Correspondence

A Concealment Method for Shape Information in MPEG-4 Coded Video Sequences

Shahram Shirani, Berna Erol, and Faouzi Kossentini

***Abstract*—In this paper, we propose a new method for error concealment of shape information in MPEG-4 video bit streams that are transmitted over error prone channels. The proposed method employs a MAP estimator with an MRF as the image *a priori* model. The MRF is designed for binary shape information and its parameters are adapted based on the information of neighboring blocks. Our experimental results show that the proposed concealment method restores missing shape blocks with high accuracy. Compared to the median filtering method, our method restores 20% more missing shape data, with a much greater subjective improvement. The proposed algorithm requires relatively small number of integer multiplications and additions and simple logic operations, making it suitable for real-time implementations.**

***Index Terms*—Error concealment, error resilience coding, Markov random fields, MPEG-4, post-processing, shape.**

I. Introduction

COMPRESSED multimedia data streams transmitted over error prone channels, such as wireless networks and the Internet, are usually corrupted by channel errors. MPEG-4, which is the latest ISO visual coding standard, offers error resilience tools that help localization and isolation of the erroneous data and partial recovery of the remaining data [1]. Concealment of the errors, however, is not specified in the MPEG-4 standard, and it is therefore a subject of ongoing research. MPEG-4 supports an object-based representation of video by allowing the coding of the shape information of arbitrarily shaped video objects along with the objects' texture and motion information. The concealment of errors in the texture information of the MPEG-4 coded video is similar to what has been vastly investigated in the frameworks of MPEG-2 and H263 [2]–[6]. However, efficient concealment of the shape information in the MPEG-4 coded video has, so far, not been addressed. In this paper, we propose a method for concealment of errors in the shape information of an MPEG-4 coded video object. An adaptive Markov Random Field (MRF), which is designed for binary shape information, is proposed as the image *a priori* model. The proposed image model is used along with a Maximum a Posteriori (MAP) estimator to recover the missing shape information. The proposed concealment method successfully reconstructs the missing shape data with good computation-performance tradeoffs.

II. MPEG-4: Video Compression and Error Resilience

MPEG-4 achieves an object-based representation by defining audio-visual objects and coding them into separate bit streams [1], [7], [8]. A video object (VO) is an arbitrarily shaped video segment that has a semantic meaning. Temporal instances of video objects are referred to as video object planes (VOPs), which are defined by their textures (luminance and chrominance values) and shapes. Similar to MPEG-1 and MPEG-2, MPEG-4 supports intra coded (I-), temporally predicted (P-), and bi-directionally predicted (B-) VOPs.

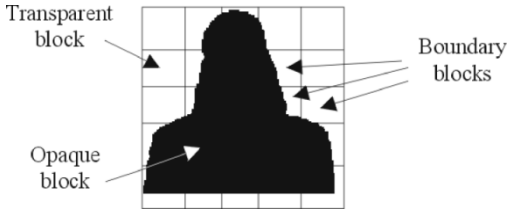

Fig. 1. Binary alpha plane.

In MPEG-4, VOPs are divided into macroblocks, such that they are represented with the minimum number of macroblocks within a bounding rectangle. Texture coding of macroblocks is very similar to the coding of frames in MPEG-2: The luminance and chrominance blocks of each macroblock are coded using motion compensation, DCT, quantization, and variable length coding (VLC).

The shape of a VOP is described by a binary alpha plane as shown in Fig. 1, which indicates whether or not a pixel belongs to a VOP. The binary alpha planes are divided into 16×16 blocks. The shape data associated with each of these 16×16 blocks is transmitted in the bit stream along with the texture information that corresponds to the same area. The blocks that are inside the VOP are transmitted as opaque and the blocks that are outside the VOP are transmitted as transparent blocks in the bit stream. The boundary blocks, i.e., blocks that contain pixels both inside and outside the VOP, are either intra or inter coded. In intra shape coding, the pixels inside the boundary blocks are raster order scanned and the corresponding binary shape data is context-based arithmetic coded. In inter shape coding, the boundary block is first predicted from the temporally previous or future VOP via motion estimation and compensation in integer pixel accuracy. Then, the associated shape motion vector is coded predictively, and the difference between the current and the predicted shape blocks is arithmetic coded.

