Available online at www.sciencedirect.com
ScienceDirect
Signal Processing: *Image Communication* 21 (2006) 890–903
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Joint near-lossless compression and watermarking of still images for authentication and tamper localization

Roberto Caldelli[a,*], Francesco Filippini[a], Mauro Barni[b]

[a]*Department of Electronics and Telecommunications, University of Florence, Via di Santa Marta, 3, 50139 Florence, Italy*
[b]*Department of Information Engineering, University of Siena, Siena, Italy*

Received 3 August 2005; received in revised form 28 August 2006; accepted 30 August 2006

Abstract

A system is presented to jointly achieve image watermarking and compression. The watermark is a fragile one being intended for authentication purposes. The watermarked and compressed images are fully compliant with the JPEG-LS standard, the only price to pay being a slight reduction of compression efficiency and an additional distortion that can be anyway tuned to grant a maximum preset error. Watermark detection is possible both in the compressed and in the pixel domain, thus increasing the flexibility and usability of the system. The system is expressly designed to be used in remote sensing and telemedicine applications, hence we designed it in such a way that the maximum compression and watermarking error can be strictly controlled (near-lossless compression and watermarking). Experimental results show the ability of the system to detect tampering and to limit the peak error between the original and the processed images.

Keywords: Image authentication; Near-lossless JPEG; Digital watermarking; Tamper localization; Remote sensing; Medical imagery

1. Introduction

The demand for image authentication and for effective means to control image integrity has been steadily increasing in the last years. Such a demand is due to the ease with which digital images can be tampered with thus compromising their credibility as faithful pictures of the scene they represent. Several techniques have been developed in order to prevent or at least detect unwanted alteration of digital images. Among them, digital watermarking has gained more and more popularity due to its versatility and its potential to localize tampering and the possibility (at least theoretical) to distinguish between different kinds of manipulations (usually split into allowed and not allowed manipulations). Two possible approaches can be distinguished, one based on (semi) fragile watermarking and the other relying on robust watermarking. Authentication through fragile watermarking [15,5] is accomplished by inserting within the image a watermark that is readily altered or destroyed as soon as the host image undergoes any manipulations. The alteration or deletion of the watermark allows to discover that the image has been modified, whereas the correct recovery of the hidden information permits to prove the integrity of the image and, possibly, to establish its origin. Some techniques

*Corresponding author. Tel.: +39 0554796380; fax: +39 055494569.
E-mail address: caldelli@lci.det.unifi.it (R. Caldelli).

0923-5965/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2006.08.006

permit also to localize the altered zones on a block basis [8,6]. Systems based on robust watermarking [4,11] assume that the watermark is not affected by image manipulations. Specifically, a summary of the to-be-authenticated image is computed and embedded within the image itself (possibly together with additional information about the origin of the image). Subsequently, the hidden information is recovered and compared with the actual content of the image: a mismatch reveals that the image has been tampered with.

In this paper we focus on authentication and tamper localization through fragile watermarking. Specifically, our system is built by relying on a scheme originally developed by Wong [14] and successively improved by Fridrich [5] with a better logo structure to prevent attacks. This method, that in the sequel will be called, for sake of simplicity, Fridrich's method, embeds the watermark in the least significant bits (LSB) of the host image. The choice of Fridrich's algorithm is justified by its security features and its good localization capabilities (more details on this scheme are given in Section 2).

Together with the demand for integrity verification, the demand for image compression is everyday more pressing. The great majority of the images exchanged in digital format are stored in a compressed format, with lossy compression being definitely much more popular than lossless compression. Hence, a first crucial choice must be made to decide whether to embed the watermark in the raw domain (i.e. before compression takes place) or in the compressed domain (e.g. by jointly coding and watermarking the image). In the context of image authentication through fragile watermarking, joint coding and watermarking is highly desirable, since otherwise the fragile nature of the watermark will identify image compression as an unwanted manipulation hence failing to distinguish between (allowed) compression and (not allowed) tampering. On the other hand, tying the watermarking system to a particular coding format limits the flexibility of the authentication scheme, since the watermark is likely not to survive lossless format changes, e.g. conversions from the coded and the raw format. It is one of the goals of the system presented in this paper to embed the watermark in the compressed domain, while still allowing its recovery in the raw pixel domain.

Though lossy compression is by far the most popular coding strategy used today, in some application scenarios the loss of information accompanying the compression process cannot be tolerated or, at least, must be strictly controlled. This is the case of remote sensing and medical applications. In both cases the risk of discarding useful information calls for the adoption of lossless compression, however the large amount of data acquired by sensors during earth observation missions and the large volume of images produced by modern telemedicine applications [10,3] make the use of efficient lossy coding algorithms unavoidable. In order to control the amount of information lost during the compression process, a class of algorithms capable of strictly controlling the compression loss have been devised and grouped under the term near-lossless compression, whose main requirement is that of insuring that the maximum error between the original and the compressed image does not exceed a fixed threshold. In the same line, the concept of near-lossless watermarking has been introduced recently to satisfy the strict requirements set by the remote sensing scenario [2,1]. In this paper we propose a system that permits to jointly compress and watermark the to-be-protected image in a near-lossless fashion; thus resulting particularly suited for remote sensing and medical applications.

Specifically, Fridrich's authentication algorithm [5] is modified so as to make it compliant with the JPEG-LS coding standard. The JPEG-LS [9,13] is a lossless/near-lossless image coding scheme based on differential pulse code modulation (DPCM) [12]. In the near-lossless mode each pixel of the reconstructed image differs from the corresponding original pixel by up to a preset (usually small) amount, called Δ in the following. By slightly modifying the quantization process, our system is able to embed an LSB message similar to that used by Fridrich directly in the compressed domain, thus keeping complete compliance with the JPEG-LS. At the same time, the maximum amount of distortion introduced by the watermark can be strictly controlled thus satisfying the near-lossless requirement. As already said, the watermark can be recovered both in the compressed and in the raw domain, thus increasing the flexibility of the system and its practical usability. Finally, the security features of Fridrich's algorithm are retained together with its localizing properties (the localization accuracy being reduced only slightly).

The rest of the paper is organized as follows. In Section 2, after a brief review of the JPEG-LS

standard, the proposed watermark embedding algorithm is described. In Section 3, watermark detection is considered. In Section 4, security issues are discussed. Section 5 is devoted to the presentation of experimental results. Finally, some conclusions are drawn in Section 6.

2. Encoding phase

As we already said, the main goal of the new watermarking scheme is to grant robustness against near-lossless JPEG image compression, while maintaining the usual features of an authentication technique. This aim is achieved by designing a system which is based on the JPEG-LS coding standard. In order to generate compressed data and simultaneously authenticate them, a secure fragile watermarking technique, that in our approach has been individuated in the technique developed by Fridrich [5], has been integrated within the JPEG-LS.

