

基于宏块类型的检错方法

方勇^{①②*}, 吴成柯^①, 郑济昌^②

① 西安电子科技大学 ISN 国家重点实验室, 西安 710071;

② 韩国汉阳大学电子与通信工程系, 首尔 133791

* E-mail: yfang79@gmail.com

收稿日期: 2004-12-03; 接受日期: 2007-08-24

国家自然科学基金(批准号: 60532060)和韩国 BK21 资助项目

摘要 提出一种基于宏块类型信息的视频传输差错检测方法。对已解码帧间宏块, 计算接收到的残差图像的绝对值之和, 并通过运动补偿参考块的纹理复杂度来估计当前宏块的帧内纹理复杂度。比较接收到的帧间残差与帧内纹理复杂度, 当帧间残差比帧内纹理复杂度大于某阈值时, 即认定该宏块传输有误, 并对该宏块进行错误隐藏。对已解码帧内宏块, 通过检测当前宏块与周围宏块的连接平滑程度来发现错误宏块。实验结果表明新算法能有效地消除出现较大错误的宏块。结合新算法与错误隐藏, 解码端的重建图像质量提高了 0.5~1 dB 左右。

关键词

MPEG-4
视频传输
差错检测
错误隐藏

当前主要的视频压缩标准(如 MPEG-x 和 H.26x)都是基于运动估计(ME)、DCT 变换与可变长编码(VLC)结合的框架之上的。在提高编码效率的同时, 这一框架使得视频码流对传输错误很敏感。一方面, VLC 的使用导致了错误的空间传播; 另一方面, 运动补偿的使用进一步导致了错误的时间传播。尽管信道编码技术的应用大大提高了码流的抗误码性能, 但是在视频码流中的随机比特错误仍是不可避免的, 因而编码层的抗误码技术(如容错编码及错误隐藏等)是必要的。容错编码是指编码端通过增加码流的冗余来增强码流的抗误码性能, 例如在码流中插入同步头^[1], 使用反向可解变长码(RVLC)^[2], 部分反向可解码流^[3]及容错熵编码(EREC)^[4]等。错误隐藏是指在已出错的情形下, 解码端通过各种途径来掩盖错误, 减小错误对图像质量的影响。

由于 VLC 的应用, 当解码器发现码流有错时, 通常不能确定错误的位置, 因为误码可能发生在此之前的任意地方, 因而在进行错误隐藏之前, 错误的定位是很重要的, 这就是差错检测。差错检测可以分为两步: 基于语法的检测和基于内容的检测。前者用以发现码流中有无出错; 若发现码流中有错, 后者可用以推测出错误出现的地方。基于内容的检测可以在频域进行,

也可在像素域进行。在文献[5, 6]中, 相邻行像素的差值被用来检测 PCM 和 DPCM 编码的图像的传输错误。在文献[7]中, Mitchell 和 Tabatabai 提出的算法可以检测出一个块中的单个 DCT 系数的错误。然而该算法是基于 DCT 变换加定长码(FLC)结构的。在文献[8]中, Lam 和 Reibman 提出了一种频域的检测方法。以上这些算法对于 ME, DCT 加 VLC 的编码框架并不适用。此外它们运算较复杂并没有利用到视频数据的帧间相关性。

在近来的 H.263, MPEG-2 和 MPEG-4 等标准中, 差错检测通常在传输层进行^[9, 10]。例如将视频流打包在 RTP 分组中, 给每个分组标上序号。这样在接收端可以提取 RTP 分组的序号来检测视频流传输是否出错。然而这些处理方法需要底层的支持, 在某些情形下并不适用。

本文提出了一种像素域的基于内容的差错检测方法。该方法计算已解码帧间宏块的残差的绝对值之和; 并用运动补偿参考块的纹理复杂度来估计当前宏块的纹理复杂度。通过比较帧间残差与帧内纹理复杂度来发现错误解码的宏块。

1 主要的差错检测方法

在解码过程中, 解码器通过分析视频码流的语法结构来检测码流有无出错。当出现以下几种情形时, 解码器将断定码流有错:

- 1) 收到非法的 VLC 码;
- 2) 在一个块中解出了多于 64 个 DCT 系数;
- 3) 对帧间宏块, 运动矢量超出预定的范围。

当发现码流出错, 解码器将不再往下解, 而是直接寻找下一个同步头。此时在该视频包里的宏块可以分为 3 类: 正确解码的宏块、错误解码的宏块和不可解码的宏块。对于出错的视频包的处理方法有 3 种: 对该视频包中所有宏块均进行错误隐藏(无论其中的宏块可解与不可解); 仅对不可解码宏块进行错误隐藏(无论已解码宏块有没有被正确解码); 将正确解码宏块与错误解码宏块区分开来, 对错误解码宏块及不可解宏块进行错误隐藏。显然第 3 种方法是最好的选择。

错误可能出在码流中的任何部分, 例如宏块类型、运动矢量及变换系数等。由于数据分组的使用, 头信息及运动矢量通常被放在与变换系数不同的包里, 使用更高保护程度的前向纠错编码予以保护, 故在本文中我们仅考虑变换系数的错误检测, 并假定运动矢量及头信息能被正确无误地接收。对于宏块类型及运动矢量错误的检测与隐藏, 可以参考文献[11, 12]。

一般来说, 交流分量的错误较直流分量的错误难检测, 因此我们的方法仅限于交流分量的错误检测。此外由于在视频码流中大部分为帧间宏块, 因而我们的方法着重于帧间宏块的错误检测。

2 基于宏块类型的差错检测方法

2.1 编码过程的简要回顾

原始图像(P 帧)被分割为 16×16 的宏块。首先进行运动估计寻找每个宏块的最佳预测块, 使当前宏块与参考块对应像素的差值的绝对值之和(SAD)最小。

$$SAD(x, y) = \sum_{i=0, j=0}^{15, 15} |orig(i, j) - ref(i+x, j+y)|, \quad (1)$$

其中 $orig(i, j)$ 指当前宏块, $ref(i+x, j+y)$ 指参考块。使得 $SAD(x, y)$ 最小的 (x, y) 即为当前宏块的运动矢量。记 $SAD(x, y)$ 的最小值为 SAD_{orig} 。然后, 计算宏块的直流分量

$$MB_DC = \sum_{i=0, j=0}^{15, 15} orig(i, j) / 256. \quad (2)$$

接下来计算当前宏块的纹理复杂度

$$MB_Comp = \sum_{i=0, j=0}^{15, 15} |orig(i, j) - MB_DC|. \quad (3)$$

最后比较 MB_Comp 与 SAD_{orig} 。若满足

$$MB_Comp < (SAD_{orig} - C), \quad (4)$$

(其中 C 为一常数, MPEG-4 校验模型中设为 512), 则帧内(INTRA)编码类型被选择, 否则帧间(INTER)类型被选择。若宏块为帧间编码, 则对当前宏块与最佳匹配参考块的残差进行变换、量化与编码。记编码之后的残差为 $diff_{enc}(i, j)$ 。我们计算 $diff_{enc}(i, j)$ 的绝对值之和 SAD_{enc} ,

$$SAD_{enc} = \sum_{i=0, j=0}^{15, 15} |diff_{enc}(i, j)|. \quad (5)$$

由于是有损压缩, 故 $SAD_{enc} < SAD_{orig}$, 并且 SAD_{enc} 随着量化步长 Qp 增大而减小。图 1 显示了对 QCIF 格式 Foreman 序列第 2 帧编码时, 量化步长对 $(SAD_{orig} - SAD_{enc})$ 的影响(图 2~4 与此类似)。图 1 中横坐标为宏块号, 纵坐标为每一宏块的 $(SAD_{orig} - SAD_{enc})$ 。从图中我们清楚地看到随着量化步长增大, $(SAD_{orig} - SAD_{enc})$ 显著增大。

