

基于 ZIG-ZAG 交织的 H. 264/ AVC 容错编码算法

姜 东, 李 波, 李 炜, 宋建斌

(北京航空航天大学计算机学院数字媒体实验室, 北京 100083)

摘 要: 本文提出了一种基于 ZIG-ZAG 交织的灵活宏块排列算法 ZFMO, 按 ZIG-ZAG 扫描方式沿次对角线方向上交织排列宏块, 从而在低位率下达到了编码效率和容错性能的最佳平衡点. 实验结果表明, ZFMO 在低丢包率的网络环境下比当前 H. 264/ AVC 的推荐算法具有更好的编码效率和容错性能. 经过率失真优化后的 ZFMO 算法性能进一步提高.

关键词: 视频容错编码; FMO; H. 264/ AVC; 率失真优化

中图分类号: TP339 **文献标识码:** A **文章编号:** 0372-2112 (2006) 11-1941-06

An Error Resilient H. 264/ AVC Video Coding Algorithm Using ZIG-ZAG Interleaved FMO

JIANG Dong, LI Bo, LI Wei, SONG Jian bin

(Digital Media Lab of School of Computer Science and Technology, Beihang University, Beijing 100083, China)

Abstract: In accordance with the increasing requirements for video transmission over unreliable channels, the issue of error resilient video coding technology has been paid enough attention. Based on ZIG-ZAG interleaved flexible macroblock ordering, a novel algorithm called ZFMO is proposed in this paper. Different with the traditional raster scan order, ZFMO achieves a good trade-off between coding efficiency and error resilience by rearranging macroblocks in ZIG-ZAG scanning order and partitioning the macroblocks into different slice groups in an interleaved way. Experimental results show that ZFMO outperforms H. 264/ AVC recommended algorithm in low packet loss environments. The using of ZFMO influences intra frame prediction and motion estimation because of the macroblocks rearrangements, so the coding mode selection must be reconsidered. The rate distortion optimization based ZFMO decreases the redundancy bitrate while maintain recovery video quality and error resilience performance. Simulation results based on H. 264/ AVC reference software JM9 suggest that the performance of rate distortion optimized ZFMO can be further improved.

Key words: error resilient video coding; flexible macroblock ordering; rate distortion optimization; H. 264/ AVC

1 引言

随着网络技术和多媒体技术应用的广泛和深入, 在不可靠信道上传输视频的需求日益增长, 视频编码和传输中的错误控制问题已经引起了广泛关注.

当前基于 DCT 的预测编码和基于 VLC 的熵编码大大地提高了信源编码的效率, 但同时也使得视频流对于传输错误十分敏感. 在不可靠信道如 Internet 和无线网络的情况下, 信道出现突发噪声、随机噪声的可能性大大增加, 导致视频传输中数据包的丢失^[1, 2]. 如果不加控制措施, 数据包的丢失引起错误扩散, 导致视频质量的严重下降. 因此, 研究容错视频编码和传输技术, 对于解决由于信道误码引起接收端视频质量下降以及由此带来的误码扩散问题具有十分重要的意义.

H. 264/ AVC 标准把视频流的容错性能与视频编码效率摆在了同等重要的地位加以考虑^[3, 4]. 灵活的宏块排列 (Flexible Macroblock Ordering) 以及基于率失真优化 (Rate Distortion optimization) 的模式选择被证明是有效的容错工具^[5-7]. 本文提出了一种基于 ZIG-ZAG 交织的灵活宏块排列算法 ZFMO, 依 ZIG-

ZAG 扫描方式在次对角线方向上交织排列宏块, 从而在低丢包率的网络环境下达到编码效率和容错性能的最佳平衡点.

本文的组织结构如下, 第 2 节对当前 H. 264/ AVC 标准中推荐的 FMO 算法性能进行评估, 第 3 节介绍本文算法 ZFMO, 第 4 节针对 ZFMO 进行率失真优化, 第 5 节为实验结果, 最后对本文进行了总结.