2.1. JPEG-LS in brief

Before describing the proposed watermarking algorithm let us sketch the JPEG-LS standard. JPEG-LS is a typical example of coder based on spatial pixel prediction followed by quantization of the prediction error and entropy coding (specifically Golomb–Rice coding is used). The aim of the spatial prediction is to decorrelate the pixel values providing an approximately white sequence of prediction errors. Prediction is performed according to a causal neighborhood (the image is scanned left to right and top to bottom) as depicted in Fig. 1, where the brightness Ix of the current pixel is predicted by relying on the pixels in position Ra, Rb and Rc (the value of Rd is used only for context modelling, see below). In the sequel we will indicate the predicted value of the pixel in position x by Px, the prediction error by E and the quantized prediction error by Q_E. Pixel values are reconstructed by adding back the dequantized prediction error Q_R to the predicted value Px. Note that due to quantization the reconstructed pixel value

$$Rx = Px + Q_R \tag{1}$$

is different form the original value Ix, hence in order to keep the decoder and encoder synchronized, the encoder calculates the predicted value Rx by relying on the predicted values of the pixels in the causal neighborhood.

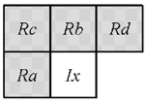

Fig. 1. A causal context for JPEG-LS prediction.

According to JPEG-LS terminology, prediction is formulated as an inductive inference problem sometimes referred to as *modelling*. In particular, the modelling approach JPEG-LS relies on is based on the notion of context, which is determined by the four reconstructed samples Ra, Rb, Rc, Rd, belonging to a neighborhood of the current sample Ix (see Fig. 1).

Consequently, each sample value is conditioned to the context and also the probability distribution used to encode the samples is determined by the context. Though two different encoding modes are available in JPEG-LS, the watermarking method proposed here considers only the so-called *regular mode*. In the *regular mode*, the prediction procedure works as follows. First of all, a test for the presence of a horizontal or vertical edge is performed on the pixels belonging to the context, then the predicted value Px is computed according to

$$Px = \begin{cases} \min(Ra, Rb) & \text{if } Rc \geqslant \max(Ra, Rb), \\ \max(Ra, Rb) & \text{if } Rc \leqslant \min(Ra, Rb), \\ Ra + Rb - Rc & \text{otherwise.} \end{cases} \tag{2}$$

The predicted value Px is chosen by switching among three simple predictors: if a vertical edge on the left of the current location is detected, the predictor tends to pick Rb, if a horizontal edge above the current location is detected the predictor tends to pick Ra, finally if no edge is detected the predictor tends to pick the value $(Ra + Rb - Rc)$. At the end of this phase, a fixed predicted value is found. Notice that Rd is not used in this phase, since it is employed in the adaptive part of the predictor [13].

After this procedure, the prediction error $E = Ix - Px$ is computed and, in the near-lossless coding ($\Delta > 0$), the error is quantized (Q_E) according to the following rule:

$$Q_E = \left\lfloor \frac{E + \Delta}{2\Delta + 1} \right\rfloor, \tag{3}$$

where Δ is the maximum guaranteed preset error between the original and compressed images.

The proposed watermarking scheme works by modifying the quantized prediction as shown in

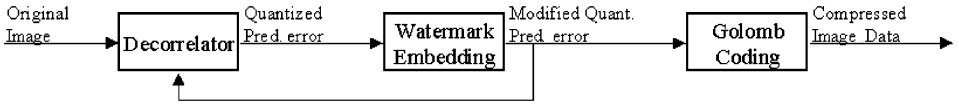

Fig. 2. Simplified block diagram of the proposed methodology.

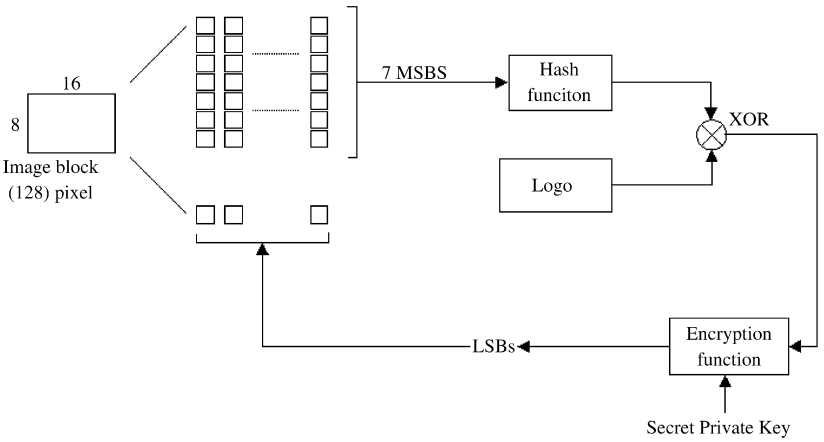

Fig. 3. Watermark embedding diagram of Fridrich's algorithm [5].

Fig. 2. Note that in order to keep the encoder in pace with the decoder the reconstructed values used by the predictor are obtained by adding back the watermarked prediction errors.

2.2. Watermark generation

Let us first summarize how the watermark is generated in the secure fragile watermarking technique developed by Fridrich [5] (a block diagram of this approach is given in Fig. 3). During watermark embedding, the algorithm proceeds by dividing the image into 8×16 pixel blocks and by separately modifying the LSBs of each block. To do so, the seven most significant bits (MSBs) of the pixels in the block are hashed by using a proper hash function. Then, a binary logo carrying information about the block position, image index and possibly other information relevant to the image is constructed, and is XORed with the hash. After that, the XORed result is encrypted using a secret-key-dependent encryption function, and inserted into the LSBs of the same block.

In the watermark detection phase, the to-be-authenticated image is divided again into 8×16 blocks and for each block the following procedure is applied. The seven MSBs of each pixel are extracted and hashed, while the LSBs are decrypted by using the secret key. In the end, the hashed MSBs and the result of LSBs decryption are XORed to obtain back the logo. Block-wise image authentication is achieved through an automatic examination of the logo. In this way, the watermarking scheme is robust to authentication attacks, such as stego-image attack, multiple stego-image attack and Holliman–Memon attack [8,6] (see Section 4); furthermore, localization of image tamper is granted.

By taking into account the JPEG-LS and Fridrich's algorithms, we developed a watermarking system that allows a near-lossless compression of the image and, at the same time, permits to insert a watermark into the to-be-authenticated image. To do so, the encoding procedure of the JPEG-LS algorithm has been modified in order to integrate the watermarking system while maintaining compliance with the JPEG-LS standard.

