

部分反向可解比特流(PBDBS)的自适应切换点定位

高绍帅, 涂国防

(中国科学院研究生院信息科学与工程学院, 北京 100039)

摘 要: 本文针对部分反向可解比特流(PBDBS)^[1]中的切换点定位问题, 提出了一种自适应定位方法, 即比特位置法. 本方法与 H.263 编解码器和 PBDBS 方法作了比较. 实验证明, 此方法可取得较好的容错性能, 可应用于公共交换电话网(PSTN)和无线信道上的视频传输.

关键词: 部分反向可解比特流; 容错编码; 错误掩盖; 鲁棒视频传输; H. 263

中图分类号: TN911. 22 **文献标识码:** A **文章编号:** 0372-2112 (2004) 04-0562-05

Adaptive Localization of Switch Point of Partial Backward Decodable Bit Stream (PBDBS)

GAO Shao-shuai, TU Guo-fang

(School of Information Science & Engineering, Graduate School of Chinese Academy of Science, Beijing 100039, China)

Abstract: This paper puts forward an adaptive localization method that can resolve switch point localization problem in the partial backward decodable bit stream (PBDBS), e.g. bit position method. And we compare this method with the H.263 codec and the original PBDBS method. It has been found that this method can obtain better error resilient performance and can be applied in video transmission over Public Switch Telephone Network (PSTN) and wireless channel.

Key words: PBDBS; error resilient coding; error concealment; robust video transmission; H.263

1 引言

H.263^[2]压缩的视频信号在有噪声干扰的信道(如无线信道)上传输时, 会出现比特错误. 由于使用了可变长编码(VLC)与运动补偿(MC), 使得出现比特错误后解码器失去同步, 直到下一个同步码的出现, 从而导致一连串宏块(MB)丢失, 并将扩散到预测帧(P帧)直到下一个帧内帧(I帧), 才能停止错误扩散. 因此如何限制因解码器失去同步造成的错误扩散就成了一个非常重要的问题.

解决错误扩散的一个有效的方法是利用可逆变长码(RVLC)^[6~9], RVLC可以使编码流在两个方向上可解. 当传输比特流出现错误时, 解码器会自动寻找下一个同步码, 并反向解码, 从而降低错误扩散的程度. 这种方法已被包含在 H.263 + , H.263 + +^[3,4]以及 MPEG-4^[5]标准中. RVLC的缺点^[9]有: (1)降低编码效率; (2)增加了码字的错误敏感性; (3)增加了解码的复杂度.

我们提出一种新的限制错误扩散的方法^[1], 即部分反向比特流方法, 简称 PBDBS 方法. 这种方法可使同步码在两个方向上同步, 大大提高了同步码的效率, 与原始的 H.263 编解码器相比, 在同等比特率的情况下, 码流的容错能力大大增强.

本文针对 PBDBS 方法的切换点定位问题, 提出了一种解决办法, 即比特位置法. 此方法可自适应地改变切换点的位置, 从而使视频比特流整体错误率降低, 进一步降低了错误扩散的程度, 容错性能有所增强. 同时我们将比特位置法同 H.263 和 PBDBS 方法比较, 实验证明, 比特位置法可取得比 PBDBS 方法更好的主观质量和客观质量.

2 H.263 语法结构与错误扩散

H.263 语法结构可分成四个层次: 图像层、块组层、宏块层和块层. 视频序列中每帧图像可以使用帧内编码模式(第一帧只能使用这种模式), 也可以使用帧间编码模式(使用运动补偿预测). 每帧图像可划分为一系列的块组, 通常每个块组包含一行宏块, 每个宏块由 16×16 个像素组成, 宏块包括 4 个亮度块和两个色度块, 每个块均由 8×8 个像素组成. 在 H.263 建议中, 每个宏块又可以使用两种编码模式: 帧内编码模式和帧间编码模式.

每个块组都可以有一个块组头(可选项), 块组头中包含有同步码——块组起始码(GBSC), 当编码比特流出现错误后, 由于使用了可变长编码(VLC), 会导致后面的比特流不可解, 直到下一个同步码的出现. 解码器会自动向后寻找同步

收稿日期: 2002-02-25; 修回日期: 2003-03-20

基金项目: 国家自然科学基金(NO. 60172041; NO. 6027056); 国家 863 计划(NO. 2002AA121070)

码,并继续解码,而出现错误宏块及之后的宏块则被丢掉,从而导致错误扩散.另外由于使用了运动补偿(帧间编码模式),错误的宏块还会扩散到后面的图像帧,直到下一个帧内帧的出现.这些都会使图像质量严重退化,因此有必要限制错误的扩散.RVLC 有效地解决了使用 VLC 导致的扩散问题,但存在降低编码效率等缺点,促使我们去研究一种更为简单易行的方法.