2 H. 264/ AVC FMO 性能评估

2.1 FMO 算法思想

传统的视频编码标准中一幅图一般都是被分成大小固定的宏块, 并且在这些标准中解码过程的定义都是以宏块为基本单位, 而 Slices 则是按扫描顺序排列的宏块序列. 在这种情况下, 如果不使用灵活的宏块排列 (Flexible Macroblock Ordering, FMO), 一幅图一般被切分成如图 1 所示的一个或几个切片组 (Slices).

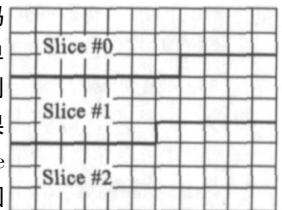


图 1 传统编码标准中一幅图的划分

FMO 是一种允许编码器以传统扫描顺序以外的其它顺序将宏块划分到切片组的灵活的宏块组织方式^[8]. 在 H. 264/AVC 中, 切片组是自包含的, 即每个切片组都包含完整的参数集并存储在压缩码流中, 在解码端每个可单独解码. 通过对切片概念的重新定义, FMO 改变了以往对于图像的划分方法. 每个切片就是一组宏块的集合, 它通过参数集定义了每个宏块到切片的映射表, 该映射表记录了每个宏块在图中的位置以及该宏块属于哪个切片组.

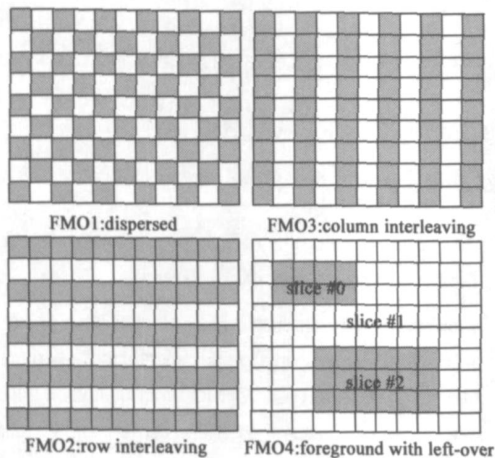


图 2 几种宏块扫描模式

在 FMO 方式下, 一幅图中宏块的扫描模式可以有多种. 当前, H. 264/AVC 的验证模型 JM9.0 中有 7 种模式, 图 2 所示. 分别为 interleaved, dispersed, foreground with left over, box-out, raster scan, wipe 以及 explicit 自定义模式. 本文选取了容错性能较好的 interleaved, dispersed, foreground with left over 几种, 并添加了 column interleaved 模式, 然后对这几种排列方式进行了对比测试.

(1) FMO1: 分散 (dispersed) 模式, 即空间相邻宏块被划分到不同的切片, 也就是类似棋盘状的排列方式. 整幅图被分成两个切片组, 这样传输时即便有一个切片丢失, 通过插值的方式进行差错掩盖仍然可以得到可以接受的解码图像. 在丢包率较高的网络下, dispersed 模式具有非常明显的优势.

(2) FMO2: 行交织 (row interleaved) 模式, 偶数行的宏块被分到一个切片, 奇数行宏块被分到另一个切片. 该模式对于竖直纹理较多的图像具有很好的容错效果.

(3) FMO3: 列交织 (column interleaved) 模式, 偶数列的宏块被分到一个切片, 奇数列的宏块被分到另一个切片. 该模式对于水平纹理较多的图像具有很好的容错效果.

(4) FMO4: 前景-背景 (foreground with left over) 模式, 它可以分为一个或多个“前景”切片组, 其余部分作为一个切片, 这种排列模式可以作感兴趣区域的编码应用.

2.2 FMO 性能测试

图 3 表示了测试序列 coastguard.qcif 在丢包率为 5% 的网络下使用 FMO1 与没有使用 FMO 两种情况下的解码端恢复图像质量对比, 结果表明使用 FMO 具有非常明显的优势. 图 4 表示 coastguard.qcif 序列在 3%, 5%, 10% 和 20% 共 4 种不同丢包率下 3 种 FMO 模式 (FMO1, FMO2, FMO3) 容错性能测试结

果, 表明 FMO1 的性能总体优于其他模式, 在高丢包率网络下 FMO1 的性能优势更加突出.

