

一种基于条带划分的低延迟帧内编码方法

姚春莲^{1,2} 阮秋琦² 姜 东² 万丽莉²

(北京工商大学计算机与信息工程学院 北京 100037)¹

(北京交通大学现代信息科学与网络技术北京市重点实验室 北京 100044)²

摘 要 现有编解码系统所产生的固有延迟比较大,例如对于标清视频,系统延迟要大于 260ms,因此难以用于对延迟有严格要求的系统中。为此,研究了一种新的低延迟帧内编解码结构,在该结构中,将输入的视频划分成若干个条带,在采集端以条带为单位进行采集,以有效降低采集延迟;同时,给出了以条带为基本单位的 I 帧编码方法,对每一编码条带下采样,将其分成 4 个子图,以子图为单位进行预测,以提高预测的精度,进而降低编码延时。实验表明,新的编码结构在采集和编码过程中能有效降低帧内编码延迟,在保证恢复图像质量的同时,可以将延迟降低到 150ms。

关键词 低延迟,条带,采集,编码,帧内,视频

中图法分类号 TP912.3, TN919.81 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.11.037

Slice-based Partition Low Delay Intra Frame Coding Method

YAO Chun-lian^{1,2} RUAN Qiu-qi² JIANG Dong² WAN Li-li²

(School of Computer and Information Engineering, Beijing Technology and Business University, Beijing 100037, China)¹

(Key Laboratory of Advanced Information Science and Network Technology of Beijing, Beijing Jiaotong University, Beijing 100044, China)²

Abstract The inherent delay of the standard codec system is relatively large, for example, for standard definition video, the system delay exceeds 260ms. It is very difficult to be applied in the system that has strict requirement on delay. On the basis of existing standards, a new low delay intra coding structure was provided in the paper. The input video is divided into a number of slices, and slice is the basic code unit at the acquisition end. Slice-based acquisition method can effectively reduce the acquisition delay, at the same time, I-frame coding method is different from existing standard in which each acquisition slice is the basic coding unit and it is divided into four sub-pictures. In order to improve the prediction accuracy, one of sub-picture was chosen as prediction unit. Experiments show that the new structure can effectively reduce intra coding delay and acquisition delay, which can be reduced to 150ms.

Keywords Low delay, Slice, Acquisition, Coding, Intra, Video

1 引言

随着网络技术和终端处理能力的不断提高和发展,为了不断提高压缩性能,ITU 和 ISO 组织推出了一系列视频编码标准,包括 ITU 组织的 H.26x 系列和 ISO 组织的 MPEG-x 系列标准,以及最新制定的 HEVC 标准。最新的 HEVC 标准致力于满足用于用户的 1) 高清, 2) 3D, 3) 移动无线, 以满足新的家庭影院、远程监控、数字广播、移动流媒体、便携摄像、医学成像等新领域的需求。在 HEVC 标准中有多种配置模式,包括 HE(High Efficiency)高性能、LC(Low Complexity)低复杂度配置。

这些编码标准主要分为两类:一类是面向存储的,另一类是面向传输的。以 H 系列为主是面向传输的,以 Mpeg 系列为主是面向存储的。视频编码标准虽然制定的组织不同,完成的年代及应用的背景方面也不相同,但是采用的基本编码

框架却是相同的,即多采用运动补偿+DCT 的基本框架。这种框架下视频帧一般分为 I(Intra-frame)、P(Predictive-frame)、B(Bidirectionally predicted-frame) 3 种类型:I 帧通过变换、量化等过程完成编码;P 帧以前向已编码帧的重建图像为参考,进行运动补偿后编码残差;B 帧则以前后双向已编码帧的重建图像为参考,进行运动补偿后编码残差。

在 3 种类型的视频帧中,虽然 I 帧的个数比较少,但其每帧编码位数却远多于 P、B 帧,在最终生成的码流中也占有相当比例,在 H.264 标准中,I/P 帧的压缩比分为:1:P=1:3~5,也即 I 帧的位率是 P 帧的 3~5 倍;而在 HEVC 标准中,随着新的复杂技术的引入,I 帧与 P 帧的压缩比进一步扩大,在某些视频中,可以达到 1:10。对于输出码率恒定(CBR)的视频流,I 帧的码率突然增大,将直接导致后续 P/B 帧的编码位数的减少,进而影响到恢复图像和预测图像的质量。提高 I 帧的压缩比,不仅对于视频质量的稳定性与连续性起到了至关