3 部分反向比特流(PBDBS)方法

为了降低因使用 VLC 导致的错误扩散,我们提出了一种新的方法,即部分反向比特流(PBDBS)方法,这种方法是将块组分成两个部分,每个部分都包含整数个宏块,第一部分使用原始的编码方法;第二部分用原始方法编完后,将比特流反向放置,使其可以反向解码.具体编解码方法如下.

3.1 编码器

PBDBS 方法的编码过程:首先使用原始的 H.263 编码器对块组中的宏块进行编码.在编完块组中的一半宏块时,将其比特位置记为 j ;在编完块组中所有宏块时,同样要将其比特位置记为 k .然后将从比特位置 $j+1$ 到 k 的所有比特逐个翻转放置,如图 1 所示.

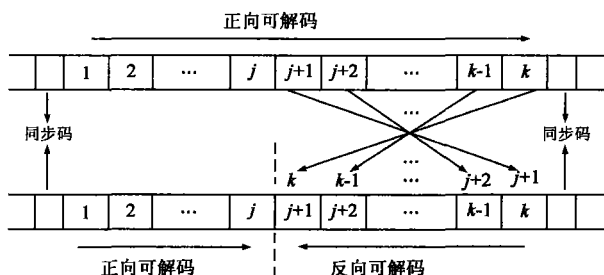


图 1 PBDBS 方法的编码过程

3.2 解码器

在解码器端有两种情况,下面我们分别加以讨论:

(1)在解完所有正向可解码的宏块后,没有发现错误.在这种情况下,解码过程恰好是上面编码器的逆过程.即:当解完块组中一半宏块后,将其比特位置记为 j ,然后向后寻找同步码(也就是寻找下一个块组头),找到后将这个同步码前面的那个比特位置记为 k .最后将从比特位置 $j+1$ 到 k 的所有比特逐个翻转(图 2),再对此翻转后的比特流解码,解出剩下的宏块.

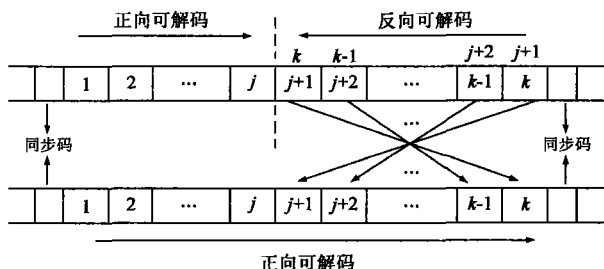


图 2 在解完正向可解宏块之后未发现错误时 PBDBS 方法的解码过程

(2)在解完所有正向可解码的宏块之前发现了错误(通过解码器的错误检测,可参考第 5 部分).此时解码器会立即停止解码,将当前解码器指针所指的比特位置记为 i ,同时将宏块在块组中的位置定为 $\frac{m}{2} + 1$ (假如块组包含有 m 个宏块),然后向后寻找同步码,同样要将此同步码之前的那个比特位置记为 k .最后将从比特位置 $i+1$ 到 k 的所有比特逐个反向放置(图 3),再将此翻转后的比特流解出另一半宏块.

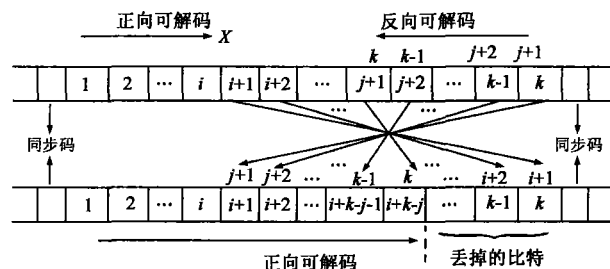


图 3 在解完正向可解宏块之前发现错误时 PBDBS 方法的解码过程

值得注意的是:PBDBS 方法由于采用了比特流反向放置的方法,因此可能会导致在从正向解码到反向解码过渡的比特流中出现错误的同步码,即可能会出现同步码误判的问题.但在实际的实验过程中这种情况极少发生,而且我们在检测到同步码后,都要再检验一下其后的 GN 和 GFID(图 6),看是否跟期望出现的值相同,如不相同,则寻找下一个同步码,从而大大降低了同步码误判的概率.因此,PBDBS 方法几乎不引入同步码误判的问题.

