Robust Streaming of Offline Coded H.264/AVC Video Via Alternative Macroblock Coding

Xiaosong Zhou, *Member, IEEE*, and C.-C. Jay Kuo, *Fellow, IEEE*

***Abstract*—An error resilient video streaming scheme that transmits offline coded H.264/AVC video through erroneous channels, called the alternative macroblock coding (AMC) scheme, is proposed in this work. In the AMC scheme, each macroblock can be reconstructed from one default and several alternative versions coded using different predictions. During the transmission, the sender tracks ACK or NACK messages from the receiver to detect transmission errors. When the reference used in the default version of a macroblock that has not been transmitted is corrupted by errors, one of its alternative versions that has the correctly received reference is selected to replace the default version in the output bit stream to stop error propagation. The AMC scheme is designed to ensure that the bit stream replacement during streaming will not cause significant mismatch at the decoder end and neither introduce large bit rate overhead to the transmitted bit stream. Furthermore, adaptive quantization selection and bit stream replacement methods are developed to improve the rate-distortion performance of received video. It is demonstrated by experimental results that the AMC scheme is effective in reducing error propagation in offline coded H.264/AVC video.**

***Index Terms*—AVC, error resilience, H.264, video coding, video streaming.**

I. INTRODUCTION

VIDEO streaming offers an effective means in distributing video data for entertainment and information services. Many communication networks (wired, wireless and cellular networks) start to provide sufficient bandwidths for transmission of high quality video data. At the same time, modern video compression techniques help reduce the bit rate of transmitted video stream substantially. The emerging video coding standard H.264/AVC [1] is shown to outperform previous coding standards in terms of compression efficiency [2]–[5]. It is expected that H.264/AVC will be widely used in video streaming applications in the near future. The media industry aims to provide video services for individuals at any place and any time. As more and more electronic devices and portable devices have built-in video display capability, there will be a tremendous growth in video streaming applications.

In typical streaming applications such as video on demand (VOD) and Internet streaming, video streams are compressed offline and stored in the streaming server. Upon user's request, the compressed stream is transmitted to the client device, where it is decoded and played back in real time. One main problem with video streaming applications is that it is difficult to achieve guaranteed end-to-end quality of service (QoS) over the entire streaming process [6], which is especially true for wireless networks. Data packets may be lost, corrupted or delayed during the transmission, and these transmission errors can degrade the perceptual quality of received video substantially over a long period due to error propagation. It is important to develop effective error resilient streaming techniques for reliable delivery of offline compressed H.264/AVC video over various communication networks.

Various tools were introduced in the past to improve error resilience of compressed video. However, few of them can be used to effectively stop error propagation in offline coded video and maintain a high video quality level in an erroneous environment. Existing error resilience tools are reviewed in Section II-B. Some of them insert redundant or side information into the bit stream in offline encoding to guide the decoder to reconstruct corrupted data or make the bit stream more robust to propagation errors. However, most of these solutions only work well for certain channel conditions. They are not adaptive to the varying channel status. Another group of tools has been designed for one-to-one conversational visual communication applications that allow a real-time handshaking mechanism between the encoder and the decoder. That is, by utilizing the feed-back information such as ACK and NACK messages, a real-time encoder can mark corrupted or lost video data in its reference frame buffer and avoid the use of them to predict future frames. As a result, a video stream of higher quality can be generated on the fly without being affected by errors occurring in previous frames. These methods are effective in reducing error propagation and highly adaptive to varying channel conditions.

However, the tools described above are not directly applicable in the streaming of offline coded H.264/AVC video for two reasons. First, the H.264/AVC encoder has a very high complexity and it is too expensive to re-encode all video frames based on the channel condition. Second, it is often that a streaming server should serve multiple users. It is difficult for the server to adopt real-time encoding to meet the need of different users in the video streaming context. Thus, a new error resilient scheme is developed in this work to achieve a similar error propagation reduction effect for offline coded H.264/AVC video using a concept called alternative macroblock coding (AMC). In the AMC scheme, the video stream can be adjusted during the streaming

Manuscript received October 22, 2006; revised July 23, 2007. This work was supported in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement EEC-9529152. This paper was recommended by Associate Editor W. Zhu.

X. Zhou is with Apple Inc., Cupertino, CA 95014 USA (e-mail: xiaosong@gmail.com).

C.-C. Jay Kuo is with Department of Electrical Engineering and Integrated Media Systems Center, University of Southern California, Los Angeles, CA 90089-2564 USA (e-mail: cckuo@sipi.usc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2008.918762

process yet without the need to re-encode the entire video stream so that the complexity is significantly lowered for streaming servers. Besides its excellent error resilience, the AMC scheme does not degrade coding efficiency of transmitted video when no errors occur and only introduces a relatively small amount of overhead in bit rates when there are errors. It works effectively under varying channel conditions with different bit error rates and network delays. It serves as a generic tool for robust video streaming over erroneous channels.

The basic idea of the proposed AMC scheme is stated below. Each macroblock can be reconstructed from one default and several alternative versions coded using different predictions. During the transmission, the sender tracks ACK or NACK messages from the receiver to detect transmission errors. When the reference used in the default version of a macroblock that has not been transmitted is corrupted by errors, one of its alternative versions that has the correctly received reference is selected to replace the default version in the output bit stream to stop error propagation. The encoding of alternative macroblocks is designed to ensure that the replacement will not cause significant mismatch to have an influence on subsequent frames. To achieve excellent error resilience as well as high coding efficiency, two methods are developed to create alternative predictions in the encoding process. The first one utilizes the multiple-reference-frame feature in H.264/AVC while the second one uses concealed reference frames to generate alternative predictions. They can be used to meet different application scenarios. In addition, adaptive quantization selection and bit stream replacement methods are developed to work together with the proposed scheme to further improve the rate-distortion (R-D) performance of the received video.

The rest of this paper is organized as follows. The background of the error resilient video coding problem is described in Section II. An overview of the proposed error resilient video coding system is presented in Section III. Then, the proposed AMC scheme is discussed in Section IV. Adaptive quantization selection and bit stream replacement methods are proposed to further improve the performance of the AMC system in Section V. Experimental results of the AMC scheme are shown in Section VI. Finally, concluding remarks are given in Section VII.

II. Background Review

A. Error Propagation

In most existing video coding standards including H.264/AVC, the video stream is compressed using motion-compensated predictive coding. As a result, transmission errors can cause chain effect in the received video stream that is called "error propagation," and the associated errors "the propagation (drift) errors." For example, if transmission errors occur in one received frame, parts of that frame are corrupted and cannot be reconstructed or displayed correctly. Furthermore, some blocks in the next frame will be affected and cannot be reconstructed correctly as well if their predictions are generated from the corrupted areas in the previous frame. The reason is that the decoder is not able to obtain correct predictions as required. Similarly, the third frame may also be affected by the incorrect reconstruction of the second frame. As long as future frames are coded by predictions from these corrupted or affected data, such propagation errors will continue. The magnitude of the errors tends to attenuate as they propagate but the attenuation speed is low [7]. Error propagation degrades the peak signal-to-noise ratio (PSNR) and subjective quality of received video for a long period of time, which is one of the biggest challenges in error resilient video transmission. If propagation errors can be eliminated or largely reduced, the impact of transmission errors will be much limited and the quality of received video can be well preserved.

B. Review of Error Resilient Tools

Various error resilient tools have been introduced in the past to improve the error resilience performance of compressed video for previous video coding standards. A large number of error resilient tools are designed to offer better protections to the compressed video data against transmission errors. For example, error resilient entropy coding methods such as reversible variable length coding (RVLC) [8] and others [9], [10] can help partially recover corrupted video packets. The compressed video data can also be classified based on their importance and given different degrees of protection. For example, scalable or layered coding methods [11]–[14] are used to classify compressed video data into different layers, and more protection can be given to the base layer which provides the basic quality level. Data embedding [15] offers another approach that embeds important data such as motion vectors into discrete cosine transform (DCT) coefficients so that the embedded data can be extracted if they are lost or corrupted during transmission. In multiple hypothesis coding scheme [16], a macroblock is coded using multiple motion vectors and references, thus a single lost or corrupted reference will do less damage to its reconstruction. All these methods help reduce the damage of errors and most of them can be adapted to work for H.264/AVC. However, the performance is achieved at the cost of redundant transmission or quality degradation. More importantly, when the video stream is coded offline, it is difficult to decide where and how much the protection should be applied. As a result, the effect of these protective methods is limited.

One way to reduce error propagation in previous standards is to insert intra macroblocks in temporally coded (P- or B-) video frames. The inserted intra macroblocks will not be affected by errors in previous frames because they are not coded using motion compensation. This technique is known as "intra refreshing" [17]–[19], and it is shown to be effective in certain scenarios. However, as intra macroblocks in H.264/AVC are coded by intra predictions generated from their neighboring areas, the insertion of intra macroblocks may fail to stop error propagation since its neighboring blocks might be corrupted or affected by errors. Therefore, simply inserting intra macroblocks is not sufficient in stopping error propagation for H.264 video. Additionally, it is difficult to decide where and how many intra macroblocks need to be inserted without the knowledge of actual errors that occur during transmission. Another problem is that intra macroblocks usually have much lower coding efficiency than inter macroblocks so that the overall coding efficiency of the video stream may be degraded significantly if a large number of intra macroblocks are inserted.

