

# 一种基于分层率失真优化的容错性视频转码算法

韩 镇<sup>1)</sup> 胡瑞敏<sup>1),2)</sup> 陈 皓<sup>2)</sup> 李 明<sup>2)</sup>

<sup>1)</sup>(武汉大学计算机学院 武汉 430072)

<sup>2)</sup>(武汉大学国家多媒体软件工程技术研究中心 武汉 430072)

**摘 要** 容错性视频转码能够在已压缩视频流中嵌入容错性工具,以增强视频流面向无线信道的抗误码能力。但是容错性工具的嵌入在降低失真的同时也会增加码率,因此需要进行率失真优化。针对这一问题,文中提出了一种基于分层率失真优化的容错性转码算法。该算法通过在帧层和宏块层分别对重同步标记的插入和帧内宏块的刷新进行不同粒度的优化来提高算法的灵活性,通过在帧层考虑帧内宏块刷新、帧内预测和运动矢量预测的影响来提高算法的精确性,通过在宏块层减少候选模式的数量来降低算法的复杂度。实验表明,该算法与基于单层率失真优化的算法相比,信噪比可获得 0.6~1.1dB 的增益,复杂度最多可降低 25%。

**关键词** 视频转码;容错性;率失真优化;重同步标记插入;帧内宏块刷新

**中图法分类号** TP391 **DOI号**: 10.3724/SP.J.1016.2009.01152

## Error Resilient Video Transcoding Based on Layered Rate Distortion Optimization

HAN Zhen<sup>1)</sup> HU Rui-Min<sup>1),2)</sup> CHEN Hao<sup>2)</sup> LI Ming<sup>2)</sup>

<sup>1)</sup>(School of Computer, Wuhan University, Wuhan 430072)

<sup>2)</sup>(National Multimedia Software Engineering Research Center, Wuhan University, Wuhan 430072)

**Abstract** Error resilient video transcoding can insert error resilient tools in the compressed video to enhance error resilience of the video over wireless channels. However the error resilience insertion increases bit rate while reducing distortion. So the rate distortion optimization is required. In order to solve this problem, this paper proposes an error resilient video transcoding algorithm based on layered rate distortion optimization. This algorithm improves the flexibility by performing rate distortion optimization with different granularity for synchronization marker insertion and intra macroblock refresh on the frame layer and macroblock layer respectively. Moreover this algorithm improves the accuracy by accounting for the intra refresh, intra prediction, motion vector prediction on the frame layer and reduces the complexity by reducing the candidate macroblock modes on the macroblock layer. The simulation results show that the proposed algorithm improves PSNR by 0.6~1.1dB and reduces the complexity by about 25% compared with the algorithms based on single layer rate distortion optimization.

**Keywords** video transcoding; error resilient; rate distortion optimization; synchronization marker insertion; intra macroblock refresh

收稿日期: 2008-01-18; 最终修改稿收到日期: 2009-04-23. 本课题得到国家自然科学基金(60472040)、部属应用创新计划(2005YYCXHBST114)资助。韩 镇,男,1980年生,博士研究生,主要研究方向为视频编码和转码。E-mail: hanzhen\_2003@hotmail.com。胡瑞敏,男,1964年生,博士,教授,博士生导师,主要研究领域为多媒体通信、数字信号处理等。陈 皓,男,1982年生,博士研究生,主要研究方向为无线视频编码。李 明,男,1975年生,博士,副教授,主要研究方向为无线视频通信和编码。

# 1 引言

由于网络环境的异构性(如信源的异构性、信道的异构性、终端的异构性等),通用多媒体存取成为一个亟待解决的问题. 视频转码正是解决这一问题的有效技术<sup>[1-3]</sup>. 狭义的视频转码是指将已压缩的视频流从一种格式(如空间分辨率、帧率、码率等)转换为另一种格式的操作. 广义的视频转码则包括在已压缩视频流中嵌入信息(如数字水印、容错性工具等)的操作. 容错性视频转码就是指在已压缩视频流中嵌入容错性工具的转码<sup>[1-3]</sup>. 容错性视频转码能够有效增强视频流面向无线信道的抗误码能力,因此对于无线异构网络环境下的通用多媒体存取具有重要意义,在移动娱乐、移动监控等领域具有广泛的应用价值.

基本的容错性视频转码分为时域容错和空域容错两种<sup>[4-7]</sup>. 前者主要是对输入视频流进行帧内宏块刷新,后者主要是对输入视频流插入重同步标记. 虽然容错性转码可以有效地增强视频流的抗误码能力,降低无线环境下解码端的视频重建失真,但是同时也会增加码率;帧内宏块的刷新会由于减少当前帧与参考帧之间的预测而增加时域上的冗余;重同步标记的插入会增加额外的头信息,同时也会由于禁止重同步单元之间的预测而增加空域上的冗余. 因此利用率失真优化在失真与码率之间取得平衡是容错性转码的关键问题. 近年来已有研究人员对率失真优化与容错性转码的结合进行了有益尝试. 文献[8]针对已有算法的不足提出了一种基于帧层率失真优化的容错性转码算法,将帧内宏块刷新率和重同步标记插入率同时作为优化变量. 文献[9]则提出了一种基于宏块层率失真优化的容错性转码算法,但只对帧内宏块的刷新进行优化,对重同步标记的插入则通过设定阈值来选择. 上述算法使得容错性转码的效率有所提高,但仍然在以下方面存在着问题:(1)灵活性. 文献[8-9]提出的帧层和宏块层算法均是针对单一层次的,因此重同步标记的插入和帧内宏块的刷新难以在不同层次进行不同粒度的优化. 这使得算法的灵活性降低.(2)精确性. 在文献[8]提出的帧层算法中:对于失真模型,没有考虑帧内宏块刷新对信源失真的影响;对于率模型,没有考虑帧内预测和运动矢量预测对比特数的影响;对于拉格朗日乘子,没有给出具体的计算公式. 这使得

帧层算法的精确性降低.(3)复杂度. 在文献[9]提出的宏块层算法中,没有利用转码输入视频的信息减少率失真优化候选模式的数量. 这使得宏块层算法的复杂度增高.

