基于块截断编码和运动补偿的纯软件视频编解码算法

王业奎,涂国防
(中国科大研究生院电子学部,北京 100039)

摘 要: 本文报告了一种新的纯软件视频编解码算法. 利用改进的块截断编码和基于一步预测运动估计的运动补偿技术,该算法能够在现有的普通 PC 机上实时实现全彩色视频电话图像序列的编解码,比特率小于 100 Kbps. 与已有的其他同类算法相比,本文的算法在保持实时性的前提下,比特率大大下降,图像质量明显提高.

关键词: 纯软件视频; 块截断编码; 运动补偿; 亚抽样与插值
中图分类号: TN919.8 **文献标识码**: A **文章编号**: 0372-2112 (2001) 02-0275-04

Software-Only Video Codec Based on Block Truncation Coding and Motion Compensation

WANG Ye-kui, TU Guo-fang
(Dept. of Electronics, Beijing Graduate School of USTC, Beijing 100039, China)

Abstract: A novel software-only video codec is presented in this paper. By using modified block truncation coding and motion compensation technique based on one-step predictive motion estimation, the proposed codec can compress real-time videophone pictures below the bit-rate of 100 kbps on current PC. Compared to other similar algorithms, the presentation has substantially improved performance in bit-rate and PSNR at the premise of keeping the real-time property.

Key words: software-only video; block truncation coding; motion compensation; decimation/interpolation

1 引言

纯软件视频编解码是指在普通计算机上实现数字视频的压缩与解压缩,而无需特殊的硬件. 随着 VLSI 技术的快速发展,CPU 计算能力不断增强,纯软件视频越来越受到人们的重视,以代替依赖于专门的压缩卡和图形加速器的硬件方案. 限制现有的视频标准算法(如 MPEG-1、MPEG-2、MPEG-4、H.261 与 H.263)作为纯软件视频的瓶颈是计算复杂度过高. 为了达到纯软件视频的目标,目前有两种研究方向:一种是研究标准算法的快速实现[1];另一种是研究不同于标准算法的低复杂度算法,如利用游程编码的方法[2]、基于块截断编码(BTC)的方法[3,4]和逐像素的条件差分补充法[5]. 文献[3,4]中基于 BTC 的方法的压缩比不高,图像质量不够好;而文献[5]的算法只适用于慢运动低复杂度的图像序列,当运动速度与复杂度上升时,比特率会急剧上升.

本文提出了一种新的纯软件视频编码方法,称为基于块截断编码(BTC)[6,7]与运动补偿的纯软件视频编解码算法. 首先对 BTC 作了重要改进:(1)在多值量化的 BTC[4]与保护边缘的亚抽样算法[8]的基础上,提出了利用已重建数据的插值方法;(2)对 BTC 的量化重建值进行预测熵编码. 然后提出了一种基于三步搜索算法(3SS)[9]的一步预测搜索算法(1SPS). 本文提出的纯软件视频编解码算法可以在普通 PC 机上实时实现比特率小于 100 kbps 的全彩色的视频电话图像序列(CIF: 352 ×288、YUV420,10fps)的编解码,重建图像质量良好.

2 BTC 简介

BTC 是一种简单、快速的低复杂度图像编码算法,最初由 E.J. Delp 和 O. R. Mitchell 提出[6]. 自第一篇报道 BTC 的文章发表以后,已经有 600 多篇杂志或会议文章、40 多篇博士论文以及一本关于 BTC 的专著面世. 1987 年,BTC 曾作为 JPEG 压缩标准的侯选算法直到最后一轮.

BTC 的主要缺点是压缩比不高(原始 BTC 算法的压缩比为 4∶1). 但由于其实用性及快速性(比 DCT 等变换编码快至少 5 倍以上[3]),BTC 及其改进得到了广泛的应用,尤其是在基于软件的多媒体视频应用方面,如 SUN 公司的 CellB 视频格式[10]、Xmovie[11]以及 DEC 公司的 SMP[12].

在 BTC 中,首先将图像分解成 $n \times n$ 大小的互不重叠的块,然后这些块将分别由不同的二值量化器进行量化. 量化器的阈值与两个重建值由一个块的局部统计特性决定. 一般,阈值选为块均值,两个重建值分别是块中大于或小于阈值的所有像素的均值.

