Improved Error Resilient H.264 Coding Scheme Using SP/SI Macroblocks

Xiaosong Zhou, Wei-Ying Kung and C.-C. Jay Kuo

Integrated Media Systems Center and Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089-2564

ABSTRACT

An error resilient H.264 coding scheme using SP/SI macroblocks is presented in this work. It is able to generate alternative SP macroblocks utilizing multiple reference frames [2] and the concealed versions of the corrupted frames [3]. These alternative macroblocks are used to replace the original ones in the output video stream to protect them from being affected by previous errors detected at the decoder side. The introduced bit bate is further reduced by adjusting quantization levels adaptively. Specifically, different versions of alternative SP macroblocks can be coded using different quantization levels, which is associated with different levels of error resilience performance and different bit rate consumptions. A proper alternative version is encoded according to the importance of the macroblock. The importance of the macroblock is measured by its influence on subsequent frames if the macroblock is not correctly reconstructed. Accordingly, fewer bits are used to replace those macroblocks with less importance. Simulation results demonstrate the proposed approach achieves an excellent error resilient capability and an improvement in reducing the bit rate overhead.

Keywords: H.264, video transmission, error resilience, error recovery, error concealment, SP/SI macroblocks

1. INTRODUCTION

Video communication applications become more popular than ever these days. All existing communication networks (wired, wireless and cellular networks) start to provide sufficient bandwidths for transmission of high-quality video data. At the same time, emerging video coding standards such as H.264 help reduce the bit rate of compressed video substantially. As more and more electronic devices and portable devices have the built-in video displaying capability, there will be a tremendous growth in these applications in the near future.

However, the transmission of compressed video over erroneous channels is still a challenging problem. Transmission errors or corrupted video data packets can degrade the perceptual quality of received video substantially. Moreover, new challenges have been raised by new applications. Better video quality is always desired and boosted by a larger display screen with higher resolutions. On the other hand, wireless networks, which contain inevitable bit errors, are used more widely and will continue to grow. With all these reasons, it is extremely important to design an effective error resilient coding and transmission scheme that is able to preserve the high quality level of compressed video over erroneous channels such as wireless networks. Our work aims at the design of such a scheme that is able to work with the state-of-the-art video coding standard – H.264.

In this work, an error resilient H.264 coding and transmission scheme using SP/SI macroblocks is presented. The scheme is able to provide excellent error resilience performance and help preserve the high quality level of the received video. It is able to work effectively under high bit error rates and different network delays. It is specifically designed for H.264, and can be adapted to any other coding standards that provide SP/SI or similar coding features.

The rest of this paper is organized as follows. In Sec. 2, some background information is reviewed. In Sec. 3, the proposed scheme is discussed in detail, where we first describe the overall error resilience scheme and explain how it works to reduce or eliminate propagation errors. Afterward, the two encoding methods of generating alternative SP/SI macroblocks are reviewed. We will then explain how to improve these encoding methods in terms of coding efficiency.

Detailed implementation notes and discussions are also provided in this section. In Sec. 4, experimental results are presented to show the performance of the improved scheme. Concluding remarks are given in Sec. 5.

2. BACKGROUND REVIEW

2.1 Overview of H.264 Video Coding Standard

H.264 (or MPEG4 Part-10 AVC) is an emerging video coding standard proposed by the Joint Video Team (JVT). It is shown to outperform other existing video coding standards in terms of coding efficiency. H.264 uses the hybrid block-based motion compensation and transform coding model as existing standards such as H.263 and MPEG-4 [4]. Furthermore, a number of new features and capabilities have been introduced in H.264 to effectively improve the coding performance. The standard provides more coding options for motion-compensated predictive coding. It allows 7 block sizes to be used for motion estimation. It also defines quarter-pel motion vector resolution in order to generate more precise predictions. In addition, more than one previous frame can be stored as reference frames for both P and B frames. The standard also utilizes intra predictive coding to enhance the compression of intra-coded macroblocks. An intra coded macroblock or block is predicted from its neighboring pixels and only the prediction mode and resulting residuals are encoded. A number of other coding features and techniques are also introduced in H.264 to help providing better coding performance. For example, the in-loop de-blocking filter plays an important role in enhancing the subjective quality of the decoded video; context-based binary arithmetic coding (CABAC) is shown to outperform variable length coding (VLC) and can be used to further reduce the bit rates during the entropy coding process.

2.2 Error Propagation and Exiting Error Resilience Tools

In most video transmission scenarios, unless a dedicated link that provides a guaranteed level of quality-of-service (QoS) is available between the source and the destination, data packets may be lost or corrupted [5] during transmission. As a result of predictive coding in the time domain, such transmission errors can affect not only the current frame but also subsequent frames. We call this phenomenon error propagation, and the associated errors the propagation (drift) errors. Error propagation can degrade the PSNR and subjective quality of received video for a long period of time and it is the biggest challenge for error resilience in video transmission. If propagation errors can be completely eliminated or largely reduced, the quality level of received video can be well preserved.

Various tools have been introduced in the past to improve the error resilience of compressed video. Many of these tools can also be adapted to work for H.264 [12-20]. One way to reduce error propagation is to insert intra coded macroblocks in P or B frames randomly or with certain patterns. It is known as intra refreshing. This technique was shown to be effective for previous standards in certain cases. However, as intra macroblocks in H.264 are coded with intra predictions, the insertion of intra macroblocks may fail to stop error propagation since the referenced neighboring macroblocks might be corrupted as well. Data partition and data embedding intend to give more protections to more important parts in the bit stream to reduce the damage of transmission errors, and error concealments tools [6] can also be implemented in the H.264 decoder to enhance the video quality of corrupted frames, but they are not designed to completely stop the error propagation.

