

一种适用于 H. 264 的时域差错掩盖改进算法

安维嵘 张旭东

(清华大学电子工程系多媒体信号处理实验室, 北京 100084)

摘要 视频压缩码流在信道传输时, 由于受到信道带宽或者稳定性的影响, 容易发生数据的损坏或者丢失, 这样不仅会对当前的视频帧产生影响, 而且差错会延续到随后的视频帧, 因此, 需要采用某种技术来降低差错的影响。针对这一问题, 在对最新视频压缩标准 H. 264 研究的基础上, 基于 H. 264 标准的框架, 对已有的差错掩盖算法进行了改进, 提出了适合 H. 264 编码标准的时域子块匹配差错掩盖算法。该算法首先采用 8×8 的子块代替 16×16 的宏块, 作为差错掩盖的运算单元, 然后对不同的子块采用不同的边界像素, 利用边界匹配算法, 并通过改进的 $1/4$ 像素精度菱形搜索法在参考帧内找到最佳匹配块。实验结果证明, 由于该算法有效地利用了 H. 264 压缩码流里的信息, 因此, 同传统的时域差错掩盖算法相比, 对差错信号有更好的恢复效果。

关键词 差错掩盖 H. 264 视频压缩 边界匹配

中图分类号: TN919.81 文献标识码: A 文章编号: 1006-8961(2004)09-1124-06

A Modified Temporal Error Concealment Algorithm Designed for H. 264

AN Wei-rong, ZHANG Xu-dong

(Department of Electronic Engineering, Multimedia Signal Processing Lab, Tsinghua University, Beijing 100084)

Abstract When compressed video stream is transmitted over internet or wireless network, it is sensitive to packet loss and error occurrence due to the limited bandwidth and the instability of the channel, especially for motion compensation-based video coding standard. An error in one frame not only debases the reconstructed quality of the current frame, but propagates to the succeeding frames which use current frame as their reference frame. As a result, error concealment algorithms are developed for hiding erroneous blocks by exploiting high correlation in video sequence. In this paper, a new temporal error concealment algorithm which suits for H. 264, the latest international video coding standard, is proposed. In this algorithm, an erroneous macroblock is firstly divided into four 8×8 blocks, which work as the concealment unit instead of the macroblock. Then, a modified boundary matching algorithm is exploited to recover the lost motion vectors with different boundary pixels for different blocks. Finally, the best-matched blocks are obtained using one-quarter pixel diamond search method. Experimental results show that the proposed algorithm makes use of the information in H. 264-compressed bit stream efficiently, and has a better reconstruction quality for erroneous blocks than conventional temporal error concealment algorithms.

Keywords error concealment, H. 264, video coding, boundary matching

1 引言

视频压缩码流在 Internet 或无线信道传输时, 由于受信道带宽及信道稳定性的影响, 会发生数据包的丢失或误码, 尤其是在基于运动补偿的视频编码标准中, 一帧数据中发生的错误不仅影响到当前帧的图像重建质量, 而且会延续到随后的预测帧, 从

而使得大量的视频信号发生错误。这就需要采用某种差错恢复方法来恢复错误的视频信号。

进行差错恢复的一种简单有效方法是在解码端引入差错掩盖技术^[1]。所谓差错掩盖技术就是利用视频信号的时域或者空域相关性, 对损坏或者丢失的视频信号进行恢复的一种技术, 其包括空域差错掩盖和时域差错掩盖技术两种, 其中, 空域差错掩盖技术主要利用了视频序列的空域相关性, 它是通过

有效邻域的空间插值算法来估计出当前丢失的宏块;时域差错掩盖技术则主要利用了视频序列的时域相关性,首先通过某种算法估计出丢失的运动矢量,然后用参考帧中相应的宏块代替当前丢失的宏块。由于差错掩盖技术工作在解码端,并不需要增加额外的码流开销,所以在视频数据的差错恢复中,差错掩盖技术得到了广泛的应用。