针对上述不足,本文提出了一种基于分层率失真优化的 H.264 容错性转码算法,首先在帧层对重同步标记的插入,即片的长度进行粗粒度的率失真优化,然后在宏块层对帧内宏块的刷新,即宏块的编码模式进行细粒度的率失真优化,从而提高算法的灵活性. 在进行帧层率失真优化时,充分考虑帧内宏块刷新、帧内预测和运动矢量预测的影响,并推导出具体的拉格朗日乘子公式,从而提高帧层算法的精确性. 在进行宏块层率失真优化时,充分利用转码输入视频流的当前帧信息和转码输出视频流的前一帧信息,减少率失真优化候选模式的数量,从而降低宏块层算法的复杂度.

## 2 帧层率失真优化

本文借鉴文献[10]中的宏块层信源信道联合率失真模型,建立如式(1)所示的帧层信源信道联合率失真模型:

$$s^* = \arg \min \{ D_{FR}(s) + \lambda_{FR} R_{FR}(s) \} \\ = \arg \min \{ D_{FR}(s) + \lambda_{FR} (R_{FR}(s_0) + \Delta R_{FR}(s)) \} \quad (1)$$

其中, $s, s_0$ 分别为当前帧的片长度和帧长度, $D_{FR}(s)$ 为当前帧片长度为 $s$ 时的信源信道联合总失真, $R_{FR}(s), R_{FR}(s_0)$ 和 $\Delta R_{FR}(s)$ 分别为当前帧片长度为 $s, s_0$ 时的比特数以及前者相对于后者的比特增益, $\lambda_{FR}$ 为当前帧的拉格朗日乘子. 其中当前帧未出现误码丢包时,解码端不进行错误掩盖,信源信道联合总失真 $D_{FR}(s)$ 由信源失真和扩散失真组成;当前帧出现误码丢包时,解码端进行错误掩盖,信源信道联合总失真 $D_{FR}(s)$ 则由掩盖失真组成. 因此式(1)所示的帧层信源信道联合率失真模型可以进一步表示如下:

$$s^* = \arg \min \{ (1 - p(s))(D_{FR,s} + D_{FR,p}) + \\ p(s)D_{FR,c} + \lambda_{FR} (R_{FR}(s_0) + \Delta R_{FR}(s)) \} \quad (2)$$

其中, $p(s)$ 为当前帧片长度为 $s$ 时的丢包率, $D_{FR,s}, D_{FR,p}$ 和 $D_{FR,c}$ 分别为当前帧的信源失真、扩散失真和掩盖失真.

需要指出的是,基于帧层和基于宏块层的率失真优化是两个相互影响的过程:对片的长度进行帧层优化时,需要先确定宏块的编码模式;而对宏块的

编码模式进行宏块层优化时,又需要先确定片的长度.为了解决这一悖论,本文采取了宏块预刷新策略,其基本思想是:在对片长度进行帧层优化之前,根据宏块的刷新前率失真函数值进行预刷新,刷新前率失真函数值越大,则预刷新的可能性就越大.由于宏块刷新前后变化的主要是扩散失真和比特数,因此刷新前率失真函数的定义如下:

$$J_{MB} = D_{MB,p} + \lambda_{MB} R_{MB} \quad (3)$$

其中,  $D_{MB,p}$ ,  $R_{MB}$  和  $\lambda_{MB}$  分别为宏块的刷新前扩散失真、比特数和拉格朗日乘子.  $D_{MB,p}$  根据文献[10]中的扩散失真公式由转码输入流中的运动矢量计算得到,  $R_{MB}$  由转码输入流解码得到,  $\lambda_{MB}$  则由转码输入流中的量化步长计算得到.

预刷新的步骤如下:首先根据式(3)计算前一帧实际刷新宏块的刷新前率失真函数值,求得其平均值  $\overline{J'_{MB}}$ ;然后根据式(3)计算当前帧帧间宏块的刷新前率失真函数值  $J_{MB}$ ;如果  $J_{MB}$  大于  $\overline{J'_{MB}}$ ,则此帧间宏块被预刷新为帧内宏块.需要指出的是,预刷新策略只是对编码模式的预估计,并不是对宏块进行实际编码.

## 2.1 失真模型

由式(2)可知失真模型由  $D_{FR,s}$ ,  $D_{FR,p}$ ,  $D_{FR,c}$  和  $p(s)$  构成,其中掩盖失真  $D_{FR,c}$  可以基于转码输入流中的运动矢量根据文献[10]中的掩盖失真公式计算得到;丢包率  $p(s)$  可以根据文献[11]中的误码率/丢包率转换公式和文献[12]中的片长度/丢包率曲线计算得到;对于帧间宏块,扩散失真  $D_{FR,p}$  可以基于转码输入流中的运动矢量根据文献[10]中的扩散失真公式计算得到;对于帧内宏块,扩散失真  $D_{FR,p}$  则可以通过采用受限帧内预测来消除.因此下面主要讨论信源失真  $D_{FR,s}$ .

由于预刷新并不进行实际编码,当前帧信源失真  $D_{FR,s}$  需要通过文献[13]中的信源率失真模型求得.当前帧转码前后的信源率失真模型分别如下所示:

$$R_{FR,1} = a_{FR} \log_2 \left( \frac{b_{FR,1}}{D_{FR,s,1}} \right) \quad (4)$$

$$R_{FR,2} = a_{FR} \log_2 \left( \frac{b_{FR,2}}{D_{FR,s,2}} \right) \quad (5)$$

其中,  $R_{FR,1}$  和  $R_{FR,2}$  分别为当前帧转码前后的比特数,  $b_{FR,1}$  和  $b_{FR,2}$  分别为当前帧转码前后的方差,  $D_{FR,s,1}$  和  $D_{FR,s,2}$  分别为当前帧转码前后的信源失真,  $a_{FR}$  为当前帧信源率失真模型参数.

由式(4)、(5)可得

$$\frac{R_{FR,1}}{R_{FR,2}} = \log_2 \left( \frac{b_{FR,1}}{D_{FR,s,1}} - \frac{b_{FR,2}}{D_{FR,s,2}} \right) \quad (6)$$

由式(6)进一步推导可得

$$D_{FR,s,2} = \frac{b_{FR,2}}{\frac{b_{FR,1}}{D_{FR,s,1}} - 2^{\frac{R_{FR,1}}{R_{FR,2}}}} \quad (7)$$

其中,  $R_{FR,1}/R_{FR,2}$  由转码前后的码率之比得到,  $D_{FR,s,1}$  由如下所示的信源失真模型<sup>[13]</sup>求得

$$D_{FR,s,1} = \sum_{MB=0}^n \left( \frac{Q_{MB,1}^2}{12} \right) \quad (8)$$

其中,  $Q_{MB,1}$  为转码前当前帧宏块的量化步长,由转码输入流解码得到,  $n$  为当前帧宏块数目.

