Garcia Freitas *et al. Journal of the Brazilian Computer Society* (2016) 22:5
DOI 10.1186/s13173-016-0046-3

Journal of the
Brazilian Computer Society

RESEARCH Open Access

Secure self-recovery watermarking scheme for error concealment and tampering detection

Pedro Garcia Freitas*, Ronaldo Rigoni and Mylène C. Q. Farias

Abstract

Background: In this paper, we present a method for protecting and restoring lost or tampered information in images or videos using secure watermarks. The proposed method consists of a combination of techniques that are able to detect image and video manipulations. Contrary to most existing watermarking schemes, the method can identify the exact position of the tampered region. Furthermore, the method is capable of restoring the manipulated information and retrieve the original content. This set of capabilities make it possible to use the proposed method in error concealment and digital tampering applications.

Methods: The proposed method is employed as both an error concealment algorithm and a tampering detection algorithm. The proposed method is divided into two stages. At the encoder side, the method generates a binary version (watermark) of the original picture (image or video frame) using a halftoning technique. Then, a quantization index modulation technique is used to embed this watermark into the protected picture. At the decoder side, after the lost or tampered regions are identified, the original content is recovered by extracting the watermark corresponding to the affected areas. An inverse halftoning algorithm is used to convert the dithered version of the picture into a good-quality multi-level approximation of the original content.

Results: First, we test the method in error concealment applications, using a set of still images and H.264 videos. Then, we test the proposed method for tampering detection and content retrieval applications, again considering both images and videos. We compare the proposed method with several other several state-of-the-art algorithms. The results show that the proposed method is fast, robust, and accurate.

Conclusions: Our results show that we can use a single approach to tackle both error concealment and tampering detection problems. The proposed method provides high levels of security, high detection accuracy, and recovery capability, and it is robust to several types of attacks.

Keywords: Quantization index modulation, Watermarking, Error concealment, Tampering detection

Background

The flexibility of digital images and videos is both a blessing and a curse. Digital technologies make it possible to create high-quality pictures, animations, games, and special effects with an amazing realism. Digital pictures (images and videos) can be enhanced, compressed, transmitted, translated across different standards, and displayed in a variety of devices. Because of the significant advances in compression and transmission techniques, it is possible to deliver high-quality visual content to the end user in many different ways. As a consequence, a variety of delivery services have been created in the last years, such as direct TV broadcast satellite, digital broadcast television, and IP-based video streaming.

In content delivery applications, video and image signals are transmitted in a compressed format [1] and they are divided into packets before transmission. Unfortunately, during the transmission over wired and wireless channels, some packets may be lost or delayed. These transmission losses cause various types of visible degradations that may

*Correspondence: sawp@sawp.com.br
Department of Computer Science, University of Brasília, Campus Universitário Darcy Ribeiro, 70919-970 Brasília, DF, Brazil

affect the quality of the perceived content. In video signals, transmission losses cause spatial and temporal degradations. These degradations may again affect the overall perceived video quality, what influences the acceptability and popularity of the application.

To minimize the effect of transmission errors, error resiliency techniques are often used. These techniques can be classified as forward, interactive, and concealment techniques [2]. Forward techniques add redundant data to the video. Although this increases the amount of data to be transmitted, *forward* methods have the advantage of not requiring interaction between the encoder and the decoder. On the other hand, *interactive* techniques use a feedback channel that allows the decoder to request the encoder to resend data. A transmission of redundant data avoids wasting resources but may introduce delays and locks that make it unsuitable for real-time video streaming applications [3]. In contrast to these approaches, *concealment* techniques usually do not increase the amount of data to be transmitted or require a side channel [4]. They can be implemented with or without a modification to the encoder. Therefore, concealment techniques are very attractive for real-time applications that require a low bit-rate and delay [5].

Although there are several possible approaches, most error concealment algorithms use error prediction techniques, like interpolation [6, 7]. For example, Lin et al. [8] use an advanced interpolation technique. Their algorithm makes a partition decision considering information from previous frames. Ranjan et al. [9] propose a method that uses an affine transformation that uses a speed-up robust features (SURF) algorithm to find a relation between current and previous frames and, then, predict the lost data. Koloda et al. [10] proposes an error concealment technique that is based on sparse linear prediction.

Other approaches that promote error resiliency combine concealment and forward techniques. Basically, the error is concealed using information previously embedded into the original signal, what does not increase the amount of data to be transmitted. If some part of the data is lost during the transmission, the embedded information can be extracted and used to recover the original data. The signal acts as a channel (host) to transmit the hidden data. Often, this approach is related to dirty paper coding [11], specially when the data hiding technique is the quantization index modulation (QIM) algorithm [12, 13].

The use of data hiding reduces the decoder's complexity. One example of this approach is the technique proposed by Yin et al. [14] that uses data hiding to embed a set of features extracted from the signal at the encoder. Similarly, Chung et al. [15] use a reversible data hiding technique to insert motion vectors (MVs) into zero quantized discrete cosine transform (QDCT) coefficients and perform intra-frame error concealment for H.264/AVC coded video sequences. A general method for recovering missing DCT coefficients in DCT-transformed images is proposed by Li et al. [16]. The work of Xu et al. [17] is an improvement of Chung's method, which exploits the number of coefficients that needs to be modified to extract the hidden data. Their work considers the residual blocks produced by the data hiding algorithm. Wang et al. [18] present an image restoration method that is based on on a linear optimization model and restores part of the image from structured side information.

A slightly different approach was proposed by Adsumilli et al. [19]. Their technique uses a spread-spectrum watermarking algorithm to embed a dithered version of a picture frame into the host video. Navak et al. [20] improved Adsumilli's technique by adding motion estimation vectors with edge-correlated information. Both schemes embed a low-resolution version of the picture or video frame into the original video content using a spread-spectrum watermarking technique. At the receiver side (decoder), the embedded watermark is extracted from the received video frame and is used as a reference for reconstructing the original signal. Unfortunately, the disadvantage of these two works is that the quality of the reconstruction is low because of the low capacity of the spread-spectrum data hiding scheme.

In addition to error concealment techniques, another important concern for image and video applications is tamper detection and copyright protection. Nowadays, there are several softwares for editing images and videos, making it very easy to alter (tamper) content without leaving any clear sign. Several techniques have been proposed with the goal of detecting tampered areas and reconstructing the original content [21]. These techniques share similar goals with the error concealment algorithms. The difference is that in tampering detection the "non-original" areas are deliberately modified, while in the error concealment case the "non-original" areas are damaged or lost. Tampering detection techniques can be divided into no-reference, reduced-reference, and full-reference approaches. Full-reference approaches require the full original picture to determine whether the picture is tampered. Reduced-reference approaches need some information of the original to determine the tampering but do not require the full original picture. No-reference approaches detect tampering using only the tested picture, i.e. they do not require any information from the original content. For transmission applications, the original is not available. Therefore, no-reference approaches are the most adequate ones. Most no-reference techniques are specialized in detecting only one type of tampering [22, 23], what is not generally not very useful in practical applications.