A number of feedback-based error resilience schemes [7-9] were also introduced. In these schemes, the sender (encoder) obtains the error feedback information from the receiver (decoder) so that it avoids referencing the corrupted areas/frames to encode subsequent frames accordingly. They are effective in reducing error propagation but all of them require the video source to be encoded in real time. For many scenarios, the video source (especially those for commercial video contents) is encoded off-line before the transmission takes place. Furthermore, real-time H.264 encoding demands great computational power and can be expensive. Therefore, these schemes may not work well for H.264 applications.

2.3 SP/SI Picture coding in H.264

SP and SI picture coding for the H.264 video coding standard was first introduced by Karczewicz and Kurceren in [10] and [11]. The motivation of the SP/SI picture is to enable flexible stream switching and splicing capabilities for H.264 coded video. An example of stream switching using SP pictures is shown in Fig. 1, where an image sequence is coded into two bit streams (A and B) with different bit rates and quality levels. It is assumed that the five frames shown in the figure are consecutive P frames (P_{A1}~P_{A5} and P_{B1} ~ P_{B5}) and each of these P frames is coded by forward prediction from its preceding frame. During the transmission, if there is a need (e.g. changing bandwidth) to switch the transmitting stream from stream A to stream B, we will encounter some problems by sending out the coded P frames in stream B directly. For example, if we switch the two streams at the point of the third frame by simply sending P_{B3} instead of P_{A3}, the decoder is not able to reconstruct P_{B3} correctly since the reference frame used to encode P_{B3}, i.e. reconstructed $P_{B2,}$ is not available in the reference frame buffer of the decoder and the decoder may attempt to use reconstructed frame P_{A2} as the reference frame to decode P_{B3}, which may result in significant errors.

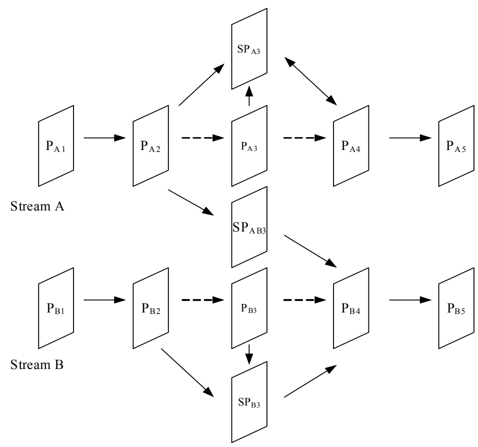


Fig. 1: Stream switching using SP pictures.

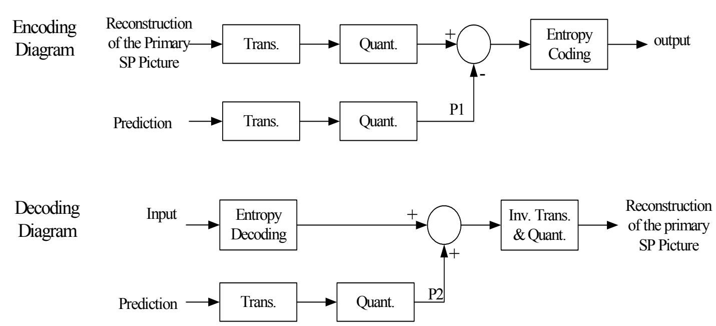


Fig. 2: The encoding/decoding diagram of secondary SP pictures.

This problem can be solved using SP pictures. First, the third frames in the two streams are coded into two SP frames instead of P frames, namely SP_{A3} and SP_{B3}, which are called primary SP pictures. When the bit stream is being switched, a secondary SP picture SP_{AB3} that is coded using P_{A2} as the reference frame will be transmitted instead of SP_{A3} or SP_{B3}. The way to encode SP_{AB3} ensures that an identical reconstruction as that of the SP_{B3} can be obtained by decoding it so that the stream switching process will not introduce any mismatch between the encoder and decoder. After that frames in stream B can be sent directly starting from the fourth frame. Thus, the two streams are successfully switched. Similarly, another secondary SP frame SP_{BA3} can be created in order to switch from stream B to stream A. We can create this kind of SP frames at certain points of the video stream to enable switching at these points.

We are not going to discuss the primary SP picture coding in detail. Basically, the reconstruction of the primary SP picture can be quantized and de-quantized without loss at certain quantization level Q_{SP}, which is going to be used in the encoding process of the secondary SP picture. The special way to encode the macroblock in the secondary SP picture SP_{AB3} is the reason that its reconstruction can be identical to the reconstruction of another coded version – the primary SP picture SP_{B3} in our example. Fig. 2 shows the simplified encoding and decoding diagrams for such macroblocks. The exact reconstruction of the primary SP picture (SP_{B3}) is being coded instead of the original video content. The prediction to be used can be chosen arbitrarily and it is obtained from P_{A2} in our example. Another difference of this coding method from general motion-compensated macroblock coding is that quantization is processed before calculating the residuals. The quantization level used has to be equal to Q_{SP} set by the primary SP picture coding as mentioned above. By decoding such an SP macroblock, the exact reconstruction in the primary SP picture can be obtained as explained below. First, the transformed and quantized different predictions will not introduce any mismatch (the value of P_1 in the encoder and P_2 in the decoder diagram are exactly the same). Second, as the reconstruction value can be quantized and de-quantized without any loss at the quantization level Q_{SP}, no quantization error will be introduced in the process. These SP pictures have better coding efficiency than I frames. Thus, it is efficient to use them for the stream switching purpose.