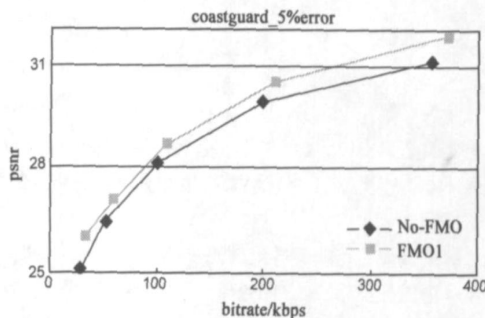


图 3 FMO1 与 NO-FMO 对比结果

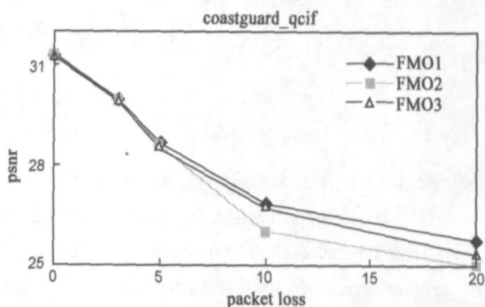


图 4 FMO1、FMO2 与 FMO3 对比结果

3 基于 ZIG-ZAG 交织的算法 ZFMO

3.1 ZFMO 算法思想

不同的宏块组织方式可导致不同的编码效率和容错能力. 现有的帧内预测机制规定帧内预测不应超出切片组范围, 如果将一些直接相邻的宏块分配到同一切片组内, 则可以利用空间相关性进行帧内预测, 此时将获得较高的编码效率; 如果空间相邻的宏块被分配到不同的切片组, 原有的空间相关性被打破, 帧内预测的有效性大大降低, 因此编码效率将随之降低. 但此时由于相邻宏块被分配在不同的切片组, 解码端不会出现连续丢失的区域, 基于解码端的差错掩盖可取得非常显著的效果.

本文 1.2 节的测试结果表明, 在 H. 264/AVC 标准中推荐使用的多种模式中, FMO1 (dispersed) 的宏块排列方式具有优势. 其棋盘状的排列方式分散了传输错误对一幅图带来的影响, 也有利于解码端的差错掩盖. 在丢包率较高的网络环境下其性能更好. 但该模式存在的问题是完全打乱了宏块原有的排列方式, 使得帧内预测的有效性大大降低, 从而降低了编码效率. 在低丢包率的环境下 ($\leq 5\%$), H. 264/AVC dispersed 模式并不能达到编码效率和容错性能的最佳平衡点.

为了达到编码效率和容错性能的最佳平衡点, 本文提出了一种基于 ZIG-ZAG 交织的排列方式 ZFMO. ZIG-ZAG 扫描是一种按次对角线方向依次扫描的方式, 在次对角线方向上进行交织排列宏块, 如图 5 所示, 非边缘区域的每个宏块, 与其直接相邻的 4 个宏块 (图 6 所示) 经过重排后, 被分配到同一切片组内的宏块数和分配到另一切片组的宏块数各占一半, 这样就同时兼顾了编码效率和容错性能.

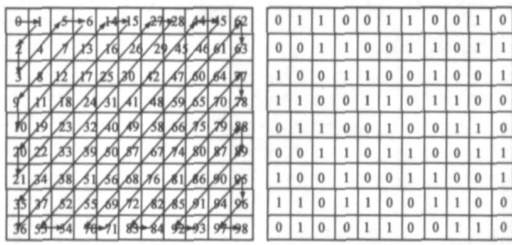


图 5 基于 Zig-zag 扫描的交织排列

图 5 基于 Zig-zag 扫描的交织排列

3.2 ZFMO 编码效率与容错性能分析

使用 FMO 在一定程度上降低了同一个切片组内相邻宏块的相关性,由于帧内预测限制在切片组内部,FMO 的引入打乱了宏块的顺序从而使帧内预测的有效性有所降低.因此 I 帧的编码效率会有一定程度的降低.另外,P 帧也会因为 I 帧而受到间接影响,但由于运动估计时其参考帧已经按正常的排列方式恢复重建,因此受到的影响很小.