Similarly to the error concealment techniques proposed by Adsumilli et al. [19] and Navak et al. [20], one popular

no-reference tampering detection approach consists of using data hiding. Imaizumi et al. [24] detect and locate tampered areas in images using a reversible data hiding and a low cost scheme with efficiently multiplexed layers. The method proposed by Xu et al. [25] embeds data directly into the encrypted H.264/AVC video bitstream. Phadikar et al. [26], on the other hand, proposes a tamper detection and correction scheme based on a novel semi-fragile data hiding technique. This method is based on an integer wavelet transform and a quantization index modulation (QIM) algorithm. Tong et al. [27] also use a watermarking algorithm to detect and localize tampered areas. Lin et al. [28] present an authentication and recovery method for tampered images.

Among the methods available in the literature, few approaches address the problem of restoring the original content with good quality. Dadkhah et al. [29] propose an effective tamper detection that uses a singular value decomposition (SVD) in a self-recovery algorithm. In a more recent work, Som et al. [30] propose a discrete wavelet transform (DWT) watermarking scheme for tamper detection, localization, and restoration targeted at *cropping* attacks. Although these methods have a good performance (in terms of quality), they are very specialized and only work for one or two types of attacks.

In this paper, we present a method that works both as an error concealment technique and a tampering detector with restoration capability. The proposed method is based on watermarking and halftoning techniques. At the encoder side, the method generates a binary version of the original picture (image or video frame) using a halftoning technique. Then, a watermarking technique is used to embed this mark into the host content. The watermarking technique used is a modification of the QIM that achieves a higher data hiding capacity than traditional methods. For tampering detection, a ciphered key is also embedded into the host video to allow spatial and temporal localization of tampered regions. At the decoder side, after the modified (degraded or tampered) regions are identified, the original content is recovered by extracting the halftone image corresponding to the affected areas. An improved inverse halftoning algorithm is used to convert the dithered picture into a good quality approximation (colored) of the original picture. The quality of the reconstructed areas is higher than for other algorithms available in the literature. The tampering method is generic enough to recover tampered areas independently of the type of attack. Finally, the proposed method is among the few methods in the literature that also work for video signals, detecting and reconstructing (spatially and temporally) modified regions in videos with a good perceived quality and few temporal artifacts.

The rest of this paper is divided as follows. In the "Halftoning" section, the halftoning method is described. In the "Watermarking embedding" section, the watermarking embedding stage is explained. In the "Watermarking extraction" and "Inverse halftoning" sections, the watermarking extraction and inverse halftoning stages are detailed. In the "Error concealment algorithm" and "Tampering detection algorithm" sections, we describe the implementation and simulation results of the two main applications of the proposed system: error concealment and tampering detection algorithm. Finally, the "Conclusions" section presents the conclusions.

Methods

Halftoning

Halftoning is a technique for converting multi-level images into binary images using patterns of white and black dots [31]. 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 printing processes.

In this work, a halftoning algorithm is used to generate a dithered version of each picture (still image or video frame), which is later embedded into the host picture itself. At the decoder, if any loss or tampered area is detected, the dithered version of the picture is recovered and used to restore the content back to its original state. Therefore, the quality of the restored image or video depends strongly on the efficiency of the halftoning algorithm. One of the contributions of this work is the design of a halftoning and an inverse halftoning algorithms that are able to generate simple dithered images that can be later inverted with a good quality.

In order to guarantee a good quality, the halftoning algorithm should be able to represent the largest possible number of intensity levels with a minimum number of bits. A simple halftoning technique that satisfies these requirements is the *ordered dithering algorithm*. This class of algorithms generates dithered pictures with sets of pixel clusters that have a predictable pattern (Fig. 1).

To promote high data hiding capacity, we propose a combinatorial dispersed-dot dithering pattern. This dithering method is capable of generating all combinations of bits necessary to represent a number of intensity levels [32, 33]. 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 a dispersed-dot dithering, we can represent 10 distinct intensity levels. On the other hand, using a 3×3 combinatorial matrix allows for up to 512 intensity levels.

Pictures reconstructed from dispersed-dot dithered images can be slightly blurred. To avoid blurring artifacts, we apply an unsharp-masking edge enhancement filter before generating the halftoning picture. This filtering generates an image with enhanced details (I_{eh}). Then,

Fig. 1 Block diagram of the halftoning algorithm used to obtain the ordered dithering pattern

we quantize I_{eh} with eight distinct intervals, using the following equation:

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

where x and y are the horizontal and vertical spatial coordinates, respectively, c refers to the color channel ($1 \leq c \leq 3$), and I_Q is the resulting dithered image. To enable restoration, up to 3 bits can be embedded in each color channel without causing visible degradations [34]. This means that the dithered mark (I_Q) is represented with a total of 9 bits per pixel. Then, we substitute each value of $I_Q(x,y,c)$ by the corresponding 3-bit combinatorial dispersed dot patterns.

The human visual system (HVS) is less sensitive to degradations in the blue channel [35]. Taking that into account, we use 2 bits to represent the dithered version of the blue channel and 3 bits to represent the dithered versions of the red and green channels. In other words, for the red and green channels, we substitute each 8-bit pixel of I_Q by one of the 3-bit dot patterns shown in Fig. 2b. For the blue channel, we substitute each 8-bit pixel of I_Q by one of the 2-bit dot patterns shown in Fig. 2a.

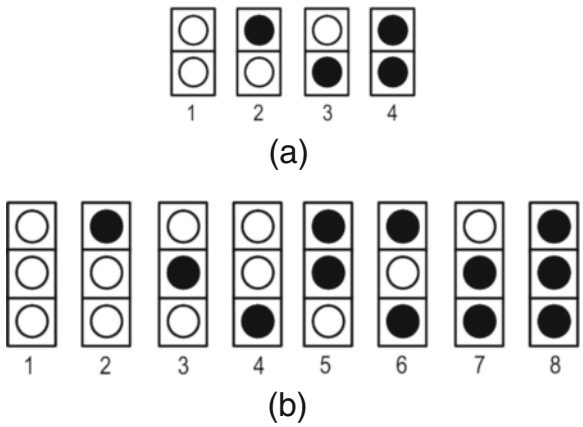

Fig. 2 Combinatorial dispersed dot patterns: **a** 2-bit dispersed dot patterns used for mapping four intensity levels and **b** 3-bit dispersed dot patterns used for mapping eight intensity levels

Watermarking embedding

After generating the mark using the techniques described in the previous section, the next stage consists of embedding the mark into the host picture. 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 distribute the mark using a *split-flip* operation. This operation consists of splitting the halftone image into sub-blocks and flipping them to different spatial regions. More specifically, we divide the picture into $k \times k$ sub-blocks, rotate each sub-block by 180° and shuffle the regions. Figure 3 shows an example of the process for the three-color channels of the image Lena using 32×32 blocks.

