

如下两点改进：(1) 当低通滤波器通过边缘时，位于边缘后的每个输入值在输入低通滤波器之前都减去阶梯的高度 h ；(2) 在滤波器的中心抽头通过边缘之后，对每一个在边缘之前的输入值都要加上阶梯高度 h 。于是得到基于边缘特征小波变换系数计算公式。

$$\begin{aligned}
 c_0 &= f_3x_0 + (f_2 + f_4)x_1 + (f_1 + f_5)x_2 + (f_0 + f_6)x_3 \\
 c_1 &= g_1x_0 + (g_0 + g_2)x_1 + g_3x_2 + g_4x_3 \\
 c_2 &= f_1x_0 + (f_0 + f_2)x_1 + f_3x_2 + f_4x_3 + f_5x_4 + f_6x_5 \\
 &\vdots \\
 c_{i-3} &= f_0x_{i-6} + f_1x_{i-5} + f_2x_{i-4} + f_3x_{i-3} \\
 &\quad + f_4x_{i-2} + f_5x_{i-1} + f_6x_i - hf_6 \\
 c_{i-2} &= g_0x_{i-4} + g_1x_{i-3} + g_2x_{i-2} + g_3x_{i-1} + g_4x_i - hg_4 \\
 c_{i-1} &= f_0x_{i-4} + f_1x_{i-3} + f_2x_{i-2} + f_3x_{i-1} \\
 &\quad + f_4x_i + f_5x_{i+1} + f_6x_{i+2} - h(f_4 + f_5 + f_6) \\
 c_i &= g_0x_{i-2} + g_1x_{i-1} + g_2x_i + g_3x_{i+1} + g_4x_{i+2} \\
 &\quad + h(g_0 + g_1) \\
 c_{i+1} &= f_0x_{i-2} + f_1x_{i-1} + f_2x_i + f_3x_{i+1} + f_4x_{i+2} \\
 &\quad + f_5x_{i+3} + f_6x_{i+4} + h(f_0 + f_1) \\
 c_{i+2} &= g_0x_i + g_1x_{i+1} + g_2x_{i+2} + g_3x_{i+3} + g_4x_{i+4} \\
 c_{i+3} &= f_0x_i + f_1x_{i+1} + f_2x_{i+2} + f_3x_{i+3} + f_4x_{i+4} \\
 &\quad + f_5x_{i+5} + f_6x_{i+6} \\
 &\vdots \\
 c_{N-1} &= (g_0 + g_4)x_{N-3} + (g_1 + g_3)x_{N-2} + g_2x_{N-1}
 \end{aligned} \tag{2}$$

式(2)表示成矩阵形式(略)

2.2 改进方法和标准方法正变换结果比较

本文在 Matlab6.5 平台上进行了标准小波变换和基于边缘特征的改进小波变换试验比较,变换系数结果如图 2 所示。其中子图 1 是输入信号 x , 输入信号包含几个理想的阶梯函数(表示图像列向量的边缘点)。子图 2 为用 Daubechies4 (db4) 进行标准一级一维小波变换的结果, 当对此阶梯函数低通滤波时, 低通滤波器的平均特性使得阶梯函数变成跨越低通子带的几个系数, 变得模糊不清。相似地, 当对此阶梯函数进行高通滤波时, 它会在变换的高通子带引起几个尖峰值。在压缩时, 如果这些尖峰被粗量化或者置 0, 在低比特传输中, 此阶梯函数就会严重失真, 相应的逆变换比较结果将在本文后面给出。子图 3 是基于边缘提取小波变换的结果, 在这种变换方法中, 正变换和逆变换都要求预先知道理想阶梯(或边缘)的位置。