本文中采用时域差错掩盖技术,针对最新的国际视频压缩标准 H. 264 的特点,提出了基于 8×8 块的边缘匹配预测算法,由于该算法充分利用了 H. 264 码流中提供的相关信息,并通过改进的 $1/4$ 像素菱形搜索算法和边缘匹配算法找到最佳匹配块来作为丢失块的估计,因此对差错信号有更好的恢复效果。实验结果验证了该算法的优越性。

2 时域差错掩盖技术

时域差错掩盖技术是差错掩盖算法中采用最多的一种方法,该技术的关键是如何合理地估计出错误宏块的运动矢量。其中最简单的算法是把运动矢量设为零,即直接用上一帧的同一位置的宏块来替换当前受损宏块,称之为时域直接替换法。时域差错掩盖技术一般的预测方法见图 1。



图 1 运动矢量的估计

当前受损宏块运动矢量记作 V_C ,周围宏块的运动矢量相应的记为 $V_{A_{left}}, V_A, V_{A_{right}}$ 等。 V_C 的值要根据周围宏块的状态来决定。当 A, B, L 宏块未出错时,则将该 3 个宏块的运动矢量加权平均作为当前宏块的预测值;如果 A, B, L 都有错误,则由当前宏块周围可能尚未受损的宏块,根据不同的算法插值得到当前块的运动矢量;当周围的宏块都损坏时,则可以将前一帧的同一位置处宏块的运动矢量作为当前宏块的运动矢量。

这种算法在后面的比较中用到,本文取周围正确宏块的平均运动矢量作为当前宏块的运动矢量,称之为邻域均值法。

3 H. 264 视频编码中的差错掩盖

3.1 最新视频压缩标准 H. 264^[2]

H. 264 视频压缩标准是在 H. 263 的基础上改进而来,但它仍然是基于运动估计、运动补偿的视频压缩算法。该标准除了保留了 H. 263 标准的框架外,还对压缩的具体细节做了大量的改进^[3,4]。

H. 264 码流由网络适配层(network adaptive level, NAL)和视频编码层(video coding level, VCL)两部分构成,其中 VCL 层是以片(slice)为单位的,每一个 VCL 加上 NAL 头,即构成一个 NAL 单元,其中,NAL 层既作为码流的同步信号,又包含了片头信息。H. 264 中仍采用熵编码方法,这样当视频码流中有一个 bit 发生了错误,就将导致以后的码流一直错误,直到下一个同步信号,所以在 H. 264 中数据的丢失仍是以片为单位对齐的。

H. 264 采用了更小的块进行运动估计和运动补偿,其允许使用 $4 \times 4, 4 \times 8, 8 \times 4, 8 \times 8, 8 \times 16, 16 \times 8$ 和 16×16 等 7 种自适应可选块大小。这样不仅进一步减小了残差信号的大小,同时提供了更多可选择的运动信息。

H. 264 采用 $1/4$ 像素运动估计来提高运动估计的精度,该方法既降低了码率,又提高了视频图像的质量。

H. 264 采用了 4×4 的整数变换来代替传统视频编码中的离散余弦变换(DCT),由于该变换全部采用定点运算,因此一方面提高了运算速度,另一方面消除了对 DCT 小数部分进行截取带来的误差。

H. 264 采用了多帧运动估计来代替原来的单帧运动估计算法,由于采用该方法,使得搜索的范围比以前更大,因此预测精度得到进一步提高。

本文提出的差错掩盖算法,就充分利用了 H. 264 的上述特性。

3.2 子块时域匹配搜索算法

文献[5]提出采用边界搜索的方法进行差错掩盖,该方法在 H. 263 视频压缩码流中取得了较好的视频恢复效果。基于该算法,本文根据 H. 264 视频压缩标准的新特点,提出了适合于 H. 264 视频编码的边缘匹配预测算法,并通过实验验证了其优越性。

本文算法采用了基于 8×8 块的时域匹配搜索算法。之所以采用 8×8 的块,而不用 4×4 的块,是因为考虑到算法的复杂性和对系统实时性的要求, 8×8 的块大小既提高了块的预测精度,同时在运算

