Error Concealment Using a Halftoning Watermarking Technique

Sibgrapi paper ID: 102153

Abstract—**In this paper, we propose an error concealment technique for H.264 coded videos. The algorithm is targeted at compressed videos degraded after packet losses caused by transmission over an unreliable channel. We use a combination of watermarking and halftoning (dithering) techniques. At the encoder side, a dithered version of each video frame is embedded into the video using a watermarking technique. The watermarking technique used by the proposed algorithm is a modified version of the Quantization Index Modulation (QIM) algorithm, which provides a good data hiding capacity. At the decoder side, the algorithm identifies which packets of the video were lost, extracts the corresponding mark, and applies an inverse halftoning technique to estimate the original content. The algorithm is fast and has a good performance, being able to restore content with a better quality than the default H.264 error concealment algorithm.**

Keywords-**Error concealment, H.264, QIM, watermarking, halftoning.**

I. Introduction

The Internet provides a best effort model, which means losses may happen when packets are being transmitted. In the majority of applications, errors caused by losses are undesirable. In particular, for video transmission, errors may decrease the quality of received content, affecting the level of acceptability and popularity of the service. Error concealment algorithms are frequently used to minimize the effect of transmission errors on video content. These algorithms use the temporal and spatial information of the received ‘pixels’ to estimate the ‘lost’ data. Most error concealment methods are implemented as a part of coding and decoding algorithms, like H.264 [1].

Various approaches have been proposed to conceal transmission errors in videos and images, with interpolation being the most common technique [2], [3]. The main advantage of interpolation algorithms is their low complexity. Another advantage is that they do not require changes to the video encoder. However, the performance of these algorithms is generaly limited by the compression rate and the type of content that was lost.

A different approach consists of using watermarking techniques to embed redundant information into the video or image (host signal). At the decoder/receiver, this information can be extracted, making it possible to recover content lost during the transmission. For example, the works of Adsumilli *et al.* [4] and Phadikar and Maity [5] use watermarking techniques to self-embed a dithered halftoned version of the content. On the other hand, the work of Nayak *et al.* [6] uses a hybrid method composed of multiple techniques, including watermarking, to conceal lost information in compressed videos. In general, the watermarking techniques provide a better quality of restored content than interpolation techniques. On the other hand, these techniques frequently introduce distortions into the host signal and require that modifications be added to the video encoder and decoder, causing an increase in complexity.

In this paper, we propose a watermarking based error concealment technique that is simple, fast, and recovers lost content with a very good quality. The algorithm is implemented together with the H.264 codec, what reduces complexity and avoids introducing distortions into the signal. The paper is divided as follows. In Section II, we describe the proposed system, detailing the halftoning, watermark embedding and extraction, inverse halftoning and reconstruction stages. In Section III, we present the simulation results. Finally, in Section IV, we present our conclusions.

II. Proposed Algorithm

The proposed algorithm has the goal to recover portions of the video lost in a transmission over an unreliable channel. At the encoder side, a halftoning algorithm generates a dithered halftoned version of each video frame. This information is embedded into the video using a *fragile* watermarking technique that is a modified version of the Quantization Index Modulation (QIM) algorithm [7]. At the decoder side, the algorithm identifies which packets of the video were lost. Then, for each lost packet, the original content is recovered by extracting the mark corresponding to the affected area and using an inverse halftoning algorithm to convert the dithered version of the frame into a colored multi-level approximation of the original.

The block diagram of the proposed algorithm is depicted in Fig. 1. Notice that this block diagram depicts stages of the H.264 codec combined with stages of the proposed error concealment algorithm. The idea here is to show that the proposed algorithm can be implemented together with the H.264 codec (or any other compression algorithm) or by itself. In this section, we describe in further details the stages of the proposed error concealment algorithm.