Similarly, SI pictures are encoded and used under the stream switching scenario. The only difference is that SI pictures do not use any reference frames, the macroblocks are coded by using intra predictions, which means that in the encoding diagram shown in Fig. 2, the prediction is generated by intra prediction from neighboring blocks.

The possibilities of utilizing SP/SI encoding for the error resilient purpose was mentioned before. But all of them were based on SP/SI coding at the picture level, which may not work effectively. Besides, there has been no work in the literature that offers a concrete error resilient scheme based on SP/SI coding idea.

3. PROPOSED ERROR RESILIENCE SCHEME

3.1 Error Resilient Transmission Scheme

Fig. 3 provides a general diagram of video data transmission, where the sender (typically, a media server) sends compressed video streams to the receiver(s). Upon receiving the video stream, a real-time decoder at the receiver side decodes the video stream and sends the output to display devices. One common problem for the system is that the potential transmission errors may cause a failure of guaranteed QoS, especially in wireless networks. The simple retransmission scheme provided by the network may not help since it will introduce intolerable delay for video. Fig. 4 shows how video frames are coded using temporal predictions. As shown in the figure, if transmission errors occur in one frame, parts of that frame are corrupted and cannot be decoded or displayed correctly. Furthermore, some macroblocks in the next frame will not be reconstructed correctly as well if they use these corrupted area as their predictions. Similarly, the third frame will also be affected by the incorrect reconstruction of the second frame. As long as the future frames are coded by predictions from these affected frames, such propagation errors will keep going.

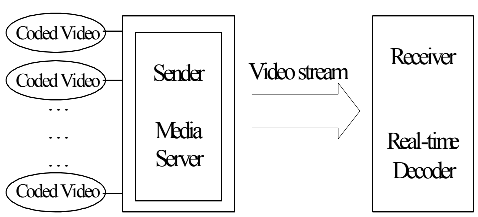


Fig. 3: Video transmission systems.

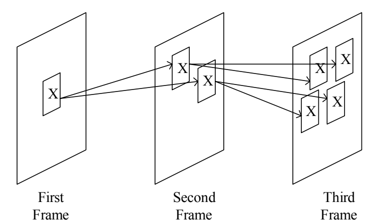


Fig. 4: An example of error propagation.

However, if the macroblocks in the subsequent frames do not reference the corrupted area in the first frame, the situation described above will not take place and the transmission error will not affect any of the future frames. As a result, error propagation is eliminated. There are certain ways to ensure this in the encoding process. Suppose that a real-time encoder is working on-line and it receives feedback from the decoder, it can simply mark the corrupted areas in its reference frame buffer according to the feedback and avoid using them as the predictions for future frames. By such a mechanism, we guarantee a 'clean' coded video stream without being affected by previous errors. Unfortunately, in most application scenarios, video streams are coded off-line before transmission starts since real-time encoding is usually not required and the cost for implementing a real-time encoder is higher. This is especially true for the H.264 encoder. Therefore, if an effective scheme, which works by following a similar principle as stated above, can be developed for off-line H.264 encoders, the error propagation problem will be solved.

We propose such an error resilience transmission scheme that is able to eliminate or largely reduce propagation errors for off-line coded H.264 video. In our scheme, a feedback channel between the sender and the receiver is required, as shown in Fig. 5. The feedback channel is used to report any detected errors to the sender from the receiver. Accordingly, the sender adjusts the encoded bit stream of subsequent video frames by avoiding referencing the corrupted parts.

For each macroblock in the bit stream, we encode multiple alternative versions for it. These alternatively coded versions are encoded using different predictions from each other, and from that of the original coded version. They are also coded in a way such that reconstructions of these alternatively coded versions are identical or very close to that of the original version. The original bit stream or the original coded version of the macroblock is transmitted by the sender when no error is detected. But transmission errors may occur during the transmission, as explained above, if the prediction of the originally coded version is corrupted by errors, the macroblock will not be correctly reconstructed at the decoder side and the error will start to propagate. But if one of the alternatively coded versions is not affected by any errors, as shown in Fig. 6 we may substitute the original version with it and transmit it to the receiver, and the decoder will be able to reconstruct the macroblock correctly. In our scheme, the sender identifies macroblocks that are affected in the subsequent bit stream according to the error feedback and replace them with their proper alternatively coded versions, and thus error propagation is stopped or largely reduced in the subsequent frames.

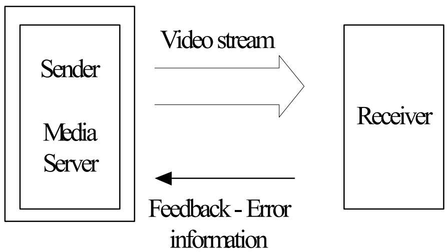


Fig. 5: The interaction between the sender and the receiver.

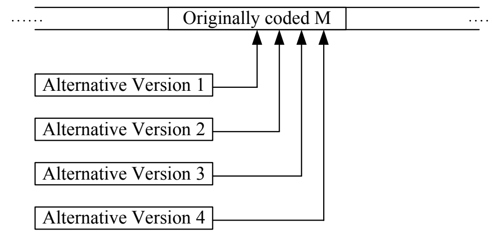


Fig. 6: Bit stream replacement.

