Error Resilient H.264 Video with SP/SI Coded Macroblocks

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

Department of Electrical Engineering and Integrated Media Systems Center
University of Southern California, Los Angeles, CA 90016
E-mails: xiaosonz@usc.edu, weiyingk@ieee.org, cckuo@sipi.usc.edu

***Abstract*—A novel H.264 error resilient scheme aiming at stopping or reducing error propagation based on the SP/SI macroblocks is proposed in this work to maintain the quality of transmitted video through erroneous channels. In the proposed scheme, for each encoded macroblock, we encode additional predicted versions for it using different reference frames or different prediction methods and save them as SP/SI macroblocks. During transmission, these SP/SI macroblocks are used to replace the original macroblocks in the output video stream if they are affected by previous errors detected by the receiver. The way to encode these SP/SI macroblocks ensures that such a replacement will not cause any mismatch at the decoder side. It is confirmed by experimental results that the proposed scheme is effective in reducing error propagation so as to enhance the error resilient capability of H.264 video. This scheme introduces a small amount of overhead in the bit rate only when there are transmission errors, and does not have an overhead when no error occurs.**

I. INTRODUCTION

H.264 [1] (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. Its capability to provide high-quality compressed video at a broad range of bit rates makes it a competitive candidate for various multimedia applications. Most applications require the transmission of a compressed video stream through certain communication networks, where transmission errors often cause significant PSNR decrease and subjective quality degradation of received video. Thus, it is important to design an effective error resilient coding and transmission scheme for H.264 video that is able to maintain the high quality provided by this new standard in spite of transmission errors.

A. Overview of H.264 Standard

H.264 uses the hybrid block-based motion compensation and transform coding model as existing standards such as H.263 and MPEG-4 [2]. Furthermore, a number of new features and capabilities have been introduced in H.264 to effectively improve the coding performance [10].

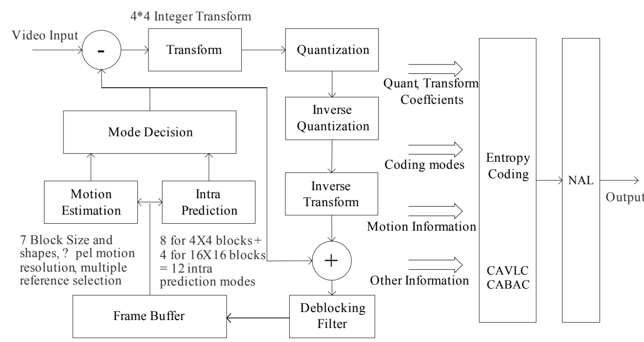


Fig. 1: The encoder diagram of H.264.

The encoding diagram of H.264 is shown in Figure 1. The standard provides more coding options for motion-compensated predictive coding. It allows more choices of block sizes to be used for motion estimation of each macroblock (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4 as shown in Figure 2). It also defines quarter-pel motion vector resolution in order to generate more precise predictions. Furthermore, more than one previous frame can be stored as reference frames for both P and B frames. Such flexibilities and new coding features improve coding efficiency substantially.

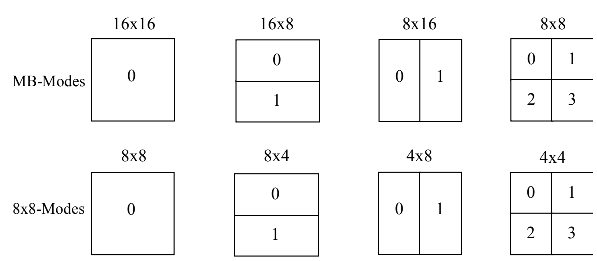


Fig. 2: Block sizes for motion estimation in H.264.

The standard also utilizes intra predictive coding to enhance the compression of intra-coded macroblocks and slices. Intra macroblocks are predicted from its neighboring blocks and only the prediction mode and resulting residuals are encoded. As shown in Figure 3, a typical intra 4x4 block can be predicted from its surrounding pixels, namely A ~ M, at the same time, the standard provides different ways that represent different directions to generate the intra prediction from these pixels, which is also shown in the figure. The direction that generates the best matching prediction may be chosen as the final prediction mode to encode the block.

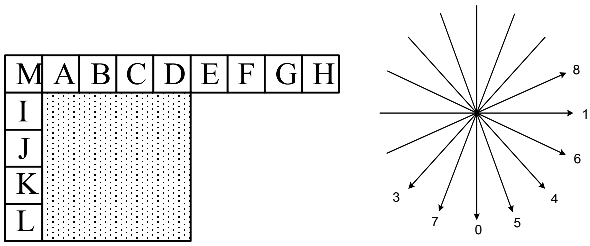


Fig. 3: Intra prediction modes for 4x4 blocks.

A number of other coding features and techniques are also introduced in H.264 to help provide 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.

B. Error Resilient Tools for H.264 and Our Proposed Scheme

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 destination, data packets may be lost or corrupted [3] 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 drift (propagation) errors. Various error resilient tools were introduced in the past to avoid and/or reduce the effect of error propagation, and some of them can be modified so as to work for H.264. However, the new coding features and very high compression ratio of H.264 create a lot of problems for these existing tools to function effectively.