设 F_n 表示视频序列中的第 n 帧图像数据, $F_{n,i}, F_{n,j}$ 分别表示第 n 帧内的第 i, j 个宏块.设一帧数据共分 k 个切片组,切片组集合用 U 表示,图像分辨率为 $W \times H$,则 ZFMO 的映射

$$F_{n,i} \in \begin{cases} U_0, & \text{if } F_{n,i} \% k = 0 \\ U_1, & \text{if } F_{n,i} \% k = 1 \\ \dots\dots \\ U_k, & \text{if } F_{n,i} \% k = k \end{cases}$$

此时 $F_{n,i}$ 为 ZIG-ZAG 扫描顺序.关于 k 的取值,即切片组的个数,从容错的角度考虑,每帧数据分的包数越多,意味着从整帧的角度看丢失的概率越低.但从网络的角度考虑,分包越多则使网络包头信息占的比重越大,从而相当于增加了码流量,更容易引起网络拥塞;另外,切片组过多,会使编码效率严重降低.结合参考文献[9]对不同的切片组的实验结果,本文每帧数据分为 2 个切片组.

设 $C(F_{n,i}, F_{n,j})$ 表示两宏块 $F_{n,i}$ 和 $F_{n,j}$ 的相似度,用正常扫描时的两个宏块的相距距离来度量,如图 7 所示.它表示宏块 $F_{n,i}$ 可以被 $F_{n,j}$ 进行帧内预测的有效性.

$$C(F_{n,i}, F_{n,j}) = \begin{cases} c_1, & |i-j| \% (W/16) = 1 \\ c_2, & |i-j| \% (W/16) = 2 \end{cases}$$

$R(F_{n,i}, F_{n,j})$ 表示两宏块的可替代度,用距离来度量,它表示在宏块 $F_{n,j}$ 替代丢失的 $F_{n,i}$ 的有效性.

$$R(F_{n,i}, F_{n,j}) = \begin{cases} r_1, & |i-j| \% (W/16) = 1 \\ r_2, & |i-j| \% (W/16) = 2 \end{cases}$$

自然图像是平滑的,如果没有剧烈运动或者不是位于物体的边缘.空间上相邻像素或时间上前后相邻帧对应位置像素色度值的变化是平滑的.该结论是视频编码中进行帧内预测和帧间运动估计的基础,同时也是解码端进行差错掩盖的基础.以图 7 中中间宏块为例,若对其进行帧内预测,显然利用直接与其相邻的上、左宏块(斜线所示)更为有效.但若在传输中该宏块丢失,在解码端对其进行差错掩盖时,则不希望出

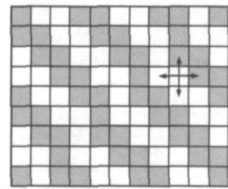


图 6 ZFMO 宏块排列方式

现连续丢失的区域,丢失宏块越分散,越有利于差错掩盖.

由 FMO1 和 ZFMO 的映射关系可以看出,FMO1 经过宏块重排后,完全打乱原有宏块的排列顺序,使其呈棋盘状,导致原先相邻的宏块被分散到不同的切片组中,降低了帧内预测的效率,同时也降低了后续 P 帧的编码效率.而 ZFMO 则同

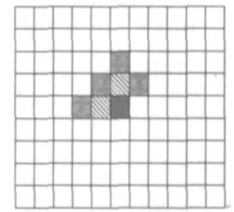


图 7 ZFMO 对编码效率与容错性能的影响

时照顾到了编码效率和容错性能.在低丢包率的网络环境下,编码端的编码效率对解码端的恢复图像质量影响更大,因此在低丢包率情况下 ZFMO 可取得比 FMO1 更好的效果;而在高丢包率的网络环境下,编码效率的提高对解码端重建图像质量影响权重有所降低.基于解码端的差错掩盖此时对重建图像质量的提高具有决定性的作用,因此,高丢包率的网络下考虑的重点应该是如何方便解码端的差错掩盖.此时,FMO1 具有理想的容错性能.

3.3 ZFMO 的实现复杂度分析

ZFMO 算法仅在编码前重新排列宏块,对编码计算复杂度的影响非常小,在解码端,由于解码前需恢复宏块的正常顺序,会引入少量延迟,该延迟在绝大多数应用的许可范围之内.ZFMO 算法导致帧内预测的编码效率有所降低,而帧间运动估计和运动补偿则基本不受影响.