A. Halftoning Stage

Halftoning is a technique for converting multi-level images into binary images using patterns of white and black dots [8]. This technique creates the illusion of seeing multiple intensity levels in a binary image, what makes it suitable for applications where only a reduced number of levels is available, such as newspapers, fax machines, and document

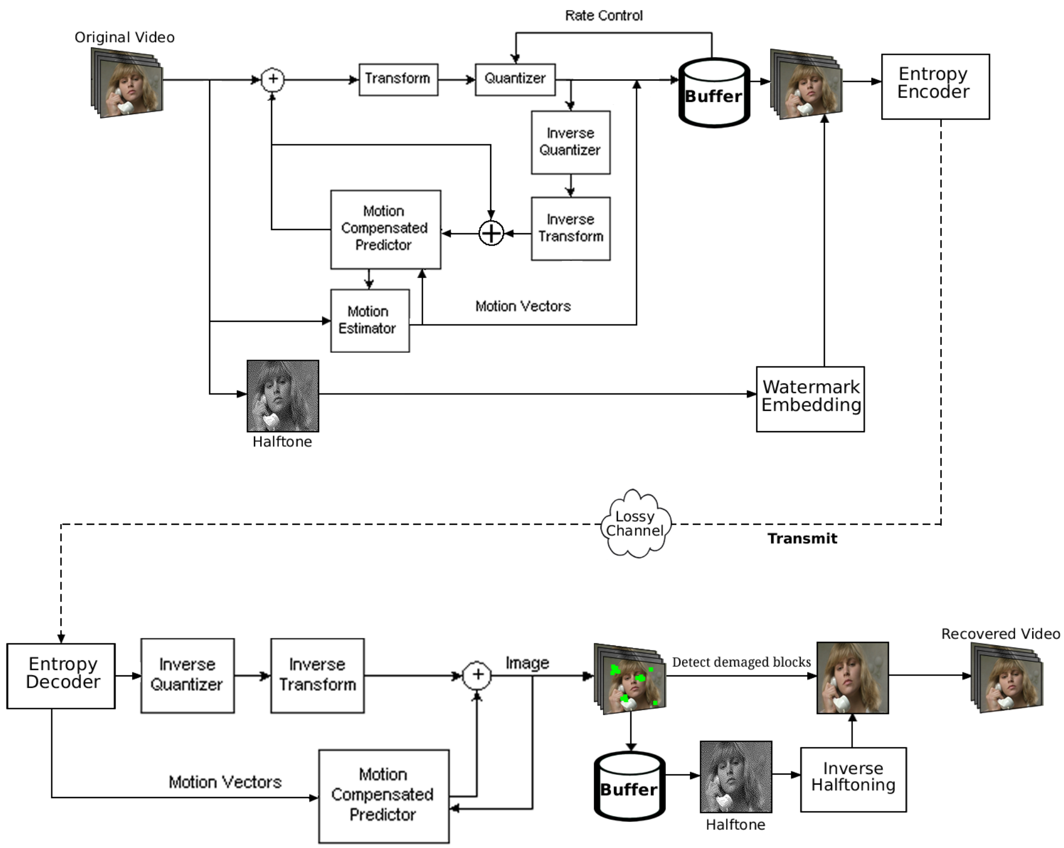


Fig. 1. Block diagram of the proposed system, which detects lost regions in a video and restores the original content.

printing processes. In the literature, there are several halftoning algorithms that can be roughly divided into two major classes: ordered dithering and error diffusion [9], [10].

Error diffusion algorithms compare the pixel intensity values with a fixed threshold. The resulting error between the output value and the original value is distributed among the neighboring pixels according to predefined weights. This provides interesting visual results and eliminates the local correspondence between the halftoned image and original gray level. *Ordered dithering* algorithms generate halftoned images by comparing each pixel value with a set of pixel clusters. The spatial patterns used by the chosen set of pixel clusters determine the quality of the dithered halftoned image. Both ordered dithering and error diffusion techniques have advantages and disadvantages. The choice of the type of halftoning algorithm often depends on the target application.

In this work, an ordered dithering algorithm is used to generate a dithered halftoned version of each picture frame, which is later embedded into the video itself. This type of halftoning technique is chosen because it uses sets of pixel clusters that have a predictable pattern. For each pixel of the picture, the algorithm only uses the pixel value and a fixed number of spatial patterns to generate the halftoned picture. Given that our target application requires generating a multi-level picture from a halftoned picture with (possibly) lost content, using an easily predictable pattern is very important.