One way to stop error propagation is to insert intra coded macroblocks in P or B frames randomly or with certain patterns. This is known as intra refresh [6~9]. This technique was shown to be effective for existing standards in certain cases. However, since intra macroblocks in H.264 are coded with intra predictions, the insertion of intra macroblocks may fail to stop error propagation since their neighboring blocks that were used to predict these macroblocks might be affected by transmission or propagation errors. ARQ based re-transmission schemes were proposed in [17] but they can only be used when long end-to-end delay is tolerable, which is not the case for most video transmission applications. Data embedding [16] is another method that was modified to work for H.264. It can hide important information within compressed video data, but the resulting PSNR degradation may not be suitable for the high quality requirement. Data partitioning and FMO (flexible macroblock ordering) [11] are supported in the H.264 standard to enhance error resilience when encountering burst errors. Besides, spatial and/or temporal error concealment tools [12] can also be implemented in the H.264 decoder to reduce artifacts caused by error propagation. However, their effectiveness largely depends on the video characteristics and transmission error patterns, and they are not designed to stop or eliminate error propagation.

Our scheme aims at completely stopping and/or largely reducing propagation errors. It can maintain a very high quality level of decoded video. Its bit rate overhead is small as compared with other error resilient tools when there are transmission errors, and there will be no overhead when no error occurs. Our scheme is able to function effectively with different transmission delays. It does not require real-time encoding and the additional computational complexity is very low for both the encoder and the decoder. Our scheme borrows ideas from the SP/SI picture concept [4] and makes some modification so that it serves as an error resilient coding tool at the macroblock level. Our scheme is designed specifically for H.264 and can be adapted to any other coding standards that provide SP/SI or similar coding features.

C. Paper Organization

The rest of this paper is organized as follows. In Section II, a brief introduction to SP/SI picture coding for H.264 is given. In Section III, the proposed scheme is discussed in detail, where we describe how to encode the SP/SI macroblocks for error resilience and illustrate how to replace the original bit stream with these alternatively coded macroblocks. We also discuss the interaction between the sender and the receiver and explain how this scheme works with different transmission delays. In Section IV, we present some experimental results to show the performance of the proposed scheme. We also compare its performance with some other error resilient tools such as intra refresh. Concluding remarks and future research directions are provided in Section V.

II. SP/SI PICTURE CODING OF H.264

SP and SI picture coding for the H.264 video coding standard was first introduced by Karczewicz and Kurceren in [4] and [5]. 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 Figure 4, 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 (PA1~PA5 and PB1 ~ PB5) 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 PB3 instead of PA3, the decoder is not able to reconstruct PB3 correctly since the reference frame used to encode PB3, i.e. reconstructed PB2, is not available in the reference frame buffer of the decoder and the decoder may attempt to use reconstructed frame PA2 as the reference frame to decode PB3, which results in significant errors.

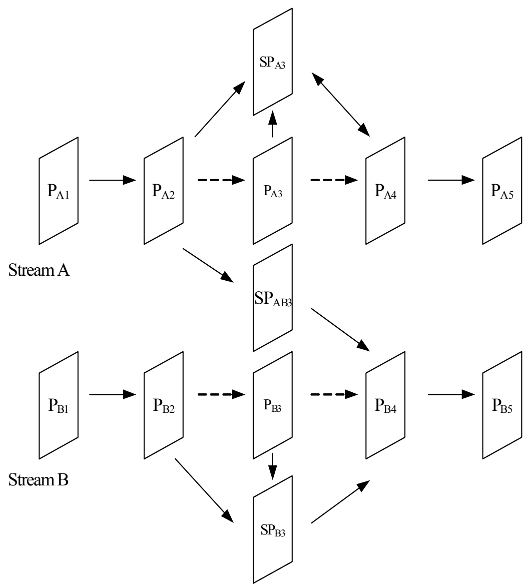


Fig. 4: Stream switching using SP pictures.

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 SPA3 and SPB3, which are called primary SP pictures. When the bit stream is being switched, a secondary SP picture SPAB3 that is coded using PA2 as the reference frame will be transmitted instead of SPA3 or SPB3. The way to encode SPAB3 ensures that an identical reconstruction as that of the SPB3 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 SPBA3 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-quantize without loss at certain quantization level QSP, 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 SPAB3 is the reason that its reconstruction can be identical to the reconstruction of another coded version – the primary SP picture SPB3 in our example. Figure 5 shows the simplified encoding and decoding diagrams for such macroblocks. As shown in the diagram, the exact reconstruction of the primary SP picture (SPB3) is being coded instead of the original video content. The prediction to be used is obtained from PA2 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 QSP set by the primary SP picture coding as mentioned before. By decoding such a 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 P1 in the encoder and P2 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 QSP, 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.

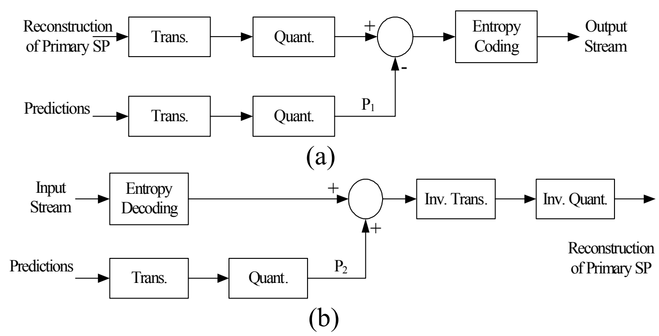


Fig. 5: (a) Encoding and (b) decoding of SP macroblocks in secondary SP pictures.

