

文章编号:1001-9081(2005)07-1626-03

## 应用提升格式实现的基于行小波变换的图像压缩算法

张宏伟,刘正光,陈红新

(天津大学 电气与自动化工程学院,天津 300072)

(alleckzhang@163.com)

**摘要:**内存需求量大、计算复杂度高问题很大程度上限制了 JPEG2000 的应用。基于行小波变换的图像压缩算法以累进方式完成列向小波变换,在不影响变换结果的前提下降低了对存储容量的需求。应用三项加法单元形式的提升格式代替原基于行的小波变换算法中的 Mallat 算法,充分利用了提升格式的全替换特性,加快了计算速度,节省了内存。同时针对基于行的小波变换的特点,设计了相应的上下文模板,可以简洁、高效地进行概率估计。应用该方法对 JPEG2000 进行改进,可大大提高其实用性。

**关键词:**基于行的小波变换;提升格式;基于上下文的算术编码

**中图分类号:** TN919.81 **文献标识码:** A

## Lifting-schemed line-based wavelet image compression

ZHANG Hong-wei, LIU Zheng-guang, CHEN Hong-xin

(School of Electrical Engineering and Automation, Tianjin University, Tianjin 300072, China)

**Abstract:** Due to the large requirement for memory and the high complexity of computation, JPEG2000 can not be used in many conditions. The line-based wavelet transform was proposed and accepted because it required lower memory without affecting the result of wavelet transform. In this paper, the improved lifting scheme was used to perform wavelet transform to replace Mallat algorithm in the original line-based wavelet transform. The corresponding context-based arithmetic coding was discussed here too. As a result, considerable reduction of memory and computational costs can be achieved.

**Key words:** line-based wavelet image compression; lifting scheme; context-based arithmetic coding

### 0 引言

作为新一代静态图像压缩标准, JPEG2000 与现有标准相比在主客观质量上都有无可比拟的优势。然而,内存需求量大、计算复杂度高因素大大限制了它的应用,如:在数码相机、打印机等中的应用。

由 Chrysafis 提出的基于行小波变换的图像编码算法<sup>[1]</sup>以传统的 Mallat 算法为基础,它在小波变换中渐进地完成列向滤波,变换中只需存储几行数据,可节省大量内存。提升格式是 Sweldens 在 20 世纪 90 年代首先提出的<sup>[2]</sup>,与 Mallat 算法相比,具有计算量小、速度快,具有全替换特性,已成为构造第二代小波的关键技术。在本文中,应用提升格式代替 Mallat 算法实现基于行的小波变换,同时考虑到提升格式的特点,应用三项加法单元实现提升格式,充分利用了提升格式的优势。

与基于行的小波变换相适应,设计了基于行小波变换的上下文形成方式,以最简洁的方式形成上下文,可以有效地进行概率估计,进一步节省内存。

### 1 基于行小波变换的图像压缩算法

传统的小波变换,在读取整幅图像数据后,先对所有行、再对所有的列进行小波变换,继而得到整幅图像的小波变换系数。基于行的小波变换算法不同,每当读取一行新的数据时,先对它进行行小波变换。在得到一定数量的行之后(这

个数量由滤波器长度决定),就开始进行列向小波变换。此算法只需存储很少数量的图像数据,并且图像数据只在进行小波变换时存储,一旦不再需要就被从内存中释放掉,这样可以大大节省内存。



图 1 基于行的三次小波变换

本文中,所需内存的大小用图像数据行的数量表示(即大小为  $B$  的内存可存储  $B$  行的图像数据)。在基于行小波变换的图像编码过程中,按行顺序读取图像数据,在读取每行的同时进行行小波变换。设低通滤波器的长度为  $L = 2S + 1$ ,读取并行变换完  $S + 1$  行数据后,经过左延拓,大小为  $L$  的滤波缓存  $B_F$  已满,由 Mallat 算法可知,此时可以对第 0 行数据进行列向滤波。接下来读取新的两行数据,然后对第一行数据进行列向

收稿日期:2004-12-29 基金项目:天津市自然科学基金资助项目(023601011)