原始的 H.263 解码器的解码过程如图 4 所示.比较图 3 和图 4,我们可以看到,在解完块组中一半宏块前发现错误时,PBDBS 方法仍然可以解出后半宏块,而原始的解码器则是丢掉块组中错误宏块后的所有宏块.可见 PBDBS 方法可以大大降低由于丢失同步而导致的错误扩散.

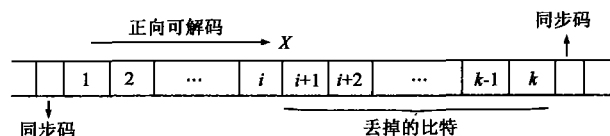


图 4 原始的 H.263 解码过程

4 PBDBS 方法的自适应切换点定位

4.1 原始的 PBDBS 切换点定位方法

在 PBDBS 方法中,一个很重要的问题是选择块组中正向可解宏块和反向可解宏块的个数,也就是选择由正向可解码到反向可解码的切换点的位置问题.我们假定每个宏块的出错概率相等^[1],并设为 p_e ,假定切换点位置为第 k 个宏块之后(图 5).由于块组中出错宏块之后的宏块都会丢失,因此在一个块组(包含 m 个宏块)中丢掉宏块数的期望值为:

$$E = \sum_{i=1}^k (k-i+1)p_e + \sum_{i=1}^{k-k} (m-k-i+1)p_e \\ = \left[\frac{(1+k)k}{2} + \frac{(m-k+1)(m-k)}{2} \right] p_e$$

$$= \left(k^2 - mk + \frac{m^2 + m}{2} \right) p_e \quad (1)$$

$$k = \frac{m}{2} \quad (2)$$

对上式求导并使 $E'_k = 0$, 则可得到最佳的切换点位置为: 即: 当切换点位于块组的中央时, 丢掉的宏块数最少。

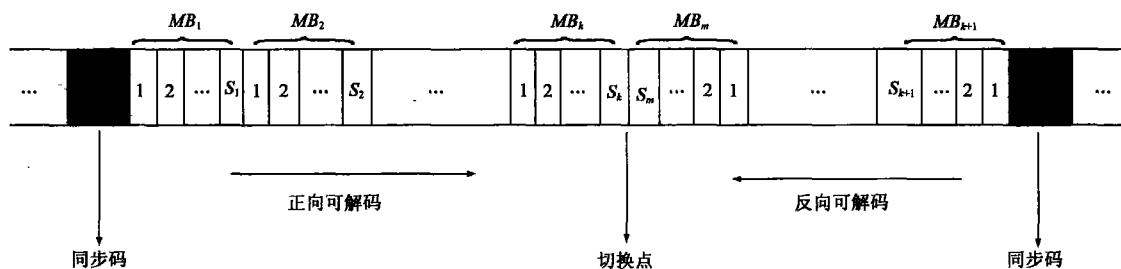


图 5 PBDDBS 中的块组示意图

4.2 自适应 PBDDBS 切换点定位方法——比特位置法

由于上面的推导是建立在假定每个宏块出错的概率相等的基础上, 而实际情况是每个宏块的出错概率并不相等, 因此把切换点放在块组中央并不是最佳的方案。如果能够确切地知道每个宏块的出错概率, 则可以得到更好的结果, 这促使我们提出了一种自适应切换点定位方法, 即比特位置法。这种方法就是针对如何得到宏块实际出错概率的。

假定信道错误为随机比特错误, 即每个比特出错的概率相等, 且相互独立, 则宏块所占的比特数越多, 其错误率也越大。如果我们能够得到每个宏块的比特数, 则可根据此比特数求出最佳的切换点位置。

4.2.1 比特位置法的具体过程 (1) 在编码的过程中, 将每个宏块的起始比特位置记住。编完一个块组后, 可得到每个宏块所占的比特数 S_i 及块组所占的比特数 G_{bit} ; (2) 用每个宏块的比特数 S_i 来表示宏块的错误率, 从第一个宏块开始对前 j