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 Figure 5, 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.

III. PROPOSED ERROR RESILIENT SCHEME BASED ON SP/SI MACROBLOCKS

A. *Motivations*

Figure 6 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 in real-time and sends the output to display devices.

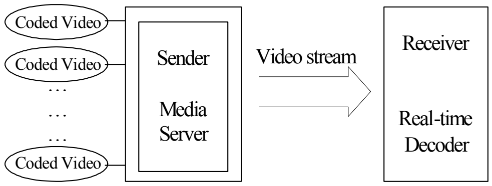


Fig. 6: Video transmission systems.

One common problem for the system is that transmission errors cannot be avoided because most existing networks do not provide a guaranteed QoS, especially for wireless networks. The simple retransmission scheme provided by the network may not help since it will introduce unwanted delay for video, which is often not acceptable by these applications. Figure 7 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 errors will keep going, which is usually referred to as error propagation.

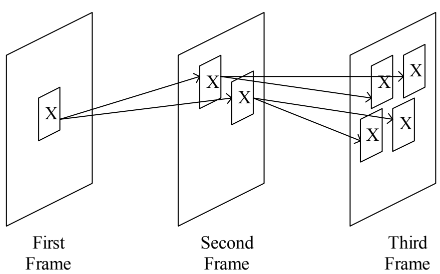


Fig. 7: An example of error propagation.

However, if no macroblocks in the next or further following frames are coded by referencing 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 for predictions when encoding future frames. Ideas and schemes similar to this were proposed for previous standards [13~15]. By utilizing 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 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.

In this section, we propose such an error resilient scheme to work for off-line H.264 encoders. The scheme is proven to be effective in eliminating or largely reducing propagation errors. It will be presented in the next subsection.

B. Alternative SP/SI Encoding

H.264 provides a lot of flexibilities for the encoder. A general H.264 encoder explores a large set of possible coding options, features and their combinations in order to select the best-fit coding mode for each macroblock during the encoding process. For example, a macroblock within a motion-compensated frame can be predicted by any of the available reference frames in the reference buffer. The block sizes used for motion estimation can be chosen from any of the possible combinations as depicted in Figure 2. Furthermore, it can also be coded into an intra macroblock if necessary.

As a matter of fact, in the RDO (rate-distortion model) of the current H.264 reference encoder developed by JVT, each macroblock is coded into every possible coding mode first. Then, the R-D costs of these different coded versions are compared. Finally, the best mode is selected as the final coding choice. Therefore, virtually each macroblock is coded into a number of different versions.

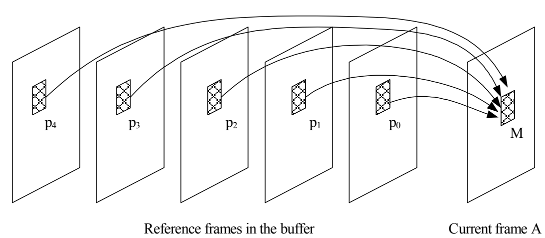


Fig. 8: Prediction in each reference frame.

1) Alternative SP Macroblock Encoding with Multiple Reference Frames

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 Figure 8, the five best matching blocks (P0 ~ P4) 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.

Suppose that P1 is chosen as the final prediction and, during the transmission of the encoded video stream, P1 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, if P1 were not chosen as the final prediction, the situation would be different. As discussed earlier, an off-line encoder does not have any information about existing errors at the decoder side so that it is not able to decide which prediction to use to avoid error propagation.

However, we can modify the encoder so that it saves all of the five coded versions of macroblock M during the encoding process instead of saving only the final choice. We call the final coded version chosen by the encoder the originally coded version and the four others as alternatively coded versions. Under the error-free channel condition, the bit stream to be transmitted only consists of the originally coded version, which has no difference from the bit stream generated by a typical encoder. When the bit stream is under transmission in an erroneous channel, the sender may get feedback from the receiver, which contains information regarding previous transmission errors before sending out the bits for macroblock M. If such information indicates that macroblock M's original prediction is affected by errors, the sender may send one of the alternatively coded versions to avoid error propagation.

However the simple idea as stated above has its own problem. That is, the four alternatively coded versions may have different reconstruction values, which may be different from that of the originally coded version, too. Suppose that some macroblocks in future frames are coded using macroblock M or part of it as their reference, the reconstruction of the originally coded version of macroblock M will be used for motion estimation. Therefore, if the sender sends out an alternatively coded version, it will cause the mismatch between the encoder and the decoder, and the future frames may not be correctly decoded as a result.

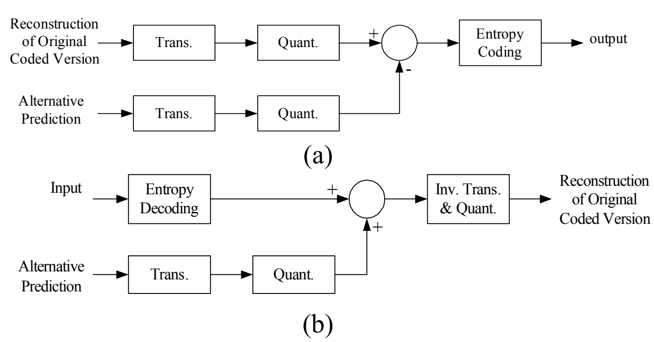


Fig. 9: (a) Encoding and (b) decoding for alternative SP/SI macroblocks.

