

基于快速提升小波变换的改进型 SPIHT 算法 在图像压缩中的应用

向泽智, 张兆宁

(中国民航学院空管学院, 天津, 300300)

摘要: 为保证图像传输的实时性和在保证图像质量下获得高压缩比,对图像压缩技术提出了新的改进型算法。提出的基于快速提升(lifting)小波变换^[1-2]的改进型 SPIHT(Set Partitioning in Hierarchical Trees)^[3]图像编码算法,引入快速提升小波变换以改进变换过程,且利用量化编码后出现许多连续零这一特点,将熵编码与 SPIHT 编码统一进行,一次完成,熵编码采用变长零游程方法。仿真结果表明,图像经该方法编码后峰值信噪比和效率上都得到了提高。该方法可以应用于电力系统中远程图像监控系统的图像数据传输与存储。

关键词: 图像压缩; 快速提升小波变换; 熵编码

中图分类号: TN919.81 文献标识码: A 文章编号: 1003-4897(2004)09-0020-04

0 引言

随着电力系统变电站自动化程度的不断提高,变电站将逐步向无人值守的方向发展,为提高电力生产和电力供应的高保障度及设备的安全可靠运行与防火、防盗、防破坏能力,同时加强设备的监视和自动化传输数据的能力,采用远程图像监控系统^[4]将具有非常深远的意义。图像监控计算机是变电站远程数字图像监控系统的核心,其主要功能之一就是图像数据的压缩处理。因此,为了节约传输和存储成本,采用一个高质量高压缩比的图像压缩编码算法很有必要。本文为此研究和提出了新的改进型算法来满足这一要求。

自 20 世纪 80 年代末期 S. Mallat 首次将小波变换引入图像处理以来^[5],小波变换以其优良的时频局部化能力及去相关能力在图像压缩编码^[6]领域得到了广泛应用,并取得了良好的效果。通常,基于小波变换的图像压缩编码大致经过 3 个关键步骤:首先选取合适小波母函数与滤波器,对图像信号实施小波变换;其次对变换后的小波系数进行量化编码,从而得到二进制符号流;最后对符号流进行熵编码以得到图像压缩后的比特流。其中,量化编码阶段占有举足轻重的地位,这是因为实际上所有的信息损失都发生在量化编码阶段。

近年来,人们围绕小波系数的量化编码开展了大量研究工作,提出了一系列新量化编码算法。其中,由 Said 和 Pearlman 提出的基于分层树集合分割

排序(SPIHT)静止图像压缩编码算法^[3],被认为是目前最先进的方法之一,该算法采用了空间方向树(spatial orientation tree, SOT)、集合 $D(i, j)$ 和 $L(i, j)$ 更有效地表示这样的系数结构,从而提高了编码效率。然而,理论分析和仿真结果表明,该算法也存在一些不足,具体表现为:

1) SPIHT 编码算法所采用的快速小波变换,需要跟庞大的图像数据做卷积运算,计算复杂,影响了压缩算法的实时性。

2) 在仿真中发现图像小波变换后量化时出现了大量的连续零,影响压缩能力。

针对以上不足,本文将提出一种基于快速提升小波变换的改进型 SPIHT 图像编码算法,该算法作了两项改进,一是引入快速提升小波以改进变换过程,进一步提高 SPIHT 编码的效率;二是将熵编码与 SPIHT 编码统一进行,一次完成,熵编码采用变长零游程方法。仿真结果表明:本文算法在压缩速度、压缩后图像质量等方面均优于基于零树结构的 SPIHT 编码算法。

1 零树结构(Zerotree Structure)^[7]

一般来说,图像经过小波变换后,图像的大部分能量都集中在最低精度的子图像里。并且,更有用的是,各子图像的小波系数间存在着空间自相似性,这一点比幅值顺序在图像编码中更具有吸引力。例如,若较低精度的子图像(如 LH_3)的一系数 $LH_3(x, y)$ 相对某阈值 T 为不重要系数,则系数 $LH_2(2x, 2y)$ 、 $LH_2(2x+1, 2y)$ 、 $LH_2(2x, 2y+1)$ 、 $LH_2(2x+1, 2y+1)$ 相对某阈值 T 为不重要系数的概率为 98.2%。

基于这种特点,有学者提出了一种新的数据结

基金项目: 中国民航总局教育基金资助项目(03-3-08)