2.3. Watermark embedding phase

After watermark generation, the quantized prediction errors are modified in order to insert the watermark into the image and, finally, the corrected quantized prediction errors are Golomb–Rice coded and the compressed image obtained. More specifically, we proceed as follows. Let us consider an image of $D_R \times D_C$ pixels, consisting of blocks each of 8×16 pixels (i.e. $D_R/8$ stripes of blocks). For the

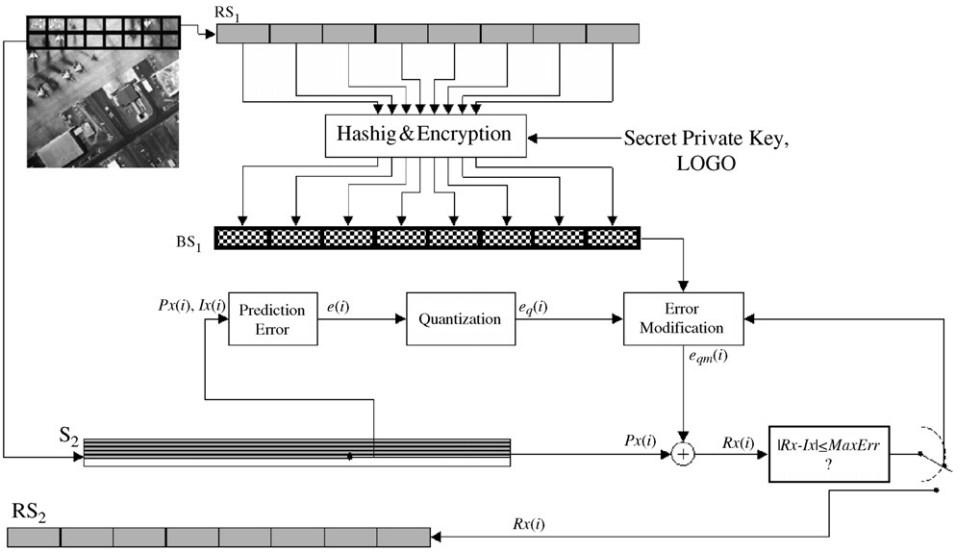

Fig. 4. Stripe watermark embedding scheme.

first stripe S_1, the reconstructed samples Rx are computed and stored to form a reconstructed sample stripe RS_1, which is formed by $D_C/16$ blocks each of 8×16 pixels. Then, each reconstructed sample block is processed by the watermarking system whose output is an 8×16 binary matrix (the authenticating message). When all the reconstructed sample blocks of the reconstructed stripe RS_1 have been processed, an $8 \times D_C$ binary stripe BS_1 is created. At the end of this process, for each sample in position (i,j) in the second stripe S_2 of the image, the quantized prediction error is calculated. Then, in order to insert the watermark into the image pixel in position (i,j), the quantized prediction error is modified by altering its LSB according to the corresponding bit of the authenticating message of the previous stripe.

In order to allow watermark recovery directly on the reconstructed pixel values[1] the parity of Rx has to be checked before performing any modification. For sake of clarity, let us give an example and let us suppose that Rx assumes an odd value and that a bit 0 has to be inserted (if a bit 1 has to be embedded no action is needed). To do this, the original Q_E is augmented or decreased by one quantization level to change its parity to obtain a $Q_{E1} = Q_E \pm 1$. By applying dequantization we obtain

$$Q_{R1} = Q_{E1} \cdot (2\Delta + 1) \tag{4}$$

and then

$$Rx_1 = Px + Q_{R1}. \tag{5}$$

The choice of increasing or decreasing the quantized error is made by choosing the option that minimizes maximum error between the original and the compressed and marked image. The quantization step $(2\Delta + 1)$ being an odd value, the modified parity is transferred to Q_{R1} and consequently to Rx_1 as required.

Fig. 4 summarizes the steps of the authentication procedure for the first stripe of pixels. First of all, it is possible to notice the *Hashing&Encryption* block which performs the hash of the seven MSBs[2] and the LSBs encryption for each image block in order to generate the binary stripe BS_1. Consequently, the prediction error of the successive stripe is computed, quantized and modified. As said before, the goal of the block that compares the difference between Rx and Ix is to choose the best modified prediction error, which limits the MaxError between the original and the authenticated image.

Through the above procedure, the authentication information of a stripe is embedded into the reconstructed samples of the stripe below. Finally, the reconstructed value Rx is stored to form the second reconstructed sample stripe RS_2, whereas Golomb–Rice coding of the modified quantized

[1]Watermark recovery on the quantized prediction error is straightforward.

[2]We could also hash all the eight bits, however we have considered only seven MSBs both to maintain coherence with Fridrich's algorithm and to hash only the bits belonging to the original image content.

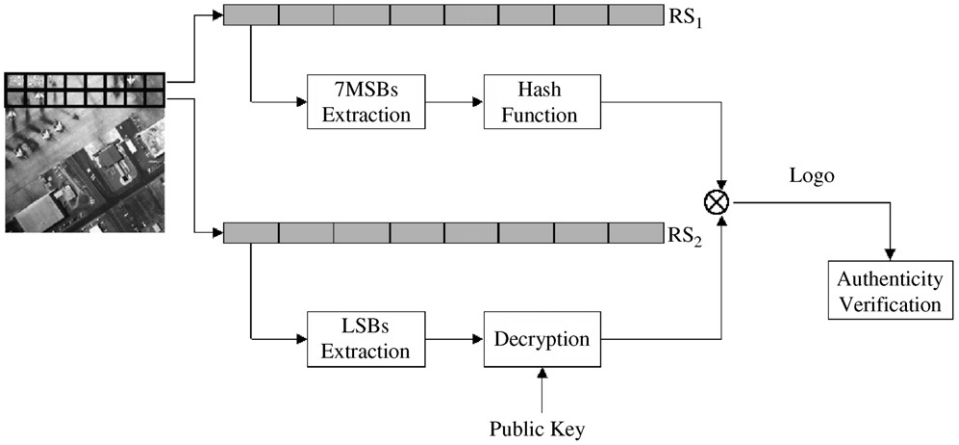

Fig. 5. Watermark detection scheme: the scheme represents the block-wise procedure that is followed to check image authenticity.

prediction error is performed. When all reconstructed samples Rx of the second stripe S_2 have been computed and stored, RS_2 has been constructed. The result of this process is RS_2 that has been modified according to BS_1. Generally, by following this procedure each RS_{i+1} is modified on the basis of BS_i and the watermarked–compressed image is generated. It is important to note that, in this system, the authentication binary matrix BS_i is embedded into the subsequent reconstructed sample stripe RS_{i+1}. This approach has been adopted because the JPEG-LS is based on a sequential procedure, whereas Fridrich's algorithm works block-wise. In the JPEG-LS algorithm, for each sample of the input image the corresponding reconstructed sample is found. If we desire to watermark this reconstructed value using Fridrich's algorithm, the binary matrix must be calculated previously; at the same time, this binary matrix can be only computed if all the reconstructed samples belonging to the block are known. This requirement contrasts with the sequential flow of the JPEG-LS.

