

基于提升小波变换的 SPIHT 编码算法研究

范文兵, 陈 晖, 陈 燕

FAN Wen-bing, CHEN Hui, CHEN Yan

郑州大学 信息工程学院, 郑州 450052

College of Information Engineering, Zhengzhou University, Zhengzhou 450052, China

E-mail: chenhui1218@yahoo.com.cn

FAN Wen-bing, CHEN Hui, CHEN Yan. Study of SPIHT image coding based on lifting wavelet transform. *Computer Engineering and Applications*, 2008, 44(5): 38-39.

Abstract: Due to the computational complexity of SPIHT, the integer-to-integer lifting scheme, which could simplify the computational process, was adopted instead of the normal wavelet transform. Integrating the characteristic of Lifting Wavelet Transform and DCT transform, the thesis advanced a new scheme. Experiments show that the PSNR value is more than that of the original SPIHT at the same bit rate.

Key words: Lifting Wavelet Transform; DCT transform; SPIHT algorithm

摘 要: 针对小波变换和多级树集合分裂图像编码具有浮点数运算、算法复杂等特点, 采用整数实现的提升格式代替了原来的小波变换, 并结合提升小波变换和 DCT 变换的特点, 提出了一种改进方案。仿真结果表明, 在相同的较低比特率情况下, 与原算法相比, 峰值信噪比有明显提高, 取得了更优的压缩效果。

关键词: 提升小波变换; DCT 变换; SPIHT 编码算法

文章编号: 1002-8331(2008)05-0038-02 **文献标识码:** A **中图分类号:** TP391

1 引言

Swrlidens 提出了一种新的小波构造方法——提升方法, 基于提升方法的小波变换不依赖于傅立叶变换, 计算简单, 易于硬件实现, 可以实现图像的完全无损编码, 被称为第二代小波变换。Said 和 Pearlman 根据 Shapiro 零数编码算法 EZW 的基本思想, 提出了一种性能更优的图像压缩算法, 即多级树集合分裂算法 (Set Partitioning in Hierarchical Trees, SPIHT) 的编码算法, 它的一个重要思想就是“编码、解码采用同样的规则对小波系数按其幅值分布区间进行重排”, 也即集合划分。在对 SPIHT 算法和静止图像编译码技术认真研究的基础上, 根据小波变换和 DCT 变换优缺点, 提出了一种新的算法, 用 MATLAB7.0 程序仿真实现, 以便得到更好的压缩效果。

2 提升小波变换算法

提升算法的基本思想在于通过一个基本小波逐步构建出一个更加良好性质的新小波。相对于 Mallat 算法而言, 提升算法是一种更为快速有效的小波变换实现方法, 他不依赖于傅立叶变换, 完全在空域完成了对正交小波滤波器的构造。具有以下优点: 能够实现更快速的小波, 仅有正负号的差别; 描述非常简单, 可以避免使用傅里叶变换。

构造双正交小波, 使用基本的多项式差补来获取信号的高频分量, 再通过构建尺度函数来获取信号的低频分量。提升算法通常有三个基本步骤: 分裂 (Split)、预测 (Predict)、更新 (Update), 如图 1 所示。

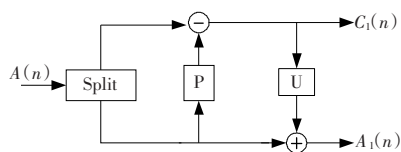


图 1 提升小波的原理图

第一分裂过程, 是将数据集 A_0 根据其序数的奇偶性分解成两个子集, 偶数点集合定义为 A_1 , 奇数点集合定义为 C_1 (即惰性小波变换, Lazy Wavelet)。即: $A_1(n) = A_0(2n)$; $C_1(n) = A_0(2n+1)$, ($n \in Z$)。

第二步是预测提升, 主要是消除分裂后留下的冗余。用 A_1 预测 C_1 , 预测误差形成新的 C_1 。设一个独立于数据集的预测算子 P , 使得 $C_1 = C_1 - P(A_1)$ 。

