Techniques for the Transport of MPEG-4 Video over Wireless Networks

Bo YAN and Kam W. NG

Abstract—In this paper, we explained the techniques for the transport of MPEG-4 video over wireless networks. As a new object-based video compression standard, MPEG-4 has been proved to be suitable for wireless applications. However, because of the characteristics of wireless channels, such as burst bit errors, some new error-resilient video coding techniques are required to be incorporated into MPEG-4 to improve the coding efficiency. These techniques are discussed in this paper including suggestions for future research.

Index Terms—Error resilient video coding, MPEG-4, UEP, Wireless channel.

I. INTRODUCTION

In the last decade, mobile communication has grown rapidly. With the explosive growth, the need for the robust transmission of multimedia information—audio, video, text, graphics, and speech data—over wireless links is becoming an increasingly important application requirement. Since the video data may occupy more bandwidth than the other media data during transmission, it should be given more consideration in the wireless multimedia communication. As we know, in order to be transmitted over the wireless network, the source video data must be compressed before transmission. During the last decade, many video compression standards have been proposed for different applications, such as MPEG-1, MPEG-2, H.26X and so on. But these traditional video compression techniques cannot efficiently meet the requirement for video transmission in wireless networks. In the last few years, MPEG-4 has been proposed, which is suitable for wireless networks because of its characteristics [1][2][3]. In this paper, the techniques of MPEG-4, which make it suitable for applications in wireless multimedia communication, will be introduced. This paper is divided into three sections. Firstly, an overview of the MPEG-4 standard is given. In this section, the basic characteristics of MPEG-4 will be described, and the basic coding and decoding procedures will also be explained. Then we will give the reasons why MPEG-4 is qualified for wireless multimedia communication. Secondly, it is well known that wireless channels are error-prone, burst errors of which can degrade the video quality greatly. In order to alleviate the effect, some error-resilient video coding techniques are incorporated in MPEG-4, including error detection, recovery and concealment, which will be described. Finally, we can draw a conclusion that these improvements are not enough and the future work will be discussed.

II. OVERVIEW OF MPEG-4

MPEG-4 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group), the committee that also developed the Emmy Award winning standards known as MPEG-1, MPEG-2 and MPEG-7. Among these standards, except for MPEG-7, the others are video compression standards.

MPEG-4 is a new object-based method concerned about video and audio compression, which is different from MPEG-1 and MPEG-2.

![Fig. 1. The Structure of MPEG-4 Encoder](image-url)

Fig. 1 outlines the basic approach of the MPEG-4 video algorithms to encode not only the rectangular but also arbitrarily shaped input image sequences. The basic coding structure involves shape coding (for arbitrarily shaped video objects) and motion compensation as well as DCT-based texture coding (using standard 8x8 DCT or shape adaptive DCT). A very important advantage of MPEG-4, which is a content-based coding approach, is that the compression efficiency can be significantly improved for some video sequences by using appropriate and dedicated object-based motion prediction “tools” for each object in a scene. There are many motion prediction techniques, which can be used to allow efficient coding and flexible presentation of the objects as...
follows [1].
1) Standard 8x8 or 16x16 pixel block-based motion estimation and compensation.
2) Global motion compensation based on the transmission of a static “sprite”. A static sprite is a possibly large still image, describing panoramic background. For each consecutive image in a sequence, only 8 global motion parameters describing camera motion are coded to reconstruct the object. These parameters represent appropriate affine transform of the sprite transmitted in the first frame.

There are four hierarchically organized classes in the MPEG-4 visual standard as follows [2]:
1) Video Session: Each video session (VS) is made up of one or more Video Objects (VO), corresponding to the various objects in the scene.
2) Video Object: Each of the VO's can have several scalability layers (spatial, temporal, or SNR), corresponding to different Video Object Layers (VOL).
3) Video Object Layer: Each VOL consists of an ordered sequence of snapshots in time, called Video Object Planes (VOP);
4) Video Object Plane: Each VOP is a snapshot in time of a VO for a certain VOL.

This way, one VS can have several VO's, each of these VO's having several VOL's, which are the sequences in time of several VOP's. And finally, each VOP is characterized by its shape, motion and texture.

