

H.264 高清视频解码实时错误掩盖算法

谢 涛, 李志华, 黄轶伦

(浙江大学数字技术及仪器研究所, 杭州 310027)

摘 要: 针对 IP 网络丢包条件下的 H.264 高清视频实时解码问题, 分析高清视频码流的特点, 提出一种实时错误掩盖算法。该算法利用丢失片的边缘宏块信息, 以垂直距离为权值加权平均预测得到错误宏块的运动矢量, 进而完成错误掩盖。实验表明, 与 Joint 模型中的错误掩盖算法相比, 该算法提升了重建图像的主观质量和客观质量, 计算复杂度较低, 错误掩盖效果较好, 适用于高清实时解码。

关键词: H.264 标准; 错误掩盖; 高清视频; 实时解码; 运动矢量估计

Real-time Error Concealment Algorithm for H.264 High Definition Video Decoding

XIE Tao, LI Zhi-hua, HUANG Yi-lun

(Institute of Advanced Digital Technology and Instrument, Zhejiang University, Hangzhou 310027, China)

【Abstract】 Aiming at real-time decode H.264 video on IP network with packet losses, an effective error concealment algorithm is proposed depending on the analysis of high definition video streams. By use of the edge macro-blocks information of the lost slice, the motion vector of corrupted macro-block is predicted, and the error concealment is completed. Experimental results show that, compared with the error concealment of Joint Model(JM), the proposed algorithm improves the objective quality and subjective quality of reconstructed images. The algorithm does not increase the complexity of the decoding, but achieves better recovery results. It is ideal for real-time decoding for high definition video.

【Key words】 H.264 standard; error concealment; high definition video; real-time decoding; motion vector estimation

DOI: 10.3969/j.issn.1000-3428.2011.14.003

1 概述

随着处理器能力的提升,在视频监控和视频会议等领域,720 P(1 280×720 像素)和 1 080 P(1 920×1 080 像素)等高清分辨率使用得越来越多。在 IP 网络上实时传输标准带宽为数兆到数十兆的高清 H.264 码流,信道拥塞或电平干扰等问题引起的视频丢包或误码,会导致解码图像质量严重降低。因此,需要在解码端引入错误掩盖技术以改善重建图像质量,使之从主观上更加接近原始视频。

错误掩盖算法主要包括空域错误掩盖和时域错误掩盖。空域错误掩盖^[1]利用视频图像的空间冗余,使用错误宏块周围可用的像素对错误宏块进行掩盖,主要方法包括双线性插值和方向插值等。时域错误掩盖根据参考帧重建出错误宏块,其关键在于运动矢量的估计。例如时域替换^[2]方法是运动矢量为零的时域掩盖方法,它的运算量极低,易于实现,但是在运动量大或运动复杂的场景下很容易产生错位感,恢复质量一般。边界匹配算法(Boundary Matching Algorithm, BMA)^[2]利用边界匹配度估计错误宏块的运动矢量,但是运算量较大,不适于实时解码应用。

针对实时解码应用中的错误掩盖,研究人员提出了很多算法。文献[3]提出了基于周围宏块运动矢量的掩盖方法。该方法通过取中值和均值等一系列运算预测错误宏块运动矢量,但该方法利用已掩盖的错误宏块信息预测其他错误宏块,容易造成差错累积。文献[4]将丢失的片(Slice)划分成子片,以子片为基本单位进行运动矢量估计和运动补偿重建。该方法运算量很小,但是由于处理单位尺寸变大,在复杂运动场景下,其预测精度容易降低。文献[5]提出的方法以运动矢量的斜率为权值,计算周围块运动矢量的加权平均得到错误块