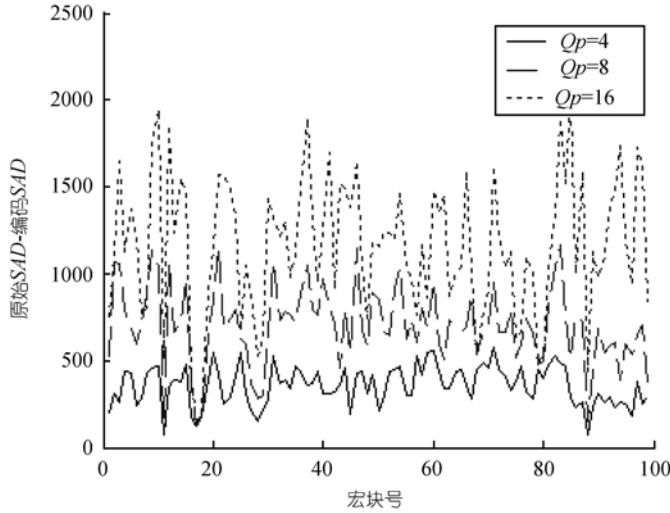


图 1 量化步长对 SAD_{enc} 的影响

2.2 宏块的错误检测

在解码端, 假定当前宏块能被解码, 且该宏块为帧间类型。对之进行 IDCT 变换得到残差

$diff_{dec}(i, j) = diff_{enc}(i, j) + err(i, j)$, 其中 $err(i, j)$ 为传输误差). 我们计算 $diff_{dec}(i, j)$ 的绝对值之和 SAD_{dec} :

$$SAD_{dec} = \sum_{i=0, j=0}^{15, 15} |diff_{dec}(i, j)|. \quad (6)$$

若传输无误, 则有 $SAD_{dec} = SAD_{enc}$. 然而由于存在传输错误, SAD_{dec} 与 SAD_{enc} 可能并不相等. SAD_{dec} 与 SAD_{enc} 的差异是由 $err(i, j)$ 来决定的:

1) 若宏块残差的高频交流分量丢失, 则 $err(i, j)$ 将使得 $diff_{dec}(i, j)$ 更平坦, 此时 $SAD_{dec} < SAD_{enc}$;

2) 若解出了多余的高频交流分量, 则 $err(i, j)$ 将使得 $diff_{dec}(i, j)$ 变得起伏更大, 此时 $SAD_{dec} > SAD_{enc}$;

以上两种情形对重建图像质量的影响是不一样的. 第 1 种情形事实上对图像质量的影响是很小的, 甚至不可见, 因为人眼对高频分量较不敏感, 并且在此种情形下, 错误隐藏并不能起到很大的作用. 而第 2 种情形对重建图像质量的影响则可能很恶劣, 尤其是出现了多余的较大的高频分量时. 因而以下仅考虑第 2 种情形.

现在我们回忆宏块类型的判决过程, 当(4)式成立时, 帧内编码类型将被选择. 在此很自然想到将 SAD_{dec} 代入(4)式中, 当传输无误时应该有

$$SAD_{dec} < (MB_Comp + C). \quad (7)$$

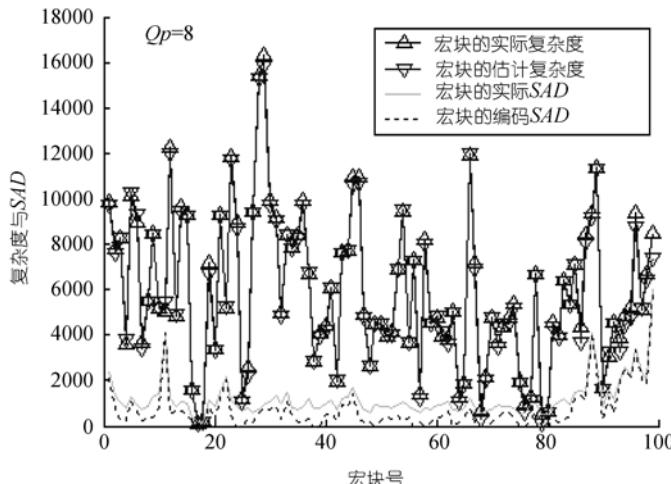


图 2 MB_Comp_{ref} 与 MB_Comp 的比较

当传输有误使得码流中出现了多余的高频分量时, SAD_{dec} 将增大, 从而可能使得(7)式不成立, 此时即可断定该宏块出错(因为在帧间残差较大的情形下, 一个明智的编码器将对该宏块进行帧内编码). 至于码流中虽然有错, 并且也多出了不该有的高频分量, 但是依然满足(7)式的情形, 此时错误并不严重, 可以忽略, 故以下不予进一步讨论.