量上又可以满足实时性的要求。

算法具体描述如下:

当发现某一宏块发生错误时,则按以下步骤进行操作:

- ①将宏块分成4个 8×8 的块。
- ②对每一个块,根据该宏块周围块的运动信息,按某种准则,选择适合当前块的初始运动矢量。
- ③对不同的块,选择不同的匹配边界点,并将该边界点作为下一步匹配搜索中计算绝对帧差和(sum of absolute difference)SAD的参考点。
- ④以初始运动矢量确定的位置作为搜索的中心,采用 $1/4$ 精度菱形搜索算法来找到满足最小SAD时的运动矢量,并将其作为最佳运动矢量。

⑤插值得到 $1/4$ 精度最佳预测块。

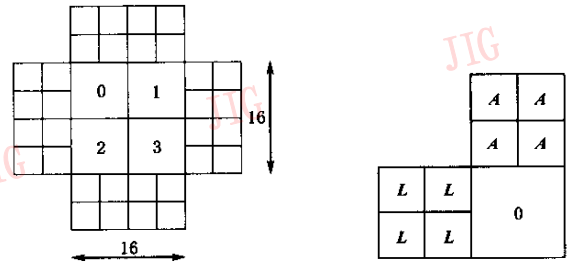
⑥用最佳预测块代替当前的错误块,并把得到的最佳运动矢量保存,作为其他块的参考运动矢量。

下面,对以上步骤的几个关键问题做详细的解释。

(1) 初始运动矢量的选择

由于在H.264中,每一个 4×4 的块都可能有自己的运动矢量,因此,初始运动矢量有多种选择方法。在本文中,对每一个 8×8 的块(如图2(a)所示的0,1,2,3),都采用其邻近正确块的平均运动矢量作为其初始运动矢量。以0块为例,如图2(b)所示,首先,将4个A块(4×4)的运动矢量进行平均,记为 \bar{V}_A ,同样,将4个L块(4×4)的平均运动矢量记为 \bar{V}_L ,然后判断A块和L块的运动矢量是否指向同

一个参考帧,如果是,则0块的初始运动矢量为 $V_0 = (\bar{V}_A + \bar{V}_L)/2$,否则, \bar{V}_A 和 \bar{V}_L 都作为当前块的初始运动矢量。其他3个块的预测方法与此相似。



(a) 分块信息

(b) 0块初始运动矢量选择

图2 初始运动矢量的选择

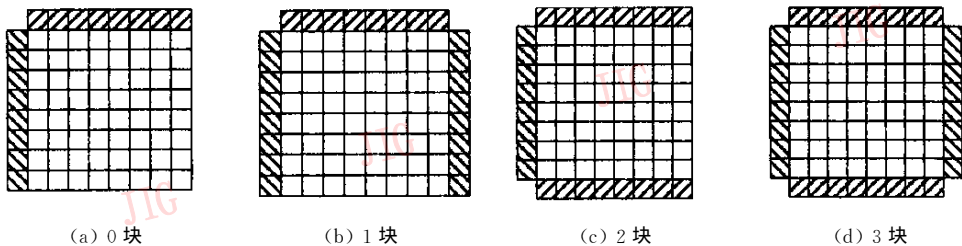
(2) 块匹配检测算法

根据4个块的位置不同,其匹配边界的选择也不同。对于0块(图2(a)中所标示),边界像素采用该块左侧和上侧的16个像素,见图3(a)中阴影部分,其他3个块的边界,分别如图3(b)、(c)、(d)所示。之所以这样选择不同的匹配边界点有以下3点原因:

①保证每一个块的边界在块匹配检测时是可以得到的;

②保证4个块在进行匹配时,相互之间能最佳匹配,以避免每一个块检测到最佳位置,但块与块之间存在块效应的问题;