个宏块的比特数进行累加, 当累加值 $\sum_{i=1}^j S_i$ 大于块组比特数 G_{bit} 的一半时, 则比较 j 个宏块累加值与 $j-1$ 个宏块累加值 $\sum_{i=1}^{j-1} S_i$, 如果与块组比特数的一半接近, 则按照对应的宏块位置选择切换点。

值得注意的是, 由于每个块组的切换点不再相同, 因此有必要对块组头加以改造, 把切换点位置信息加入其中, 如图 6 所示。其中 SP(Switch Point) 为切换点位置信息, 共 5 个比特。

GSTUF	GBSC	GN	SP	GSBI	GFID	GQUANT
-------	------	----	----	------	------	--------

图 6 比特位置法的块组头

4.2.2 比特位置法的理论依据 参照图 5, 假定第 i 个宏块的比特数为 $S_i, i = 1, 2, \dots, m$; 随机比特错误率 (BER) 为 p , 并假定当比特流中的一个比特出错后会导其后的比特不可解。则可得到整个块组比特流丢掉比特的期望值为:

$$E = \sum_{j=1}^{S_1} j p + \sum_{j=S_1+1}^{S_1+S_2} j p + \dots + \sum_{j=S_1+\dots+S_{k-1}+1}^{S_1+\dots+S_k} j p + \sum_{j=1}^{S_{k+1}} j p + \dots + \sum_{j=S_{k+1}+\dots+S_{m-1}+1}^{S_{k+1}+\dots+S_m} j p = \left[\frac{(S_1 + \dots + S_k)(S_1 + \dots + S_k + 1)}{2} \right] p$$

$$+ \left[\frac{(S_{k+1} + \dots + S_m)(S_{k+1} + \dots + S_m + 1)}{2} \right] p$$

$$= \left[\frac{\theta_k^2 + \theta_k}{2} + \frac{(G_{bit} - \theta_k)^2 + (G_{bit} - \theta_k)}{2} \right] p$$

$$= \left[\theta_k^2 - G_{bit} \cdot \theta_k + \frac{G_{bit}^2 + G_{bit}}{2} \right] p$$

其中 $\theta_k = S_1 + S_2 + \dots + S_k, G_{bit} = S_1 + \dots + S_k + S_{k+1} + \dots + S_m$ 。

把式(3)对 θ_k 进行求导并使 $E'_{\theta_k} = 0$, 则可得到:

$$\theta_k = \frac{G_{bit}}{2} \quad (4)$$

这就是说, 当前面 k 个宏块比特数的累加值等于块组比特数的一半时, 块组比特流的错误率最低。事实上 $\theta_k = G_{bit}/2$ 的情况不大可能出现, 但可找到一个 k , 使得 θ_k 与 $G_{bit}/2$ 最接近, 从而得到最佳的切换点。

5 错误检测和掩盖

PBDDBS 方法取得比 H. 263 编解码器更好的容错性能, 但仍会出现宏块丢失, 导致图像质量的退化, 因此必须对丢失的宏块进行掩盖。错误掩盖必须知道丢失宏块的位置, 而这由错误检测来完成。

5.1 错误检测

可以分成两个步骤:

首先, 可由 H. 263 解码器检错方法确定是否出现了错误, H. 263 视频解码的检错方法有:

(1) 出现非法的 VLC 码;

(2) 编码结构和语法错误。例如: 解出的块数据中 DCT 系数的个数超过了 64;

(3) 对于帧间编码帧, 当解出的运动矢量超过规定范围时, 也可以判断出现了错误。

其次, 上面的方法仅提供是否出错的信息, 而不能提供错误的准确位置, 因为在解码器报告错误之前可能已经发生了错误, 只是未能发现。为了找到错误的准确位置, 就必须对错误宏块之前的各宏块进行检测, 以便找到最早出错的宏块^[10]。由于视频信号中存在相关性, 因此可利用图像中块与块之间的相关性来检错。这里我们采用了一种利用块间像素连贯性检测错误的方法。具体描述如下:

从错误宏块之前的宏块开始依次向前检测,对于每个宏块,我们对 4 个亮度块分别检测(检测顺序为 4,3,2,1).具体说,就是检测亮度块最上边的一行和最左边的一列像素,与其相邻的一行(上边)和一列(左边)像素比较,分别求出其平均绝对误差(MAD)值,任意一个 MAD 值超出了预定值,则判断此宏块出错,如图 7 所示.