Fig. 3 Original dithered versions of each color channel of the image Lena (*left*) and the corresponding split-flip versions (*right*)

For videos signals, we also perform an embedding temporal distribution by inserting the mark corresponding to the current picture frame in a previous picture frame, located 1 s before. By distributing the mark spatially and temporally, a 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. The mark is also encrypted applying a XOR cipher to protect it from being extracted by an unauthorized user.

To store the watermark, we use a QIM-based approach. Among the available watermarking methods, QIM is one of the methods with the best performance [12]. The QIM algorithm inserts a mark into a host signal by quantizing it with a uniform scalar quantizer. The standard quantization operation with step size δ is given by the following equation:

$$Q(x,\delta) = \text{round}\left(\frac{x}{\delta}\right),$$

where round(.) denotes the mathematical operation of rounding a value to the nearest integer. The watermarked pixel is obtained using the following equation:

$$s(x) = Q(x,\delta) + d(m),$$

where $d(m)$ is the perturbation value, which depends on the mark signal m to be embedded.

In this work, we propose a modification of the QIM algorithm that inserts an integer mark instead of a function of the 1-bit mark. The modulation function is set to $d(m) = m$. So, the integer dithered image (I_{dth}) is inserted in each pixel of the corresponding color channel of the original picture using the following equation:

$$I_{\text{m}}(x,y,c) = Q(I_o(x,y,c),\delta) + I_{\text{dth}}(x,y,c),$$

where I_{o} is the original color channel of the picture frame, I_m is the resulting watermarked color channel, δ is the quantization step, and c is the corresponding color channel.

Watermarking extraction

The watermarking extraction is performed at the receiver side (decoder) after a possible transmission or tampering. When a lost or tampered region is detected by the decoder, the mark is extracted using the following equation:

$$\hat{I}_{\text{dth}}(x,y,c) = I_{\text{m}}(x,y,c) \bmod \delta,$$

where I_{m} is the watermarked color channel and $\hat{I}_{\text{dth}}$ is the recovered mark. Then, an authorized user can decipher the encrypted mark using the symmetric key provided by the owner of the content.

To detect if a picture region is lost or tampered, the extracted watermark is compared with the corresponding "host" content. For tampering detection, we extract the watermark and compute its inverse halftoning version. By computing the structural similarity (SSIM) between blocks of inverse halftoning and host content, it is expected that SSIM is higher for blocks where there are no tampers. On the other hand, if the image is tampered, the structural similarity between host and restored blocks are lower. For error concealment, the position of the lost areas are identified by the decoder and the damaged areas are substituted by the recovered watermark after an inverse halftoning process.

Inverse halftoning

If a region is classified as lost or tampered, we search the correspondent mark in the appropriate location as depicted in the "Watermarking embedding" section. To generate a multi-level colored picture, we use an inverse halftoning algorithm [33]. The block diagram of the inverse halftoning algorithm is shown in Fig. 4.

Given that I_{th} is the dithered 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 dithered 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 to the indices of the dot patterns. Once this interval is found, we randomly select a value within it, generating a slightly noisy picture I_{inv}.

Next, I_{inv} is filtered using a Gaussian low-pass and a Laplacian-of-Gaussian, generating I_{gauss} and $I_{\log}$,

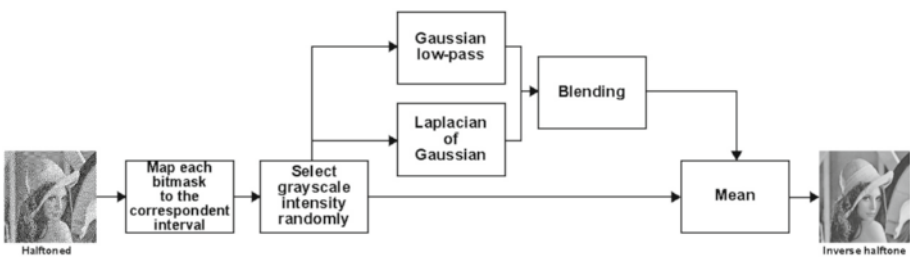

Fig. 4 Block diagram of the inverse halftoning algorithm

respectively. The resulting pictures are used to compose another picture, I_{bld}, given by the following equation:

$$I_{\text{bld}}(x,y,c) = \gamma \cdot I_{\text{gauss}}(x,y,c) + (1-\gamma) \cdot I_{\log}(x,y,c),$$

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

When we combine all three channels, the resulting picture contains visible color distortions. If, on the other hand, we use γ as a constant matrix, I_{bld} is a blurred version of I_{inv}. To minimize color distortion and keep the details of the original image, we make a composition of I_{inv} and I_{bld}, using the following equation:

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

where $\hat{I}_{\text{o}}$ is the recovered 8-bit version of original video frame, I_{o}. The inverse halftoning process can be applied to videos and images alike. In the next sections, we detail the application of the proposed system for error concealment and tamper detection.

Error concealment algorithm

We used the techniques described in the previous sections to design an error concealment algorithm. The algorithm is designed to be integrated into a compression codec but can also be used independently of the codec technology. Since decoders are able to identify which packets were lost, there is no need for a key code. To increase the quality of restored areas, for each color channel, we used dithered watermarks of 3 bits (see the "Methods" section, Fig. 2b).

The block diagram of this algorithm is depicted in Fig. 5. The idea here is to show that the proposed algorithm is intended to be implemented together with the codec (or any other compression algorithm). For images, the buffer is not necessary because the image stores the watermark in itself using only the split-flip operation (see the "Watermarking embedding" section).

Tampering detection algorithm

We use the techniques described in previous sections to design a tampering detection algorithm that is able to detect tampered regions and recover the original content. For that, we use 1 bit (out of the 8 bits) to store a secret key code (see the "Watermarking embedding" section). As described earlier, the dithered versions of the red and green channel use 3 bits each, while the dithered version of the blue channel uses only 2 bits. Figure 6 shows the block diagram of the tampering detection algorithm.

Results and discussion

In this section, we present the simulations of the error concealment algorithm and the tampering detection algorithm, for videos and images.