As pointed out previously, in case of content loss the halftoned version of the picture frame is used to restore the video back to its original state. The quality of the restored video depends on the halftoning algorithm and of the number of intensity levels it is able to represent. So, in order to guarantee a good quality, the halftoning algorithm must be able to represent the largest possible number of intensity levels with a minimum number of bits. Unfortunately, the data hiding capacity of the host picture frame is relatively low. Our tests show that a maximum of 3 bits per pixel can be embedded in the host picture frame without causing visible degradations.

Because of this limited data hiding capacity, we cannot use classical ordered dithering patterns. So, we propose to use *combinatorial dispersed-dot patterns* which are capable of generating all combinations of bits necessary to represent the intensity levels, but require less bits than classical ordered dithered patterns. This way, we can increase the number of mapped intervals without increasing the size of the dot-pattern matrix. For example, using a 3×3 Bayesian matrix to generate the dispersed-dot dithering, we obtain 10 distinct levels. On the other hand, using a 3×3 combinatorial matrix allows for $2^{3\times 3}$ distinct configurations, that translates into up to 512 intervals. Since 3 bits are available for our application, we can have 2^3 combinations and, consequently, 8 different intervals, as shown in Fig. 2.

Pictures reconstructed from dispersed-dot dithered halftoned

(a) (b) (c)

Fig. 3. Halftoning algorithm: (a) Original frame, (b) Edge enhanced frame, (c) Halftoned frame.

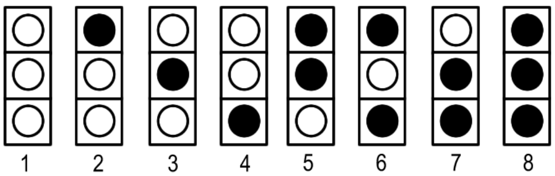


Fig. 2. Combinatorial dispersed-dot patterns used for generating dithered images with 3 bits, which allows mapping 8 intervals.

images can be slightly blurred. So, before we generate the halftoned picture we apply an unsharp-mask edge enhancement filter to the original picture frame (I_o), generating an image with enhanced details (I_{eh}). In Fig. 3(b), an example of this enhancement step is depicted for a frame of the video "Foreman" presented in Fig. 3(a).

Then, we quantize I_{eh} using 8 distinct intervals, using the following equation:

$$I_q(x,y,c) = \left\lceil \frac{8}{255} I_{eh}(x,y,c) \right\rceil, \tag{1}$$

where x and y are the horizontal and vertical spatial dimensions, respectively, c refers to the color channel ($1 \le c \le 3$), and I_q is the quantized image. Next, we substitute the value of each pixel in I_q by the corresponding combinatorial dispersed-dot patterns, showed in Fig. 2, generating the haltoned image I_{dth}. Finally, for each color channel, the 3 bits are converted to an integer number. In Fig. 3(c) the halftoned picture corresponding to the frame shown in Fig. 3(a) is presented.

B. Watermarking Embedding Stage

After generating the mark using the techniques described in the previous section, the next stage consists of embedding it into the host image or video frame. But, in order to make it possible to recover the original content, the dithered mark corresponding to a specific region cannot be embedded in the same spatial and temporal position of the host image or video frame. Therefore, we spatially distribute the mark over the host picture using a *split-flip operation*, which consists of splitting the halftoned image into sub-blocks and flipping them to a different spatial region. More specifically, we divide the halftoned picture in 64×64 blocks, rotate each sub-block by 180 degrees, and shuffled the regions. Fig. 4 shows one example of the process for the three color channels of the image "Lena".

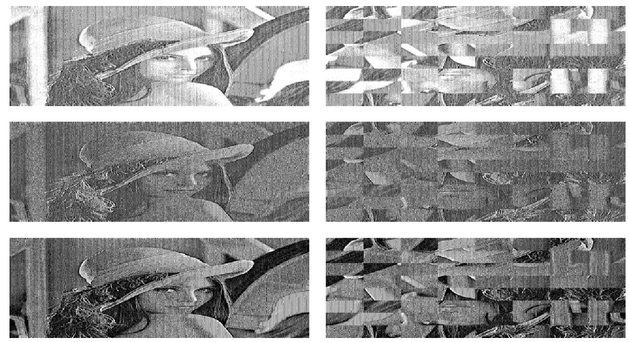


Fig. 4. Original dithered halftoned versions of each color channel of the image "Lena" (left) and the corresponding split-flip versions (right).

