

# 基于宏块模式预测的时域错误隐藏算法

朱冰莲, 刘剑东

(重庆大学通信工程学院, 重庆 400044)

**摘要:** 针对压缩视频码流在无线信道传输过程中由于数据丢失或错误而导致的重构图像质量的问题, 提出一种时域错误隐藏方法。在对边界匹配函数改进的基础上, 根据 H.264 视频标准具有灵活的宏块分割模式的特点, 利用受损宏块与其周围宏块的相关性预测分块模式进行运动矢量估计。实验结果表明, 在相同的 RTP 包丢失率情况下, 该算法与其他算法相比, 能恢复出更高质量的图像。

**关键词:** H.264 视频标准; 错误隐藏; 无线视频通信; 宏块模式

## Temporal Error Concealment Algorithm Based on Macroblock Mode Prediction

ZHU Bing-lian, LIU Jian-dong

(College of Communication Engineering, Chongqing University, Chongqing 400044)

**【Abstract】** Aiming at the problem that the compressed video stream will inevitable be corrupted because of the wireless channel errors, which may degrade the reconstructed image quality, this paper presents an efficient temporal error concealment method which based on improved border-matching function and flexible macroblock mode of H.264 video standard. It utilizes the relativity between lost macroblock and other surrounding macroblocks to predict the block mode, and estimates the motion vector of the lost macroblock depending on the block mode. Simulation results under 3GPP/3GPP2 wireless channel show the proposed method performs better quality than other approaches with the same RTP packet loss rate.

**【Keywords】** H.264 video standard; error concealment; wireless video communication; macroblock mode

### 1 概述

随着视频通信的广泛应用, 如何有效地实现高质量和稳定的视频传输成为研究人员追求的目标。H.263, MPEG-4 以及最新的 H.264 等视频压缩编码标准均采用了运动估计与变长编码等预测编码技术, 高效压缩后的视频数据对传输中的误码更加敏感, 降低了视频质量。

错误隐藏是针对视频传输误码提出的解决方案。可分为 2 类: 空域错误隐藏和时域错误隐藏<sup>[1-3]</sup>。空域错误隐藏利用邻近的正确块进行像素线性内插来恢复丢失块, 但这种方式容易使图像变得模糊。时域错误隐藏利用参考帧中的估计块来恢复丢失块, 通常能取得比空域错误隐藏更好的效果。

本文以 H.264 视频标准为基础提出一种有效的时域错误隐藏方案, 并在 3GPP/3GPP2 无线信道环境下进行仿真实验。

### 2 边界匹配代价函数及其改进

#### 2.1 传统的边界匹配代价函数

在传统的边界匹配算法中<sup>[4]</sup>, 如果通过运动矢量估计从参考帧中恢复出来的丢失块的边界像素与邻块的边界像素差值最小, 则可以认为当前运动矢量为最佳运动矢量。如图 1 所示, 灰色部分为采用运动矢量恢复后丢失块的像素边界,  $(X_0, Y_0)$  为丢失块左上角像素的坐标, 丢失块的大小为  $N \times N$ 。

丢失块上下左右 4 个相邻边界的绝对误差和为

$$Cost_{BAM} = \sum_{k=0}^{N-1} \left[ \sum_i |P_{ik} - P_{ik}^r| \right] \quad i \in \{T, B, L, R\} \quad (1)$$

其中  $T, B, L, R$  分别表示丢失宏块上下左右 4 个相邻的宏块;

$P$  表示与受损宏块相邻 4 块的边界像素;  $P^r$  表示候选运动矢量对应参考帧中候选宏块的 4 个边界像素, 使  $Cost_{BAM}$  值最小即为最佳运动矢量。

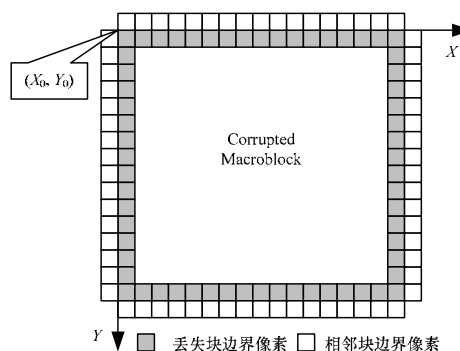


图 1 边界匹配算法

#### 2.2 边界匹配代价函数的改进

通过实验发现用边界匹配算法得到的运动矢量有时并非人眼感觉最佳的选择。图 2(a)中的方框为使用边界匹配算法恢复的丢失块, 其丢失块与邻块边界像素绝对误差为 78, 图 2(b)为正确块, 其边界像素绝对误差为 82。可见虽然采用