收稿日期:1999-11-29;修回日期:2000-05-15
基金项目:国家自然科学基金(No. 69771022)

3 改进的 BTC

为了后面叙述的方便,先给出几个定义:

活动性 任意像素集合 S 按 BTC 原理计算得到的两个量化重建值与均值之差的较大值,称为该集合的活动性,记为 ACT_S,如式(1)所示.

$$ACT_S = \max\{Mean - RL0, RL1 - Mean\} \quad (1)$$

其中 $RL0$ 与 $RL1$ 分别为较小与较大的重建值,$Mean$ 为集合 S 的像素均值,$\max\{P\}$ 表示取集合 P 中的最大值. 注意:$RL0 \le Mean \le RL1$.

N 值量化 重建的像素集合中最多有 N 个重建值,每一重建值都是块中某个像素子集的均值,简记为 QN. 单值量化(Q1)表示以集合均值 $Mean$ 作为块中所有像素的重建值;二值量化(Q2)将集合 S 按 $Mean$ 分为 $S0$ 与 $S1$ 两个子集,其中 $S0$ 为像素值小于 $Mean$ 的像素集合,$S1$ 为像素值大于 $Mean$ 的像素集合,$S0$ 与 $S1$ 的重建值($RL0$ 与 $RL1$)分别为各自的均值;四值量化(Q4)指分别将二值量化得到的 $S0$ 与 $S1$ 子集再进行二值量化,从而得到 $S00$、$S01$、$S10$ 和 $S11$ 四个子集,每个子集的重建值($RL0$、$RL1$、$RL2$ 与 $RL3$)都是该子集的均值;依次类推,可以得到 $Q8$、$Q16$ 等等.

水平梯度与竖直梯度 在一个 $m \times n$ 的图像块中,水平梯度 GH 与竖直梯度 GV 分别定义为式(2)和(3).

$$GH = \frac{1}{m(n-1)} \sum_{i=0}^{m-1} \sum_{j=0}^{n-2} |I(i,j) - I(i,j+1)| \quad (2)$$

$$GH = \frac{1}{(m-1)n} \sum_{i=0}^{m-2} \sum_{j=0}^{n-1} |I(i,j) - I(i+1,j)| \quad (3)$$

其中 $I(i,j)$ 为位置 (i,j) 处像素的亮度值.

(1)改进后 BTC 的编码方案

首先将灰度图像分成互不重叠的 16 ×16 大小的块,两个色度分量分成对应的 8 ×8 块(类似 H.263 等算法中的宏块划分),然后计算每个灰度块的活动性(ACT). 若 $ACT < th\_1$,进行 16 ×16 的单值量化,两个 8 ×8 的色度块也进行单值量化,记为 B16Q1;否则,继续将灰度块按四叉树法则分解为 4 个 8 ×8 块.

对每个 8 ×8 块(包括灰度块),也是先计算块的活动性 ACT,然后:

①若 $ACT < th\_2$,单值量化(B8Q1);

②若 $th\_2 \le ACT < th\_3$,二值量化(B8Q2);

③若 $ACT \ge th\_3$,计算四值量化(B8Q4)参数:若 B8Q4 中四个像素子集的活动性都小于 th_3,则四值量化(B8Q4),否则八值量化(B8Q8).

其中,th_1、th_2 与 th_3 为判决阈值.

(2)亚抽样与插值

为了提高压缩比,在多值(二值以上)量化前先进行亚抽样,解码时再进行相应的插值. 如果所有的块都采用同样的亚抽样方法[3,4,13,14],会使图像的边缘产生不同程度的失真. 文献[8]中的亚抽样方案对不同类型的块采用不同的亚抽样方法,以保护图像的边缘与细节. 本文的亚抽样与插值算法不仅能够保护图像的轮廓细节,而且可以在插值时利用已重建的相邻块图像数据,进一步提高图像质量. 具体方案如下:

①对 B16Q1 及 B8Q1 这两类活动性极低的平滑块进行单值量化,编码时不需要亚抽样. 为了减小单值量化引起的块效应,解码时只将右下方最大子块的值重建为均值 $Mean$;最左列和顶行的像素则通过利用 $Mean$ 与左、上二相邻块中已解码的数据进行插值(重建为相邻像素值的均值或中值)得到.

②对 B8Q2 这类活动性较低的块,采用 1/4 亚抽样,抽样矩阵如图 1(a)所示. 解码时,最左列和顶行的像素通过利用本块解码数据与左、上二相邻块中已解码的数据进行插值得到,然后其他丢弃的数据再由其在本块中已解码的相邻像素插值得到.