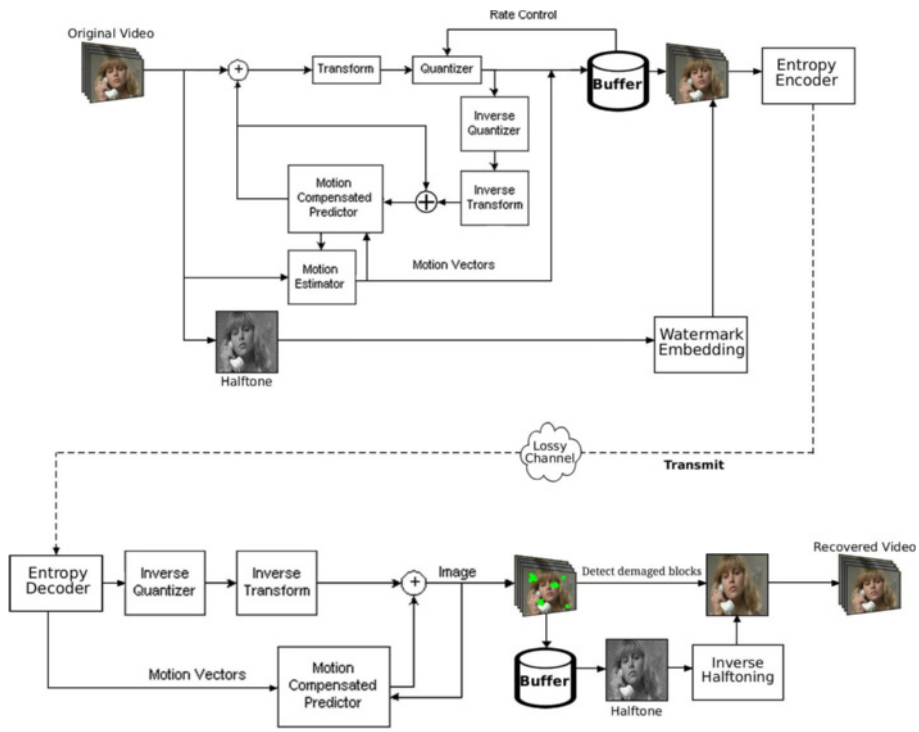

Fig. 5 Block diagram of error concealment algorithm for videos

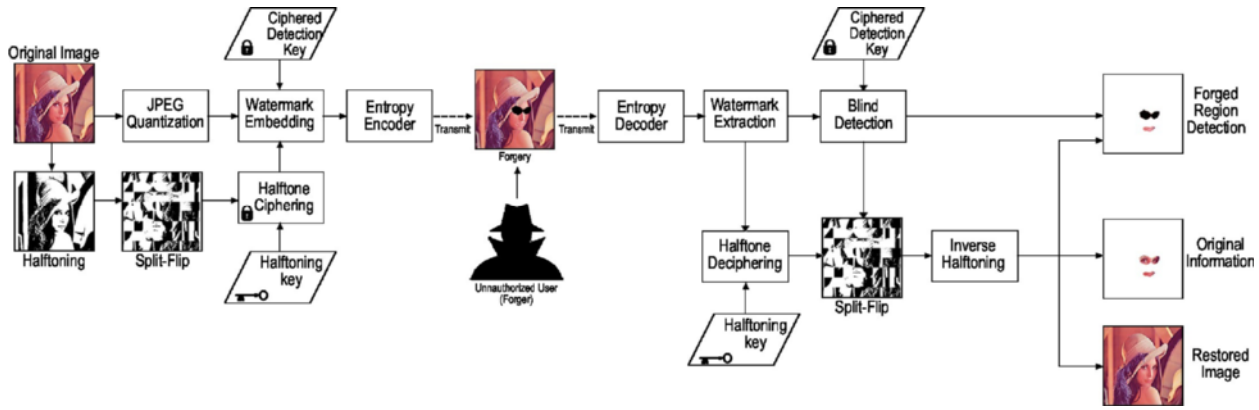

Fig. 6 Block diagram of tampering detection algorithm

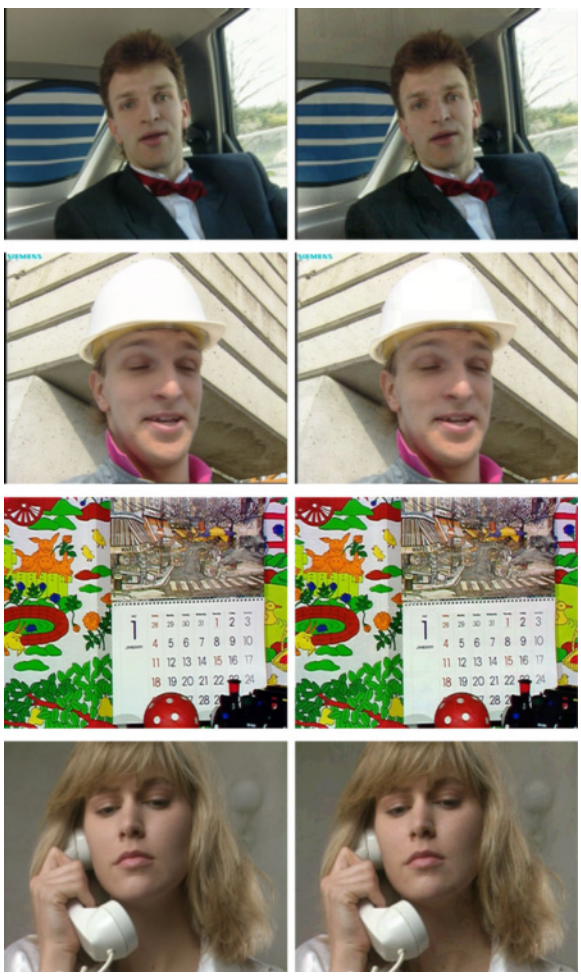

Fig. 7 Comparison of the original (*left*) and marked (*right*) images. From top to down: "Carphone," "Foreman," "Mobile," and "Suzie"

Quality of watermarking algorithm

In Fig. 7, we show a comparison between original and watermarked picture frame for the videos "Carphone," "Foreman," "Mobile," and "Suzie," which are publicly available . The format of these videos are YUV 4:4:4 color CIF (352 × 288, progressive), with each video containing around 300 frames each. From this figure, we can observe that, due to the quantization, the addition of the mark introduces very small distortions, which are hardly visible.

Table 1 shows the peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [36], and universal image quality index (UQI) [37] values 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 perceptually acceptable.

Error concealment for images

First, we test our algorithm using a set of still images degraded by deleting a percentage of the image blocks of size 16 × 16 pixels. To implement this test, we divide the image in blocks of 16 × 16 and randomly discard a percentage of these blocks (the original block content is replaced by zeros). The percentage of deleted blocks varies from 5 to 25 %. Figure 8 shows three examples of damaged images restored using the proposed error concealment algorithm. The images on the left column are the originals, the images on the middle columns have 20 % of blocks

Table 1 UQI, PSNR and SSIM values calculated between original and watermarked videos