To address this problem, we introduce a special encoding method to encode the macroblock into different alternative versions but with the same or very close reconstruction value to the reconstruction value of the originally coded version. In this case, no matter which version the sender decides to send out, the same or very close reconstruction can be obtained at the decoder side and there will be no significant mismatch afterwards. This special encoding method can be realized by utilizing the SP/SI encoding. Figure 9 shows a simplified encoding and decoding diagram for such alternative SP macroblocks.

As shown in the figure, the reconstruction value of the originally coded version (selected by the encoder) is being coded instead of the raw video data. The best matching blocks in each of the reference frames (except the one selected by the encoder as the final prediction) are used as the alternative predictions to encode the macroblock. As discussed in Section II, such an encoding method ensures that these different predictions will not introduce any mismatch. Furthermore, the quantization step size is selected in such a way that there will be no significant quantization error introduced during the quantization and de-quantization process of the reconstruction value. One way to realize it is to encode the original version as a macroblock of primary SP pictures, and make the quantization level in the above encoding process identical to the one used in primary SP picture coding, just as described in Section II. However, in this way, the whole video stream needs to be coded into primary SP pictures, which degrades the coding efficiency of the originally coded video stream. Therefore, we will not encode those primary SP pictures. As we do not expect a large number of alternative SP macroblocks to be transmitted during video transmission, we can sacrifice their coding efficiency a little bit by allowing lower quantization levels. As the reconstruction values are derived from decoding the compressed video stream and they have gone through certain quantization and de-quantization processes, it is not difficult to find a relatively small quantization level for them to eliminate the quantization error or reduce it to a negligible level. In this case, the above encoding and decoding process will not introduce any significant mismatch, and the decoder is able to generate the same or a very close reconstruction as the originally coded version.

By using this method, the four alternatively coded versions can be generated and saved as SP macroblocks in the above example. 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 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 transmission errors will not propagate.

The above alternative SP macroblock encoding scheme will work well when multiple reference frames are available for prediction. Our experimental results show that it offers excellent error resilient performance. When no transmission error occurs, only the original coded stream will be transmitted so that there is no bit rate overhead. When the transmission error occurs, since the number of alternative macroblocks being transmitted is quite limited, it does not introduce a significant bit rate overhead either.

2) Alternative Encoding with Only One Reference Frame

In some applications, the reference buffer size may be restricted to one in the encoder. In this case, the above method will fail since no alternative SP macroblocks 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 from distant reference frames 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 techniques to enhance the proposed error resilient scheme.

One method to generate alternatively coded macroblocks in this situation is to encode them into SI macroblocks. Instead of using any reference frames for prediction, alternative SI macroblocks are generated using intra predictions from its neighboring blocks. The intra predictions generated are used as alternative predictions as shown in Figure 9. Similarly, multiple alternative SI macroblocks can be generated using different intra prediction modes. We call this method "SI refresh"

One problem with the SI refresh method is that the alternative SI macroblocks are vulnerable to transmission errors occurred in the current frame. Since the sender may not get the information of such errors in time, the alternative SI macroblocks it sends out may not be constructed correctly at the decoder side so that they are not able to stop error propagation. But as long as the error rate remains low, this method still has very good error resilient performance since the above problem only occurs at a very low probability.

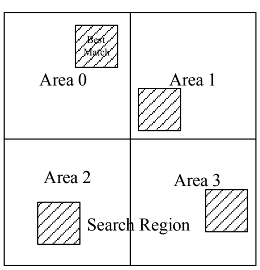


Fig. 10: Motion search for alternative matching blocks.

Another way to generate multiple alternatively coded macroblocks is to find multiple predictions or matching blocks for the macroblock within a single reference frame. These matching blocks or predictions should not overlap with each other so that they will not be affected by the same transmission errors. One way to find these alternative matching blocks is illustrated in Figure 10.

As shown in the figure, the motion search region is being divided into four sub-regions, motion estimation is performed in each of sub-regions to find a best matched block. As a result, four blocks will be found in these four sub-regions, respectively. They will be used as the alternative predictions to encode the alternative SP macroblocks. The original coded version is still coded by searching the whole region.

In this way, multiple alternatively coded SP macroblocks can be generated. This method can work along with the SI refresh method, in which case both alternative SI and SP macroblocks are generated. Their cost will be taken into consideration when the sender chooses the one to replace the original coded version. The SP macroblocks have a higher priority since they are less vulnerable to transmission errors in the current frame. By this method, the proposed error resilient scheme will work well with only one reference frame available.

C. Sender/Receiver Interactions and Stream Replacements

As mentioned above, the proposed error resilient scheme requires interactions between the sender and the receiver during the video transmission. Actually, it needs no more than a error feedback channel from the receiver to report transmission errors to the sender.

