# A Robust Error Concealment Technique using Data Hiding for Image and Video Transmission over Lossy Channels

Chowdary B. Adsumilli, *Student Member, IEEE,* Mylène C. Q. Farias, *Member, IEEE,* Sanjit K. Mitra, *Life Fellow, IEEE,* and Marco Carli, *Senior Member, IEEE*

***Abstract*— A robust error concealment scheme using data hiding which aims at achieving high perceptual quality of images and video at the end user despite channel losses is proposed. The scheme involves embedding a low resolution version of each image or video frame into itself using spread spectrum watermarking, and extracting the embedded watermark from the received video frame and using it as a reference for reconstruction of the parent image or frame, thus detecting and concealing the transmission errors. Dithering techniques have been used to obtain a binary watermark from the low resolution version of the image/video frame. Multiple copies of the dithered watermark are embedded in frequencies in a specific range to make it more robust to channel errors. It is shown experimentally that based on the frequency selection and scaling factor variation, high quality watermark can be extracted from a low quality lossy received image/video frame. Furthermore, the proposed technique is compared to its two-part variant where the low resolution version is encoded and transmitted as side information instead of embedding it. Simulation results show that the proposed concealment technique using data hiding outperforms existing approaches in improving the perceptual quality, especially in the case of higher loss probabilities.**

***Index Terms*— Image/video transmission, data hiding, digital half-toning, error concealment, wireless video.**

## I. Introduction

THE transmission of images and video over wired and/or wireless channels introduces multiple losses into the transmitted data that manifest themselves as various types of artifacts. These artifacts degrade the quality of the received image/video as they vary rapidly during the course of transmission based on the channel conditions. Therefore, there is a need for a good error concealment technique that can detect and correct (or conceal) these errors better and display a good quality image/video regardless of the channel conditions [1]. In the case of video transmission over wireless channels, adaptive error control that adapts to the approaches both at the transmitter and at the receiver has proven to be more effective [2].

Manuscript received August 21, 2003; revised August 24, 2004. This work was supported in part by a National Science Foundation grant No. CCR-0105404, in part by a University of California MICRO grant with matching supports from Philips Research Laboratories and Microsoft Corporation

Chowdary B. Adsumilli, Mylène C. Q. Farias, and Sanjit K. Mitra are with the Department of Electrical and Computer Engineering at University of California, Santa Barbara, USA. Email: (chowdary, mylene, mitra)@ece.ucsb.edu

Marco Carli is with the Department of Applied Electronics at University of Rome TRE, Rome, Italy. Email: carli@uniroma3.it

Error concealment methods use spatial and temporal information to recognize that an error has occurred. Once an error is detected, the received video stream is adjusted with an attempt to recover the original data. A number of error concealment techniques have been proposed in the literature that use either statistical methods to detect and correct errors (these are usually computationally intensive) or depend on certain critical information from the transmitter, like the re-synchronization markers, to detect these transmission errors.

In this paper, a new robust error concealment algorithm which uses watermarking is proposed. Watermarking is usually used to introduce some redundancy to the transmitted data with little increase in its bit rate during implementation [3].

The basic idea of the proposed algorithm is as follows. A frame of a video is wavelet transformed and the low-low approximation coefficients (usually second or third order) are then embedded in the frame itself during MPEG encoding. The embedded data can then be error protected unequally such that the mark signal embedded packets are given higher protection against channel errors. At the receiver, the mark is extracted from the decoded frame. The channel corrupted information of the frame is then reconstructed using the embedded mark signal. Specific areas lost through transmission are selected from the reconstructed mark signal and replaced in the original frame, thus enhancing its perceptual quality.[1] This contribution provides modified algorithms for implementation in video transmission and high detail color extensions in case of wired and wireless transmission scenarios. Also, a comparison of the proposed technique to its two-part variant (where the low resolution child image is encoded and transmitted as side information) is provided along with an extensive analysis of its performance.

The paper is organized as follows. Section II briefly points out the previous work in error concealment where similar/related approaches have been addressed. Section III describes the proposed error concealment/reconstruction algorithm using data hiding. Section IV gives a detailed explanation of the multiple scenarios under which the algorithm can be modified to make it more efficient. The simulations done using the proposed technique and the obtained results are presented in Section V. In Section VI, conclusions are drawn based on the results and suggestions for further work are discussed.

[1]Preliminary results of the performance of the algorithm's implementation for wired transmission of images have been partially reported [4].

## II. Previous Work

The use of data hiding as an error control tool was first introduced by Liu and Li [5]. They extracted the important information in an image, like the DC components of each $8 \times 8$ block, and embedded it into the host image. In the work that followed, Liu and Li's work formed the basis. Certain key features were extracted from the image and these features were encoded and data hidden in the original image either as a resilience tool or for concealment [6]-[8].

Watermarking of error correcting codes was introduced by Lee and Won [9]. Here, the parities generated by conventional error control codes were used for watermarking sequence. A region of interest (ROI) based coded bit stream embedding was employed by Wang and Ji, where the ROI DCT bit stream is embedded into the region-of-background wavelet coefficients [10]. This technique gives better results when perception based encoding is employed.

The concept was extended to video coding by Bartolini *et al.* [11]. However, they used data hiding as a tool to increase the syntax-based error detection rate in H.263 but not for recovering or correcting lost data. Munadi *et al.* extended the concept of key feature extraction and embedding to inter-frame coding [12]. In their scheme, the most important feature is embedded into the prediction error of the current frame. However, the effects on motion vectors and the loss of motion compensated errors were not addressed. Yilmaz and Alatan proposed embedding a combination of edge oriented information, block bit-length, and parity bits in intra-frames [13]. They use a minimally robust technique of even-odd signalling of DCT coefficients for embedding.

The problems with existing techniques are: (1) Only one or a few selected set of key features are used for embedding. These features may not necessarily follow the loss characteristics of the channel employed. (2) They use transform domain to encode the data that needs to be embedded, often DCT. However, if losses occur on the DC coefficient or a set of first few AC coefficients, the loss to the extracted reference would be significant and therefore may lead to reduction in concealment performance. (3) Almost all the techniques use fragile or semi-fragile data embedding schemes which are more susceptible to attacks. Our proposed technique avoids two of the three problems by embedding half-toned version of the whole reference image (instead of encoding its transform coefficients). This way, loss or errors in the data will have smaller and local effects on the reconstructed video. A possible solution to the third problem will be addressed in a future work.

A set of concealment techniques that do not use data hiding while giving similar high levels of performance have been proposed in the literature. Block based deterministic interpolation models are used for reconstruction of missing blocks in either spatial domain [14]-[16] (simplified edge extraction imposition for obtaining the directional interpolation was considered in [14] and [15] while projection onto convex sets was considered in [16]) or spectral domain [17] (where lost DCT coefficients are estimated based on spatial smoothing constraints). Li and Orchard provided a good review of these techniques and proposed a set of block-based sequential recovery techniques [18]. These work well in simplified loss scenarios where successfully received data is assumed to be reconstructed loss free. This is often not the case. A comparison of these techniques with the proposed technique is provided in Section V.

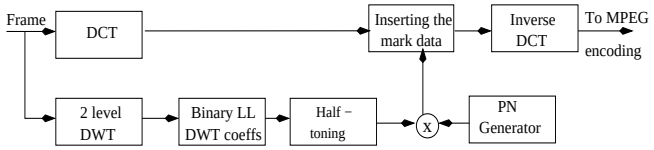



Fig. 1. Block diagram of the embedding algorithm.

## III. Error Concealment using Data Hiding

The proposed scheme can be divided into an embedding part and a retrieval part. It should be noted that the proposed technique does not overload the communication channel by requiring feedback or any retransmission of damaged blocks.

### A. The Embedding Part

The data hiding technique used here is a modified version of the Cox's watermarking algorithm [19]. Due to the limited embedding capacity of the algorithm, it is practically not feasible to embed the whole frame into itself [20]. In this work, therefore, the discrete wavelet transform (DWT) and dithering techniques have been used to reduce the amount of data to be embedded such that the algorithm embeds maximum information while still catering to the feasibility issues. Wavelets have several properties that make them good candidates for this application. Some of the important ones relevant to this algorithm are: (1) The approximation coefficients provide a good low-resolution estimate of the image, while minimizing the aliasing artifacts resulting from the reduction in resolution, and (2) The wavelet coefficients are localized, such that a corruption of a coefficient through channel errors has only local effect on the image. Dithering techniques make it possible to generate binary images which look very similar to gray level images. The technique employed here is Floyd-Steinberg error diffusion dithering algorithm [21], [22]. The DWT approximation coefficients are half-toned before being embedded.

The block diagram of the embedding algorithm is shown in Fig. 1. The operation of the embedding part can be described as follows. The 2-D DWT of the frame is first computed. A second level DWT is performed again on the approximation coefficients to obtain an image that is $\frac{1}{16}$-th the size of the original frame. A half-toned image, the marker, is then generated from the reduced size image. One marker is used for each frame. After the marker is generated, each pixel of the marker is repeated $4$ times in a $2 \times 2$ matrix format. This repetition operation allows the decoder to recover the marker from the data in a more robust fashion.