As a final observation it is worth noticing again that the watermarked image is fully compliant with the JPEG-LS standard and hence can be decompressed by means of a standard decoder.

3. Watermark detection

In order to describe the authentication process let us consider, as we did for the coding phase, a $D_R \times D_C$ image. Watermark detection starts by dividing the image into 8-row stripes each consisting of 8×16 pixel blocks, as in the embedding phase. Then, for each image stripe the following procedure is applied. First of all, in order to verify the integrity of the first image stripe S_1, the second stripe S_2 is accessed to extract the LSBs and to complete the watermark detection (see Fig. 5). For each image block belonging to the first stripe S_1, the verification procedure continues as in Fridrich's algorithm. The seven MSBs are extracted and then hashed. At the same time, the LSBs of the corresponding image block in the second stripe S_2 are extracted and decrypted by using the public key corresponding to the one used by the embedder.[3] Finally, the hashed data and the decrypted LSBs are XORed and the authenticating logo is found. The information carried by the logo permits to verify the authenticity of each image block. A similar approach is followed for the subsequent stripes. In general, by analyzing two consecutive stripes it is possible to check each image stripe and in the end to check if the image is authentic as a whole or which parts (blocks) have been manipulated.

Note that authentication is carried out directly in the non-compressed domain thus increasing the flexibility of the proposed scheme.

4. Security issues

Security issues play a central role in watermarking-based authentication. In fact, content authenticity can be compromised by an ad hoc action made by an attacker who wants to create a fake document by resorting to all the information

[3]The use of a couple of private/public keys can be imagined for an application where authenticity verification is left to an end-user, otherwise a unique secret key could be adopted.

and capabilities available to him. It is important that an authentication algorithm is robust not only when a hacker has a unique image at his disposal (*stego-image attack*) but also when he can access other supplementary knowledge; hereafter some of the main security attacks against watermarking-based authentication are listed:

- *Multiple stego-image attack*: The counterfeiter has many authenticated documents and his action aims at making changes in such a way that the detector cannot reveal them or at gaining knowledge about the secret keys used by the scheme. A particular application of this attack is well known as the *Holliman and Memon attack* [8].
- *Verification device attack*: The aim of the counterfeiter is the same as before, but, in this case, he has access to the verification device and can use it to check the integrity of any image he likes. On the basis of the answer he gets he can rearrange the applied modifications to achieve a successful result. The kind of output the hacker obtains, either a simple Yes/No or a binary map containing authentic and tampered blocks, plays a key role in determining the potentialities of the attack.
- *Cover-image attack*: The counterfeiter has multiple pairs of original and authenticated images; this can happen when one has access to the image before authentication or when an estimate of the original can be performed. Again the hacker aims at making changes in such a way that the detector cannot reveal them or at gaining knowledge about the secret keys of the scheme.
- *Chosen cover-image attack*: The counterfeiter has the authentication device at his disposal and can submit his images to the authentication process; this could lead him to violate the secrets of the system.

Since the technique presented in this paper is based on the work by Fridrich [5], it inherits all the main security features of that algorithm. In particular, due to the specific structure of the logo, robustness to all the previous security attacks, included the *Holliman and Memon* one, is granted (see [5] for a discussion about the security of Fridrich's scheme).

5. Experimental results

In this section, some experimental results are given so to evaluate the performance of the authentication algorithm.

5.1. Watermark distortion

In this subsection, image distortion due to the watermark insertion is considered. Images belonging to remote sensing and biomedical scenarios are considered.

First of all, in Fig. 6 an example of original and authenticated images ($\Delta = 2$) is given, both for the case of remote sensing (*El Toro Airfield* 512×512) and for the case of medical imaging (*RX-Chest* 512×512). In both circumstances authentication does not introduce perceptual artifacts. To carry out a more objective analysis, the peak-signal-to-noise-ratio (PSNR) between the original image and the compressed one with different values of the Δ factor has been computed both in the case of near-lossless JPEG coding and in the case of joint authentication and coding. These results are presented in the graphs of Fig. 7 for *El Toro Airfield* image and in Fig. 8 for *RX-Chest*. It can be noticed that, as expected, there is a decrement (approximately 6–7 dB for each level of Δ factor) in the value of PSNR when the authentication information is embedded within the image. This worsening is about the same for both the types of image and is almost constant when the Δ factor increases. Our primary aim being that of designing a near-lossless scheme, where the maximum error can be strictly controlled, it is important to examine how the peak error varies as a consequence of watermark insertion.

This effect is due to the fact that during the watermark embedding phase, the quantized errors are modified in order to accomplish image authentication. In particular, each quantized error is changed to obtain a reconstructed sample whose LSB is equal to that of the corresponding binary stripe. As a result of this process, the quantized prediction error is varied by one quantization step whose value is $(2\Delta + 1)$. Because two possible quantization levels exist, the one which determines the minimum distance between the reconstructed sample and the original pixel Ix is chosen. This means that the two modifications (the one due to coding and the one due to watermarking) do not add each other, in such a way that the error is at

