

# 一种对残差帧的四叉树分解编码算法

王明辉 彭立中

(北京大学数学科学学院数学系, 北京 100871)

**摘要** 为了克服在低比特率下的块效应,并适应实时编、解码的需要,提出了一种对残差帧的四叉树编码算法,该算法首先采用OBMC的方法生成预测帧以有效消除块效应,并针对残差帧的统计与视觉特性,对每个 $2 \times 2$ 的小块取灰度平均值作为最小编码单位,然后用四叉树分解方法来大致标记均值大于某阈值的块的位置,并以标量量化和自适应算术编码器编码其灰度值.与基于变换的编码方法相比,该算法简单、高效、码率低、解码质量高,是一种实时的编、解码算法.

**关键词** 视频编码 四叉树分解 标量量化 重叠的块运动补偿

**中图分类号**: TN911.81 **文献标识码**: A **文章编号**: 1006-8961(2001)07-0648-05

## A Quadtree Coding Algorithm for Residual Frames

WANG Ming-hui, PENG Li-zhong

(School of Mathematical Sciences, Peking University, Beijing 100871)

**Abstract** In this paper we propose a novel quadtree decomposition coding algorithm for residual frames in video coding. First we employ overlapped block motion compensation to reduce blocking artifacts, this leads to smoother motion prediction frames which helps to encode the residual frames more efficiently; based on the nature of human visual system, we regard each  $2 \times 2$  subblock as a basic coding unit, compute its mean value, and use a new quadtree decomposition method to label the subblocks with larger magnitude than a chosen threshold, then code them with scalar quantizer and adaptive arithmetic coder. Compared with transform-based methods, such as EZW and H. 263, the proposed algorithm is more efficient and has better performance, which indicates that it is a very effective and computationally simple technique especially for low bit rate coding. Moreover, the coding and decoding procedures are so fast that it can satisfy real-time applications.

**Keywords** Video coding, Quadtree decomposition, Scalable quantization, Overlapped block motion compensation

## 0 引言

如今数字视频编码已在许多多媒体领域得到了广泛应用,如视频会议、可视电话、视频游戏中均占有中心地位.现行的一些国际视频编码标准MPEG1, MPEG2, H. 261, H. 263等都采用了混合的编码方法,即用预测编码来消除帧间的时间冗余,而用变换编码来消除帧内的空间冗余.

但在甚低比特率下,这些标准的编码效果往往

不够理想.由于它们都是基于分块的DCT变换,因此会产生严重的块效应,而影响视觉感受.近来比较热门的DWT变换虽然没有块效应,但在图象的边缘部分会产生振铃效应,此外大量的蚊式噪声也使得图象变得模糊<sup>[1]</sup>.这是由于运动补偿残差帧中各象素点的空间相关性已大大降低,统计表明,其一阶互相关系数 $r$ 一般为 $0.3 \sim 0.5$ ,这与自然灰度图象的 $r > 0.9$ 有本质区别<sup>[2]</sup>,即残差帧是完全非平稳的,因此DCT与DWT变换的基本图象无法很好地表示其特征.由于在低比特率的情况下,只能对一小

基金项目:国家自然科学基金资助(69735020,19872006)

收稿日期:2000-05-16; 改回日期:2000-10-25

部分变换系数编码,许多高频系数被丢弃,从而造成了上述现象.文献[2]认为,任何变换编码方法都不能对残差帧进一步去除相关性,而达到能量压缩的目的.

为了克服上述缺点,本文使用重叠的块运动补偿方法生成预测帧,其所得残差帧更光滑,因而可以很好地消除块效应;本文还设计了一种新的四叉树分解编码算法,直接对残差帧进行编码.实验表明,新算法简单、高效,与 H.263 和零树小波方法(EZW)<sup>[3]</sup>相比较,在甚低比特率的情况下,实现解码复现图象的主观和客观质量都有所提高.新算法运算速度快,对于 QCIF 格式的灰度图象序列,在 PC 机上就可以实时的编码和解码.

## 1 运动估算与运动补偿

在视频编码中,运动估算和运动补偿用来去除帧间的时间冗余.本文算法首先把当前帧分成大小为  $16 \times 16$  的不重叠块,然后以 SAD (Sum of Absolute difference) 为失真度量,在  $[-16, 15.5]$  的范围内,采用块匹配方法进行完全搜索(FS),得到每一块相对于参考帧的运动矢量.为了提高预测质量,使用了半像素点精度的运动矢量,这就要求首先对参考帧进行双线性插值<sup>[4]</sup>.

