N} = 4$ 时的经过提升的 Donoho 小波为 SWE13/7 (分解低通长度为 13, 合成低通长度为 7)。

采样逼近性好的优点是可以直接把信号原始采样值作为最细尺度上的系数, 这样可以使误差足够小。这对图像压缩尤为重要, 因为一般处理都是直接把图像像素值作为最细尺度上的系数, 即以采样逼近代替理论上的投影逼近^[6]。

3 不带乘法的整数小波变换

提升格式

$$\begin{cases} d^{(i)}[n] = d^{(i-1)}[n] - \left[\sum_k \tilde{s}[k] c^{(i-1)}[n-k] + \frac{1}{2} \right] \\ c^{(i)}[n] = c^{(i-1)}[n] - \left[\sum_k \tilde{s}[k] d^{(i-1)}[n-k] + \frac{1}{2} \right] \end{cases}$$

是可将整数映射到整数的可逆小波变换,一般内部计算过程中要使用浮点数运算,取整会引起小波变换的非线性,但不影响它的完全重构性质^[9]。

提升后的 Donoho 小波也有上述的提升格式。低通部分和高通部分可以由下式得到

$$\begin{aligned} [lp(Z) \quad hp(Z)] &= [X_e(Z) \quad X_o(Z)] \begin{bmatrix} \tilde{H}_e(Z) & \tilde{C}_e(Z) \\ \tilde{H}_o(Z) & \tilde{C}_o(Z) \end{bmatrix} \\ &= [X_e(Z) \quad X_o(Z)] \times \\ &\quad \begin{bmatrix} 1 & -\tilde{S}(Z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \tilde{S}(Z) & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \end{aligned}$$

对于 SWE13/7,它的提升格式是

$$\begin{aligned} s_l^{(0)} &= x_{2l}, d_l^{(0)} = x_{2l+1} \\ d_l &= d_l^{(0)} - (-s_{l-1}^{(0)} + 9s_l^{(0)} + 9s_{l+1}^{(0)} - s_{l+2}^{(0)})/16 \\ s_l &= s_l^{(0)} + (-d_{l-2} + 9d_{l-1} + 9d_l - d_{l+1})/32 \\ d_l &= \sqrt{2}d_l, s_l = \sqrt{2}s_l \end{aligned}$$

而 D9/7 小波的提升格式是

$$\begin{aligned} s_l^{(0)} &= x_{2l}, d_l^{(0)} = x_{2l+1} \\ d_l^{(1)} &= d_l^{(0)} + \alpha(s_l^{(0)} + s_{l+1}^{(0)}) \\ s_l^{(1)} &= s_l^{(0)} + \beta(d_l^{(1)} + d_{l-1}^{(1)}) \\ d_l^{(2)} &= d_l^{(1)} + \gamma(s_l^{(1)} + s_{l+1}^{(1)}) \\ s_l^{(2)} &= s_l^{(1)} + \delta(d_l^{(2)} + d_{l-1}^{(2)}) \\ s_l &= \zeta s_l^{(2)}, d_l = d_l^{(2)}/\zeta \end{aligned}$$

其中, $\alpha = -1.586134342, \beta = -0.05298011854, \gamma = 0.8829110762, \delta = 0.4435068522, \zeta = 1.149604398$ 它们需要的乘法和加法量一样多。

在 JPEG2000 的开源项目(如 jasper, 可从 <http://www.ece.uvic.ca/~mdadams/jasper> 下载)中,对浮点数乘法的解决是将其换为定点小数乘法,实际上就是有限精度的方法。

文献[10]中比较了 D9/7 和 SWE13/7 小波在有限精度下的变换应用于图像零树编码时的效果,发现在进行整数到整数的提升格式小波变换时, SWE13/7 小波保留 4 或 6 位尾数与 D9/7 小波保留 10 或 12 位尾数的效果相当。实际上,在有限精度浮点数运算下,对于 SWE13/7 小波保留 5 位尾数就相当于是无限精度,只是最后的规范化常数 $\sqrt{2}$ 被有限