到稿日期:2013-06-25 返修日期:2013-08-16 本文受国家自然科学基金(61103124),现代信息科学与网络技术北京市重点实验室资助课题(XDXX1301),北京市自然科学基金资助项目(4123104)资助。

姚春莲(1973—),女,博士,主要研究方向为视频处理、嵌入式系统设计,E-mail:yaoc@btbu.edu.cn;阮秋琦(1944—),男,博士,教授,主要研究方向为数字图像处理、模式识别;姜 东(1976—),男,博士,主要研究方向为视频处理;万丽莉(1979—),女,副教授,主要研究方向为虚拟现实、三维立体建模。

重要的作用,而且会降低编解码系统的延迟。也即,如果 I 帧的码率过高,将会导致占用更多的网络带宽来传输压缩码流。

在视频编码过程中,尤其是移动平台下,对于带宽和复杂度都有要求的情况下,希望编解码算法达到低延迟和低复杂度。本文将重点分析低延迟,在视频的编码过程中,延迟主要由两个原因引起:

(1)编解码固有的时间,如果编解码速度都能够达到实时,那么将不会有延时产生。

(2)信道传输,如果信道足够宽,没有拥挤,也不会有延迟产生。

如果不考虑信道的问题,那么如何降低编解码的固有延迟就显得十分重要。本文将从采集输入端和编码处理过程来降低延迟,提高 I 帧的预测效果,降低码流。降低码流将会降低对缓冲区的空间需求,进而降低延迟。概括地说,主要通过改变采集和编码的基本单元,提高预测效果,降低码流,进而减小对缓冲区的需求,减少延迟。

本文将通过对视频编解码延迟进行分析,给出适合于 I 帧编码的低延迟的编码结构,在该结构中,将主要考虑到采集端、编码端两个方面。

本文后续部分包括如下内容:(1)编解码延迟分析;(2)基本条带划分的采集低延迟处理;(3)基于下采样的编码低延迟处理。

2 视频编解码延迟分析

一个编解码系统的延迟由以下几个部分组成(见图 1),即图像信息采集、压缩处理、发送缓冲区、数据链路传输、接收缓冲区、解压缩、图像数据格式转换和显示等环节的延迟。在帧率为 25 帧/秒的实时编码情况下,一帧图像采集完成后编码器在 40ms 内完成压缩处理;当编码的帧类型为 I-P 时,编码输出的位数比一般为 8:3,因此通常设置的缓冲区大小为每帧平均码流量的 3.5~4 倍,以满足 I 帧码流的缓存。通过数据链路传输后,解码端收到压缩码流就进行解压缩处理,最后在显示设备上显示。

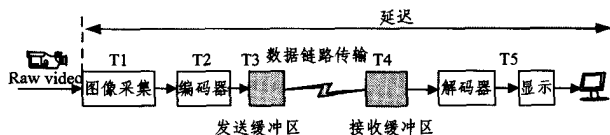


图 1 系统延迟示意图

在图 1 中,假定以下两种情况:

1)数据链路传输在整个编解码过程中都是固定的(CBR 类型信道),不计入延迟。

2)在实时采集、处理、传输系统中,接收缓冲区与解压缩过程是并行处理的,可以只计算一个时间。

在标准的编码框架下,以帧为基本单元,那么编解码系统总延迟分析如下:

采集时间 T1:采集一帧图像所需的时间,以 PAL 制标准为例, $T1=1000/25=40\text{ms}$ 。

编码时间 T2:在实时编码的情况下,编码一帧图像的时间为 $T2=40\text{ms}$ 。

码流发送时间 T3:对延迟影响比较大的部分是发送缓冲区,设置缓冲区是为了解决变码率码流的传输和存储问题。缓冲区越大,越能承受码率的波动,但造成的延迟也越大,反之亦然。以 I 帧编码为例,产生的码流可能会充满整个缓冲