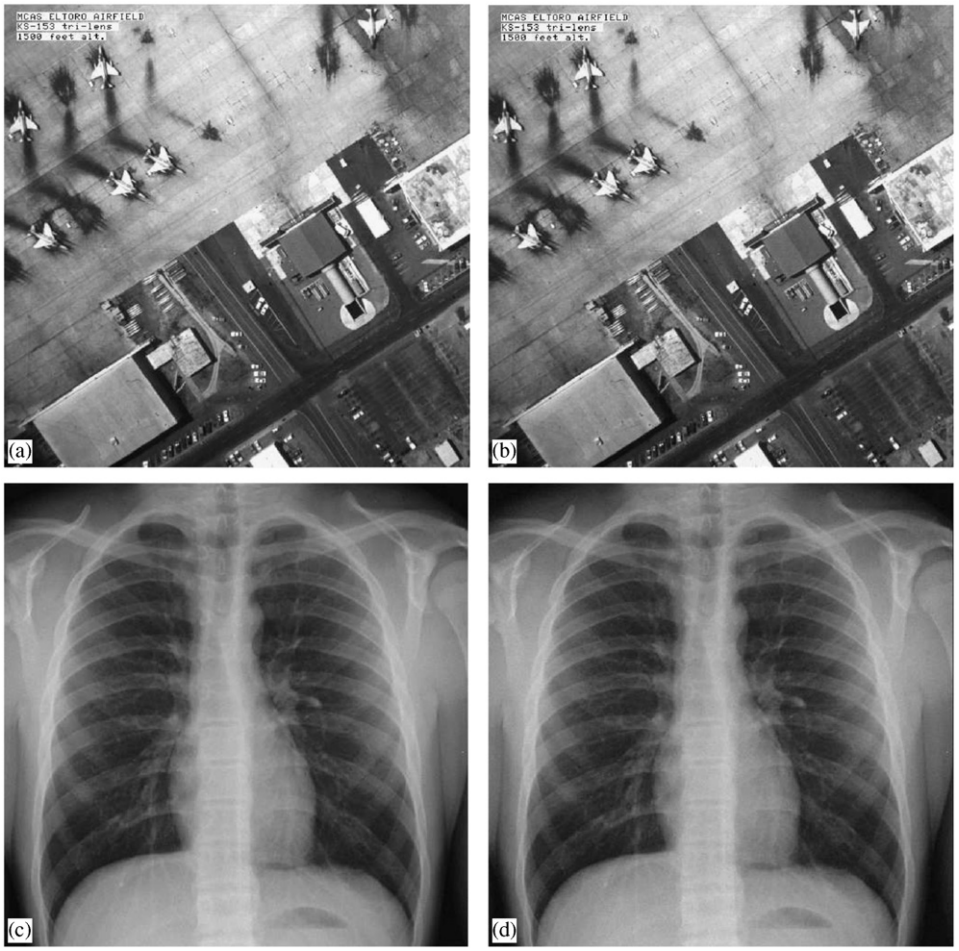

Fig. 6. *El Toro Airfield*: (a) original image and (b) authenticated image ($\Delta = 2$). *RX-Chest*: (c) original image and (d) authenticated image ($\Delta = 2$).

most $2\Delta + 1$. However, this choice is not possible in the case of pixel values that are near to 0 and 255 due to overflow and underflow problems. In this case, the choice to augment or decrease the quantized prediction error is obliged and the error could be equal to $\Delta + (2\Delta + 1)$.

In Figs. 9 and 10, the percentages of image pixels having a certain distortion error with respect to the original image for two sample images when Δ has been set to 1 and 2 are reported. It can be noticed that in all cases about 50% of the image pixels have a distortion within the Δ and almost 80% of the image pixels are at most one gray level beyond Δ.

5.2. Performance against attacks to authenticity

To examine the ability of the algorithm to ascertain image authenticity and to detect local modifications, near-lossless compressed and authenticated images have been tampered with and then authenticated.

In Fig. 11, three examples of counterfeited images are illustrated. Images in the left column have been modified by inserting some artifacts, in particular, in Fig. 11(a) an airplane originally belonging to the image has been duplicated on the airfield, while in Fig. 11(c) another airplane, a *B-52* taken from a different picture, has been added. In Fig. 11(e) a "false fracture" has been artificially induced on the right collarbone of the chest. In the corresponding right columns these alterations have been rightly detected by the proposed technique, the image blocks that the detector estimates to be altered are in black. The results demonstrate that the image authenticity is correctly verified, but the tamper localization accuracy is decreased with respect to

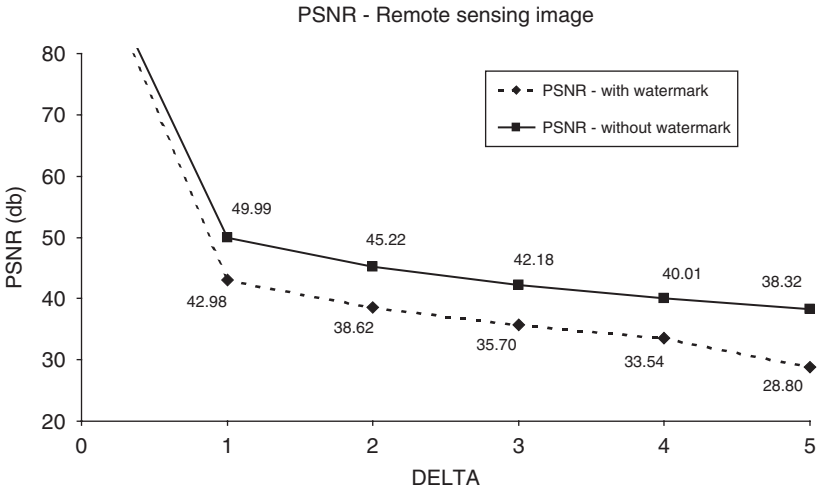

Fig. 7. *El Toro Airfield*. Graph of PSNR versus preset error Δ: continuous line JPEG-LS and dotted line JPEG-LS + WAT.

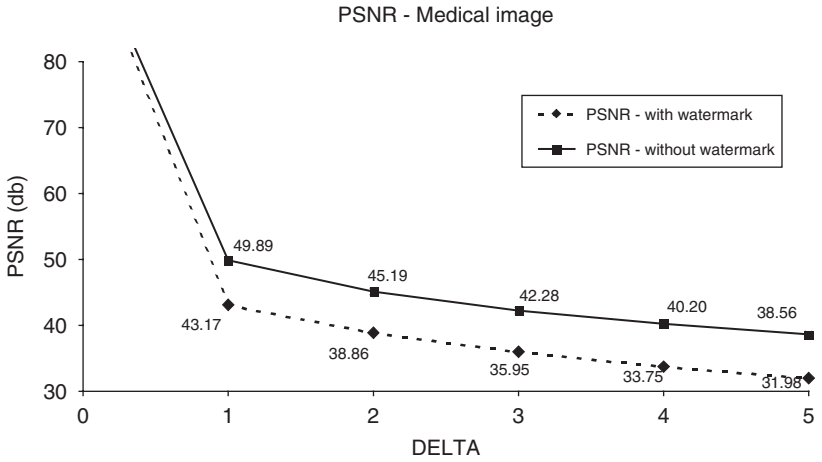

Fig. 8. *RX-Chest*. Graph of PSNR versus preset error Δ: continuous line JPEG-LS and dotted line JPEG-LS + WAT.

Fridrich's original work, in particular now accuracy is half. In fact, because the embedding procedure inserts into an image block b_i the binary map found utilizing the pixels of its upper block, it is impossible to distinguish if block modification has been applied to block b_i or to its upper neighbor. Both these circumstances lead to a non-authenticity detection.

5.3. Compression performance

Some tests, whose results are summarized in Table 1 for remote sensing and in Table 2 for medical images, have been carried out to establish the variation of compression rate between the JPEG-LS standard and the new integrated system. For each value of Δ the size of the compressed and marked/compressed images is given as a percentage of the original image. In the last column the difference in bytes between the size of the authenticated image and the image resulting from plain JPEG-LS compression is given. Interestingly this difference is much lower than the size of the embedded watermark, hence testifying the efficiency of the proposed embedding scheme.