精度所影响。

另外 $\tilde{S}(Z), S(Z)$ 的系数都是分母为 2 的幂次的有理数,如果不考虑最后乘上的 $\sqrt{2}$,上述运算可以转换成只需要位移和加法的整数运算。通常使用 Mallat 快速算法时,使用的是“原始”滤波器系数,即 $H(1) = \sqrt{2}, \tilde{H} = \sqrt{2}$ 。文献[11]中指出用零树编码时,应该使用“原始”滤波器系数进行变换,否则性能很差,也就是说乘以 $\sqrt{2}$ 不能省略。

文献[12]解决 $\sqrt{2}$ 带来的浮点数运算的方法是每一次 1 维变换(行或列),暂时不乘以 $\sqrt{2}$,但在每一级 2 维变换完成后,对 4 个子带的信号进行两次乘以 $\sqrt{2}$,即乘以 2 的处理。

由上面的论述,得到只需位移和加法的整数算法(一个水平):

(1) 设 P, U 为两个位移加法处理单元:

$$\begin{aligned} P(a, b, c, d) &= (-a + b + c - d + b \ll 3 + c \ll 3 + 8) \gg 4 \\ U(a, b, c, d) &= (-a + b + c - d + b \ll 3 + c \ll 3 + 16) \gg 5 \end{aligned}$$

其中, \ll 是左位移符号, \gg 是右位移符号。

则

$$\begin{aligned} s_l &= x_{2l}, d_l = x_{2l+1} \\ d_l - &= P(s_{l-1}, s_l, s_{l+1}, s_{l+2}) \\ s_l + &= U(d_{l-2}, d_{l-1}, d_l, d_{l+1}) \end{aligned}$$

用上式进行行变换。

(2) 再用上式进行列变换,并且规范化 $d_l = d_l \ll 1, s_l = s_l \ll 1$ 。

这样得到的结果与整数到整数的提升格式的结果一致,但完全不需要浮点数运算也不需要乘法运算。对比 D9/7 提升格式的浮点数运算,它的运算量要小得多。

4 基于行的提升格式

基于行的提升格式使用所谓的“压缩格式”,主要思想就是内存中只保留当前计算和下一阶段计算需要的信号,产生最终结果就及时输出。

文献[13]中已经实现了 D9/7 的基于行的提升格式,下面实现 SWE13/7 的基于行的提升格式。

4.1 1 维结构

提升格式基本处理元素是一个多项加法计算器,对于 SWE13/7 是 5 项输入的,即

$$Output = R_n + C_1 * (R_0 + R_3) + C_2 * (R_1 + R_2)$$

其中, R_0, R_1, R_2, R_3 为信号变量, $R_n (R_l)$ 为当前

高通(低通)信号变量, $R_{i0}, R_{i1}, R_{i2}, R_{i3}$ 为临时变量。如图 1 所示,对于 1 维正变换,每一层结果都由上一层的 4 个信号和当前信号所决定。每一阶段有 1 个新的信号输入,算法将逐个阶段进行,直到输出所有的分解信号。做完一行的分解,结果将送入行缓存,供下一步的垂直滤波使用。图 2 为 1 维逆变换示意图。

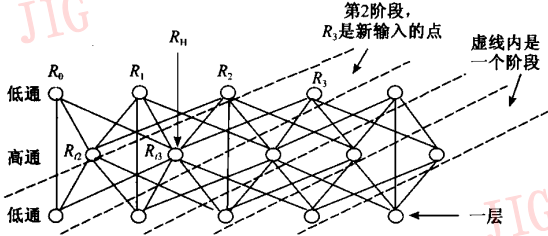


图 1 1D_DWT(正变换)示意图
Fig. 1 1D_DWT(Forward transform)

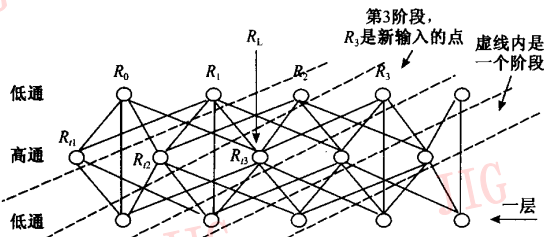


图 2 1D_IDWT(逆变换)示意图
Fig. 2 1D_IDWT(Inverse transform)