当前帧转码前的方差  $b_{FR,1}$  由当前帧未刷新宏块的方差和预刷新宏块的刷新前方差构成,两者均可以由转码输入流解码得到.当前帧转码后的方差  $b_{FR,2}$  则由当前帧未刷新宏块的方差和预刷新宏块的刷新后方差构成,其中后者由于预刷新并不进行实际编码需要根据前一帧刷新宏块刷新前后的方差估计得到.基于上述分析,  $b_{FR,1}$  和  $b_{FR,2}$  的计算公式如下所示:

$$b_{FR,1} = \sum_{MB \in A} b_{MB} + \sum_{MB \notin A} b_{MB,inter} \quad (9)$$

$$b_{FR,2} = \sum_{MB \in A} b_{MB} + \sum_{MB \notin A} b_{MB,intra} \\ = \sum_{MB \in A} b_{MB} + \sum_{MB \notin A} b_{MB,inter} \cdot \frac{\overline{b'_{MB,intra}}}{\overline{b'_{MB,inter}}} \quad (10)$$

其中,  $A$  表示预刷新宏块的集合,  $b_{MB}$  为当前帧未刷新宏块的方差,  $b_{MB,inter}$  和  $b_{MB,intra}$  分别为当前帧预刷新宏块刷新前后的方差,  $\overline{b'_{MB,inter}}$  和  $\overline{b'_{MB,intra}}$  分别为前一帧刷新宏块刷新前后方差的平均值.  $b_{MB}$ ,  $b_{MB,inter}$  和  $\overline{b'_{MB,inter}}$  由转码输入流解码计算得到,  $\overline{b'_{MB,intra}}$  由前一帧实际编码计算得到.

将式(8)~(10)代入式(7)就可以求得  $D_{FR,s,2}$ , 也就是式(2)中的  $D_{FR,s}$ .

## 2.2 率模型

由式(2)可知率模型由  $R_{FR}(s_0)$  和  $\Delta R_{FR}(s)$  构成.其中当前帧片长度为  $s_0$  时的比特数  $R_{FR}(s_0)$  由转码后的目标码率得到.下面主要讨论片长度为  $s$  时相对片长度为  $s_0$  时的比特增益  $\Delta R_{FR}(s)$ .

正如文献[12]所分析的,片的长度越小,片的数量就越多,而片结构会增加额外的片头信息,同时也会禁止片之间进行预测,因此由片数增加而导致的比特增益主要由两部分构成:片头增加的

比特数和片间预测(包括帧内预测和运动矢量预测)被禁止而增加的比特数.  $\Delta R_{FR}(s)$  的计算公式如下所示:

$$\Delta R_{FR}(s) = \frac{s_0}{s} \cdot R_{\text{head}} + n \cdot \overline{\Delta R'_{MB}} \quad (11)$$

其中,  $R_{\text{head}}$  为视频编码标准中片头的比特数,  $s_0/s$  为当片长度为  $s$  时一帧所包括的片的数目,  $\overline{\Delta R'_{MB}}$  为前一帧中宏块由于片间预测被禁止而增加的比特数的平均值,  $n$  为当前帧片间预测被禁止的宏块的数目, 也就是与参考宏块处于不同片的宏块的数目.  $n$  的计算公式如下所示:

$$n = \sum n_{MB} \quad (12)$$

如果当前宏块与参考宏块之间所有宏块的比特数之和大于片长度, 那么此宏块与参考宏块处于不同片内. 因此  $n_{MB}$  的取值如下:

$$n_{MB} = \begin{cases} 0, & \sum_{i=MB-L}^{MB} r_i \cdot \frac{R_{FR,2}}{R_{FR,1}} \leq s \\ 1, & \sum_{i=MB-L}^{MB} r_i \cdot \frac{R_{FR,2}}{R_{FR,1}} > s \end{cases} \quad (13)$$

其中,  $r_i$  为宏块  $i$  转码前的比特数, 由转码输入流解码得到;  $L$  为当前宏块与参考宏块之间的宏块数目, 由空间分辨率参数决定.

### 2.3 拉格朗日乘子

文献[10]已经证明了在信道丢包的情况下, 拉格朗日乘子  $\lambda_{FR}$  满足下列关系:

$$\lambda_{FR} = (1 - p(s)) \cdot \lambda_{FR,0} \quad (14)$$

其中,  $\lambda_{FR,0}$  为没有信道丢包时当前帧的拉格朗日乘子. 由文献[13]中的证明可知:

$$\lambda_{FR,0} = -\frac{dD_{FR,s,2}}{dR_{FR,2}} \quad (15)$$

将式(5)代入式(15)可进一步推导得到

$$\lambda_{FR,0} = \frac{D_{FR,s,2} \cdot \ln 2}{a_{FR}} \quad (16)$$

其中,  $a_{FR}$  可由式(4)进一步推导得到

$$a_{FR} = \frac{R_{FR,1}}{\log_2 \left( \frac{b_{FR,1}}{D_{FR,s,1}} \right)} \quad (17)$$

其中,  $R_{FR,1}$  由转码输入流解码得到,  $D_{FR,s,1}$  和  $b_{FR,1}$  由式(8)、(9)计算得到.

### 2.4 片长度的选择

由于片长度是灵活变化的, 并不像编码模式一样有具体的候选集. 针对这一问题, 本文采取了片长度预选择策略, 其基本思想是首先求得帧层率失真函数取极小值时的片长度, 即预选择的最

优片长度, 然后以此片长度为中心确定一个有限的候选集, 最后基于此候选集进行最终的片长度优化选择.

根据上述分析, 首先根据下式求式(2)中帧层率失真函数取极小值时的片长度, 其中与片长度  $s$  相关的变量为丢包率  $p(s)$  和比特增益  $\Delta R_{FR}(s)$ :

$$(d\{(1-p(s))(D_{FR,s} + D_{FR,p}) + p(s)D_{FR,c} + \lambda_{FR}(R_{FR}(s_0) + \Delta R_{FR}(s))\})/ds = 0 \quad (18)$$

对于丢包率  $p(s)$ , 由于与片长度  $s$  具有近似线性关系<sup>[12]</sup>, 因此可以根据下式近似计算:

$$p(s) \approx k_1 \cdot s + k_2 \quad (19)$$

其中,  $k_1$  和  $k_2$  是丢包率模型参数, 可以根据文献[11]中的误码率/丢包率转换公式和文献[12]中的片长度/丢包率曲线计算得到.