Upon inspection of the tables, it can be seen the authentication procedure leads to a slight decrement of the compression efficiency compared to that achieved by the plain JPEG-LS algorithm. This result is mainly due to the fact that in the watermarking embedding procedure the difference between smooth and non-smooth regions cannot be exploited as usually done by switching between *run*

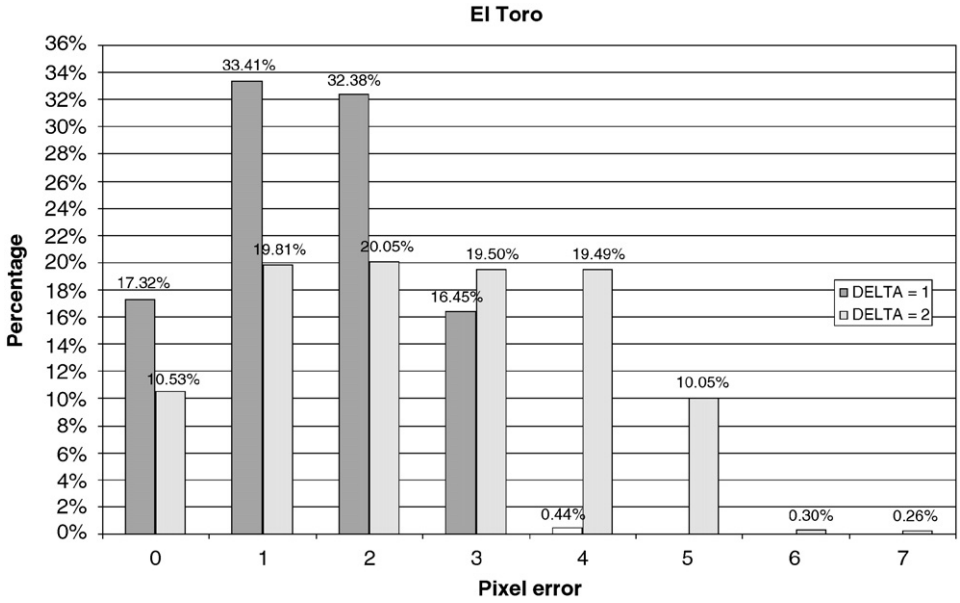

Fig. 9. *El Toro Airfield*. Histogram of the percentage of image pixels having a certain distortion error ($\Delta = 1$ dark and $\Delta = 2$ bright). The maximum error between the original image and authenticated one is $(3\Delta + 1)$, that is, 3 and 7, respectively.

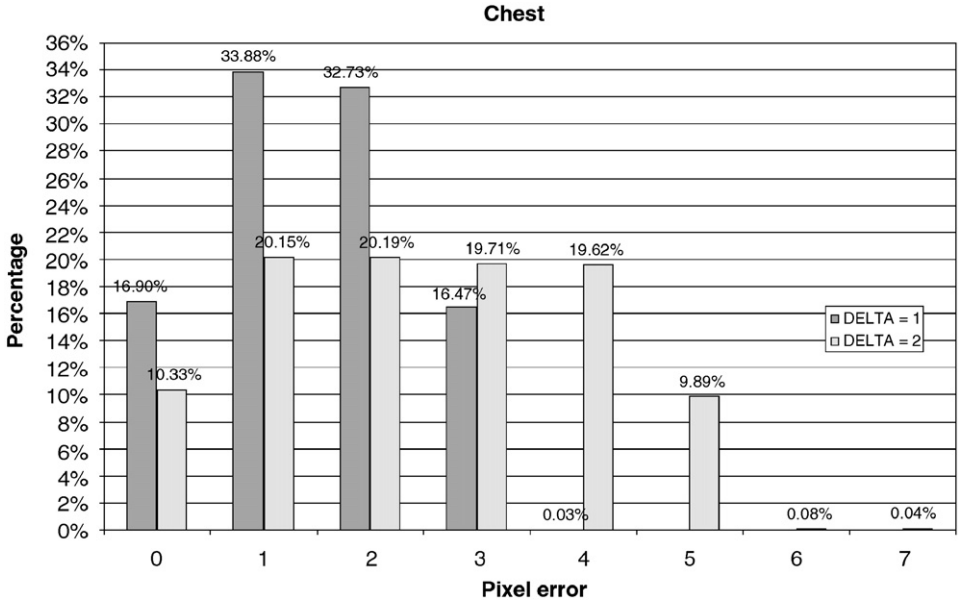

Fig. 10. *RX-Chest*. Histogram of the percentage of image pixels having a certain distortion error ($\Delta = 1$ dark and $\Delta = 2$ bright). The maximum error between the original image and authenticated one is $(3\Delta + 1)$, that is, 3 and 7, respectively.

mode and *regular mode* in the JPEG-LS coding. To confirm this claim, it has been noted that the compression rate decrease for highly textured images is less than that experienced in flat images, where the *run mode* allows to improve the compression performance.

Finally, to provide a further point of view, the rate–distortion trends obtained for the image *El Toro Airfield* are pictured in Fig. 12 (a similar behavior is registered for the image *RX-Chest*). This figure basically synthesizes values coming from the graphs in Fig. 7, regarding distortion in terms of