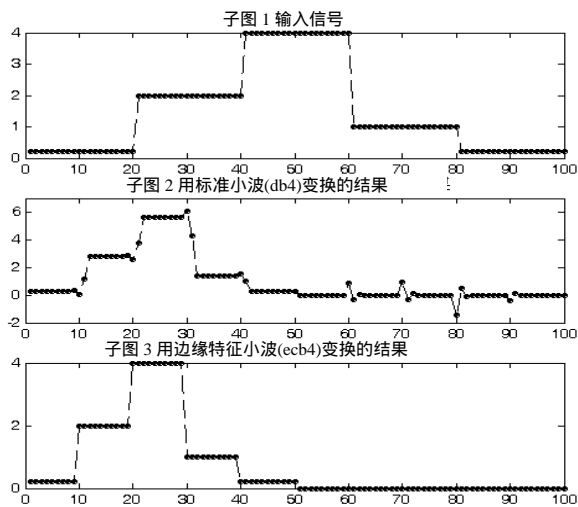


图 2 改进小波方法和标准小波变换结果比较

在基于边缘特征的小波变换中, 低通滤波器在每个位置可以看到的输入值都是已移除阶梯的信息。利用这样的低通滤波器意味着输入信号中的理想阶梯会被复制, 并且不会在低通子带中变得模糊不清。而高通滤波器从不会遇到阶梯(或

边缘), 因此高通子带中就不再存在由阶梯(或边缘)引起的尖峰值。这样即使在每个子带中的小波系数被粗量化, 阶梯(或边缘)会被无影响地保留。

2.3 逆变换

基于边缘特征的小波变换是一个线性过程, 并能用 $F_e \cdot x = c$ 形式的系统方程描述。其中: c 是小波系数向量组(在进行低通和高通重排之前的); F_e 是一个 $N \times N$ 的矩阵, 此矩阵是小波变换系数矩阵 F , 加上一个 $N \times N$ 的边缘移除矩阵 E 构成。因此利用矩阵逆变换 $\hat{x} = F_e^{-1} \cdot c$ 重建原始输入向量 x 是可行的。虽然这样改进小波逆变换会比标准小波逆变换计算复杂度高一些, 但是在单机计算能力迅速提高而网络传输带宽成为各种应用瓶颈的今天, 算法增加一些计算量来换取较好的压缩率和解压缩质量是可行的。

通过上面提出的正变换公式可知, 如果边缘抽取矩阵为 E , 则正变换可以表示为 $(F+E) \cdot x = c$ 。由此一级小波逆变换可以表示为 $\hat{x} = (F+E)^{-1} \cdot c$ 。如果将所有边缘的影响都考虑在内, 假设共有 S 条边缘, 计算公式为: $(F+E_0+E_1+\dots+E_{S-1}) \cdot x = c$ 和 $\hat{x} = (F+E_0+E_1+\dots+E_{S-1})^{-1} \cdot c$ 。

需要注意的是, 构造的矩阵 E 和基本变换矩阵 F 一样必须使滤波器抽头(taps)在矩阵边界附近适当延展。

2.4 改进方法和标准方法逆变换结果比较

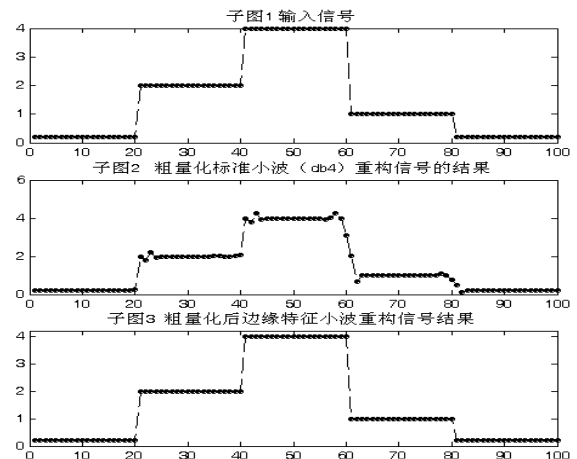


图 3 粗量化后重构(逆变换)的结果比较

图 3 是对小波变换系数粗量化后(阈值 $\text{threshold} = 0.5$), 用小波逆变换对输入信号进行重构的比较结果, 并且阈值 $\text{threshold} = 0.25, 0.125$ 结果与图 3 所示十分相似。图 4 为变换系数高频子带置零后利用小波逆变换重构的比较结果。