Mathematically, the reduced size image generated for the $i$-th frame, $\mathbf{f}_i$, can be represented as $\mathbf{m}_i$. Here, $\mathbf{f}_i$ is of size $m \times n$ and $\mathbf{m}_i$ is of size $\frac{m}{4} \times \frac{n}{4}$. The half-toning operation

is performed on $\mathbf{m}_i$ using a Floyd-Steinberg diffusion kernel $\mathbf{D}_{FS}$ given by

$$\mathbf{D}_{FS} = \frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 0 & P & 7 \\ 3 & 5 & 1 \end{bmatrix}, \quad (1)$$

where $P$ is the current pixel position. $\mathbf{D}_{FS}$ is typically applied on each $3 \times 3$ block of the reduced size image. The resulting marker is denoted as $\mathbf{w}_i$. Each pixel of $\mathbf{w}_i$ is then repeated in a $2 \times 2$ matrix format to form $\breve{\mathbf{w}}_i$. Note that $\breve{\mathbf{w}}_i$ is of size $\frac{m}{2} \times \frac{n}{2}$.

A zero mean, unit variance pseudo-noise image is then randomly generated with a Gaussian distribution and a known seed. A unique pseudo-noise image, $\mathbf{p}_i$, of size $\frac{m}{2} \times \frac{n}{2}$ is generated for each frame of the video. For a generic $i$-th frame $\mathbf{f}_i$, of a video sequence, the final watermark $\tilde{\mathbf{w}}_i$ is obtained by multiplying $\breve{\mathbf{w}}_i$ with the pseudo-noise image, $\mathbf{p}_i$:

$$\tilde{\mathbf{w}}_i = \breve{\mathbf{w}}_i . * \mathbf{p}_i \quad (2)$$

where $.*$ represents element-by-element multiplication. Note that $\tilde{\mathbf{w}}_i \in \{0, 1\}$.

The computed DCT coefficients of the luminance channel of the frame $\mathbf{f}_i$ are denoted as $\mathbf{F}_i$. The watermark, $\tilde{\mathbf{w}}_i$ is then scaled by a factor $\alpha$, and added to a set of coefficients in $\mathbf{F}_i$ starting at the initial frequencies of $(\Delta_1, \Delta_2)$. The resulting image $\mathbf{Y}_i$ is given by

$$Y_i(k + \Delta_1, l + \Delta_2) = F_i(k + \Delta_1, l + \Delta_2) + \alpha \cdot \tilde{w}_i(k, l) \quad (3)$$

where $k$ and $l$ correspond to the pixel location in the spatial domain and the coefficient location in the DCT domain. Here, $Y_i(\cdot, \cdot)$, $F_i(\cdot, \cdot)$, and $\tilde{w}_i(\cdot, \cdot)$ represent the individual component values of matrices $\mathbf{Y}_i$, $\mathbf{F}_i$, and $\tilde{\mathbf{w}}_i$, respectively. Note that $\Delta_1 \in [0, \frac{m}{2}]$ and $\Delta_2 \in [0, \frac{n}{2}]$. $\mathbf{Y}_i$ is then inverse transformed, encoded and transmitted.

In the proposed method, the final watermark is added only to the mid-frequency DCT coefficients. The range of frequencies where the watermark is inserted is strongly dependent on the application. For the purpose of delivering a high quality video through a channel, the mid-frequencies are a good choice. Inserting the watermark in the low-frequencies would cause visible artifacts in the image, while inserting it in the high frequencies would make it more prone to channel induced defects. Also, multiple copies of the marker can be inserted sequentially with various initial frequencies to make the watermark more robust to channel errors. In this case, the multiple copies are generated using independent randomly generated pseudo-noise matrices.

### *B. The Retrieval Part*

The block diagram of the retrieval technique is shown in Fig. 2. The DCT coefficients of the luminance channel of the received frame $\mathbf{y}_{ri}$, denoted by $\mathbf{Y}_{ri}$, are computed as

$$\mathbf{Y}_{ri} = DCT_2(\mathbf{y}_{ri}) \quad (4)$$

where $DCT_2$ represents the 2-D DCT operation.

These coefficients are then multiplied by the corresponding pseudo-noise image $\mathbf{p}_i$. The pseudo-noise image generated is

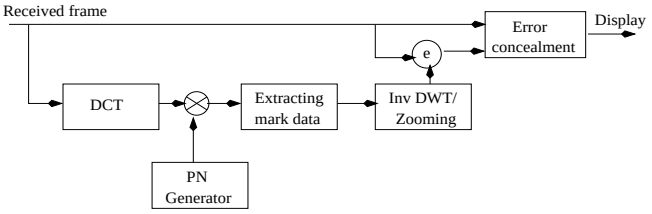



Fig. 2. Block diagram of the retrieval algorithm.

the same as that at the transmitter side. It is tacitly assumed that the receiver knows the seed for generating the pseudo-noise image and the initial frequencies, $(\Delta_1, \Delta_2)$, where the mark was inserted. An issue of concern with this assumption is that it might lead to possible synchronization problems when severe channel errors cause loss of frames. This can in turn be handled by embedding the frame order number $i$ (similar to the sequence number in packet transmission) into the frame itself. The receiver side pseudo noise generator algorithm can be driven by the recovered value while the missing frames can be detected using the missing frame order numbers.

The result of the multiplication, denoted as $\breve{\mathbf{w}}_{ri}$, is averaged over the 4 pixels ($2 \times 2$ matrix form) and the binary marker is extracted by taking the sign of this average:

$$w_{ri}(k, l) = sgn\left\{\frac{1}{4}\left(\lambda_i(k, l)\right)\right\}. \quad (5)$$

where

$$\lambda_i(k, l) = \sum_{k'=2k-1}^{2k} \sum_{l'=2l-1}^{2l} Y_{ri}(k' + \Delta_1, l' + \Delta_2) \cdot p_i(k', l') \quad (6)$$

and $Y_{ri}(\cdot, \cdot)$, $p_i(\cdot, \cdot)$, and $w_{ri}(\cdot, \cdot)$ are the individual component values of matrices $\mathbf{Y}_{ri}$, $\mathbf{p}_i$, and the extracted marker $\mathbf{w}_{ri}$, respectively. Note here that the values of $\mathbf{w}_{ri}$ greater than 0 are assigned a value of 1 and those that are equal to or less than 0 are assigned a value 0 to make the resulting image binary. Also note that while the size of $\breve{\mathbf{w}}_{ri}$ is $\frac{m}{2} \times \frac{n}{2}$, the size of $\mathbf{w}_{ri}$ is $\frac{m}{4} \times \frac{n}{4}$.

It has been shown that this approach enables a fairly large amount of hidden data to be embedded without significantly affecting the perceptual quality of the encoded image [4]. Once the binary marker is extracted, the reduced size image is obtained by inverse half-toning the watermark.

Although a number of algorithms have been proposed for inverse half-toning [23]-[25], the inverse half-toning algorithm using wavelets proposed by Xiong *et al.* [23] is employed here because of its performance and ease of operation. This process primarily involves edge extraction from the high frequency components and edge preserving noise removal of the low frequency components of the wavelet coefficients. A discrete dyadic wavelet transform using 'Haar' wavelet (as a perfect reconstruction filter) without sampling rate conversion is employed to obtain back the processed smooth marker. The whole operation can be represented as

$$\hat{\mathbf{m}}_i = HT^{-1}(\mathbf{w}_{ri}) \quad (7)$$

where $HT^{-1}$ denotes the inverse half-toning operation and $\hat{\mathbf{m}}_i$ is the extracted approximation of $\mathbf{m}_i$.

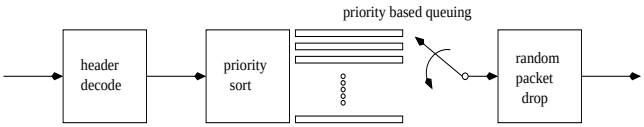



Fig. 3. Block diagram of the lossy wired transmission model.

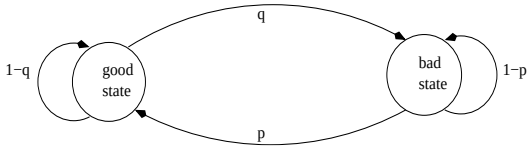



Fig. 4. States of the adopted wireless simulation model.

A 2-D inverse DWT is performed on this smooth marker to obtain an intermediate resolution image $\hat{\mathbf{g}}_i$. The values of $\hat{\mathbf{m}}_i$ form the approximation coefficients. Other high frequency coefficients are assumed to be $0$ while computing the inverse DWT.

$$\hat{\mathbf{g}}_i = IDWT_2(\hat{\mathbf{m}}_i). \tag{8}$$

Note that $\hat{\mathbf{g}}_i$ is of size $\frac{m}{2} \times \frac{n}{2}$. It is then up-sampled by a factor of $2$ and passed through a low pass interpolation filter to obtain an $m \times n$ image. The resulting image $\hat{\mathbf{f}}_i$, is compared with the current received frame $\mathbf{y}_{ri}$ to detect and conceal the corrupted blocks by substituting the appropriate data.