For videos signals, we also perform a temporal distribution of the mark by inserting the mark corresponding to the current picture frame in a previous picture frame, located 1 second before. By distributing the mark spatially and temporally, a specific region does not store the mark necessary to restore it, what increases the probability that the algorithm is able to restore the content to its original version.

Among the available watermarking methods, the Quantization Index Modulation (QIM) algorithm is one of the methods with the best performance [7]. The QIM algorithm inserts a mark into a host picture by quantizing it with a uniform scalar quantizer. The standard quantization operation with step size δ is defined by the following equation

$$Q(x,\delta) = \left\lfloor \frac{x}{\delta} \right\rfloor \cdot \delta, \tag{2}$$

where $\lfloor t \rfloor$ is the largest integer not greater than t. The watermarked pixel is obtained using the following equation

$$s(x) = Q(x,\delta) + d(m), \tag{3}$$

where $d(m)$ is the perturbation value, m is the mark signal to be embedded, and $Q(x,\delta)$ is the quantization function defined in equation 2. The value of the step size δ determines the data hiding capacity of the algorithm. The higher δ, the more bits per pixel can be inserted in the host. The value of δ also affects the level of distortion introduced in the original by the watermarking algorithm.

In this work, we propose a modification of the QIM algorithm that inserts an integer mark, instead of a function of a binary mark. In other words, the modulation function in equation 3 is set to $d(m) = m$. Therefore, the integer-halftoned mark, I_{dth}, is inserted in each pixel of the corresponding color channel of the original picture frame using the following equation:

$$I_m(x, y, c) = Q\left(I_o(x, y, c), \delta\right) + I_{dth}(x, y, c), \quad (4)$$

where I_m is the resulting watermarked picture frame, I_o is the original picture frame, δ is the quantization step, and c is the corresponding color channel.

C. Watermarking Extraction Stage

The watermarking extraction is performed at the receiver (decoder side), after the transmission of the video. First, we create a buffer containing the current picture frame and all the frames in the previous 1 second interval. After that, we are ready to restore losses in the next picture frames.

When a packet loss is detected by the H.264 decoder, we extract the corresponding mark from the buffer using the following equation:

$$\hat{I}_{dth}(x, y, c) = I_m(x, y, c) \mod \delta, \quad (5)$$

where I_m is the watermarked color channel and $\hat{I}_{dth}$ is the recovered integer-halftoned mark. Then, the extracted integer-halftoned mark is converted to a binary number of 3 bits in order to restore the halftoned picture frame.

D. Inverse Halftoning Stage

If any region is classified as "lost" in the current frame, we search the correspondent mark in the appropriate location of the buffer. Then, we use the inverse halftoning algorithm to generate a multi-level colored picture frame, which is an approximation of the original picture frame.

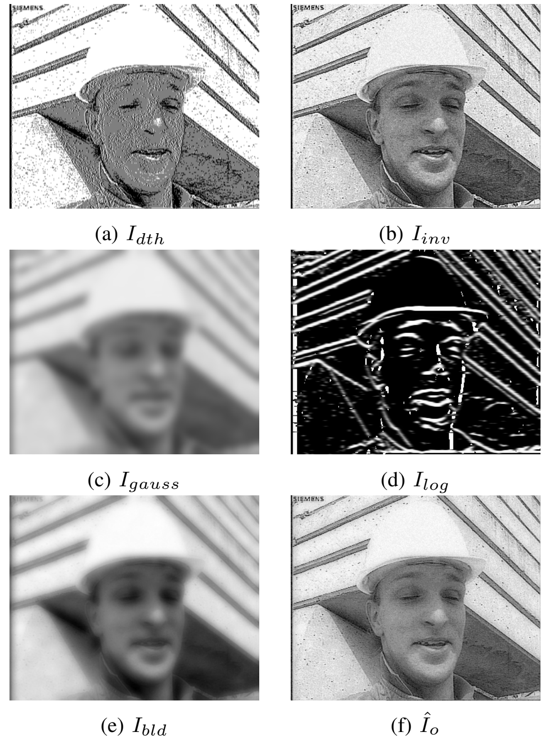
Given that I_{dth} is the halftoned picture frame, $D(p)$ is the distribution of the area surrounding the pixel p in I_{dth}. To reconstruct an 8-bit pixel from the halftoned picture, we first calculate the local distribution $D(p)$ for all pixels in I_{dth}. From this distribution, we find the corresponding mapped interval that contains the most probable pixel value in the corresponding color channel, according with the indices of the dot-patterns, shown in Fig.2. Once this interval is found, we randomly select a value within it, generating a slightly noisy picture I_{inv}.