作者简介:张宏伟(1972-),男,河北承德人,博士研究生,主要研究方向:图像压缩与编码、图像分割与识别; 刘正光(1945-),男,福建闽清人,教授,博士生导师,主要研究方向:模式识别、图像处理、电力电子技术、智能控制; 陈红新(1978-),男,江苏盐城人,博士研究生,主要研究方向:图像压缩与编码。

滤波。对于一次小波分解来说,第0行输出为低通滤波结果,接下来是高通结果,然后又是低通滤波结果,如此类推。对于多级小波变换,变换级数为 $N(0, 1, 2, \dots, n, \dots, N-1)$ ,则行标记能被 $2^N$ 整除的行,为低通滤波结果,其他的行,行标记若能被 $2^n$ 整除但不能被 $2^{n+1}$ 整除,则这一行属于第 $n$ 级变换。本文中应用的滤波器为CDF9/7滤波器,因此: $L=9, S=4$ 。图1所示为CDF9/7滤波器进行三级基于行的小波变换的过程,每一格代表一行数据,它的水平位置表示在变换中出现的顺序。当读入第12行图像数据时可得第1行二级变换结果,当读入第28行时可得第1行三级变换结果。

这种算法对内存的需求依赖于滤波器的长度、图像的宽度和小波变换的级数。每进行一级小波变换,就需要大小为 $L$ 的滤波缓存 $B_F$ 。为了保持编码器和解码器输出序列同步,每一级小波变换的高通结果不是马上输出,而是直到最高级小波变换完成后再输出。这样,就需要附加的同步缓存 $B_S$ 存储实时的高通滤波结果。全部缓存包括 $B_F$ 和 $B_S$ 两部分:

滤波缓存的大小与小波变换的级数 $N$ 成比例:

$$B_F = N(2S + 1) = NL \quad (1)$$

每级小波变换(最高级除外)需要一大大小为 $S(2^{N-i} - 1)$ 的同步缓存, $i$ 代表当前级。对于 $N$ 级变换,同步缓存的总量为:

$$B_S = \sum_{i=1}^N S(2^{N-i} - 1) = S(2^N - N - 1) \quad (2)$$

针对本文中应用的CDF9/7滤波器进行三级小波变换( $L=9, S=4, N=3$ ),所需内存计算如下。全部变换需内存43行,与传统小波变换相比大大节省了内存。

$$\text{滤波缓存 } B_F = 3 \times (2 \times 4 + 1) = 27$$

$$\text{同步缓存 } B_S = S + 3S = 4S = 4 \times 4 = 16$$

$$\text{总缓存 } B = B_S + B_F = 27 + 16 = 43$$

## 2 改进的基于行小波变换的图像压缩算法

如前所述,基于行的小波变换是基于传统的Mallat算法。考虑到提升格式具有很多优越性,可引入它来实现基于行的小波变换以进一步降低计算复杂度和内存需求。

由提升格式实现的小波变换由以下三步组成:分割、预测和更新。首先,应用懒惰小波将原始信号 $x_i$ 分解为不相关的两部分:偶部分 $s_i^0 \in x_{2i}$ 和奇部分 $d_i^0 \in x_{2i+1}$ ,然后应用预测算子 $P_n(k)$ 由偶部分 $s_i^0$ 预测奇部分 $d_i^0$ ,应用更新算子 $U_n(k)$ 更新偶部分 $s_i^0$ :

$$d_i^1 = d_i^0 - \sum P_n(k) \cdot s_i^{n-1} \quad (3)$$

$$s_i^n = s_i^{n-1} - \sum U_n(k) \cdot s_i^{n-1} \quad (4)$$

最后用系数 $K_0$ 和 $K_1$ 归一化输出结果:

$$s_i^N = K_0 \cdot s_i \quad (5)$$

$$d_i^N = K_1 \cdot d_i \quad (6)$$

这样,初始信号可由 $d_i^N, s_i^N, P_n(k), U_n(k), K_0$ 和 $K_1$ 表示。

Daubechies 和 Sweldens 已经证明,任何基于小波的FIR滤波器的小波变换都可以分解成为若干提升步骤的形式。对CDF9/7滤波器进行提升变换,可得以下提升步骤<sup>[3]</sup>,其中 $\alpha$ 和 $\gamma$ 是预测算子, $\beta$ 和 $\delta$ 是更新算子:

$$s_i^{(0)} = x_{2i}$$

$$d_i^{(0)} = x_{2i+1}$$

$$d_i^{(1)} = d_i^{(0)} + \alpha(s_i^{(0)} + s_{i+1}^{(0)})$$

$$s_i^{(1)} = s_i^{(0)} + \beta(d_i^{(1)} + d_{i-1}^{(1)})$$

$$d_i^{(2)} = d_i^{(1)} + \gamma(s_i^{(1)} + s_{i+1}^{(1)})$$

$$s_i^{(2)} = s_i^{(1)} + \delta(d_i^{(2)} + d_{i-1}^{(2)})$$

$$s_i = \rho \times s_i^{(2)}$$

$$d_i = d_i^{(2)} / \rho$$

由以上的提升步骤可看出,每一步都可以用 $R_{\text{output}} = R_1 + c(R_0 + R_2)$ 形式的三项加法单元表示<sup>[4]</sup>,其中 $R_0, R_1$ 和 $R_2$ 表示输入缓存, $R_{\text{output}}$ 表示输出缓存, $c$ 表示乘法算子。本文中,应用三项加法单元来实现提升运算。

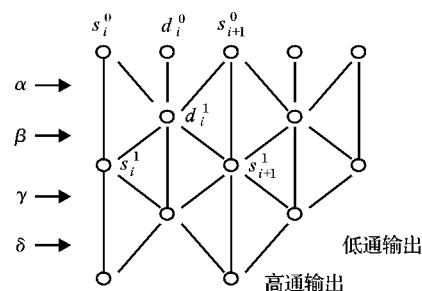


图2 应用三项加法单元实现基于行的行向小波变换

基于行的小波变换编码器每读入一行图像数据就将其进行行变换,图2所示为应用三项加法单元格式进行基于行的小波变换的行向变换过程,此变换具有全变换特性(full in-place),原始图像数据可直接被当前的变换结果替换,因此只需很少的计算步骤且不需要额外的辅助内存就可完成变换。

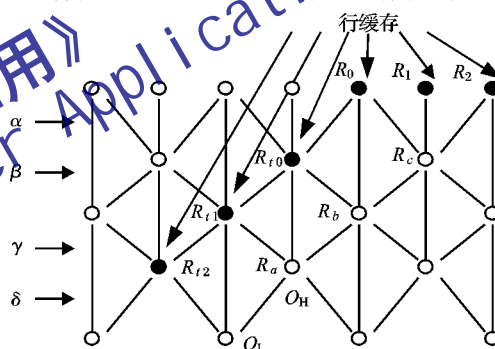


图3 应用三项加法单元实现基于行的列向小波变换

应用三项加法单元实现的基于行的渐进的列向的小波变换如图3所示。水平方向的每个点表示表示经过行变换的一行数据。垂直方向的点表示每行数据的更新。由图3,我们可按如下步骤计算低通输出 $O_L$ 和高通输出 $O_H$ :

$$O_L = \delta(R_{d_2} + R_{d_1} + R_a) =$$

$$\delta(R_{d_2} + R_{d_1}) + \delta \gamma(R_{d_1} + R_{d_0} + R_b) =$$

$$\delta R_{d_2} + \delta(1 + \gamma) R_{d_1} + \delta \gamma R_{d_0} + \delta \gamma \beta(R_{d_0} + R_0 + R_c) =$$

$$\delta R_{d_2} + \delta(1 + \gamma) R_{d_1} + \delta \gamma(1 + \beta) R_{d_0} + \delta \gamma \beta R_{d_0} +$$

$$\delta \gamma \beta \alpha(R_0 + R_1 + R_2) = \delta R_{d_2} + \delta(1 + \gamma) R_{d_1} +$$

$$\delta \gamma(1 + \beta) R_{d_0} + \delta \gamma \beta(1 + \alpha) R_{d_0} + \delta \gamma \beta \alpha R_{d_1} + \delta \gamma \beta \alpha R_{d_2} \quad (7)$$