The compressed video signal is extremely vulnerable to transmission errors. This is mainly a result of using prediction as well as variable length codes. MPEG-4 offers error resilience tools to localize the effects of errors, re-establish synchronization, and salvage the erroneous data. These tools can be divided into three groups: video packetization, data partitioning, and reversible VLC [9], [10].

When decoding a corrupted bit stream, a video decoder loses synchronization with the encoder, that is, it is unable to identify the precise location in the VOP where the current data belongs. MPEG-4 employs periodic resynchronization markers, which are different from all the valid code words, to restore synchronization between the decoder and the encoder. The macroblock data between two resynchronization markers is referred to as a video packet. The number of macroblocks in an MPEG-4 video packet may be variable, depending on the number of bits required to represent each macroblock. Each video packet contains information, such as the macroblock number and the quantization parameter value, that is necessary to restart the decoding operation in the case of an error.

Typically, when synchronization is lost, all the motion, shape, and texture data between two resynchronization words are discarded, since

Manuscript received August 19, 1999; revised May 19, 2000. This work was supported by NSERC. The associate editor coordinating the review of this paper and approving it for publication was Dr. Chung-Sheng Li.

S. Shirani is with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada

B. Erol and F. Kossentini are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: faouzi@ece.ubc.ca).

Publisher Item Identifier S 1520-9210(00)07021-8.

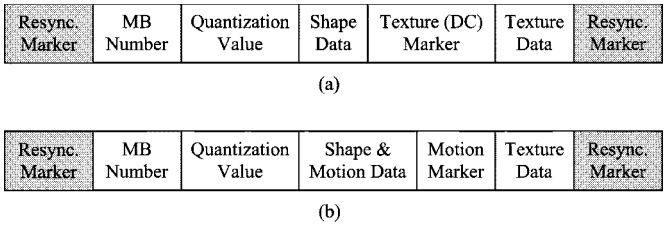

Fig. 2. Video packet structure of MPEG–4 in (a) I-VOPs and (b) P-VOPs.

the decoder does not know the exact location of the error. MPEG-4 supports the data partitioning mode, which allows the separation of motion and shape data from the texture data in a video packet. In I-VOPs, the shape information is separated from the texture data using the "dc marker." In P-VOPs, the motion and shape data are separated from the texture data using the "motion marker." Both cases are illustrated in Fig. 2. In general, an error is more likely to occur in the texture data, because the shape and motion data is represented with fewer bits than the texture data. Therefore, using data partitioning, in most cases the motion and shape information can be recovered. Hence, even if the texture information is lost, the other information can be used to conceal texture errors.

When the decoder detects an error while decoding the VLC data, it looses synchronization with the bit stream and discards all the data until the next resynchronization point. MPEG-4 addresses this problem by employing Reversible VLC's (RVLC's) that can be decoded in both the forward and reverse directions. This enables the decoder to better localize the error between two resynchronization points.

III. Concealment of Channel Errors in MPEG-4 Coded Video Data

We next discuss the various scenarios that can occur when decoding an erroneous MPEG-4 coded video bit stream. It is assumed that the error resilience tools mentioned above are employed in the MPEG-4 bit stream. We consider the I-VOPs and P-VOPs separately.

In the P-VOPs, the shape and motion data are separated from the texture data with a motion marker as seen in Fig. 2. If an error occurs in the texture part of a video packet, the decoder can use the motion information to replace the missing texture with the texture in the previous VOP. If the error occurs in the motion/shape part, then the whole video packet is discarded. A simple concealment can be done by replacing the shape and texture of the missing macroblocks with those of macroblocks corresponding to the same location in the previous VOP. An alternative method is to estimate the missing motion vectors from the motion vectors of surrounding macroblocks at the decoder and use the estimated vectors to replace the missing texture from the previous VOP [11].

Concealment of errors in I-VOPs is more critical than that in P-VOPs, simply because the I-VOPs are used for prediction and thus the errors in I-VOPs propagate. In I-VOPs, the shape data is separated from the texture information with a texture marker within a video packet as illustrated in Fig. 2. If an error occurs in the texture part of a video packet, only the texture information is lost. If an error occurs in the shape part, then both the shape and texture information are lost. Various methods have been proposed for error concealment of texture information within the frameworks of the H.263 and MPEG-2 standards. Since the representation of texture in MPEG-4 is very similar to that in H.263 and MPEG-2, those methods can be applied to an MPEG-4 coded bit stream as well. However, not much research has been done on the concealment of errors in the shape data of I-VOPs. In this paper, we propose a method for concealing the effects of shape errors in I-VOPs.