Transmission errors (for corrupted video packets) can be detected by the H.264 decoder or the network mechanisms of the receiver. Once an error is detected, a feedback message is transmitted to the sender immediately. According to this feedback information and coding parameters, the sender determines which part of the video stream is corrupted and which part of the subsequent frames to be transmitted is affected by these errors. Such interactions are illustrated in Figure 11.

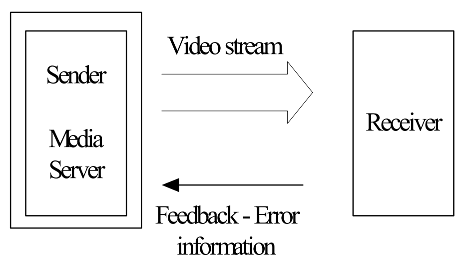


Fig. 11: The interaction between the sender and the receiver.

Whenever the sender attempts to transmit part of the video stream, it checks whether any macroblocks in it may be affected by previously detected errors. An index file generated during the encoding process contains necessary information regarding each macroblock. If the macroblock to be referred is detected in error by the receiver and fed back to the sender, the sender will look for suitable alternatively coded versions and use the best one available to replace the originally coded macroblock in the output bit stream. Again, as described earlier, such bit stream replacement will not have much negative impact on the rest of the video stream.

How soon the sender could get the feedback information will influence the performance of the above scheme. The receiver may take some time to detect errors upon receiving the video packets, and the feedback message leads to transmission delay as well. If such delay is very short (less than the frame interval), the sender is able to replace the affected macroblocks in the frame immediately next to the frame where the transmission error occurs and the error propagation phenomenon can be completely eliminated.

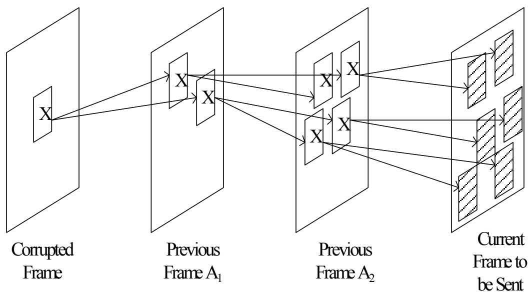


Fig. 12: Illustration of the proposed error resilient scheme in the presence of longer delay.

In cases where delay is longer, the scheme can work in a slightly different way. Figure 12 shows the corresponding scenario. When the information regarding the corrupted frame reaches the sender, two more frames (A1 and A2) have already been sent out and the transmission errors have started to 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 tries to replace these macroblocks (the shaded macroblocks as shown in the figure) in the current frame with one of their suitable alternative coded versions.

When the network bandwidth is sufficient, there is another way to make the proposed scheme work under longer delays. Upon receiving the feedback message, the sender selects and retrieves suitable alternatively coded macroblocks of the affected ones in the frames that have been sent out, and sends these alternatively coded macroblocks to the receiver directly. The receiver receives these data and is able to replace the affected macroblocks in its buffer as long as they have not been decoded yet. In this case, the possible error propagation will be stopped earlier.

D. Bit Rate Overhead and Computational Complexity

The alternatively coded SP/SI macroblocks generally consume more bits than the originally coded version. As a result, our scheme will introduce overhead in the bit rate when there exists transmission error and the bit stream replacement takes place. The main reason for the overhead is that the alternative predictions used are generally not the best prediction and the quantization level used is relatively smaller, which reduces the coding efficiency of alternative SP/SI macroblocks. Since H.264 uses the context-based entropy coding method based on previously obtained coding parameters, a number of additional bits are also inserted into the alternatively coded SP/SI macroblock to indicate the entropy coding parameters of the originally coded version to keep the consistency of the entropy coder. However, since these alternative macroblocks are only transmitted when they are used to replace part of the original bit stream, they will not influence the bit rate significantly.

The proposed scheme does not introduce much additional computational complexity, either. In most cases, the alternative predictions such as predictions in different frames are found in the normal encoding process, the only additional operation required in the encoder is to encode the SP macroblock using these predictions, which is not computationally intensive at all. At the decoder side, there is actually no additional computation needed since the complexity for decoding a SP/SI macroblock and a normal motion-compensated macroblock is about the same. The computational complexity of the stream replacement at the sender side during transmission is also quite low.

In summary, the proposed scheme uses different predictions from different reference frames for alternative SP coding if multiple reference frames are available. When only one reference frame is available, it can use intra prediction to generate multiple alternative SI macroblocks or uses different predictions from one single reference frame by a special motion estimation method to generate alternative SP macroblocks. A combined approach of these two can also be used. These alternatively coded macroblocks are used to replace the original ones in the output bit stream to stop error propagation during transmission. The examples given in this section consider the P frame coding. However, the same idea applies to B frames, too. The proposed scheme requires an error feedback channel from the receiver to the sender, and it is able to function effectively under different transmission delays. The bit rate overhead and the additional computational complexity introduced by the scheme are both low.

IV. 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. Case with Multiple Reference Frames

If multiple reference frames are available, multiple alternatively coded SP macroblocks using predictions from different reference frames can be generated and used to replace the original coded macroblock. The error resilient performance of the proposed scheme is expected to be very good since it is unlikely that the matching blocks in all reference frames are affected by transmission errors.