然而当相邻块的运动矢量不同的时候,会造成预测帧在块边界上的不连续性,由此引起残差帧的块效应,这将影响重构帧的视觉感受,而且对这些块边界的处理会大大降低编码效率,因此必须减少甚至消除块效应.采用重叠的块运动补偿(OBMC)方法来生成预测帧,就可以很好地消除块效应.

预测帧中的每一个象素点,都是5个预测值的加权和,而为了得到这5个预测值,就需要知道如下5个运动矢量:当前块(该象素点所在的块)的运动矢量( $MV_x^c, MV_y^c$ ),当前块上方块的运动矢量( $MV_x^u, MV_y^u$ ),当前块下方块的运动矢量( $MV_x^d, MV_y^d$ ),当前块左方块的运动矢量( $MV_x^l, MV_y^l$ )和当前块右方块的运动矢量( $MV_x^r, MV_y^r$ ).

设  $(x, y)$  为预测帧中某象素点的位置坐标,令  $m = x/16, n = y/16$ , 则  $(m, n)$  表示该象素点所在块处于预测帧中的位置.其中“/”代表整数除.

令  $i = x - m \times 16, j = y - n \times 16$ , 可得

$$(x, y) = (m \times 16 + i, n \times 16 + j)$$

这时预测帧中的每一点  $(x, y)$  的灰度值  $\bar{p}(x, y)$

可表示为

$$\begin{aligned} \bar{p}(x, y) = & C(x, y) \times H_c(i, j) + T(x, y) \times H_t(i, j) + \\ & B(x, y) \times H_b(i, j) + L(x, y) \times H_l(i, j) - \\ & R(x, y) \times H_r(i, j) \end{aligned}$$

式中,  $H_c, H_t, H_b, H_l, H_r$  是大小为  $16 \times 16$  的加权矩阵,其定义见文献[5].而  $C(x, y), T(x, y), B(x, y), L(x, y), R(x, y)$  是来自参考帧象素点的灰度值,定义如下:

$$C(x, y) = p(x - MV_x^c, y + MV_y^c)$$

$$T(x, y) = p(x + MV_x^t, y + MV_y^t)$$

$$B(x, y) = p(x + MV_x^b, y + MV_y^b)$$

$$L(x, y) = p(x + MV_x^l, y + MV_y^l)$$

$$R(x, y) = p(x + MV_x^r, y + MV_y^r)$$

其中  $p(\cdot, \cdot)$  表示参考帧中象素点的灰度值.

## 2 四叉树分解编码算法

残差帧是当前帧与预测帧的差值帧,也是运动补偿的误差帧,它主要反映前后两帧运动边界的信息.由于人眼对图象运动部分的空间灵敏度较低<sup>[5]</sup>,因此可以以每个  $2 \times 2$  的小块为编码单位,用其灰度平均值来代替;再者,因为残差帧只可能在运动边界、遮挡或暴露等区域出现较大的幅值,而同一场景中相邻帧的差别是很小的,所以大幅值的象素点的范围很小.此外,人类视觉具有对比灵敏度<sup>[6]</sup>,在亮度有细小差别时,只要不超过一定的辨别阈值,人眼就分辨不出来.利用这个特性,可以设立一个阈值  $T_0$ ,把幅值小于  $T_0$  的  $2 \times 2$  小块置为零,就会使大部分的小块灰度值为零.这时,如果可以有效地标记那些非零小块的位置,并对它们的灰度值进行量化和熵编码,就能既增加编码效率,又不影响重构帧的质量.

基于以上考虑,本文设计了一种针对残差帧的四叉树分解编码算法.它着眼于对残差帧中的高运动区域进行较粗的空间和灰度量化的,以提高运算速度和降低码率.

该算法首先把每个  $2 \times 2$  的小块看成一个单元,求其平均灰度值,该小块4个象素点的灰度值就用这个平均值来代替;然后根据该残差帧的统计特性,选定一个阈值  $T_0$ ,把幅值小于  $T_0$  的象素点置为零.经过这一步后,应使绝大多数的象素点为零,而少量的非零点则代表高运动区域、高对比度区域.

