

一种基于外推运动矢量的差错掩盖方法*

阮若林^{1,2}, 胡瑞敏², 曾书婧², 尹黎明¹

(1. 咸宁学院 电子与信息工程学院, 湖北 咸宁 437100; 2. 武汉大学 国家多媒体软件工程技术研究中心, 武汉 430079)

摘要: 为了克服边框匹配算法因边框匹配不准确而导致最优运动矢量的漏选, 以及差错掩盖效果明显下降的问题, 提出了一种新的基于外推运动矢量的自适应时域差错掩盖算法。首先, 根据宏块运动矢量的时空相关性, 从相邻宏块的运动矢量及其外推运动矢量中自适应地构造出一个有限的候选运动矢量集; 然后, 利用边框匹配算法从构造的候选运动矢量集中选取最佳运动矢量进行差错掩盖; 最后, 在 H. 264/AVC 的参考模型 JM12.2 中编程实现该算法。大量仿真实验结果表明, 本算法可以明显降低最优运动矢量的漏选, 在解码端取得了较为满意的差错掩盖效果。

关键词: 视频编码; 运动矢量; 外推矢量; 差错掩盖

中图分类号: TP391 **文献标志码:** A **文章编号:** 1001-3695(2012)04-1547-03

doi:10.3969/j.issn.1001-3695.2012.04.096

Error concealment method based on extrapolation motion vectors

RUAN Ruo-lin^{1,2}, HU Rui-min², ZENG Shu-jing², YIN Li-ming²

(1. School of Electronic & Information Engineering, Xianning University, Xianning Hubei 437100, China; 2. National Engineering Research Center for Multimedia Software, Wuhan University, Wuhan 430079, China)

Abstract: In order to overcome the problem of the inaccuracy of the boundary matching algorithm to lead to leakage selection of the optimal motion vector, and the effect declining of the error concealment, this paper proposed a new temporal error concealment algorithm based on extrapolation motion vectors set. Firstly, according to the temporal-spatial correlation of the motion vectors, this paper adaptively constructed a limited candidate motion vector set from neighboring macroblocks motion vectors and their extrapolation motion vectors. Secondly, it used the boundary matching algorithm to select the optimal motion vector to conceal the error from the candidate motion vectors set. Lastly, the paper realized the algorithm based on H. 264/AVC JM12.2. The simulation results show that the proposed algorithm can effectively reduce the leakage selection of the optimal motion vectors, at the same time, achieve the satisfactory error concealment effect.

Key words: video coding; motion vector; extrapolation vector; error concealment

0 引言

随着压缩视频编码技术的进一步发展, 视频编码的压缩效率越来越高, 使得高质量的视频码流可以在不同的信道中进行传输。但是, 由于传统视频压缩标准都是基于变换/预测的混合编码框架, 压缩过程中采用了预测编码和熵编码技术, 基本上很少考虑压缩数据在传输过程中有可能出现差错或丢失的问题, 使得压缩码流对传输差错特别敏感, 抵抗差错的能力非常脆弱。所以, 压缩码流在复杂多变的各种信道中进行传输时, 经常会出现数据出错或丢失的情况, 同时, 出错数据还会在时域和空域上迅速蔓延和扩散, 使得解码视频的主观质量急剧下降^[1-3]。图 1 给出了 foreman 序列在误码率为 10^{-4} 时, 没有

采用任何差错控制和掩盖技术时的解码图像质量情况。



图1 差错在时空域上蔓延示意图

差错掩盖技术是在解码端利用已知的未出错宏块信息掩盖恢复出错宏块的一种技术手段, 可以较好地恢复传输过程中差错或丢失的视频数据, 提高解码视频的主客观质量。通常来说, 差错掩盖技术可分为空域差错掩盖和时域差错掩盖两大类。空域差错掩盖技术主要是利用视频丢失信息块在空域上的周边已知信息, 对丢失信息块进行掩盖恢复; 时域差错掩盖

收稿日期: 2011-08-11; **修回日期:** 2011-09-29 **基金项目:** 国家自然科学基金资助项目(61070080); 国家重点基础研究发展计划资助项目(2009CB320906); 国家科技重大专项项目课题(2010ZX03004-003); 湖北省自然科学基金资助项目(2010CDB05103); 湖北省教育厅优秀中青年人才资助项目(Q20112805); 咸宁学院博士启动基金资助项目(BK1105)

作者简介: 阮若林(1974-), 男, 湖北红安人, 副教授, 博士, 主要研究方向为视频编解码及多媒体网络通信(rluan@foxmail.com); 胡瑞敏(1964-), 男, 湖北武汉人, 教授, 博士, 主要研究方向为多媒体技术及安防应急技术; 曾书婧(1984-), 女, 湖北通山人, 硕士, 主要研究方向为视频编解码; 尹黎明(1974-), 男, 湖北嘉鱼人, 讲师, 博士研究生, 主要研究方向为视频编解码。