Then, we filter I_{inv} with a Gaussian lowpass and a Laplacian-of-Gaussian filter [11]. The idea here is to spatially decorrelate the pixels in order to be able to process each pixel independently. This allows for an independent reconstruction, even when the neighboring pixels are missing. The resulting picture frames, I_{gauss} and I_{log}, are used to compose another picture, I_{bld}, given by the following equation:

$$I_{bld}(x, y, c) = \Upsilon\ I_{gauss}(x, y, c) + (1 - \Upsilon)\ I_{log}(x, y, c), \quad (6)$$

where Υ is the blending-ratio matrix that determines the proportion of each input filtered picture in the output.

In our simulations, we observed that using $\Upsilon = I_{inv}$ preserves the edges of the pictures. Unfortunately, when we combine all 3 channels, the resulting picture frame contains



(a) I_{dth} (b) I_{inv} (c) I_{gauss} (d) I_{log} (e) I_{bld} (f) $\hat{I}_o$

Fig. 5. Inverse halftoning algorithm steps: (a) extracted halftoned mark, (b) random inverse estimative, (c) image filtered using Gaussian low-pass filter, (d) image filtered using LoG filter, (e) blind combination of I_{gauss} and I_{log}, (f) restored image.

visible color distortions. If, on the other hand, we use Υ as a constant matrix, I_{bld} is a blurred version of I_{inv}. An example of this can be seen in Fig. 5. In Figs. 5(a)-(d), the extracted halftoned picture, the filtered images, and their combination are shown, respectively. Notice that the combination I_{bld} is a very blurred picture (see Fig. 5(d)).

Hence, it is necessary to make another composition using I_{inv} and I_{bld}, with the goal of minimizing color distortion and keeping the details of the original image. The composition, $\hat{I}_o$, is the final result of inverse halftoning process and is given by the following equation:

$$\hat{I}_o(x, y, c) = \left\lfloor \sqrt{I_{inv}(x, y, c)\ I_{bld}(x, y, c)} \right\rceil, \quad (7)$$

where $\hat{I}_o$ is the recovered 8-bit version of original video frame, I_o. This way, we recover the original content with the best visual quality possible. An example of this process for the pictures in Fig. 5 is shown in Fig. 5(e).

III. Test Results

First, we tested the degradation of the original content caused by the proposed method, in particular by the insertion of the watermark. In Fig. 6, we show a comparison between original and watermarked picture frame for the videos "Carphone", "Foreman", "Mobile", and "Suzie". These videos are publicly available of format YUV 4:4:4 color CIF (352 ×288, progressive) with around 300 frames each. From this figure, we can observe that the addition of the mark cause very small distortions in the colors of the frame due to quantization. However, this effect is not very visible without having the original frame as reference.

In Table I, we show the values of the Peak signal-to-noise ratio (PSNR), the Structural similarity (SSIM) [12], and Universal Image Quality Index (UQI) [13] metrics for the marked videos corresponding to the four originals. The values obtained with these metrics suggest that the marked videos have a good quality and that the small degradations present in these videos are visually acceptable.

Fig. 6. Comparison of the original (left) and marked (right) images. From top to down: "Carphone", "Foreman", "Mobile" and "Suzie".

Second, we tested our algorithm using a set of still images modified with a percentage of 16×16 blocks deleted. To implement this test, we divided the images in blocks of 16×16 pixels. Then, we deleted (substituted the original content by the value 0) of a fixed percentage of the blocks, with spatial positions chosen randomly. The percentage of lost blocks used in this test varied from 5% to 25%.