Error concealment techniques [20]–[24] are commonly used in decoders including H.264/AVC decoders [30] to reconstruct a better version of lost or corrupted video content in order to improve the quality of the received video. The reconstruction is usually not perfect because the concealed video frame is not the same as the original one transmitted by the sender. Therefore, the concealed data still causes error propagation if it is used as the reference for future frames although the magnitude of such error may be reduced.

Besides error concealment, a number of new coding features are adopted by H.264/AVC for error resilience. For example, macroblocks grouped in the same package may not be adjacent to each other geographically under the flexible macroblock ordering (FMO) [31], [32]. Thus, if a packet is lost or corrupted, error concealment tools can generate a better concealed reconstruction based on macroblocks that are spatially distributed. Redundant slices (RS) coded using a coarser quantization parameter than primary slices can be used to obtain a low-quality reconstruction of a slice when the primary slice is lost. Even though these methods help improve the error resilient performance, their effectiveness is very limited in reducing error propagation for offline coded H.264 video.

Feedback-based schemes [25]–[28] provide excellent error resilient performance for one-to-one conversational visual communication applications. By tracking the ACK or NACK messages sent by the receiver, the sender can identify transmission errors occurred in the past and encode the video stream adaptively on the fly. One popular approach is known as NEWPRED [29] and adopted in MPEG-4. NEWPRED selects reference frames dynamically with real-time encoding to avoid referencing to corrupted or lost data. In other words, the encoder attempts to use reference frames that are successfully received and acknowledged only. Since error propagation can be largely reduced in feedback-based schemes, the quality of the video can be well preserved. However, as these methods demand real-time encoding of the entire bit stream according to the channel condition, they are typically used in the context of conservational visual communication applications such as one-to-one video telephony or video conference. They are not suitable for video streaming applications such as video-on-demand for two reasons. First, the H.264/AVC encoder has a high computational complexity and it is costly to re-encode all video frames based on user's channel condition. Second, a streaming server may serve multiple users with the same video bit stream at the same time, dynamic real-time encoding to adapt to different channel conditions of different users is even more challenging.

Generally speaking, it is desirable to develop an error resilient scheme that can reduce error propagation of offline coded video without real-time encoding. Another feedback-based strategy is based on the idea of automatic repeat request (ARQ). Even though the ARQ scheme does not require real-time encoding of video data, it may not be suitable for some video streaming applications if the retransmission delay is too long to be acceptable. The impact of network delay on feedback-based schemes will be discussed in Section III.

In summary, even though error resilience in video transmission over erroneous channels has been studied for more than one decade, the error propagation problem for offline coded video has not been well addressed in the past. A novel error resilient scheme, called the AMC scheme, is proposed in this work to achieve this goal. It is designed to stop or largely reduce error propagation. The proposed AMC scheme does not affect coding efficiency of transmitted video when no error occurs, and introduces a relatively small amount of bit rate overhead in the presence of errors. It can also be applied together with other error resilient tools to achieve better performance.

III. AMC-Based Video Streaming System

A. System Overview

In this section, we describe an error resilient video streaming system that is able to eliminate or largely reduce propagation errors for offline coded H.264 video based on AMC. The AMC-based video streaming system consists of two main parts.

1) *Offline video stream preparation*: The source video is compressed using the standard H.264 encoding process to give a *default* bit stream at a certain bit rate, and it is stored in the streaming server before the streaming operation starts. Furthermore, we encode additional versions of each macroblock with different predictions or prediction methods. These alternatively coded versions can also be generated during the offline encoding process and stored in the server. They are coded in a special way to have close reconstruction and compression efficiency with the default version. The coding methods of these macroblocks are discussed in Section IV.
2) *Online selected bit stream replacement:* During the streaming process, the sender tracks the ACK or NACK messages to determine previous transmission errors. Based on the information, the sender replaces the default version of macroblocks in output bit stream that are affected by previous errors to avoid them referencing the corrupted regions in order to stop error propagation.

When no error occurs, the default compressed video stream is transmitted by the sender to the receiver without any changes. When a transmission error occurs, part of the transmitted video will be corrupted or lost. As explained in Section II-A, the error will propagate to subsequent frames. This error propagation phenomenon cannot be stopped by the error concealment technique at the decoder. As a result, certain macroblocks in subsequent frames that have not been transmitted will be affected if the references used in their default version are corrupted. However, if one of their alternatively coded versions uses a prediction that is not affected by any transmission error, the sender is able to substitute the default version with this alternative version in the output bit stream. Then, the decoder can reconstruct these macroblocks without being affected by previous errors. In other words, error propagation can be stopped or largely reduced by the substitution.

The macroblocks affected by previous errors can be identified by error tracking techniques [27]. There are a number of ways to implement error tracking in the proposed AMC scheme. For example, motion vectors and block types can be stored in a separate file during offline encoding and used to extract the information to track how errors propagate during the streaming process. Another way to obtain such information without an additional file is to decode the motion information in real time. Since the

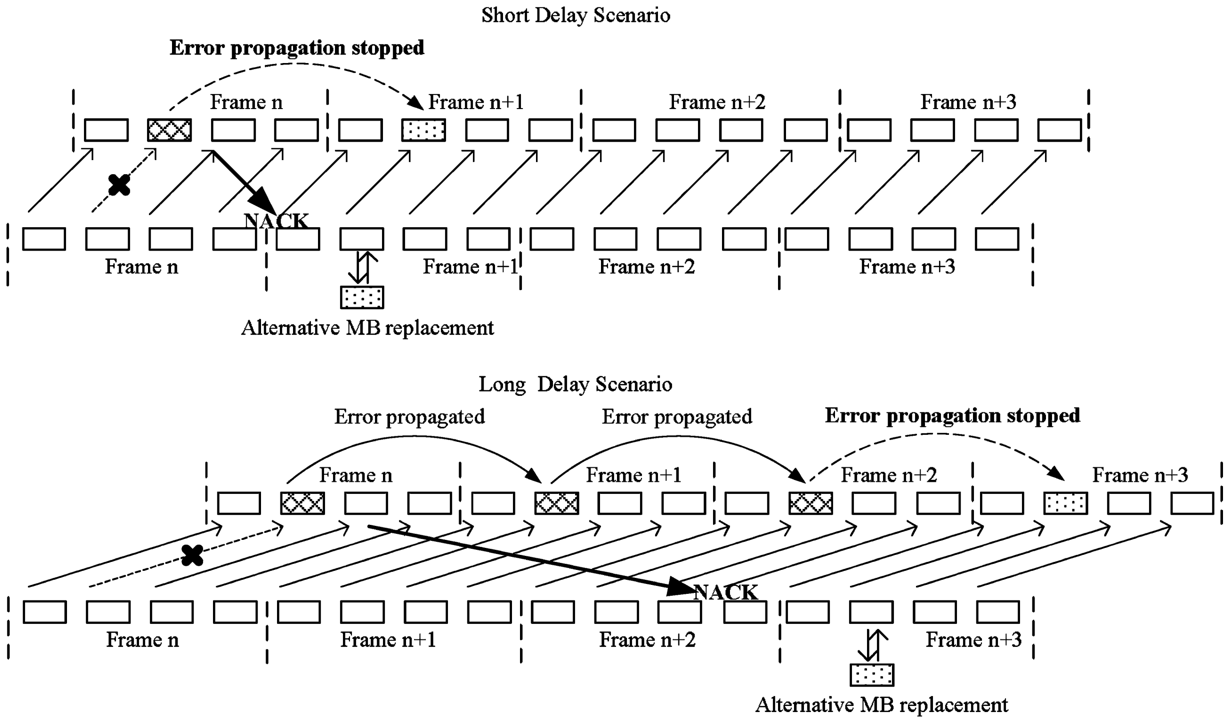


Fig. 1. Illustration of how the AMC scheme works under different network delays with the NACK messages.

complexity the H.264/AVC decoder is much lower as compared with that of the encoder, the resulting method is still efficient.

B. Relationship between AMC and Network Delay

Since the feedback-based error resilient systems, including the AMC scheme, are dependent on the ACK and NACK messages received from the receiver, network delay plays a critical role in these systems. For example, the ARQ retransmission scheme only works if a retransmitted packet arrives before its actual decoding time. In other words, when a corrupted packet is detected by the receiver, its replacement (or a retransmitted packet) has to arrive before the corrupted packet reaches the end of the decoder buffer and will be decoded immediately. Mathematically, the round-trip network delay time should be less than the queuing delay decided by the decoder buffer size. This condition could be difficult to meet if the decoder buffer size is small and the associated queuing delay is short. This is often the case if the receiver is a portable device. The ARQ retransmission scheme could also fail when the round-trip network travel time is too long.