3.2 Alternative SP/SI Encoding Scheme

In order to make the above error resilient transmission scheme work effectively, there are several major requirements on the coding of alternative macroblocks. First, the reconstructions of alternative macroblocks have to be identical or very close to the originally coded version. Otherwise, mismatch is introduced between the encoder and the decoder. Second, the encoding of alternatively coded macroblocks should not influence the original coded version such that, if no error occurs during transmission, the bit stream being transmitted should have the same bit rate and quality as being coded by a normal H.264 encoder. Finally, the bit rate overhead introduced by these alternatively coded macroblocks should be small and it should not increase the original bit rate too much, as the bandwidth is still a critical issue in video transmission.

3.2.1 Alternative SP Macroblock Encoding

The original bit stream to be transmitted without errors is exactly the same as being coded by a normal H.264 encoder. We call them originally coded versions. The alternatively coded macroblocks introduced in our scheme are encoded using secondary SP/SI macroblock coding. The diagram for the coding of such macroblocks is shown in Fig. 7.

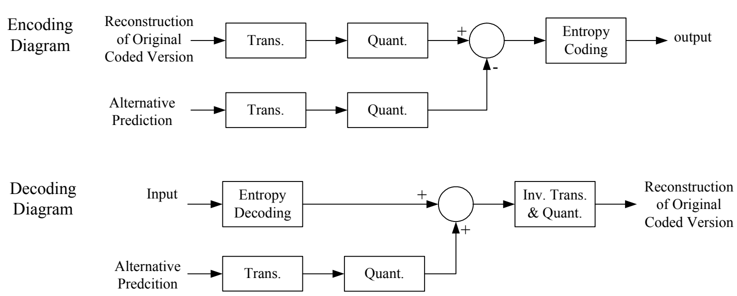


Fig. 7: The encoding/decoding diagram of alternative SP/SI macroblocks.

There are two major differences between the encoding of alternative SP/SI macroblocks and the normal inter/intra macroblocks encoding process. First, the reconstruction value of the originally coded version is being coded instead of the raw video data. Second, the transformation and quantization are done before calculating the difference, as explained previously. This is done so that different predictions being used will not influence encoding and decoding of the original reconstruction.

As the original coded versions are not encoded into primary SP macroblocks, in order to achieve identical or very close reconstruction as the original version, the quantization level has to be chosen properly such that few quantization errors will be introduced in the process. It is a trade-off between how close the reconstruction values are to the original version and the bit rate overhead introduced by using lower quantization levels. This issue will be discussed in detail in Sec. 3.3.

Multiple alternative predictions are used to generate different alternatively coded versions for the macroblock. These predictions have to be ‘good’ predictions in order to improve the coding efficiency of alternatively coded macroblocks, which means that the predictions should match the macroblock well. Although the ‘best’ prediction is usually being used by the originally coded version, it is still possible to generate different well matched alternative predictions for a given macroblock. In our previous work [2,3], we proposed two different methods to generate such predictions effectively.

3.2.2 Alternative Encoding with Multiple Reference Frames

H.264 allows multiple reference frames to be used for motion compensation. When multiple reference frames are allowed, motion estimation is performed in each available reference frame to find the best prediction. Suppose that we fix the block size for motion estimation to be 16x16, the encoder will find one best matching 16x16 block in each reference frame. As shown in Fig. 8, the five best matching blocks ($P_0 \sim P_4$) for macroblock M in frame A are determined in the five reference frames, respectively. Then, macroblock M is coded into five different versions using each of the matching blocks as predictions. The one provides the lowest R-D cost is chosen and saved as the final coded version for macroblock M and its reconstruction will be saved in the reference buffer to be used for future frames.

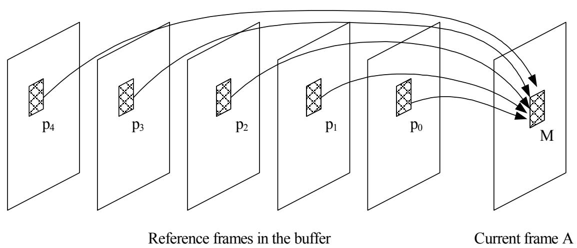


Fig. 8: Prediction in each reference frame.

Suppose that P_0 is chosen as the final prediction and, during the transmission of the encoded video stream, P_0 is corrupted due to transmission errors. Upon receiving frame A, the decoder is not able to reconstruct macroblock M correctly with a corrupted prediction. However, it is not as likely that all the other predictions $P_1 \sim P_4$ are corrupted at the same time, therefore these predictions can serve as very good alternative predictions in our error resilience scheme.

Therefore, we can modify the encoder so that it saves all of the five predictions of macroblock M during the encoding process instead of saving only the final choice. And four alternatively coded SP macroblocks can be generated using the alternative predictions $P_1 \sim P_4$. These alternative predictions are expected to match macroblock M well because they are the best matching results for each reference frame after motion search. Such alternatively coded SP macrobocks can be generated for each macroblock in the bit stream. During the transmission of the video stream, the sender always sends the originally coded macroblocks unless it receives any error information from the receiver. If it does receive any information regarding a transmission error, the sender checks the part of the video stream that is going to be transmitted to see whether there are any macroblocks that may be affected by this error. If so, an alternative version of this macroblock is chosen and transmitted instead of the original one. As long as not all five of the predictions for this macroblock are affected by transmission errors, the sender can find a suitable coded version and send it to the receiver to have the macroblock correctly decoded. Thus, the impact of the transmission error will not propagate.

The alternatively coded SP macroblocks described above can also be coded in real-time during transmission instead of being coded off-line with the original bit stream. In that case, sender will choose one alternative prediction and encode the SP macroblock when needed, such that the storage needed for the bit stream is reduced.