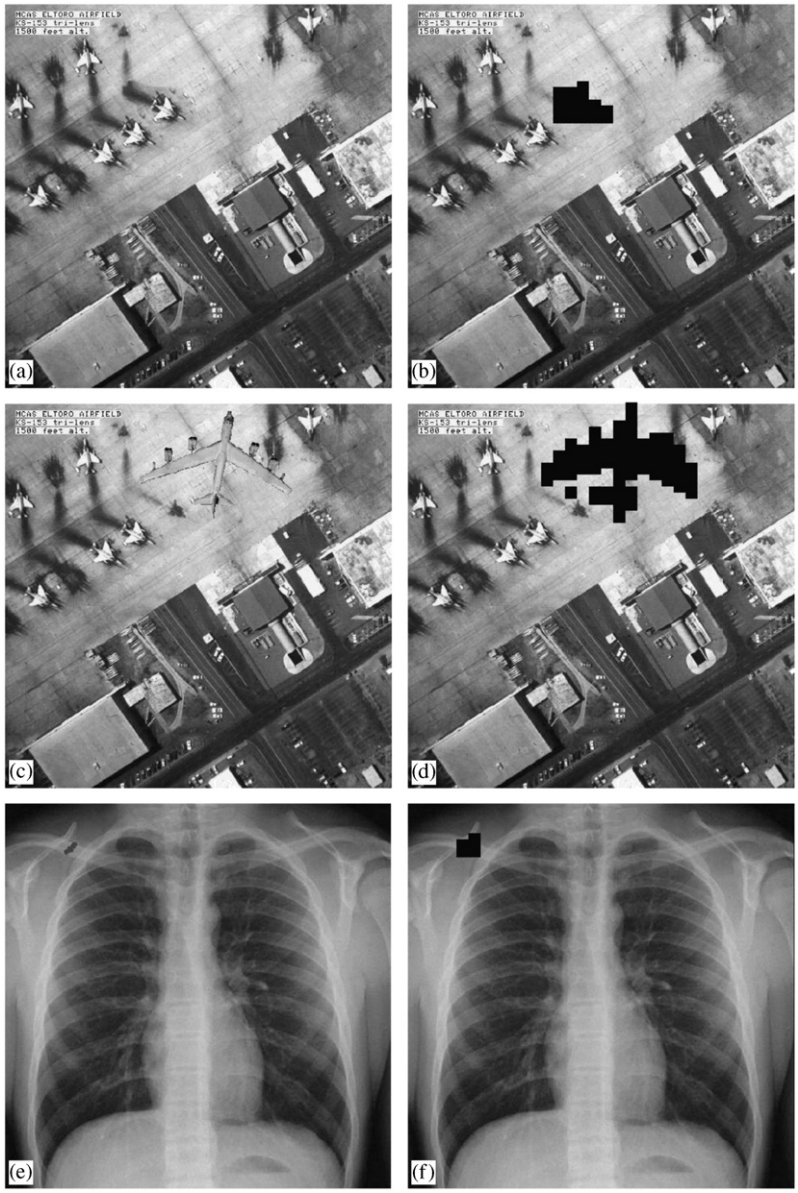

Fig. 11. *El Toro Airfield*: authenticated image after manipulation and detection of tampered zones (dark blocks) in the authenticated image, respectively: object replication (a) and (b), object insertion (c) and (d). *RX-Chest*: (e) authenticated image after manipulation. (f) detection of tampered zones (dark blocks) in the authenticated image.

PSNR, and Table 1, concerning compression rate. In addition to the two cases of simple JPEG-LS compression (bold dashed line) and JPEG-LS + WAT (dashed line), it has been considered another situation in which a signature of the same size of the watermark is attached to the header of the JPEG-LS image to convey an informative payload as the watermark does (continuous line). The dimension of this signature will have to be equal to the binary matrix embedded in the image with the proposed procedure (see Section 2.3). As pointed out in Eq. (6), the signature size depends on the dimension of the image itself ($D_R \times D_C$):

$$\text{Signature}_{\text{size}}\ (\text{bits}) = \left[\left(\frac{D_R}{8} - 1\right) \times \frac{D_C}{16}\right] \times 128, \tag{6}$$

where the amount $(D_R/8 - 1)$ represents the number of stripes contained in an image diminished by

Table 1
El Toro Airfield: output data size (percentage) with respect to the original size, obtained by JPEG-LS + WAT and JPEG-LS; byte increment (right column)

El Toro Airfield pgm 512 × 512 size: 262 159 bytes			
Δ	JPEG-LS + WAT Data size (percentage)	JPEG-LS Data size (percentage)	Increment Bytes
0	62.92	62.92	0
1	44.87	43.67	3152
2	37.63	35.29	6127
3	33.38	29.96	8983
4	30.61	26.50	10806
5	28.84	23.99	12717

Table 2
RX-Chest: output data size (percentage) with respect to the original size, obtained by JPEG-LS + WAT and JPEG-LS; byte increment (right column)

RX-Chest pgm 512 × 512 size: 262 159 bytes			
Δ	JPEG-LS + WAT Data size (percentage)	JPEG-LS Data size (percentage)	Increment Bytes
0	42.19	42.19	0
1	29.73	24.91	12641
2	25.56	19.81	15065
3	24.10	17.41	17523
4	23.50	15.68	20507
5	23.22	14.19	23690

one because the last binary stripe is not embedded and $D_C/16$ indicates the number of 8×16 blocks per stripe. In this circumstance, the image *El Toro Airfield* has $D_R = D_C = 512$ and, consequently, the signature size is 258 048 bits (32 256 bytes).

It can be observed that the JPEG-LS + WAT method permits to grant a performance slightly better than that of JPEG-LS + SIGNATURE as evidenced by the two lines of tendency; in particular, the first one allows to achieve a lower rate for a desired distortion not decreasing the PSNR under 40 dB. Anyway what is important to further highlight is that the signature, attached to the header in this manner, would not provide any real warranty for the authenticity of the image. In fact, being separated from the rest of the image, the signature could be, for instance, fraudulently deleted or substituted and nothing might be assessed in terms of image integrity; on the contrary, this is not possible with the proposed methodology. Obviously, the JPEG-LS curve outperforms the other two, but does not insert any informative payload at all.

5.4. Further improvements

Looking at the distortions applied to the images, due to watermark embedding, it could be deemed that the PSNR performance is not sufficient for specific applications, such as radiographies or

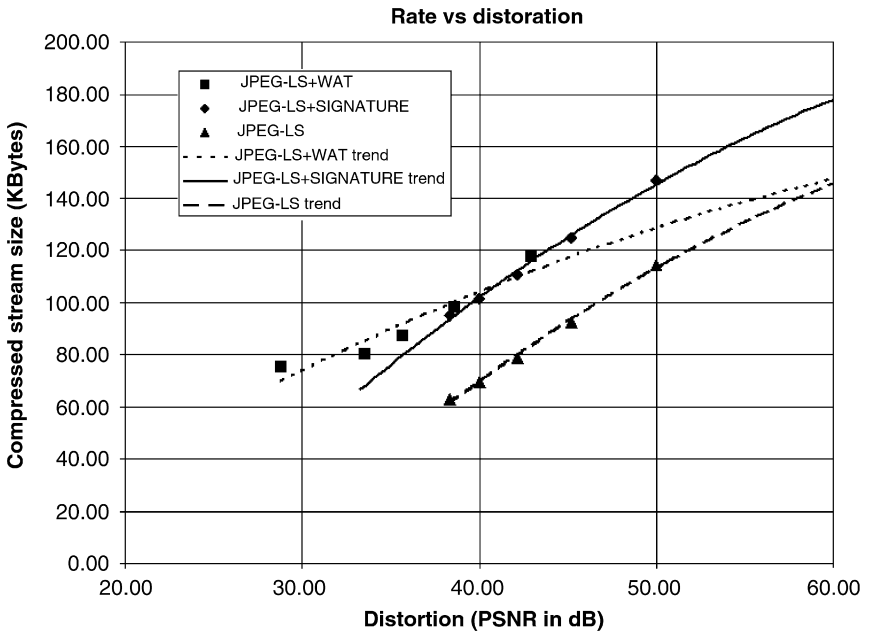

Fig. 12. *El Toro Airfield*: rate–distortion trends.

military imagery. In this case a higher visual quality can be achieved at the expense of a minor localization accuracy. For example, instead of hashing blocks of size 8×16, we could use blocks of 16×16, that is two stripes are processed together, and insert the authenticity map within a third stripe (see Fig. 13). This procedure, named *alternate watermarking*, always permits to protect the whole image but with reduced resolution. By observing Fig. 13, it can be seen that authenticity information of stripes number 1 and 2 is embedded in stripe number 3, and then authenticity information of stripes number 3 and 4 is embedded in stripe number 5 and so on. Doing so, it determines that, globally, one stripe is only JPEG-LS compressed (i.e. even stripes in dark color) and one is jointly JPEG-LS compressed and watermarked: this allows to reduce PSNR distortion and improve visual quality. In Figs. 14 (*El Toro Airfield*) and 15 (*RX-Chest*), the plots representing PSNR with respect to Δ are proposed again for the modified scheme, where an improvement of about 2 dB can be appreciated.