接下来需要把那些  $2 \times 2$  的非零小块的位置标

记出来,并对它们的平均灰度值进行标量量化和熵编码。这里采用了一种新的二叉树分解与编码方法。二叉树是一种简单的图象分级表示技术,曾成功应用于二值图象的压缩,它算法简单,易于硬件实现,可以根据内容进行自适应的区域分解,且分解结果导致某种形式的图象分割,这虽然利于其他形式的应用处理,然而直接对灰度图象应用二叉树方法进行压缩,其率失真性能明显低于基于变换的方法<sup>[7]</sup>,因此,类似于二值图象压缩,这里只用二叉树分解来大致标记非零小块的位置。

为了提高运算速度,减少内存开销,本文采用队列操作来实现二叉树的分解与编码过程。定义队列 WorkQuene,它的每一项表示二叉树的一个结点。用四元组  $(x_1, y_1, x_2, y_2)$  来表示,它代表残差帧中范围  $[x_1, x_2] \times [y_1, y_2]$  内的所有象素点的集合,如果这个集合中有非零值,那么就称该项(结点)是重要的;否则称为非重要的项。

二叉树分解与编码算法:

(1)初始化

把残差帧分成  $2^n \times 2^n$  的块,对于范围是  $[x_1, x_2] \times [y_1, y_2]$  的各块,以四元组的  $(x_1, y_1, x_2, y_2)$  形式加入到队列 WorkQuene 中,二叉树的分解过程就是对每个这样的块进行的。

(2)分解与编码

取出 WorkQuene 的第一项  $(x_1, y_1, x_2, y_2)$

分解:当  $x_2 - x_1 > 2$  时,若该项是重要的,输出符号 '1',并把以下 4 项  $\left\{ x_1, y_1, \frac{x_1+x_2}{2}, \frac{y_1+y_2}{2} \right\}$ ,  $\left\{ x_1, \frac{y_1+y_2}{2}, \frac{x_1+x_2}{2}, y_2 \right\}$ ,  $\left\{ \frac{x_1+x_2}{2}, y_2, x_2, \frac{y_1+y_2}{2} \right\}$  和  $\left\{ \frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}, x_2, y_2 \right\}$  分别加在队列的末尾。否则,该项是非重要的,输出符号 '0'。

编码:当  $x_2 - x_1 = 2$  时,表明该结点代表一个  $2 \times 2$  的小块,是编码的最小单位,应对其进行编码。若其平均灰度值为 0,则输出量化符号  $L_0$ ;若其平均灰度值不为 0,则使用下节的方法对其幅值进行量化,并输出量化符号  $L_n$  和正负符号(当大于零时,输出符号 '1',小于零时,输出符号 '0')。

(3)判断队列是否为空

如果此时队列 WorkQuene 为空,那么算法到此结束,退出运算;否则,转向步骤(2)。

算法中生成的 '0'、'1' 符号流以及量化符号流经过自适应算术编码器<sup>[8]</sup>熵编码后,得到输出码流。

需要注意的是,这里并没有完全确定所有非零  $2 \times 2$  小块的位置,实际上算法最多只对  $4 \times 4$  的小块判断是否进行分解,而对重要的  $4 \times 4$  小块的 4 个  $2 \times 2$  小块无论是否为零,都进行编码,这种做法是为了减少编码位置的比特数,从而大大提高了编码效率。

### 3 算法中阈值与量化层次的自适应选择

阈值与量化层次这两个参数的选择对于码率的大小和重构帧的质量好坏有重要影响。

阈值的大小决定了要编码的非零区域范围大小,反过来,也可以通过设定非零区域所占的比率  $r$  来控制阈值的大小。

$$r = \frac{N_1}{N}$$

式中,  $N_1$  为非零区域的象素点个数,  $N$  为该帧中全部的象素点个数。

阈值设定时,首先把残差帧中大于 255 的象素点幅值截取为 255,并求该帧象素点幅值的直方图  $Hist(i), 0 \leq i \leq 255$ 。令

$$T_0 = \arg \min \left\{ \sum_{i=0}^T Hist(i) \geq N \times (1-r) \right\}$$

其中,  $N$  为该帧的象素总个数,则  $T_0$  就是所求的阈值。

通过对不同类型的可视电话景物进行的测量可知<sup>[6]</sup>,在一帧的时间间隔内,大约只有  $< 4\%$  的象素有变化。由于在实际的视频编码过程中,是以上一帧的重构帧为参考帧的,因此这个比率有所增大,可以先设置一个初始值,如 8%,然后根据输出缓冲区的满溢情况,以及前几帧 PSNR 值的变化情况对该比率的值适当地加以调节,从而自适应的选取阈值  $T_0$ ,达到码率控制和质量控制的目的。