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

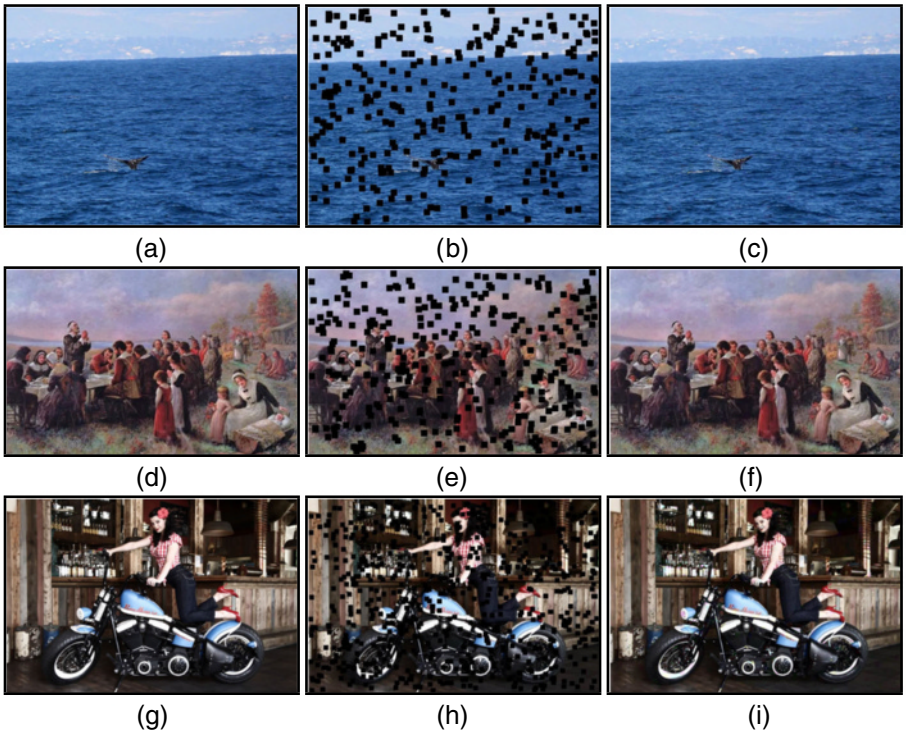

Fig. 8 Illustration of restoration of lost blocks using the proposed error concealment algorithm: **a**, **d**, **g** are the original pictures; **b**, **e**, **h** are the pictures with 20 % of content lost; **c**, **f**, **i** are the restored pictures using the proposed algorithm

discarded, and the images on the right column are the corresponding restored versions. Notice that the quality of the reconstructed image is excellent, and it is very difficult to identify the location of the restored blocks.

We also calculate the peak signal-to-noise ratio (PSNR) values between the original and restored images. Figure 9 depicts the graph of the average PSNR over all videos versus the percentage of deleted blocks. We compared the proposed algorithm with other state-of-the-art methods, such as visual clearness (VC) [38], weighted template matching (WTM) [39], multivariate kernel density estimation (KMMSE) [40], and frequency selective extrapolation with residual filtering (XFSE) [41]. It can be observed that, as expected, the PSNR values decrease as the percentage of deleted blocks increases. When compared to the other methods, the proposed method has a much better performance. These good results are due to the additional redundancy inserted by the proposed watermarking algorithm. Also, the improved inverse halftone method is able to restore textures and borders information with a good quality.

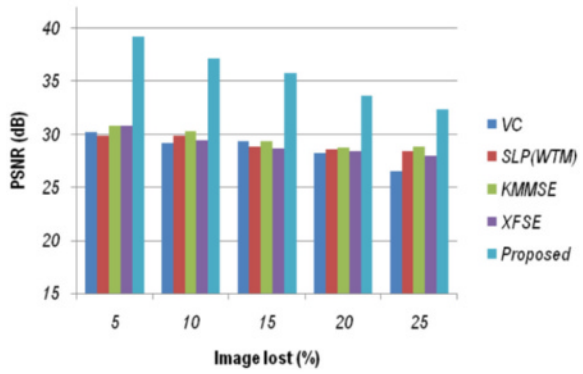

Fig. 9 Average PSNR of reconstructed images versus percentage of deleted blocks

Figure 10 shows examples of the visual results obtained by the proposed method and the KMMSE, WTM, VC, and XFSE methods. The first image on the top is the original image. The images in the first column show the results after restoration, while the images in the second column show the maps of removed blocks. From these images, we can notice that most error concealment techniques insert artifacts in restored images. The KMMSE and WTM methods (second and third row) insert blocking artifacts and visible color distortions. The image restored with the VC method (fourth row) presents the lowest quality among the tested methods, showing visible dark blocks in the restored areas. The XFSE method (fifth row) shows small color distortions and a small blocking effect. Finally,

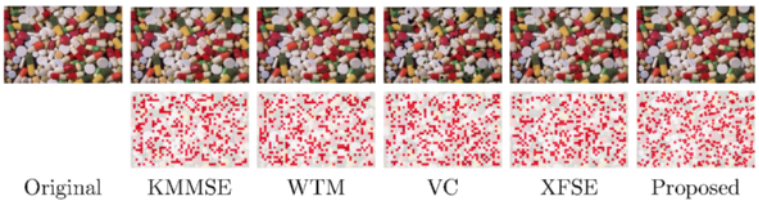

Fig. 10 Concealed results using different methods and their respective map of attacks for "pill" image

the last row of Fig. 10 shows the result of the proposed method. It presents the best results among all tested methods, showing only minor noisy artifacts in the restored areas.

Error concealment for videos

Second, we test the ability of the proposed algorithm to recover lost packets. The block diagram of this error concealment algorithm for videos is depicted in Fig. 5. We use publicly available videos in YUV 4:4:4 color CIF (352×288, progressive) format, with around 300 frames each. The videos are "Foreman," "Mobile," "Carphone," and "Suzie" [42]. We embed the proposed error concealment algorithm into a video stream. In order to simulate packet losses into a given bitstream, we use a simple model that simulates packet losses over error-prone channels. The following packet loss rates (PLR) are used: 0.5, 1, 3, 5, and 10 %.

Although the error concealment methods tested in the previous section (KMMSE, WTM, VC, and XFSE) perform well for images, they are not designed to conceal errors in compressed videos. They are designed to conceal errors only within homogeneous spatial regions and, therefore, cannot recover lost packets because these packets may contain information from several frame blocks distributed along several frames. Given the inability to compare the proposed method with these techniques, we compare it only with the standard H.264 error concealment algorithm.

Figure 11 depicts an example of the use of the proposed algorithm to mitigate lost packets for the videos "Foreman" and "Mobile." The first row of the figure shows sample frames of the original videos. The second row shows sample frames of the videos recovered with the *default* H.264 error concealment algorithm [1] for 3 % PLR. The third row shows sample frames of the videos restored using the proposed algorithm, also for 3 % PLR. As can be noticed, the proposed method is able to get rid of the most visible distortions present on videos restored using the standard H.264 error concealment algorithm, like for example blocking effects, packet losses, and false contours. In fact, we observed that, for PLR values below 5 %, the restored videos present very few distortions.

Figure 12a, b depicts the graphs of SSIM and PSNR values, respectively, for the several PLR values and for all videos. In these graphs, the continuous lines correspond to the proposed method, while the dashed lines correspond to the default H.264 error concealment algorithm. It can be observed that, as expected, PSNR and SSIM values decreases with PLR. The proposed algorithm has a better performance than the default H.264 error concealment algorithm, both in terms of PSNR and SSIM.