③对活动性较高的块(B8Q4 与 B8Q8),先计算 GH 与 GV. 若 GH 与 GV 都大于阈值 th_4,则表明块中有复杂边缘或细节,采用任何亚抽样方法都会产生失真,因而不进行亚抽样;否则,若 $GH < GV$,则表明块中只有水平边缘,采用如图 1(b)所示的 1/2 水平亚抽样,以保护水平边缘;若 $GH > GV$,则表明块中只有竖直边缘,采用如图 1(c)所示的 1/2 竖直亚抽样,以保护竖直边缘.

0 0 0 0	0 1 0 1	0 0 0 0
0 1 0 1	0 1 0 1	1 1 1 1
0 0 0 0	0 1 0 1	0 0 0 0
0 1 0 1	0 1 0 1	1 1 1 1
(a)	(b)	(c)

图 1 亚抽样示意图. 其中,1 表示保留的像素,0 表示丢弃的像素.

插值时,对于水平亚抽样块,丢弃的像素插值为左右二相邻像素的均值,其中最左列的像素插值时将用到左相邻块中已解码的数据;对于竖直亚抽样块,丢弃的像素插值为上下二相邻像素的均值,其中顶行的像素插值时将用到上相邻块中已解码的数据.

(3)量化值的编码

为了进一步提高编码效率,所有的量化值都进行预测熵编码. 方法如下:

①对于单值量化(B16Q1 与 B8Q1),编码块均值与其预测值之差,即 $\Delta Mean$. 预测方法为:(a)图像左边缘宏块(包括色度块)的均值预测为上方相邻宏块的均值,(其中左上角预测为 0),其他宏块均值预测为左相邻宏块均值;(b)灰度子块均值的预测方向如图 2 所示. 其中箭头指向的块为箭尾指向块的预测块. 图中给出了图像左上角三个宏块的情况. 左边缘宏块与图中左下宏块相同,其余与右上宏块相同. 图像左上角子块的均值预测为 0.

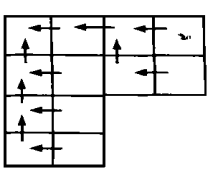

图 2 块均值的预测方向

②对于 B8Q2 块,先编码 $\Delta Mean$,然后编码 $Mean\_RL0$. 解码时,$RL1$ 由式(4)确定:

$$RL1 = (nn * Mean - (nn - n1) * RL0) / n1 \quad (4)$$

其中 nn 为亚抽样后块中像素的总数,$n1$ 是量化位图中 1 的个数.

③对于 B8Q4 块,先编码 $\Delta Mean$,然后依次编码 $Mean\_$

*RL*1、*RL*1-*RL*0、*RL*2-*Mean* 与 *RL*3-*RL*2. 这样可保证除 Δ*Mean* 外,其余都不小于 0,且有相对集中的分布.

④对于 B8Q8 块,先编码 Δ*Mean*,然后依次编码 *Mean*-*RL*3、*RL*3-*RL*2、*RL*2-*RL*1、*RL*1-*RL*0、*RL*4-*Mean*、*RL*5-*RL*4、*RL*6-*RL*5 与 *RL*7-*RL*6. 理由同上.

上述数据中,除 Δ*Mean* 外其余(记为 Δ*RL*)都用同一个 Huffman 码表进行编码. 对于视频电话图像序列(Missa 与 Claire),Δ*Mean* 和 Δ*RL* 的概率分布分别如图 3 和图 4 所示.

从图 3 和图 4 中可以看出,预测误差的分布相对集中,从而采用熵编码可以得到很高的压缩效率.

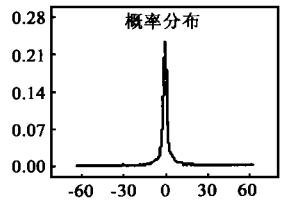

图 3　Δ*Mean* 的分布曲线

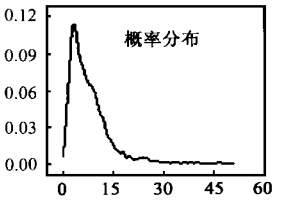

图 4　Δ*RL* 的分布曲线

以上 BTC 的改进方案既适用于图像的原始数据(帧内编码),也适用于 DPCM 数据(帧间编码). 二者有两个区别:(a)四个阈值参数不同;(b)由于帧间编码时均值与重建值本身的分布就与帧内的 Δ*Mean* 相似,因而不经过预测而直接进行熵编码.