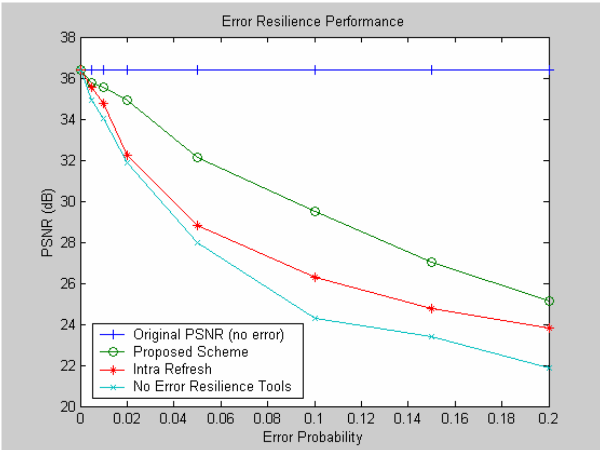


Fig. 13: Error resilience performance comparison for the case of multiple reference frames.

First, we observe the performance of the proposed scheme in the case of slice losses with a different error rate. Each slice consists of an entire row in the image frame. We encode the Foreman QCIF sequence with all forward predictions and five reference frames are allowed for prediction during the encoding process. It is assumed that short delay exists between the sender and the receiver. The performance of our scheme is compared with that of the adaptive intra refresh method (where the refreshing rate changes according to the error probability) and the H.264 video without error resilient tool. A simple error concealment tool is implemented at the decoder side, which copies the blocks in the previous frame at the same position to conceal the corrupted macroblocks in P frames and uses spatial interpolation to conceal corrupted macroblocks in I frame. The results are shown in Figure 13. As a benchmark, the original PSNR of the coded H.264 is also given in the figure.

It is clear that the proposed scheme has a much better performance. When the error probability is lower, the PSNR degradation is mainly caused by corrupted video data since most propagation errors have been eliminated. Even when the error probability reaches 10%, the PSNR value still remains around 30dB. The intra refresh method does not perform as well. This is due to the fact that some macroblocks in H.264 are coded with intra predictions so that they can also be affected by errors occurred in the neighboring areas. It does not necessarily stop error propagation. Second, since the off-line encoder does not have any information on existing errors, it cannot precisely assign those vulnerable macroblocks to be coded as intra blocks. Thus, it cannot stop error propagation efficiently. Another disadvantage of intra refresh is that the intra refresh rate is determined and set to meet the expected error probability during the encoding process. However, if the channel characteristics change during the transmission, the intra refresh method may have even worse performance. The encoder in our scheme does not require any information about channel characteristics. It can still function well with a changing error rate.

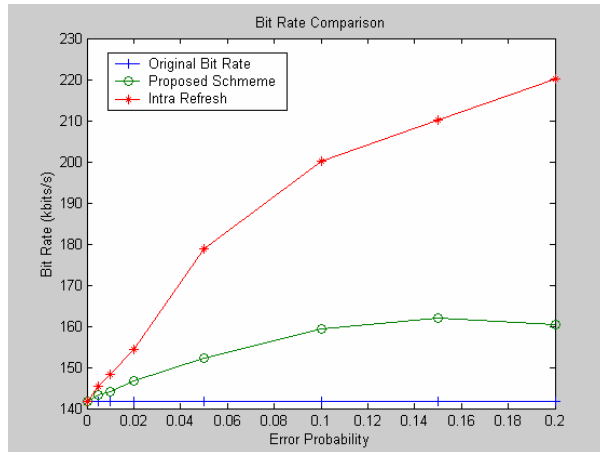


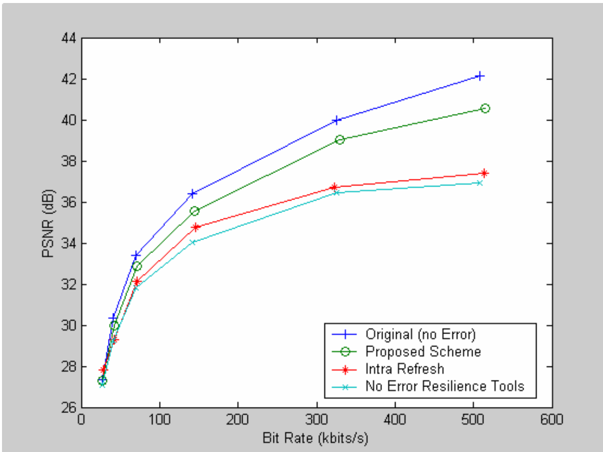
Fig. 14: Bit rate overhead comparison for the case of multiple reference frames.

We also compare the bit rate consumption of various schemes. Figure 14 shows the bit rate overheads of our scheme and the adaptive intra refresh approach as compared with the original bit rate. Apparently, the overhead introduced by our scheme is smaller. Our scheme does not introduce any bit rate overhead when no error occurs. When the error probability is very high (such as 20%), the bit rate overhead may drop because, for some macroblocks all of their alternative coded versions are affected by errors under such a high error rate. Thus, no replacement is made.