③利用与当前块相关性最大的像素集作为搜索边界,理论上可以得到最佳匹配的预测块。



(a) 0块

(b) 1块

(c) 2块

(d) 3块

图3 块匹配边界选择

在确定了匹配边界后,再以初始运动矢量为中心,寻找最佳运动矢量。搜索时,对每一个块,在参考帧内计算以下边界的绝对帧差和SAD,

$$SAD = (k_1 \sum_{i=0}^7 |C(x_0 + i, y_0 - 1) - P(x_0 + v_x + i, y_0 + v_y - 1)| + k_2 \sum_{i=0}^7 |C(x_0 - 1, y_0 + i) - P(x_0 + v_x - 1, y_0 + v_y + i)| +$$

$$k_3 \sum_{i=0}^7 |C(x_0 + 8, y_0 + i) - P(x_0 + v_x + 8, y_0 + v_y + i)| + k_4 \sum_{i=0}^7 |C(x_0 + i, y_0 + 8) - P(x_0 + v_x + i, y_0 + v_y + 8)|) / (k_1 + k_2 + k_3 + k_4) \times 8$$

其中, $C(x, y)$ 表示当前帧的像素值, $P(x, y)$ 表示参考帧的像素值, (x_0, y_0) 是当前受损块的左上角像

素坐标, (v_x, v_y) 是预测运动矢量, k_1, k_2, k_3, k_4 是开头参数, 与图 3 的阴影直接对应, 其值如表 1 所示。

表 1 k_1, k_2, k_3, k_4 值的确定

	k_1	k_2	k_3	k_4
块 0	1	1	0	0
块 1	1	1	1	0
块 2	1	1	0	1
块 3	1	1	1	1

将满足最小 SAD 的运动矢量 (v_x, v_y) 作为最佳运动矢量。

(3) 1/4 像素精度菱形搜索算法

为了在降低搜索复杂度的同时, 仍能保证搜索的精度, 本文采用了 1/4 像素精度菱形搜索算法进行搜索, 该算法是先通过对整形像素位置进行菱形搜索^[6,7]来找到最佳的整数点位置, 然后再以该点为

中心, 对 1/4 像素位置进行全搜索, 以找到最佳的分数预测位置, 搜索过程如图 4 所示, 即首先, 以上面得到的初始运动矢量所指的位置作为搜索的中心位置, 检测菱形的 9 个像素点, 如图 4(a) 所示, 如果此时 SAD 最小的位置在中心, 则检测图 4(b) 中的 4 个像素点, 并将找到的 SAD 最小的位置作为最佳整数点位置。

否则以当前 SAD 最小的点为中心, 搜索新的菱形位置(如图 4(c) 所示), 直到 SAD 最小的位置在菱形中心为止, 然后, 以该点为中心, 再搜索图 4(b) 中的 4 个点, 以便找到最佳整数点位置。

最后, 以最佳整数位置为中心, 插值得到的 1/4 像素位置的像素值如图 4(d) 所示, 并用全搜索算法找到最佳的分数位置。

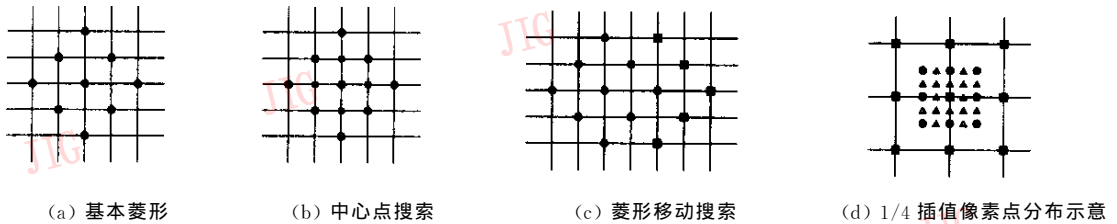


图 4 1/4 菱形搜索法

图 4(a)、图 4(b)、图 4(c) 的点都是整像素位置点, 其中圆点为第 1 步的搜索点, 方块为第 2 步的搜索点, 图 4(d) 中黑方块为整数点位置, 圆点为半像素点位置, 三角为 1/4 像素位置。