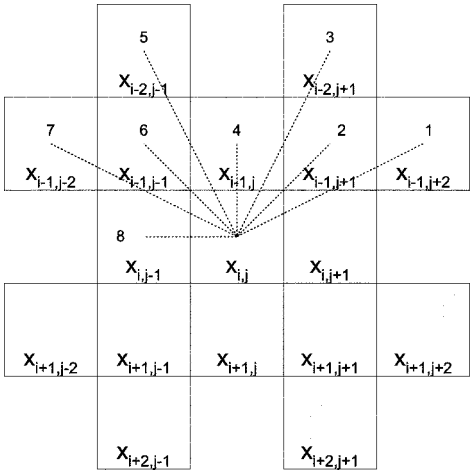

Fig. 3. A pixel, its clique c, and the eight directions. The complement of the clique c' is the dark area.

IV. Proposed Method

Statistical methods for error concealment assume that the pixel values in an image or video signal are realizations of an underlying statistical model. The Bayesian approach provides a framework for incorporating the *a priori* information by selecting the image model. Then, a Maximum a Posteriori (MAP) estimation yields the most likely image given the observed image data [3]. Here, we use a MAP estimator for restoration of missing shape information. Since the shape information is binary, we propose an appropriate form of Markov random field (MRF) as the image model.

The MAP estimation of missing data in an image assuming MRF as the image *a priori* model can be expressed by the following minimization problem

$$\hat{\mathbf{X}} = \min_{x_{i,j}} \sum_{i,j \in \mathcal{M}} V(x_{i,j}) \tag{1}$$

where $\mathcal{M}$ is the set of all missing pixels in the image and V is the potential function. The potential function characterizes the relationship between a group of pixels by assigning larger costs to configurations of pixels which are less likely to occur. The choice of the potential function is crucial to the performance of the image model. Commonly, the potential functions are selected to be of the form

$$\sum_{c \in C} V(x_{i,j}) = \sum_{k,l \in c} w_{i,j-k,l} \rho(x_{i,j} - x_{k,l})) \tag{2}$$

where c is the clique, C is the set of all cliques throughout the image, ρ is a function called the cost function, and $w_{i,j-k,l}$ is the weight assigned to the difference between the pixel values $x_{i,j}$ and $x_{k,l}$ [12]. The cost function $\rho(\cdot)$, in fact, encourages the pixels that are spatially close to each other to have the same values. The shape information is binary and $x_{i,j}$ can only assume two values. Therefore, we select the cost function of the following form

$$\rho(x) = \begin{cases} \beta: & x \neq 0 \\ 0: & x = 0 \end{cases}$$

where β is a positive constant.

For the clique, we adopt an eight pixel neighborhood shown in Fig. 3. The weight corresponding to the difference between a pixel and one of the pixels in its clique ($w_{i,j-k,l}$ in (2)) is selected adaptively, based on the likelihood of an edge in the direction of the subject pair of pixels.

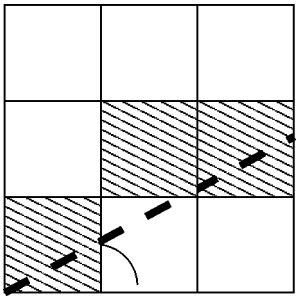

Fig. 4. A 3 × 3 window, the border pixels in it (shaded) and the best line-fit.

The rationale behind this selection is to weigh more the difference between the pixels in a direction which will cause the values of the pixels in that direction to be the same. Assuming the values $w_{i,j\to k,l}$ to be integers, the estimated value of a pixel given the values of the pixels in its clique, will be

$$\begin{aligned}\hat{x}_{i,j} &= \min_{x_{i,j}} \sum_{(k,l)\in c\cup c'} w_{i,j\to k,l}\rho(x_{i,j}-x_{k,l}) \\ &= \min_{x_{i,j}}[\underbrace{\rho(x_{i,j}-x_{i,j+1})+\cdots+\rho(x_{i,j}-x_{i,j+1})}_{w_{i,j\to i,j+1}\text{ times}} \\ &\quad +\cdots+\underbrace{\rho(x_{ij}-x_{i+1,j+2})+\cdots+(x_{i,j}-x_{i+1,j+2})}_{w_{i,j\to+1,j+2}\text{ times}}]\end{aligned}$$