4 针对 ZFMO 的率失真优化

通常一个宏块的编码模式有 Intra/ Inter 两类,Intra 宏块压缩比相对较低,即编码效率低,但遇到网络错误时其解码不需参考帧,网络容错性能好;Inter 宏块通过运动估计可大大提高编码效率,但要正确解码时必须依靠其参考帧,因此对于有丢失的网络其信道失真较大.

在没有建立反馈信道的情况下,在视频流中插入 Intra 编码的宏块是阻止传输错误扩散的有效方法^[9,10].因此,在编码阶段合理地选择编码模式不仅可以获得理想的编码效率,同时还能提高压缩码流的容错性能.ZFMO 影响了原有的帧内预测与帧间预测机制,也同时影响原有的宏块编码模式的选择.因此,针对 ZFMO 进行率失真优化,综合考虑 Intra/ Inter 两种编码模式以及不同 Inter 模式的选择对于编码效率及容错性能的影响,能获得总体更优的效果.

传统的编码模式选择通过计算整像素运动估计的 SAD 值与一个预先设定的阈值进行比较,以决定该宏块的编码模式为 INTRA, INTER, 还是 SKIP 模式.这种决策方式仅仅考虑到在视频内容变化时通过改变编码模式维持一定的编码质量,没有综合考虑位率与失真的相互约束关系,更没有照顾到码流的容错性能.

本文运用文献[12]的率失真优化策略,首先进行整像素运动估计,然后通过求得最小的 Lagrange 代价函数实现基于位率约束的优化.通过该方法得到的增益虽然不多,但是计算简单,容易实现,与 ZFMO 算法结合使用,可显著提高视频流的容错效果.

5 实验结果

5.1 测试条件

国际电信联盟为基于 IP 的网络定义了仿真工具, 具体可以参考 ITU VCEG-N79 文档“Common Conditions for wire line, low delay IP/UDP/RTP packet loss resilient testing”. 其中包含 3%, 5%, 10%, 20% 共 4 种模式的错误模式文件, 用以模拟不同的信道。

关于网络打包方案, 从容错的角度考虑, 每帧数据分的包数越多, 意味着从整帧的角度看丢失的概率越低。但从网络的角度考虑, 分包越多则使网络包头信息占的比重越大, 从而相当于增加了码流量, 更容易引起网络拥塞。因此, 本文实验每帧数据分为 2 个切片组。

解码端差错掩盖采用 H. 264/AVC 的推荐算法。其他测试条件如下。测试平台: JM9.0, 测试序列: foreman_qcif, coastguard_qcif 和 carphone_qcif. 300 frames, 25frame/s, I ... 14 P, QP = 24, 28, 32, 36, 40。

5.2 编码效率比较

图 8 所示为测试序列 foreman, coastguard 和 carphone 的编码效率比较。左图为三个测试序列的率失真曲线, 对本文算法 ZFMO, 经过率失真优化的 ZFMO (ZFMO+RDO) 和 H. 264/AVC 的推荐算法进行了对比。由结果可以看出, ZFMO 的率失真性能明显优于 H. 264/AVC JM9.0, 而经过率失真优化的 ZFMO 性能又好于 ZFMO。图 8 右图表示了 QP 值为 36 时, 两个测试序列前 62 帧的长度, 可以看出, 对于 I 帧, ZFMO 的压缩比明显高于 H. 264/AVC 推荐算法, 因为 FMO 对于 P 帧的影响很小, 因此两种算法对于 P 帧的影响不大。

