Improving Data Recovery in MPEG-4

Liyang Xu1, Sunil Kumar1, Mrinal Mandal2, Member

SUMMARY In this paper, we present an MPEG-4 decoding scheme based on reversible variable length code. The scheme is purely decoder based and compliance with the standard is fully maintained. Moreover, the data recovery scheme suggested in MPEG-4 can still be used as the default scheme. Simulation results show that the proposed scheme achieves better data recovery, both in terms of PSNR and perceptual quality, from error propagation region of a corrupted video packet, as compared to existing MPEG-4 scheme.

Key words: MPEG-4, video compression, RVLC, data recovery, error resilience

1. Introduction

In order to achieve graceful degradation in the presence of errors, the MPEG-4 video standard defines several error resilience tools, such as resynchronization markers, data partitioning, data recovery and intra-refresh [1]. The reversible variable length code (RVLC) scheme is used in MPEG-4 standard to encode the DCT coefficients of macroblocks (MBs) corresponding to texture information [1,2]. The RVLC code in MPEG-4 video codec is based on a scheme proposed by Toshiba [3], where each RVLC codeword is used to encode a group of similar-valued DCT coefficients in the form of ‘LEVEL’, ‘RUN’ and ‘LAST’. Here ‘RUN’ represents the number of consecutive DCT coefficients in scan order having the same magnitude represented by ‘LEVEL’, whereas ‘LAST’ represents the last coded (usually non-zero) symbol of the block. Because RVLC coded symbols can be decoded both in the forward and backward directions, a portion of codewords in a packet can be ‘recovered’ that would have been lost in a conventional VLC.

According to RVLC decoding strategies suggested in Annex E of the MPEG-4 video standard [2], only completely decoded MBs in a corrupted video packet are recovered. In this paper, we present an improved RVLC decoding scheme that can recover additional MBs or blocks that would have been discarded (and therefore concealed) by the MPEG-4 scheme. The proposed scheme thus achieves better data recovery, both in terms of PSNR and perceptual quality. In addition, we present more conditions for error detection than those suggested in MPEG-4, and discuss properties of error propagation in corrupted video packets.

The rest of the paper is organized as follows. The discussion on error propagation and detection is presented in Section II. The proposed RVLC decoding scheme is discussed in Section III. Sections IV and V present the experimental results and conclusions, respectively.

2. Error Propagation and Detection in MPEG-4

When an error is detected during forward decoding in a bitstream, the MPEG-4 decoder skips the subsequent bits, and jumps to the next resynchronization point (i.e., VOP start code or resync_marker), and parses the stuffing bits before backward decoding of the RVLC coded symbols. Assuming that other parts of video packet preceding the RVLC codewords are error free, an error is detected if any of the following conditions is true.

1. Decoded codeword is not listed among 169 codeword patterns and escape codes, defined in the RVLC table.
2. For escape codeword, (a) decoded value for LEVEL is zero, (b) second escape code ‘ESCAPE’ is incorrect in the forward and/or backward decoding, (c) decoded values of LAST, RUN and LEVEL are found in the RVLC table, (d) marker bit between RUN and LEVEL or LEVEL and ESCAPE is 0.
3. More than 64 DCT coefficients are decoded in a block.
4. An incorrect stuffing bit sequence is found.
5. LAST is not equal to 1 and no bits are left in the buffer.
6. All the coded blocks in a video packet cannot be decoded.
7. LAST of the first codeword in the backward direction is not equal to 1.

The error conditions 1, 2 (a and b), 3 and 4 are listed in Annex E of MPEG-4 video standard [2]. Here, we have

1 The authors are with Clarkson University, Potsdam, NY, USA 13699
2 The author is with University of Alberta, Canada
also listed a few additional error conditions that we have observed in the course of our studies.

In order to get good data recovery, it is important to detect the propagating errors as soon as they occur. Analysis of a few selected error types and their behavior are discussed below.