对于比特增益  $\Delta R_{FR}(s)$ , 由于片数目  $s_0/s$  越大, 与参考宏块处于不同片的宏块数目  $n$  就越大, 因此可以将  $n$  与  $s_0/s$  近似为成正比关系. 所以  $\Delta R_{FR}(s)$  可以根据下式进行近似计算:

$$\Delta R_{FR}(s) \approx \frac{s_0}{s} \cdot R_{\text{head}} + k_3 \cdot \frac{s_0}{s} \cdot \overline{\Delta R'_{MB}} \quad (20)$$

其中,  $k_3$  是  $n$  与  $s_0/s$  之间成正比关系的参数, 可以根据前一帧或前若干帧的数据统计得到.

将式(19)、(20)以及式(14)代入式(18)进行求导可得到预选择的最优片长度  $s$ :

$$\begin{aligned} & k_1 \cdot (D_{FR,c} - D_{FR,s} - D_{FR,p} - \lambda_{FR,0} R_{FR}(s_0)) - \\ & (1 - k_2) \lambda_{FR,0} (R_{\text{head}} + k_3 \cdot \overline{\Delta R'_{MB}}) \cdot \frac{s_0}{s^2} = 0, \\ & s = \sqrt{\frac{(1 - k_2) \cdot \lambda_{FR,0} \cdot (R_{\text{head}} + k_3 \cdot \overline{\Delta R'_{MB}}) \cdot s_0}{k_1 \cdot (D_{FR,c} - D_{FR,s} - D_{FR,p} - \lambda_{FR,0} \cdot R_{FR}(s_0))}} \end{aligned} \quad (21)$$

最后以  $s$  为中心, 将  $s, s+10, s+20, s-10, s-20$  作为候选集进行率失真函数计算, 选取最小函数值的片长度为最终的最优片长度.

## 3 宏块层率失真优化

与帧层信源信道联合率失真模型类似, 宏块层信源信道联合率失真模型如下所示<sup>[10]</sup>:

$$\begin{aligned} m^* &= \arg \min \{ D_{MB}(m) + \lambda_{MB} R_{MB}(m) \} \\ &= \arg \min \{ (1 - p(s)) (D_{MB,s}(m) + D_{MB,p}(m)) + \\ & p(s) D_{MB,c} + \lambda_{MB} R_{MB}(m) \} \end{aligned} \quad (22)$$

其中,  $m$  为当前宏块的编码模式,  $D_{MB}(m)$ ,  $D_{MB,s}(m)$ ,  $D_{MB,p}(m)$  和  $D_{MB,c}(m)$  分别为当前宏块

的总失真、信源失真、扩散失真和掩盖失真,  $R_{MB}(m)$  为当前宏块的编码模式为  $m$  时的比特数,  $\lambda_{MB}$  为当前宏块的拉格朗日乘子. 由文献[10]可知, 在信道丢包的情况下拉格朗日乘子  $\lambda_{MB}$  满足下列关系:

$$\lambda_{MB} = (1 - p(s)) \cdot \lambda_{MB,0} \quad (23)$$

其中,  $\lambda_{MB,0}$  为没有信道丢包时当前宏块的拉格朗日乘子, 可以根据 H.264 参考模型中的定义由量化步长计算得到. 由于丢包率  $p(s)$  和掩盖失真  $D_{MB,c}$  与宏块编码模式无关, 因此将式(23)代入式(22)后可以简化为

$$\begin{aligned} m^* &= \arg \min \{ (1 - p(s)) (D_{MB,s}(m) + D_{MB,p}(m)) + \\ &\quad (1 - p(s)) \lambda_{MB,0} R_{MB}(m) \} \\ &= \arg \min \{ D_{MB,s}(m) + D_{MB,p}(m) + \lambda_{MB,0} R_{MB}(m) \} \end{aligned} \quad (24)$$

为了降低宏块层算法的复杂度, 需要在对宏块进行转码之前减少率失真优化候选模式的数量. 对于转码前的帧内宏块, 不需要考虑刷新, 因此候选模式可以仍然限制为帧内模式; 对于转码前的帧间宏块, 需要考虑刷新, 因此候选模式的限制需要比较转码后帧内和帧间模式的率失真函数值来确定. 而宏块在转码前可利用的信息主要来自两方面: 转码输入流中当前宏块转码前的率失真信息和转码输出流中前一帧对应宏块转码后的率失真信息. 对于转码前的帧间宏块而言, 前者蕴涵了转码后帧间模式的率失真信息, 而后者蕴涵了转码后帧内模式的率失真信息. 因此综合利用这两方面的信息, 减少候选模式的数量是降低宏块层算法复杂度的有效方法. 下面分别对转码后帧内和帧间模式的率失真函数进行分析.

### 3.1 帧间模式率失真函数

对于转码前的帧间宏块, 转码后帧间模式的率失真函数如下所示:

$$\begin{aligned} J_{\text{inter}} &= D_{MB,s,2}(m_{\text{inter}}) + D_{MB,p,2}(m_{\text{inter}}) + \\ &\quad \lambda_{MB,0} R_{MB,2}(m_{\text{inter}}) \end{aligned} \quad (25)$$

其中,  $m_{\text{inter}}$  为当前宏块转码后的帧间模式,  $D_{MB,s,2}(m_{\text{inter}})$ 、 $D_{MB,p,2}(m_{\text{inter}})$  和  $R_{MB,2}(m_{\text{inter}})$  分别为当前宏块转码后模式为  $m_{\text{inter}}$  时的信源失真、扩散失真和比特数. 由于帧间模式的运动矢量可以从转码输入流中解码得到, 因此  $D_{MB,p,2}(m_{\text{inter}})$  可以直接根据文献[10]中的扩散失真公式计算得到. 下面主要讨论  $D_{MB,s,2}(m_{\text{inter}})$  和  $R_{MB,2}(m_{\text{inter}})$ .