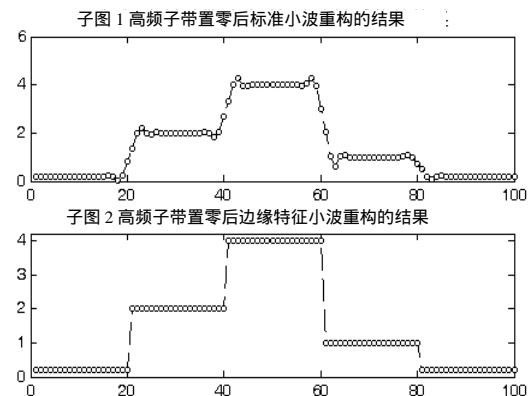


图 4 高频子带置零后重构(逆变换)的比较结果

由图可知, 无论是对高频子带置 0 还是粗量化基于边缘特征的小波变换方法, 在同等条件下都能得到比标准小波变

换更好的重构结果,并且能最大限度保留边缘信息,这正是所希望的。在改进方法中阶梯高度已经被保留在低通子带中(尺度由低通子带获得),但采样边缘的准确位置只包含在分离的边缘信息之中,而不包含在转换系数中,边缘位置信息已经被从高通子带系数中移除。

2.5 计算复杂度分析

基于边缘特征的小波正变换的计算复杂度与标准小波变换在同一个数量级,对一个长度为 N 的一维信号来说,计算复杂度都为 $O(N^2)$ 。在这里主要讨论基于边缘特征的小波逆变换的计算复杂度,根据上面的论述,逆变换中为了计算出每一个长度为 N 的信号都需求出一个 $N \times N$ 矩阵的逆矩阵。一般来说求一个 $N \times N$ 矩阵的逆矩阵的计算复杂度为 $O(N^3)$,在本方法中总的变换矩阵 F_e 的逆矩阵的计算复杂度不大于 $O(N^3)$,因为 F 是一个恒定值,而每一个 E_i 只包含两个非 0 列。在一个长度为 N 的给定的输入信号上,如果边的数量相对 N 来说要小得多,可以用 Sherman-Morrison 公式^[6]来消减计算复杂度。另一种能消减计算复杂度的方法是只对输入信号被边缘影响的部分使用矩阵逆变换。也就是说,仅仅 F_e 中包含边缘的子矩阵需要求逆。矩阵 F_e 中仅仅包含边缘的位置信息而不包含边缘高度信息,所以 F_e 的逆变换对于任何一个有相同边缘位置的信号都是有效的。对于公共边对应的 F_e 的子矩阵的逆变换可以预计算。利用以上一些技巧,计算复杂度能够被进一步消减。

3 关键技术之二:边缘编码技术

3.1 改进的斐波纳契数列压缩编码

由关键技术一中的论述可知,基于边缘特征的小波变换无论正变换还是逆变换都要求已知边缘位置的准确信息。在本文提出的改进小波变换方法在实际应用中先用边缘抽取算子从输入信号(或图像)中抽取边缘,然后用适当的方法对抽取出的边缘信息进行编码,并将编码结果存放在压缩文件的头文件中,以便在低比特率传输中优先传输边缘编码。这样无论正变换还是逆变换时,都可以从头文件中得到边缘的准确位置。改进方法的以上特点将有利于快速浏览和基于内容的检索和识别的实现,并极大地减少了高通子带的能量从而有利于改进压缩效率和压缩性能。本文给出了一种高效的边缘信息压缩编码方法,即改进的利用斐波纳契数列(一种整数数列,其中每个数等于前面两数之和)空间的多环链编码。这种编码方法的优点是:(1)实现起来相对比较简单;(2)可以实现计算开销和边缘信息失真之间的折中。