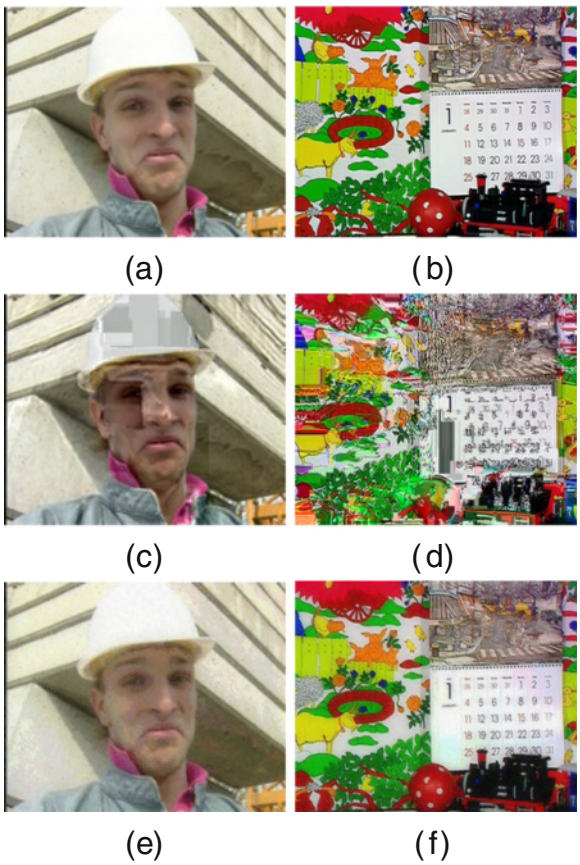

Fig. 11 From top to bottom: **a**, **b** original frames, **c**, **d** restored using the default H.264 error concealment algorithm, and **e**, **f** restored using the proposed algorithm

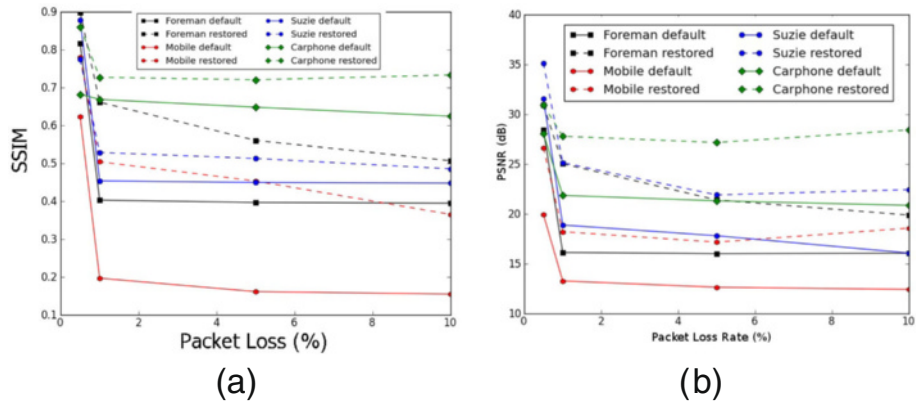

Fig. 12 a SSIM and **b** PSNR for reconstructed videos versus packet loss rate values. The label "default" refers to the standard H.264 error concealment algorithm, while "restored" refers to the proposed error concealment algorithm

Tampering detection in images

We test the proposed algorithm using still images with different characteristics: high-detailed, low-detailed, color, grayscale, documents, landscape, faces and people images, etc. Different kinds of attacks are applied to these images: blurring of selected/random areas, noise addition, cut-and-paste, region deletion, and resizing.

Figure 13 depicts examples of the use of the proposed algorithm to detect tampered regions in still images. Each column (from left to right) shows original (non-tampered) images, tampered images, regions detected as tampered, recovered tampered regions, images recovered without the secret key, and images reconstructed with the restored original content. As can be observed, the algorithm is able to detect tampered regions, independent of their size. Also, since the embedded halftone watermark is encrypted, an unauthorized user is unable to know if an image was tampered and, consequently, unable to reconstruct the image back to its original state (see Fig. 13e). The algorithm is able to restore content of tampered images with good quality, if at least 50 % of the regions of the embedded halftone are preserved (see Fig. 13f).

Tampering detection in videos

We also test the performance of the algorithm for tampering detection and recovery of digital videos. As in the previous examples, we used publicly available videos in YUV 4:4:4 color CIF (352 × 288, progressive) format with around 300 frames each, downloaded from the Video Trace Library [42] and from the Consumer Digital Video Library [43]. We tested the following attacks: blurring of selected areas, cut-and-paste, region-deletion, and object-addition. Again, for all test cases, we were able to detect tampered regions in 100 % of the cases. In

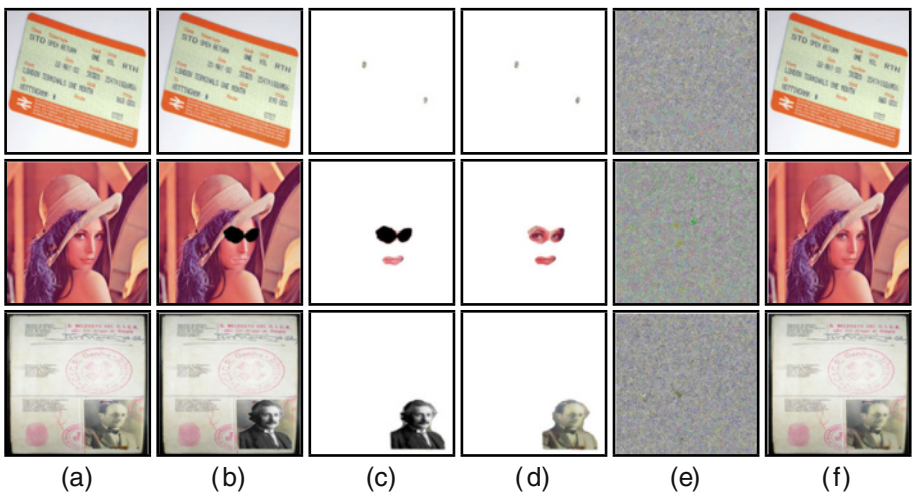

Fig. 13 Identification and restoration of tampered regions in the "Train ticket," "Lena," and "Passport" images attacked using "cut-and-paste," "crop and blur," and "cropping" attacks, respectively. From left to right: **a** original, **b** tampered, **c** regions identified as tampered, **d** restored regions, **e** unauthorized recovered mark, **f** restored image