根据文献[13]中的宏块信源失真模型可知:

$$D_{MB,s,2}(m_{\text{inter}}) = D_{MB,s,1}(m_{\text{inter}}) \cdot \frac{Q_{MB,2}^2}{Q_{MB,1}^2} \quad (26)$$

其中,  $D_{MB,s,1}(m_{\text{inter}})$  为当前宏块转码前模式为  $m_{\text{inter}}$  时的信源失真,  $Q_{MB,1}$  和  $Q_{MB,2}$  分别为转码前后当前帧宏块的量化步长.  $D_{MB,s,1}(m_{\text{inter}})$  满足如下所示的信源率失真模型:

$$R_{MB,1}(m_{\text{inter}}) = a_{MB} \log_2 \left( \frac{b_{MB,\text{inter}}}{D_{MB,s,1}(m_{\text{inter}})} \right) \quad (27)$$

其中,  $R_{MB,1}(m_{\text{inter}})$  为当前宏块转码前的比特数,  $a_{MB}$  为当前宏块信源率失真模型参数,  $b_{MB,\text{inter}}$  为当前宏块模式为  $m_{\text{inter}}$  时的方差. 由式(27)可得

$$D_{MB,s,1}(m_{\text{inter}}) = \frac{b_{MB,\text{inter}}}{2^{\frac{R_{MB,1}(m_{\text{inter}})}{a_{MB}}}} \quad (28)$$

其中,  $R_{MB,1}(m_{\text{inter}})$  和  $b_{MB,\text{inter}}$  可以从转码输入流中解码计算得到,  $a_{MB}$  则基于已实际编码宏块的数据根据 H.264 参考模型中的线性衰退模型预测得到.

将式(28)代入式(26)可求得  $D_{MB,s,2}(m_{\text{inter}})$ , 而  $R_{MB,2}(m_{\text{inter}})$  的计算公式如下:

$$R_{MB,2}(m_{\text{inter}}) = a_{MB} \log_2 \left( \frac{b_{MB,\text{inter}}}{D_{MB,s,2}(m_{\text{inter}})} \right) \quad (29)$$

将式(26)、(29)代入式(25)即可求得  $J_{\text{inter}}$ .

### 3.2 帧内模式率失真函数

对于转码前的帧间宏块, 转码后帧内模式的率失真函数如下所示:

$$\begin{aligned} J_{\text{intra}} &= D_{MB,s,2}(m_{\text{intra}}) + D_{MB,p,2}(m_{\text{intra}}) + \\ &\quad \lambda_{MB,0} R_{MB,2}(m_{\text{intra}}) \end{aligned} \quad (30)$$

其中,  $m_{\text{intra}}$  为当前宏块转码后的帧内模式,  $D_{MB,s,2}(m_{\text{intra}})$ 、 $D_{MB,p,2}(m_{\text{intra}})$  和  $R_{MB,2}(m_{\text{intra}})$  分别为当前宏块转码后模式为  $m_{\text{intra}}$  时的信源失真、扩散失真和比特数. 由于帧间宏块的运动矢量可以从转码输入流中解码得到, 因此可以确定当前宏块在前一帧的对应宏块, 从而获得其提供的帧内模式率失真信息.  $D_{MB,p,2}(m_{\text{intra}})$  可以通过采用受限帧内预测来消除, 因此下面主要讨论  $D_{MB,s,2}(m_{\text{intra}})$  和  $R_{MB,2}(m_{\text{intra}})$ .

由于当前宏块的运动搜索以  $4 \times 4$  块为最小单位, 所以获取对应宏块的率失真信息也以  $4 \times 4$  块为单位. 如果对应  $4 \times 4$  块进行了帧内编码, 则将其在最优帧内模式时的信源失真、方差作为帧内模式率失真信息; 如果对应  $4 \times 4$  块没有进行帧内编码, 则将其对应  $4 \times 4$  块在帧内模式时的信源失真、方差作为帧内模式率失真信息.

将上述获取的对应  $4 \times 4$  块在帧内模式时的信源

失真、方差记为  $D'_{B,s,2}(m_{\text{intra}})$ 、 $b'_{B,\text{intra}}$ , 则  $D_{MB,s,2}(m_{\text{intra}})$  和  $R_{MB,2}(m_{\text{intra}})$  分别通过下式计算得到

$$D_{MB,s,2}(m_{\text{intra}}) = \sum_{B=1}^{16} \left( D'_{B,s,2}(m_{\text{intra}}) \cdot \frac{Q_{B,2}^2}{Q_{B,2}'^2} \right) \quad (31)$$

$$R_{MB,2}(m_{\text{intra}}) = a_{MB} \log_2 \left[ \frac{\sum_{B=1}^{16} b'_{B,\text{intra}}}{D_{MB,s,2}(m_{\text{intra}})} \right] \quad (32)$$

其中,  $Q_{B,2}$  和  $Q'_{B,2}$  分别为当前  $4 \times 4$  块和前一帧对应  $4 \times 4$  块的量化步长。

将式(31)、(32)代入式(30)就可以求得  $J_{\text{intra}}$ 。

### 3.3 候选模式的选择

根据式(25)、(30)分别求得的转码后帧间模式和帧内模式的率失真函数值  $J_{\text{inter}}$  和  $J_{\text{intra}}$ , 可以对候选模式进行选择。由于  $J_{\text{intra}}$  是基于前一帧对应宏块的信息得到的, 因此当前宏块与对应宏块的时域相关性越强,  $J_{\text{intra}}$  的误差就越小; 反之, 误差则越大。基于上述分析, 实际的帧内模式率失真函数值应当落在一个以  $J_{\text{intra}}$  为中心、以误差为半径的区间内。所以候选模式的选择可以基于以下思路进行: 如果  $J_{\text{inter}}$  大于  $J_{\text{intra}}$  区间的最大值, 则进行刷新, 候选模式限制为帧内模式; 如果  $J_{\text{inter}}$  小于  $J_{\text{intra}}$  区间的最小值, 则不进行刷新, 候选模式限制为帧间模式; 如果  $J_{\text{inter}}$  落在  $J_{\text{intra}}$  区间内, 则无法判断是否进行刷新, 候选模式为所有模式。对于当前宏块而言, 转码前帧间模式的差分系数表征了当前宏块与对应宏块的时域相关性: 系数越小, 相关性越强,  $J_{\text{intra}}$  区间半径也就越小; 反之, 相关性越弱,  $J_{\text{intra}}$  区间半径也越大。因此可以以  $4 \times 4$  块中的非零系数比  $\rho_B$  表征此时域相关性和区间半径。候选模式选择的具体步骤如下所示:

1. 根据下式分别计算最大值  $\bar{D}_{MB,s,2}(m_{\text{intra}})$ 、 $\bar{R}_{MB,2}(m_{\text{intra}})$ :

$$\bar{D}_{MB,s,2}(m_{\text{intra}}) = \sum_{B=1}^{16} \left( D'_{B,s,2}(m_{\text{intra}}) \cdot \frac{Q_{B,2}^2}{Q_{B,2}'^2} \cdot (1 + \rho_B) \right) \quad (33)$$

$$\bar{R}_{MB,2}(m_{\text{intra}}) = a_{MB} \log_2 \left[ \frac{\sum_{B=1}^{16} (b'_{B,\text{intra}} \cdot (1 + \rho_B))}{\bar{D}_{MB,s,2}(m_{\text{intra}})} \right] \quad (34)$$

2. 根据下式分别计算最小值  $\underline{D}_{MB,s,2}(m_{\text{intra}})$ 、 $\underline{R}_{MB,2}(m_{\text{intra}})$ :

$$\underline{D}_{MB,s,2}(m_{\text{intra}}) = \sum_{B=1}^{16} \left( D'_{B,s,2}(m_{\text{intra}}) \cdot \frac{Q_{B,2}^2}{Q_{B,2}'^2} \cdot (1 - \rho_B) \right) \quad (35)$$

$$\underline{R}_{MB,2}(m_{\text{intra}}) = a_{MB} \log_2 \left[ \frac{\sum_{B=1}^{16} (b'_{B,\text{intra}} \cdot (1 - \rho_B))}{\underline{D}_{MB,s,2}(m_{\text{intra}})} \right] \quad (36)$$

3. 根据下式分别计算  $J_{\text{intra}}$  的最大值  $\bar{J}_{\text{intra}}$  和最小值  $\underline{J}_{\text{intra}}$ :

$$\bar{J}_{\text{intra}} = \bar{D}_{MB,s,2}(m_{\text{intra}}) + \lambda_{MB,0} \bar{R}_{MB,2}(m_{\text{intra}}) \quad (37)$$

$$\underline{J}_{\text{intra}} = \underline{D}_{MB,s,2}(m_{\text{intra}}) + \lambda_{MB,0} \underline{R}_{MB,2}(m_{\text{intra}}) \quad (38)$$

4. 如果  $J_{\text{inter}} > \bar{J}_{\text{intra}}$ , 则候选模式限制为帧内模式; 否则, 如果  $J_{\text{inter}} < \underline{J}_{\text{intra}}$ , 则候选模式限制为帧间模式; 否则, 则候选模式为所有模式。

## 4 实验结果及讨论

我们基于 H.264 JM12.2 参考模型和空域转码结构<sup>[2]</sup>, 对本文提出的容错性转码算法、文献[8-9]中的容错性转码算法以及 JM12.2 参考模型中的非容错性算法进行了测试。实验采用 ITU 推荐的 3G 无线视频通用仿真测试环境<sup>[14]</sup>, 无线信道误码使用六种错误模板来模拟实现, 误码率分别为  $9.3\text{E-}3$ 、 $2.9\text{E-}3$ 、 $5.1\text{E-}4$ 、 $1.7\text{E-}4$ 、 $5.0\text{E-}4$ 、 $2.0\text{E-}4$ , 测试序列使用 Hall Monitor、Coastguard、Foreman、Glasgow、Paris 等 5 个测试序列, 序列与模板的对应关系根据文献[14]中表 4 设置, 其中模板 1 和 2 针对的是流媒体型业务, 分别进行了 3 次和 2 次重传, 模板 3~6 针对的是会话型业务, 不进行重传。转码输出序列的分辨率、帧率、码率根据文献[14]中表 1 设置, 转码输入序列的分辨率、帧率与输出序列相同, 码率则为输出序列的两倍。测试序列的编码帧数均为 100 帧, 不使用多参考帧运动预测, 均采用 IPPP... 的编码结构和一片一包的打包策略, 解码端启用误码掩盖策略。每组实验均采用文献[14]中表 3 给出的 10 个错误起始位置进行 10 次测试。实验所用的机器配置为 Pentium 4 3.0GHz CPU、1.00GB 内存。

表 1 给出了每个测试序列无误码条件下的原始信噪比、10 次误码测试的平均信噪比和每个测试序列的转码时间。其中算法 1~4 分别表示本算法、帧层算法<sup>[8]</sup>、宏块层算法<sup>[9]</sup>和 JM12.2 参考模型算法。由表 1 可以看出, 与 JM12.2 参考模型算法相比, 本算法、帧层算法和宏块层算法无误码条件下的原始信噪比分别有约 0.56dB、0.62dB 和 0.45dB 的降低, 而 10 次误码测试的平均信噪比则分别有约 3.71dB、2.75dB 和 3.19dB 的增益。这主要是因为本算法、帧层算法和宏块层算法在视频流中嵌入了额外的容错性工具, 从而导致在无误码条件下信噪比降低, 而在误码条件下信噪比增高。由表 1 还可以看出, 除了 Glasgow 序列, 本算法 10 次误码测试的平均信噪比始终保持最优, 与帧层算法和宏块层算法相比分别有约 1.11dB 和 0.62dB 的增益。本算法

相对于帧层算法的增益主要来源于在帧层率失真模型中考虑帧内宏块刷新、帧内预测和运动矢量预测的影响和以宏块层的模式优化选择代替帧层的模式优化选择,相对于宏块层算法的增益则主要来源于以基于率失真优化的片长度选择代替基于固定阈值的片长度选择.通过进一步分析,我们可以发现:(1)与帧层算法相比,本算法对 Hall Monitor 序列(会话型)的提升最大,有约 2.85dB.这是因为对于 Hall Monitor 这种大部分内容为静止背景,小部分内容为运动物体的序列,仅在帧层进行模式选择难以精确地对小部分运动物体进行帧内模式刷新,因而当运动物体出现误码时会导致信噪比的显著下降.(2)与宏块层算法相比,本算法对 Foreman 序列(流媒体型)的提升最大,有 1.05dB,但是对于 Foreman 序列(会话型)的提升则相对低一些,只有 0.45dB.这是因为在仿真实验中,宏块层算法所取的片长度选择阈值,是文献[12]中基于 Foreman 序