图 5 为用 Matlab6.5 平台生成的斐波纳契数列空间多环链网格结构示意图。图中给定一个基本单位长度 l ,同时定义一个同心正方环结构,这些同心环结构的半径是 l 的斐波纳契数列倍(1,2,3,5,8,13 倍)。网格点沿着每个同心正方环以间隔 l 排列,共产生 256 个网格点。输入曲线包含一个 $X = \{x_0, x_1, \dots, x_N\}$ 点序列,最终它们被输出点的有序序列 $Y = \{y_0, y_1, \dots, y_M\}$ 所代替。在计算过程中的每一步,多环结构的原点是前一个已编码的输出点 y_{j-1} ,并按从里到外的顺序计算输入曲线和每个环的交点,其中到交点最近的网格点被选为下一个候补输出点 y_{cand} 。如果位于 y_{i-1} 和 y_{cand} 之间的 X 上的点到线段 $y_{j-1}y_{cand}$ 的距离都小于单位长度 l ,则 y_{cand} 就成为下一个输出点 y_j ,与此同时编码器输出 y_j 的多环索引值。以上方法可以近似无损的编码输入曲线,此方法中在输入曲线上所有点与输出曲线之间的距离都小于 l 。输出曲线上点的

最大间距为 $13\sqrt{2}l$,因此通过选择 l 的大小可以在输出曲线的准确度和输出曲线需要传送的点的数目(环索引的数目)之间进行折衷,即在计算精度和计算复杂度之间进行折中。

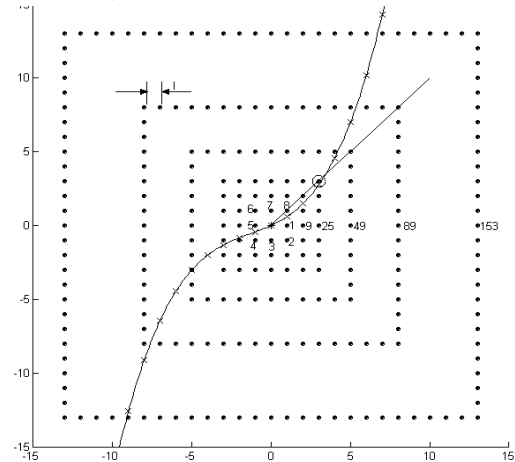


图 5 斐波纳契数列空间多环链网格结构示意图

3.2 边缘编码试验结果分析

作者对本文提出的边缘曲线编码方法进行了试验研究。首先用“canny”算子对 256×256 的 saturn 图像进行边缘抽取,并用识别算法对边缘进行线性识别,取得到的最长的 8 条边缘曲线进行编码。试验中取 $l = 2$ (pixels),在具体实现中,一般选取 l 等于数倍像素。编解码结果表 1 和表 2。

表 1 改进的斐波纳契数列压缩编码数据

编码的边缘曲线名	编码前曲线边缘点数目	编码后索引数目	编码后所需空间 (bit)	每个边缘点所需位数 (bit)
Line1	827	56	480	0.58
Line2	306	23	216	0.70
Line3	130	5	72	0.55
Line4	126	6	80	0.63
Line5	116	5	72	0.62
Line6	96	4	64	0.67
Line7	93	5	72	0.77
Line8	89	4	64	0.72
总计	1 783	108	1120	0.63

表 2 改进的斐波纳契数列压缩解码数据

编码的边缘曲线名	解码后曲线边缘点数目	解码后找到重合点的原边缘点数目	解码后找到邻接点原边缘点的数目	重合点和邻接点所占百分比 (%)
Line1	819	369	439	97.7
Line2	304	155	149	99.4
Line3	130	90	40	100
Line4	126	81	45	100
Line5	116	74	42	100
Line6	96	72	24	100
Line7	91	50	41	100
Line8	89	48	41	100
总计	1 771	939	821	98.7