The criterion for substituting the loss areas is different for images and video. The substituted areas in images are identified by packet-size blocks of lost data while in case of video, these are located using motion vectors and MCER values. An implementation issue here is that the marker needs to be scaled before the appropriate areas are substituted. The scaling can be either done throughout the image or only in the localized areas where the frame experienced packet losses. A global scaling constant is used when the image is globally scaled. In case of local scaling, the different local scaling factors are used based on the intensities of the surrounding areas. Both approaches are explored and the results presented in Section V.

## IV. Case Studies

In this section, three different extensions where the algorithm can be extended with minor modifications are presented and discussed. The extensions to the regular implementation are: (1) Extension from wired to wireless transmission, (2) Extension from gray scale to color images, and (3) Extension from images to video. The various models that are adopted to test the feasibility and effectiveness of the proposed algorithm in these three scenarios are also discussed.

### A. Wired vs. Wireless transmission

The lossy wired model that has been adopted for implementation of the proposed algorithm is shown in Fig. 3. The testing of the algorithm can be easily performed at the IP level. The embedded image is packetized with appropriate protection and header information and transmitted.

At the router level, the following algorithm is implemented. The channel (layered) header is decoded to check for a match in the source, next hop, and destination IP addresses. Then the packets are sorted into one of the multiple priority queues based on the value of *priority_byte* in the header. The transmission from this point forward is based on the priority of the packet and the current channel conditions. Only when queue $1$ is empty, packets from queue $2$ are transmitted and so on. According to this model, packet loss is introduced when the queue buffer is full for each of the queues except for queue $1$.

Once the packets are reordered based on their priorities, a delay is created in the transmission according to predefined latency values. The packets are then randomly dropped in accordance with a known probabilistic distribution (Gaussian in this case) which has a preset (controllable) mean and variance. The remainder of the packets are forwarded to the destination of a point-to-point network.

However, for wireless cases, the model in Fig. 3 does not work due to the following reasons: (1) Wireless channel has unpredictable variation with time, (2) Co- and cross-channel interferences are not accounted for, and (3) Fading and power losses are not considered in the current model. Apart from these, the model used in case of wired transmission is adapted for network traffic characteristics of Internet, typically like network congestion, which is quite unlike the case in wireless transmission.

A link layer modelling instead of network layer is adopted for wireless channel transmission scenarios. A simple point-to-point, two-state Markov chain, i.e., the Gilbert-Elliot model as shown in Fig. 4, has been adopted for wireless transmission scenarios [26]. The two states in this model can be considered as the *good* state and *bad* state or the receive state and the loss state, respectively, with predefined probabilities $p$ and $q$. This means that the Markov chain is in the *good* state if the packet is received in time without any errors and is in the *bad* state if the packet is lost during transmission due to latency or bandwidth limitations of the channel.

The parameters $p$ and $q$ are called the transition probabilities of the Markov chain between the *good* and the *bad* states. The transition matrix of this two-state Markov chain can be represented as

$$\mathbf{M}_2 = \begin{bmatrix} 1-q & q \\ p & 1-p \end{bmatrix} \tag{9}$$

The values of $p$ and $q$ are quite apart with more emphasis on the *good* state. For a typical wireless channel, the values of $p$ and $q$ would be around $0.999$ and $0.001$. Note here that $p + q = 1$. Such a model is followed here for the wireless channel simulations and the results are presented in Section V.

### B. Gray scale vs. Color

The algorithm can be extended to work in the case of color image/video transmission. In this case, differences can be seen not only in the algorithm implementation, but in the results too. The variation in results and their analysis is presented

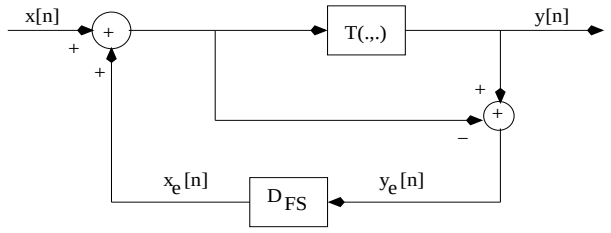



Fig. 5. Block diagram of the Error Diffusion algorithm. $D_{FS}$ is the Floyd-Steinberg kernel and $T(\cdot,\cdot)$ is the threshold operator.

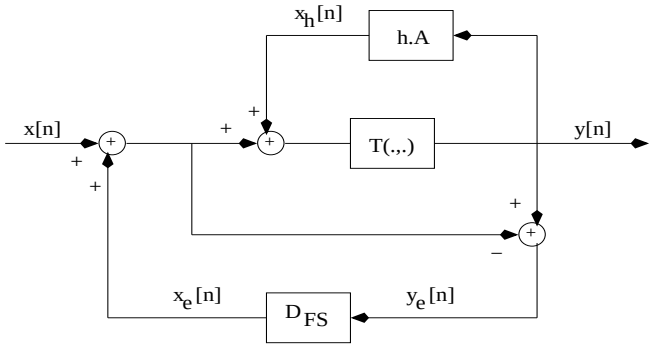



Fig. 6. Block diagram of the Generalized Error Diffusion algorithm. $h$ is the hysteresis value and $\mathbf{A}$ is a low pass filter.

in Section V while the implementation changes are discussed here.

The half-toning technique used for gray scale image processing is a simple feedback based loop as shown in Fig. 5. Here, $\mathbf{D}_{FS}$ is the Floyd-Steinberg kernel given in (1) and $T(\cdot,\cdot)$ is a threshold operation. $x_e[n]$ and $y_e[n]$, the errors in the $n$-th sample of $x[n]$ and $y[n]$ respectively, are "diffused" back into $x[n+1]$ to $x[n+4]$ samples using a feedback loop. This implementation gives a pretty good estimate of the original gray scale image as a binary image.

For the case of color images/videos, we use color dithering [27]. Here, the color space is divided into $4$ subspaces Cyan (C), Magenta (M), Yellow (Y), and black (K). The original RGB color image $\mathbf{S}_1$ is converted into an CMYK image $\mathbf{S}_{2new}$ using a transformation given by

$$\begin{aligned} \mathbf{S}_{2old} &= 1 - \mathbf{S}_1 \\ K &= min\left\{[\mathbf{S}_{2old}]^T\right\} \\ \mathbf{S}_{2new} &= \mathbf{S}_{2old} - K \end{aligned} \tag{10}$$

where $\mathbf{S}_1 = [R\ G\ B]^T$ and $\mathbf{S}_2 = [C\ M\ Y]^T$. The black values $K$ are initially calculated using *old* $C$,$M$,$Y$ values and then, the *new* $C$,$M$,$Y$ values are obtained by modifying the *old* ones using these $K$ values. The error diffusion shown in Fig. 6 is then applied to the CMYK images individually to obtain a color half-toned image.

The block diagram of the generalized error diffusion algorithm used for color image half-toning, shown in Fig. 6, is quite similar to Floyd-Steinberg dithering technique in Fig. 5 except that it has another positive feedback loop [28]. The sample $y[n]$, created from threshold operation on $x[n]$, goes through a scaled lowpass filter $\mathbf{A}$ with a predefined *hysteresis* value $h$. Since the sample values after threshold process are sharper in the CMYK space, this lowpass filtering operation is required. $\mathbf{A}$ is chosen to be a $3 \times 3$ lowpass filter to simplify implementation with respect to the $3 \times 3$ Floyd-Steinberg kernel $\mathbf{D}_{FS}$. $h$ is chosen to be $0.5$ in our implementation.

After the error-diffusion process, the image is converted back to RGB color space by an approximate inverse of the transformation defined in (10). The conversion can be represented as:

$$\begin{aligned} \mathbf{S}_{2new} &= \mathbf{S}_{2old} + K \\ \mathbf{S}_1 &= 1 - \mathbf{S}_{2new}. \end{aligned} \tag{11}$$

Either the R, G, and B channels or the Y, Cb, and Cr channels of the half-toned color marker signal can now be embedded into the luminance component of the original image/video frame. Each of these cases with individual and dependent variations in $\alpha$ are considered and the results are presented and analyzed in Section V.

### C. Image vs. Video

There are several methods in which the previous algorithm can be extended to video. Due to the facts that the algorithm can be implemented on a frame-by-frame basis and the effects of watermarking on motion vectors is minimal (shown later), this algorithm can be employed in almost all of the currently existing video communication codecs like MPEG or H.263.[2] However, implementing the algorithm on a frame-by-frame basis is practically not viable (even though it gives very high PSNR values after reconstruction) because it increases the pre- and post-processing complexity of the video transceivers.