然而, 由于解码端并不知道原始的宏块, 因而也无从计算 MB_Comp . 为解决这一问题, 我们使用当前宏块的运动补偿参考块 $MC_Ref(i, j)$ 来预测 MB_Comp . 记 $MC_Ref(i, j)$ 的帧内纹

理复杂度为 MB_Comp_{ref} . 仿真结果表明, MB_Comp_{ref} 是 MB_Comp 的极好近似. 从图 2 中可以清楚看到, MB_Comp_{ref} 与 MB_Comp 相当吻合. 与 $(SAD_{orig} - SAD_{enc})$ 不同, $(MB_Comp - MB_Comp_{ref})$ 大体分布在 0 附近, 与 0 的偏差随量化步长 Qp 的增大而略有增大(图 3).

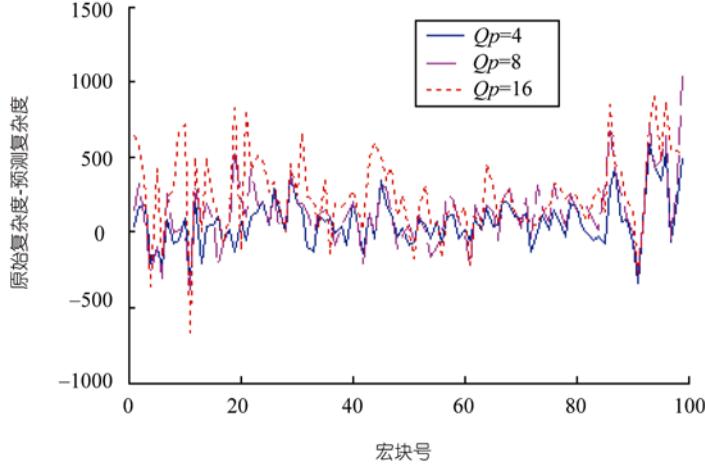


图 3 量化步长对 MB_Comp_{ref} 的影响

将 MB_Comp_{ref} 代入(7)式, 即为

$$SAD_{dec} < (MB_Comp_{ref} + C'), \quad (8)$$

其中 C' 为一常数. 当(8)式不满足时, 即认定该宏块有错. (8)式的合理性. 是基于以下的假定: 当 $SAD_{orig} < (MB_Comp + C)$ 的前提下, 则有 $SAD_{enc} < (MB_Comp_{ref} + C')$. 为验证这一假定, 同时为确定 C' , 我们进行了仿真. 从图 4 中我们看到 $(MB_Comp_{ref} - SAD_{enc})$ 与 $(MB_Comp - SAD_{orig})$ 相当吻合, 这为前述假定提供了现实可行性. 同时据此, 设定 $C' = C$.