**作者简介:** 朱冰莲(1959 -), 女, 副教授、博士, 主研方向: 通信信号处理, 图像信号处理, 生物医学信号处理; 刘剑东, 硕士研究生  
**收稿日期:** 2009-03-04 **E-mail:** zhubl@cqu.edu.cn

边界匹配算法虽然得到了更小的绝对误差值，但恢复效果与正确图像相比，还有一定差距。



(a)采用边界匹配恢复的块 (b)正确块

图2 边界匹配算法的恢复误差

为减少误匹配，本文提出一种改进的匹配代价函数。对于2个相邻块的边界，假如它不是图像的边缘或是没有近似垂直的图像边缘通过这个边界，则可认为紧临这个边界两边的像素是单调变化的，该算法如图3所示。

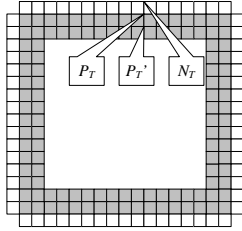


图3 改进的边界匹配算法

以上边界为例，可以得到：

$$P_T' - P_T \approx P_T - N_T \quad (2)$$

$$2P_T \approx P_T' + N_T \quad (3)$$

其中， $N_T$  为上邻块的下边界像素； $P_T$  为丢失块的上边界与  $N_T$  对应列的重建像素； $P_T'$  为丢失块上边界的第2行与  $N_T$  对应列重建像素。根据假设，可推导出改进的代价函数：

$$Cost_{BAM} = \sum_{k=0}^{N-1} \left[ \sum_i \left| 2P_{ik} - P_{ik}' - N_{ik} \right| \right] \quad i \in \{T, B, L, R\} \quad (4)$$

其中， $N_{ik}$  为4个邻块中与丢失块相邻的边界像素； $P_{ik}$  为丢失块的4个边界重建像素； $P_{ik}'$  为丢失块4个边界的第2行重建像素。使式(4)最小的运动矢量即为最佳运动矢量。相比传统的边界匹配函数，由于考虑了相邻块边界的像素值变化趋势，起到了增加匹配约束条件的作用。实验表明，这种改进在一定程度上提高了丢失块的匹配精度，恢复图像更加平滑。

### 3 基于H.264分块策略的时域错误隐藏方案

#### 3.1 树型结构的运动补偿

与传统的视频标准相比，H.264采用了更灵活的宏块分割方式<sup>[5]</sup>，每个宏块的亮度分量可按4种方式划分，即按16×16, 16×8, 8×16, 8×8的方式对每个子块进行运动补偿。如果选择8×8模式，那么该子块还可以划分为更小的块，例如8×4, 4×4等。H.264宏块分割模式如图4所示。

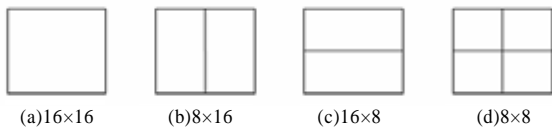


图4 H.264宏块分割模式

在H.264的编码过程中，每个子块的运动矢量和分块模式信息均需编码传输。选择大的分块尺寸意味着运动矢量和分块模式需要较少的比特数，适合于图像纹理比较平滑的区域。反之，选用小的分块尺寸则需要更多的比特来表示运动

量和分块模式的值，适用于含有丰富细节的区域。

#### 3.2 基于分块模式预测的时域错误隐藏算法

鉴于H.264灵活的分块策略，本文提出有针对性的时域错误隐藏方案：首先计算整个宏块的边界SAD，然后判断丢失宏块的分块模式，根据不同的分块模式并结合前文提出的改进边界匹配函数进行时域的错误隐藏，完成错误隐藏后根据不同的分块模式改写丢失块的状态信息。图5为不同分块模式下的错误隐藏。

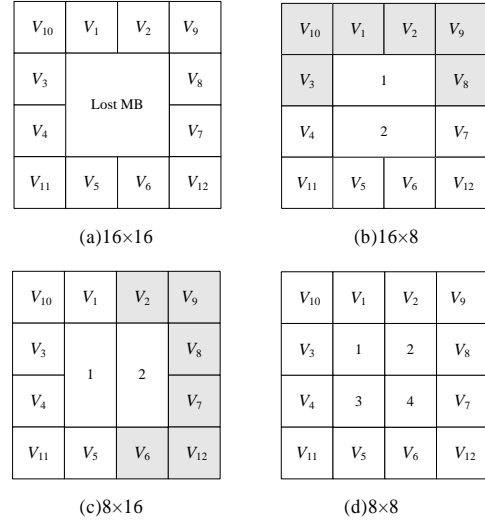


图5 不同分块模式的错误隐藏

方案的具体过程如下：

##### (1) 计算宏块的边界像素绝对误差

将整个丢失宏块作为整体进行运动矢量估计，候选运动矢量包括丢失宏块周围12个8×8块的运动矢量  $V_i, 1 \leq i \leq 12$  (见图5(a))，以及这些运动矢量的中值运动矢量  $V_{median}$ ，和当前帧前一帧相同位置的运动矢量  $V^{n-1}$ 。保存计算得到的最佳运动矢量和对应边界像素绝对误差的值。