区的 3/4,所以发送缓冲区延迟一般为 $T3=3 \times 40=120\text{ms}$,在缓冲区被充满的极端情况下可能会导致 $140\text{ms} \sim 160\text{ms}$ 的延迟。

码流接收时间 T4:解压缩图第一帧图像并进行图像数据格式转换的时间,这里记为 $T4=25\text{ms}$ 。

显示时间 T5:解压缩后送入显示设备,显示一帧图像的时间为 $T5=40\text{ms}$ 。

将上述时间进行累计,可以得到编解码总延迟为 $T=T1+T2+T3+T4+T5=265\text{ms}$ 。

在上述延迟中,I 帧作为关键帧类型,其编解码延迟是最大的,同时对整个编解码延迟的贡献也是最大的,降低 I 帧编码过程中的延迟,降低其产生码率将对延迟产生重要的影响。

3 基于条带的采集低延迟划分结构

从第 2 节的分析可知,编解码的延迟是比较大的,对于实时直播、移动终端等应用来说,是较难满足系统延迟要求的。为了减少现有编解码系统的延迟,本文中提出一种基于条带(Slice)的降低采集延迟的编码结构,如图 2 所示。在该结构中,在采集完成一个条带后将直接进行编码。如果输入的图像类型为 I 帧,将被化整为零,编码的单位由原来采集完一帧进行编码变成一个条带,每个条带的大小一般以宏块行或者编码单元为基本单位。

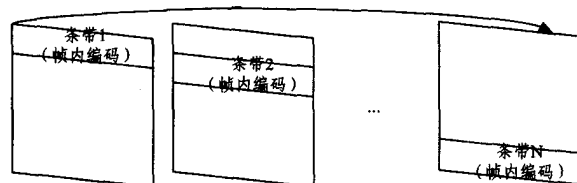


图 2 基于条带的采集低延迟编码划分方式

$$N_{\text{slice_MB}} = N_{\text{frame}} / N_{\text{MB}}$$

$$N_{\text{slice_CU}} = N_{\text{frame}} / N_{\text{CU}}$$

式中, N_{frame} 为一帧图像的扫描行数, N_{MB} 表示以宏块为基本单位所包含的行数,一般情况下, $N_{\text{MB}}=16$; $N_{\text{slice_MB}}$ 为以宏块为单位所划分出的 SLICE 个数。同理, N_{CU} 表示以编码单元 CU 或 LCU 为基本单位所包含的行数,一般情况下, $N_{\text{CU}}=32$ 或 64 ; $N_{\text{slice_CU}}$ 为以编码单元为单位所划分出的 SLICE 个数。

对于 P 帧或 B 帧图像,意味着,P/B 图像帧中将有一个条带采用 Intra 编码方式,其余的条带仍采用原有 P 帧的 Inter 编码(帧间编码)。这样就不存在原来 I 帧和 P 帧之间的较大码流差别,使每帧图像的压缩码流量相对平稳,可以设定较小的发送缓冲区。此外,可以在采集完一个条带后就进行编码,以降低原来的采集延迟。

这里需要特别区别两种类型的视频,一种是隔行扫描方式,另一种为逐行扫描方式。下面将对这两种情况进行分析。

(1)隔行扫描:25 帧/秒,分辨率为 720×576 的隔行视频序列, $N_{\text{frame}}=576$,以宏块行为单位进行条带划分, N_{MB} 包含纯奇数行或者纯偶数行, $N_{\text{MB}}=16$,一帧图像共有 $N_{\text{slice_MB}} = N_{\text{frame}} / N_{\text{MB}} = 36$ 条带,则基于条带的系统延迟为:

采集一个条带就可进行编码,采集所需时间为: $T1=40\text{ms}/36=1.11\text{ms}$ 。

在基于条带(Slice)的编解码器中,一帧图像的压缩码流量相对平稳,可以设定缓冲区大小为 1.25~1.5 帧,发送缓冲区延迟一般为 $T3=40\text{ms}$;在缓冲区被充满的极端的情况下,可能会导致 $50\text{ms} \sim 60\text{ms}$ 的延迟。