构:零树结构,也就是空间向量树结构(spatial orientation trees),如下图 1 所示。

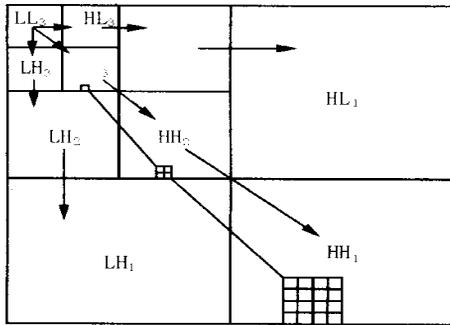


图 1 零树结构示意图

Fig. 1 Schematic diagram of zerotree structure

由图 1 可知,这种数据结构是一种金字塔结构,精度高的子图像相应于金字塔的低层,而精度低的子图像相应于金字塔的高层。当沿着金字塔的最高层向低层观察时,会发现高层的每个像素点在低层相应的方向会有 4 个“孩子”存在。而这 4 个“孩子”在其低层的相应方向也会有 4 个“孩子”存在。它们就构成了一种家族关系。这种关系在图中如箭头所示,当然这种关系也有例外的,如图中 LL_3 的部分,它就没有“子孙”存在。

2 基于快速提升小波变换的改进型 SPIHT 图像编码算法

下面将通过引入快速提升小波变换,采用基于零树结构的 SPIHT+游程统一编码措施,提出一种基于快速提升小波变换的改进型 SPIHT 图像编码算法。

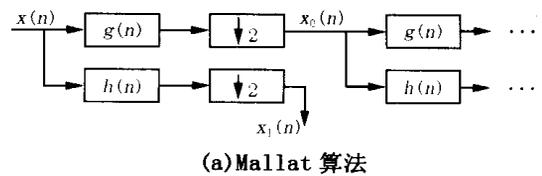
2.1 快速提升小波变换的引入

由于快速小波变换算法(Mallat 算法)在实际应用中存在着计算复杂、内存需求量较大、不能对所有尺寸图像进行变换等弱点,本文将采用另外一种快速小波变换实现算法,即“提升”(lifting)算法。

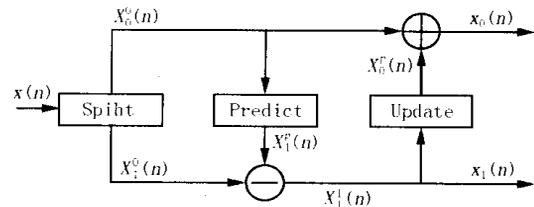
快速提升小波变换算法的基本思想是将 Mallat 算法中的每一级滤波运算分解为分裂(split)、预测(predict)和更新(update)3 个过程,如图 2 所示。

其中,分裂是将输入信号根据其序数的奇偶性分为以下 2 组:
$$\begin{cases} X_0^0(n) = x(2n) \\ X_1^0(n) = x(2n+1) \end{cases}$$
, 预测是将滤波器 P 作用于偶信号 $X_0^0(n)$ 得到奇信号的预测值 $X_1^0(n)$, 再将该预测值与奇信号 $X_1^0(n)$ 相减以得到奇信号的预测误差 $X_1^1(n)$, 因此

$$\begin{cases} X_1^0(n) = \sum_k p_k X_0^0(n-k) \\ X_1^1(n) = X_1^0(n) - X_1^0(n) \end{cases} \quad (1)$$



(a) Mallat 算法



(b) Lifting 算法

图 2 Mallat 算法与 Lifting 算法相互关系示意图

Fig. 2 Relationship of Mallat algorithm and Lifting algorithm

更新是将滤波器 U 作用于奇信号的预测误差 $X_1^1(n)$, 得到偶信号 $X_0^0(n)$ 的预测值 $X_0^0(n)$, 然后用来对偶信号进行校正。

$$\begin{cases} X_0^0(n) = \sum_k u_k X_1^1(n-k) \\ X_0^1(n) = X_0^0(n) + X_0^0(n) \end{cases} \quad (2)$$

式(1)与(2)中 k 的取值范围分别对应于滤波器 P 和 U 的阶数。

Mallat 算法任一组小波滤波器 ($h(n)$, $g(n)$) 均可以通过因式分解得到相应的一组或多组预测滤波器 (P) 和更新滤波器 (U), 并可在分裂以后进行多次的预测和更新操作。