where c' is the complement of the clique shown in Fig. 3. To minimize the value of the potential function, the number of terms with an estimated value that is different from that of the neighboring pixel, should be minimized. Therefore, the estimated value should be equal to the median of the following vector

$$\begin{aligned}\hat{x}_{i,j} = \text{median}[&\underbrace{x_{i,j+1},\cdots,x_{i,j+1}}_{w_{i,j\to i,j+1}\text{ times}}, \\ &\cdots,\underbrace{x_{i+1,j+2},\cdots,x_{i+1,j+2}}_{w_{i,j\to i+1,j+2}\text{ times}}].\end{aligned} \tag{3}$$

The likelihood of edges in each of the eight directions is computed using blocks around the missing block. In this way, the available shape information in a larger area is exploited in the concealment process. To determine the likelihood of edges in each of the eight directions, edges in the blocks surrounding the missing block, whose directions imply that they pass through the missing block, are determined.

Since the edge information of the shape data is embedded in its borders, we first separate the borders of the shape data in the adjacent blocks. To do this, we use a morphological transform called the boundary transform [13]. If all the four neighboring pixels (above, bellow, left and right) of a pixel are inside the shape, then the pixel is declared inside the shape. Otherwise, the pixel is assigned to the border of the shape. A 3 × 3 window is then centered at each border pixel and the angle of the best line-fit to the border pixels in the window is computed. This in fact gives the direction of the edge at the pixel in the center of the window. Fig. 4 shows a typical window and the best line-fit and the angle of the line. There are eight counters corresponding to eight directions as shown in Fig. 3. The counter corresponding to the direction of the detected edge (best line-fit) is incremented if the extension of the best line fit passes through the missing block. The procedure is repeated for all the border pixels in the blocks to the right, left, below and above the missing block (if applicable). The weights required in (3) are obtained by

$$w_{i,j-k,l} = c_m \tag{4}$$

Fig. 5. Shape of the first I-VOP of the AKIYO video object.

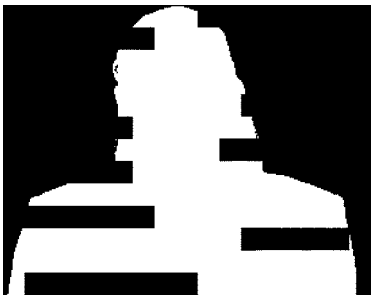

Fig. 6. Shape of the AKIYO video object missing 30% of the shape blocks.

Fig. 7. Reconstructed shape of the first I-VOP of the AKIYO video object.

where c_m is the counter corresponding to direction m, and the direction m corresponds to the direction formed by (i,j) and (k,l). Finally, the proposed shape error concealment method can be summarized as follows.

1) Determine the edges in the neighboring blocks and assign them to eight equally spaced directions. Compute the corresponding counter for each direction,
2) Use (4) to find a set of weights for each missing block,
3) Use (3) to obtain an estimate of each missing pixel employing the weights obtained in the previous step, and
4) Iteratively re-estimate the missing pixels using (3) until convergence.

In the case where adjacent blocks are lost, the reconstruction algorithm is applied recursively. Blocks with the maximum number of correctly decoded neighboring blocks are reconstructed first, and the rest of the blocks are reconstructed recursively. This guarantees that the best possible estimation accuracy is achieved.

The computational load of the proposed method consists of the computations required for finding the boundary pixels, finding the slope of the best line-fit for each of the boundary pixels, and the estimation of missing pixels using (3). For every pixel belonging to the shape in a neighboring block of a missing block, three XOR operations are required to determine whether or not the pixel belongs to the border. Finding the slope of the best line-fit for each of the boundary pixels in the 3 × 3 window requires approximately ten integer multiplications

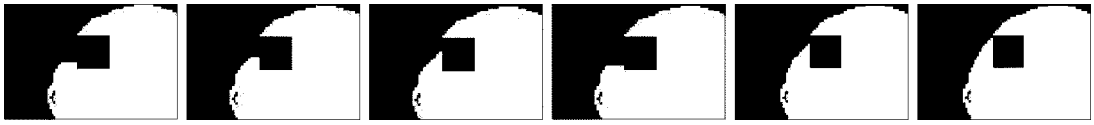

Fig. 8. Intermediate results for the reconstruction of block number 23 in the first I-VOP of the AKIYO video object after (a) 2, (b) 3, (c) 6, (d) 9, (e) 11, and (f) 15 iterations.