为了解释清楚这个过程,附上部分 C 代码(边界采用偶延拓,即超出部分以偶对称点代替)。

预测子过程,其中, R_H 为高通, R_0, R_1, R_2, R_3 为低通, $type$ 为 1 是正变换, 否则是逆变换。

```
SWE_13_7_P(int * Ri, int R0, int R1, int RH, int R2, int R3, int type)
```

```
{
//边界处理, N 是空值标记
if (R0 - N == 0) R0 = R1;
if (R0 - N == 0 && R1 - N == 0)
{R0 = R3; R1 = R2; }
if (R3 - N == 0) R3 = R2;
if (R3 - N == 0 && R2 - N == 0)
{R2 = R1; R3 = R0; }
//P 过程
R0 = -R0 + R1 + R2 - R3;
R1 = R1 << 3;
R2 = R2 << 3;
R0 = (R0 + R1 + R2 + 8) >> 4;
if (type == 1) *Ri = RH - R0;
```

```
else *Ri = RH + R0;
}
//更新子过程,其中, RL 为低通, R0, R1, R2, R3 为高通, type 为 1 是正变换, 否则是逆变换
SWE_13_7_U(int * Ri, int R0, int R1, int RL, int R2, int R3, int type)
{
略 //类似 SWE_13_7_P 的过程
}
//1 维正变换。输入信号为 inSig, 输出低通信号为 lowSig[lowLen], 输出高通信号为 highSig[highLen]
1D_DWT(int * inSig, int * lowSig, int lowLen, int * highSig, int highLen)
{
//初始化
lk = 0; hk = 0; ikl = 0; ikh = 1;
//初始化信号变量, N 是空值标记
R0 = N; R1 = inSig[ikl]; ikl += 2;
R2 = inSig[ikl]; ikl += 2; R3 = inSig[ikl]; ikl += 2;
//输入第一个高通信号变量
RH = inSig[ikh]; ikh += 2;
//计算第一个高通信号 opH, 并输出到
SWE_13_7_P(&opH, R0, R1, RH, R2, R3, 1);
highSig[hk++] = opH;
//初始化临时变量
Ri0 = N; Ri1 = N; Ri2 = N; Ri3 = opH;
//循环过程, 计算出所有高通为止
while(hk < highLen)
{
//信号变量左轮换, 空出 R3, 输入信号至 R3
R0 = R1; R1 = R2; R2 = R3;
if(ikl < inLen)
{R3 = inSig[ikl]; ikl += 2;}
else R3 = N;
//输入当前高通信号变量
RH = inSig[ikh]; ikh += 2;
//计算高通信号 opH, 并输出到
SWE_13_7_P(&opH, R0, R1, RH, R2, R3, 1);
highSig[hk++] = opH;
//临时变量左轮换, 空出 Ri3, 输入高通信号至 Ri3
Ri0 = Ri1; Ri1 = Ri2; Ri2 = Ri3; Ri3 = opH;
//计算低通信号 opL, 并输出到
SWE_13_7_U(&opL, Ri0, Ri1, R0, R2, Ri3, 1);
lowSig[lk++] = opL;
}
//至少还有一个低通未计算
//临时变量左轮换, 空出 Ri3, 因为已没有新的高通信号,
```

加入空值标记

$R_{10} = R_{11}; R_{11} = R_{12}; R_{12} = R_{13}; R_{13} = N;$

$SWE_13_7_U(\&opL, R_{10}, R_{11}, R_{12}, R_{13}, 1);$

$lowSig[lk + +] = opL;$

//如果信号长度为奇数,则还有一个低通

$if(inLen \% 2 == 1)$

{ $R_{10} = R_{11}; R_{11} = R_{12}; R_{12} = R_{13}; R_{13} = N;$

$SWE_13_7_U(\&opL, R_{10}, R_{11}, R_{12}, R_{13}, 1);$

$lowSig[lk + +] = opL;$

}

}

//1 维逆变换。输入低通信号为 $lowSig[lowLen]$,输入高通信号为 $highSig[highLen]$,输出信号为 $outSig$

$1D_IDWT(int * outSig, int * lowSig, int lowLen,$

$int * highSig, int highLen)$

{

略 //类似于 1D_DWT 的过程,并参考图 2

}

对比文献[12]中的算法,在每一个阶段 D9/7 需要 8 次乘法,8 次加法,而 SWE13/7 只需要 6 次移位,14 次加法,计算量要小很多。

4.2 2 维结构

不同于 1 维的情形,此时压缩格式里的信号变量变成信号行,临时变量变成临时行,垂直滤波的“信号”是从水平滤波时的行缓存里得到的。

具体算法类似于 1 维的情形,不过 $R_0, R_1, R_2, R_3, R_H(R_L), R_{10}, R_{11}, R_{12}, R_{13}$ 都要换成指向行缓存的指针,子过程 $SWE_13_7_P, SWE_13_7_U$ 应变成对输入的行进行垂直滤波的过程,输入输出换成整行的输入输出。