Due to many ‘0’ valued DCT coefficients (in the scan order) at the end of the block, the number of encoded coefficients in a block is often much less than 64. As a result, a propagating error that results in an invalid value of RUN in a codeword may not be detected when the total number of decoded coefficients for a block is not more than 64. Note that sometimes errors in a codeword generate new codewords whose LEVEL values are not significantly different from the original values. Such errors do not propagate beyond the corrupted block and hence may not be detected. However, if propagating errors change the values of LAST in one or more symbols, they may be detected in the later part (or at the end) of the video packet. Errors listed in 1 and 2 can usually be detected early. However errors listed at 3 can be detected only at the end of the block, whereas errors listed at 4 through 7 can be detected only at the end of the video packet.

3. The Proposed Data Recovery Scheme

The MPEG-4 standard (Annex E) suggests four RVLC decoding strategies depending on the following two conditions: (i) total number of bits decoded in the forward and backward directions, and (ii) total number of completely decoded MBs in the forward and backward directions [2]. Note that only completely decoded MBs are considered as recovered in MPEG-4. Furthermore, all intra-coded MBs in a corrupted video packet are discarded, even when they have been fully decoded using the above procedure. This is probably done to avoid possibility of error propagation to inter-coded MBs in the subsequent frames. The standard suggests that the MBs of a corrupted video packet, which could not be recovered using RVLC, should be concealed. In this section, we present a scheme that further improves the performance of RVLC decoding and achieves better data recovery. The data recovery in the proposed scheme is based on block (8x8 pixels) instead of MB as in MPEG-4. It decodes blocks until the last error-free block, and then applies fitness function to remove the potentially corrupted blocks. The steps in the proposed scheme are as follows.

1) Read a video packet and perform forward decoding
2) If no error is detected in the packet, go to step 1 (i.e., read the next packet).
3) If error is detected in the packet header, discard the packet and go to step 11.
4) If error is detected in the RVLC symbols, record forward_code_cursor that indicates the error location, and is used to compute the last error-free block B_{n+1} as shown in Fig. 1(a). Here n+1 represents index of the block where the error was detected.
5) Go to the end of packet, parse stuffing bits, and perform RVLC decoding in the backward direction.
6) Record backward_code_cursor, which indicates the error location and is used to compute the last error-free block B_{m+1} as shown in Fig. 1(a).
7) If forward_code_cursor> backward_code_cursor, exchange the two variables.
8) Decode the blocks between [B_{m}, B_{m+1}] and [B_{m-1}, B_{m}] as shown in Fig. 1(a). Go to step 10. The last completely decoded error free block in each direction (B_{m} and B_{n} in Fig. 1(a)) is thus discarded because it may potentially contain error(s) due to error propagation. Alternatively, we can employ a more conservative decoding scheme, as illustrated in step 9.
9) During the forward decoding, if there is at least one MB between the partially decoded MB (i.e., MB_{n-1}) and the last MB (i.e. MB_{n+1}) recovered by MPEG-4 (as shown in Fig. 1(b)), only the completely decoded MBs (i.e. [MB_{m-1}, MB_{m+1}]) are recovered as in MPEG-4 scheme. The same approach is applied in backward direction. Compared to the decoding strategy in MPEG-4, the proposed (more conservative) RVLC scheme can retrieve up to four additional blocks from the otherwise discarded MB, in each direction. It happens when error is detected in the last (chrominance) block and all six blocks (i.e., four luminance and two chrominance blocks) of an MB are coded. The first chrominance block is also discarded as explained in Step 8.
10) Perform fitness inspection (discussed below) on the range of [p1, p2] in forward direction, as shown in Fig. 1(c), which is the additional data recovered as compared to MPEG-4 RVLC scheme. The same scheme is applied in backward direction.
11) Conceal all discarded blocks, including the blocks that are not recovered by the proposed RVLC decoding scheme.
12) Go to step 1.