and ten integer additions per border pixel. These computations are noniterative and the total number of operations depends on the complexity of the shape in the blocks around a missing block. In our simulations, the average number of border pixels per neighboring block is 22. Therefore, this stage requires approximately $22 \times 4 \times 10 = 880$ addition and 880 multiplication operations per missing block. The last part of the reconstruction operation, which is the estimation of missing pixels using (3), is iterative. Since the shape information is binary data, the median operation can be implemented with two simple counters (one for 0 and one for 1), which requires only 16 additions. On average, six iterations are required for the proposed reconstruction algorithm to converge for a block. Considering that the median operation needs to be performed for every missing pixel, $16 \times 6 \times 16 \times 16 = 24\,576$ additions are required per missing block for the last part of the algorithm. For a video object plane with size of 160 × 160 and 30% blocks missing, $10 \times 10 \times 0.3 \times (24576 + 880) = 763\,680$ additions and $10 \times 10 \times 0.3 \times 880 = 26\,400$ multiplications are necessary in order to reconstruct its shape. Computing IDCT for a 160×160 VOP requires approximately 250K additions and 70K multiplications [14]. Considering that IDCT takes approximately 50% of the MPEG-4 decoding operation, the computations required for our proposed shape reconstruction algorithm are comparable to that of MPEG-4 decoding. Therefore, our algorithm is suitable for real-time implementations.

V. Experimental Results

We tested our proposed shape reconstruction method on various corrupted MPEG-4 bit streams which contain video objects coded at 5000 bits/VOP. The error resilience tools of MPEG-4, more specifically using video packets and data partitioning, are applied during the encoding. Error simulation is performed at the decoder: While decoding the bit stream, the decoder ignores the video packets randomly with a given packet loss percentage. Then, the proposed concealment algorithm is applied to the erroneous I-VOP binary shape data to recover the missing shape blocks.

The similarity of the erroneous and error concealed shape data to the original shape data is measured using the following ratio

$$\eta = 1 - \frac{n_d}{n_t} \tag{5}$$

where n_d is the number of pixels that are different between the restored shape and the original one, i.e., the Hamming distance, and n_t is the total number of pixels in the bounding box.

Fig. 5 shows the binary shape data of the first I-VOP of the AKIYO video object. The bounding box of the VOP is of size 272×208. The length of each video packet is selected to be 500 bits. As mentioned before, the number of blocks in each packet is different and depends on the size of coded data of each block. For example, in this case, the first ten packets have 9, 14, 3, 14, 2, 15, 2, 2, 13, and 3 macroblocks, respectively. Fig. 6 shows the shape of the I-VOP with 30% of the shape blocks missing, corresponding to 6 missing video packets. The result of the proposed error concealment method is presented in Fig. 7. The shape similarity measure η is 88% for the erroneous shape data shown in Fig. 6, and 99% after using the proposed shape concealment method. Fig. 8(a)–(f) show the intermediate results for the reconstruction of block number 23 of the VOP object shown in Fig. 6 after 2, 3, 6, 9, 11, and 15 iterations. The block is located on the left side of the head of the object.

Fig. 9. Shape of the first I-VOP of the BREAM video object.

Fig. 10. Shape of the BREAM video object missing 25% of the shape blocks.

Fig. 11. Reconstructed shape of the first I-VOP of the BREAM video object.

Next, we present the performance of our concealment technique on the first and the 120th VOPs of the BREAM video object and on the first VOP of the WEATHER video object. Fig. 9 shows the binary shape data of the first VOP of the video object BREAM. The size of the bounding rectangle is 272 × 192. The size of the video packets is selected to be 700 bits. The coded I-VOP contains 44 video packets. Fig. 10 shows the binary shape information of the I-VOP with 25% of the blocks missing. The shape data after concealment is given in Fig. 11. The similarity measure η is 86% before, and 99% after, concealment.

The 120th VOP of the BREAM video object is given in Fig. 12, which is the size of 144×192. The video object consists of 19 packets and each video packet contains 700 bits. Figs. 13 and 14 show the same VOP with 29% of the blocks missing and after applying our reconstruction method, respectively. η is 81% before, and 98% after, concealment.

Fig. 12. Shape of the 120th VOP of the BREAM video object.