性能明显优于 H. 264/AVC JM9.0, 而经过率失真优化的 ZFMO 性能又好于 ZFMO。

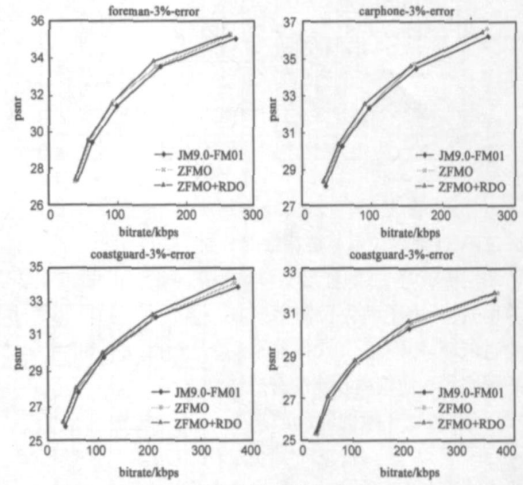


图 9 ZFMO 算法的容错性能

5.4 冗余度比较

容错编码的目标是在网络出现错误时保证接收端基本的解码质量, 这不可避免的通过添加冗余数据来实现。因此, 一个容错算法的冗余度大小也是衡量该算法的指标。冗余度 (Redundancy rate) 是容错编码器的另一个指标, 指使用容错手段后额外增加的位率, 它体现了使用该容错措施的代价。

$$\text{冗余度: Rate} = \frac{R^* - R}{R} \times 100\%$$

其中, R 为没有容错措施时的编码位率, R^* 为添加容错后的编码位率。本文比较了两种算法与没有使用 FMO 时的冗余度, 结果如表 1 所示。

表 1 ZFMO 与 JM9.0 FMO1 的冗余度比较

bitrate(kbps)	JM9.0 FMO1		ZFMO		NO-FMO	
	bitrate	redundancy	bitrate	redundancy		
foreman	QP= 28	97.17	6.87%	92.15	1.35%	90.92
	QP= 32	62.44	11.46%	58.33	4.12%	56.02
coastguard	QP= 28	109.67	8.48%	105.69	4.54%	101.1
	QP= 32	59.65	12.87%	56.23	6.40%	52.85
carphone	QP= 28	95.45	9.20%	90.42	3.44%	87.41
	QP= 32	57.45	10.35%	53.86	3.46%	52.06

5.5 主观质量比较

图 10 选取了测试序列 Foreman 中的几帧容错效果, 丢包率为 3%, 分别为第 119 帧和第 235 帧, 左图为 JM9.0 FMO1 算法, 右图为 ZFMO 算法。可以看出, 与 JM9.0 FMO1 算法相比, ZFMO 算法对于 Foreman 面部区域的恢复效果有明显改善。图 11 选取了测试序列 Coastguard 中的几帧容错效果, 分别为第 5 帧、第 44 帧, 左图为 JM9.0 FMO1 算法, 右图为 ZFMO 算法。可以看出, 右图中 Coastguard 的船部纹理细节更加清晰, 有了明显的改善。