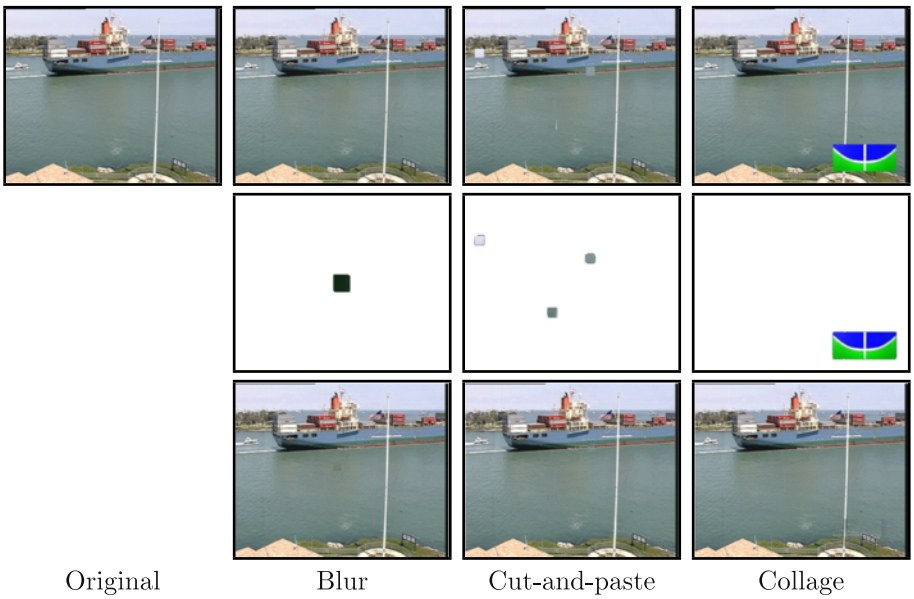

Fig. 14 Identification and restoration of tampered regions for sample frames of video "container." From top to bottom: tampered frames, identified tampers, and restored frames

terms of reconstruction, the algorithm was able to recover tampered regions with good quality.

Figure 14 depicts examples of the use of the proposed algorithm to detect tampered regions for three different types of attacks. Figure 14a shows the original frame content. Figure 14b, e, h shows the frames attacked using blurring, cut-and-paste, and object-addition, respectively. In addiction, Fig. 14c, f, i shows the tampered areas using these attacks, while the Fig. 14d, g, j shows the frames with the original content restored. Notice that the algorithm is able to detect tampered regions, independent of their size, position or (visual) similarity to the original content. Also, the restored content has a very good quality.

In Table 2, we compare the spatial and temporal efficiency of the proposed technique with seven methods available in the literature [44–50]. In this table, the efficiency is computed as the ratio between the amount of tampered frames and the amount of detected tampered frames. The table shows the percentage of the cases in which an attacked frame was detected as being tampered, independent of the size of the attacked areas.

Table 2 Mean efficiency of spatial detections per frame for some methods (%)

Method	Spatial attack	Temporal attack
Zhi-yu and Xiang [44]	50.49	–
Hsu et al. [45]	98.34	–
Lin et al. [46]	82.05	–
Pan and Lyu [47]	100	–
Subramanyam and Emmanuel [48]	85.01	99.5
Amerini et al. [49]	98.17	–
Wang and Farid [50]	–	100
Proposed Framework	97.86	96.53

Conclusions

This paper presents a secure variable-capacity self-recovery watermarking scheme. In the proposed scheme, it is possible to implement both an error concealment algorithm and a tampering detection algorithm. The scheme is based on watermarking and halftoning techniques. In order to increase the data hiding capacity, this work proposed a simple modification of the QIM watermarking algorithm. To obtain higher quality restored areas, improved inverse halftoning algorithms are also proposed. A secret key code is embedded to the host content to identify the spatial and temporal positions of tampered regions, taking advantage of the lower sensitivity of the HVS to degradations in the blue color channel. Above all, the proposed scheme not only achieves variable-capacity, higher security, higher detection accuracy, and strong recovery ability but also can resist collage attack and mean attack.

Future works include a further increase of the data hiding capacity with the goal of embedding even more information. With that, the quality of the restored content can be increased and additional bits can be used for protection of the data against tampering. For example, using some bits to embed additional temporal information can help counter other attacks, such as frame shuffle.

Acknowledgements
This work was supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), and by the University of Brasília (UnB).

Authors' contributions
RR developed the methods presented in this paper for his master degree dissertation at the University of Brasília (UnB), Brazil. PGF wrote most of the text, figures, and diagrams presented in this manuscript. MCQF is the research adviser and was responsible to guide the research, the writing, and the revising of this manuscript. All authors read and approved the final version of this work.

Competing interests
The authors declare that they have no competing interests.

Received: 21 April 2016 Accepted: 8 September 2016
Published online: 22 September 2016