Fig. 13. Shape of the 120th VOP of the BREAM video object missing 29% of the shape blocks.

Fig. 14. Reconstructed shape of the 120th VOP of the BREAM video object.

Fig. 15. Shape of the first I-VOP of the WEATHER video object.

The first VOP of the WEATHER video object is given in Fig. 15. The size of the VOP is 160×224. The size of the video packets is set to 1000 bits. Fig. 16 shows the shape of the I-VOP when 35% of the macroblocks are missing. Fig. 17 shows the result of the proposed error concealment method. The value of η is 87% and 99% before and after the concealment of shape information, respectively.

Finally, we compare our concealment method with an iterative median filtering method where each pixel in a missing block is set to the median of pixels around it (in its clique and its complement) until the content of a missing block does not change in two consecutive iterations. Figs. 18 and 19 show the result for the first VOP of the AKIYO and the 120th VOP of the BREAM video objects, respectively. As can be

Fig. 16. Shape of the WEATHER video object missing 35% of the shape blocks.

Fig. 17. Reconstructed shape of the first I-VOP of the WEATHER video object.

Fig. 18. Reconstructed shape of the first I-VOP of the AKIYO video object using the median method.

Fig. 19. Reconstructed shape of the 120th VOP of the BREAM video object using the median method.

seen from the figures, the median method is unable to recover a large portion of the missing pixels. If we define the reconstruction improvement as $(M_p - M_m)/M$, where M is the number of pixels that are damaged as a result of a packet loss, M_p and M_m are the number of pixels restored by our proposed method and the median method, respectively, the improvement of our proposed method over the median method is 17% for the AKIYO and 22% for the BREAM video objects. Comparing Figs. 18 and 19 with Figs. 7 and 14, it is clear that the subjective improvement of our proposed method over the median method is very significant.

VI. Conclusion

This paper presents an error concealment method for shape information in the MPEG-4 coded video sequences. A maximum *a Posteriori* estimator, which employs an adaptive Markov random field, is used to restore the missing shape information. The proposed concealment method successfully reconstructs the missing shape data providing about 20% improvement compared to median filtering method. The subjective improvement achieved by our algorithm is even much more significant. Moreover, reasonable computational complexity of our algorithm makes it suitable for real-time applications.

References

[1] *Coding of audio-visual objects: Video*, ISO/IEC JTC!/SC29/WG11, Jan. 1999.

[2] Y. Wang, Q. F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. Commun.*, vol. 41, pp. 1544–1551, Oct. 1993.

[3] P. Salama, N. Shroff, E. J. Coyle, and E. J. Delp, "Error concealment in encoded video streams," in *Signal Recovery Techniques for Image and Video Compression and Transmission*, A. K. Katsaggelos, Ed. Boston, MA: Kluwer, 1998.

[4] W. Kwok and H. Sun, "Multi-directional interpolation for spatial error concealment," *IEEE Trans. Consum. Electron.*, vol. 39, pp. 455–460, Aug. 1993.

[5] Y. Wang and Q. F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, pp. 974–997, May 1998.

[6] S. Shirani, F. Kossentini, and R. Ward, "An adaptive markov random field based error concealment method for video communication in an error prone environment," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, vol. VI, Mar. 1999, pp. 3117–3120.

[7] *MPEG-4 overview*, http://drogo.cselt.stet.it/mpeg/standarads/mpeg-4/mpeg-4.htm, July 1998.

[8] B. Erol, A. Dumitras, and F. Kossentini, *Emerging MPEG Standards: MPEG-4 and MPEG-7, in Handbook of Image and Video Processing*: Academic Press, 2000.

[9] R. Talluri, "Error resilient video coding in the MPEG-4 standard," *IEEE Commun. Mag.*, vol. 26, pp. 112–119, June 1998.

[10] J. Liang and R. Talluri, "Tools for robust image and video coding in JPEG2000 and MPEG-4 standards," *Proc. SPIE*, vol. 3653, pp. 40–51, Jan. 1999.

[11] M. R. Frater, W. S. Lee, and J. F. Arnold, "Error concealment for arbitrary shaped video objects," in *Proc. Int. Conf. Image Processing*, vol. 3, Chicago, Oct. 1998, pp. 507–511.

[12] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, Nov. 1984.

[13] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[14] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images," *Trans. IEICE*, pp. 1095–1097, 1988.