##### (2) 判断丢失宏块的分块模式

为了更准确地判断出丢失宏块的分块模式，须找到与之相关性最强的宏块，本文采用运动矢量相关度来判断宏块相关性，即假设2个宏块的运动矢量分别为  $V_1=(x_1, y_1)$  和  $V_2=(x_2, y_2)$ ，定义2个宏块的相关度为

$$D = |V_1 - V_2| = |x_1 - x_2| + |y_1 - y_2| \quad (5)$$

其中， $D$  越小，则认为2个宏块相关性越强。利用丢失宏块周围4个邻块的运动矢量 ( $V_T^n, V_B^n, V_L^n, V_R^n$ ) 以及前一帧中与丢失宏块位置相同的宏块  $V^{n-1}$  和周围4个邻块的运动矢量 ( $V^{n-1}, V_T^{n-1}, V_B^{n-1}, V_L^{n-1}, V_R^{n-1}$ ) 来进行计算。其中， $n$  表示当前帧， $n-1$  表示前一帧。分别计算前一帧中与丢失宏块位置对应的宏块与其上下左右4个邻块的相关度和丢失宏块周围4个邻块两两之间的相关度：

$$D_i^{n-1} = |V^{n-1} - V_i^{n-1}|, \quad i \in \{T, B, L, R\} \quad (6)$$

$$D_{ij}^n = |V_i^n - V_j^n|, \quad i, j \in \{T, B, L, R\}, \quad i \neq j \quad (7)$$

则丢失宏块与其周围4个邻块的相关度可表示为

$$D_i = 3D_i^{n-1} + \sum_j D_{ij}^n, \quad i, j \in \{T, B, L, R\}, \quad i \neq j \quad (8)$$

选择具有最小相关度的宏块为丢失宏块的最相似宏块，把它的分块模式作为丢失宏块的预测分块模式。

##### (3) 不同分块模式的错误隐藏

根据步骤(2)中分块模式预测的结果，选取不同的模式进

行错误隐藏。

如果预测得到分块模式为  $16 \times 8$ ，则将丢失块分为上下 2 个  $16 \times 8$  子块依次进行运动矢量估计(如图 5(b)所示)。对于 1 块和 2 块，其候选运动矢量分别为

$$MV_1 \in \{V_1, V_2, V_3, V_8, V_9, V_{10}, V_{median}\}$$

$$MV_2 \in \{V_4, V_5, V_6, V_7, V_{11}, V_{12}, V_{median}, MV_1\}$$

将 1 块的最优运动矢量作为 2 块的候选运动矢量，同时将 1 块的下边界也作为 2 块的匹配边界。计算宏块边界像素的绝对误差，如果该值小于之前保存的宏块边界像素绝对误差值，则认为此模式更好，其运动矢量为最优运动矢量。

如果预测得到分块模式为  $8 \times 16$ ，如图 5(c)所示。则 4 个子块的候选运动矢量分别为

$$MV_1 \in \{V_1, V_3, V_4, V_5, V_{10}, V_{11}, V_{median}\}$$

$$MV_2 \in \{V_2, V_6, V_7, V_8, V_9, V_{12}, V_{median}, MV_1\}$$

1 块的最优运动矢量作为 2 块的候选运动矢量，其右边界也作为 2 块的匹配边界。同样计算宏块边界像素的绝对误差，并与之前保存的宏块边界像素绝对误差值相比较，得到最优的运动矢量。

如果预测得到分块模式为  $8 \times 8$ ，将丢失块分为 4 个子块进行运动矢量估计，如图 5(d)所示，则 4 个子块的候选运动矢量分别为

$$MV_1 \in \{V_1, V_3, V_{10}, V_{median}\}$$

$$MV_2 \in \{V_2, V_8, V_9, V_{median}, MV_1\}$$

$$MV_3 \in \{V_4, V_5, V_{11}, V_{median}, MV_1\}$$

$$MV_4 \in \{V_6, V_7, V_{12}, V_{median}, MV_2, MV_3\}$$

1 块的右边界可作为 2 块的匹配边界，其下边界可作为 3 块的匹配边界，2 块的下边界与 3 块的右边界又可以作为 4 块的匹配边界。计算宏块的边界像素误差，与保存的宏块边界像素绝对误差值比较后得到最优的运动矢量。