4 实验结果

为了评测本文提出的差错掩盖算法的性能, 使用 Container、Foreman、Silent 和 Susie 4 个标准 QCIF (176×144) 测试序列进行了不同算法的对比实验。实验以 H. 264 Jm61e 作为测试平台, 熵编码模式采用了基于内容的自适应二进制算术编码(context-based adaptive binary arithmetic coding), 并采用 1/4 像素精度运动矢量, 帧率为 25 fps, 每一帧视频图像分为 9 个片, 即每一行宏块为一片。

各测试序列的空域插值法、时域直接替换法、邻域均值法、本文方法等各算法重建图像的峰值信噪比(PSNR)见图 5。

从图 5 中可以看出, 本文的差错掩盖算法对不同的测试序列所表现的性能不同。

其中, Container、Silent、Susie 视频序列是背景无变化的视频序列, 三者的区别为 Container 序列

中只有物体的缓慢的平移, 并且运动的幅度很小; Susie 是中等速度的运动, 而且运动的方向是随机的; Silent 序列中有一个动作幅度比较大的运动。

Foreman 序列中, 不但存在背景的抖动和变化, 而且前景也发生变化。

从图 5 可以看出, 在 Container 序列中, 时域直接替代法、邻域均值法及本文所提出的方法重建图像的峰值信噪比曲线基本重合, 这说明在运动幅度很小的平移运动过程中, 三者性能的优劣相近。

但是, 在运动较大情况下, 尤其是存在背景变化和抖动情况下, 本文提出的算法性能明显提高, 这是因为, 本文算法采用了更小的搜索块, 并根据块的位置确定了不同的边缘区域, 从而使得搜索到的块和当前丢失的块更接近; 同时, 由于本文算法采用了菱形搜索法, 扩大了运动搜索的范围, 因而符合大运动位移的视频特性。对 Foreman 序列, 同邻域均值法相比, 本文算法的峰值信噪比值提高了 0.4~1.2 dB。

下面, 以 Foreman 序列的第 15 帧为例, 从主观视觉角度对本文的算法和空域插值法、时域替代法、邻域均值法做比较。数据丢包率设为 45%, 重建图像如图 6 所示。