T2、T4、T5 的计算方法与基于帧编码方式类似,这样编解码过程的总延迟为 $T = T1 + T2 + T3 + T4 + T5 = 146.11\text{ms}$,这比基于帧编解码方式减少了 100ms 以上。

(2)逐行扫描:条带可以是逐行划分,也可以是隔行划分。

逐行扫描中的逐行划分方式,以宏块行为单位进行条带划分, $N_{MB} = 16$, N_{MB} 包含奇数行偶数行相间的行,即从第 1—16 行作为第一个条带,依次类推,可以将一帧图像划分成若干个条带。

但考虑到本文后续的编码结构,将采用隔行划分的形式。 N_{MB} 只包含奇数行或偶数行,即对于第一个 SLICE 将是由第 1 行、第 3 行、第 5 行等奇数行构成,第二个 SLICE 将是由第 2 行、第 4 行、第 6 行等偶数行构成。

以上逐行扫描方式中的两种条带划分方式,延迟相差不大,第一种方式与隔行扫描方式相同;第二种方式比第一种方式在采集时多需要一个条带,采集所需时间为: $T1 = (40\text{ms}/36) \times 2 = 2.22\text{ms}$ 。

注:本文后续内容是基于隔行扫描的条带划分方式展开的。

4 基于条带的采集低延迟划分结构

编码过程中 I 帧压缩码流的大小对延迟、视频序列的稳定性都有重要影响。I 帧一般采用预测的方式进行编码,即通过已编码图像预测待编码图像的过程。在基于块的预测过程中,先将图像分为大小相等的块,已编码块作为基本预测单元,预测待编码块。

4.1 标准的帧内预测模式

在标准的编码过程中,采用了多模式的预测技术,从多个角度进行预测。H. 264 标准中采用帧内预测算法^[4],该算法以 16×16 的宏块及 4×4 的块为基本预测单元,对块内各点在 9 个方向上进行预测。以 4×4 块为例,如图 3(a)所示, $a, b, \dots, p$ 为当前待预测块,周围 17 个点 $Q, I, \dots, P$ 为已编码点。对 $a-p$,逐点沿图 3(b)所示的 0,1, ..., 8 及方向 2(DC 预测)共 9 个方向取 P-Q-H 中的点,用适当的预测公式计算预测值,并与原始采样值做差分,差值最小的模式为最终预测模式,最后对预测残差系数进行 DCT 编码。

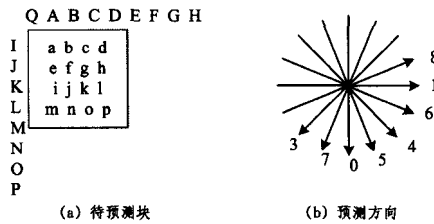


图3 JVT 中的空间相邻点预测

在最新的 HEVC 标准中,引入了 CU(编码单元)、PU(预测单元)、TU(变换单元) 3 个新的概念。编码单元类似于 H. 264/AVC 中的宏块概念,其大小最大可以为 64×64 ;预测单元是进行预测的基本单元;变换单元是进行变换和量化的基本单元。3 个单元的分离,使得变换、预测和编码各个处理环节更加灵活,也更符合视频图像的纹理特征。预测单元的大小可以为 $4 \times 4, 8 \times 8, 16 \times 16, 32 \times 32, 64 \times 64$,块的大小不同,预测的模式也不相同,分别对应 17,34,34,34 和 3 种模式可选。如图 4 所示,统一的帧内预测角度为: $+/- [0, 2, 5, 9, 13, 17, 21, 26, 32]/32$,模式的选择标准同 JVT 标准一样。