References

1. Yang SH, Tsai JC (2010) A fast and efficient H. 264 error concealment technique based on coding modes. In: 2010 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB). IEEE. pp 1–4. http://ieeexplore.ieee.org/document/5463091/
2. Wang Y, Zhu QF (1998) Error control and concealment for video communication: a review. Proc IEEE 86(5):974–997
3. Nasiopoulos P, Coria-Mendoza L, Mansour H, Golikeri A (2005) An improved error concealment algorithm for intra-frames in h. 264/avc. In: Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium On. IEEE. pp 320–323. http://ieeexplore.ieee.org/document/1464589/
4. Cui Z, Gan Z, Zhan X, Zhu X (2012) Error concealment techniques for video transmission over error-prone channels: a survey. J Comput Inf Syst 8(21):8807–8818
5. Al-Mualla ME, Canagarajah CN, Bull DR (2001) Multiple-reference temporal error concealment. In: Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium On, vol. 5. IEEE. pp 149–152. http://ieeexplore.ieee.org/document/922007/
6. Zhou J, Yan B, Gharavi H (2011) Efficient motion vector interpolation for error concealment of h.264/avc. IEEE Trans Broadcast 57(1):75–80
7. Sun H, Liu P, Wang J, Goto S (2011) An efficient frame loss error concealment scheme based on tentative projection for H.264/AVC. Springer. http://link.springer.com/chapter/10.1007/978-3-642-15696-0_37
8. Lin TL, Yang NC, Syu RH, Liao CC, Tsai WL (2013) Error concealment algorithm for hevc coded video using block partition decisions. In: Signal Processing, Communication and Computing (ICSPCC), 2013 IEEE International Conference On. pp 1–5. doi:10.1109/ICSPCC.2013.6664106
9. Ranjan A, Midya A, Chakraborty J, Sengupta S (2014) Video error concealment using speeded up robust features and affine transformation. In: Students' Technology Symposium (TechSym), 2014 IEEE. pp 72–75. doi:10.1109/TechSym.2014.6807917
10. Koloda J, Ostergaard J, Jensen SH, Sanchez V, Peinado AM (2013) Sequential error concealment for video/images by sparse linear prediction. IEEE Trans Multimed 15(4):957–969. doi:10.1109/TMM.2013.2238524
11. Costa MHM (1983) Writing on dirty paper (corresp.) IEEE Trans Inf Theory 29(3):439–441. doi:10.1109/TIT.1983.1056659
12. Chen B, Wornell GW (2001) Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Trans Inf Theory 47(4):1423–1443
13. Cox I, Miller M, Bloom J, Fridrich J, Kalker T (2008) Digital watermarking and steganography 2nd edn.. Morgan Kaufmann Publishers Inc., San Francisco
14. Yin P, Liu B, Yu HH (2001) Error concealment using data hiding. In: Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference On. vol. 3. pp 1453–14563. doi:10.1109/ICASSP.2001.941204
15. Chung KL, Huang YH, Chang PC, Liao H-YM (2010) Reversible data hiding-based approach for intra-frame error concealment in h.264/avc. IEEE Trans Circ Syst Video Technol 20(11):1643–1647. doi:10.1109/TCSVT.2010.2077577
16. Li S, Karrenbauer A, Saupe D, Kuo C-CJ (2011) Recovering missing coefficients in dct-transformed images. In: 2011 18th IEEE International Conference on Image Processing, IEEE. pp 1537–1540. http://ieeexplore.ieee.org/document/6115738/
17. Xu D, Wang R, Shi YQ (2014) An improved reversible data hiding-based approach for intra-frame error concealment in h. 264/avc. J Vis Commun Image Represent 25(2):410–422
18. Wang H, Ho AT, Li S (2014) A novel image restoration scheme based on structured side information and its application to image watermarking. Signal Process Image Commun 29(7):773–787
19. Adsumilli CB, Farias MCQ, Mitra SK, Carli M (2005) A robust error concealment technique using data hiding for image and video transmission over lossy channels. IEEE Trans Circ Syst Video Technol 15(11):1394–1406
20. Nayak CK, Surendran J, Merchant SN, Desai UB, Sanyal S (2010) Error concealment of h.264 encoded video through a hybrid scheme. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, MEDES '10. ACM, New York. pp 189–195
21. Redi JA, Taktak W, Dugelay JL (2011) Digital image forensics: a booklet for beginners. Multimedia Tools Appl 51(1):133–162
22. Ng TT, Chang SF, Sun Q (2004) Blind detection of photomontage using higher order statistics. In: Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on. IEEE Vol. 5. pp V-688–V-691. http://ieeexplore.ieee.org/document/1329901/
23. Peng F, Wang X-l (2010) Digital image forgery forensics by using blur estimation and abnormal hue detection. In: 2010 Symposium on Photonics and Optoelectronics. IEEE. pp 1–4. http://ieeexplore.ieee.org/document/5504476/
24. Imaizumi S, Taniguchi K (2014) Hierarchical image authentication based on reversible data hiding. Bull Soc Photogr Imag Japan 24(1):1–5
25. Xu D, Wang R, Shi YQ (2014) Data hiding in encrypted h. 264/avc video streams by codeword substitution. IEEE Trans Inf Forensic Secur 9(3-4):596–606
26. Phadikar A, Maity SP, Mandal M (2012) Novel wavelet-based qim data hiding technique for tamper detection and correction of digital images. J Vis Commun Image Represent 23(3):454–466
27. Tong X, Liu Y, Zhang M, Chen Y (2013) A novel chaos-based fragile watermarking for image tampering detection and self-recovery. Signal Process Image Commun 28(3):301–308
28. Lin SJ, Lin JC (2010) Authentication and recovery of an image by sharing and lattice-embedding. J Electron Imaging 19(4):043008–043008
29. Dadkhah S, Manaf AA, Hori Y, Hassanien AE, Sadeghi S (2014) An effective svd-based image tampering detection and self-recovery using active watermarking. Signal Process Image Commun 29(10):1197–1210
30. Som S, Palit S, Dey K, Sarkar D, Sarkar J, Sarkar K (2015) Applied Computation and Security Systems: Volume Two. In: Chaki R, Saeed K, Choudhury S, Chaki N (eds). Springer, New Delhi. pp 17–37
31. Soleimany H, Sharifi A, Aref M (2010) Improved related-key boomerang cryptanalysis of AES-256. In: 2010 International Conference on Information Science and Applications. IEEE. pp 1–7. http://ieeexplore.ieee.org/document/5480302/
32. Knuth DE (1987) Digital halftones by dot diffusion. ACM Trans Graph 6:245–273
33. Freitas PG, Farias MCQ, de Araujo APF (2011) Fast inverse halftoning algorithm for ordered dithered images. In: 2011 24th SIBGRAPI Conference on Graphics, Patterns and Images. IEEE. pp 250–257. http://ieeexplore.ieee.org/document/6134739/
34. Rigoni R, Freitas PG, Farias MC (2016) Detecting tampering in audio-visual content using qim watermarking. Inf Sci 328:127–143
35. Mullen KT (1985) The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings. J Physiol 359:381
36. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612. doi:10.1109/TIP.2003.819861
37. Wang Z, Bovik AC (2002) A universal image quality index. IEEE Signal Proc Lett 9(3):81–84. doi:10.1109/97.995823
38. Koloda J, Sánchez V, Peinado AM (2013) Spatial error concealment based on edge visual clearness for image/video communication. Circ Syst Signal Process 32(2):815–824
39. Koloda J, Ostergaard J, Jensen SH, Peinado AM, Sanchez V (2012) Sequential error concealment for video/images by weighted template matching. In: Data Compression Conference (DCC), 2012. pp 159–168. doi:10.1109/DCC.2012.24

40. Koloda J, Peinado AM, Sanchez V (2013) On the application of multivariate kernel density estimation to image error concealment. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference On. pp 1330–1334. doi:10.1109/ICASSP.2013.6637867
41. Koloda J, Seiler J, Kaup A, Sanchez V, Peinado AM (2014) Frequency selective extrapolation with residual filtering for image error concealment. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On. pp 1976–1980. doi:10.1109/ICASSP.2014.6853944
42. Video Trace Library of Arizona State University (ASU). http://trace.eas.asu.edu/. Accessed Apr 2013
43. The Consumer Digital Video Library (CVDL). http://www.cdvl.org/. Accessed Apr 2013
44. Zhi-yu H, Xiang-hong T (2011) Integrity authentication scheme of color video based on the fragile watermarking. In: Electronics, Communications and Control (ICECC), 2011 International Conference On. pp 4354–4358. doi:10.1109/ICECC.2011.6067709
45. Hsu CC, Hung TY, Lin CW, Hsu CT (2008) Video forgery detection using correlation of noise residue. In: Multimedia Signal Processing, 2008 IEEE 10th Workshop On, IEEE. pp 170–174. http://ieeexplore.ieee.org/document/4665069/
46. Lin E, Eskicioglu AM, Lagendijk RL, Delp EJ (2005) Advances in digital video content protection. Proc IEEE 93(1):171–183
47. Pan X, Lyu S (2010) Region duplication detection using image feature matching. IEEE Trans Inf Forensic Secur 5(4):857–867
48. Subramanyam A, Emmanuel S (2012) Video forgery detection using hog features and compression properties. In: Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop On, IEEE. pp 89–94. http://ieeexplore.ieee.org/document/6343421/
49. Amerini I, Ballan L, Caldelli R, Del Bimbo A, Serra G (2011) A sift-based forensic method for copy–move attack detection and transformation recovery. IEEE Trans Inf Forensic Secur 6(3):1099–1110
50. Wang W, Farid H (2007) Exposing digital forgeries in video by detecting duplication. In: Proceedings of the 9th Workshop on Multimedia & Security, ACM. pp 35–42. http://dl.acm.org/citation.cfm?id=1288876