A trade-off issue that needs to be considered is a reduction in the pre- and post-processing complexity against the loss of some efficiency of the error concealment that the algorithm provides. We can achieve this trade-off in two ways. In the first method, we exploit the motion vector information of the video [29]. It can be seen from Fig. 7 that the effect of embedding the watermark in consecutive frames does not cause noticeable differences to the motion vectors. Previous work also suggests that any small effects that this process creates can be used to enhance the error concealment at the receiver by coding and transmitting these differences [30]. Hence, we can embed the low resolution versions of the I-frames inside themselves with little change in the motion vector information of the P- and the B-frames provided the P-frames are embedded in themselves.

The effects of the algorithm on motion vectors can be seen in Fig. 7. The original $k$-th and $(k+1)$-th frames of the *Table tennis* video are shown in Fig. 7(a) and (b), respectively. Fig. 7(c) shows the original motion vectors between these two frames, and Fig. 7(d) shows the effect of watermarking on the motion vectors when the watermark is embedded only in the $k$-th frame. Note that these effects are seen in the areas (mid-frequency range of the frame) where the watermark is embedded, specifically in the area of the table in this case. Fig. 7(e) shows the effects of watermarking on motion vectors

[2]Implementation in H.263 is much more straight forward based on the fact that the algorithm can be extended to work for YUV (4:2:2) and (4:2:0) color video frames.

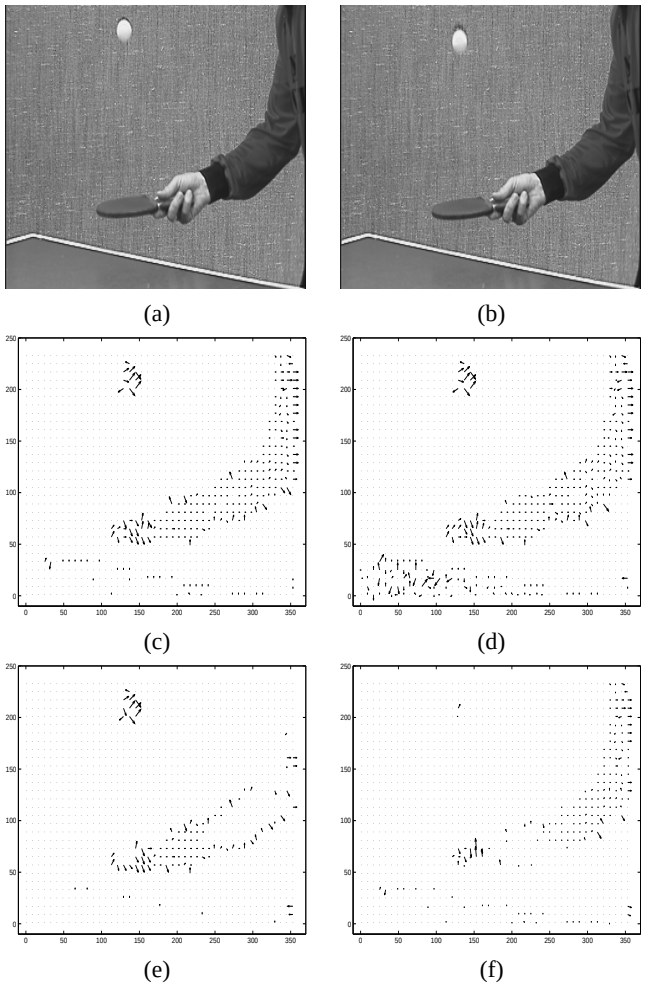


Fig. 7. (a) Original $k$-th frame of table tennis sequence, (b) original $(k+1)$-th frame, (c) motion vectors between the above frames, (d) motion vectors when the watermark is embedded in only the $k$-th frame, (e) motion vectors when the watermark is embedded in both the frames, and (f) difference in motion vectors for (c) and (e).

when the mark is embedded in both the frames. This figure shows that embedding in both frames removes the effects of data hiding on motion vectors. Fig. 7(f) shows the difference between Figs. 7(c) and (e).

However, this algorithm can only conceal the spatial errors introduced in the transmission while the temporal errors continue to be present. These latter errors in the video can be easily reduced by a simple modification of the proposed technique. Instead of embedding the I-frames in themselves, we embed the low resolution version of the subsequent P-frame in the current I-frame. This not only enhances the error concealment at the receiver for the current and subsequent P- and B-frames, but also preserves the motion information by giving a reference for the prediction. For enhanced error concealment in P- and B-frames, their motion vectors can also be embedded along with the low resolution versions of the frames. A disadvantage of this modification would be that it does not take care of the loss of information in the I-frame itself.

In the second method to achieve the processing-efficiency trade-off, we can embed the non I-frames in the motion vectors itself. Previous work suggests that an arbitrary one-dimensional signal can be embedded into the motion vectors of a video and recovered at the receiver without much loss [31]. This can be adapted to the proposed algorithm such that the watermark can now be embedded into the motion vectors and can be recovered without much loss. To reduce the coding overhead in this case, we can now embed every alternate P-frame (instead of every P-frame) in the motion vectors that correspond to the frame. These techniques have not yet been explored and are under investigation.

## V. Experimental Results and Analysis

The algorithm proposed in Section III with extensions proposed in Section IV is implemented. For wired channel transmission, conventional UDP protocol with packet loss is considered, whereas a simulated point-to-point lossy link layer model is used for wireless cases. The following assumptions are made for simplicity with regard to the implementation of the algorithm: (1) The binary loss probability of the channel is assumed to be constant for a given network bandwidth, (2) The source transmission rate is assumed to be less than the maximum channel bandwidth, (3) No re-transmissions occur, and (4) Bit errors over successfully received packets are negligible.

'Haar' wavelet is used for calculating the two-level approximate wavelet coefficients. For wired scenarios, the compressed output stream is vectorized and multiplied with a vector that is randomly generated using a Gaussian distribution. About $1000$ varying packet loss simulations have been generated independently for each transmission and their statistical average is taken to obtain the probability loss vector.

In the case of wireless transmission, the channel and its losses are simulated using the *ns-2* simulator [32]. The power levels for the wireless transmission are kept almost constant. A constant fading is assumed for each of the packet transmissions in the individual channels of a multichannel scenario. It should be noted here that a standard CDMA$2000$ system [33] is followed due to its efficiency of operation in the link layer when compared to other wireless systems.

Table I summarizes the results of the experiment for wired and wireless transmissions. In case of color images, the Composite PSNR (CPSNR) is calculated at the receiver. The symbol * indicates the images/video frames that are implemented using the wired transmission simulations. Note that the percentage improvement due to localized error concealment in the case of wireless transmission is much more pronounced than in the case of wired transmission. For each value of image/video frame, the PSNR of the received image (PSNR$_{rec}$), the PSNR of the error concealed image (PSNR$_{erc}$), and the PSNR of the local-scaled error concealed image (PSNR$_{loc}$) are noted.

When the received frame is error concealed using the proposed algorithm, the scaling operation performed on the concealed data blocks is fixed. Instead, if localized interpolation is performed on the reconstructed blocks by considering the neighboring pixels that were correctly received, an

TABLE I

PERFORMANCE OF THE PROPOSED ALGORITHM. PSNR (IN DB) FOR A FIXED MEAN LOSS 15% AND VARIANCE 2.5%

| *Frame/Image* | PSNR$_{rec}$ | PSNR$_{erc}$ | PSNR$_{loc}$ | % inc |
|---|---|---|---|---|
| Sail | 19.4923 | 26.5968 | 29.6880 | 11.6 |
| Cameraman* | 19.8271 | 27.6852 | 28.9047 | 4.4 |
| News | 21.9273 | 27.2865 | 30.5244 | 11.9 |
| Psycho | 15.6776 | 25.4005 | 29.4129 | 15.8 |
| Table tennis | 16.8707 | 24.3020 | 27.6268 | 13.7 |
| Flower | 17.8046 | 21.6195 | 24.6611 | 14.1 |
| Football | 19.1367 | 27.7806 | 30.9214 | 11.3 |
| Stefan | 20.1159 | 26.5788 | 29.4279 | 10.7 |
| Coastguard* | 20.0866 | 25.6650 | 27.0271 | 5.3 |
| Hockey | 19.3017 | 24.3303 | 33.2081 | 36.5 |
| Girl | 18.1319 | 25.4701 | 31.7092 | 24.5 |
| Frank | 20.0172 | 23.0964 | 29.0501 | 25.8 |
| Surf | 18.1977 | 23.5479 | 30.4909 | 29.5 |
| Gold hill* | 18.0111 | 27.1874 | 28.4486 | 4.6 |

*wired cases

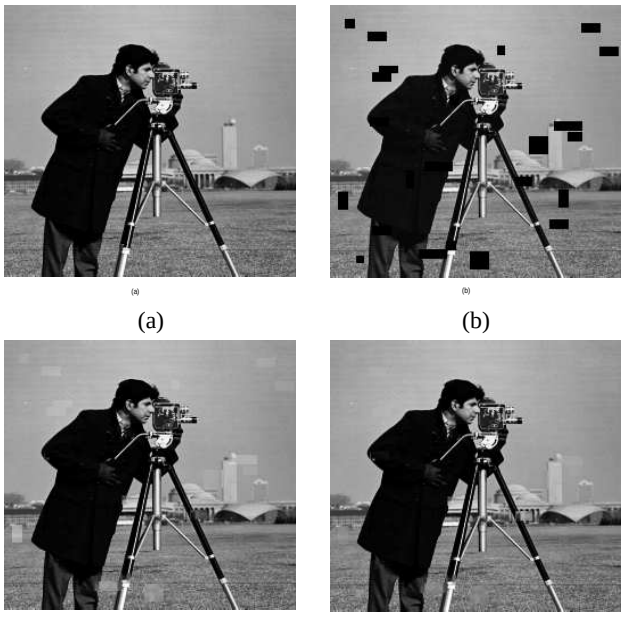
improvement in the visual quality of the restored frame is observed (as can be seen in Table I).