例如正变换时(如图 3 所示),需要 4 个信号行来存储水平滤波时得到的水平低通或水平高通信号,1 个信号行来存储当前垂直高通信号,4 个临时行来存储垂直滤波得到的垂直高通信号。这比文献[13]中 D9/7 需要的 3 个信号行和 3 个临时行要耗费大,但是是可以接受的。在每一个阶段,1 个水平

信号行被消耗同时加入新的水平信号行,1 行新的低通输出,1 个临时行被消耗同时加入新的临时行,总的来说有 3 行进行了数据更新,而 D9/7 在一个阶段总的有 5 行进行了数据更新。可见 SWE13/7 相比 D9/7 的数据更新量较小。值得注意的是,所有经过垂直滤波后的信号都要左移 1 位(乘以 2)。

同时可以通过这种算法来实现高速并行的硬件结构,如图 4 所示。

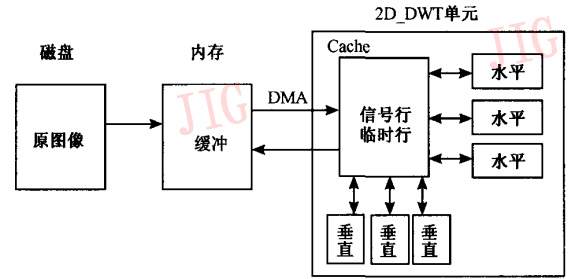


图 4 基于行的提升格式的并行硬件结构

Fig. 4 Line-based lifting scheme parallel hardware structure

5 实验结果

对 512×512 的 8bpp 的 Lena 图像进行图像压缩的数值实验,分别使用 D9/7 和 SWE13/7 小波,编码量化方式使用 EZW(嵌入零树编码)。程序中用了 6 个水平的小波分解。实验结果如表 1 ~ 表 4,图 5 ~ 图 8 所示。

表 1 D9/7 小波,浮点数变换的压缩效果

Tab.1 D9/7 wavelets, float point transform

	比特率 (bpp)			
	0.125	0.25	0.5	1.0
PSNR (dB)	30.4066	33.3431	36.4226	39.5816
MSE (dB)	59.2130	30.1141	14.8191	7.1601

表 2 D9/7 小波,整数到整数变换的压缩效果

Tab.2 D9/7 wavelets, integer to integer transform

	比特率 (bpp)			
	0.125	0.25	0.5	1.0
PSNR (dB)	30.3265	33.1876	36.0977	38.8580
MSE (dB)	60.3154	31.2119	15.9703	8.4583

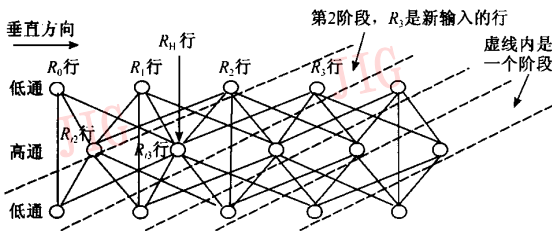


图 3 2D_DWT(正变换)示意图

Fig. 3 2D_DWT(Forward Transform)

表 3 SWE13/7 小波, 浮点数变换的压缩效果

Tab.3 SWE13/7 wavelets, float point transform

	比特率 (bpp)			
	0.125	0.25	0.5	1.0
PSNR (dB)	30.564 1	33.487 0	36.467 6	39.685 5
MSE (dB)	57.104 9	29.132 9	14.666 4	6.990 9



图 5 0.125bpp 下 D9/7 浮点数变换

Fig.5 D9/7 float point transform in 0.125bpp

表 4 SWE13/7 小波, 整数变换 (加法位移) 的压缩效果

Tab.4 SWE13/7 wavelets, integer transform (adder-shift only)