列(会话型)给出的推荐值,因此宏块层算法的性能对于 Foreman 序列(会话型)较好,对于 Foreman 序列(流媒体型)则较差.这正说明宏块层算法的适应性较差,而本算法则能在不同的测试条件下均保持较好的性能.(3)与帧层算法和宏块层算法相比,本算法对 Glasgow 序列均只有略微的提升.这是因为 Glasgow 序列存在频繁的场景切换(由 30f/s 转为 7.5f/s 后 100 帧中包括 13 帧场景切换帧),本身已具有较强的容错性,进一步提升的空间较小.除了信噪比,由表 1 还可以看出,本算法的转码时间与宏块层算法相比减少了约 25%,与帧层算法基本相当.这是因为宏块层的率失真优化需要进行实际编码,而帧层的率失真优化并不需要,所以复杂度主要集中在宏块层,而本算法通过重用信息减少了宏块层率失真优化中候选模式的数量,降低了宏块层算法的复杂度,因此本算法的转码时间与宏块层算法相比能有较明显的减少,与帧层算法相比则基本相当.

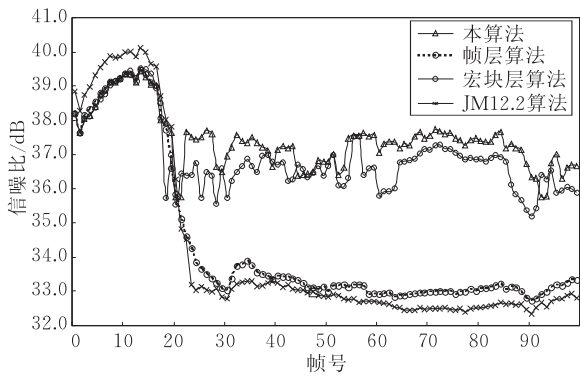
表 1 信噪比和转码时间的比较

图像序列	算法	模板 1/3/5 信噪比		模板 2/4/6 信噪比		运行时间/s
		原始信噪比/dB	平均信噪比/dB	原始信噪比/dB	平均信噪比/dB	
Hall Monitor 会话型	1	41.16	37.323	41.07	38.251	8.085
	2	41.29	34.604	41.27	35.219	7.988
	3	40.99	36.775	40.98	37.560	11.765
	4	41.43	31.866	41.43	34.715	11.813
Coastguard 流媒体型	1	33.70	29.989	33.51	29.037	11.797
	2	33.68	29.433	33.67	27.833	11.202
	3	33.75	29.691	33.59	28.623	15.625
	4	34.10	26.113	34.10	23.794	15.820
Foreman 会话型	1	34.45	30.551	34.50	31.092	12.611
	2	33.92	29.664	33.93	30.270	12.466
	3	34.68	30.264	34.69	30.602	16.637
	4	35.31	26.465	35.31	26.546	16.912
Foreman 流媒体型	1	34.42	30.557	34.23	29.340	11.790
	2	34.37	29.925	34.28	28.946	11.411
	3	34.84	29.277	34.58	28.591	16.299
	4	35.31	25.229	35.31	22.596	16.329
Glasgow 流媒体型	1	28.77	27.201	28.41	26.443	11.361
	2	28.76	27.162	28.34	26.351	10.850
	3	28.86	27.177	28.47	26.380	16.102
	4	29.15	25.694	29.15	24.840	16.252
Paris 会话型	1	31.81	28.269	31.82	28.560	39.109
	2	31.89	27.829	31.85	27.912	38.300
	3	31.88	27.603	31.92	27.790	52.637
	4	32.00	26.788	32.00	27.377	54.826

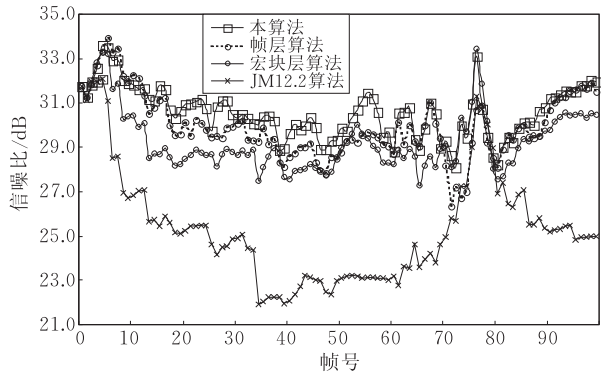
图 1 给出了 Hall Monitor 序列(会话型)和 Foreman 序列(流媒体型)的单帧信噪比,两者分别是在错误模板 4 和 1 下进行 10 次测试得到的平均值.由图 1 我们可以发现,在误码条件下本算法的单帧信噪比也基本上保持最优.图 2 和图 3 比较了 Hall Monitor 序列(会话型)和 Foreman 序列(流媒

体型)解码重构图像的主观质量.由图 2 和图 3 我们可以看出,本算法的解码重构图像具有较好的主观质量,既通过精确的帧内模式刷新避免了误差扩散导致的黑斑和阴影,又通过适当的片长度选择避免了丢包导致的掩盖失真.





(a) Hall Monitor序列(会话型)



(b) Foreman序列(流媒体型)