表 1 是对 8 条边缘曲线编码的结果,因为索引号在 256 以内,所以可用一个字节 (uint8) 来存储索引号,但每条边缘曲线还要存储起点和终点,这需要 4B。每条曲线编码后所需存储空间如表 1。从表 1 可知,对较长边缘曲线来说,编码后用于存储一个边缘点的空间平均小于 1 个位。但因为存储边缘起点和终点就要 32 个位,对于较短的边缘曲线来说,每个边缘点所占空间会增大,如果边缘点数小于 32 个,每个边缘点的存储空间肯定大于 1 个位。多环链编码对于较长的曲线进行编码是十分有效的,这些边缘恰好是重要边缘信息。而多环链编码方法对于只包含一个线段的边缘编码并不理想,因为没有信息可用于预测线段的角度并且存储起点和终

(下转第 224 页)

对于 $2N \times 2N$ 乘法器，可以直接利用类似的树型结构实现。但使用组合模块方式更加合理，因为一方面这样可以避免拉大进位链，更为关键的是可以突破 FPGA 器件容量的限制。如对于 FLEX10K20，其包含 EAB 模块的数量只有 6 个，其所能实现的最大的对称乘法器只能是 8×8 。

基于模块化的思想， $2N \times 2N$ 乘法器可定义为

$$P = YX = (Y_2 2^N + Y_1) (X_2 2^N + X_1)$$

$$= Y_2 X_2 2^{2N} + (Y_1 X_2 + Y_2 X_1) 2^N + Y_1 X_1$$

其中，下标 2 和 1 分别代表操作数的高位部分和低位部分。

通过模块化方法，可以采用 4 个 8×8 乘法器和 1 个 16 位加法器及 1 个 32 位加法器实现 16×16 乘法器。同理，可以采用模块的方法使用 8×8 乘法器构造 32 位乘法器。

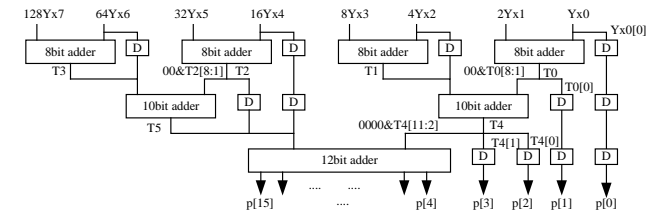


图 4 改进的 8 位树型乘法器

3 验证与结论

32 位树型乘法器采用一片 Xilinx xc2s400e-6ft256 器件实现，其规模为 145 000 门。用 VHDL 语言作 RTL 级描述，用 XST 进行综合，在 ModelSim 中进行仿真，代码已在 Xilinx xc2s400e-6ft256 器件上实现，结果验证正确。系统允许最大

(上接第 179 页)

点会占用太多空间，实际应用中将抽取算子得到的边缘曲线按长度进行排序，根据具体要求只对较长的曲线进行编码。

表 2 是对表 1 中的编码结果进行解码的数据，由表 2 可知此边缘编码算法解码后保持了很高的准确性。解码后，原来的边缘点基本可以找到重合点或邻接点（对在解码所得边缘点中找不到重合点的原边缘点寻找其邻接点），邻接点与原边缘点的误差位 1 或 $\sqrt{2}$ (pixels)，这相对于基于边缘特征的小波变换来说，是可以接受的误差。为了进一步改进斐波纳契数列压缩编码的性能和效率要注意以下技术细节问题：

(1) 多环链网格点的索引编号。对网格点进行索引编号很容易，比如可以如图 4 所示进行编号，从里向外沿每个环按顺时针方向进行编号，对每个环来说，正左方的网格点为环索引编号的起始点，即由里向外，每个环的正左方网格点的编号，依次为 1, 9, 25, 49, 89, 153。在实际应用中，环索引利用连续曲线的可预测性进行索引编号，因此网格点的编号要同“一直向前”的策略相一致，即环索引编号的起始点 1, 9, 25, 49, 89, 153 应沿着前面已编码的线段延伸的直线方向排列，这样每一个索引反映了一个相对于前面已编码线段的相对角度偏移量而不是绝对偏移量。因此索引出现的分布状态倾向于最小角度变化，改进了索引的压缩效率。这 256 个环索引再加上表示“边缘结束”的第 257 个索引可以用整数算术编码技术进行编码。