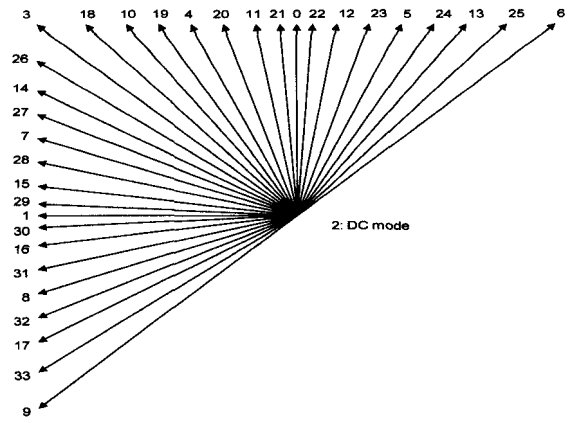


图4 HEVC 标准中的帧内预测模式

4.2 本文预测模式与标准模式预测距离分析

在进行预测时,当前像素与其参考像素的相关性越强,预测残差(SAD)也就越小,预测效果越好。

标准中的预测方法,可以说是穷举式的,这种方式可以用图 5(a)来表示;为了克服该方法的缺点,本文提出了一种如图 5(b)所示的基于最近像素点的分层预测模式,首先利用黑色像素预测图中灰色点,其余像素由编码后重建的灰色点内进行第二级预测。

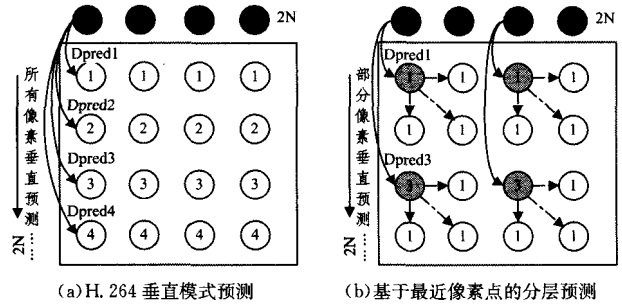


图5 不同预测方式下像素的预测距离

本文中引入平均预测距离这个概念(即所有当前像素与其参考像素距离的平均值,用 D_{pred} 表示),并用平均距离对标准方法与本文方法的预测距离进行分析。

设当前待编码的基本块大小为 $2N \times 2N$ (通常 $N=2, 4, 8, 16, \dots$), $D_{i,j}$ 表示坐标为 (i, j) 的像素与其参考像素之间的距离,该数据块的平均预测距离 D_{pred} 可按如下步骤求得。

首先计算单列像素的预测距离 D_{col} :

$$D_{col} = D_1 + D_2 + \dots + D_{2N} = \sum_{i=1}^{2N} D_i$$

数据块的总预测距离 D_{total} 为各列预测距离之和:

$$D_{total} = D_{col_1} + D_{col_2} + \dots + D_{col_{2N}}$$

$$= \sum_{j=1}^{2N} (D_{col_j}) = \sum_{j=1}^{2N} \sum_{i=1}^{2N} D_{i,j} \quad (1)$$

(1)图 5(a)为 H. 264 的帧内预测垂直模式,该模式下同一行所有像素的预测距离相等,即 $D_i = i$ 。图中当前像素点上标注的数字即是该点的预测距离,因此该数据块所有像素的平均预测距离 D_{pred_A} 为:

$$D_{pred_A} = D_{total_A} / (2N \times 2N) = \frac{1}{4N^2} \sum_{j=1}^{2N} \sum_{i=1}^{2N} D_{i,j}$$

$$= \frac{1}{4N^2} [2N \times (1 + 2 + \dots + 2N)]$$

$$= (4N^3 + 2N^2) / 4N^2 \quad (2)$$

(2)图 5(b)中箭头标记了每个像素的参考像素点,与图

5(a)相同,其一级像素点的预测距离由其与参考像素的距离决定;由于第二级预测的所有点均由相邻点预测得出,因此预测距离均为1。

本文方法的平均预测距离计算如下:首先,第一级预测距离 D_{level_1} 为第一级中 N 列像素点预测距离总和:

$$D_{level_1} = D_{col_1} + D_{col_3} + \dots + D_{col_2N-1} = \sum_{j=1}^N D_{col_j}$$

由于第一级预测中采取的是隔行隔列预测,因此其中每一列像素点的预测距离 $D_{col_j} = 1 + 3 + \dots + 2N - 1$,代入上式可以得到:

$$D_{level_1} = \sum_{j=1}^N D_{col_j} = N \times (1 + 3 + 5 + \dots + 2N - 1) = N^3$$