4　运动补偿

运动补偿的目的是利用图像序列相邻帧间的位移矢量减少预测误差,从而提高预测编码的效率. 其方法是通过传送图像相邻帧(一般以 16 ×16 的宏块为计算单元)的位移矢量 D 和位移帧差 *DFD* 来补偿相邻帧间的变化. *DFD* 由式(5)决定:

$$DFD = I(i, j, t) - I(i + dx, j + dy, t - \tau) \qquad (5)$$

其中,*I* 为像素值,D = *dx* + *dy* 为当前帧相对于前帧在位置(*i*,*j*)处的位移矢量,*t* 为当前帧的时间,τ 为两帧之间的延时.

当 *DFD* 满足一定的条件时,可以只传送 D,而不需要传送 *DFD*. 本文采用类似 H.263 测试模型 TMN5[15] 中的条件,如式(6)所示.

$$\max\left\{ \left| \overline{DFD}(i) \right| , i = 0 \sim 15 \right\} < th\_5 \qquad (6)$$

其中 $\overline{DFD}(i)$ 为宏块的 16 个 4 ×4 子块中第 *i* 块的平均 *DFD*,*th*_5 为阈值.

运动补偿的关键是运动估计. 最常用的运动估计技术是块匹配算法(BMA). 直接采用 BMA 算法计算复杂度高,很难应用于软件实时编解码. 近年来,出现了许多关于 BMA 的快速算法,其中三步搜索算法(3SS)[9] 是应用最为广泛的一种.

基于视频电话图像序列的慢变特性,假定相邻帧的位移矢量在每个方向上的增量都不超过 1.5 个像素(该假定较好符合实际情况),如式(7)所示.

$$\left.\begin{aligned} | dx - dx' | \le 1.5 \\ | dy - dy' | \le 1.5 \end{aligned}\right\} \qquad (7)$$

其中,*dx* ′+ *dy* ′= D 为前帧的位移矢量. 这样,在运动估计时,首先将当前宏块的位移矢量 D 预测为前帧对应宏块的运动矢量 D′,然后进行 3SS 算法中的最后一步,即在(*dx*′,*dy*′)及其 8 个相邻位置进行搜索. 上述方法称为一步预测搜索算法(1SPS). 最后再按 TMN5 的方法进行半像素搜索.

5　实验结果

在 PC/Pentium-Ⅱ-350 计算机上模拟了本文的纯软件视频编解码系统(简称算法 *C*). 根据率失真曲线以及主观质量,依次确定了帧内及帧间个阈值参数的取值,结果如表 1 所示.

表 1　阈值参数的选取

参数	*th*_1	*th*_2	*th*_3	*th*_4	*th*_5
帧内	2	4	8	10	\
帧间	4	5	10	15	2

为了便于与文献[3]和[4]中的算法(简称算法 *A* 和 *B*)进行比较,表 2 给出了 *Miss*_*A* 灰度序列(QCIF,8 帧)的压缩比 *Cr* 和峰值信噪比 *PSNR*,其中 *PSNR* 的单位是 dB. 表 3 给出了本文算法对 Claire 灰度序列(QCIF,50 帧,30 帧/秒)及 Miss_*A* 彩色序列(CIF,50 帧,10 帧/秒)的结果. 对于彩色图像,由于压缩比定义可能有歧义,本文用编码比特率(比特/像素,bpp)来衡量其压缩效率. 图 5 给出了 Miss_*A* 原始图像及本文算法的解码图像.

从表 2 可以看出,本文算法无论在压缩效率还是在信噪比方面,都比其他同类算法好得多. 相对于算法 *A*,帧内编码的压缩比稍有提高,*PNSR* 则提高了 9.4dB;序列编码的压缩比提高了 50 %以上,*PSNR* 提高了 6.9dB 以上. 相对于算法 *B*,帧内编码的压缩比有所下降,*PNSR* 则提高了 4.9dB;序列编码的压缩比提高了 50 %以上,*PSNR* 提高了 1.5dB. 从表 3 可以看出,本文提出的纯软件视频编解码算法完全可以在普通桌面计算机上实现通信比特率小于 100kbps 的全彩色视频的实时编解码. 对于 10 帧/秒的 Miss_*A*(CIF)彩色序列,每帧净编、解码时间之和为 42ms,编解码只占了整个系统资源的 42 %;对于 30 帧/秒的 Claire(QCIF)灰度序列,每帧净编、解码时间之和为 6.4ms,编解码只占了整个系统资源的 19.2 %. 由图 5 可以看出,本文算法的解码图像的主观质量良好.