**Fitness inspection:** Since error(s) may not be detected immediately in the bitstream, our scheme sometimes recovers the corrupted blocks considering them as error-free. In order to avoid this, we have implemented a fitness function that uses the smoothness constraint in the spatial neighborhood to check if the additional blocks recovered by the proposed RVLC decoding scheme(s) are actually error-free [5]. In this scheme, we assume that the MBs recovered by using MPEG-4 decoding strategy are error-free. We first conduct fitness inspection in the forward
direction to test the blocks in the range $[p_1, p_2]$, where $p_1$ represents the last encoded block (of the last MB) recovered by MPEG-4, and $p_2$ represents the block before error is detected by the proposed decoding scheme. We then calculate a smoothness measure across the boundary of block $X$, denoted by $f_{\text{inter}}(X)$, as follows.

$$f_{\text{inter}}(X) = \sum_{i=0}^{7} |x_i - x'_i|$$

where $x_i$ and $x'_i$ represent the pixels on the internal and corresponding external boundary (one pixel wide) of block $X$. We calculate $f_{\text{inter}}$ only for the four-connected ‘believable’ neighboring blocks of $X$, where the set of believable (spatially) neighboring blocks includes both the correctly received and the concealed blocks. We also calculate a smoothness measure inside the block $X$, $f_{\text{intra}}(X)$, as follows.

$$f_{\text{intra}}(X) = \sum_{i=0}^{7} \sum_{j=0}^{7} \left( |x_{i,j} - x'_{i,j}| + |x_{i,j} - x'_{i+1,j}| \right)$$

where $x_{i,j}$ is the element of the $i_{th}$ row and $j_{th}$ column in the block $X$.

If $f_{\text{intra}}(X) > t_1$ or $f_{\text{intra}}(X) > t_2$, where $t_1$ and $t_2$ are two empirically chosen thresholds, the block is marked as corrupted. If the block is healthy, the next block is tested, unless a corrupted block is found. The fitness inspection is then carried out in the backward direction. The corrupted blocks in a video packet, not recovered by RVLC decoding, are concealed.

The additional computational overhead introduced by the proposed scheme is the cost of the fitness function, which is at most 128 additions for an inter MB and 256 additions for an intra MB. Note that the fitness inspection is required only for a few blocks lying between $p_1$ and $p_2$, as shown in Fig. 1(c).

4. Simulation Results and Discussions

We have used Microsoft reference software [4], for implementing MPEG-4 (version 1) video codec’s simple visual profile. We have applied the MPEG-4 and the proposed RVLC scheme on 352x288 test sequences, ‘akiyo’, ‘foreman’, ‘mother daughter’, ‘flower’ and ‘coastguard’ to evaluate the performance. However, we discuss the results only for test sequence ‘foreman’ due to space limitation. The I-VOP frame was repeated after every 9 P-VOP frames. Other encoder parameters were: 10 frames/second, fixed quantizer of 10, 48Kbps, and video packet size = 1200 bits. The values of fitness function thresholds were chosen as $t_1 = 80$ and $t_2 = 320$. Table 1 shows the five error modes used for introducing the random and burst errors in our simulation, as suggested in [1]. The sequence was tested 5 times for every error type. We have not shown the results of the random error of BER=10E-2, because in this case errors are fairly spread out and almost all MBs are corrupted.

The blocks or MBs marked as corrupted after RVLC decoding are concealed by using the maximally smooth scheme [6]. Fig. 2 shows a decoded frame of ‘foreman’ sequence obtained using the MPEG-4 and the proposed scheme. Fig. 2(a) shows the original frame. Fig. 2(b) shows the corrupted frame without any error concealment. In Figs. 2(c) and 2(d), the DC coefficients are available for concealing corrupted blocks. It is observed from Fig. 2(d) that the proposed scheme provides perceptual improvement in the areas (pointed by arrows) of eyes, right eyebrow, mouth, the boundary between neck and collar (right side) and the building (in the upper half of the frame), by recovering the corresponding blocks. These blocks could not be recovered by MPEG-4 scheme resulting in degradation in subjective quality as shown in Fig. 2(c). The average PSNR of the whole test sequence for various error types is shown in Table 2. It is observed that the proposed scheme achieves a PSNR improvement in the range of 0.5 to 1.5 dB as compared to MPEG-4, except in the case of error type 4 in which the error is very limited. However, we achieved an average PSNR improvement of only about 0.2 to 0.5 dB when only first frame was encoded as I-VOP and the packet size was 500 bits. The improvement in PSNR thus depends on the packet size. As illustrated in Fig. 3, the proposed scheme consistently achieves better PSNR performance as compared to MPEG-4 for almost all frames for a representative error type. The PSNR performance of another version of the proposed scheme (Step 9 in Section III) is similar, as the proposed scheme recovers additional MBs only occasionally. We also noticed similar PSNR performance for other error types for all the tested sequences.