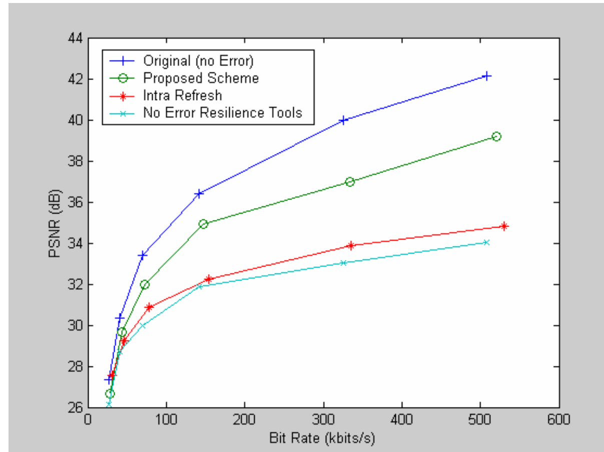
To further prove the performance of the proposed scheme, we tested its R-D performance with respect to different channel error probabilities and compared it with the adaptive intra refresh method with the same test sequences and conditions. The original R-D performance without errors and the R-D performance of the codec without error resilient tools are also plotted for comparison. As shown in Figure 15, our scheme outperforms intra refresh at all error rates. Especially, when the error probability is low (in Figures 15 (a) and (b)), its R-D performance is quite close to the error-free R-D curve, which means that error propagation is largely reduced or completely eliminated in these cases.

B. *Case with A Single Reference Frame*

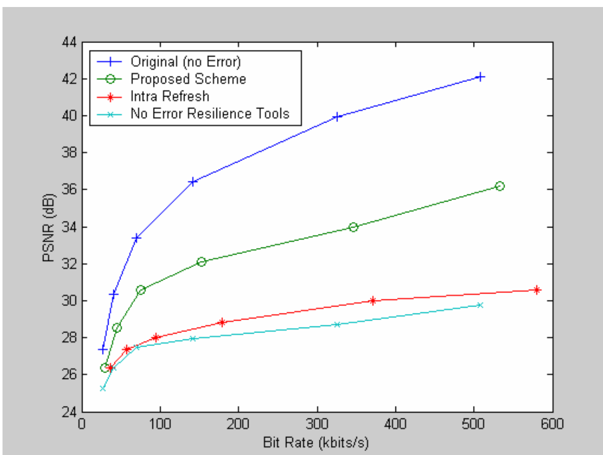
When there is only one reference frame available, we can use both SI refresh and the alternative SP macroblocks coded by predictions from the single reference frame at the same time to achieve better performance. Here, we tested this hybrid approach by comparing the R-D performance at different error probabilities with intra refresh. The results are shown in Figure 16.



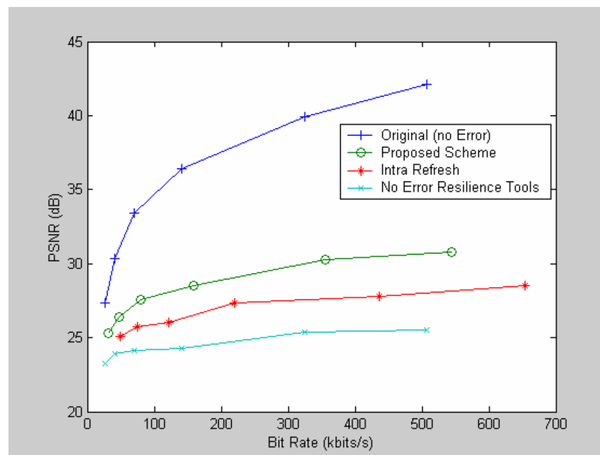
(a) Error probability = 1%



(b) Error probability = 2%

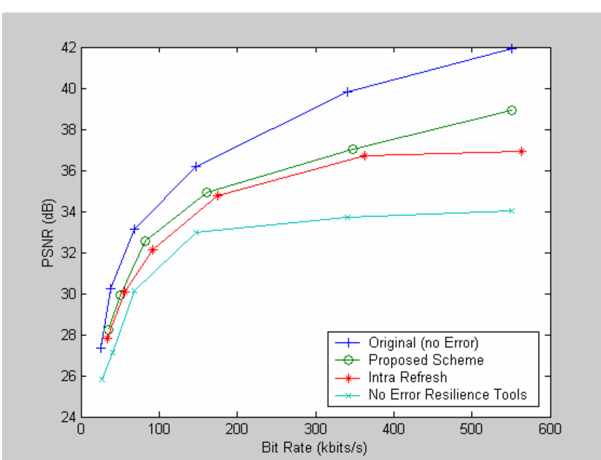


(c) Error probability = 5%

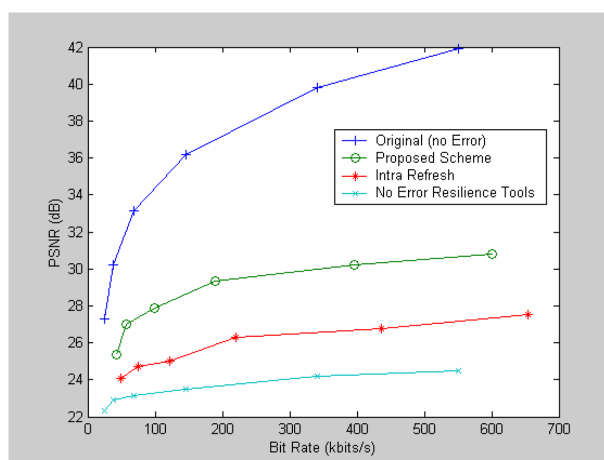


(d) Error probability = 10%

Fig. 15: Comparison of the R-D performance with respect to multiple frames under different error probabilities.