表 2　三种算法对 Miss_*A*(QCIF)灰度序列的结果

算法	*A*	*B*	*C*
Cr(帧内)	10.6	18.5	11.1
PSNR(帧内)	31.9	35.4	40.3
Cr(平均)	< 30.0	29.9	45.1
PSNR(平均)	< 30.0	35.4	36.9

表 3　本文算法对 Claire(QCIF)灰度序列与 Miss_*A*(CIF)彩色序列的结果

编码结果	平均编码比特率(bpp)	平均 *PSNR*(dB)	平均净编码时间(ms/frame)	平均净解码时间(ms/frame)	通信比特率(kbps)
Claire	0.074	33.65	2.6	3.8	54.8
Miss_*A*	0.093	37.53	19.2	22.8	91.9

6　结论

本文报告了一种基于块截断编码和运动补偿的纯软件视

图 5 Miss_A 的原始图像(左)与对应解码图像(右):
上为第 1 帧,中为第 25 帧,下为第 50 帧.

频编解码算法.该算法可以在现有的普通桌面计算机上实现全彩色视频电话图像的实时编解码处理,比特率小于 100kbps,图像质量良好.与现有的同类算法相比,压缩比提高 50 %以上,重建图像的信噪比提高 1.5～6.9dB.本算法可广泛应用于多媒体视频通信.

参考文献:

[1] P. Bahl, P. Gauthier, and R. Ulichney. Software-only compression, rendering, and playback of digital video [DB/OL]. http://www.digital.com/info/DTJK04/DTJK04SC.TXT.

[2] H. C. Huang and J. L. Wu. Novel real-time software-based video coding algorithm [J]. IEEE Trans. Consumer Electron., Aug. 1993, 39: 570 - 580.

[3] H. C. Huang and J. L. Wu. Real-time software-based moving picture coding system [J]. Signal Processing: Image Comm., 1994, 6: 173 - 187.

[4] 涂国防,王业奎. 自适应多值量化的亚抽样块截断编码 [J]. 电子科学学刊, July 1999, 21(4): 506 - 510.

[5] Y. J. Chiu and T. Berger. A software-only videocodec using pixelwise conditional differential replenishment and perceptual enhancements [J]. IEEE Trans. CSVT, Apr. 1999, 9(3): 438 - 450.

[6] E. J. Delp and O. R. Mitchell. Image compression using block truncation coding [J]. IEEE Trans. Comm., Sept. 1979 COM-27: 1335 - 1342.

[7] B. V. Dasarathy. Image data compression: block truncation coding [C]. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[8] 王业奎. 一种保护图像轮廓细节的自适应亚抽样和插值方法 [J]. 中国图像图形学报, Dec. 2000, 5(12): 1002 - 1005.

[9] T. Koga, K. Iinuma, A. Iijima, and T. Ishiguro, Motion-compensated interframe coding for video conferencing [A]. Proc. NTC81 [C], New Orleans, LA, 1981: C9.9.1 - 9.6.5.

[10] W. K. Pratt. Developing visual applications XIL: an imaging foundation library [Z], SUN, 1998.

[11] R. Keller, W. Effelsberg, and B. Lamparter. Xmovie: architecture and implementation of a distributed movie system [J]. ACM Trans. Info. Sys., Oct. 1995, 13(4): 471 - 499.

[12] B. K. Neidecker-Lutz and R. Ulichney. Software motion pictures [J]. Digital Technical Journal, Spring 1983, 5(2): 1 - 9.

[13] Y V R. Rao and C. Eswaram. A new algorithm for BTC image bit plane coding [J]. IEEE Trans. Comm., Jun. 1995, 43(6): 2010 - 2011.

[14] B. Zeng and Y. Neuvo. Interpolative BTC image coding with vector quantization [J]. IEEE Trans. Comm., Oct. 1993, 41(10): 1436 - 1438.

[15] Video codec test model [Z]. TMN5, ITU-T/SG15, Jan. 1995.

作者简介:

王业奎 1973 年 11 月出生,1995 年本科毕业于北京理工大学自动控制系,1998 年硕士毕业于中国科大研究生院(北京)通信与电子系统专业,现攻读信息与信号处理专业博士学位.主要研究兴趣为数字图像与视频压缩及传输等.

涂国防 中国科大研究生院(北京)电子学部教授、博士生导师.主要从事现代数字通信、数据压缩、图像处理、ISDN 等方面的研究与教学工作.