第二级预测距离 D_{level_2} 为第一级预测得到的所有点(共 $2N$ 个)各自分别对应的3个二级预测点的预测距离之和,即 $D_{level_2} = 2N \times 3$ 。

因此,采用图5(b)所示方法整个预测像素块的预测距离 D_{total_B} 为:

$$D_{total_B} = D_{level_1} + D_{level_2} = N^3 + 6N$$

平均预测距离 D_{pred_B} 为:

$$D_{pred_B} = D_{total_B} / (2N \times 2N) = (N^3 + 6N) / 4N^2 \quad (3)$$

(3)令 λ 为 D_{pred_A} 、 D_{pred_B} 两者比值,可以得出:

$$\lambda = D_{pred_A} / D_{pred_B} = \frac{4N^2 + 2N}{N^2 + 6}$$

上式表明, D_{pred_A} 、 D_{pred_B} 两者有数倍差距, N 值越大差距越大。

对比式(2)和式(3)可以看出,分层预测方法大幅减小了平均预测距离。对于 4×4 大小的预测块($N=2$),可以计算出 $D_{pred_A}=2.5$, $D_{pred_B}=1.25$,两者相差一倍;对于 16×16 这样较大的预测块($N=8$),两种方法的平均预测距离分别为8.5和2.2,与H.264方法相比,基于最近像素点的分层预测方法平均预测距离下降近75%,优势更加明显。本节仅以标准中的垂直预测模式为例进行了理论分析,同理,该方法可适用于其他预测模式。

图6为逐行序列被分解为4张CIF格式子帧后的情况。分解方式为:将原图 2×2 单位中的4个像素点分别划分到4个子帧中,形成水平和垂直分辨率减半的基本子帧、水平子帧、垂直子帧和对角子帧。显然,4个子帧都各自不同程度地包含了原图信息并且相互之间具有强相关性,该特性表明利用子帧间预测可更有效地去除冗余。

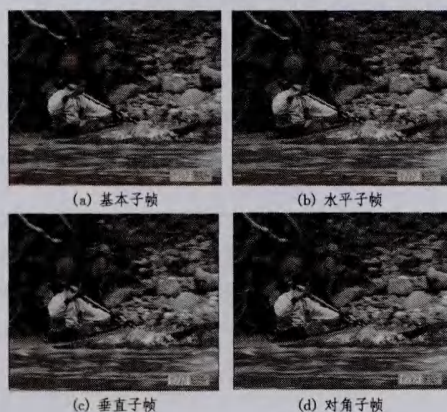


图6 Boat序列某1帧的空域分解

4.3 基于条带的预测编码

在进行预测时,最需要考虑的是,低分辨率子图恢复成高

分辨率原图的过程。在本文中,低分辨率子图的大小是以条带的 $1/4$ 划分的,高分辨率原图为一个条带的大小。

事实上,这可以看作图像放大的过程,进行图像放大时,通常的做法是根据已知采样点预测逼近未采样点。插值是函数逼近的一种重要方法,在图像处理中常用来放大图像。根据采样定理,当采样间隔足够小,即 $\tau \leq \frac{1}{2s_0}$ (τ :采样间隔, s_0 :

采样函数的截止频率)时,采样函数可以完全无失真地从采样数据中恢复。但实际采样频率通常达不到以上标准,因此通过插值不可能完全无损地得到未采样点。插值函数与实际采样函数之差称为估值误差。估值误差是反映恢复图像失真度的一个重要参数,估值误差越小,恢复图像失真度越小。估值误差大小由采样图像分辨率和插值函数核决定。通常情况下,采用同一插值函数,图像采样分辨率越高,采样间距越小,插值效果越精确,估值误差越小。在采样分辨率确定的条件下,可通过改进插值函数,使其进一步逼近采样函数,从而减小估值误差。图像处理中常用的插值算法有:邻近像素复制、线性插值、立方卷积等,一些精度更高、自适应的插值算法^[4,5]在医学、遥感等领域应用也非常普遍。这些算法中,双线性插值因计算复杂度低、插值精度相对较高而被广泛采用。