A sample result for the wired transmission case is shown in Fig. 8 for the *Cameraman* image with the parameter values: $\alpha$ = 3.6, mean loss probability = 0.15, loss variance = 2.5%. The received frame had a PSNR$_{rec}$ = 19.83 and the error concealed image had a PSNR$_{erc}$ = 27.69. These frames with the original frame are shown in Figs. 8(a), (b) and (c). Fig. 8(d) has been obtained by localized scaling error concealment. Here, the error-concealed image is locally scaled by using a localization kernel of size $8 \times 8$ or $16 \times 16$ (based on the size of the lost data area). The PSNR obtained by performing this localized scaling operation on the error-concealed image is PSNR$_{loc}$ = 28.90. However, better results in terms of PSNR are expected if the kernel size is varied.

In Fig. 9, a sample result for the wireless transmission simulation are presented for the *Sail* image with the parameter values: $\alpha$ = 5, average loss probability = 0.12, and loss variance = 5%. The received image had a PSNR$_{rec}$ = 19.30 (the value of 19.4923 that is listed in the table is CPSNR, i.e., it is obtained for the color image). The original, received and error-concealed images (PSNR$_{erc}$ = 28.17) are shown in Figs. 9(a), (b) and (c) respectively and the localized error concealed image with $8 \times 8$ kernel (PSNR$_{loc}$ = 29.91) is represented in Fig. 9(d).

A sample set of 25 iterations for error concealment and localized concealment algorithms on *Sail* image (due to randomness in loss for wireless scenarios) is represented in Fig. 10 with variations in packet loss probability. As seen from these simulations, the quality of the error-concealed image is much better (approximately $6-8$ dB improvement) for all cases. The localized error concealment algorithm achieved higher quality (approximately $2-3$ dB over the error concealment algorithm). Note that the improvements in the PSNR values increased with increasing loss probabilities. The mean and standard deviations of these curves are shown in Fig. 11. Note that the variation of standard deviations was higher at lower packet losses and lower at higher packet losses.

Since 4 copies of the marker are embedded in the image/video frame, it can be extracted with high quality even



(a) (b) (c) (d)

Fig. 8. (a) Original frame, (b) Received frame with mean loss probability = 0.15, variance = 2.5%; PSNR = 19.8271, (c) Error concealed frame; $\alpha$ = 3.6; PSNR = 27.6852, and (d) Localized scaling error concealed frame; PSNR = 28.9047.

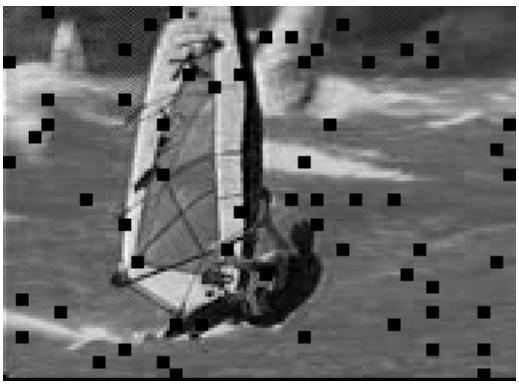
at higher loss probabilities of $0.4 - 0.6$. The variation of the watermark quality with the loss probability for the same sample set of 25 simulations is plotted in Fig. 12. The mean and the standard deviations of the extracted watermark quality are plotted in Fig. 13. Note that while the mean decreased, the standard deviations increased non-linearly with increasing packet loss probabilities. This suggests that the variation in the decrease in extracted watermark quality is non-linear, i.e., the decrease in quality varies less at lower loss probabilities and changes more at higher loss probabilities even with high robustness. It can be shown that finding an optimum value of the scaling parameter, $\alpha$, might compensate for this non-linear variation.

### A. Scaling parameter variation

The value of the scaling parameter $\alpha$ is changed to vary the strength of the embedded watermark. The parameter $\alpha$ in this case defines the robustness of the watermark with respect to channel errors. As $\alpha$ increases, it not only enhances the quality of the received marker, but also makes the embedded watermark more visible thus reducing the quality of the received image/frame. Hence, it is important to choose an optimum value of $\alpha$ such that there is a balance between the quality of the received image/video frame and the amount of the information embedded for the quality of the received watermark to be high.

The results of the experiment with $\alpha$ variation can be seen in Fig. 14. The *Sail* image in Fig. 14(a) is embedded with $\alpha$ value of 1 while that in Fig. 14(b) is embedded with $\alpha$ value of 10 at the extreme case. The embedded signal content

(a)



(b)

(c)

(d)

Fig. 9. (a)The original image, (b)The received image with mean loss probability of $0.12$ and variance $5\%$ (PSNR=19.3003), (c) The error corrected image using the data hiding algorithm with $\alpha = 5$; (PSNR=28.1702), and (d) The localized error corrected image with kernel size $8 \times 8$; (PSNR=29.9137).

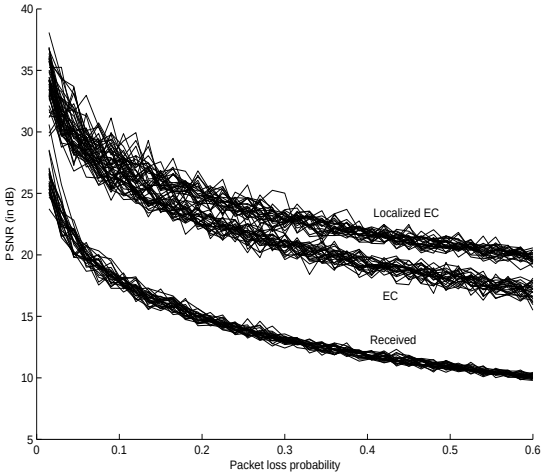



Fig. 10. Comparison of the quality of the received frame vs. the error concealed (EC) and the localized error concealed frames with variation in loss probability for a sample set of $25$ simulations.

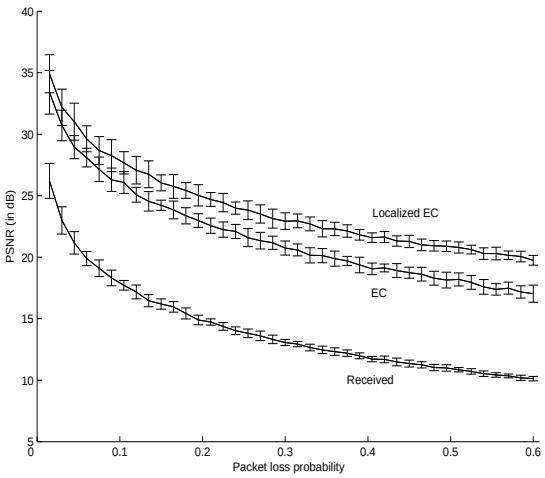



Fig. 11. The mean and standard deviation of the curves in Fig. 10.

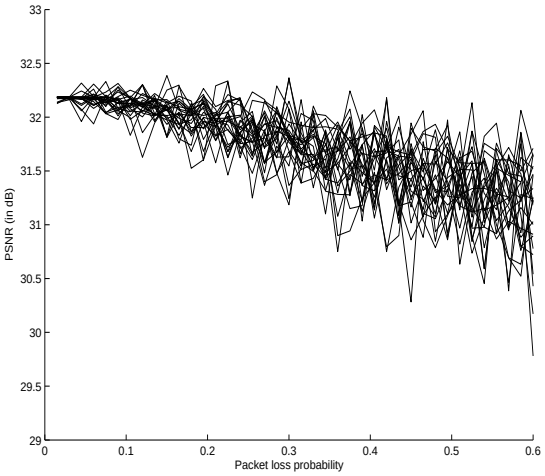



Fig. 12. The variation of the quality of the extracted watermark with increasing packet loss probabilities for the same sample set as in Fig. 10.