$$O_H = R_a = \gamma(R_{d_1} + R_{d_0} + R_b) =$$

$$\gamma R_{d_1} + \gamma R_{d_0} + \gamma \beta(R_{d_0} + R_0 + R_c) =$$

$$\gamma R_{d_1} + \gamma(1 + \beta) R_{d_0} + \gamma \beta R_{d_0} + \gamma \beta \alpha(R_0 + R_1 + R_2) =$$

$$\gamma R_{d_1} + \gamma(1 + \beta) R_{d_0} + \gamma \beta(1 + \alpha) R_{d_0} + \gamma \beta \alpha R_{d_1} + \gamma \beta \alpha R_{d_2} \quad (8)$$

由上述计算可以看出, $O_L$ 和 $O_H$ 只与信号数据 $R_0, R_1, R_2$ 和临时数据 $R_{d_2}, R_{d_1}, R_{d_0}$ 有关。在多级小波变换中,只需暂存LL子带结果来进行下一级变换,而且每一级子带的长度以2的指数倍递减。本文中,为了与Mallat算法比较,我们进行三级小波变换。

$$\text{信号数据缓存为: } B_{LS} = 3 \times (1 + \frac{1}{2} + \frac{1}{4}) = \frac{21}{4}$$

$$\text{临时数据缓存为: } B_{LT} = 3 \times \frac{1}{2} \times (1 + \frac{1}{2} + \frac{1}{4}) = \frac{21}{8}$$

应用三项加法单元所需的内存为 63/8 行,而应用 Mallat 算法所需内存为 43 行。应用三项加法单元形式的提升算法大大节省了内存。

### 3 基于行小波变换的上下文的形成

基于行的小波变换渐进地产生小波系数并将其暂存。在最高级的小波变换完成后,所有输出结果形成塔式结构。由于小波变换的降采样特性,在每一级小波变换后,每一行数据的宽度变为它的一半。如对图像数据进行三级小波变换,塔式结构中:一级变换子带(HL1, LH1, HH1)的大小为 4 行  $\times$  1/2 图像宽;二级变换子带(HL2, LH2, HH2)的大小为 2 行  $\times$  1/4 图像宽;三级变换子带(LL3, HL3, LH3, HH3)的大小为 1 行  $\times$  1/8 图像宽。

上述塔式结构中的每个子带的系数单独编码,采用基于上下文的算术编码方式。上下文模型选用 JPEG2000 推荐的模型<sup>[5]</sup>,被编码系数的上下文只与其 8 个相邻系数有关,可简洁高效地进行概率估计。按照这种方式,要对当前系数编码必须知道它的前一行数据和后一行数据。第二级变换和第三级变换子带的数据缺少这些信息,因此在编码过程中我们对当前系数的前一行数据和后一行数据进行暂存,如图 4 所示。在对当前行编码完成后,当前行被移入前一行,而后一行被移入当前行。读入新的一行作为后一行数据。与基于行的小波变换相适应,这种上下文形成方式可以滚动使用内存,以最少的内存有效地进行概率估计。应用这种算法进行编码,不需要存储全部图像信息,进一步节省了内存。

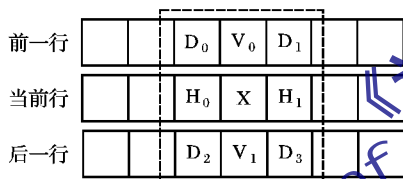


图4 基于行小波变换的上下文模型

### 4 实验结果及结论

我们将本文所提出的算法嵌入 JPEG2000 标准第 5 部分推荐的参考软件包“Jasper”中,并应用图像“Lena”,“Water”,“Cats”等 JPEG2000 测试图片对该算法进行了验证。

图 5 所示为图像“Lena”应用本文的算法在不同比特率要求下进行压缩后的 PSNR 值,其中小波变换为四级。表 1 所示为应用本文算法和 JPEG2000 算法压缩不同图像时所需的内存。从上面的结果可以看出,与 JPEG2000 比较,本文所述算法有一点轻微的质量下降,但其所用内存只为 JPEG2000 的 10%。