	比特率 (bpp)			
	0.125	0.25	0.5	1.0
PSNR (dB)	30.570 8	33.484 8	36.442 2	39.552 0
MSE (dB)	57.016 3	29.147 5	14.752 2	7.209 1



图 6 0.125bpp 下 D9/7 整数到整数变换

Fig.6 D9/7 integer to integer transform in 0.125bpp



图 7 0.125bpp 下 SWE13/7 浮点数变换

Fig.7 SWE13/7 float point transform in 0.125bpp



图 8 0.125bpp 下 SWE13/7 整数变换

Fig.8 SWE13/7 integer transform in 0.125bpp

6 结 论

无论浮点数, 还是整数变换 (D9/7 只能实现整

数到整数变换), EZW 编码时 SWE13/7 的性能都超过了 D9/7。浮点数变换的优势主要来源于 SWE13/7 在采样逼近和有限精度运算上较优。而在整数变换上, 由于 SWE13/7 是可逆变化, 所以效

果好(当然也有取整带来的一些非线性因素)。D9/7的整数到整数变换比D9/7的浮点数变换效果差很多,这说明有限精度下D9/7的性能急剧下降。SWE13/7进行整数变换时,在0.125比特率上,甚至超过了SWE13/7的浮点数变换,这是因为左移舍位(取整)带来了非线性的因素。在计算量方面,SWE13/7的不带乘法提升格式的计算量比D9/7的带浮点数乘法的提升格式要小得多,其基于行的提升格式的计算量也小得多。

参考文献 (References)

- Deslauriers G, Dubuc S. Symmetric Iterative interpolation processes [J]. *Constr. Approx.*, 1989, 5(1):49~68.
- Sweldens W, Schröder P. Building your own wavelets at home [A]. In: *Proceedings of ACM SIGGRAPH Wavelets in Computer Graphics* [C], Course notes, France, New Orleans, 1996: 15~37.
- Shensa Mark J. The discrete wavelet transform: wedding the a trous and mallat algorithms [J]. *IEEE Transactions on Signal Processing*, 1992, 40(10): 2464~2482.
- Sweldens Wim. The lifting scheme: A custom-design construction of biorthogonal wavelets [A]. In: *Proceedings of SPIE* [C], San Diego, CA, USA, 1995, 2569:68~79.
- Zhou Xian-bo, Lin Wei. About some problems on wavelet regularity [J]. *Acta Mathematica Sinica*, 2002, 45(6): 1069~1078. [周先波, 林伟. 关于小波正则性的几个问题 [J]. *数学学报*, 2002, 45(6): 1069~1078.]
- Mallat Stephane. *A Wavelet Tour of Signal Processing* (2nd ed) [M]. New York: Academic Press, 1999.
- Daubechies Ingrid. *Ten Lectures on Wavelets* [M]. Philadelphia, PA: SIAM, 1992.
- Tian J, Wells R O Jr, Odegard J E, et al. Coifman wavelet systems: approximation, smoothness, and computational algorithms [A]. In: *Computational Science for the 21st Century* [C], Tours, France, 1997.
- Calderbank A R, Daubechies I, Sweldens W, et al. Wavelet transform that map integers to integers [J]. *Applied and Computational Harmonic Analysis*, 1998, 5(3):332~369.
- Grangetto M, Magli E, Olmo G. Finite precision wavelets for image coding: lossy and lossless compression performance evaluation [A]. In: *ICIP00-IEEE International Conference on Image Processing* [C], Vancouver, Canada, 2000, 3: 206~209.
- Kim H, Li C C. Lossless and lossy image compression using biorthogonal wavelet transforms with multiplierless operations [J]. *IEEE Transactions on Circuit and Systems II: Analog and Digital Signal Processing*, 1998, 45(8):1113~1118.
- Qiao Shi-jie, Zhi Gui-lian. A fast wavelet transform image coding [J]. *Journal of Image and Graphics*, 2001, 6(5):434~438. [乔世杰, 智贵连. 一种快速的小波变换图象编码算法 [J]. *中国图象图形学报*, 2001, 6(5):434~438.]
- Chang Wei-Hsin, Lee Yew-San, Pen Wen-Shiaw, et al. A line-based memory efficient and programmable architecture for 2D DWT using lifting scheme [A]. In: *IEEE International Symposium on Circuits and Systems (ISCAS) 2001* [C], Sydney, Australia, 2001: 330~333.