量化层次决定了对非零块灰度幅值的量化步长。设量化层次为  $Q_{Level}$ ,该残差帧中幅值  $g$  的最大值为  $g_{max}$ ,则量化步长  $Q_{Step}$  为

$$Q_{Step} = \frac{g_{max} + 1 - T_0}{Q_{Level} - 1}$$

该量化器把灰度值为零的块量化为符号  $L_0$ ,再把非零块的灰度幅值在区间  $[T_0, g_{max} + 1]$  内均匀量化,即把介于区间  $[T_0 + Q_{Step} \times (n-1), T_0 + Q_{Step} \times n)$ ,  $n \in [1, Q_{Level} - 1]$  的灰度幅值量化为符号  $L_n$ 。

由于人眼对于运动物体的分辨率比较低,因而对于非零块的灰度幅值不必很细的量化。可以规定:  $4 \leq Q_{Step} \leq 20, Q_{Level} \leq L$ 。在此条件下,对  $Q_{Step}$  从 4 开

始,每次加 4,直到条件得到满足为止.实验表明,当压缩比为 70~80:1 时,取  $L=8$  即可,当然, $L$  也可以根据  $T_0, Q_{step}$  的大小以及输出缓冲区的满溢情况适当地加以调节.

### 4 实验结果与分析

本文针对若干个 QCIF 格式的 8bit 灰度图象序列做了一些实验,为方便起见,本文的四叉树算法用 QuadTree 表示.

首先对 Carphone 序列的前 100 帧进行编码,以比较 QuadTree 与 EZW<sup>[3]</sup>算法的性能.编码时,每 20 帧用 EZW 方法压缩 16 倍做帧内编码,对其余各帧分别用 QuadTree 和 EZW 算法做帧间编码.其中,QuadTree 选择参数  $r=10\%, L=8$ ;截取 EZW 编码码流,使得两种算法每帧所用的比特数相同,实验结果如图 1 所示,其中 QuadTree 的平均 PSNR 为 31.18,而 EZW 算法的平均 PSNR 为 30.48,即 QuadTree 算法在码率相同的情况下,可以得到更高的客观图象质量.

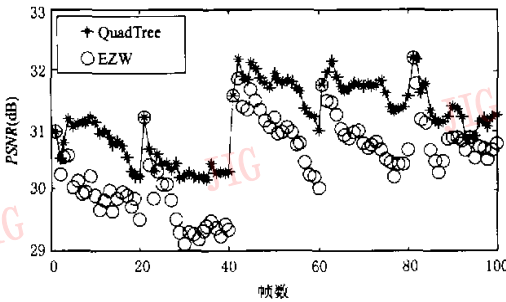


图 1 对 Carphone 序列的前 100 帧用 QuadTree 与 EZW 方法编码所得 PSNR 的对比结果

其次,对 Carphone 序列的前 21 帧,每一帧用原序列中的前一帧为参考帧,用块匹配方法做运动估算,用 OBMC 做运动补偿,得到当前帧的预测帧,残差帧就是它与当前帧的差值.设 QuadTree 算法参数  $r=3\%, L=8$ ,用 QuadTree 算法进行编码;同时,给出基于 TMN1.6 的 H. 263<sup>[9]</sup>编码器的编码结果,并通过调节参数  $Q_p$  的大小,使得两种算法对每一帧编码的比特数大致相同.表 1 显示了实验结果:20 个残差帧的平均编码字节数为 68.6,相当于 0.021 7bits/pixel;用 QuadTree 算法编码残差帧,所得重构帧的平均 PSNR 值为 35.39dB;用 H. 263

算法在相同码率下编码残差帧,所得重构帧的平均 PSNR 值为 35.12dB.由此可见,在高压缩比和相同比特率下,本算法与 H. 263 算法相比可以得到更高的图象质量.

表 1 QuadTree(四叉树)与 H. 263 算法在相同比特率下的对比结果

序列帧号	近似的编码 Byte 数	四叉树重构帧的 PSNR(dB)	H. 263 重构帧的 PSNR(dB)
2	79	33.15	33.29
3	79	34.43	34.53
4	89	35.43	35.11
5	61	35.67	35.18
6	91	37.59	37.46
7	69	34.45	34.20
8	108	36.21	36.00
9	63	34.63	34.29
10	71	35.40	34.98
11	59	35.14	34.64
12	54	34.77	34.49
13	49	36.01	35.55
14	54	35.57	35.27
15	64	36.44	36.22
16	50	34.67	34.02
17	54	36.05	35.68
18	71	36.82	36.71
19	69	35.57	35.30
20	51	33.90	33.60
21	87	35.99	35.83

表 2 给出了 QuadTree 与 H. 263 算法对多个不同的 300 帧图象序列在相同比特率下的对比结果,其中帧频为 10f/s,两种算法均每 20 帧用 H. 263 方法做一次帧内编码.