The proposed AMC scheme does not work on the corrupted or lost video data directly and does not retransmit any data. Instead, it adaptively adjusts the proper portion of the remaining bit stream that has not yet been transmitted. Fig. 1 shows an example of how the AMC scheme works under different network delays by utilizing the NACK messages. In this example, a NACK message is sent to the sender as soon as the receiver detects a lost packet in frame n. (Since packets may arrive out of order at the receiver side, a moving packet window is utilized to determine packet loss.) Under the short-delay scenario, the NACK message arrives before the sender transmits frame $n+1$. Therefore, the sender can identify macroblocks in frame $n+1$ that refer to the lost packet by tracking the motion vectors and replace them with proper alternative macroblocks that are not affected by the lost packet. As a result, transmitted frame $n+1$ is no longer affected by the error, and the associated error propagation is stopped. Other subsequent frames will not be changed until a new error is detected.

Consider another example that has a longer network delay, say, the NACK message for a macroblock in frame n is received after the sender has transmitted k frames, i.e., $n+1, n+2, \ldots, n+k$. On one hand, it is too late to correct the errors that have already been propagated to these k frames. On the other hand, the sender can still stop error propagation using the AMC scheme from frame $n+k+1$. It determines macroblocks in frame $n+k+1$ that refers to error corrupted macroblocks in previous frames and replaces them with their proper alternative versions to stop error propagation. The AMC scheme can also work with the ACK messages in a similar manner.

Generally speaking, the AMC scheme can be applied to both short and long delay scenarios as long as error tracking is effective. We have so far presented an overall framework of the AMC-based video streaming system but have not yet described ways to generate alternative macroblocks, which is the main subject in the next section.

IV. Alternative Macroblock Coding (AMC) Schemes

Before we elaborate the AMC scheme, it is important to discuss three desirable features of the scheme so that AMC can be effective in offline coded H.264/AVC video streaming.

1) *Reconstructions of alternative macroblocks should be very close to the default coded version.* Since the entire bit stream is compressed offline, the replacement will affect subsequent frames if there exists a large difference between

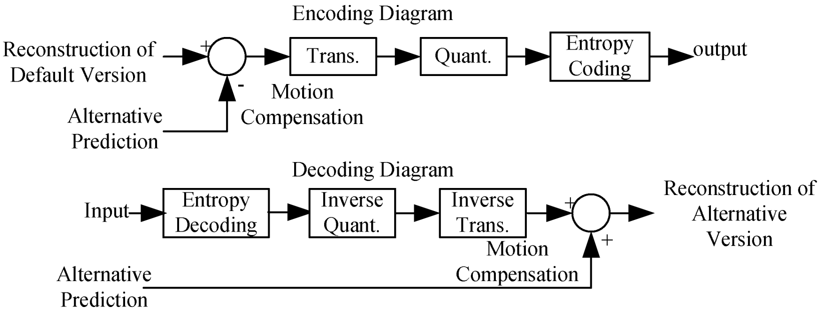


Fig. 2. General framework of alternative macroblock encoding (top) and decoding (bottom).

the default version and other reconstructions. Suppose that a number of macroblocks are replaced by their alternative versions in frame n. After transmission of the new version of frame n, the transmitter will continue to send the default coded stream of subsequent frames to the receiver. If the alternative macroblocks used in the replacement are very different from their default versions, there will be a mismatch that in turn affects the decoding of subsequent frames when these reconstructions are used as the reference.

2) *The coding overhead should be as low as possible*. If no error occurs during transmission, it is desirable that the bit stream being transmitted is the same at that obtained by a normal H.264/AVC encoder. If the error occurs, the coding of alternative macroblocks should not increase the original bit rate much since the video transmission bandwidth is a critical performance metric. The alternative macroblocks should be coded with high coding efficiency such that the introduced bit rate overhead is small.

3) *Independence of default and alternative versions*. The encoding of the default and alternative versions should be independent of each other and different methods should be used to encode alternative macroblocks to increase the diversity of the underlying bit stream.

The specific coding schemes of alternative macroblocks are described in this section to realize the above three features.

A. *Process of AMC*

For standard motion-compensated predictive coding schemes such as MPEG-2, H.263 and H.264/AVC, a general framework to encode and decode alternative macroblocks is shown in Fig. 2. To get a reconstruction close to the default version, the reconstructed default version instead of raw video data is used as the input as shown in the figure. The alternative macroblocks are coded using alternative predictions that are different from the default one. If the difference between the reconstruction of the default version and the alternative prediction is small, alternative macroblocks can be coded efficiently.

When the H.264/AVC extended profile is adopted for the streaming application, we can use the secondary SP macroblock coding process to encode alternative macroblocks. The SP/SI picture coding in the H.264/AVC standard was introduced by Karczewicz and Kurceren [33] to enable flexible stream switching and splicing in video streaming. There are two types of SP/SI pictures, i.e., primary and secondary SP/SI pictures, and they are always used in pairs. The macroblocks in these pictures are coded differently from those in the normal I- or P-pictures. Here, only the coding process and syntax of the secondary SP macroblocks are adopted [34] for the coding of alternative macroblocks as shown in Fig. 3. It is worthwhile to emphasize that the reconstruction of the default coded version is used as the input.

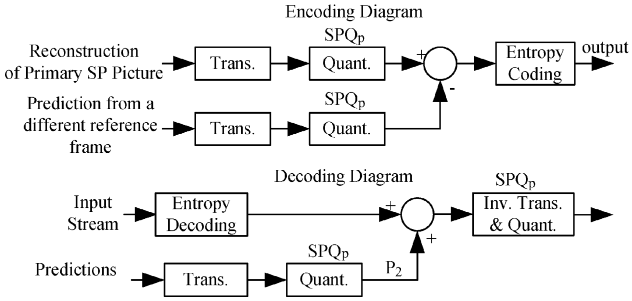


Fig. 3. Encoding/decoding diagram of alternative macroblocks using the secondary SP picture coding syntax.

When compared with the normal motion-compensated coding, one major difference of the secondary SP encoding process is that the transform and the quantization modules are done before residual calculation. With such a coding method, quantization errors of the prediction value will not affect the reconstruction of secondary SP macroblocks. As shown in the figure, value P_1 in the encoding process and value P_2 in the decoding process are completely identical. By combining the encoding and decoding processes of alternative macroblocks in Fig. 3, we can simplify the overall process and depict it in Fig. 4. As shown in the figure, the only possible difference of the input (i.e., reconstruction of the default version) and the output (i.e., reconstruction of an alternative version) is caused by the forward/inverse transforms and the quantization process. Thus, although alternative macroblocks are coded using different predictions and/or prediction methods, their reconstructions are always identical. This feature allows selecting different alternative versions without introducing quality fluctuation.

The quantization level SPQ_p can be chosen properly so that only very small quantization errors are introduced in the process. It is a tradeoff between the closeness of the reconstruction to the default values and the bit rate overhead due to smaller quantization step size. This issue will be discussed in detail in Section V. Multiple alternative predictions are used to generate different alternatively coded versions for the macroblock. Although the best prediction is often the one used

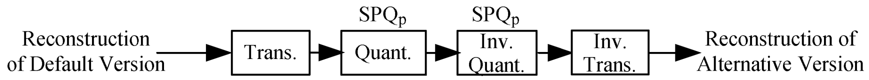

Reconstruction of Default Version
Trans.
SPQp
Quant.
SPQp
Inv. Quant.
Inv. Trans.
Reconstruction of Alternative Version

Fig. 4. Integrated encoding/decoding process of alternative macroblocks.

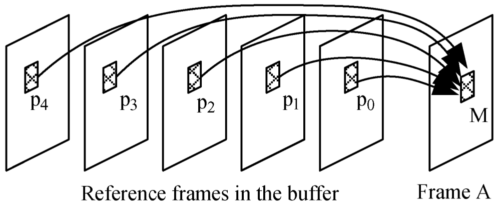


Fig. 5. Prediction from multiple reference frames.

in the default coded version, it is still possible to get good alternative predictions for a given macroblock. We propose two different methods to generate such predictions in Section IV-B and IV-C.

B. Alternative Prediction With Multiple Reference Frames

H.264/AVC allows multiple reference frames (or long term 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 16 × 16, the encoder will find one best matching 16 × 16 block in each reference frame. As shown in Fig. 5, the five best matching blocks $(P_0 - P_4)$ 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 (default) coded version for macroblock M and its reconstruction will be saved in the reference buffer to be used for future frames.

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 other predictions $P_1 - P_4$ are corrupted at the same time. Thus, these predictions can serve as good alternative predictions in the AMC scheme.