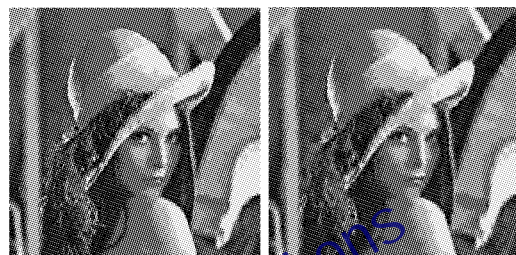
在本文中,我们应用提升算法代替 Mallat 算法实现基于

行的小波图像压缩。由于应用三项加法单元实现提升步骤,可以充分利用提升格式的优点,大大减少了内存需求,降低了计算复杂度。针对基于行的小波变换缺少全局信息的特点,我们设计了相应的上下文模板,可以高效地进行概率估计,进一步减少了内存需要和计算复杂度。



(a) 原图

(b) bpp=0.473, PSNR=60.86



(c) bpp=0.397, PSNR=53.26

(d) bpp=0.159, PSNR=27.68

图5 图像 Lena 在不同比特率下压缩时的 PSNR 值

表1 内存需求

图像	大小	JPEG2000	基于行
Water	2.87M (1465 $\times$ 1999)	36.1M	2.3M
Cats	6.15M (3072 $\times$ 2048)	73.5M	4.9M
Lena	66K (256 $\times$ 256)	906K	33K

#### 参考文献:

- [1] CHRYSAFIS C, ORTEGA A. Line based, reduced memory, wavelet image compression [J]. IEEE Transactions on Image Processing, 2000, 9(3): 378-389.
- [2] SWELDENS W. The Lifting Scheme: A new Philosophy in Biorthogonal Wavelet Construction [A]. Wavelet Applications in Signal and Image, Processing III[C], 1995. 68-79.
- [3] DAUBECHIES I, SWELDENS W. Factoring wavelet transforms into lifting steps [J]. Fourier Analysis and Applications, 1998, 4(3): 247-269.
- [4] CHANG W-H, LEE Y-S, PENG W-S, et al. A line-based, memory efficient and programmable architecture for 2D DWT [A]. IEEE Proceedings of ISCAS 2001[C], 2001, 4. 330-333.
- [5] ISO/IEC JTC 1/SC 29/WG 1, ISO/IEC FCD15444-1: JPEG2000 image coding system[S]. <http://www.jpeg.org/FCD15444-1.htm>, 2000.

(上接第 1604 页)

实验中只对图像采用了最简单的均匀划分,这使得其压缩比不高,但是我们所提出的方法不难与其他划分方式如四叉树或 HV 等结合以提高压缩比;另外,也可与其他的加速方法如 Saupe 特征向量等相结合以进一步提高编码速度。

#### 参考文献:

- [1] FISHER Y. Fractal Image Compression[J]. Fractals, 1994, 2(3).
- [2] HRUTGEN B, STILER C. Fast Hierarchical Codebook Search for Fractal Coding of Still Images[A]. Proceedings of EOS/SPIE Visual Communications PACS Medical Applications'93[C]. Berlin, Germany, 1993.
- [3] CONCI A, AQUICO FR. Fractal Image Coding by Multi-Scale Se-

lection Based on Block Complexity[J]. Journal for Geometry and Graphics, 1999, 3(1): 57-65.

- [4] 赵耀, 王红星, 袁保宗. 一种吸引子图像编码的快速算法[J]. 铁道学报, 2000, 22(2): 35-38.
- [5] 李杰, 付萍, 刘金国. 基于复合分类的快速分形图像压缩编码[J]. 计算机辅助设计与图形学学报, 2002, 14(4): 348-350.
- [6] TONG CS, PI M. Fast Fractal Image Encoding based on Adaptive Search[J]. IEEE Transactions on Image Processing, 2001, 10(9): 1269-1277.
- [7] 薛东辉, 朱耀庭, 朱光喜, 等. 基于 DFBIR 场的图像边缘提取的一种新方法[J]. 信号处理, 1996, 12(1): 18-21.