3.2.3 Alternative Encoding with Concealed Reference Frame

The above alternative SP macroblock encoding scheme works well when multiple reference frames are available for prediction. However, in some applications, the reference buffer size may be restricted to one. In this case, the above method will fail since no alternative SP predictions can be generated. Even if the encoder does allow multiple reference frames, the above method may not work well sometimes when there is only one frame available for reference. For example, the second frame that immediately follows an I frame could only use one frame for prediction. Sometimes, alternative predictions may generate a large residual and then introduce a high bit rate increase if being transmitted (which may happen during scene changes). Under these situations, we develop the following technique to enhance the proposed error resilient scheme utilizing the error concealment tools.

Although the error concealment tools implemented at the decoder cannot be used to completely stop error propagation, such tools still help reduce artifacts of corrupted video frames and improve the PSNR of received video. Corrupted video frames concealed by the decoder are stored as references for subsequent frames. Since these concealed frames are different from the reference frames used at the encoder, the mismatch is introduced and the subsequent frames become vulnerable to propagation errors. However, if the encoder obtains the exact concealed frames based on the error feedback information and the knowledge of error concealments tools used by the decoder, such a mismatch can be eliminated using the following coding methods.

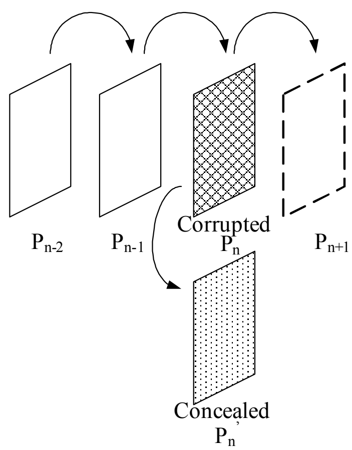


Fig. 9: Error concealment of the decoder.

Fig. 9 shows the video frames received at the decoder during the streaming of the encoded H.264 stream. Suppose that a transmission error occurs in frame P_n and corrupts the received frame. Upon detection of the transmission error, the decoder conceals the corrupted frame using error concealment tools and stores the concealed P_n' as a reference frame in its buffer. At the same time, the error information is sent back to the sender using the feedback channel. After receiving the feedback for frame P_n, the sender is able to generate the concealed frame P_n' to conceal motion vectors and other parameters exactly the same as that of the decoder using the same error concealment tool.

By calculating the dependency between frames P_n and P_{n+1}, the sender can identify macroblocks in frame P_{n+1} that are affected by the mismatch between P_n and P_n'. An alternative SP macroblock can be generated in real time for it using predictions from the concealed reference frame P_n' with the same method as shown in Fig.7. It is expected that the concealed reference frame can also generate a well matching prediction for the macroblock. This SP macroblock will then be used to replace the originally coded version in the bit stream and sent to the receiver to stop error propagation.

Both of the two approaches described above are able to effectively generate alternative coded SP macroblocks to eliminate or largely reduce error propagation. They can be combined and used together to make the scheme work more robustly. A third option, alternative SI macroblocks can also be used when both of the two approaches fail. The alternative SI macroblocks are coded without any difference from SP macroblocks, only that the alternative predictions used are generated from intra prediction instead of inter prediction.

3.3 Adaptive Alternative SP/SI Macroblock Coding

As mentioned above, the selection of quantization levels to encode SP macroblocks is critical in our proposed error resilience scheme. It is a trade-off between how close the reconstruction values are to the original version and the bit rate overhead introduced using lower quantization levels.

If the quantization level is chosen too large, significant mismatches may exist among the reconstructions of alternative SP/SI macroblocks and that of the originally coded version. As shown in Fig. 10, the originally coded version of macroblock M (M_0) is replaced by one of its alternative versions (M_p). As macroblock M may be referenced by other macroblocks in subsequent frames and the reconstruction of M_0 is used for their prediction, if there is significant mismatch between the reconstruction of M_0 and M_p, the subsequent macroblocks may not be reconstructed correctly and it will introduce errors in subsequent frames.

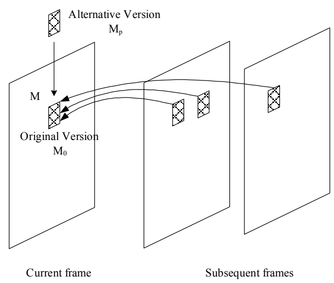


Fig.10: Influence of mismatch on subsequent frames.

In our previous work [2,3], the quantization level is fixed to ensure a pseudo-lossless compression of the reconstruction values. The same low quantization level is used for the alternatively coded versions of each macroblock. The approach is able to guarantee that no significant mismatch is introduced among the reconstructions of different alternative SP macroblocks and the original coded macroblock. But it is not efficient because the quantization level is usually very low and will introduce a relatively large bit rate overhead for the alternatively coded SP macroblocks. Since the number of macroblocks being replaced into these SP macroblocks is quite limited, and usually very low in most transmission scenarios, the overall bit rate overhead to the transmitted bit stream is not very significant. However, the scheme can be further improved if the SP macroblocks are encoded more efficiently and the bit rate overhead is further reduced.

In this work, we propose an adaptive method to select the quantization levels. The decision is made adaptively according to the importance of each macroblock to be replaced. For example, if a macroblock is not being referenced by any part of subsequent frames, it is considered not important, as the mismatches of its different reconstructions do not affect subsequent frames and thus we can use larger quantization levels to encode its alternative SP/SI macroblocks. For instance, all the macroblocks in a frame immediately before an intra frame can be viewed as not important.