设: (x, y) 为 S 中的点; $r(x, y)$ 为 (x, y) 点的像素值; Π_1 为 S_1 中点的集合; Π_2 为 S_2 中点的集合; $f(\Pi_1)$ 为插值函数。 $\forall (x, y) \in \Pi_2$, 通过插值函数计算其近似值 $r'(x, y) = f(\Pi_1)$ 。设 $\Delta(x, y) = r'(x, y) - r(x, y)$ 为估值误差。当插值函数足够精确时,有: $\Delta(x, y) \rightarrow 0$ 。因此,以 $r'(x, y)$ 作为 $r(x, y)$ 的预测点将取得很好的预测效果。因而,考虑到预测的精度和复杂度,本文选用双线性插值来构造对应位置的预测点。

4.4 性能分析

除了本文第3节的延迟分析,还采用本文方法对H.264标准baseline profile进行实现、测试并与标准算法结果相比较。对于恢复图像质量,本文仍采用传统的峰值信噪比(PSNR)作为衡量指标。通过调整量化参数,分别测得各序列在标准算法及本文算法在相同PSNR(dB)值时对应的编码位数(kb/frame),纵坐标表示恢复图像的PSNR值。

针对测试序列,本文从Cif序列中选择了运动剧烈的coastguard、相对静止的mother、纹理复杂的flower,测试结果如表1所列。

表1 本文算法及标准算法测试结果比较

序列	PSNR (dB)	平均每帧编码位数 (kb/frame)		减少位数 (kb/frame)	下降百分比
		本文算法	H.264		
Coastguard	28	24	29.5	5.5	17.7%
	31	52	59	7	11.3%
	34	96	105	9	8.3%
Mother	33	16	17.5	1.5	8.3%
	35	21	23.8	2.8	11.2%
	38	41	44	3	6.8%
Flower	26	52	68.2	16.2	22.8%
	30	94	119	25	20.4%
	34	147	185	38	20.2%

从表1的结果可以看出,采用本文算法后,图像PSNR值一定时,平均每帧编码位数较标准算法减少1.5~38kb,约占码流量的6%~23%。标准算法在带宽较低条件下,对纹理复杂的图像进行帧内编码时,编码位数增长十分迅速,常导致图像质量急剧下降或出现连续跳帧,采用本文算法后编码位

数明显减少,以上现象得到缓解,如表1中 Flower 在低带宽下码率下降尤为明显。

因此,本文的方法是在不降低图像质量的前提下,降低码流,进而降低延迟。

结束语 本文首先对现有编码标准的固有延迟进行了分析,以标准清晰度的视频为例,编解码延迟要在 260ms 以上,无法满足低延迟系统的需求。因而,本文从采集与编码两个环节来降低延迟,首先,采集端将每帧图像以条带为单位进行采集,条带的大小与视频分辨率和延迟要求相关;然后,以条带为单位进行编码,为了提高预测精度,进一步降低延迟,将每个条带划分成 4 个子图,以其中一个子图为预测单元进行帧内预测,这样可以有效提升预测的精度。本文提出的基于条带的低延迟编解码结构能有效降低编解码系统的固有延迟,对于标准清晰度视频,可以降低到 150ms。

参考文献

- [1] 郑翔,叶志远,周秉锋. JVT 草案中的核心技术综述[J]. 软件学报,2004,15(1):58-68
- [2] 毕厚杰. 新一代视频压缩编码标准-H. 264/AVC[M]. 北京:人民邮电出版社,2005:78-89
- [3] Wiegand T, Sullivan G J, Bjontegaard G, et al. 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] Sullivan G J, Ohm J-R, Han W-J, et al. Overview of the High Efficiency Video Coding (HEVC) Standard[J]. IEEE Transac-