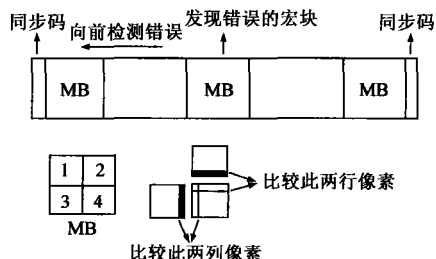


图 7 向前检测错误的具体过程

最上一行像素与其相邻像素的 MAD 为:

$$MAD = \frac{1}{8} \sum_x |I(x, y - y \bmod 8) - I(x, y - y \bmod 8 - 1)| \quad (5)$$

最左一列像素与其相邻像素的 MAD 为:

$$MAD = \frac{1}{8} \sum_y |I(x - x \bmod 8, y) - I(x - x \bmod 8 - 1, y)| \quad (6)$$

其中 x, y 为像素的横纵坐标, mod 为模运算.

当 $MAD \geq$ 预定值时,则判断宏块出错,然后检测前一个宏块;反之,则依次算出后面的 MAD. 只有当整个宏块各个块的 MAD 都小于预定值时,才可判断此宏块未出错. 此时停止错误检测,并确定最先出现错误的宏块为:未出错宏块之后的宏块. 这里我们取预定值为 40.

5.2 错误掩盖

我们利用视频信号时间和空间相关性进行掩盖,并可分成两种情况:

(1)第一帧图像. 没有参考帧,利用相邻像素值来掩盖.

$$\hat{I}(x, y) = \frac{1}{W_t + W_b + W_l + W_r} [W_t \cdot I(x, y - y \bmod 16 - 1) + W_b \cdot I(x, y - y \bmod 16 + 16) + W_l \cdot I(x - x \bmod 16 - 1, y) + W_r \cdot I(x - x \bmod 16 + 16, y)] \quad (7)$$

其中 x, y 为像素的横纵坐标, mod 为模运算, W_t, W_b, W_l, W_r 为权重系数,值为 0(对应宏块丢失或不存在)或 1(对应宏块已正确解码),下标 t, b, l 和 r 分别表示上下左右四个宏块.

(2)其他帧图像. 利用相邻宏块(帧间帧)或前一帧相同位置宏块(帧内帧)的运动矢量,找到前一帧对应像素区域进行掩盖.

对于帧间帧,运动矢量由相邻宏块的运动矢量的均值来估计.

$$\hat{MV}_{i,j} = \frac{1}{W_t + W_b + W_l + W_r} [W_t \cdot MV_{i,j-1} + W_b \cdot MV_{i,j+1} + W_l \cdot MV_{i-1,j} + W_r \cdot MV_{i+1,j}] \quad (8)$$

其中 i, j 为宏块的横纵坐标, W_t, W_b, W_l, W_r 的定义同上.

对于帧内帧,运动矢量取前一帧相同位置宏块的运动矢量,即:

$$\hat{MV}_{i,j}(t) = \hat{MV}_{i,j}(t - \tau) \quad (9)$$

其中 τ 为帧延时.

6 实验结果及分析

我们在开放的 H. 263 测试模型 TMN-5 中实现了比特位置法,并与 H. 263 标准编码器及原始的 PBDBS 方法做了比较. 在我们的测试中,我们使用了两个标准的四分之一公共中间格式(QCIF)测试序列,分别为“news”和“container ship”(帧率为 25 帧/秒,共 300 帧). 编码模式为:第一帧使用帧内编码模式,后面的图像帧每隔 30 帧使用帧内编码模式刷新一次,其余图像帧使用帧间编码模式. 量化参数 QUANT^[2]固定为 6. 我们使用峰值信噪比(PSNR)来衡量图像的客观质量,PSNR 被定义为:



图 8 三种方法在出现错误(相同位置)后第一帧“container ship”的重建图像,从左到右依次是 H. 263, PBDBS 和比特位置法上面一行为掩盖前的图像,下面一行是掩盖后的图像.