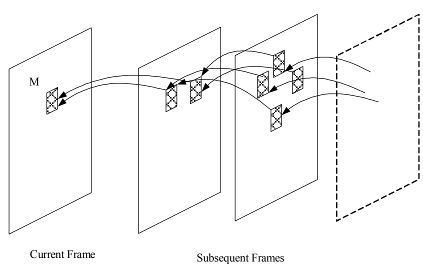


Fig. 11: Reference chain of macroblock M

The importance of the macroblock is measured by the number of macroblocks in its reference chain. As shown in Fig. 11, macroblock M is referenced by a couple of blocks in the next frame, while they are being referenced by a number of blocks in their subsequent frames, and such dependencies may continue to grow. We call this dependency the reference chain of macroblock M. We count the number of 4x4 blocks existing in this reference chain of macroblock M within a certain number of frames (*e.g.* 5 frames). As the original bit stream is coded off-line, such information can be easily obtained and it is used as the importance measurement for macroblock M.

Therefore, according to different levels of importance, we may use different quantization levels accordingly. The level classification and thresholds can be defined to fit the needs of different applications. A simple strategy used in our experiments is to define three different levels: most important, important and not important. If the number of 4x4 blocks in the reference chain of a macroblock is less than 16 within the five-frame interval, it is classified as not important and its alternative version will be encoded with the same quantization level as the original bit stream. If the number is above 80, it is considered important and the quantization level used for encoding its alternative version is chosen to ensure pseudo-lossless compression. The actual quantization step is not necessarily 1 in order to reduce the quantization error to a negligible number, instead it could be much higher. This is because that the reconstruction value of the originally coded version has gone through a certain quantization and de-quantization process, and it is friendly to another quantization process. The macroblocks fell into the middle level can be coded using a quantization level in between (chosen as +8 to lossless level as optimized in our experiments). The overall coding efficiency of the alternative SP/SI macroblocks can be improved effectively using the above approach.

3.4 Implementation of the Proposed Error Resilience Scheme

The proposed error resilience scheme can be implemented and functioning effectively in any applications that require error resilient transmission of off-line coded H.264 video. The original bit stream is coded in the same way as a normal H.264 encoder. The alternative SP/SI macroblocks can be encoded on-line or off-line, it depends on the requirements of the application. Only a few modifications need to be made to the H.264 encoder in order to generate these alternative SP/SI macroblocks since the SP/SI encoding scheme is already included in a standard H.264 encoder.

A number of important functionalities are implemented at the sender side. The sender is responsible of identifying macroblocks that are corrupted or affected by errors, selecting proper alternatively coded versions and replacing the macroblocks in the bit stream. When the network delay is relatively low, the sender receives the error feedback immediately after the receiver detects an error and it is able to replace the macroblocks in the next frame immediately. When the network delay is higher, the scheme has to be adjusted accordingly so as to work effectively.

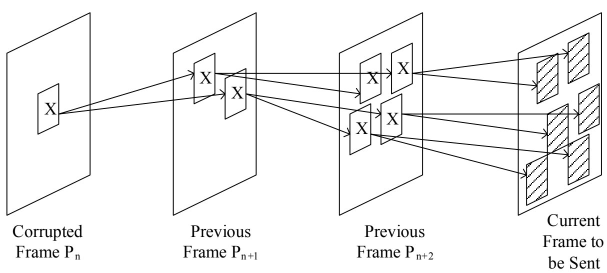


Fig. 12: Long delay scenarios

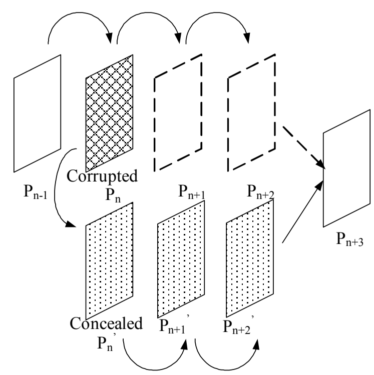


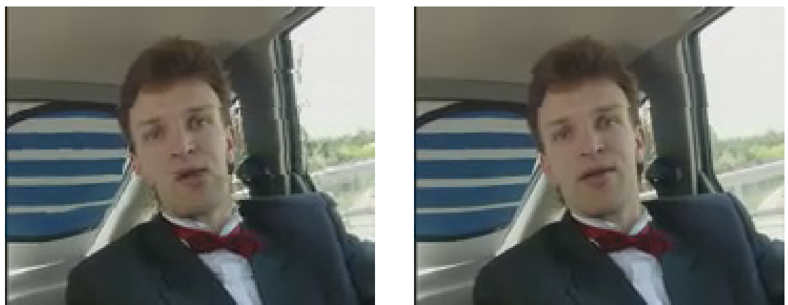
Fig. 13: Long delay scenarios with concealed reference frames

As shown in Fig. 12, when the sender receives the feedback regarding the transmission error occurring at frame P_n, two subsequent frames P_{n+1} and P_{n+2} have already been transmitted to the receiver and it is expected that errors will propagate in these two frames. In order to stop the error propagation, the sender first identifies macroblocks in these two frames that have been affected by transmission errors, and then determines which macroblocks in the current frame may be affected by them. Finally, the sender replaces these macroblocks (the shaded macroblocks as shown in the figure) in the current frame with one of their suitable alternative coded versions. If the alternative prediction using concealed reference frame is being used, the sender also generates concealed versions of the intermediate frames. As shown in Fig. 13, the sender still generates the concealed version of P_n as P_n'. Furthermore, based on P_n' and the coding parameters, it also generates frame P_{n+1}' and P_{n+2}' that are identical to the reconstructed frames of P_{n+1} and P_{n+2} at the decoder. Then, using these re-generated reference frames, the sender encodes the affected macroblocks in P_{n+3} into SP macroblocks and sends them in the output bit stream to stop error propagation. By following this method, our scheme works effectively under long network delays.