技术则主要是利用视频丢失信息块在时域上(主要是已解码的前一帧或前几帧)的相应块及周边已知信息,对丢失信息块进行掩盖恢复。由于当前先进视频编码标准都是基于块的编码技术框架,编码过程大量使用了时域运动估计技术,因此,时域差错掩盖通常能够获得更好的掩盖效果,也是差错掩盖技术研究的热点。

时域差错掩盖是基于运动补偿的时间预测,它通过恢复受损块的运动信息来进一步恢复出整个块的像素信息。现有的时域差错掩盖算法的基本思路是利用边框匹配算法(boundary matching algorithm, BMA)从候选运动矢量集中选择最优运动矢量进行差错掩盖,候选运动矢量集越大,掩盖效果越好。在目前已有的方法中,Wang 等人^[1]提出的差错掩盖方法是只利用受损块相邻块运动矢量,不考虑外推运动矢量,然后利用边框匹配算法从相邻运动矢量中选择最佳运动矢量,但是在物体边界处,由于系数的剧烈变化,采用边缘平滑性原则来进行掩盖常常会导致误差。而彭强等人^[2]则将外推运动矢量和受损块相邻运动矢量同时纳入候选预测运动矢量集,然后结合边框匹配算法从中选取最佳运动矢量,此种方法显然与文献[1]中的方法存在同样的问题,同时由于候选矢量集过大,增加了计算复杂度。在文献[4,5]中各自提出了相应的自适应差错掩盖同样也存在类似问题。因此,为了尽量减小边框匹配带来误差和计算的复杂度,提高差错掩盖的效果,本文提出了一种基于外推运动矢量的自适应时域差错掩盖方法。

1 基于自适应候选运动矢量集的差错掩盖算法

1.1 候选运动矢量集的构造

目前,在实际编码标准中广泛采用的时域掩盖算法主要有边框匹配、解码端运动估计以及自适应掩盖算法等,都是基于整个丢失宏块来恢复运动信息,效率不高,不能取得很好的效果,如文献[1,2]都出现不同程度的掩盖误差。因此,本文充分利用宏块运动矢量的时空相关性,从相邻宏块的运动矢量和外推运动矢量中自适应地构造一个有限的候选运动矢量集合,减少了候选运动矢量集,从而可以有效提高差错掩盖的效率。同时,由于充分考虑到了宏块运动矢量的时空相关性,可以有效避免最优候选运动矢量的漏选,从而可以保证取得较好的掩盖效果。在此基础上,再利用边框匹配算法从候选运动矢量集中选取最佳运动矢量进行掩盖。

本算法并不是将受损宏块的外推运动矢量和相邻运动矢量全部作为候选运动矢量,而是根据宏块运动矢量的时空相关性,自适应地构造一个有限的候选运动矢量集合。构造方法如下:

- a) 如果宏块运动矢量的时域和空域相关性均较强,以外推运动矢量和相邻运动矢量构造矢量集;
- b) 如果宏块运动矢量的空域相关性均较强,以相邻运动矢量构造矢量集;
- c) 如果宏块运动矢量的时域相关性均较强,以外推运动矢量构造矢量集。

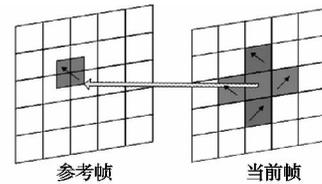


图2 宏块运动矢量的时空相关性示意图

以 8×8 的子块作为掩盖单位,由于连续的视频帧之间存在很强的时域相关性,参考帧某一宏块的运动,会在后继帧中与前帧保持着同样的运动趋势。这种特性只有在运动物体出现、消失在视频图像帧中或发生场景切换时才会例外,因此,本文将利用此规律来估计受损块的运动矢量。

1.2 自适应候选运动矢量集的差错掩盖算法

假设当前参考帧为第 n 帧,第 n 帧中的第 k 个受损块表示为 Mb_n^k ;令受损块的参考帧为前一帧,表示为第 $n-1$ 帧,其中 Mb_{n-1}^i 表示第 $n-1$ 帧的第 i 个宏块, MV_{n-1}^i 表示该块的运动矢量, Eb_n^j 表示 Mb_{n-1}^i 按 MV_{n-1}^i 运动趋势在第 n 帧得到的外推预测宏块。计算 Eb_n^j 与 Mb_n^k 相交的像素个数,取其中使得两者相交个数最大的 Eb_n^j 的运动矢量作为外推运动矢量。