We 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. Four alternatively coded SP macroblocks can be generated using the alternative predictions $P_1 - P_4$ along with one default version coded using P_0. These alternative predictions are expected to match macroblock M well since they provide the best match in each reference frame via motion search. Such alternatively coded SP macroblocks can be generated for each macroblock in the bit stream. During the video streaming process, the sender always sends the default coded macroblocks unless an error is detected. If it does detect an 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 this is the case, the bit stream of the alternative version is used to replace the original bit stream containing the default version, and then the modified bit stream is transmitted. As long as not all of the predictions saved for this macroblock are affected by transmission errors, the sender can always find a suitable coded version to have the macroblock correctly decoded. Thus, the error propagation can be stopped.

C. Alternative Prediction With One Reference Frame

The above alternative SP macroblock encoding scheme works well when multiple reference frames are available for prediction. However, in some mobile applications, the reference buffer size may be restricted to one due to the limited memory size of the receiver. 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 IDR frame could only use one frame for prediction. Sometimes, alternative predictions from other reference frames may generate a large residual and then introduce a high bit rate overhead if being transmitted (which may happen during scene changes). Under these situations, a different method to generate alternative predictions is desired.

One possible solution is that we generate multiple predictions from the same reference frame. For example, we can divide the motion search region into four parts and find the best-matched prediction from each part as alternative predictions. On one hand, these predictions may not provide good predictions. On the other hand, since they are mutually exclusive from each other they are less likely to be corrupted by the same error. Another way is to generate alternative SI macroblocks instead, where the alternative prediction is generated using intra prediction. With these two methods, alternative macroblocks can be generated effectively. However, their coding efficiency is degraded due to poor predictions [35].

In the following, we propose a more effective technique to generate alternative macroblock when only one reference frame is available by exploiting error concealment techniques. Although the error concealment tools implemented at the decoder cannot be used to completely stop error propagation, they 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 correct reference frames used at the encoder, there is a mismatch and subsequent frames may be vulnerable to propagation errors. However, if the encoder has the knowledge of error concealments tools used by the decoder, such a mismatch can be eliminated as described below.

Fig. 6 shows video frames received at the decoder during the transmission of the encoded H.264 stream. Suppose that a transmission error occurs in frame P_n so that it is corrupted at the

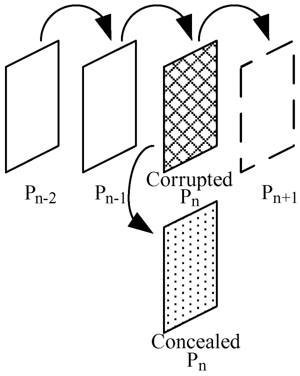


Fig. 6. Error concealment of the decoder.

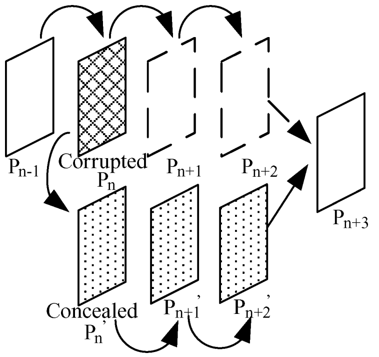


Fig. 7. Concealed multiple reference frames for the long delay scenario.

receiver. Upon detection of the transmission error, the decoder conceals the corrupted frame using error concealment tools and stores concealed P_n' as a reference frame in its buffer. At the same time, after detecting the error in frame P_n by tracking ACK or NACK messages, the sender can generate the concealed frame P_n' exactly the same as that in the receiver by using the same error concealment methods. Then 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 these macroblocks by using predictions from concealed reference frame P_n' with the same coding method as shown in Fig. 3. The concealed reference frame can generate a reasonable prediction and the coded SP macroblock will then be used to replace the default version in the bit stream and sent to the receiver to stop error propagation.

In the long network delay scenario, the sender can generate concealed versions of intermediate frames. As shown in Fig. 7, the sender generates concealed frames P_n', P_{n+1}' and P_{n+2}' that are identical to concealed frames at the decoder. Thus, using these regenerated concealed frames as new references, the sender can encode the affected macroblocks in P_{n+3} into SP macroblocks and send them to the receiver to stop error propagation. Consequently, the AMC scheme still works for the long network delay scenario. Please note that it only requires re-encoding of the macroblocks that are affected by previous errors in the frame to be transmitted next. It does not require re-encoding of any other part of the bit stream unless new errors are detected. Since the number of the macroblocks to be re-encoded is limited, the introduced computational overhead is relatively small.

To summarize, both approaches described above can be used to generate alternative coded SP macroblocks to eliminate or reduce error propagation in various coding and delay scenarios. They can also be combined to make the integrated scheme more robust.

V. Adaptive Alternative Macroblock Coding (AAMC)

A. *Adaptive Quantization for Alternative Macroblocks*

The encoding process of the AMC scheme ensures that the difference between the reconstruction values of default and alternative versions is very small. The mismatch can be further reduced by properly choosing the quantization step size when encoding the alternative macroblocks. A simple experiment is given below to shed light on how the mismatch affects the quality of subsequent frames and the impact of using different quantization step sizes.

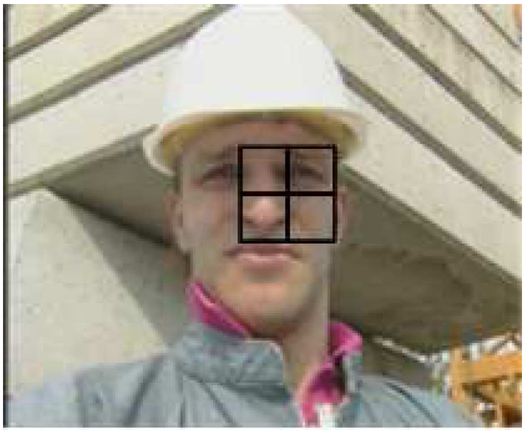


Fig. 8. Four macroblocks affected by an error in its previous frame as the reference.

The Foreman QCIF sequence is used in the experiment as shown in Fig. 8, where the quantization level to encode both the default bit stream and the alternative version is 28. It is assumed that an error occurs in the previous frame so that four macroblocks in the current frame need to be replaced with their alternative versions. The error is selected such that the four macroblocks to be replaced are in the region of interest (ROI) of the frame. The PSNR value of subsequent frames after the replacement is measured and compared with the original PSNR. The result is shown in Fig. 9. We see from the figure, the first frame after replacement is affected most, which has more than 0.1 dB degradation in PSNR. The other subsequent frames are also affected but in a smaller degree. By viewing the reconstructed video sequence after replacement, there is no noticeable perceptual quality degradation. This experiment proves that the AMC coding scheme is able to reduce the mismatch introduced by bit stream replacement.

Furthermore, we compare the PSNR degradation under different quantization levels. Four quantization levels are used to encode alternative versions. The average PSNR value of 60 subsequent frames after replacement is calculated. The results are shown in Fig. 10. The PSNR value of the original video stream is provided for comparison. As shown in the figure, the PSNR

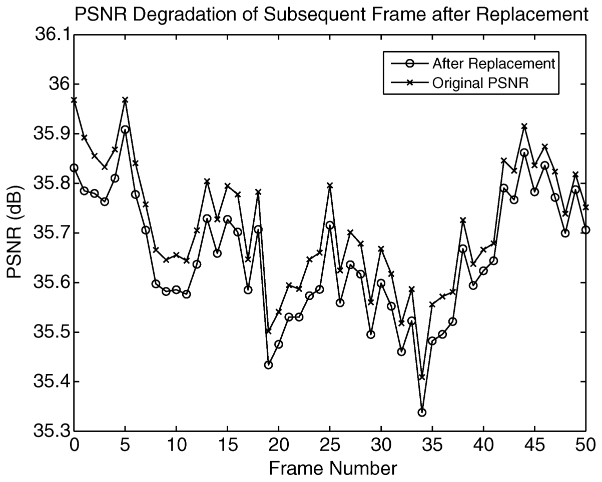


Fig. 9. Comparison of the PSNR value of subsequent frames after replacement for alternative SP/SI macroblocks.

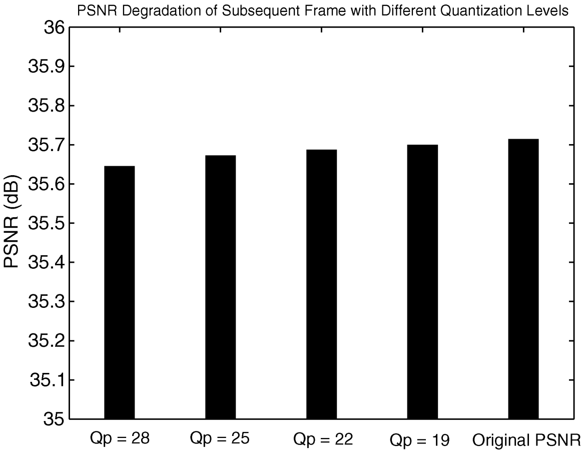


Fig. 10. Comparison of the average PSNR value of subsequent frames under different quantization levels for alternative SP/SI macroblocks.

degradation is further reduced via the use of a smaller quantization step size, which is desirable in the AMC scheme. However, more bits is needed to encode the alternative macroblocks with smaller quantization step size. Although the number of alternative macroblocks used in the replacement is limited, they may still introduce considerably large bit rate overhead if the quantization step size used is too small.