Fig. 7 shows three examples of the restoration of lost blocks

Video	UQI	PSNR	SSIM
Suzie	0.90069	41.20822	0.97837
Carphone	0.89881	38.91160	0.97383
Foreman	0.88333	39.17669	0.97625
Mobile	0.93128	38.19763	0.97615

TABLE I
PSNR AND SSIM VALUES CALCULATED BETWEEN ORIGINAL AND WATERMARKED VIDEOS.

using the proposed error concealment algorithm. The images on the upper line have 20% of the blocks deleted, while the images on the bottom are the corresponding restored versions. Notice that the quality of the reconstructed images is very good and it is difficult to identify the location of the restored blocks. We calculated the PSNR and SSIM values between the original and restored images of the test cases. Figs. 8 and 9 depict the graphs of SSIM and PSNR versus the percentage of deleted blocks for all test images. It can be observed that, as expected, the quality of the images decreases with the percentage of deleted blocks.

Third, we tested the ability of the proposed algorithm to recover lost packets from H.264 compressed videos. We embed the proposed error concealment algorithm in a H.264 codec to obtain a protected video stream. In order to simulate packet losses in a given bitstream, we use a simple model that simulates packet losses over error-prone channels. The loss model discards groups of H.264 packets (Network Abstraction Layer units – NALs) from the middle of the bitstream, according to the desired packet loss rate (PLR) values. All packets are treated equally regarding their susceptibility to loss, i.e. we did not focus on specific types of packets, such as those carrying intra-coded slices. We tested PLR values of 0.5%, 1%, 3%, 5%, and 10%.

For these test cases, we calculated the PSNR and the SSIM values between the original and restored videos. The Figs. 10 and 11 depict the graphs of the values of PSNR and SSIM, respectively, versus the PLR values for all tested videos. In these graphs, the continuous lines represent the comparisons using the proposed method and the dashed lines represent the comparisons using the default H.264 error concealment algorithm. It can be observed that, as expected, the quality of the videos decreases with the PLR value. Also, the proposed algorithms always has a better performance (both in terms of PSNR and SSIM) than the default H.264 error concealment algorithm.

Fig. 12 depicts an example of the use of the proposed algorithm to mitigate lost packets for the videos "Foreman" and "Mobile". The first column of the figure shows sample frames of the original videos. The second column shows sample frames of the videos recovered with the default H.264 error concealment algorithm for 3% PLR. The third column shows sample frames of the videos restored using the proposed algorithm, also for 3% PLR. The method is able to get rid of common visible distortions that are commonly seen on videos restored using the standard H.264 error concealment algorithm, like for example blocking effects, packet losses, and

Fig. 7. Illustration of restoration of lost blocks using proposed error concealment algorithm: (top) pictures with 20% of content lost and (bottom) restored pictures using the proposed algorithm.

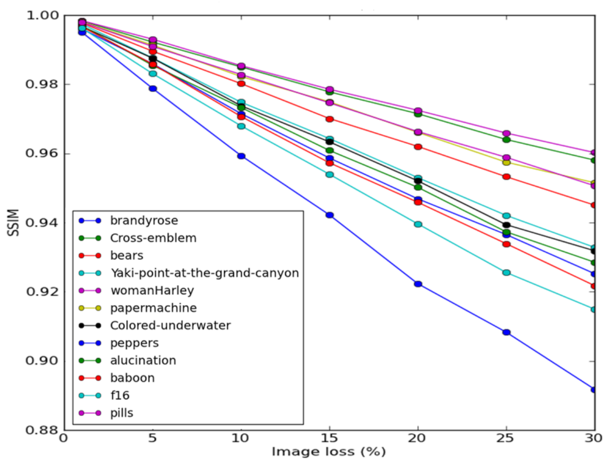


Fig. 8. SSIM values for reconstructed images versus percentage of deleted blocks.