这时候候选预测运动矢量集包括外推运动矢量和相邻块运动矢量,分别求它们的差值:

- a) 如果所有的差值均小于设定的阈值 $Threshold_1$,就使用边框匹配准则选择使得边框匹配误差和最小运动矢量 MV^* 作为受损块的运动矢量。其中边缘匹配准则公式如下所示:

$$MV^* = \arg \min_{MV \in C} (D_U + D_L + D_D + D_R) \quad (1)$$

$$D_U = \sum_{i=0}^7 |F_n(x+i, y) - f_n(x+i, y-1)| \quad (2)$$

$$D_L = \sum_{i=0}^7 |F_n(x, y+i) - f_n(x-1, y+i)| \quad (3)$$

$$D_D = \sum_{i=0}^7 |F_n(x+i, y+7) - f_n(x+i, y+8)| \quad (4)$$

$$D_R = \sum_{i=0}^7 |F_n(x+7, y+i) - f_n(x+8, y+i)| \quad (5)$$

$$F_n(x, y) = f_{n-1}(x + mv^x, y + mv^y) \quad (6)$$

其中: n 表示当前帧; $n-1$ 表示参考帧; (x, y) 表示受损块 Mb_n^k 左上角坐标; C 为受损块的候选预测运动矢量集; MV 是从 C 中选取的用来进行边框匹配的运动矢量; $f_n(x, y)$ 表示相邻块边界像素值; $F_n(x, y)$ 则表示受损块像素的运动补偿值; i 表示边界的八个像素; D_U, D_L, D_D, D_R 分别表示上、左、下、右像素与运动补偿值的绝对差分和; MV^x 表示 MV 的横向运动矢量值; MV^y 表示 MV 的纵向运动矢量值。

- b) 如果其中有差值大于设定的阈值 $Threshold_1$,说明要么是外推运动矢量出现误差,要么是相邻块与受损块之间并不平滑,有边缘信息,此时利用边缘匹配准则就有可能出现误差。通过前一帧的信息判断相邻块与受损块是否同一物体来对候选预测运动矢量集进行筛选。假设受损块 Mb_n^k 的相邻块为 Mb_{n-1}^m ,找到该块在前帧对应的块 Mb_{n-1}^i ,假设 Mb_{n-1}^i 的相邻块为 Mb_{n-1}^p, Mb_{n-1}^q 的位置与当前帧受损块 Mb_n^k 和其相邻块 Mb_{n-1}^m 的位置关系有关,若相邻块在受损块的上、下、左、右,则 Mb_{n-1}^p 对应 Mb_{n-1}^i 的下、上、右、左块,求 Mb_{n-1}^p 与 Mb_{n-1}^i 两者运动矢量的

绝对差值,如果差值大于设定的阈值 Threshold_2 ,就认为该相邻块与受损块不是同一物体,否则为同一物体。

c)重复上述过程直至所有相邻块都与受损块进行了是否同一物体的判断。如果最后相邻块与受损块均不是同一物体,直接选取外推运动矢量作为受损块运动矢量进行差错掩盖;否则,由于判断同一物体并不是绝对正确,还需再次进行步骤 d) 的判断。

d)将这些经过判断后与受损块是同一物体的相邻块的运动矢量与外推运动矢量求差值。若差值均大于设定阈值 Threshold_3 (本文中取值 5),由这些使得差值大于设定阈值的运动矢量构成候选预测运动矢量集,利用边框匹配准则从中选取最优的运动矢量作为受损块的运动矢量进行差错掩盖;否则从中选取使得差值小于阈值的运动矢量构成候选预测运动矢量集,利用边框匹配准则从中选取最优的运动矢量作为受损块的运动矢量进行差错掩盖。

2 仿真结果与分析

为测试基于运动矢量外推的时域差错掩盖的掩盖性能,本实验采用 H. 264 的参考软件 JM12.2^[6] 作为编解码器,视频传输采用 3GPP/3GPP2 无线 IP 信道^[7],对参考文献[1,2]中提出的方案以及本文方案进行比较和分析,并采用峰值信噪比 PSNR 值作为客观质量评价,同时比较解码视频的主观质量效果。