As observed from the experiment, we find it important to choose the quantization step size properly to obtain a better R-D tradeoff. In the following, we propose a method to select the quantization step size adaptively to improve the overall coding efficiency. The Lagrange R-D cost function

$$J = D + \lambda R \tag{1}$$

is often used in a video encoder to make coding decision to achieve the best R-D performance of a coded video stream. In an erroneous environment, the received video stream cannot be perfectly reconstructed due to transmission errors and error propagation. The above R-D cost function can be adapted to the context of error resilient coding by measuring the expected distortion caused by errors (D_e) and the actual transmission bit rate.

When a macroblock is replaced by its alternative version, the distortion caused by error propagation from an earlier error is replaced by the distortion caused by the mismatch of the reconstruction of the alternative macroblock, which is denoted by D_a. The actual transmission bit rate for the macroblock is the bit rate for the alternative macroblock denoted by R_a. Thus, the problem to minimize the R-D cost of the alternative macroblock can be formulated as

$$\min_{Q_p} J_a, \text{ where } J_a = D_a + \lambda R_a. \tag{2}$$

Since the distortion caused by the replacement will affect subsequent frames, we select the quantization parameter to lower the resultant distortion. To achieve the best overall R-D performance, we can set up the following optimization problem:

$$\min_{Q_p} J'_a, \text{ where } J'_a = D_a + D_{\text{prop}} + \lambda R_a \tag{3}$$

and where $D_{\text{prop}} = \sum_{i=1}^{N} D_i$ denotes the distortion propagated to subsequent N frames. Here, we use N to represent the number of remaining frames in the group of pictures (GOP) since the mismatch will not affect the next GOP. The exact value of D_i can be calculated in an offline encoding process. To reduce the complexity, they can be estimated by

$$D_{\text{prop}} = \sum_{i=1}^{N} \alpha^i D_a, \tag{4}$$

where α is the attenuation factor of error propagation. Its calculation is discussed in [7]. By using (3), the quantization step size of the alternative macroblock can be adaptively determined to achieve the best R-D performance.

B. Adaptive Bit Stream Replacement

In the proposed AMC scheme, if a macroblock is affected by an error, it is replaced by an alternative version in the output bit stream. The replaced macroblock can be correctly reconstructed at the decoder side with little mismatch from the reconstruction of the default version. In most cases, the mismatch error is much smaller than the propagation error if there is no replacement so that error propagation can be largely reduced or eliminated by the bit stream replacement. However, there are occasions where the propagation error is very small. For example, consider the case where a macroblock is decomposed into 16 submacroblocks of size 4 × 4 and only one of the 4 × 4 blocks has the reference to a corrupted region. With the help of an error concealment tool, the propagation error in this macroblock could be very small. Under this scenario, it may not be necessary to replace the bit stream with an alternative macroblock. Besides, the bit rate overhead is reduced by transmitting fewer alternative macroblocks since they tend to consume more bits than the default version. It is desirable to make decision on the bit stream replacement adaptively based on actual errors in the streaming process.

To achieve the best R-D performance of the received video stream, a macroblock will be replaced only when the R-D cost of the received macroblock after replacement is smaller than that without replacement. This criterion can be written as

$$J_a < J_e \tag{5}$$

where

$$J_e = D_e + \lambda R \tag{6}$$

is the R-D cost of the macroblock without replacement. The distortion D_e is the distortion caused by the propagation error due to previous corrupted or lost video packets. As discussed in Section III, it can be calculated by some error tracking technique. The bit rate R is the bit rate of the default coded macroblock.

In summary, adaptive quantization parameter selection and adaptive bit stream replacement improve the R-D performance of the AMC scheme. They also provide a basis to develop simple yet efficient methods to achieve similar performance. For example, the decision on the quantization parameter is based on the impact of the alternative macroblock to subsequent frames in the same GOP. Instead of calculating the R-D cost, the impact of the macroblock can be evaluated by other "easy-to-collect" data such as the number of remaining frames in the GOP and the number of macroblocks/blocks that will be affected by the mismatch of the alternative macroblock. Similarly, to determine how much a macroblock has been damaged and decide whether it is worthwhile to do bit stream replacement, the number of partitions (in terms of 4 × 4 blocks) in the macroblock affected by previous errors can be used as a rough measure. These low-complexity methods are under our current investigation.

VI. Experimental Results

The H.264 JM 8.6 reference code [36] was modified to implement the proposed AMC coding scheme. The default error concealment methods implemented in the decoder of the reference codes were turned on. In the experiments, various channel characteristics were examined. Error patterns were generated to simulate both random and burst errors with a broad range of error rates occurring in real world applications. The list of test conditions conducted in the experiment is shown in Table I. Both random and burst errors generated by the Gilbert–Elliot packet erasure channel model [37] with an average length of 3 packets were simulated. For each test condition, experiments were repeated over 200 times on randomly generated error patterns.

In the simulation, each video frame was divided into five slices and each packet contained one slice of the compressed video data. We also tested the performance of the proposed scheme with the flexible macroblock ordering (FMO) of the checkerboard pattern. FMO changes the pattern of corrupted macroblocks and affects the performance of the AMC scheme accordingly. Since a concealed video frame was used in the AMC scheme for the one reference frame case, it is interesting to test its dependency on FMO as well.

TABLE I
List of Test Conditions

Test Condition	Error Type	FMO
T1	1×10^{-3} Random Packet Loss	Yes
T2	1×10^{-3} Random Packet Loss	No
T3	5×10^{-2} Random Packet Loss	Yes
T4	5×10^{-2} Random Packet Loss	No
T5	5×10^{-3} Burst Packet Loss (Length = 3)	Yes
T6	5×10^{-3} Burst Packet Loss (Length = 3)	No
T7	1×10^{-2} Burst Packet Loss (Length = 3)	Yes
T8	1×10^{-2} Burst Packet Loss (Length = 3)	No
T9	5×10^{-2} Burst Packet Loss (Length = 3)	Yes
T10	5×10^{-2} Burst Packet Loss (Length = 3)	No

A. Quality Comparison

In this experiment, the quality of received video streams using different error resilient methods is evaluated to demonstrate the superior performance of the proposed AMC scheme. We consider two AMC schemes, which generate alternative predictions based on "multiple reference frames" and "concealed reference frames." They are called AMC-1 and AMC-2, respectively. The scheme uses only the error concealment tools at the decoder (without AMC) is used as the benchmark. We also show other reference data in Tables II and III such as the original coded stream without any error, and the received video stream using ideal real-time re-encoding in the presence of errors. For the ideal re-encoding scheme, whenever an error occurs, the rest of the video stream is completely re-encoded to avoid referencing to corrupted data so that error propagation is completely eliminated. All schemes were simulated under the short network delay scenario as discussed in Section III-B.

Secondary SP macroblocks are used to encode alternative macroblocks. The quantization step size to encode all default versions is set to 28 and the quantization level used to encode alternative macroblocks in the experiments is set to 22. For each test sequence, the size of the group of pictures (GOP) is set to 60. Only the first frame of the GOP is coded as the intra picture while all other frames are coded as P-pictures. The average PSNR of received Foreman and Akiyo QCIF sequences under each test condition are shown in Tables II and III, respectively. The PSNR improvement of the AMC-1 scheme over the benchmark is listed as well.

The PSNR values of AMC-1 and AMC-2 are identical in most test cases, which confirms the design objective of the AMC scheme; namely, the same reconstruction can be obtained using different predictions in AMC. The slight difference between their PSNR results at high error rates (e.g., test conditions T3, T4 and T10 for the Akiyo sequence) is caused by the rare case where AMC-1 fails to find an alternative version for replacement since all the alternative versions are affected by errors. However, the coding bit rates of AMC-1 and AMC-2 do depend on the chosen prediction method. A better prediction will lead a lower bit rate overhead. This point will be examined in Section VI-B.