的运动矢量,但是以斜率为权值不能很好地反映周围块运动信息和错误块运动信息的关系。

本文通过对高清视频码流的特点分析,提出了一种低运算量的实时错误掩盖算法。

2 高清视频码流的特点分析

为分析高清视频码流中运动分割的特点,将高清序列缩放成更小的分辨率,使用相同的配置文件进行编码,码流中宏块的各种运动分割的分布如图 1 所示。可以看出,随着分辨率的变大,码流中 16×16 块的比例越来越大,而 8×8 块以下的比例越来越小。在高清视频中,除了运动复杂的宏块,其他宏块绝大多数都使用 8×8 块以上的运动分割,因此,在进行错误掩盖的过程中,为了效率与质量的平衡,将最小的错误掩盖单位设为 8×8 块。

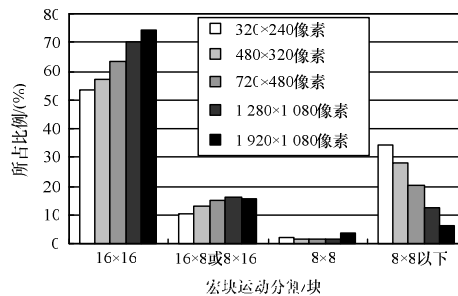


图1 不同分辨率下各种宏块运动分割的分布

基金项目: 国家自然科学基金资助项目(61001170)

作者简介: 谢 涛(1985—),男,硕士研究生,主研方向:视频编解码;李志华,讲师;黄轶伦,副教授

收稿日期: 2010-12-30 **E-mail:** tao_xie@live.cn

本文对 P 帧码流中的宏块间距离和宏块间运动矢量差异进行了统计,结果如图 2 所示。可以看出,宏块间距离越近,其运动矢量的相关性越强。因此,解码器可以利用距离当前错误宏块较近的正确宏块信息进行错误掩盖。

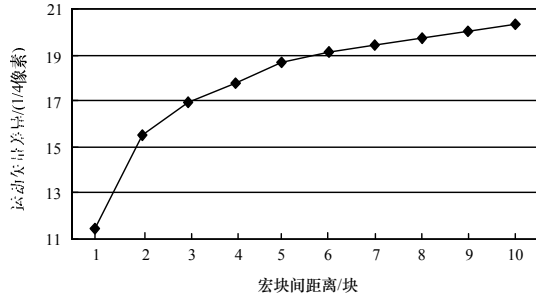


图 2 宏块的运动差异和宏块垂直距离的关系

令丢失片上方正确解码的宏块为上边缘,下方正确解码的宏块为下边缘。由于高清视频中的片一般呈长条形,通常丢失片中的错误宏块距离上下边缘较近,而左右边缘在大多数情况下不存在或者由于距离比较远而与错误宏块的运动相关性不大。因此,本文算法直接利用上下边缘的宏块运动信息估计出错误宏块的运动矢量,从而重建出错误宏块。

3 高清视频实时错误掩盖算法

3.1 基于距离加权平均的运动矢量估计

当丢失片的上下边缘都存在时,根据第 2 节对宏块间距离与宏块间运动矢量的相关性分析,距离当前错误宏块近的边缘宏块信息对错误掩盖的帮助较大。因此,本文算法以垂直距离为权值的加权平均预测方法来估计错误宏块的运动矢量,再利用估计运动矢量完成运动补偿,替换错误宏块,达到错误掩盖的目的。

如图 3 所示,丢失片中当前正在处理的错误宏块为 X ,与之相关的正确解码宏块分别有 A (顶部宏块)、 C (顶部偏左宏块)、 D (顶部偏右宏块)、 B (底部宏块)、 E (底部偏左宏块)、 F (底部偏右宏块)。宏块错误掩盖单位最小可为 8×8 子块,每个子块有一个用于错误掩盖的运动矢量。