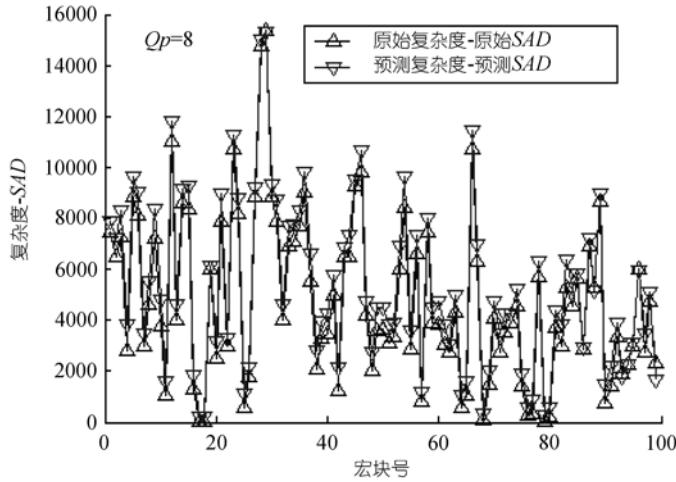


图 4 复杂度与 SAD 的预测差值与实际差值的比较

值得一提的是, 对出错的视频包采用的 3 种处理方法(扔掉所有宏块; 保留所有可解码宏块; 仅保留正确解码宏块)均可看作(8)式的特例。当令 $C' = -\infty$ 时, 此时(8)式恒不成立, 该视频包中所有宏块将视为有错, 这对应的是扔掉所有宏块的情形。反之, 当令 $C' = \infty$ 时, 此时(8)式恒成立, 该视频包中所有可解码宏块将视为正确, 这对应的是保留所有可解码宏块的情形。

2.3 算法的应用

我们的算法的一个先决条件是要正确收到 $MC_Ref(i, j)$ 。这要求在此之前的所有帧(同一帧组之内)的当前宏块位置附近的像素都被正确接收到。由于错误的随机性, 在连续帧里相邻位置出现错误的可能性并不大, 因而这一条件可以认为是基本满足的。对于帧内宏块, 由于缺乏参考块, 在我们的仿真中比较当前宏块与周围宏块的直流分量, 若存在较大差异, 则认为该宏块有错。此外, 使用边界像素来测试当前宏块与周围宏块的连接平滑度。若当前宏块与相邻宏块的边界像素存在较大差异, 则该宏块也被认为有错。对出错的帧内宏块, 使用帧内插值的方法进行错误隐藏。对于宏块类型及运动矢量等信息, 由于所占带宽较低, 可采用数据分组加不等错误保护(UEP)来传输。

3 仿真结果及讨论

通常错误检测需要和错误隐藏结合起来使用, 才能在实际应用中收到好的效果。人们已经提出了多种错误隐藏的方法, 例如, Wang 等^[13]提出的大平滑恢复(MSR)方法以及 Sun 等人^[14]提出的基于凸集投影的空间插值的方法等。以上的方法运算较为复杂, 而在我们的仿真中使用的错误隐藏方法相对简单。对帧内宏块, 使用加权像素平均的帧内插值的方法进行错误隐藏, 其权值与被预测的像素和相邻宏块像素的距离成反比; 对帧间宏块, 运动补偿参考块被直接复制取代当前宏块。

我们使用 MPEG-4 的编码程序进行仿真。帧组大小为 50, 即每 50 帧插入 1 个 I 帧。每一行的 11 个宏块构成一个视频包。为了更好地演示我们提出的方法的效果, I 帧和各种头信息及运动矢量置于与 P 帧 DCT 系数不同的包中进行传输。我们使用 QCIF 格式的 Foreman, Coastguard 和 Container 3 组图像序列进行仿真。为验证我们的算法对信道特征变化的适应能力, 仿真在 6 种不同的传输误码率(如表 1 所示)下进行。

表 1 3 种方法的客观对比

误码率	Foreman			Coastguard			Container		
	NED	NER	ED	NED	NER	ED	NED	NER	ED
0	34.92			33.76			34.66		
1.35×10^{-3}	26.23	26.01	26.71	24.34	23.59	24.63	29.51	29.05	29.68
1.24×10^{-3}	26.56	26.32	26.99	24.52	23.80	24.81	29.72	29.18	29.85
9.21×10^{-4}	27.47	27.18	27.79	25.11	24.59	25.46	30.36	29.79	30.52
8.77×10^{-5}	33.13	33.00	33.21	30.44	30.11	30.67	33.80	34.01	33.89
1.12×10^{-4}	32.97	32.87	33.10	29.96	29.76	30.19	33.67	33.38	33.71
8.99×10^{-4}	27.52	27.20	27.81	25.14	24.66	25.52	30.40	30.40	30.61
平均值	28.98	28.76	29.27	26.59	26.09	26.88	31.24	30.97	31.38