We observe a significant PSNR improvement of the AMC scheme over the benchmark from these two tables. The improvement becomes higher as the packet loss rate goes higher. When compared with the reference data, the quality of the received stream with AMC is very close to that of the ideal real-time

TABLE II
PSNR VALUE (dB) OF THE RECEIVED FOREMAN QCIF SEQUENCE

	Proposed Schemes		Comparison		References	
Test Condition	**AMC - 1**	**AMC - 2**	**Benchmark**	**PSNR Gain**	**Original**	**Re-encode**
T1	35.69	35.69	35.62	+0.07	35.71	35.70
T2	35.70	35.70	35.60	+0.10	35.71	35.70
T3	35.25	35.25	31.65	+3.60	35.71	35.52
T4	35.31	35.31	30.34	+4.97	35.71	35.47
T5	35.65	35.65	35.12	+0.53	35.71	35.69
T6	35.65	35.65	35.01	+0.64	35.71	35.69
T7	35.55	35.55	34.03	+1.52	35.71	35.66
T8	35.61	35.61	33.95	+1.66	35.71	35.64
T9	35.26	35.26	30.70	+4.56	35.71	35.49
T10	35.27	35.27	30.55	+4.72	35.71	35.43

TABLE III
PSNR VALUE (dB) OF THE RECEIVED AKIYO QCIF SEQUENCE

	Proposed Schemes		Comparison		References	
Test Condition	**AMC - 1**	**AMC - 2**	**Benchmark**	**PSNR Gain**	**Original**	**Re-encode**
T1	38.39	38.39	38.31	+0.08	38.42	38.42
T2	38.40	38.40	38.30	+0.10	38.42	38.42
T3	38.02	38.03	34.26	+3.76	38.42	38.18
T4	38.04	38.05	34.23	+3.81	38.42	38.18
T5	38.32	38.32	37.98	+0.34	38.42	38.40
T6	38.34	38.34	37.90	+0.44	38.42	38.39
T7	38.13	38.13	37.10	+1.03	38.42	38.29
T8	38.18	38.18	36.92	+1.26	38.42	38.27
T3−T9	37.95	37.95	34.15	+3.80	38.42	38.19
T4−T10	38.01	38.02	33.93	+4.08	38.42	38.19

re-encoding method. This indicates that AMC can stop error propagation for offline coded video in a similar manner as feedback-based methods for online coded video, which was discussed in Section II-B.

By enabling FMO, macroblocks in the same packet are scattered in a video frame. The performance of error concealment tools is enhanced and a single packet loss usually has less damage to the overall PSNR. Thus, the benchmark PSNR values with FMO enabled are higher than that without FMO. On the contrary, the PSNR values of the AMC schemes are slightly lower when FMO is enabled. The quality of alternative macroblocks is not affected by the pattern or positioning of previously corrupted macroblocks. But when the corrupted macroblocks are more scattered spatially, the number of macroblocks in subsequent frames that are affected by them generally increases. As a result, the number of alternative macroblocks used in replacement becomes higher and the small mismatch introduced by them slightly degrades the overall R-D tradeoff of received video.

Fig. 11 shows the PSNR values as a function of the frame number in a typical run under test condition T_3 for Foreman QCIF sequence. The PSNR values of 50 frames starting from the fifth frame in a GOP is shown. The two AMC schemes have the identical PSNR performance so that they share the same curve. We see from the figure that the quality of each individual frame of the AMC scheme is significantly better than that of the benchmark. For the benchmark, the frames closer to the I frame have relatively better quality while the quality of subsequent frames drops more and more as propagation errors originated from multiple transmission errors keep adding up. In contrast, the quality of video frames in the AMC scheme is well maintained. To compare the visual performance, several reconstructed frames coded

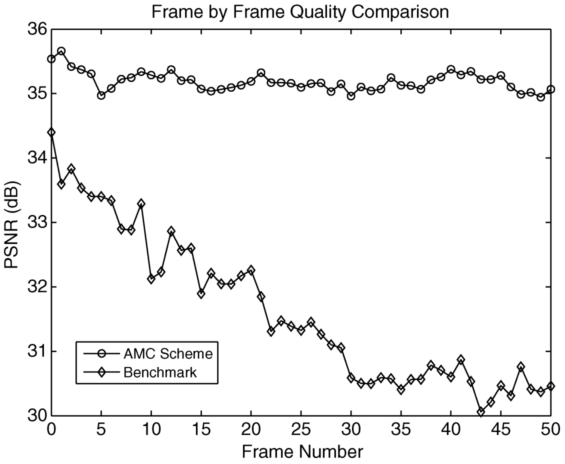


Fig. 11. The PSNR value as a function of the frame number for the foreman QCIF sequence under test condition T_3.

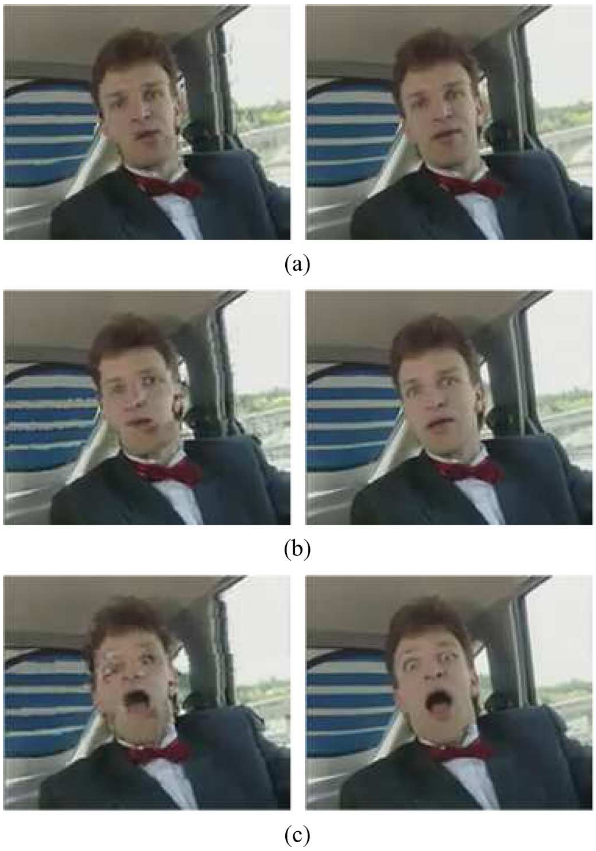


Fig. 12. Comparison of three reconstructed frames coded by the AMC scheme (right) and the benchmark (left). (a) The 10th frame after the error occurrence. (b) 30th frame after the error occurrence. (c) 60th frame after the error occurrence.

by AMC and the benchmark under test case T_3 for the Carphone QCIF sequence are compared in Fig. 12. As shown in the figure, the AMC scheme provides significantly better visual quality than the benchmark. Even at the 60th frame in the GOP, we still do not observe any significant artifact in the reconstructed frame. On the other hand, the reconstructed frame of the benchmark has been greatly distorted. This example clearly demonstrates the advantage of AMC in stopping error propagation.

As shown in the results, the AMC scheme is able to preserve the quality of the received video stream under high error rates and with large GOP sizes (say, 60 frames). In traditional video streaming applications, the GOP size is usually set to be a smaller value in order to confine the damage due to error propagation, which in turn sacrifices coding efficiency as more intra-coded pictures are adopted.

When a smaller GOP size is enforced by some applications, AMC is still an effective method in stopping error propagation for quality improvement of received video. The PSNR values of the AMC scheme, the benchmark and the original PSNR with different GOP sizes under test case T_3 are shown in Fig. 13. We see that a larger PSNR gain is achieved by the AMC scheme. The difference ranges from 2.25 to 3.65 dB for a GOP size from 15 to 50 frames.

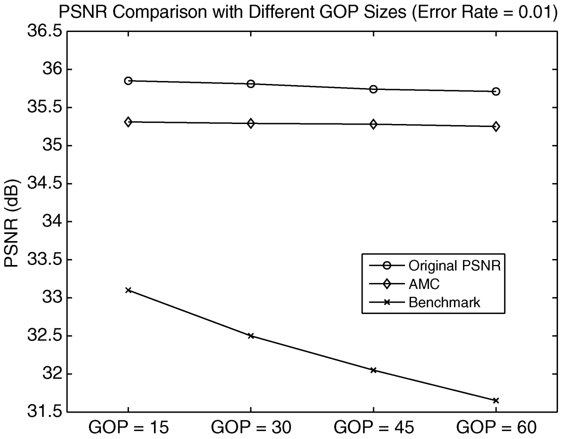


Fig. 13. PSNR value as a function of the GOP size for the Foreman QCIF sequence under test condition T_3.

TABLE IV
BIT RATES (kbps) OF TRANSMITTED VIDEO FOR THE FOREMAN QCIF SEQUENCE

Test Condition	AMC-1	AMC-2	Original
T1	99.70 (0.67%)	99.87 (0.83%)	99.04
T2	99.44 (0.40%)	99.62 (0.59%)	99.04
T3	113.90 (15.1%)	119.07 (20.2%)	99.04
T4	109.41 (10.5%)	121.64 (22.8%)	99.04
T5	102.11 (3.21%)	106.03 (7.10%)	99.04
T6	100.44 (1.45%)	111.25 (12.37%)	99.04
T7	105.02 (6.08%)	109.07 (10.2%)	99.04
T8	103.15 (4.19%)	120.14 (21.3%)	99.04
T9	113.78 (14.9%)	124.77 (26.0%)	99.04
T10	108.02 (9.11%)	137.33 (38.7%)	99.04

B. Bit-Rate Overhead Comparison

Since the bandwidth is an expensive resource for video streaming applications, it is important that an error resilient scheme does not have a large bit rate overhead during transmission. To illustrate this, we consider the coding of the Foreman QCIF sequence as an example. The original bit rate of coded Foreman QCIF sequence is 99.04 kbps as shown in the four column of Table IV. Please note that we simulated the FMO effect by using the standard bit stream and mapping errors to the corresponding macroblocks associated with a specific FMO pattern. Consequently, there is no bit rate overhead of FMO under this particular implementation.