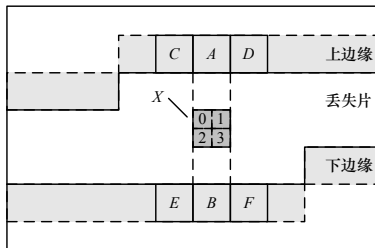


图 3 宏块分割和相关宏块的搜索

以 $X0$ 子块为例说明每个子块的运动矢量估计过程,其余子块类推。

首先利用图 3 中的上边缘宏块 A 、 C 、 D 计算当前子块 $X0$ 的上方预测矢量 mv_{top} 。令 $mv_0 = mv_{A2}$, $mv_1 = mv_{C3}$, 若宏块 A 在垂直方向上有分割,则 $mv_2 = mv_{A3}$; 否则, $mv_2 = mv_{D2}$ 。这 3 个矢量的均值和方差计算如式(1)和式(2):

$$\overline{mv} = \sum_{i=0}^2 mv_i / 3 \quad (1)$$

$$\Delta = \sum_{i=0}^2 |mv_i - \overline{mv}|^2 / 3 \quad (2)$$

如式(3)所示,根据方差的大小决定 mv_{top} 的预测方法。如果当前 3 个矢量方差小,则直接使用 A 宏块中子块 2 的运动矢量 mv_{A2} ,否则用 3 个 mv_{A2} 、 mv_{C3} 、 mv_{D2} 的中值预测 mv_{top} 。

$$mv_{top} = \begin{cases} mv_{A2} & \text{if } \Delta \leq \tau \\ \text{median}(mv_{A2}, mv_{C3}, mv_{A3}) & \text{if } \Delta > \tau \end{cases} \quad (3)$$

同理,利用图 3 中的下边缘宏块 B 、 E 、 F 计算出下方预测矢量 mv_{btm} ,然后以错误宏块到顶部宏块的距离 D_{top} 和到底部宏块的距离 D_{btm} 为权值进行加权平均来估计 $X0$ 子块的运动矢量:

$$mv_0 = (mv_{top} \times D_{btm} + mv_{btm} \times D_{top}) / (D_{btm} + D_{top}) \quad (4)$$

3.2 高清视频实时错误掩盖算法流程

本文提出的高清视频实时错误掩盖算法的流程如图 4 所示。在整帧解码结束时,检查是否发生码流丢失或解码错误,若发生,开始错误掩盖,对丢失的片依次执行,直至完成所有的丢失片。

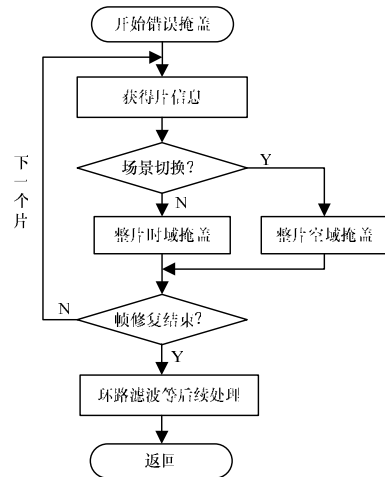


图 4 高清实时错误掩盖算法的流程

在处理 P 帧中的丢失片时,首先根据丢失片上下边缘的 Intra 块百分比来判断场景检测。判断方法如式(5)所示:

$$\text{switchFlag} = \begin{cases} \text{true} & \text{if } N(\text{intra}) / N(\text{mb}) > \sigma \\ \text{false} & \text{if } N(\text{intra}) / N(\text{mb}) \leq \sigma \end{cases} \quad (5)$$

若检测到 P 帧发生场景切换,视频前后帧之间的时域相关性不大,使用空域掩盖算法较为合适。本文使用的空域掩盖算法基于像素点,利用上下边缘的可用像素进行线性插值。对 I 帧中的丢失片使用相同的空域掩盖算法。

若检测到 P 帧中未发生场景切换,则使用本文提出的基于距离加权平均的时域掩盖算法。时域错误掩盖算法从左到右按列依次处理,每列从上到下按宏块依次掩盖。每个宏块的优化处理过程如下:

(1)搜索顶部和底部的可用宏块,计算 mv_{top} 和 mv_{btm} 。向上搜索,得到宏块 A 、 C 、 D 。若 A 、 C 、 D 这 3 个宏块都不存在,或者存在宏块都是 Intra 块,则认为 mv_{top} 不可用;否则,若 3 个宏块中有 Intra 块,则 mv_{top} 计算过程中不包括此宏块。利用类似的方法搜索底部宏块 B 、 E 、 F ,计算 mv_{btm} 。

(2)决定宏块子块划分。若顶部宏块 A 或底部宏块 B 至少有一个满足:存在且是 Inter 块,并且宏块垂直方向有分割,则将当前列的所有宏块分为 2 列子块。

若 mv_{top} 和 mv_{btm} 都存在且差异较大,则将宏块在水平方向上划分为 2 行子块,判断方法如式(6)所示:

$$\text{divideFlag} = \begin{cases} \text{true} & \text{if } |mv_{top} - mv_{btm}| / N > \lambda \\ \text{false} & \text{if } |mv_{top} - mv_{btm}| / N \leq \lambda \end{cases} \quad (6)$$

其中, N 为宏块 A 和宏块 B 相距的宏块数; λ 为判断阈值。

这样处理使在运动简单的场景下每个宏块的子块数减小,从而降低算法运算量。

(3)子块运动矢量估计。若 mv_{top} 和 mv_{btm} 都可用, 采用如式(4)的加权平均预测子块的运动矢量, 若 mv_{top} 或 mv_{btm} 只有一个可用, 利用可用的信息预测当前列所有块的运动矢量。若 mv_{top} 和 mv_{btm} 都不可用, 使用左边相邻块的运动矢量进行预测。

(4)使用估计运动矢量完成运动补偿, 掩盖错误宏块。

4 实验结果与分析

为了评价提出的实时错误掩盖算法的有效性, 测试了4个序列(rush_hour、parkrun、shields和stockholm), 序列的分辨率均为1080P, 长度为100帧。编码时, 设I帧周期为10, I帧之间采用P帧编码, 参考帧数目为1, 每帧约22个片, 量化参数固定为28。分别对各个码流模拟随机丢包, 产生丢包码流, 再对每个丢包码流施以各种错误掩盖, 用于对比的错误掩盖方法包括:

- (1)时域替换方法^[2]: 零矢量时域掩盖。
- (2)JM14.0: 对I帧采用双线性插值做空域掩盖, 对P帧采用BMA方法做时域掩盖, 详细的处理过程见文献[2]。
- (3)本文算法, 式(3)中阈值 T 为80, 式(5)中阈值 σ 为70%, 式(6)中阈值 λ 为8。

4.1 主观质量和客观质量对比

图5是rush_hour码流的第3帧发生丢包后, JM14.0和本文算法的结果对比。从图5可以看出, JM14.0的掩盖结果中产生了很多条状马赛克, 而本文算法则表现得更加平滑。