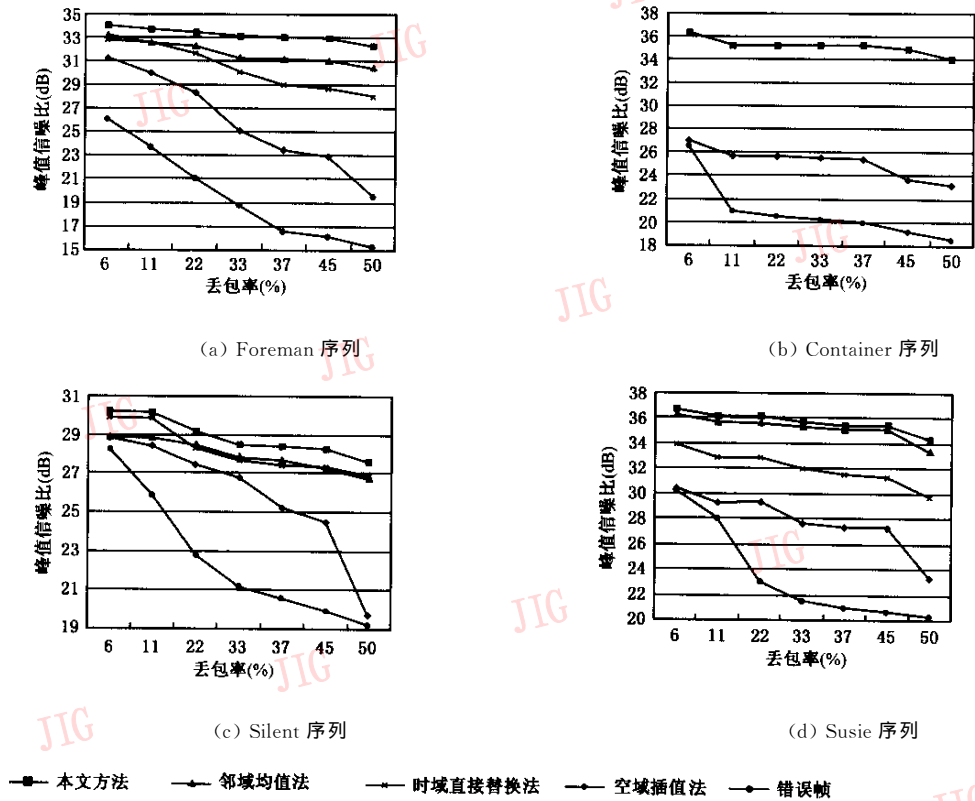


图 5 各掩盖方法峰值信噪比曲线

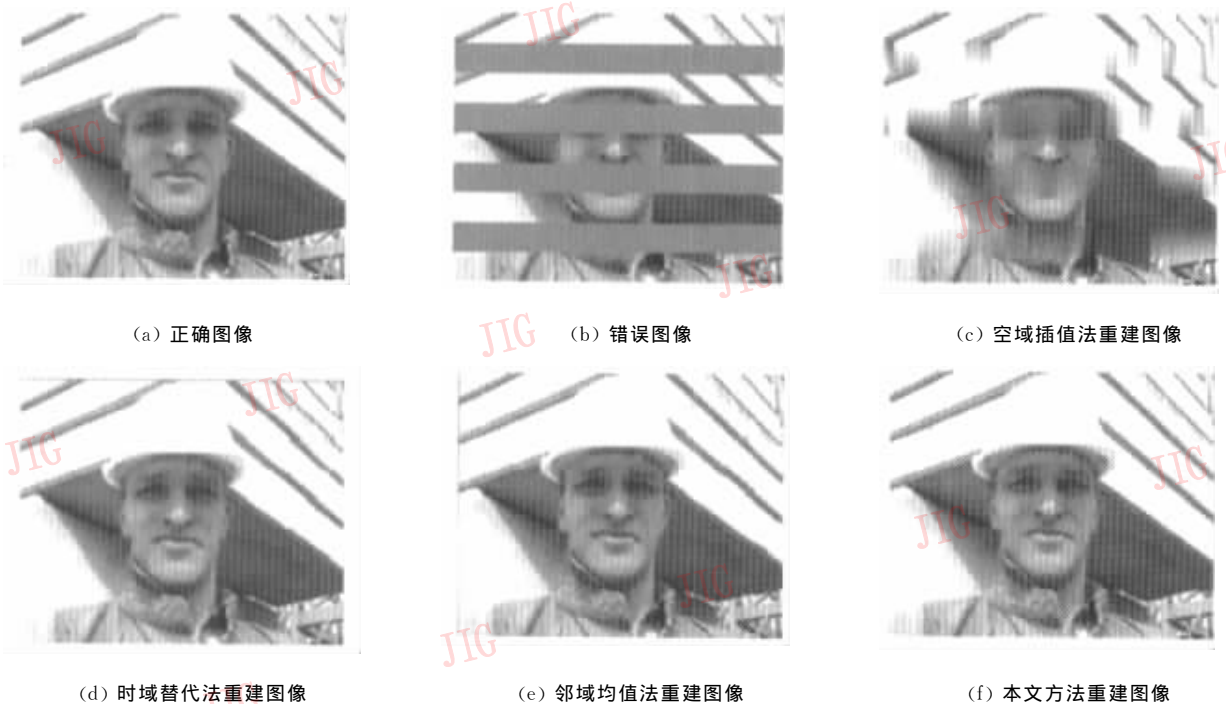


图 6 不同算法重建的 Foreman 第 15 帧图像

从图 6 可以看出,时域替代法和邻域均值法重建的视频图像,在背景处有明显的块效应,而本文的方法却减轻了块效应。

由于本文算法引入了 $1/4$ 菱形搜索算法,因此在提高差错恢复效果的同时,也增加了算法的复杂度。通过上文算法的描述可以看出,本文算法同邻

域均值法比较,复杂度主要增加在边界像素的插值和 1/4 菱形搜索算法上,而这两部分同块的 1/4 插值相比,运算量很少。下面以 Foreman 的第 15 帧为例来比较一下算法的复杂度。试验平台仍采用本实验室优化的 Jm61e 解码器平台,在 P3 700MHz, 128M 内存的 PC 上进行测试。在 Release 模式下,解码 QCIF 序列, I 帧平均需要 10~15ms, P 帧和 B 帧需要 20~25ms。若丢包率设为 45%, 则各算法耗时比较结果见表 2。

表 2 各种算法耗时统计

	算 法				
	无差错	空域插值法	时域替代法	邻域均值法	本文方法
P 帧时间 (ms)	20~25	25~30	25~30	25~30	28~33

由表 2 可以看出,如果所有帧都发生错误,那么采用本文算法,则仍可以达到 QCIF 的实时解码速度(25~30 fps)。在丢包率降低,配置更高的 PC 上,时间开销会进一步降低。

5 结 论

本文根据最新视频压缩标准 H. 264 的新特点,提出了子块边缘匹配差错掩盖算法,由于该算法充分利用了 H. 264 标准中新信息,并将 1/4 精度菱形搜索算法引入到差错掩盖中,因此提高了差错恢复的效果。

由于 H. 264 的标准还在不断地完善和改进中,因此如何进一步结合 H. 264 的算法特点来提高视频的差错恢复性能,以进一步降低因信道带宽和信道噪声而引起的视频传输差错,仍然值得作更深入的研究。

参 考 文 献

1 Wang Yao, Zhu Qin-fan. Error control and concealment for video communication: A review [J]. Proceedings of the IEEE, 1998, 86(5): 974~996.

2 ITU-T SG16. Draft ITU-T recommendation and final draft international standard of joint video specification[S/OL], 14496-10 AVC, ISO/IEC, ftp://standard.pictel.com/video-site/jvt-experts/2003-05-Geneva/JVT-G050r1.zip, 2003~5.