The results on bit rate overhead of the AMC-1 and AMC-2 schemes are shown in the second and the third columns of Table IV. The bit rate overhead is calculated and shown inside the parenthesis. We see that the bit rate overhead of AMC-1 and AMC-2 is very small (i.e., less than 1%) when the packet loss rates are low (e.g., under test conditions T1 and T2). The overhead is only introduced when alternative macroblocks are transmitted for the replacement purpose. Typically, alternative macroblocks consume more bits than the default version since a smaller quantization step size is used to encode them. When the packet loss rate becomes higher, the number of replacements increases and the bit rate overhead increases accordingly. Even

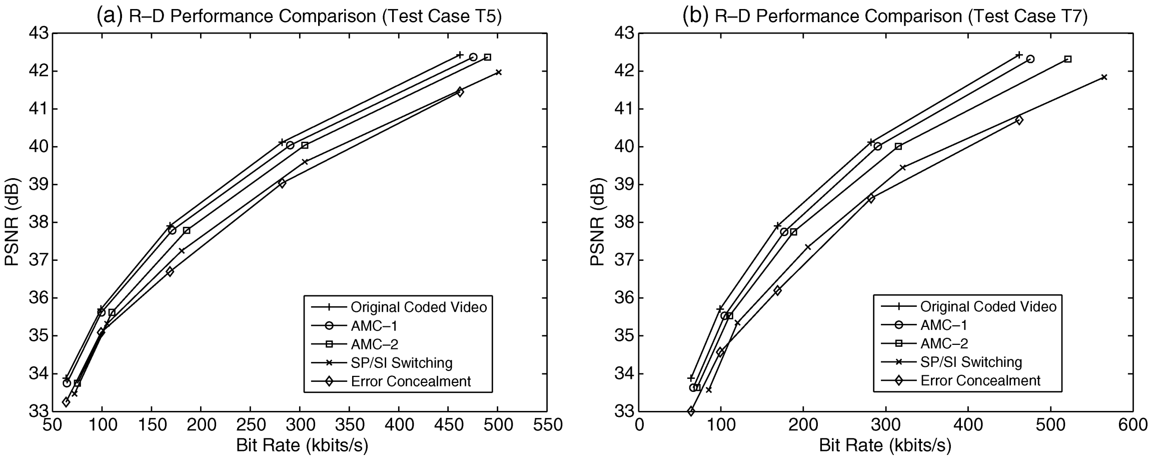


Fig. 14. Quality-rate performance comparison of several methods for the foreman QCIF sequence under test conditions (a) T5 and (b) T7.

at a very high packet loss rate such as 5%, the overall bit rate overhead is still around 10%–20% of the original bit rate for AMC-1, which is very good by considering the achieved error resilience performance shown in Section VI-A. AMC-2 has a larger bit rate overhead than AMC-1 since it would be more difficult to find a good reference from a concealed version of the previous reference frame than a correct reference frame. This is especially true when the error rate is high.

Similarly, as more alternative macroblocks are used and transmitted when FMO is enabled, the bit rate overhead for AMC-1 is higher than the case without FMO. For AMC-2, the relationship between FMO and the bit rate overhead is more complex since the concealed reference frame is used to generate alternative predictions. As FMO improves the performance of error resilience tools, better prediction can be generated. As a result, alternative macroblocks can be coded more efficiently when FMO is enabled. This is especially true when the error rate is relatively high. Thus, although the number of alternative macroblocks transmitted increases when FMO is enabled for AMC-2, the bit rate overhead is smaller when the error rates becomes higher as shown in the result.

As the error rate goes higher, AMC-2 starts to introduce much larger overhead. When the transmission network is suffering from high packet loss rates, sending more bits in these cases actually may worsen the situation. One possible solution is to use AMC-1 that has a lower overhead as compared with AMC-2 when the error rate becomes close to or exceed 5%. A more complete solution to this problem demands further examination in the future.

C. Quality-Rate Performance Comparison

To further demonstrate the error resilience performance of the proposed scheme, we compare AMC-1, AMC-2 and another error resilience method developed with a similar principle. Since there are few existing error resilience tools that utilize the feedback mechanism and work for offline coded video as AMC does, we implemented the idea in [33] and used its SP/SI pictures for error resilience for fair comparison. In this scheme, some frames in the sequence (called switching points) are coded into primary SP pictures or associated secondary SI pictures in offline encoding. Primary SP pictures are always transmitted when there are no errors. When the sender receives the feedback about an error, SI pictures (or slices) are transmitted instead of the primary SP picture to stop error propagation at these switching points. We further improve this scheme by allowing the use of secondary SP pictures coded using long term reference frames to reduce the bit rate overhead. We call this scheme the SP/SI switching scheme. Since primary SP pictures affect coding efficiency and error propagation can only be stopped at these switching points, it is critical to select the frequency of switching points in a sequence properly. The optimal frequencies for test conditions T5 and T7 are found to be one SP/SI switching point per 7 and 4 frames, respectively. They are used in the SP/SI switching method in the corresponding experiments.

The quality-rate performance of methods mentioned above under test conditions T5 and T7 is compared in Fig. 14. The curves of the original coded video (i.e., with no transmission errors) and the reconstructed video using error concealment are also shown in the same figure. It is obvious that the original coded video has the best performance. However, the gap between the original coded video and the proposed AMC1/AMC2 is very small even in the presence of erroneous channels. AMC1 and AMC2 have much better performance than SP/SI switching scheme. The SP/SI switching method suffers from coding efficiency degradation due to the use of primary SP pictures. They are also less effective in reducing error propagation since it can only be done at fixed switching points which are preselected during offline encoding. In contrast, coding efficiency is preserved for the most part of the video sequence in AMC1 and AMC2 except for the macroblocks being replaced. Furthermore, error propagation can be stopped immediately after the sender receives the error feedback, which can be done anywhere in the compressed sequence. Finally, we see that AMC1 has slightly better performance than AMC2, which is achieved at the expense of a larger memory buffer that allows multiple reference frames.

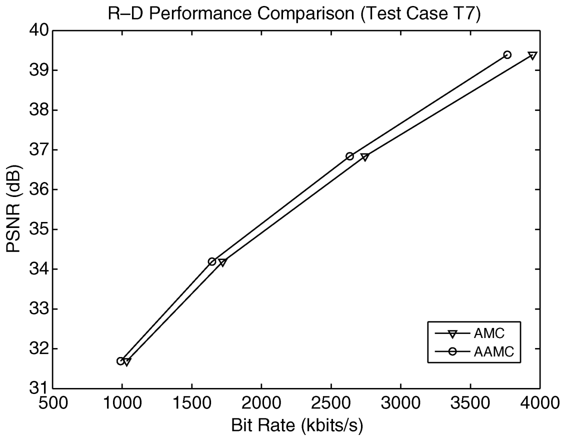


Fig. 15. Comparison of the R-D performance of AMC and AAMC for the mobile CIF sequence.

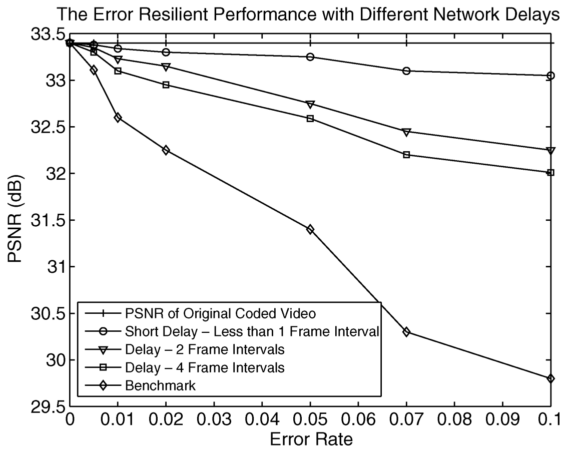


Fig. 16. Error resilient performance for the foreman QCIF sequence under different network delays.

D. Adaptive AMC Schemes

In this subsection, we compare the R-D performance of the AMC-1 scheme with and without using adaptive quantization selection and adaptive bit stream replacement. They are denoted by AMC and AAMC, respectively. The Mobile CIF sequence was used in the test under test case T_7. As shown in Fig. 15, AAMC yields better R-D performance than AMC. Most of the improvements are attributed to reduced overhead in bit rates.

E. Long Delay Scenarios

Finally, we test the AMC scheme under different network delay scenarios. The delay is defined to be the interval between sender's transmitting a corrupted frame and the reception of its corresponding error message. It is measured in terms of the number of frame intervals. For example, when the frame rate is 15 fps, one frame interval is 66 ms. As shown in Fig. 16, the AMC scheme provides good error resilient performance under different network delays. The PSNR degradation with longer delays is caused by the fact that errors already propagate in frames that have been transmitted before the sender receives the error message from the receiver.