tions on Circuits and Systems for Video Technology, 2012,22(12):1649-1668

- [5] Ohm J-R, Sullivan G J, Schwarz H, et al. Comparison of the Coding Efficiency of Video Coding Standards-Including High Efficiency Video Coding (HEVC)[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012,22(12):1669-1684
- [6] Bossen F, Bross B, Sühning K, et al. HEVC Complexity and Implementation Analysis[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012,22(12):1685-1672
- [7] Lin P X, Rahardja S, et al. Fast Intra Mode Decision Algorithm for H. 264/AVC Video Coding [J]. International Journal of Computer Science and Network Security, 2007,7(1):356-361
- [8] Li Bo, Li Wei, Tu Ya-ming. A fast block matching algorithm using smooth motion vector field adaptive search technique[J]. Journal of Computer Science and Technology, 2003,18(1):14-21
- [9] Yang Zhi, Cai Hua, Li Jiang. A framework for fine-granular computational-complexity scalable motion estimation[C]//IEEE International symposium on Circuits and Systems (ISCAS). 2005:5473-5476
- [10] Yu Dan-chu, Huang Sung-cheng. Study of Reprojection Methods in Terms of Their Resolution Loss and Sampling Errors[J]. IEEE Transaction on Nuclear Science, 1993,40(4):1174-1178
- [11] Rodrigues L, Leandro Borges D, Marcos Goncalves L. A Locally Adaptive Edge-preserving Algorithm for Image Interpolation[C]//XV Brazilian Symposium on Computer Graphics and Image Processing, 2002:300-305

(上接第 156 页)

//在应用的 dwr. xml 中自动增加的对类的定义

```
<dwr>
  <allow>
    <create creator="new" javascript="ScheduleQuery">
      <param name="class"
        value="cn.edu.fudan.tr23.ScheduleQuery"/>
    </create>
  </allow>
</dwr>
//页面构件 js 中插入 DWR 调用,调用业务构件中
//ScheduleQuery 类的 getUserCourse 方法,取得返回值后调用
//getUserCourse_callBack 回调函数
function getUserCourse(username){
  ScheduleQuery. getUserCourse (username, getUserCourse _callBack);
}
```

图 10 采用基于 DWR 模式下的新增代码

5.3 分析讨论

应用该技术后,Web 应用构件的组装不再需要关注组装细节,一切连接代码的生成、编译等工作都交由组装引擎来完成,保证了实用性和简便性;另外,亦满足了 Web 应用中异步通信、局部刷新的需求。但由于该方法在设计时定义的限制较多,导致适用性较差,例如:当前该技术仅支持页面构件、业务构件和 Web Service 构件的组装,难以满足未来所需,限制了 Web 应用构件开发者的创造性;在本技术实现中,也未对页面构件中亦存在服务接口、提供功能服务给其他页面构件的可能性做出考虑。

结束语 本文基于软件复用的思想为 Web 应用的高效

开发提供了一种途径。我们首先对 Web 应用的构件类型以及这些不同类型构件之间的组合方式进行了分析,随后提出了一套基于 AJAX 的 Web 应用构件组装技术。该技术主要包括两种实现模式,1)采用 jQuery 调用 Servlet 的模式,2)采用基于 DWR 技术的模式。使用这两种模式都能够实现前端页面构件与后端业务构件或 Web Service 构件的交互。基于该技术,我们开发了采用不同实现模式的组装工具,并使用一个选课网站系统作为实例描述了组装的过程与结果。由于本文所提出的技术仍然具有局限性,因此,我们计划在以后的工作中涵盖更多类型的构件实体及其组装类型。同时,逐步改进工具使其更具有可用性。

参考文献

- [1] Jacobson I, Griss M, Jonsson P. Software reuse: architecture, process and organization for business success[M]. ACM Press/Addison-Wesley Publishing Co., 1997
- [2] Fuqing Y, Hong M, Keqin L. Software Reuse and Software Component Technology [J]. Acta Electronica Sinica, 1999,27(2)
- [3] George T H, William T C. Component-based software engineering: putting the pieces together[M]. Reading: Addison-Wesley, 2001
- [4] 陈霄,吴毅坚,彭鑫,等. 采用构件组装技术协同开发 Web 应用的方法[J]. 计算机科学与探索, 2013,7(2):114-125
- [5] Direct Web Remoting [OL]. <http://directwebremoting.org/dwr/index.html>. Accessed:5 June 2013
- [6] Serrano N, Aroztegi J P. Ajax frameworks in interactive Web apps[J]. Software, IEEE, 2007,24(5):12-14
- [7] 张栗,张凤元,危胜军. 基于 DWR 框架的 Web 应用的设计与实现[J]. 计算机技术与发展, 2008,18(8):84-87