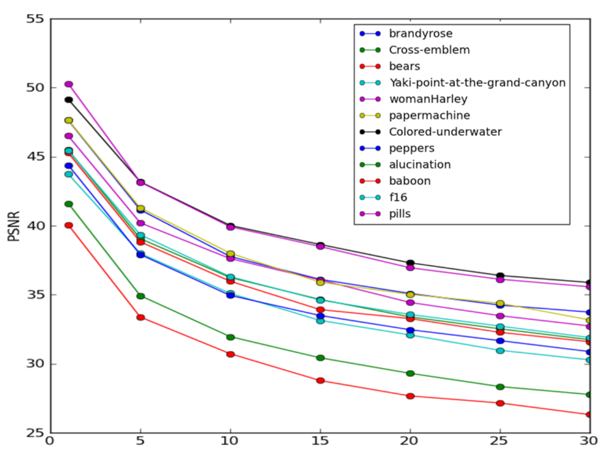


Fig. 9. PSNR values for reconstructed images versus percentage of deleted blocks.

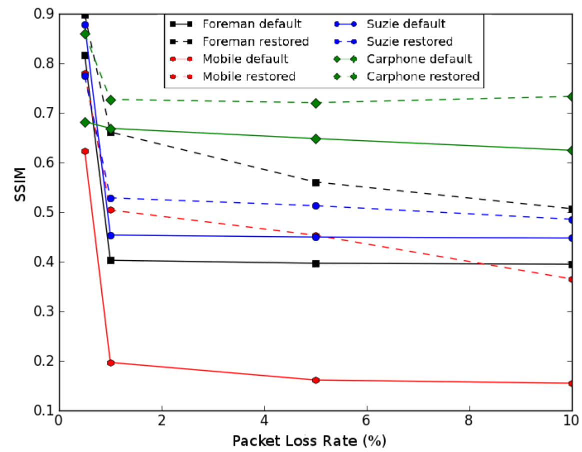


Fig. 10. SSIM values for for reconstructed videos versus packet loss rate (PLR) values.

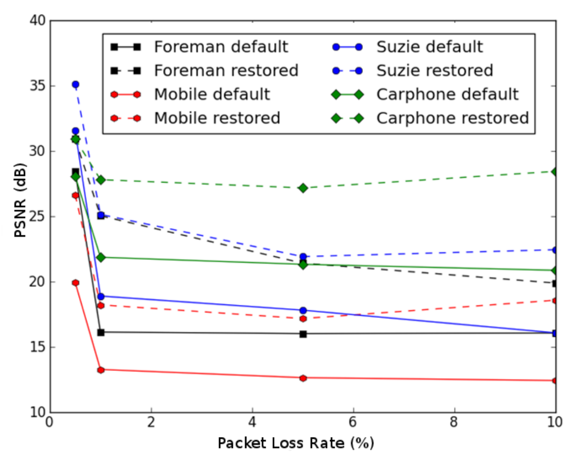
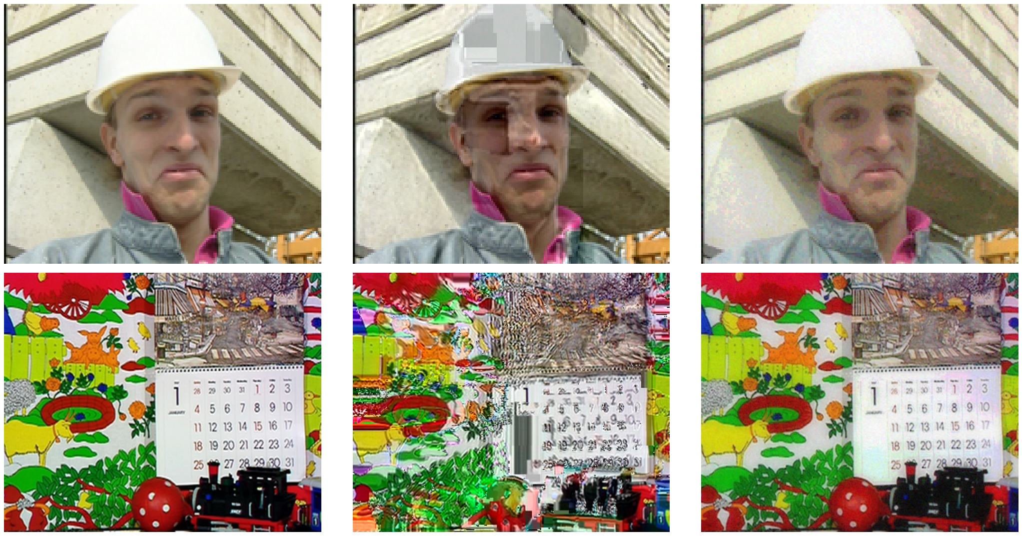


Fig. 11. PSNR values for reconstructed videos versus packet loss rate (PLR) values.

false contours. In fact, we observed that, for PLR values below 5%, the restored videos present few distortions in comparison to the original videos.