3 Wiegand Thomas, Sullivan Gary J, Bjontegaard Gisle, et al. Overview of the H. 264/AVC video coding standard[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 560~576.

4 张旭东, 卢国栋, 冯健编. 图像编码基础和小波压缩技术——原理、算法和标准[M]. 北京:清华大学出版社, 2004.

5 梁柱, 王兆华. H. 263 视频编码流的时域错误掩盖[J]. 中国图象图形学报, 2002, 7A(11): 1204~1208.

6 Zhu S, Ma K K. A new diamond search algorithm for fast block-matching motion estimation[J]. IEEE Transactions on Image Processing, 2000, 9(3): 525~529.

7 Tourapis Alexis M, Au O C, Liou Ming L. Predictive motion vector field adaptive search technique (PMVFAST)—enhancing block based motion estimation[A]. In: Proceedings of Visual Communications and Image Processing 2001 (VCIP-2001) [C], San Jose, CA, USA January, 2001: 883~892.



安维嵘 1979 年生, 2001 年获北方工业大学电子工程系工学学士学位, 现为清华大学电子工程系信息与通信工程专业硕士研究生。主要研究方向为视频压缩、差错控制、高速信号处理。

E-mail: awr01@mails.tsinghua.edu.cn



张旭东 1962 年生, 先后获山东大学电子系理学学士和清华大学电子工程系工学硕士和博士学位, 曾任香港理工大学短期研究员一年, 现任清华大学电子工程系副教授, 信号处理与网络传输研究所副所长。主要研究方向为多媒体信号处理、小波理论及应用、统计信号处理、高速数字信号处理系统。发表学术论文 50 余篇, 其中 20 余篇被 SCI 或 EI 检索。

E-mail: zhangxd@mail.tsinghua.edu.cn