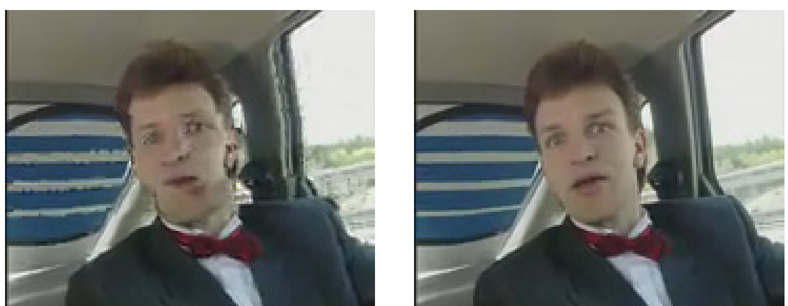
In summary, our proposed scheme is able to effectively eliminate or largely reduce the propagation errors using the alternative SP/SI macroblock coding scheme and the transmission scheme described above. When no error occurs, the original bit stream is transmitted and no bit rate overhead is introduced. When the transmission error occurs, as the alternative SP/SI macroblocks are encoded with well-matched predictions and with adaptively selected quantization levels, their bit consumption is largely reduced. As the number of these alternative macroblocks to be transmitted is limited, the overall bit rate overhead is small.

4. EXPERIMENTAL RESULTS

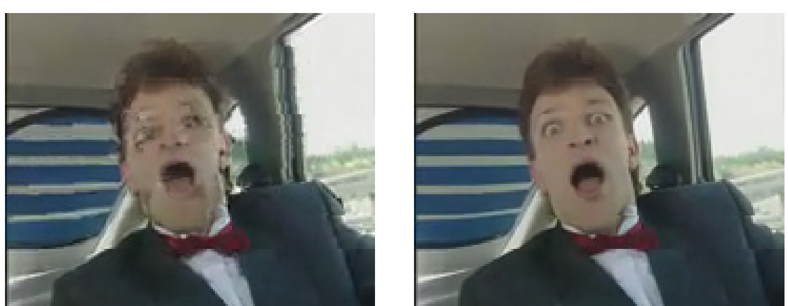
To evaluate the performance of the proposed scheme, we tested it under different scenarios with different channel characteristics. In order to demonstrate its effectiveness in real-world applications, we simulated the network packet loss conditions by generating error patterns similar to the practical ones. For instance, burst errors often occur in wireless channels. Such burst errors may corrupt one whole slice in a video frame or even more. We simulated such burst errors by generating the corresponding error pattern. The experiment results are reported below.



(a) The 10th frame after error starts to occur



(b) The 30th frame after error starts to occur



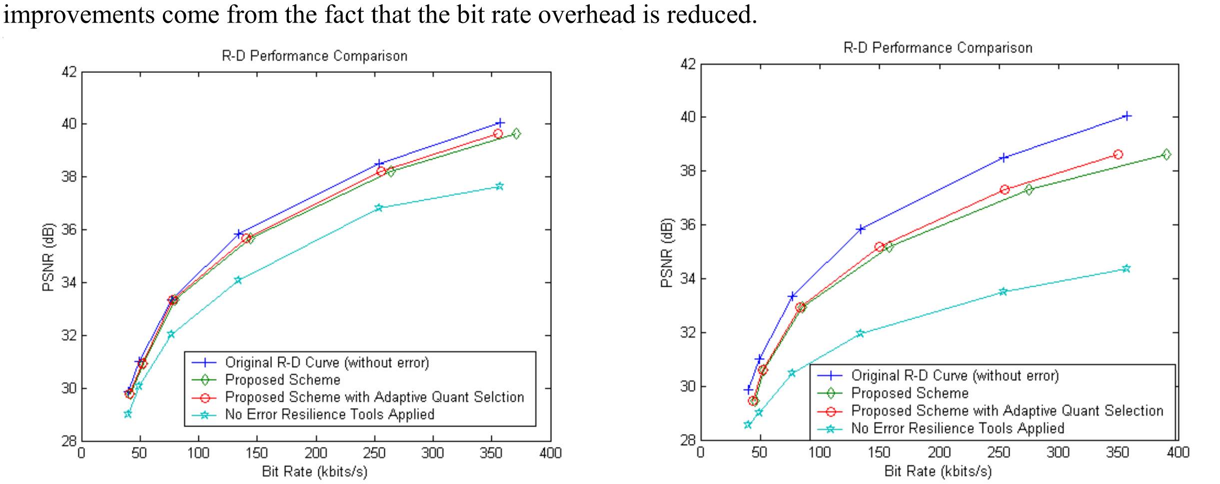
(c) The 60th frame after error starts to occur

Fig. 14: Comparison of reconstructed frames with (right) and without (left) the proposed scheme

First, we want to demonstrate the error resilience performance of our proposed scheme. As shown in Fig. 14, the reconstructed video frames at the decoder side with and without our proposed scheme are compared. In our test, the error concealment tools described in [6] and implemented in the reference codec are used to conceal the corrupted frames. The burst error length is one slice and the error rate is 5% of all slices. The sequence is encoded into 'IPPPPP…' pictures.

As shown in the results, our scheme provides an excellent error resilience performance, even for the 60th frame after the error starts to occur, we still do not observe any significant artifacts in the reconstructed video frame. On the other hand, the reconstructed video frame without our scheme being implemented has been largely distorted.