6 总结

FMO 和基于率失真优化的模式选择是容错编码的有效工具。FMO 以极小的计算量和较少的冗余在丢包网络下可获

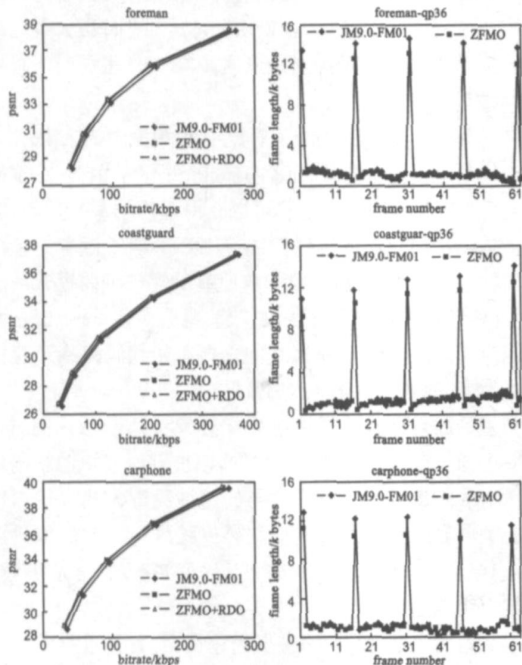


图 8 ZFMO 算法的编码效率

5.3 容错性能比较

图 9 所示为三个测试序列的容错性能比较, 在低丢包率下(3%, 5%)的率失真曲线。由结果可以看出, ZFMO 的率失真



图 10 Foreman 序列的主观质量比较

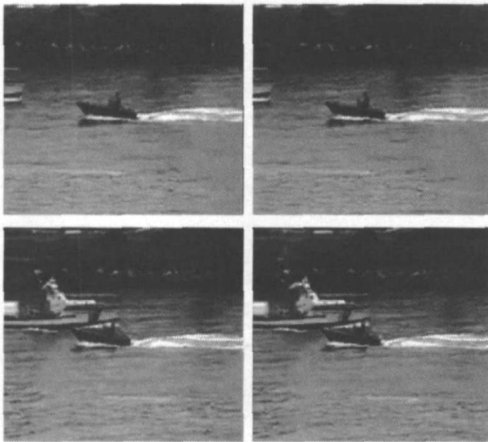


图 11 Coastguard 序列的主观质量比较

得非常显著的增益。基于率失真优化的模式选择在一个给定的位率约束下进行合理的选择编码策略,从而同时提高了编码效率和容错性能。本文提出的基于率失真优化的 ZFMO,在中低位率,低丢包率的网络下具有明显的优势,ZFMO 可以在高丢包率网络下具有优势的 H. 264/ AVC FMO1 互为补充,在不同的网络状况下自适应选择,从而提高视频传输对不同信道的适应能力。

参考文献:

- [1] Guy Cote, Shahram Shirani, Faouzi Kossentini. Optimal mode selection and synchronization for robust video communications over error prone networks[J]. IEEE Journal on Selected Areas in Communications, 2000, 18(6): 952- 965.
- [2] T Stockhammer, D Kontopodis, T Wiegand. Rate distortion optimization for JVT/ H. 26L video coding in packet loss environment[A]. Proceeding of Packet Video Workshop [C]. Pittsburgh, PY, 2002.
- [3] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, Ajay Luthra. 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] M Karczewicz, R Kurceren. A proposal for SP-Frames [A]. ITU-T Video Coding Experts Group Meeting [C]. Eibsee, Germany, 2001.
- [5] Y-K Wang, MM Hannuksela. Gabbouj. Error robust inter/ intra mode selection using isolated regions [A]. Proceeding of International Packet Video Workshop [C]. Nantes, France, 2003.
- [6] T Wiegand, H Schwarz, A Joch, F Kossentini. Rate constrained coder control and comparison of video coding standards [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7) : 688- 703.
- [7] R Zhang, SL Regunathan, K Rose. Video coding with optimal inter/ intramode switching for packet loss resilience [J]. IEEE Journal on Selected Areas in Communications, 2000, 18(6) : 966 - 976.
- [8] Jom Ostermann, Jan Bormans, Peter List, Detlev Marpe. Matthias Narroschke, Fernando Pereira, Thomas Stockhammer, Thomas Wedi. Video coding with H. 264/ AVC: Tools, performance, and complexity [J]. IEEE Circuits and Systems Magazine, 2004, 4(1) : 7- 28.
- [9] Thomas Wiegand, B Girod. Lagrangian multiplier selection in hybrid video coder control [A]. Proceeding of International Conference on Image Processing [C]. Thessaloniki, Greece, 2001. 542- 545.
- [10] Thomas Stockhammer, Stephan Wenger. Standard compliant enhancement of JVT coded video for transmission over fixed and wireless IP [A]. Proceeding of the International Workshop of Advanced Methods for Multimedia Signal Processing [C]. Capri, Italy: 2002. 645- 656.
- [11] Carlos Miguel Tavares Calafate, Manuel Perez Malumbres. Evaluation of the H. 264 codec internal report [R]. DISCA UPV (Universidad Politcnica de Valencia), 2003.
- [12] GJ Sullivan, T Wiegand. Rate distortion optimization for video compression [J]. IEEE Signal Processing Magazine, 1998, 15 (6) : 74- 90.

作者简介:



姜 东 男, 1976 年生于山东, 北京航空航天大学计算机学院在读博士, 研究方向为视频通信、视频容错编码与传输。
E-mail: duff99@163.com

李 波 男, 1966 年出生于四川, 北京航空航天大学计算机学院教授、博士生导师, 研究方向为数字视频处理、多媒体、智能信息处理与嵌入式数字图像处理。

李 炜 男, 1973 年出生于陕西, 北京航空航天大学计算机学院博士, 研究方向为视频编码、多媒体系统与视频编码的 VLSI 设计。