The proposed scheme can sometimes recover corrupted blocks or MBs. However, the frequency of the resulting degradation in PSNR of the frame is less than 5%. Even when the PSNR does degrade, the degradation is less than 0.3 dB.

The RVLC-based data recovery scheme suggested in MPEG-4 Annex E is relatively more conservative in terms of recovering blocks from the error propagation region. Therefore the use of fitness inspection whose main function is to reject potentially erroneous blocks is not required with it.
5. Conclusions

In this paper, we have proposed an improved RVLC decoding scheme that recovers more blocks (and sometimes more MBs) from error propagation region of corrupted video packets, as compared to the MPEG-4 scheme. It has been shown that the proposed scheme achieves a better data recovery, both in terms of PSNR and perceptual quality. We have also presented a few more conditions for error detection than those suggested in MPEG-4 and also discussed properties of error propagation in corrupted video packets. Although the proposed scheme is only a simple modification of the MPEG-4 scheme, it achieves reasonable improvement in the quality of the reconstructed frames in the presence of errors, with negligible computational overhead. Since the proposed scheme is employed only in decoder, compliance with the standard is fully maintained.

REFERENCES


Table 1: The error types

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Bitrate (bits/sec)</th>
<th>Burst or Random</th>
<th>BER</th>
<th>Burst time</th>
<th>Free Time*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>48,000</td>
<td>Burst</td>
<td>10E-2</td>
<td>10ms</td>
<td>50ms</td>
</tr>
<tr>
<td>2</td>
<td>48,000</td>
<td>Burst</td>
<td>10E-3</td>
<td>10ms</td>
<td>50ms</td>
</tr>
<tr>
<td>3</td>
<td>48,000</td>
<td>Burst</td>
<td>10E-2</td>
<td>1ms</td>
<td>50ms</td>
</tr>
<tr>
<td>4</td>
<td>48,000</td>
<td>Burst</td>
<td>10E-3</td>
<td>1ms</td>
<td>50ms</td>
</tr>
<tr>
<td>5</td>
<td>None</td>
<td>Random</td>
<td>10E-3</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

* Free Time is the minimum interval in which no bit error happens in the bitstream. This parameter is used in burst errors.

Table 2: Comparison of PSNR for P-VOP and I-VOP in the test sequence

<table>
<thead>
<tr>
<th>Seq.</th>
<th>Test sequence: foreman_cif (352X288)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P-VOP</td>
</tr>
<tr>
<td>Error Type</td>
<td>MPEG-4 RVLC</td>
</tr>
<tr>
<td>1</td>
<td>28.31 dB</td>
</tr>
<tr>
<td>2</td>
<td>32.45 dB</td>
</tr>
<tr>
<td>3</td>
<td>32.54 dB</td>
</tr>
<tr>
<td>4</td>
<td>34.82 dB</td>
</tr>
<tr>
<td>5</td>
<td>27.20 dB</td>
</tr>
</tbody>
</table>

Figure 1. Illustration of improved RVLC data recovery scheme: (a) shows the proposed block-based RVLC decoding scheme; (b) shows the situation that the proposed RVLC scheme is the same as MPEG-4 RVLC scheme; (c) shows the MBs between p1 and p2 that are needed to be inspected by fitness function.
Figure 2. Intra-VOP frame of ‘foreman’ test sequence, (a) original frame, (b) corrupted frame, (c) reconstructed frame using MPEG-4 scheme, (d) reconstructed frame using the proposed scheme. The areas pointed by arrows are recovered by the proposed scheme instead of being concealed. The area in the circle represents the recovered blocks with error(s), which are not detected by the fitness function.

Figure 3. Comparative performance of the RVLC decoding schemes in terms of PSNR.