In our previous work, we have shown that our proposed scheme outperforms other error resilience tools such as intra refreshing. In our following experiments, we compared the R-D performance of our scheme with and without using adaptive decision on quantization levels for alternative SP/SI macroblock coding. The same slice loss pattern is simulated while FMO (flexible macroblock ordering) is enabled to improve the performance of error resilience. The H.264 reference codec version 7.6 and Foreman QCIF sequence is used in the experiments. As shown in Fig. 15, our scheme with adaptive quantization level selection yields a better R-D performance than before, most of the improvements come from the fact that the bit rate overhead is reduced.



(a) Error Rate = 1%

(b) Error Rate = 5%

Fig. 15: R-D performance comparison of the proposed scheme with and without adaptive quantization level selection.

5. CONCLUSION

The proposed scheme is able to provide high error resilient performance under different scenarios as shown in the experiments. As most multimedia applications demand effective quality control while the quality requirement becomes higher, the proposed scheme offers a great tool for H.264 coded video streams to maintain the high quality level in an error-prone environment. The bit rate overhead of our scheme is small. The adaptive quantization level selection scheme helps further reduce the bit rate overhead. It does not result in any overhead in bit rates when there is no transmission error, since the proposed scheme does not require any significant syntax changes for the H.264 standard. It aims to work for off-line H.264 encoders. Our scheme can work effectively even when long or varying transmission delays exist. It is specifically designed for H.264, and can be adapted to any other coding standards that provide SP/SI or similar coding features.

ACKNOWLEDGEMENTS

The research has been funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, under the Cooperative Agreement No. EEC-9529152. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

REFERENCES

1. ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, Document JVT-G050, 8th Meeting: Geneva, Switzerland, May 2003.
2. X. Zhou, W.-Y. Kung and C.-C. J. Kuo, “Error resilient H.264 video with SP/SI coded macroblocks”, *The 14th International Packet Video Workshop*, Irvine, CA, December 2004.
3. X. Zhou, W.-Y. Kung and C.-C. J. Kuo,, “A Robust H.264 Video Streaming Scheme for Portable Devices”, *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.
4. International Standard Organization, “Information Technology-Coding of Audio-Visual Objects, Part2---Visual”, ISO/IEC 14496-2.
5. Y. Wang, S. Wenger, J. Wen and A. K. Katsaggelos, “Error Resilient Video Coding Techniques”, *IEEE Signal Processing Magazine*, July 2000, pp. 61-82.
6. Y. K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta and M. Gabbouj, “The error concealment feature in the H.26L test model”, *IEEE International Conference on Image Processing*, September 2002, pp. 729-732.
7. T. C. Wang, H. C. F and L. G. Chen, “Low-delay and error-robust video transmission for video communcations”, *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 12, pp 1049-1058, December 2002.
8. E. Steinbach, N.Farber and B.Girod, “Standard compatiable extension of H.263 for robust video transmission in mobile environments”, *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 7, pp 872-881, December 1997.
9. W. Wada, “Selective recovery of video packet loss using error concealment”, *IEEE J. Select. Areas Commun*., vol. 7, pp. 807-814, June 1989.
10. M. Karczewicz and R. Kurceren, Document VCEG-L27, “A Proposal for SP frames ”, 12th Meeting: Eibsee, Germany, 9-12 January, 2001
11. R. Kurceren and M. Karczewicz , Document VCEG-O47, “New Macroblock Modes for SP-frames”, 15th Meeting: Pattaya, Thailand, 4-6 December, 2001
12. P. Haskell and D. Messerschmitt, “Resynchronization of Motion - compensated Video Affected by ATM Cell Loss”, *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* vol.3, pp.545-548, March 1992.
13. J. Liao and J. Villasenor,“Adaptive Intra Update for Video Coding over Noisy Channels”, in Proc. *IEEE Int. Conf. On Image Processing (ICIP),* vol.3, pp. 763-766, Sept.1996.
14. K. Imura and Y. Machida, “Error Resilient Video Coding Schemes for Real-time and Low bitrate Mobile Communications”, *Signal Processing: Image Communication*, 14(1999), pp.519-530.
15. C. S. Park, J. Ye and S. U. Lee, "Lost Motion Vector Recovery Algorithm", *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1994, pp. 229-232
16. M.-J. Chen, L.-G. Chen, and R.-M. Weng, "Error Concealment of Lost Motion Vectors with Overlapped Motion Compensation", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 560-563
17. Z. Wang, Y. Yu, and D. Zhang, "Best Neighborhood Matching: An Information Loss Restoration Technique for Block-Based Image Coding Systems", *IEEE Trans. on Image Processing*, vol. 7, no. 7, pp. 1056-1061
18. W.-Y. Kung, C.-S. Kim, C.-C. J. Kuo, "Edge recovery and selective directional interpolation for spatial error concealment", *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP),* April 2003.
19. L.-W. Kang and J.-J. Leou, “An Error Resilient Coding Scheme for H.264 Video Transmission Based on Data Embedding”, *IEEE International Conference on Acoustics, Speech, and Signal Processing,* May 2004.
20. M.H. Chen, Y. He, and R.L. Lagendijk, “A Fragile Watermark Error Detection Scheme For Wireless Video Communications”, *IEEE Transactions on Circuit and Systems for Video Technology*, 2004
21. Video and image compression resources and research (VCODEX)," H.264 / MPEG-4 Part 10 Tutorials", *http://www.vcodex.com/*
22. M. Karczewicz and R. Kurceren,"The SP- and SI-Frames Design for H.264/AVC", *IEEE Trans. on circuits and systems for video technology*, vol. 13, no. 7, pp.637-644