表 1 三种方法在错误掩盖后不同错误率下的 PSNR(dB)比较

错 误 率	Container ship			News		
	H.263	PBDBS	比特位置	H.263	PBDBS	比特位置
0.0001	29.48	30.10	30.36	27.99	28.95	29.06
0.0005	20.37	21.39	21.51	19.59	20.56	20.71
0.001	16.95	17.66	17.96	15.63	16.98	17.31

$$\text{PSNR} = 10 \log_{10} \frac{(255)^2}{\sum_x \sum_y (X_1(x, y) - X_2(x, y))^2} \quad (10)$$

其中 X_1, X_2 分别表示原始和解码后的图像。

图 8 为三种方法主观质量的比较。从图中可以看出,在出现错误后,H.263 会导致块组中错误宏块之后所有宏块都丢失,而 PBDBS 方法则仍然可以解出后面一半的宏块。从比较得出:比特位置法比 PBDBS 方法少丢了一个宏块,进一步降低了错误扩散的程度。从掩盖后的结果可以看出,PBDBS 方法和比特位置法都可以取得比原始的 H.263 方法更为满意的图像质量。

表 1 为三种方法客观质量的比较。为了消除错误造成影响的随机性,我们对每个视频序列都重复测试了 100 次,并算出 PSNR 的平均值。从表中可以看出,PBDBS 方法比基本的 H.263 方法最多可以高出 1 个多 dB,而比特位置法又可比原始的 PBDBS 方法高零点几个 dB。

无论从主观还是客观质量上看,PBDBS 方法比基本的 H.263 方法具有更好的容错性能,比特位置法由于能够自适应地改变切换点位置,从而增强了 PBDBS 方法的容错性能。

7 结论

本文提出了一种 PBDBS 方法的自适应切换点定位方法——比特位置法。由于比特位置法考虑了宏块的实际错误概率,能够自适应地改变切换点的位置,因而可使视频比特流的整体容错性能进一步提高。实验证明,比特位置法在原始 PBDBS 方法的基础上,进一步减少了错误扩散的程度,解码后的图像质量比原始的 PBDBS 方法有所提高。这种方法可以被应用到 PSTN 或无线信道上的视频传输。同时还可以与其他的容错编码方法相结合以得到更好的结果,如数据分割(Data partitioning)和分层编码等。这也是我们下一步将要要做的工作。

参考文献:

- [1] Shaoshuai Gao, Guofang Tu. Robust H.263 + video transmission using partial backward decodable bit stream (PBDBS) [J]. IEEE Trans. Cir-

cuits Syst. Video Technol., Feb. 2003, in print.

- [2] ITU Telecommunications Standardization Sector, ITU-T Recommendation H.263, Video coding for low bit rate communication [S]. 1995.
- [3] ITU Telecom. Standardization Sector of ITU, Video coding for low bit rate communication, Draft ITU-T Recommendation H.263 Version 2 [S]. Sept. 1997.
- [4] ITU Telecom. Standardization Sector of ITU, Draft for H.263 + Annexes U, V, and W to Recommendation H.263 [S]. 2000.
- [5] ISO/IEC FDIS 14496-2, Generic Coding of audio-visual objects: (MPEG-4 video), Final Draft International Standard [S]. Document N2502, 1999.
- [6] J Wen, J Villasenor. Reversible variable length codes for robust image and video transmission [A]. In Proc. IEEE 31st Asilomar Conf. Signals, Systems & Computers [C]. 1998, 2: 973 - 979.
- [7] J Wen, J Villasenor. A class of reversible variable length codes for robust image and video coding [A]. In IEEE Proc Int. Conf. Image Processing [C]. 1997. 65 - 68.
- [8] J Wen, J Villasenor. Reversible variable length codes for efficient and robust image and video coding [A]. In Proc IEEE Data Compression Conf. [C]. 1998. 471 - 480.
- [9] Jennifer L H. Webb. Efficient table access for reversible variable-length decoding, IEEE Trans. Circuits Syst. Video Technol. [J]. Aug. 2001, 11 (8): 981 - 985.
- [10] 高绍帅, 涂国防. 基于 H.263 视频解码的错误检测和掩盖[J]. 通信学报, Dec. 2001, 22(12): 82 - 87.

作者简介:

高绍帅 男, 1976 年生于山东德州, 中国科学院研究生院信息科学与工程学院在读博士生, 主要研究方向为: 容错视频编码与通信, 联合信源信道编码等. Email: shaoshuai_gao@mails.gscas.ac.cn.

涂国防 男, 1954 年生于湖南长沙, 中国科学院研究生院信息科学与工程学院常务副院长、教授、博士生导师, 主要从事数据压缩、图像处理、无线通信等方面的研究与教学工作。