Although MPEG-4 is an object-based standard, its texture coding is still block-based and somewhat similar to the traditional video coding standards. While coding, the first step in encoding one arbitrary VO is finding a rectangular bounding box that completely contains the object to be encoded. Then this bounding box, corresponding to a VO, is divided into blocks of 16x16 pixels called macroblock (MB), which are then encoded one by one. In every MB, there are four 8x8 luminance blocks and one 8x8 block for each sub-sampled (in both directions) chrominance, therefore giving a total of 6 blocks per MB.

Firstly, the shape coding is applied. Each MB is analyzed and classified according to three possible classes: transparent (MB outside the object but inside the bounding box), opaque (MB completely inside the object) or border (MB over the border). The shape coding method called Content-based Arithmetic Encoding (CAE) is applied on the border MBs [4].

Secondly, similar to other traditional methods, the motion information is encoded by means of motion vectors. Each MB can either have one or four motion vectors. When four motion vectors are used, each of them is associated with an 8x8 block within the MB. With the motion vectors, the decoder can know which block of pixels in the previous VOP is closest to the current one and it can be used for the prediction of the texture. In the receiver, the decoder will use the motion vectors for motion compensation.

Then the texture data can be encoded by two modes: intra and inter. For the intra mode, a given MB is encoded by itself (with no temporal prediction), only exploring the spatial redundancies. On the other hand, for the inter mode, motion compensation is used to explore the temporal redundancy, and the difference between the current and prediction MB is encoded. Both the absolute texture value (intra coding) and the differential texture value (inter coding) are then encoded using the DCT transform. Then the DCT coefficients are encoded by run-length coding and variable-length coding (VLC). Instead of the regular VLC, reversible VLC can be used to code the texture, if error resilience is a requirement.

At this stage we get all the encoded information including shape, motion and texture [2][5].

Wireless multimedia applications face technical challenges, which are significantly different from the problems typically encountered with desktop multimedia applications. One important reason is that the wireless channel is error-prone with bits errors and burst errors due to fading and multi-path reflections.

MPEG-4, designed as an adaptive representation scheme that also accommodates very low bitrate applications, is very appropriate for wireless multimedia applications. Concretely, MPEG-4 is useful because many error-resilient tools are incorporated into MPEG-4, which guarantee the correctness of the data in the error-prone wireless channel [6].

III. ERROR-RESILIENT VIDEO CODING IN MPEG-4

Wireless channels are typically noisy and suffer from a number of channel degradations such as bits errors and burst errors due to fading and multi-path reflections. The channel errors can affect the compressed video bit-stream very severely. On the one hand, if the decoder loses information about a frame or a group of pixels, it won’t be able to decode the frame or the pixels. And it also cannot decode the frames or the pixels coded using them. On the other hand, the decoder can lose synchronization with the encoder because of the lost information, which leads to the remaining bit-streams being incomprehensible. Therefore for a robust video compression method in the wireless applications, resynchronization and robust techniques are necessary in the encoder.

Generally the MPEG-4 decoder can apply the syntactic and semantic error detection techniques to enable the video decoder to detect when a bitstream is corrupted by channel errors. In motion compensation and DCT, the decoder can detect bitstream errors by applying the checks as follows [7]:
1) The motion vectors are out of range;
2) An invalid VLC table entry is found;
3) The DCT coefficient is out of range;
4) The number of DCT coefficients in a block exceeds 64.

After errors are detected, some techniques incorporated into MPEG-4 can be applied which can provide the important properties such as resynchronization, data recovery, and error concealment. They are as follows:
1) Video packet resynchronization;
2) Data partitioning (DP);
A. Video Packet Resynchronization

When errors occur in the bit-stream, the video decoder decoding the corrupted stream may lose synchronization with the encoder, i.e. the precise location of the stream in the video is uncertain. It will degrade the decoded video quality very greatly and make it unusable.

A traditional scheme to solve this problem is to introduce resynchronization markers in the bit-streams at various locations. When the decoder detects errors, it can hunt for the next resynchronization marker to regain synchronization. In previous video coding standards such as MPEG-1, MPEG-2 and H.263, the resynchronization markers are fixed at the beginning of each row of macro-blocks. But the quantity of information between two markers is not fixed because of the variable length coding as shown in Fig. 2 [8].