(a) (b) (c)

Fig. 12. Comparison between the watermarked frame and the restored frame after a lossy transmission: (a) original frames, (b) restored frames using default H.264 error concealment algorithm, (c) restored frames using the proposed algorithm.

IV. Conclusions

In this paper, we present a new technique for error concealment, which combines watermarking and halftoning algorithms. The proposed system is able to recover the original content with very good visual quality. The protected (watermarked) videos present good quality, showing few visible distortions. The proposed algorithm is fast and restores the picture frames with a better visual quality than the standard H.264 error concealment algorithm.

Future works include further investigating techniques to enhance the quality of marked pictures by reducing noise degradations created in the watermarking stage. Moreover, the proposed algorithm can be modified to take into account aspects of the human visual system (HSV), such as how human viewers perceive colors and motion. For example, we can insert the watermark in less salient regions, so that the degradation caused by the watermarking algorithm are less visible.

References

[1] S.-H. Yang and J.-C. Tsai, "A fast and efficient h.264 error concealment technique based on coding modes," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2010 IEEE International Symposium on*, march 2010, pp. 1 –4.

[2] J. Zhou, B. Yan, and H. Gharavi, "Efficient motion vector interpolation for error concealment of h.264/avc," *Broadcasting, IEEE Transactions on*, vol. 57, no. 1, pp. 75 –80, march 2011.

[3] H. Sun, P. Liu, J. Wang, and S. Goto, "An efficient frame loss error concealment scheme based on tentative projection for h.264/avc," pp. 394–404, 2011, 10.1007/978-3-642-15696-037. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15696-037

[4] C. Adsumilli, M. Farias, S. Mitra, and M. Carli, "A robust error concealment technique using data hiding for image and video transmission over lossy channels," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 11, pp. 1394 – 1406, nov. 2005.

[5] A. Phadikar and S. Maity, "Roi based error concealment of compressed object based image using qim data hiding and wavelet transform," *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 2, pp. 971 –979, may 2010.

[6] C. K. Nayak, J. Surendran, S. N. Merchant, U. B. Desai, and S. Sanyal, "Error concealment of h.264 encoded video through a hybrid scheme," in *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, ser. MEDES '10. New York, NY, USA: ACM, 2010, pp. 189–195. [Online]. Available: http://doi.acm.org/10.1145/1936254.1936286

[7] B. Chen and G. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *Information Theory, IEEE Transactions on*, vol. 47, no. 4, pp. 1423–1443, 2001.

[8] D. E. Knuth, "Digital halftones by dot diffusion," *ACM Trans. Graph.*, vol. 6, pp. 245–273, October 1987. [Online]. Available: http://doi.acm.org/10.1145/35039.35040

[9] J.-M. Guo and H. Lee, "Watermarking in halftone images with mixed halftone techniques," *Int. J. Imaging Syst. Technol.*, vol. 17, no. 5, pp. 303–314, Feb. 2007. [Online]. Available: http://dx.doi.org/10.1002/ima.v17:5

[10] D. Anastassiou and K. S. Pennington, "Digital halftoning of images," *IBM Journal of Research and Development*, vol. 26, no. 6, pp. 687 –697, nov. 1982.

[11] J. S. Chen, A. Huertas, and G. Medioni, "Fast convolution with laplacian-of-gaussian masks," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-9, no. 4, pp. 584 –590, july 1987.

[12] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600 –612, april 2004.

[13] Z. Wang and A. C. Bovik, "A universal image quality index," *Signal Processing Letters, IEEE*, vol. 9, no. 3, pp. 81–84, 2002. [Online]. Available: http://dx.doi.org/10.1109/97.995823