(a) Error probability = 1%



(b) Error probability = 10%

Fig. 16 R-D performance comparison – Single reference frame

As shown in the figure, our scheme still outperforms intra refresh under these scenarios. However, its R-D performance is worse than that of the same scheme but with multiple reference frames. This is mainly due to the fact that both alternatively coded SI macroblocks and SP macroblocks coded by different predictions in a single frame usually generate a large residual, which increases the bit rate consumption of the proposed scheme.

V. CONCLUSION AND FUTURE WORK

The proposed scheme can provide high error resilient performance under different scenarios as shown in the section of experimental results. It is able to completely eliminate error propagation when the error rate of the network does not exceed an unacceptable level. As most multimedia applications demand effective quality control while the quality requirement grows higher, the proposed scheme offers a great tool to 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 as compared with other error resilient tools. It does not result in any overhead in bit rates when there is no transmission error. Besides, 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. The additional computational complexity introduced is low for both the encoder and the decoder. It is specifically designed for H.264, and can be adapted to any other coding standards that provide SP/SI or similar coding features.

Although the bit rate overhead introduced by our scheme is relatively low, it may be reduced further by improving the proposed scheme. For example, the quantization level used to encode alternative SP/SI macroblocks can be selected adaptively. Besides, the two types of alternative SP/SI macroblocks can be generated (one with higher bit cost and one with lower bit cost) and coded with different quantization levels. The one with a lower quantization level and a high bit cost will guarantee good error resilience performance, but if the macroblock being replaced is less important (it is not used as reference for subsequent frames), we may sacrifice this macroblock by transmitting a lower cost version instead during bit stream replacement. Or, we may not attempt to replace this macroblock at all because even if this macroblock is not constructed correctly, it will not affect subsequent frames and error propagation is stopped automatically. Such adaptive decisions on bit stream replacement will help reduce the bit rate overhead further. We will investigate this issue and some other aspects of the proposed scheme in depth in our future work.

REFERENCES

[1] ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, Document JVT-G050, 8th Meeting: Geneva, Switzerland, May 2003.

[2] International Standard Organization, “Information Technology-Coding of Audio-Visual Objects, Part2---Visual”, ISO/IEC 14496-2.

[3] Yao Wang, Stephan Wenger, Jiangtao Wen and Aggelos K. Katsaggelos, “Error Resilient Video Coding Techniques”, IEEE Signal Processing Magazine, pp. 61-82, July 2000.

[4] Marta Karczewicz and Ragip Kurceren, Document VCEG-L27, “A Proposal for SP frames ”, 12th Meeting: Eibsee, Germany, January 2001.

[5] Ragip Kurceren and Marta Karczewicz , Document VCEG-O47, “New Macroblock Modes for SP-frames”, 15th Meeting: Pattaya, Thailand, December 2001.

[6] ITU-T Rec. “Video Coding for Low Bitrate communication”, ITU-T Recommendation h.263, Geneva, Switzerland, 1998.

[7] Paul Haskell and David 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.

[8] Judy Y. Liao and John D. Villasenor,“Adaptive Intra Update for Video Coding over Noisy Channels”, in Proc. IEEE Int. Conf. On Image Processing (ICIP), vol.3, pp. 763-766, September 1996.

[9] Koji Imura and Yutaka Machida, “Error Resilient Video Coding Schemes for Real-time and Low bitrate Mobile Communications”, Signal Processing: Image Communication, 14(1999), pp.519-530, May 1999.

[10] Xiaosong Zhou, Eric Q. Li, and Yen-Kuang Chen, “Implementation of H.264 Decoder on General-Purpose Processors with Media Instructions,” SPIE Conf. on Image and Video Communications and Processing, vol. 5022, pp. 224-235, January 2003.

[11] Stephan Wenger, Document JVT-D063, “FMO 101”, JVT 4th Meeting: Klagenfurt, Austria, July 2002.

[12] Ye-Kui Wang, Miska M. Hannuksela, Viktor Varsa Ari Hourunranta, and Moncef Gabbouj, “The error concealment feature in the H.26L test model”, IEEE Int. Conf. on Image Processing, pp. 729-732, September 2002.

[13] Tu-Chih Wang, Hung-Chi Feng and Liang-Gee 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.

[14] Eckenhard Steinbach, Niko Farber and Bernd 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.

[15] Masahiro Wada, “Selective recovery of video packet loss using error concealment”, IEEE J. Select. Areas Commun., vol. 7, pp. 807-814, June 1989.

[16] Li-Wei Kang and Jin-Jang 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.

[17] Paolo Bucciol, Enrico Masala and Juan Carlos De Martin, “Perceptual ARQ for H.264 Video Streaming over 3G Wireless Networks”, Proceedings of IEEE Int. Conf. on Communications (ICC), vol. 3, pp. 1288-1292, June 2004.

[18] Video and image compression resources and research (VCODEX)," H.264 / MPEG-4 Part 10 Tutorials", http://www.vcodex.com/

[19] Marta Karczewicz and Ragip 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, July 2003.