MPEG-4 provides a similar method of resynchronization with one important difference: for the MPEG-4 encoder, the resynchronization markers are not restricted to be inserted at the beginning of each row of macroblocks. The encoder has the option of dividing the image into video packets, each of which is made up of an integer number of consecutive macroblocks. In this way, these macroblocks can span several rows of macroblocks in the image and even include partial rows of macroblocks. Generally, the MPEG-4 encoder can insert the resynchronization markers periodically at every K bits as shown in Fig. 3, which divides the bit-stream into packets of data that are independent of other named video packets.

B. Data partitioning

After detecting an error between two resynchronization markers, usually all macroblocks in the video packet have to be discarded. This is due to the fact that between two resynchronization markers, the motion and DCT data for each macroblock are all coded together. So when errors are detected, the decoder is not sure where the errors occur, in the motion part, texture part or both. Fig. 4 shows the organization of the video data within a video packet for a typical video compression scheme without data partitioning. For a video packet, it should be noted that the motion vectors are predictively coded with respect to the neighboring motion vectors; hence, only the motion vector differences are coded. DCT data comprises the 64 DCT coefficients actually encoded via zig-zag scanning and run-length-encoding, and then a VLC table.

In order to avoid discarding all the video data in the VP while detecting an error, MPEG-4 incorporates the data partitioning mode which partitions the data in a video packet into a motion part and a texture part separated by a motion boundary marker (MBM), as shown in Fig. 5. The MBM marks the end of the motion part and the beginning of the texture part. The MBM is computed from the motion VLC tables using a search program such that this marker word is hamming distance 1 from any possible valid combination of the motion VLC tables. The word is uniquely decodable from the motion VLC tables. It should be noted that the MBM is only computed once based on the VLC tables and is fixed in the standard. In MPEG-4, the MBM is a 17-bit word whose value is 1 1111 0000 0000 0001.
The number of macroblocks (NMB) in the video packet can obviously be derived after encountering the MBM. When an error is detected in the texture session, the decoder will discard all the texture data in the video packet and conceal the error by motion compensation of the NMB motion vectors. When an error is detected in the motion session, all the data in the video packet will be discarded by the decoder, because the texture data is coded based on the motion data and it will be useless without the motion data.

If no error is detected in the motion and texture sections of the bitstream, but the resynchronization marker is not found at the end of decoding all the macroblocks of the current packet, an error is flagged. In this case, only the texture data in the VP need to be discarded. The motion data can still be used for the NMB macroblocks since we have higher confidence in motion vectors because of the detection of the MBM.

The data partitioning scheme is only used for I-VOP and P-VOP. And for the two cases the content in the motion and texture part is different. For I-VOP, the motion part contains the coding mode and DC’s coefficients and the texture part is the AC’s coefficients. For P-VOP, the motion part is the motion vectors and the texture part contains the DCT coefficients [7][10][12].

C. Reversible variable Length codes (RVLC)

As mentioned above, the texture part is coded with variable length code. Therefore during decoding, the decoder has to discard all the texture data up to the next resynchronization marker when it detects an error in the texture part because of losing synchronization. RVLC can alleviate the problem and recover more DCT data from a corrupted texture partition. The RVLC is a special VLC, which can be decoded in both the forward and reverse direction. When the decoder detects an error while decoding the texture part in the forward direction, it can find the next resynchronization marker to decode the texture part in the backward direction until an error is detected. Therefore only the data between the two error locations is discarded, and other data in the packet can be recovered. In Fig. 6, only the data in the shaded area is discarded. It should be noted that the RVLC scheme can only be used with the help of resynchronization and data partitioning techniques [7][10][13].

D. Header Extension Code (HEC)

At the beginning of each video packet, there is the most important information for decoding—the header data, which contains information about the spatial dimensions of the video data, the time stamps associated with the decoding and presentation of the video data, and the mode in which the current video object is encoded (INTRA or INTER). If the header information is corrupted by the channel errors, the whole frame has to be discarded. So the MPEG-4 standard introduces the technique named Header Extension Code to reduce the sensitivity of the header data, as shown in Fig. 5. In the technique, a 1-bit field called HEC is inserted for every video packet. If the bit is set, the header information of the video frame is repeated following the HEC. The duplicated information can ascertain the decoder to get the correct header data. The HEC usually is used in the first video packets of the VOP, but not in all of them [7][10].