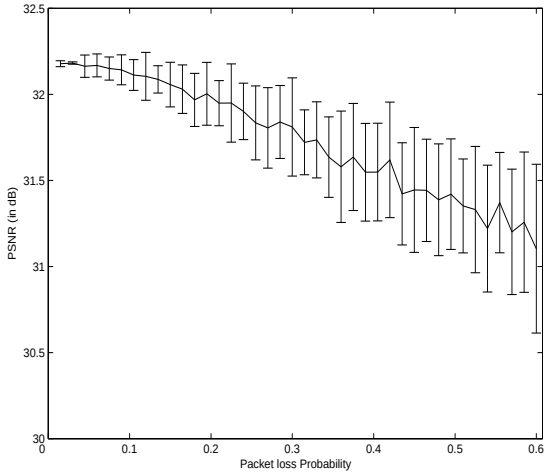
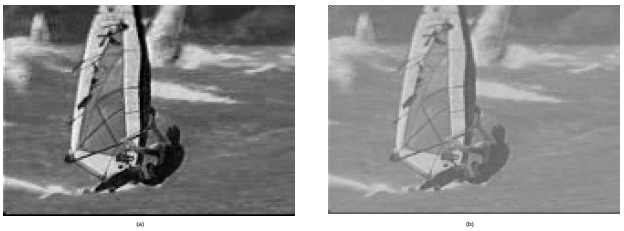



Fig. 13. The mean and standard deviation of the extracted watermark quality in Fig. 12.



(a) (b)

Fig. 14. Results with different values of $\alpha$ (a) $\alpha$ = 1, and (b) $\alpha$ = 10 (extreme cases).

in Fig. 14(b) increased multi fold thus highly decreasing the frame quality.

The extracted watermark from the received image, however, has quality variations that follow a different trend. Fig. 15 shows the variation of the quality of the extracted watermark for packet loss probability = $0.09$. Note that the quality converges to a maximum value ($32.11$ dB in the case of *Sail* image). The value of $\alpha$ at which this maximum is achieved, $\alpha_{thresh}$, can be defined as the smallest $\alpha$ after which there is no substantial increase in the quality for increasing $\alpha$. The value of $\alpha_{thresh}$ for the *Sail* image is $4.74$.

Since increasing $\alpha$ would decrease the quality of the received image/video frame, choosing $\alpha$ = $\alpha_{thresh}$ would be a start in finding the optimum $\alpha$. However, even at $\alpha_{thresh}$, we are not guaranteed a high quality of the received image. Here, the optimum is found graphically by plotting the quality variations of the received image and the extracted watermark against $\alpha$ variation and finding the point of intersection of these two curves based on the assumption that the transmitter has a good estimate of the channel losses (this can be achieved by either using the feedback from the receiver or expecting the channel to follow a particular probabilistic distribution that closely approximates the real time channel losses). Fig. 16 shows this plot for the *Sail* image for a mean packet loss probability of $0.09$. As expected, the optimum value of $\alpha$ (the point of intersection, $4.19$) lies below the value of

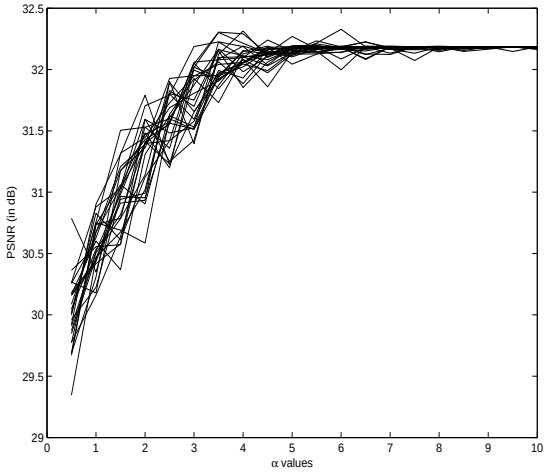



Fig. 15. Result of watermark quality variation with $\alpha$ for a sample packet loss probability of $0.09$. The plot also shows the convergence of watermark quality for higher values of $\alpha$.

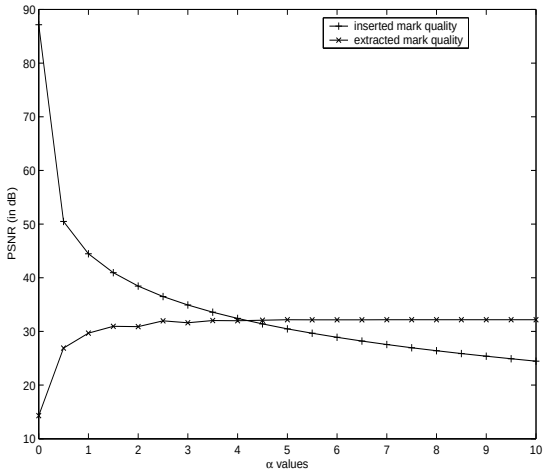


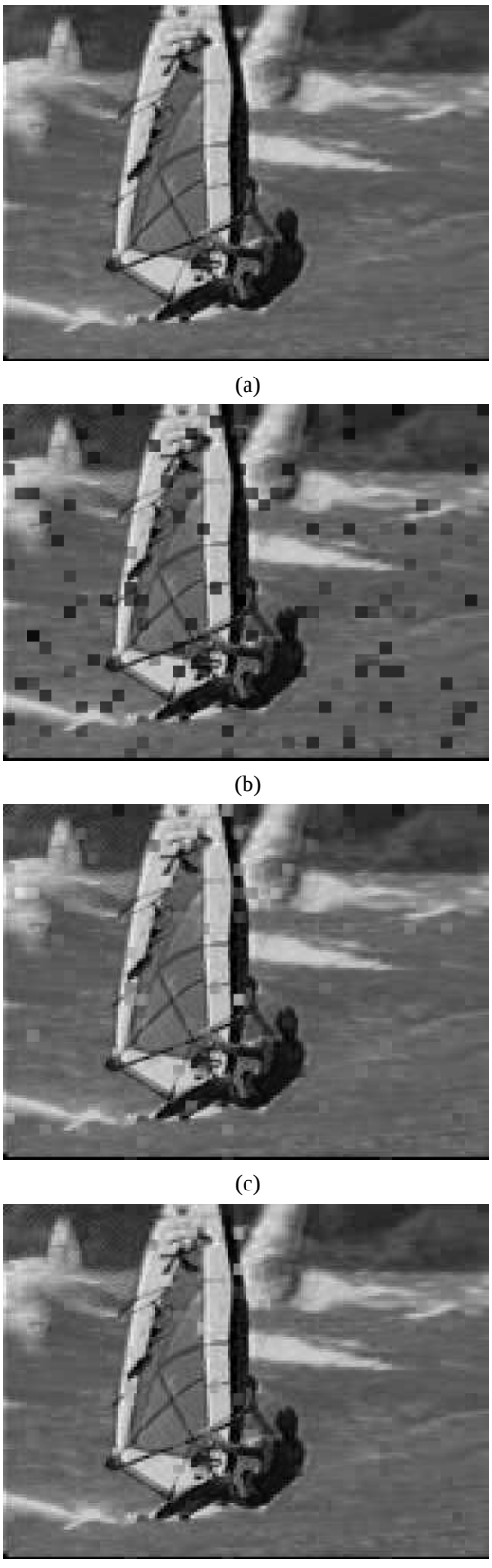

Fig. 16. Comparison of extracted watermark quality with degrading frame quality after watermark insertion.

$\alpha_{thresh}$. Also, since this value is close to $\alpha_{thresh}$, the non-linear marker quality fluctuations seen in Fig. 13 are reduced if not eliminated.

### *B. Effects on chrominance components*

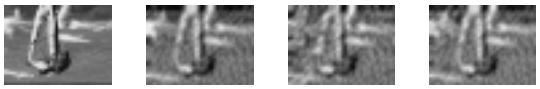
For the case of color images/video, losses are simulated both in the RGB color space and the YCbCr color space. In the latter case, even though the losses that affect all three channels are considered, those that effect only the luminance channel are given higher priority because most of the embedded information in the YCbCr (YUV in case of H.263) color space is mainly concentrated in the luminance channel.

The result of the algorithm implementation on the *Sail* color image are shown in Fig. 17. The original color image is shown in Fig. 17(a), while Fig. 17(b) represents the received image with errors due to wireless channel occurring in all three channels of YCbCr color space (PSNR$_{rec}$ = $19.49$). The error concealed image with PSNR$_{erc}$ = $26.60$ and the localized



(a) (b) (c) (d)

Fig. 17. (a)The original color image, (b)The received image with mean loss probability of 0.15 and variance 3% with loss occurring in all 3 channels (PSNR=19.4923), (c) The error concealed image using the data hiding algorithm (PSNR=26.5968), and (d) The localized error corrected image (PSNR=29.6880).