表 2 QuadTree 与 H. 263 算法对多个序列在相同压缩比下编码的对比结果

图象序列	比特率	H. 263 的 PSNR(dB)	QuadTree 的 PSNR(dB)
Container	17.4kbps	30.54	30.58
Akiyo	13.2kbps	32.31	32.20
Mother-daughter	13.2kbps	32.55	32.98
Silent-voice	19.0kbps	30.28	30.14
News	19.8kbps	29.52	29.56

由表 2 可见,这两种算法在实际编码时的平均 PSNR 值大致相同,但从主观质量来看,QuadTree 算法减少了块效应和振铃效应,而且在运动激烈的地方图象质量虽有所降低,但在运动缓慢的地方图象质量却得到提高.由于这符合人类视觉系统的特性,因而具有较好的视觉感受.图 2 给出了对 Carphone 序列编码时,第 70 帧的复现图象,其中,图 2(a)是原始图象,图 2(b)是 H. 263 的复现图象,图 2(c)是 QuadTree 的复现图象.



图2 Carphone序列第70帧的复现图像

最后在CPU时钟频率为300MHz的PC机上,测试了QuadTree算法的运算速度,采用二维对数搜索方法(2D Logarithmic search)<sup>[10]</sup>做运动估计,并对多个300帧的图象序列进行了编解码实验,表3显示了QuadTree、H.263与EZW算法编解码的速度,从表3可看出QuadTree算法的编解码速度比H.263、EZW算法有较大提高,完全可以满足实时应用的需要。

表3 3种算法对不同图象序列的编解码速度的对比结果

图象序列	H.263(f/s)		EZW(f/s)		QuadTree(f/s)	
	编码速度	解码速度	编码速度	解码速度	编码速度	解码速度
Container	12.76	29.53	7.97	18.75	20.60	48.23
Akiyo	12.71	29.94	8.17	18.30	20.70	46.94
Foreman	12.33	29.07	7.92	18.58	19.77	44.96
News	12.62	29.38	8.00	18.76	20.28	47.37

## 5 总结

本文提出了一种对残差帧进行四叉树分解的编码算法,即使用四叉树分解方法来标记高运动区域,并结合标量量化和自适应算术编码器进行编码,通过对人眼视觉特性与残差帧的统计特性的分析,设计了新的四叉树分解策略,并通过自适应地选择参数,达到了降低码率,增强图象质量的目的,新算法执行效率高,可以进行实时的编码与解码。

### 参考文献

- 1 John G Apostolopoulos, Nikil S Jayant. Postprocessing for very low bit-rate video compression. IEEE Trans on Image Processing, August 1999, 8(8): 1125~1129.

- 2 Peter Strobach. Tree-structured scene adaptive coder. IEEE Trans on Communications, April 1990, 38(4): 477~486.
- 3 Shapiro J M. Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans on Signal Processing, 1993, 41(2): 3445~3462.
- 4 ITU-T Recommendation H.263. Video coding for low bit rate. May 1996.
- 5 Ke Shen, Edward J Delp. Wavelet base rate scalable video compression. IEEE Trans on Circuits and System for Video Technology, 1998, 9(1): 109~121.
- 6 姚庆栋, 毕厚杰等. 图象编码基础. 北京: 人民邮电出版社, 1984.
- 7 Eli Shusterman, Meir Feder. Image compression via improved quadtree decomposition algorithms. IEEE Trans on Image Processing, March 1994, 3(2): 207~215.
- 8 Ian H Witten, Radford M Neal, John G Cleary. Arithmetic coding for data compression. Communication of the ACM, 1987, 30(6): 520~540.
- 9 <http://huizen.dds.nl/~roalt/h263.html>.
- 10 Jain J R, Jain A K. Displacement measurement and its application in inter-frame image coding. IEEE Trans on Communications, 1981, COM-29(12): 1799~1808.



王明辉 1973年生,1996年获北京大学数学科学学院信息科学系学士学位,现为北京大学数学科学学院博士生,研究兴趣为小波分析、图象压缩、视频编码。



彭立中 1944年生,瑞典斯德哥尔摩大学博士,现为北京大学数学科学学院教授,博士生导师,基础数学系主任,研究领域为调和分析、小波分析、图象处理等。