E. Unequal Error Protection Technique (UEP)

The MPEG-4 video compression standard has incorporated several error resilience tools to enable detection, recovery and concealment of errors specially for wireless applications. Although these tools are efficient for combating bit errors when the bit rate errors (BERs) are less than 10⁻³, the typical wireless channels often have much higher BERs. Therefore, channel coding is required to reduce the number of errors in the compressed bit stream that is sent to the MPEG-4 video decoder. The technique of Unequal Error Protection (UEP) is introduced into the compressed bit stream to ensure that fewer errors in the important portions of the bitstream will occur.

In order to perform the Unequal Error Protection in the compressed video streams, the “data partitioning tool” is exploited in the video packet. For a video packet, the information bits are divided into three partitions, each of which has a different sensitivity to channel errors and may thus be protected with a different channel coding rate.

As far as the I-VOP is concerned, the partitions consist of a header, DC DCT coefficients and AC DCT coefficients. And for P-VOP, the partitions contain a header, a motion partition and a texture partition. Among these three partitions, the information bits of the header are the most important, since the whole packet will be discarded if the header is received with errors. So the header partition should be strongly protected with the lowest rate. For I-VOP, DC coefficients have an higher subjective importance than AC coefficients, thus they should be protected with a lower rate. Similarly, for P-VOP, the motion part should be more protected than the texture data, since without the texture data the motion-compensation can conceal the errors without too much degradation of the reconstructed picture [14][15][16][17].

Fig. 7 illustrates the protection scheme described above, in which R₁, R₂ and R₃ represent the bit rates of the three partitions respectively and they satisfy the condition: R₁<R₂<R₃. To realize it, the Rate Compatible Punctured Convolutional (RCPC) codes are used [18]. In this case, the codes considered are obtained by puncturing the same “mother” code and only one coder and one decoder are required for performing the
coding and decoding of the whole bitstream [19].

### Table 1

<table>
<thead>
<tr>
<th>Bit Rate Errors</th>
<th>QCIF Format</th>
<th>CIF Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>EEP (dB)</td>
<td>UEP (dB)</td>
</tr>
<tr>
<td>0.05</td>
<td>34.7</td>
<td>34.8</td>
</tr>
<tr>
<td>0.1</td>
<td>32.3</td>
<td>33.5</td>
</tr>
<tr>
<td></td>
<td>30.8</td>
<td>31.8</td>
</tr>
<tr>
<td></td>
<td>26.2</td>
<td>26.7</td>
</tr>
</tbody>
</table>

In Section A. Conclusions component (Y) of the frame needs to be considered.

To evaluate the video quality, only the PSNR of the luminance component (Y) of the frame needs to be considered.

### IV. CONCLUSIONS AND FUTURE WORK

Generally there are two kinds of video formats often used, i.e. Common Intermediate Format (CIF) and Quarter CIF (QCIF). The sizes of their images are 352x288 and 176x144 respectively.

In the signal processing field, the video quality is often evaluated in terms of PSNR, Peak Signal Noise Ratio, in dB, defined as:

$$\text{PSNR} = 20 \log_{10} \frac{255}{\text{RMSE}}$$

where RMSE is the square root of the Mean Square Error (MSE):

$$\text{MSE} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |f(i, j) - F(i, j)|^2}{M \times N}$$

where we assume $f(i, j)$ and $F(i, j)$ are the source and the reconstructed images, containing MxN pixels each [20]. To evaluate the video quality, only the PSNR of the luminance component (Y) of the frame needs to be considered.

### A. Conclusions

In Section III, to improve the video quality against the bit errors, some techniques including resynchronization, DP, RVLC and HEC have been discussed. Actually all the techniques have been evaluated thoroughly and independently verified by two parties before being accepted into the MPEG-4 standard. The techniques are rigorously tested over a wide variety of test sequences, bit rates and error conditions. The compressed bitstreams are corrupted using random bit errors, packet loss errors, and burst errors. To provide statistically significant results, 50 tests are performed for each of the mentioned error conditions. In each test, errors are inserted in the bitstreams at different locations. This is achieved by changing the seed of the random number generators used to simulate the different error conditions [7][21][22].