本实验采用 QCIF 的 foreman 和 CIF 的 bus、Paris 序列,采用码率控制,其中 QCIF 的固定码率为 64 kbps, CIF 的固定码率为 128 kbps,编码帧数为 100 帧,顺序为 IPPP。将 H. 264 的 NAL 单元封装成 RTP 包,再经模拟无线信道传输,得到误码码流,而且将阈值 Threshold_1 、 Threshold_2 和 Threshold_3 的取值设定为 5。然后分别用文献[1,2]以及本文方案进行掩盖,将编码得到的数据放到 ITU 推荐的 3G 无线视频通用测试环境进行丢包的模拟,使用误码模式文件来模拟实际信道的比特误码,其中 bus、Paris 序列使用比特率为 128 kbps、误比特率为 5.0×10^{-4} 、移动速度为 3 km/h 的误码模式文件,而 foreman 序列则使用比特率为 64 kbps、误比特率为 5.1×10^{-4} 、移动速度为 3 km/h 的误码模式文件。最后,利用解码器解码丢包模拟后的数据得到解码客观数据和主观图像序列。针对每个误码模式文件进行 10 次实验,每次实验采用不同的误码起始位置,取 10 次实验的平均值作为最终的实验结果。对传统率失真刷新算法和本文算法的抗误码能力进行比较。每个测试序列测试 100 帧,分别统计各种码率情况的 PSNR 值。其中 bus、Paris 序列解码峰值信噪比如图 3、4 所示,foreman 序列解码峰值信噪比如图 5 所示。

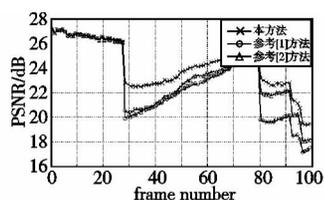


图3 Bus序列三种方案下PSNR值比较

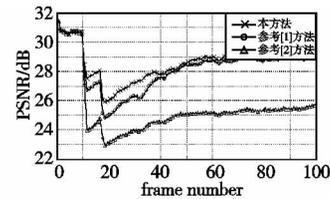


图4 Paris序列三种方案下PSNR值比较

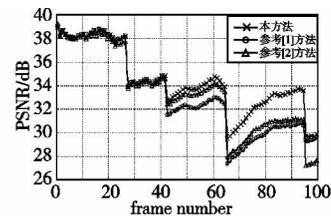


图5 Foreman序列三种方案下PSNR值比较

从图 3~5 给出的 PSNR 曲线可以看出,本文算法掩盖后的视频 PSNR 曲线与文献[1,2]提出算法掩盖后的视频 PSNR 曲线相比,获得了较大的性能提升。

从表 1 可知,本文算法相对文献[1]的方法,PSNR 值平均有 0.93 dB 的增益,相对文献[2]的方法,PSNR 值平均有 1.63 dB 的增益。因此,本文提出的方法取得了较大的增益,具有较强的差错掩盖和恢复能力。

表1 本文方法和文献[1,2]方法的平均 PSNR 值

序列	平均 PSNR 值/dB		
	本文方法	文献[1]	文献[2]
Bus_cif	24.26	22.93	23.38
Paris_cif	28.55	28.17	25.4
Foreman_qcif	34.34	33.26	33.47

3 结束语

由于差错掩盖的结果直接影响到视频接收端最终获得的视频质量,本文针对现有常用的有效时域差错掩盖方法都是基于整个丢失宏块来恢复运动信息,效率不高,不能取得很好的掩盖效果的现象,充分利用宏块运动矢量的时空相关性,能够根据外推运动矢量与各相邻运动矢量之间的差值以及受损块与相邻块是否属于同一物体,自适应地构造候选运动矢量集,从而可以有效提高差错掩盖的效果。同时,由于充分考虑到了宏块运动矢量的时空相关性,因此可以取得较好的掩盖效果。实验结果表明,这种技术具有较强的差错掩盖和恢复能力,并获得了比较满意的主观结果,可以在某种程度上弥补现有技术的一些不足之处。

参考文献:

- [1] WANG Ye-kui, HANUKSELA M M, VARSA V, et al. The error concealment feature in the H. 26L test model[C]//Proc of IEEE International Conference on Image Processing. 2002:729-732.
- [2] 彭强, 诸昌铃. 视频传输差错时域掩盖方法研究[J]. 铁道学报, 2004, 26(4):55-59.
- [3] 彭强, 张蕾, CHEN J X. 视频传输差错掩盖技术研究现状与发展趋势[J]. 西南交通大学学报, 2009, 44(4):473-483.
- [4] 黄治华, 易本顺. 一种适用于 H. 264 的时域差错掩盖算法[J]. 计算机应用, 2009, 29(5):1355-1358.
- [5] 马鑫, 杨小康, 宋利. 自适应时域差错掩盖方法[J]. 中国图象图形学报, 2007, 12(10):1782-1785.
- [6] JVT reference software H. 264/AVC JM12.2 [EB/OL]. http://bs.hhi.de/~suehring/tml/download/old_jm/.
- [7] VARSA V. ITU-T SG16 Q. 6 Doc. VCEG-N80, Common test conditions for RTP/IP over 3GPP/3GPP2[S]. 2001.