图 2(b) 仅表示了一组滤波器, 其最后结果为:

$$\begin{cases} x_0(n) = X_0^1(n) \\ x_1(n) = X_1^1(n) \end{cases}$$

综上, 快速提升小波变换具有计算更快捷、能够在当前位置完成小波变换从而节省内存、能对任意尺寸图像进行变换等优点。

2.2 SPIHT 编码原理^[3]

SPIHT 将小波系数用空间方向树 (spatial orientation tree, SOT) 的形式组织起来。空间方向树的特点是: 如果父节点的系数值小于量化门限 T , 那么该树的所有节点系数有很大概率都小于 T (由小波系数在不同分解级的分布特性决定)。如果这个事实成立, 就只需用 1 bit (0) 表示这些节点在当地编码平面的值, 从而达到压缩的效果。

SPIHT 算法定义了三个链表 LIS, LIP 和 LSP。其中 LIS (List of Insignificant Sets) 是不重要的像素集链; LIP (List of Insignificant Pixels) 是不重要的像素链;

LSP(List of Significant Pixels) 是重要的像素链。它的简要编码步骤如下:

1) 初始化过程。确定初始量化门限 T , 初始化 $n=0$, 初始化 LSP, LIP 和 LIS。

2) 第 n 级编码的分类过程

对于 LIP 链表中的每个元素, 如果包含的 4 个系数幅度值都小于 T , 那么输出“0”; 否则输出“1”, 编码每个系数的符号和最高有效位, 并将其转移到 LSP 链表中。

对于 LIS 链表中的每个元素, 如果所有叶子节点的幅度都小于 T , 输出“0”, 否则输出“1”, 分裂空间方向树, 并相应更新 LIP 和 LSP 链表。

3) 第 n 级编码的细化过程

对 LSP 的每个元素(不包括在同一编码分类过程新加入 LSP 表的元素), 输出其在该级编码平面的值。

4) $n = n + 1, T = T/2$ 转到 2), 进行下一级编码。

一旦编码输出比特数达到给定值, 可以直接中止编码过程, 从而可以实现精确的码率控制。SPIHT 解码过程十分简单, 只需将编码过程中的输出改为输入即可。

2.3 SPIHT+游程统一编码方案及实现

在上述量化后, 出现了许多连续的零, 为了提高压缩比, 将其游程编码。考虑到效率问题, 将游程编码与量化统一进行, 一次完成。这里采用变长的零游程方案: 在 SPIHT 方法中由于有正、负重要系数、重要像素数据集和非重要像素及其数据集, 以及结束 END 共五种类别, 故需要用 3 位二进制来编码, 而 3 位二进制可以表示 8 种状态, 那么对连续零进行游程编码时, 可以充分利用未用的 3 个状态, 将其作为游程零个数的标识码, 采用变码长的方式来充分提高压缩比。游程长度标识码与系数类型码统一

编码, 如表 1 所示。

本方案编码过程如下:

1) 对图像进行三级快速提升小波变换;

2) 找出其余部分数据中绝对值最大的值, 求出其对应的二进制数的位数 B ;

3) 输出阈值 $T = 2^B$;

4) 按照 SPIHT 编码步骤扫描数据, 全部扫描完成后转至步骤 10), 上次扫描为重要系数的点不再参与扫描和编码;

5) 数据为非重要像素或其数据集为 0, 则统计连续 0 的个数 C , 然后转至步骤 4);

6) 非重要像素或其数据集的个数 C 不为零, 则输出对应次要系数根的系数类型码和个数代码;

7) 数据为正重要系数或负重要系数则输出对应系数类型码和幅值代码, 然后转至步骤 9);

8) 数据为非重要像素和非重要像素数据集, 则输出对应系数类型码;

9) 将 C 清零, 然后转至步骤 4);

10) 检查是否达到码率要求, 若没有则 $B = B - 1$ 转至 3), 否则结束 SPIHT 编码。

表 1 SPIHT 量化系数类型表

Tab. 1 Type of SPIHT quantification coefficient		
系数类型	代 码	后续数据位数
正重要系数	101	$\log_2(T/8)$
负重要系数	110	$\log_2(T/8)$
重要像素数据集	100	无
非重要像素和非重要像素数据集	000	3(有 1~8 个根)
	001	5(有 9~32 个根)
	010	7(有 33~128 个根)
	011	9(有 129~512 个根)
结束标志码	111	无