图 1 单帧信噪比的比较



(a) 原始图像



(b) 帧层算法



(c) 宏块层算法



(d) 本算法

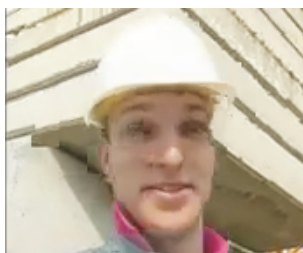


(e) JM12.2算法

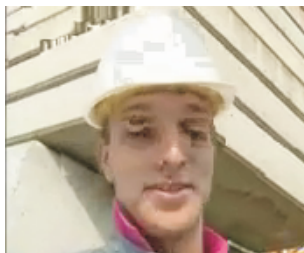
图 2 Hall Monitor 序列主观质量的比较



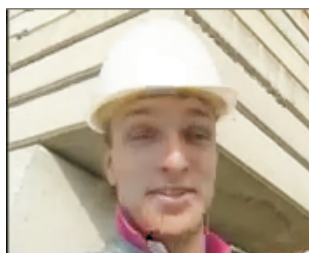
(a) 原始图像



(b) 帧层算法



(c) 宏块层算法



(d) 本算法



(e) JM12.2算法

图 3 Foreman 序列主观质量的比较



## 5 结 论

针对现有基于单层率失真优化的容错性视频转码算法在灵活性、精确性、复杂度方面的不足,本文提出了一种基于分层率失真优化的容错性视频转码算法.该算法通过在帧层和宏块层分别对重同步标记的插入和帧内宏块的刷新进行不同粒度的优化来提高算法的灵活性,通过在帧层考虑帧内宏块刷新、帧内预测和运动矢量预测的影响来提高算法的精确性,通过在宏块层减少候选模式的数量来降低算法的复杂度.仿真实验结果表明,本算法的平均信噪比与基于帧层率失真优化的算法<sup>[8]</sup>和基于宏块层率失真优化的算法<sup>[9]</sup>相比有较为显著的提高,并且对于不同特征的序列始终保持最优,同时本算法的复杂度与宏块层算法相比降低了约 25%,与帧层算法基本相当.作为最新一代视频编码标准,H.264 提供了一些新的容错工具,如灵活的宏块排序 FMO 等,因此在下一步的工作中我们将尝试在基于率失真优化的容错性转码框架中引入这些新的工具.

## 参 考 文 献

- [1] Vetro A, Christopoulos C, Sun H F. Video transcoding architectures and techniques: An overview. *IEEE Signal Processing Magazine*, 2003, 20(2): 18-29
- [2] Ahmad I, Wei X H, Sun Y et al. Video transcoding: An overview of various techniques and research issues. *IEEE Transactions on Multimedia*, 2005, 7(5): 791-804
- [3] YU Sheng-Fa, Chen Zeng-Ping, Zhuang Zhao-Wen. Review of video transcoding for network video applications. *Journal on Communications*, 2007, 28(1): 111-118(in Chinese)  
(余圣发, 陈曾平, 庄钊文. 针对网络视频应用的视频转码技术综述. *通信学报*, 2007, 28(1): 111-118)
- [4] De los Reyes G, Reibman A R, Chuang J C et al. Video transcoding for resilience in wireless channels//*Proceedings of the International Conference on Image Processing (ICIP'98)*. Chicago, USA, 1998: 338-342
- [5] Dogan S, Cellatoglu A, Uyguroglu M et al. Error-resilient video transcoding for robust internetwork communications using GPRS. *IEEE Transactions on Circuits and Systems for Video Technology*, 2002, 12(6): 453-464
- [6] Chiou H-J, Lee Y-R, Lin C-W. Content-aware error-resilient transcoding using prioritized intra-refresh for video streaming. *Journal of Visual Communication and Image Representation*, 2005, 16(4): 563-588
- [7] Eminsoy Sertac, Dogan Safak, Kondo Ahmet M. Transcoding-based error-resilient video adaptation for 3G wireless networks. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007(2): 29-39
- [8] Vetro A, Xin J et al. Error resilient video transcoding for wireless communications. *IEEE Wireless Communications*, 2005, 12(4): 14-21
- [9] Seong Hwan Jang, Jayant N. An efficient error resilient technique for applications of one-way video using transcoding and analysis by synthesis//*Proceedings of the IEEE Global Telecommunications Conference Workshops*. Texas, USA, 2004: 428-432
- [10] Zhang Y, Gao W, Lu Y et al. Joint source-channel rate-distortion optimization for H.264 video coding over error-prone networks. *IEEE Transactions on Multimedia*, 2007, 9(3): 445-454
- [11] Wenger S. On the equivalence of BER and packet loss rates. *JVT Doc JVT-B025*, Geneva, Switzerland, 2002
- [12] Stockhammer T, Hannuksela M M, Wiegand T. H.264/AVC in wireless environments. *IEEE Transactions on Circuits and Systems on Video Technology*, 2003, 13(7): 657-673
- [13] Wiegand T, Girod B. Lagrange multiplier selection in hybrid video coder control//*Proceedings of the International Conference on Image Processing (ICIP'01)*. Thessaloniki, Greece, 2001: 542-545
- [14] Roth G, Sjöberg R, Liebl G et al. Common test conditions for RTP/IP over 3GPP/3GPP2. *ITU-T SG16 Doc VCEG-N80*, Santa Barbara, CA, USA, 2001



**HAN Zhen**, born in 1980, Ph. D. candidate. His research interests include video coding and transcoding.

**HU Rui-Min**, born in 1964, Ph. D., professor, Ph. D. supervisor. His research interests include multimedia communication and digital signal processing.

**CHEN Hao**, born in 1982, Ph. D. candidate. His research interests focus on wireless video coding.

**LI Ming**, born in 1975, Ph. D., associate professor. His research interests include wireless video communication and coding.

Background

The work in this paper is one of the series studies of Adaptive Wireless Video Transmissions in the National Multimedia Software Engineering Research Center. The main objective of the project is to establish effective schemes to translate video adaptively through wireless heterogeneous networks. As a key technology to solve this problem, wireless video transcoding can make the video content adaptive to the capabilities of diverse networks and client devices. Thus, it is one of the essential components for current and future multimedia systems that aim to provide universal access in a wireless heterogeneous network environment, such as mobile e-business, mobile entertainment, mobile surveillance. The center has worked in the wireless video transcoding including rate control and mode decision for transcoding since 2005. These works are supported by National Natural Science Foundation of China (No. 60472040) and Innovation Plan of Ministry of Public Security (No. 2005YYCXHBST114).

As one of key technologies of wireless video transcoding, error resilient video transcoding can insert error resilient tools in the compressed video to enhance error resilience for video transmission in a wireless heterogeneous network environment. However the error resilience insertion increases bit

rate while reducing distortion. So the rate distortion optimization is required. In order to solve this problem, researchers have proposed some error resilient video transcoding algorithms based on single layer rate distortion optimization. However the flexibility, accuracy and complexity of these algorithms still need to be improved. So this paper proposes an error resilient video transcoding algorithm based on layered rate distortion optimization. This algorithm improves the flexibility by performing rate distortion optimization with different granularity for synchronization marker insertion and intra macroblock refresh on the frame layer and macroblock layer respectively. Moreover this algorithm improves the accuracy by accounting for the intra refresh, intra prediction, motion vector prediction on the frame layer and reduces the complexity by reducing the candidate macroblock modes on the macroblock layer. The simulation results show that the proposed algorithm improves PSNR remarkably and achieves the best PSNR compared the reference algorithm while maintaining low complexity. The next step is to introduce the novel error resilient tools adopted by H.264 in the video transcoding scheme.