(a) (b) (c) (d)

Fig. 18. (a)The original watermark color image, (b)The received watermark for a mean loss probability of 0.15 and variance 3% when RGB channels are embedded with $\alpha$=5, (c) The extracted watermark for the same loss when YCrCb channels (4:2:0) are embedded with $\alpha$=5 , and (d) The extracted watermark when YCrCb (4:2:0) channels are embedded with $\alpha_Y$=3.75, $\alpha_{Cb}$=4.5, and $\alpha_{Cr}$=5. Note that (b), (c), and (d) are inverse half-toned images.

scaled error concealed image with PSNR$_{loc}$ = 29.69 are shown in Figs. 17(c) and (d) respectively.

Here, the embedded color dithered watermark is in RGB color space and all three channels are embedded into the luminance channel of the image. The watermark can also be converted into YCbCr color space before it is embedded. This way, the implementation of the algorithm at the receiver is much simpler. Besides, it helps in the compression of data to be embedded. For example, instead of embedding the entire color information, we can sub-sample the chrominance components and embed the 4:2:2 or 4:2:0 watermark in the luminance channel of the original image/video frame. This proves to be effective for codecs like H.263 where such kind of sub-sampling is usually adopted.

The extracted watermark quality difference between embedding the RGB channels of the watermark and embedding the YCbCr channels is very little. This can be seen from Fig. 18. The original color watermark, the extracted color watermark from the received image (mean loss probability = 0.15, loss variance = 5%) when RGB channels of the watermark are embedded ($\alpha$ = 5), and when YCbCr channels are embedded ($\alpha$ = 5) are shown in Figs. 18(a), (b) and (c) respectively. In fact, the simplicity in the implementation and compression are the only reasons that make embedding sub-sampled chrominance components more appealing.

The quality of the extracted color watermark can be further improved if different $\alpha$ values are used to embed each of the Y, Cb, Cr channels independently. Fig. 18(d) shows the extracted color watermark when the $\alpha$ values for luminance and the chrominance components are 3.75, 4.5, and 5 respectively. In this case, a reasonable improvement is seen in the quality of the extracted watermark when compared to Figs. 18(b) and (c). The values of $\alpha_Y$, $\alpha_{Cb}$, and $\alpha_{Cr}$ are randomly chosen here and may not be optimum for each of the individual channel's reconstruction. However, better results are expected if such kind of optimization is performed.

### *C. Comparison with other techniques*

Table. II shows the comparison of the proposed technique and the techniques presented in [14]-[18] (obtained from [18]). Note that the proposed algorithm gives an improvement of 1-3 dB over the existing error concealment techniques.

An alternate method to achieve high level of error concealment performance is to transmit the low resolution version of the image/video frame as a encoded side information. The encoded bits of the low resolution image would be generated

TABLE II
PERFORMANCE COMPARISON IN PSNR (DB).

| *Image* | Sun's | Salama's | Wang's | Park's | Li's | Ours |
|---|---|---|---|---|---|---|
| Lena | 23.93 | 23.99 | 24.41 | 24.96 | 26.46 | 28.23 |
| Pepper | 22.19 | 23.69 | 24.06 | 24.48 | 27.25 | 29.47 |
| Zelda | 26.35 | 27.13 | 26.40 | 27.36 | 28.33 | 29.08 |
| Baboon | 17.46 | 18.98 | 19.02 | 17.42 | - | 21.92 |

in the same manner as proposed. They would comprise the side information and transmitted through the same channel instead of embedding them in the source. This approach yield results that are comparable to the proposed technique. However, there are three problems with this approach:

- Side information can be appended to the encoded video, but this addition incurs higher transmission bandwidth. If however, the compression or encryption of the video is increased to accommodate for the side information without increasing the bandwidth, there would be an increase in computational complexity. Data hiding provides a seamless alternative to achieve this trade-off [34].
- Since the effects of the wireless channel are approximated by independent or burst errors from packet losses, there would be significant loss incurred in the low resolution reference. Therefore, recovering the lost data of the high resolution image would be very difficult if either the areas where the losses occur match or higher losses occur in the zoomed up low resolution image.
- More often than not, the compressed mark embedded image requires more bits to be encoded than the compressed original due to the higher entropy of the data hiding process. However, the increase in the number of bits is very small. For example, a *Hockey* video frame of size $240 \times 320$ was $7899$ bytes after compression, while its marker was $1085$ bytes. The mark embedded image was $8015$ bytes after compression. This implies that the transmission of side information bits would require $969$ bytes more (and therefore higher bandwidth) than the encoded data hidden signal.

Simulations have been carried out for the case of transmitting the low resolution image as a data hidden watermark (ECDH) and as encoded side channel information (ECSI) in wireless scenario. Fig. 19(a) gives the PSNR versus loss rate curves of the two techniques. It can be seen that ECDH performs better at higher loss rates (up to 3-4 dB improvement), while the performance of both the techniques is comparable at lower loss rates (up to about $0.2$). Note that at very low packet loss probabilities ($< 0.04$), ECSI tends to give a better performance. This may be due to the fact that the end result of the ECDH suffers from minor data hiding defects, which are rarely visible. Fig. 19(b) shows the variation in the PSNR values of the received low resolution reference images.

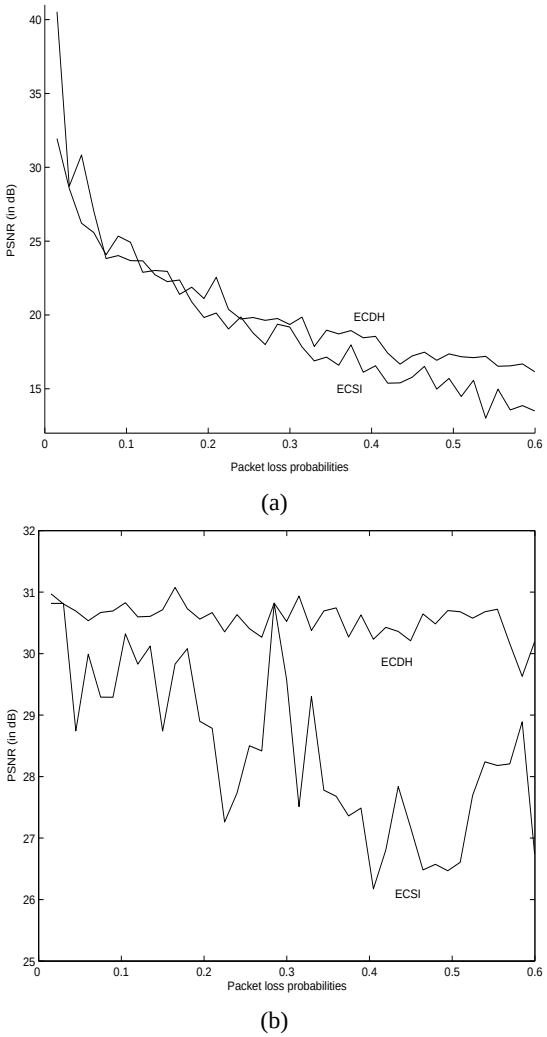



Fig. 19. PSNR vs. loss rate curves for the techniques of error concealment using data hiding (ECDH) and error concealment using side information (ECSI) for the *Sail* image. (a) PSNR of the reconstructed image and (b) PSNR of the retrieved low resolution images.

## VI. CONCLUDING REMARKS

The data hiding technique employed for error concealment is efficient in protecting the low-low wavelet coefficients of the frame from transmission errors by embedding multiple scaled copies of these coefficients in the frame itself. A localized scaling error concealment technique is implemented for improving the perceptual quality of the video frames affected with packet losses. Simulation results with analysis have been presented for various loss rates. Wired to wireless, gray scale to color, and image to video extensions to the proposed algorithm are presented. Possible future work is in the video processing extension. It includes exploiting motion vector information to embed the watermark of the subsequent P-frame in the current I-frame. Another future direction on this would be to identify the areas in the image/video frame where there is a little or no motion and embed the watermark in those areas. This might prove to be useful since viewers often tend to focus on the areas that contain high motion.

Another possible future direction would be to devise a watermarking technique that is robust for this application. Spread spectrum watermarking was considered here as a proof of concept due to its ease of implementation in DCT domain. However, it is not ideal for this application due to its

large spreading gain and higher entropy. Quantization based watermarking may be an alternative. A robust watermarking technique that is targeted for error control applications still needs to be developed.

## VII. Acknowledgements

The first author would like to thank the reviewers for their valuable and insightful suggestions at various stages of the paper revisions.

## References

[1] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communications," *Proc. IEEE, Special Issue on Multimedia Signal Processing*, 1998, pp. 974-997.

[2] S.B. Wicker, *Error Control Systems of Digital Communication and Storage*, Printice Hall, 1995.

[3] T. Kalkar, "Considerations on watermarking security," *Proc. IEEE Workshop on Multimedia Signal Processing*, Cannes, France, October 2001, pp. 201-206.

[4] C.B. Adsumilli, M.C.Q. Farias, M. Carli, and S.K. Mitra, "A hybrid constrained unequal error protection and data hiding scheme for packet video transmission," *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, vol. 5, Hong Kong, April 2003, pp. 680-683.

[5] Y. Liu and Y. Li, "Error concealment for digital images using data hiding," *Proc. Ninth DSP Workshop*, Texas, October 2000.