解码是编码的逆过程, 解码与编码顺序相反, 这样实现了零树量化+游程的编码和解码。



(a) Lena 原图



(b) 本文算法恢复图



(c) SPIHT 恢复图

图 3 标准图像 Lena(512 × 512 × 8 bit) 与恢复图

Fig. 3 Original image and decoded(512 × 512 × 8bit) Lena image

3 仿真结果

为了验证本文算法的高效性,以下在 PentiumIV 计算机上,以 $512 \times 512 \times 8\text{bit}$ 标准图像 Lena 为例,进行了 7 级小波分解与重构仿真,并与 SPIHT 算法进行了比较,仿真结果如表 2 和图 3 所示。其中,小波分解重构采用了著名的双正交 9/7 样条小波滤波器^[8]。表 2 给出了两种算法在终止阈值 $T=8$ 时的峰值信噪比 (PSNR) 和编解码时间对比,其中码率都为 0.2 bpp 保持不变。从仿真结果中可以看出利用本文算法峰值信噪比提高 0.2 dB,减少编解码时间且解码时间快于编码时间。

表 2 仿真结果

比较指标	本文算法	SPIHT
PSNR/ dB	32.9	32.7
编码时间/ s(近似)	2.14	2.86
解码时间/ s(近似)	2.02	3.01

4 结论

将本文算法应用在电力系统的远程图像监控系统中具有对图像的高压缩比、编解码速度快、图像质量好、抗干扰能力强等优点。仿真结果基本令人满意。因此,该方案在对远程图像监控系统中的图像压缩是非常有效的。

参考文献:

[1] Daubechies I, Sweldens W. Factoring Wavelet Transforms in

to Lifting Steps[J]. J Fourier Anal, 1998, 4(3):245-267.

[2] Sweldens W. The Lifting Scheme: A Construction of Second Wavelets[J]. J Math Anal (SIAM), 1996, 29(2): 511-546.

[3] Said A, Pearlman W. A New Fast and Efficient Image Coding Based on Set Partitioning in Hierarchical Trees[J]. IEEE Trans on Circuits & Systems for Video Technology, 1996, 6(3):243-250.

[4] 唐慧明,张健 (TANG Hui-ming, ZHANG Jian). 变电站远程图像监控系统设计 (Design of Vedio Surveillance System for Substation) [J]. 电力系统自动化 (Automation of Electric Power Systems), 2001, 25(7):56-58.

[5] Mallat S. A Theory for Multiscale Signal Decomposition: The Wavelet Representation [J]. IEEE Trans on Patt Anal and Mach Intell, 1989, 11(7):674-693.

[6] Antonioni M, Barlaud M, Mathieu P, et al. Image Coding Using Wavelet Transform[J]. IEEE Trans Image Processing, 1992, 1(2):205-220.

[7] Shapiro J M. Embedded Image Coding Using Zerotree of Wavelet Coefficients[J]. IEEE Trans on Signal Processing, 1993, 41(12):3445-3462.

[8] 周建鹏,杨义先 (ZHOU Jian-peng, YANG Yi-xian). 图像压缩编码中的小波变换技术 (Wavelet Transforming Technology in Image Compression Coding) [J]. 电路与系统学报 (Proceedings of Circuit and System), 1997, 2(2):58-65.

收稿日期: 2003-08-27; 修回日期: 2004-01-04

作者简介:

向泽智(1978-), 硕士研究生, 研究方向为图像压缩;

张兆宁(1964-), 博士后, 教授, 从事神经网络、小波分析理论与应用研究。

The SPIHT algorithm based on fast lifting wavelet transform in image compression

XIANG Ze-zhi, ZHANG Zhao-ning

(Civil Aviation University of China, Tianjin 300300, China)

Abstract: This paper presents the improved algorithm according to image compression technology to pledge the real time of the image transmission and gain the high compression ratio under the image quality. The improved SPIHT image coding algorithm based on fast lifting wavelet transform introduces fast lifting wavelet transform technology to improve transform course; because of many consecutive zeroes appearing in SPIHT quantification coding, the improved algorithm employs the simultaneous encoding of entropy and SPIHT with entropy coding adopting run length changeable coding. Experimental results show that encoding by this method can improve PSNR and efficiency. This method can be applied in the image data transmission and storage of remote image surveillance systems.

Key words: image compression; fast lifting wavelet transform (FLWT); entropy coding