For each test, the peak signal-to-noise ratio, (PSNR) of the video decoded from the corrupted stream and the original video is computed. Then the average PSNR of the 50 runs is computed for each frame in the sequence. To evaluate the performance of the proposed techniques, the average PSNR values generated by the error-resilient video codec with and without each error resilient tool are compared. The techniques are also compared on the basis of the number of frames discarded due to errors and the number of bits discarded. The comparison results have proved that these techniques have improved the video quality greatly in error-prone channels [7].

However, these techniques are not enough for wireless channels with high bit errors rate (BERs), so the channel coding method, UEP, is introduced. To test the performance of the UEP against others such as EEP (Equal Error Protection), in which all partitions are protected equally, many experiments have been made based on the wireless channel. The results of them have shown that the UEP technique can improve the quality of the reconstructed video at a high channel error rate more greatly than EEP. Table I shows the results of the experiments made by Wendi [15] in GSM channel. In the table, UEP produces higher average PSNR for the reconstructed video for both CIF and QCIF images at high channel error rates, which verify the good performance of the UEP. Usually in many cases, UEP may leave more errors in the channel decoded bitstream than EEP, but these errors are in less important portions of the video packet and degrade less the video quality.

<table>
<thead>
<tr>
<th>Total Bandwidth (kbps)</th>
<th>EEP (dB)</th>
<th>UEP (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>17.2</td>
<td>21.9</td>
</tr>
<tr>
<td>120</td>
<td>26.8</td>
<td>27.2</td>
</tr>
<tr>
<td>130</td>
<td>32.4</td>
<td>32.5</td>
</tr>
</tbody>
</table>

Cai has made other experiments [14] to compare the coding performance between the EEP and UEP, in which the source coding rate is fixed at 100kbps and channel coding rate is varying. Table II shows the simulation result. For example, total bandwidth of 110 kbps represents 100 kbps of fixed source coding rate and 10 kbps of channel coding rate. In this case, channel coding redundancy is 10%. Similarly, the redundancy of other cases can be derived. The experimental results demonstrate that in bandwidth-stringent cases, UEP is much better than EEP with the gain of nearly 5 dB at 10% redundancy. When the channel coding redundancy increases to 20% or more, the performance of EEP is nearly the same as that of UEP. This is because high redundancy channel coding provides virtually error free environment for all classes of the video bitstream so that the advantage of UEP is diminishing.

### B. Future work

In future, the work will be focused on how to further improve the efficiency of the error-resilient video coding. The proposals are as follows.

Firstly, future work about UEP to improve the performance should consider some issues concerned with the characteristics
of the wireless channels and video signals. In wireless communication, the power problem is an important factor, which greatly affects the communication performance. Since the UEP will generally consume more power than EEP, the power problem should be taken into account for wireless communication when applying the UEP. This is particularly necessary when the bandwidth is not quite stringent and the gain of UEP over EEP is not so significant as shown in Table II at the rate of 120 kbps or over. There is a tradeoff between power and video quality. Therefore a decision scheme needs to be designed to determine whether UEP should be adopted under a given channel condition.

Secondly, to improve the performance of UEP, it is necessary to get the optimal allocation of channel coding rates among different classes of video data. In order to carry out the optimal allocation, the relationship of the error sensitivity among different partitions of the video packet is required. Without such analytical relationship, it is difficult to design the optimal allocation. Better results may be achieved if rates are chosen according to accurate sensitivity studies on the bitstream. This will also be one part of our future research work.

Finally as mentioned in Section III, by applying the RVLC in the DCT data partition, the decoder can get more data to decode when errors occur. Experimental results have proved that it has improved the performance of the MPEG-4 coder greatly in the wireless channels [21]. Currently, the RVLCs are used only for the DCT data partition. In the future work, the RVLCs may be introduced into the motion data partition to code the motion vectors. Since the motion part is more important than the DCT part, we expect that it will further improve the efficiency of the error-resilient video coding of MPEG-4.

In conclusion, although many techniques have been introduced into the MPEG-4 standard, which aim at providing strong supports for robust transmission of the video data in wireless multimedia communication, actually they are not enough at all for all applications. So there is still a lot of research work to be pursued to improve the performance of video transmission over wireless channels.

REFERENCES

[18] Hagenauer J., "Rate-compatible punctured convolutional codes (RCPC codes) and their applications", IEEE Transactions on Communications, Volume: 36 Issue: 4, April 1988 pp.389 –400