(4)将得到的最优运动矢量根据不同的分块模式进行  $1/4$  像素精度的插值运算，恢复出丢失宏块，保存它的运动矢量与分块模式信息，并改写该宏块的状态信息为已完成错误隐藏。

基于分块模式预测的时域错误隐藏算法的流程见图 6。

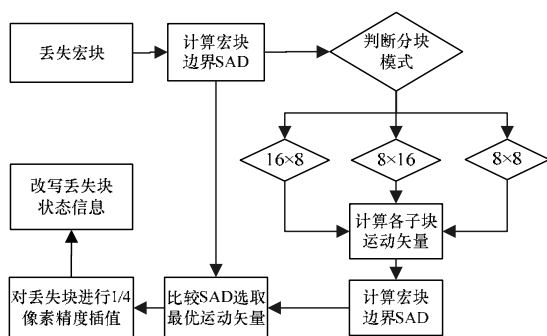


图 6 基于分块模式预测的时域错误隐藏算法流程

#### 4 实验结果

为验证该算法，本文采用 H.264 的参考编解码软件 JM10.1 作为测试平台，编码序列采用 IPPP... 帧结构，序列长度为 100 帧，I, P 帧的量化参数均为 28。使用 FMO 技术将错误宏块转化为分散形式。测试的信道环境为 3GPP/3GPP2 无线信道传输模型，传输速率约为 64 Kb/s。测试序列选用 QCIF 格式中等速率的 Foreman 和速率较低的 Carphone。分别使用时域取代<sup>[4]</sup>(TR)，H.264 参考软件中的算法(JM)，以及本文的

方法作为对比。以亮度分量的峰值信噪比 PSNR 为比较标准，结果见表 1 和表 2。

表 1 Foreman 速率下的 PSNR 值

算法	RTP 包丢失率(PLR)			
	3.14%	5.54%	10.03%	15.21%
TR	29.74	28.25	27.32	25.03
JM	31.95	30.21	29.07	26.83
本文算法	32.76	31.37	30.80	28.95

表 2 Carphone 速率下的 PSNR 值

算法	RTP 包丢失率(PLR)			
	2.96%	5.54%	10.24%	14.37%
TR	32.30	29.97	28.29	27.62
JM	33.78	31.04	29.14	28.38
本文算法	33.99	32.73	30.72	30.22

可见，针对不同视频序列，本算法在相同包丢失率的情况下相比其他算法均有不同程度的提高。限于篇幅，本文仅给出 Foreman 序列的对比图。图 7 为 Foreman 序列在约 15% 的包丢失率时，各算法 0~100 帧的 PSNR 值比较，可见本算法优于其他算法。

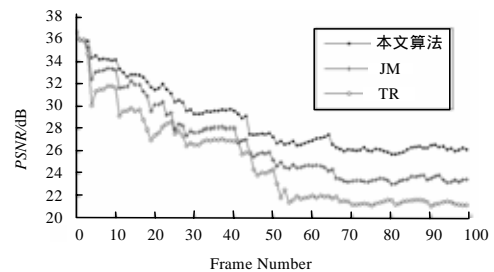


图 7 Foreman 序列本文算法与其他算法的逐帧比较

图 8 给出了 Foreman 序列包丢失率为 15.21%、传输速率为 64 Kb/s、帧数为第 54 帧时，采用不同算法恢复的单帧图像比较。从主观视觉效果来评判，本算法也明显优于其他算法。在解码速度方面，本文方法比 H.264 的方法增加了约 7% 的时间，复杂度并不高。

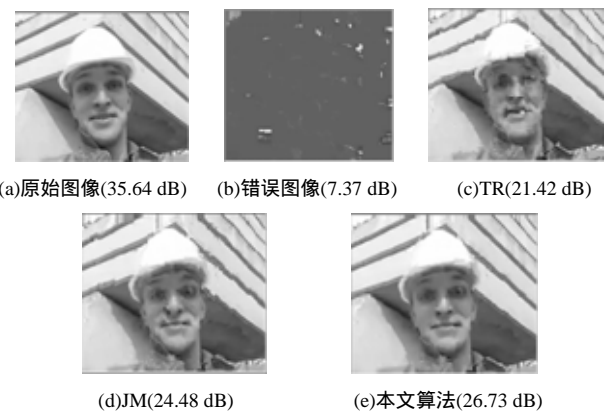


图 8 Foreman 序列不同算法的恢复图像

#### 5 结束语

本文针对视频码流经过无线信道传输数据丢失的情况，提出一种基于宏块分割模式预测的时域错误隐藏方案。实验结果表明本文的方案与 H.264 方法相比，在相同的 RTP 包丢失率情况下均有更好的图像恢复效果。本文的方法完全基于解码端，不需要编码端支持，具有广泛的适用性。

(下转第 234 页)