VII. Conclusion and Future Work

A novel error resilient H.264 video coding scheme designed to reduce error propagation caused by transmission errors for offline coded video was proposed for streaming applications. The new scheme was shown to provide excellent error resilient performance. Since it can work effectively under a broad range of error rates and network delays, the proposed scheme offers a great tool to maintain the streamed H.264 video at a high quality level in an error-prone environment. The proposed scheme only modifies the transmitted stream whenever an error is detected. Thus, when there is no transmission error, there is no bit rate overhead. Even if there are transmission errors, the bit rate overhead is still low. Furthermore, adaptive quantization scheme and bit stream replacement were developed to further improve the performance of the proposed AMC scheme. The AMC scheme does not demand any significant syntax changes for the H.264 standard. The general idea can be adapted to any other coding standards through slight modification. In the future, we will continue to investigate a generic error resilience framework for robust H.264 video streaming by exploiting its rich syntax.

Acknowledgment

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] *Coding of Audiovisual Objects-Part 10: Advanced Video Coding*, ISO/IEC 14496-10:2003, ITU-T Rec. H.264, 2003.

[2] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, 2004.

[3] N. Kamaci and Y. Altunbasak, "Performance comparison of the emerging H.264 video coding standard with the existing standards," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2003, vol. 1, pp. 345–348.

[4] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[5] P. Lambert, W. De Neve, P. De Neve, I. Moerman, P. Demeester, and R. Van de Walle, "Rate-distortion performance of H.264/AVC compared to state-of-the art video codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 134–140, Jan. 2006.

[6] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Process. Mag.*, vol. 17, no. 4, pp. 61–82, Jul. 2000.

[7] N. Färber, K. Stuhlmüller, and B. Girod, "Analysis of error propagation in hybrid video coding with application to error resilience," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 1999, vol. 2, pp. 550–554.

[8] J. Wen and J. Villasenor, "A class of reversible variable length codes for robust image and video coding," in *Proc. IEEE Int. Conf. Image Process.*, 1997, vol. 2, pp. 65–68.

[9] T. J. Ferguson and J. H. Rabinowitz, "Self-synchronizing Huffman codes," *IEEE Trans. Inf. Theory*, vol. IT-30, no. 4, pp. 687–693, Jul. 1984.

[10] W. M. Lam and A. R. Reibman, "Self-synchronizing variable length codes for image transmission," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 1992, vol. 3, pp. 477–480.

[11] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 5, pp. 771–781, Jun. 1989.

[12] I. Kondi, F. Ishtiaq, and A. Katsaggelos, "Joint source-channel coding for scalable video," in *Proc. IEEE Int. Conf. Image Process.*, Jan. 2000, pp. 324–335.

[13] U. Horn, B. Girod, and B. Belzer, "Scalable video coding with multiscale motion compensation and unequal error protection," in *Proc. Int. Symp. Multimedia Commun. Video Coding*, Oct. 1995, pp. 475–482.

[14] R. M. Pelz, "An unequal error protected p8 kbit/s video transmission for DECT," in *Proc. Vehicular Technol. Conf*, 1994, pp. 1020–1024.

[15] L.-W. Kang and J.-J. Leou, "An error resilient coding scheme for H.264 video transmission based on data embedding," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, May 2004, vol. 3, pp. 257–260.

[16] Y. O. Park, C.-S. Kim, and S.-U. Lee, "Multi-hypothesis error concealment algorithm for H.26l video," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2003, vol. 3, pp. 465–468.

[17] P. Haskell and D. Messerschmitt, "Resynchronization of motion—Compensated video affected by ATM cell loss," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, Mar. 1992, vol. 3, pp. 545–548.

[18] J. Liao and J. Villasenor, "Adaptive intra update for video coding over noisy channels," in *Proc. IEEE Int. Conf. Image Process.*, 1996, vol. 3, pp. 763–766.

[19] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.

[20] Y. Wang, Q. F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. Commun.*, vol. 41, no. 10, pp. 1544–1551, Oct. 1993.

[21] M.-J. Chen, L.-G. Chen, and R.-M. Weng, "Error concealment of lost motion vectors with overlapped motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, pp. 560–563, Jun. 1997.

[22] W.-Y. Kung, C.-S. Kim, and C.-C. J. Kuo, "A spatial-domain error concealment method with edge recovery and selective directional interpolation," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2003, vol. 2, pp. 145–148.

[23] C. S. Park, J. Ye, and S. U. Lee, "Lost motion vector recovery algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 1994, pp. 229–232.

[24] Z. Wang, Y. Yu, and D. Zhang, "Best neighborhood matching: An information loss restoration technique for block-based image coding systems," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1056–1061, Jul. 1998.

[25] W. Wada, "Selective recovery of video packet loss using error concealment," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 7, pp. 807–814, Jun. 1989.

[26] S. Fukunaga, T. Nakai, and H. Inoue, "Error resilient video coding by dynamic replacing of reference pictures," in *Proc. IEEE GLOBECOM*, Nov. 1996, vol. 3, pp. 1503–1508.

[27] E. Steinbach, N. Färber, and B. Girod, "Standard compatiable extension of H.263 for robust video transmission in mobile environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 6, pp. 872–887, Dec. 1997.

[28] T. C. Wang, H. C. F. , and L. G. Chen, "Low-delay and error-robust video transmission for video communcations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1049–1058, Dec. 2002.

[29] *Information Technology-Coding of Audio-Visual Objects, Part2—Visual*, ISO/IEC 14496-2, 2002.

[30] Y. K. Wang, M. M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," in *Proc. IEEE Int. Conf. Image Process.*, 2002, vol. 2, pp. 22–25.

[31] S. Wenger and M. Horowitz, *FMO: Flexible Macroblock Ordering* May 2002, Doc. JVT-C089.

[32] S. Wenger and M. Horowitz, "Flexible MB ordering—A new error resilience tool for IP-based video," presented at the Int. Workshop Digital Commun., Capri, Italy, Sep. 2002.

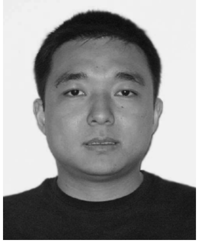
[33] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 637–644, Jul. 2003.

[34] X. Zhou, W.-Y. Kung, and C.-C. J. Kuo, "Error resilient H.264 video with SP/SI coded macroblocks," presented at the 14th Int. Packet Video Workshop, Irvine, CA, Dec. 2004.

[35] X. Zhou, W.-Y. Kung, and C.-C. J. Kuo, "A robust H.264 video streaming scheme for portable devices," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, vol. 4, pp. 3263–3266.

[36] (2008). JVT Reference Software ver. 8.6 [Online]. Available: http://bs.hhi.de/seuhring/tml/download

[37] L. N. Kanal and A. R. K. Sastry, "Models for channels with memory and their applications to error control," *Proc. IEEE*, vol. 66, no. 7, pp. 724–744, Jul. 1978.



Xiaosong Zhou (M'00) received the B.S. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2000, the M.S. degree in electrical and computer engineering from Purdue University, Indianapolis, IN, in 2002, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 2006.

He worked as summer Intern at Intel China Research Center in Beijing in 2002 and Intel Shanghai Software Laboratory in 2003. He also worked as a Research Intern at IBM Watson Research Center, New York, in 2004. He is currently a Senior Software Engineer at Apple Inc., Cupertino, CA. His research interest are in the areas of multimedia data compression, streaming and digital video processing/analysis.

Dr. Zhou is a member of the Society of Motion Picture Television Engineers. He served as technical reviewer for various IEEE conference and journal papers in 2003–2007.

C.-C. Jay Kuo (S'83–M'86–SM'92–F'99) received the B.S. degree from the National Taiwan University, Taipei, in 1980 and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1985 and 1987, respectively, all in Electrical Engineering. He is Director of the Signal and Image Processing Institute (SIPI) and Professor of Electrical Engineering, Computer Science and Mathematics at the University of Southern California (USC). His research interests are in the areas of digital image/video analysis and modeling, multimedia data compression, communication and networking, and biological signal/image processing. He is coauthor of about 140 journal papers, 730 conference papers and 9 books.

Dr. Kuo is a Fellow of SPIE and a member of ACM. He is Editor-in-Chief for the *Journal of Visual Communication and Image Representation,* and Editor for the *Journal of Information Science and Engineering*, *LNCS Transactions on Data Hiding and Multimedia Security* and the EURASIP *Journal of Applied Signal Processing*. He was on the Editorial Board of the IEEE *Signal Processing Magazine* in 2003–2004. He served as Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING in 1995–98, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 1995–1997 and IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING in 2001–2003. Dr. Kuo received the National Science Foundation Young Investigator Award (NYI) and Presidential Faculty Fellow (PFF) Award in 1992 and 1993, respectively. He received the Northrop Junior Faculty Research Award from the USC Viterbi School of Engineering in 1994. He received the best paper award from the multimedia communication Technical Committee of the IEEE Communication Society in 2005. He is an IEEE Signal Processing Society Distinguished Lecturer in 2006. He is also Advisor to the SMPTE (Society of Motion Picture Television Engineers)-USC student chapter.