当原始的 C_1 和 A_1 相关性大时, 通过合理的预测, 能使新的 C_1 具有很低的能量分布。这里已经可以用更小的子集 A_1 和小波子集 C_1 来代替原信号集 A_0 。重复这个分解过程, 将 A_1 进一步分解成 A_2, C_2 一直继续下去, 直到指定的 n 分解, 形成系

基金项目: 河南省教育厅自然科学基金 (No.2007510019)。

作者简介: 范文兵 (1969-), 男, 博士, 副教授, 主要研究领域: 数字图像处理及中间件的研究; 陈晖 (1982-), 男, 硕士研究生, 主要从事数字图像处理及多媒体技术的研究。

收稿日期: 2007-06-07 修回日期: 2007-08-28

数集 $\{A_n, C_n, \dots, C_1\}$ 来取代原数据集 A_0 。但是, 在这种分解过程中, 希望最后的系数子集 A_n 中保留 A_0 中的某些全局性质。但是分裂过程不一定能够使这个性质保持下来, 因此, 在每一级分解中增加了第三步。

第三步是更新过程, 其目的是使某些全局性质得以保障。更新的思想是要找一个更好的子集 A_1 , 使得它保持原图的某一标量特性 $Q(A_1)$ (例如均值、消失矩等)不变, 即要求: $Q(A_1)=Q(A_0)$ 。为了使正反变换一致, 须构造一个算子 U 对 A_1 进行更新, 定义为: $A_1=A_{i+1}+U(C_{i+1})$ 。

重复以上步骤, 得到如下小波变换公式:

$$\begin{cases} \{A_{i+1}, C_{i+1}\} = \text{Split}(A_i) \\ C_{i+1} = C_{i+1} - P(A_{i+1}), i=1, 2, \dots, n \\ A_{i+1} = A_{i+1} + U(C_{i+1}) \end{cases}$$

对于提升算法, 有了正变换, 很容易得到反变换, 只需改变加减号。这是提升算法的一个优良特性。反变换过程为:

$$\begin{cases} A_{i+1} = A_{i+1} - U(C_{i+1}) \\ C_{i+1} = C_{i+1} + P(A_{i+1}), i=1, 2, \dots, n \\ A_i = \text{Join}(A_{i+1}, C_{i+1}) \end{cases}$$

综上所述, 该分解方式与离散小波变换的思想是一致的, 将序列分解成更紧凑的表示方式, 分裂方式、预测器的不同选择, 等价于选取了不同的滤波器组。

3 SPIHT 编码算法

SPIHT 算法吸取了 EZW 零树的许多思想, 通过方向树最有效地表示有效值映射, 通过对树的划分, 将尽可能多的无效值系数汇集在一个子集中, 用一个单位符号表示。SPIHT 算法编码思想是通过递减的阈值序列生成一系列重要图(或称位图)来逐步逼近每一个小波系数。为了有效地编码每一个位图, 对于任意节点 (i, j) , 引入如下四种集合:

- $O(i, j)$: 节点 (i, j) 的四个直接后续节点的坐标集合;
- $D(i, j)$: 节点 (i, j) 的所有后续节点的坐标集合;
- $L(i, j)$: 节点 (i, j) 的除直接后续以外的所有后续节点的坐标集合, 即 $L(i, j)=D(i, j)-O(i, j)$;
- H : 在小波塔形分解中的最高层节点的坐标集合。

除了根节点之外, 对所有子节点 (i, j) , 有 $O(i, j)=\{(2i, 2j), (2i, 2j+1), (2i+1, 2j), (2i+1, 2j+1)\}$

另外, SPIHT 算法采用了 3 个链表来记录编码信息:

LIS: 不重要集合链表(List of Insignificant Sets); 每个记录都是坐标 (i, j) 形式, 它代表一个集合 $D(i, j)$ 或 $L(i, j)$ 。

LIP: 不重要像素集合链表(List of Insignificant Pixels); 每个记录也是坐标 (i, j) 形式, 它代表 (i, j) 位置有一个无效值。

LSP: 重要像素集合链表(List of Significant Pixels); 每个记录也是坐标 (i, j) 形式, 它代表 (i, j) 位置有一个有效值。

对变换后小波系数的分类排序以及加细算法如下:

3.1 初始化

输出 $n=\lfloor \log_2(\max |x_{i,j}|) \rfloor$, 置 LSP 为空表, 将所有最高层节点 $(i, j) \in H$ 放入 LIP 表中, $(i, j) \in H$ 中带有子孙的加入 LIS, 并作为 $D(i, j)$ 类集合。

3.2 分类扫描过程

3.2.1 对 LIP 中每个记录 (i, j) 作

- (1) 输出 $S_n(i, j)$

(2) 若 $S_n(i, j)=1$, 则将 (i, j) 移到 LSP 表中, 并输出 $X_{i,j}$ 的符号位。

3.2.2 对于 LIS 中每个记录 (i, j) 作

(1) 如果这个记录代表一个 $D(i, j)$ 类集合, 则输出 $S_n(D(i, j))$; 如果 $S_n(D(i, j))=1$, 对每一个 $(k, l) \in O(i, j)$ 作:

(1.1) 输出 $S_n(k, l)$

如果 $S_n(k, l)=1$, 则将 (k, l) 加入到 LSP 表中, 并输出 $X_{k,l}$ 的符号;

如果 $S_n(k, l)=0$, 则将 (k, l) 加入 LIP 表中;

(1.2) 如果 $L(i, j)$ 不为空集合, 则将 (i, j) 移到 LIS 表的尾部, 并表明它是 $L(i, j)$ 类型集合, 转移至步骤 2.2.2; 否则, 将 (i, j) 移出 LIS 表;

(2) 如果这个记录代表 $L(i, j)$ 类集合, 则输出 $S_n(L(i, j))$;

如果 $S_n(L(i, j))=1$, 则

(2.1) 将每个 $(k, l) \in O(i, j)$ 加入到 LIS 尾部, 并标记为 $D(k, l)$ 类型;

(2.2) 从 LIS 中移去 (i, j) 项。

3.3 细化过程

对于 LSP 表中所有节点 (i, j) , 移去那些在最后一次排序过程中加入的节点, 输出 $|X_{i,j}|$ 的第 n 个最高有效位。

3.4 量化步长更新

$n=n-1$, 转移至步骤 2。

4 基于提升小波变换的 SPIHT 编码改进方案

通过大量的实验数据分析, 对于在低比特率情况下压缩高频图像, 容易丢失大量细节信息, 重构图像相对模糊, 效果不是很理想。针对这一问题, 结合提升小波的优点, 提出了一种改进方案, 使用提升小波代替原小波算法, 并利用 DCT 变换只需要做实数的乘法和加法, 计算复杂度适中, 计算速度快; 对大多数自然图像, DCT 能将最多的信息放到最少的系数上去等优点, 把 DCT 变换应用在提升小波变换之前, 对小波变换前的高频子带先做处理, 使变换前的图像在高频域上能量更加集中。

5 实验结果

本文采用 256×256 的 Lena.bmp 和 512×512 的 Baboon.bmp 标准测试图像对本算法进行测试。采用 CDF/97 提升小波, 进行 7 级小波分解, 压缩结果如图 2。将被测试图像在不同比特率(bpp)下的峰值信噪比 PSNR (dB) 值相比较, 试验结果如表 1。

表 1 两种算法的 PSNR 结果比较

	Lena.bmp		Baboon.bmp			
	bpp	原算法/dB	改进算法/dB	bpp	原算法/dB	改进算法/dB
低 频 图 像	0.1	19.855	23.222	0.1	18.946	19.545
	0.2	21.719	25.084	0.2	19.519	20.040
	0.3	23.008	26.500	0.3	19.917	20.373
	0.4	24.326	27.322	0.4	20.417	20.865
	0.5	25.258	28.141	0.5	21.028	21.233
	0.6	26.180	28.940	0.6	21.500	21.620
	0.7	27.097	29.610	0.7	21.970	22.003
	0.8	28.107	30.289	0.8	22.423	22.357
	0.9	28.900	30.826	0.9	22.969	22.709
	1.0	29.713	31.222	1.0	23.477	23.137

由实验数据可以看出, 本文设计的基于提升小波变换的 SPIHT 压缩算法和原算法相比较, 能明显的提高 PSNR 值, 压