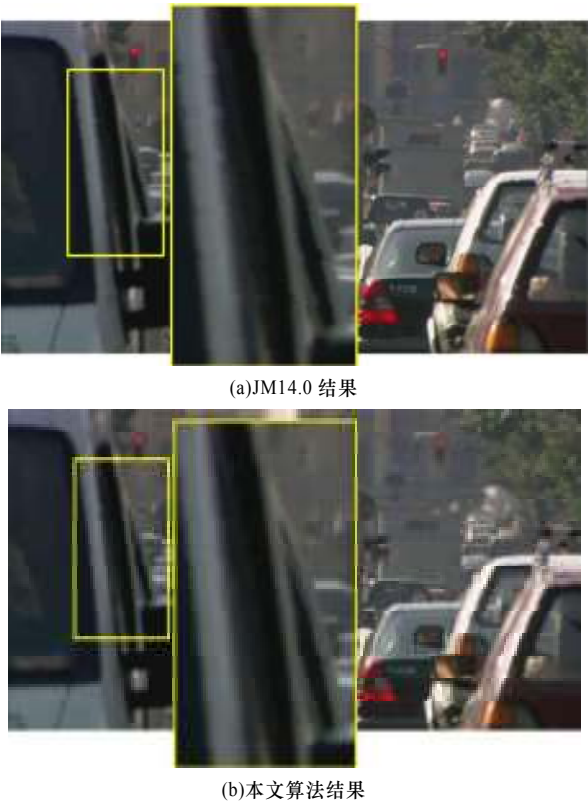


图5 rush_hour 序列错误掩盖结果对比

为了验证算法在不同网络条件下的适用性, 对比了不同丢包率下视频恢复的情况。图6是rush_hour和shields序列在不同丢包率下, 亮度峰值信噪比(PSNR)的对比。容易看出, 本文算法在所有的丢包条件下都比其他算法得到更高的PSNR, 因为本文算法很好地利用了上下边缘正确解码宏块的运动信息, 使得错误宏块的运动矢量估计误差更小。本文算法在不同网络条件下均表现出更好的恢复效果。

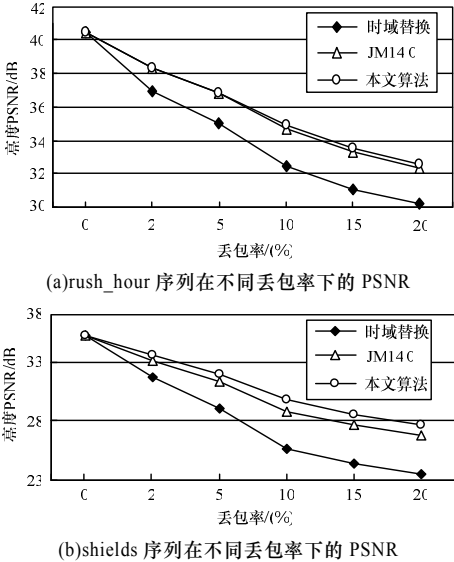


图6 不同丢包率下的 PSNR 对比

为了验证算法的普适性, 测试了多个不同运动强度和不同纹理复杂度的序列在相同丢包率下的视频恢复情况。如图7所示是不同序列的PSNR对比。可以看出, 本文算法对多种码流都能取得比其他对比方法更好的效果。

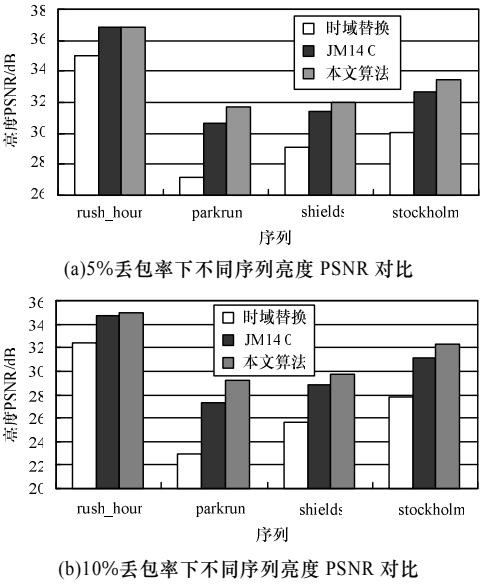


图7 不同序列的 PSNR 对比

4.2 算法执行效率

TMS320-DM6467^[6]是专为高清视频多媒体应用设计的SoC, 单片内集成了ARM926子系统、C64x+DSP子系统和高清视频图像协处理器(HDVICP)子系统。错误掩盖算法执行在C64x+DSP子系统和HDVICP子系统上。解码1080P的码流时, 各种丢包率下的解码效率如表1所示。第1列是测试的高清序列, 第2列是正确解码的效率, 第3列、第4列分别是丢包率为10%和20%条件下解码器的效率。从表1可以看出, 本文提出的错误掩盖算法不会增加解码器的运算量, 可用在其他实时解码器中。

表1 基于DM6467的算法效率统计				MCPS
序列	0%丢包率	10%丢包率	20%丢包率	
rush_hour	18.4	18.6	18.7	
stockholm	17.5	17.6	17.8	

(下转第17页)