当视频包出错时, 解码端尝试了 3 种不同的处理方法。第 1 种方法, 扔掉当前视频包中所有宏块, 并对当前视频包中所有宏块进行错误隐藏, 该方法以 NED 标识。第 2 种方法, 保留当前视频包中所有已解码宏块, 对当前视频包中未解码宏块进行错误隐藏, 该方法以 NER 标识。第 3 种方法, 使用我们提出的方法找出错误宏块, 之后保留已正确解码的宏块, 对解码错误及未解码的宏块进行错误隐藏, 该方法以 ED 标识。

仿真的客观结果包含在表 1 中。表 1 中的结果以灰度图像的峰值信噪比(PSNR)来衡量, 单位为分贝(dB)。从表 1 中可以清楚的看到, 应用我们的错误检测方法, 图像质量有了显著的提高。在 3 种方法中, 方法 2 的效果最差, 这说明在很多情形下错误解码宏块对图像质量的影响相当恶劣, 有甚于未解码的宏块。方法 3 的图像质量较方法 2 平均提高了 0.5~1 dB 左右, 较方法 1 平均提高了 0.3 dB 左右。

4 结论

本文提出了一种利用宏块类型信息和帧间信息的错误检测方法。该方法与错误隐藏的结合, 有效地消除了解码错块, 从而提高了解码端重建图像的质量。与其他方法相比, 此方法简单易行。

参考文献

- 1 Ferguson T J, Ranowitz J H. Self-synchronizing Huffman codes. *IEEE Trans Inf Theory*, 1984, 30(4): 687—693 [\[DOI\]](#)
- 2 Takishima Y, Wada M, Murakami H. Reversible variable length codes. *IEEE Trans Commun*, 1995, 43(2): 158—162 [\[DOI\]](#)
- 3 Gao S S, Tu G F, Robust H.263+ video transmission using partial backward decodable bit stream. *IEEE Trans Circ Syst Vid Tech*, 2003, 13(2): 182—187 [\[DOI\]](#)
- 4 Redmill D W, Kingsbury N G. The EREC: an error resilient technique for coding variable-length blocks of data. *IEEE Trans Image Process*, 1996, 5(4): 565—574 [\[DOI\]](#)
- 5 Ngan K N, Stelle R. Enhancement of PCM and DPCM images corrupted by transmission errors. *IEEE Trans Commun*, 1982, 30(1): 257—265 [\[DOI\]](#)
- 6 Rose K M, Heiman A. Enhancement of one-dimensional variable-length DPCM images corrupted by transmission errors. *IEEE Trans Commun*, 1989, 37(4): 373—379 [\[DOI\]](#)
- 7 Mitchell O R, Tabatabai A J. Channel error recovery for transform image coding. *IEEE Trans Commun*, 1981, 29(12): 1754—1762 [\[DOI\]](#)
- 8 Lam W M, Reibman A. An error concealment algorithm for images subject to channel errors. *IEEE Trans Image Process*, 1995, 4(5): 533—542 [\[DOI\]](#)
- 9 Fan C, Cui H J, Tang K. Chessboard-interpolation-based multiple description video coding. *Sci China Ser F-Inf Sci*, 2004, 47(1): 66—74
- 10 Wang G J, Zhang Q, Zhu W W, et al. An end-to-end robust approach for scalable video over the Internet. *Sci China Ser F-Inf Sci*, 2004, 47(2): 246—259
- 11 Hong M C, Kondi L, Scwab H, et al. Video error concealment techniques. *Signal Process-Image Commun*, 1999, 14(6): 437—492
- 12 Sun H, Challapali K, Zdepski J. Error concealment in digital simulcast AD-HDTV decoder. *IEEE Trans Consum Electr*, 1992, 38(3): 108—117 [\[DOI\]](#)
- 13 Wang Y, Zhu Q F, Shaw L. Maximally smooth image recovery in transform coding. *IEEE Trans Commun*, 1993, 41(10): 1544—1551 [\[DOI\]](#)
- 14 Sun H, Kwok W. Concealment of damaged block transform coded images using projections onto convex sets. *IEEE Trans Image Process*, 1995, 4(4): 470—477 [\[DOI\]](#)