[6] Y. Shao, L. Zhang, G. Wu, and X. Lin, "Reconstruction of missing blocks in image transmission by using self-embedding," *Proc. Intl. Symp. on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, May 2001, pp. 535-538.

[7] P. Yin, B. Liu, and H.H. Yu, "Error concealment using data hiding," *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, vol. 3, May 2001, pp. 1453-1456.

[8] C.-S. Lu, "Wireless multimedia error resilience via a data hiding technique," *Proc. Intl. Workshop on Multimedia Signal Processing*, Virgin Islands, December 2002, pp. 316-319.

[9] J. Lee and C.S. Won, "A watermarking sequence using parities of error control coding for image authentification and correction," *IEEE Trans. on Consumer Electronics*, vol. 46, no. 2, May 2000, pp. 313-317.

[10] J. Wang and L. Ji, "A region and data hiding based error concealment scheme for images," *IEEE Trans. on Consumer Electronics*, vol. 47, no. 2, May 2001, pp. 257-262.

[11] F. Bartolini, A. Manetti, A. Piva, and M. Barni, "A data hiding approach for correcting errors in H.263 video transmitted over a noisy channel," *Proc. IEEE Workshop on Multimedia Signal Processing*, October 2001, pp. 65-70.

[12] K. Munadi, M. Kurosaki, and H. Kiya, "Error concealment using digital watermarking technique for interframe video coding," *Proc. Intl. Technical Conf. on Circuits/Systems, Computers, and Communications*, July 2002, pp. 599-602.

[13] A. Yilmaz and A.A. Alatan, "Error concealment of video sequences by data hiding," *Proc. Intl. Conf. on Image Processing*, vol. 3, September 2003, pp. 679-682.

[14] P. Salama, N.B. Shroff, E.J. Coyle, and E.J. Delp, "Error concealment techniques for encoded video streams," *Proc. Intl. Conf. on Image Processing*, vol. 1, October 1995, pp. 9-12.

[15] J.W. Park and S.U. Lee, "Recovery of corrupted image data based on the NURBS interpolation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, October 1999, pp. 1003-1008.

[16] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projection onto convex sets," *IEEE Trans. on Image Processing*, vol. 4, April 1995, pp. 470-477.

[17] Y. Wang and Q. Zhu, "Signal loss recovery in DCT-based image and video codecs," *Proc. SPIE Conf. on Visual Communications and Image Processing*, vol. 1605, November 1991, pp. 667-678.

[18] X. Li and M.T. Orchard, "Novel sequential error-concealment techniques using orientation adaptive interpolation," *IEEE Trans. on Circuits and Systems on Video Technology*, vol. 12, no. 10, October 2002, pp. 857-864.

[19] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, 1997, pp. 1673-1687.

[20] M.D. Swanson, M. Kobayashi, and A.H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proc. IEEE*, vol. 86, no. 6, 1998, pp. 1064-1087.

[21] J. Buchanan and L. Streit, "Threshold-diffuse hybrid halftoning methods," *Proc. Western Computer Graphics Symposium (SKIGRAPH)*, 1997.

[22] R.W. Floyd and L. Steinberg, "An adpative algorithm for spatial grayscale," *Proc. Society of Information Display*, vol. 17, no. 2, 1976, pp. 75-78.

[23] Z. Xiong, M.T. Orchard, and K. Ramachandran, "Inverse halftoning using wavelets," *IEEE Trans. on Image Processing*, vol. 8, no. 10, 1999, pp. 1479-1483.

[24] M. Mese and P.P. Vaidyanathan, "Look up table (LUT) inverse halftoning," *IEEE Trans. on Image Processing*, vol. 10, no. 10, 2001, pp. 1566-1578.

[25] M.-Y. Shen and C.-C. Kuo, "A robust nonlinear filtering approach to inverse halftoning," *Journal of Visual Comm. and Image Representation*, vol. 12, 2001, pp. 84-95.

[26] J.J. Lemmon, "Wireless link statistical bit error model," *Dept. of Commerce (Communication and Information)*, NTIA Report 02-394, 2002.

[27] D.L. Lau and G.R. Arce, *Modern Digital Halftoning*, Marcel Dekker, 2001.

[28] H.R. Kang, *Color Technology for Electronic Imaging Device*, SPIE Optical Engineering Press, 1997.

[29] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, vol. 83, no. 6, 1995, pp. 858-876.

[30] A.M. Tekalp, *Digital Video Processing*, Printice Hall, 1995.

[31] K.A. Peker, A.A. Alatan, and A.N. Akansu, "Low-level motion activity features for semantic characterization of video," *Proc. IEEE Intl. Conf. on Multimedia and Expo.*, New York City, August 2000, pp. 801-804.

[32] The network simulator (1995), Retrived April 28, 2003 [online] from http://www.isi.edu/nsnam/ns/

[33] S. Dennett, *The cdma2000 ITU-R RTT candidate submission (0.18)*, chair TR45.5.4, 1998.

[34] M. Wu, "Multimedia data hiding," *Ph.D. Dessertation*, Princeton University, June 2001.

**Chowdary B. Adsumilli** Chowdary B. Adsumilli received his B.Tech degree in Electronics and Communications Engineering from Jawaharlal Nehru Technological University, Hyderabad, India, in 1999 and the M.S. degree in Electrical and Computer Engineering at University of Wisconsin, Madison in 2001. He is currently a Ph.D. candidate in the Image Processing Laboratory in Electrical and Computer Engineering at University of California, Santa Barbara. His research interests include image and video processing, multimedia communications, error resilience and concealment, watermarking, video quality assessment, and wireless video. He has been a member of IEEE since 1995.

**Mylène C. Q. Farias** Mylène Farias received her B.Sc. degree in Electrical Engineering in January 1995 from Universidade Federal de Pernambuco (UFPE) - Brazil and her M.Sc. degree also in Electrical Engineering in July 1998 from the Universidade Estadual de Campinas (UNICAMP)- Brazil. In September 2004, she received her Ph.D. in Electrical and Computer Engineering from the University of California, Santa Barbara. She is currently a Post-Doctoral Researcher in the departments of Electrical and Computer Engineering and Psychology in the University of California, Santa Barbara, under the supervision of Profs. Sanjit K. Mitra and John M. Foley repectively. Her current interests include video quality assessment, video processing, multimedia, watermarking, and information theory.

**Sanjit K. Mitra** Sanjit K. Mitra received the B.Sc. (Hons.) degree in Physics from the Utkal University in 1953, the M.Sc. (Tech.) degree in Radio Physics and Electronics from the Calcutta University in 1956, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of California, Berkeley, in 1960 and 1962, respectively. He has been a Professor of Electrical and Computer Engineering at the University of California, Santa Barbara since 1977, where he served as Chairman of the Department from July 1979 to June 1982. He has served IEEE in various capacities including service as the President of the IEEE Circuits and Systems Society in 1986 and as a Member-at-Large of the Board of Governors of the IEEE Signal Processing Society from 1996-99. He has published over 600 papers in signal and image processing, twelve books, and holds five patents.

Dr. Mitra is the recipient of the 1973 F.E. Terman Award and the 1985 AT&T Foundation Award of the American Society of Engineering Education, the 1989 Education Award, and the 2000 Mac Van Valkenburg Society Award of the IEEE Circuits & Systems Society, the 1996 Technical Achievement Award and the 2001 Society Award of the IEEE Signal Processing Society, the IEEE Millennium Medal in 2000, the McGraw-Hill/Jacob Millman Award of the IEEE Education Society in 2001, and the 2002 Technical Achievement Award of the European Association for Signal Processing (EURASIP). He is the co-recipient of the 2000 Blumlein-Browne-Willans Premium of the the Institution of Electrical Engineers (London) and the 2001 IEEE Transactions on Circuits & Systems for Video Technology Best Paper Award. He is a member of the U.S. National Academy of Engineering, an Academician of the Academy of Finland, a member of the Norwegian Academy of Technological Sciences, a foreign member of the Croatian Academy of Sciences and Arts, and a foreign member of the Academy of Engineering of Mexico. He has been awarded honorary doctorate degrees from the Tampere University of Technology, Finland, and the 'Politehnica' University of Bucharest, Romania. Dr. Mitra is a Fellow of the IEEE, AAAS, and SPIE, and a member of EURASIP.

**Marco Carli** Marco Carli (SM'05) received the Laurea degree in telecommunication engineering from the University of Roma, 'La Sapienza,' Rome, Italy, in 1996. He was with Datamat-Systems Engineering Company until 1997. He is an associate researcher with the University of Rome, 'Roma Tre', where he has been also lecturer for the graduate course telecommunication systems and multimedia communications. From August 2000 he is a visiting research associate with the Image Processing Laboratory, at the University of California, Santa Barbara (UCSB).

His research interests are in the area of digital signal and image processing and in multimedia communications. He is a member of the Technical Committees of several IEEE conferences and a reviewer for IEEE Journals.