(2) 边缘编码的开始点。在实际应用中，图像通过边缘抽取算子的处理会得到多条包含边缘信息的曲线。多环链编码器必须用每个边缘曲线的开始点进行初始化，为了这个目的，我们利用算术差分偏移量编码边缘的开始端点。编码器计算一系列边缘开始端点之间水平和垂直偏移量的平均值，并将此信息与在头文件中的边缘全部编码一起传送。编码器将每个从前面继承的边缘的水平和垂直偏移量与平均值之间的差值进行编码。

时钟频率为 107.945MHz，占用了 1 019 个 Slices(21%)，1 898 个 Slice Flip Flops(19%)，1 227 个 4 输入查找表(12%)和 1 个全局时钟(25%)等。与之相比较，采用移位累加方式和非组合方式构建的 32 位乘法器数据如表 2 所示。

表 2 采用不同方式构建 32 位乘法器的性能对比

结构	组织方式	最大时钟频率 MHz	Slices	Slice Flip Flops	4 input LUTs
移位累加方式	非模块化	82.583	195	237	370
优化的树型方式	非模块化	93.174	582	1 088	1 024
优化的树型方式	模块化	107.945	1 019	1 898	1 227

基于实验验证的结果，采用模块化的优化树型结构可构建出基于 FPGA 的具有最佳性能的 32 位乘法器。

参考文献

- 1 Wiatr K. Implementation of Multipliers in FPGA Structures. Quality Electronic Design, International Symposium, San Jose, CA:2001-03: 415-420
- 2 Gallagher W L. High Radix Booth Multipliers Using Reduced Area Addertrees. Signals, Systems and Computers, Conference Record of the 28th Asilomar Conference, Pacific Grove, CA, 1994, 1: 545-549
- 3 Uwe Meyer-Baese. 数字信号处理的 FPGA 实现. 北京:清华大学出版社, 2003-01
- 4 傅志晖, 程东方, 梅其元等. 32 位浮点阵列乘法器的设计及算法比较. 微电子学, 2003, 33(3): 190-195
- 5 Xilinx. Spartan-IIIE Data Sheet. 2003-07

4 结束语

本文虽然给出了基于边缘特征的改进小波变换方法，但并没有给出完整的利用此改进小波变换的图像压缩算法，这也正是我们今后需要研究的内容。但通过对基于边缘特征小波变换的优点的分析不难发现，无论是对图像进行快速浏览及基于内容的图像识别和检索的实现，还是对图像压缩效率和性能的提高，改进小波变换法都是有所帮助的。

参考文献

- 1 Said A, Pearlman W A. A New Fast and Efficient Image Code Based on Set Partitioning Inhierarchical Trees [J]. IEEE Trans. Circuits and System for Video Technology, 1996,6(6)
- 2 Schilling D, Cosman P C. Preserving Step Edges in Low Bit Rate Progressive Image Compression [J]. IEEE Trans. Image Processing, 2003, 12(12): 1473-1484
- 3 Schilling D, Cosman P. Fea Preserving Image Coding for Very Low Bit Rates. In : Proc. of 2001 IEEE Data Compression Conf. (DCC), J. A Storer and M. Cohn, Eds., Snowbird, UT, 2001, 103-112
- 4 Fan G, Cham W K. Model-based Edge Reconstruction for Low Bit-rate Wavelet Compressed Images [J]. IEEE Trans. Circuits Syst. for Video Technol., 2000, 10 (1): 120-132
- 5 Schilling D, Cosman P. Edge-enhanced Image Coding for Low Bit Rates. In : Proc. of Int. Conf. Image Processing (ICIP-99), 1999, 3: 747-751
- 6 Rosin P, West G. Nonparametric Segmentation of Curves Into Various Representations [J]. IEEE Trans. Pattern Anal and Machine Intell., 1995, 17(12):1140-1153
- 7 秦前清, 杨宗凯. 实用小波分析. 西安:西安电子科技大学出版社, 1995