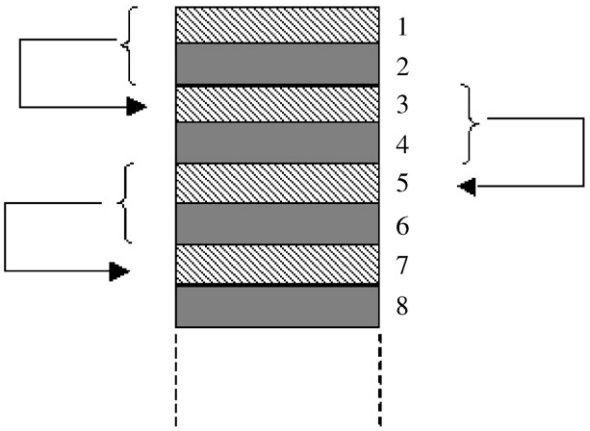

Fig. 13. Alternate watermark embedding procedure.

A similar trade-off between tamper localization accuracy and distortion can be achieved through other solutions, e.g. by applying an XOR operation to two consecutive image blocks of size 8×16 before using Fridrich's algorithm, thus reducing the authentication payload to be embedded. A further solution might foresee the adoption of concepts stemming from parity/syndrome or matrix embedding [7].

6. Conclusions

The system we presented in this paper permits to jointly compress and watermark a still image to allow subsequent tamper localization. The system was designed to take into account the peculiarities of application scenarios requiring that the degradation of the original image content is strictly controlled (near-lossless compression and watermarking). Particular care was paid to insure the security of the system. While the proposed system was specifically designed and tested to work on remote sensing and telemedicine imagery, its use is not limited to these scenarios. On the contrary, thanks to the compliance with the JPEG-LS coding standard and the possibility of retrieving the watermark even in the raw pixel domain, we believe that

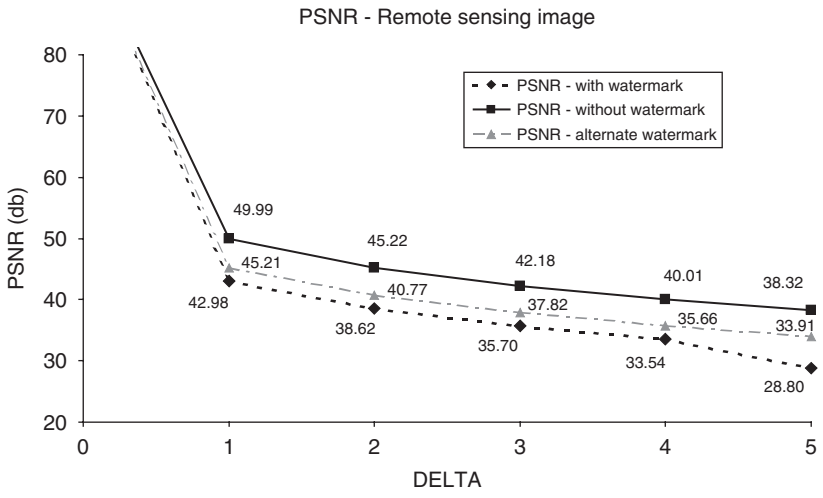

Fig. 14. *El Toro Airfield*. Graph of PSNR versus preset error Δ: continuous line JPEG-LS, dotted line JPEG-LS + WAT and dashed line alternate watermark.

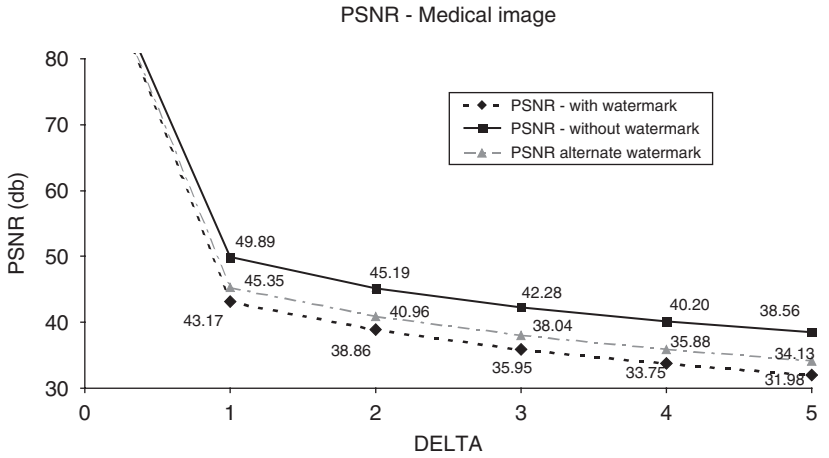

Fig. 15. *RX-Chest*. Graph of PSNR versus preset error Δ: continuous line JPEG-LS, dotted line JPEG-LS + WAT and dashed line alternate watermark.

our system can find application in a variety of real-life scenarios.

References

[1] M. Barni, F. Bartolini, V. Cappellini, E. Magli, G. Olmo, Watermarking-based protection of remote sensing images: requirements and possible solutions, in: Proceedings of SPIE Conference on Mathematics of Data/Image Coding, Compression, and Encryption IV, with Applications, S. Diego, CA, July 29–August 3, 2001.

[2] M. Barni, F. Bartolini, V. Cappellini, E. Magli, G. Olmo, Near-lossless digital watermarking for copyright protection of remote sensing images, in: IEEE International Geoscience and Remote Sensing Symposium, 2002, IGARSS'02, vol. 3, 24–28 June 2002, pp. 1447–1449.

[3] K. Chen, T.V. Ramabadran, Near-lossless compression of medical images through entropy-coded DPCM, IEEE Trans. Med. Imaging 13 (3) (September 1994) 538–548.

[4] J. Fridrich, Image watermarking for tamper detection, in: Proceedings of the ICIP98, IEEE International Conference on Image Processing, vol. II, Chicago, IL, October 1998, pp. 404–408.

[5] J. Fridrich, Security of fragile authentication watermarks with localization, in: Security and Watermarking of Multimedia Contents, Proceedings of the SPIE, vol. 4675, San Jose, CA, January 2002, pp. 691–700.

[6] J. Fridrich, M. Goljan, N. Memon, Further attacks on Yeung–Mintzer fragile watermarking scheme, in: Security and Watermarking of Multimedia Contents, Proceedings of the SPIE, San Jose, CA, 24–26 January 2000, pp. 428–437.

[7] J. Fridrich, D. Soukal, Matrix embedding for large payloads, in: E.J. Delp III, P.W. Wong (Eds.), Proceedings of the SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII, vol. 6072, San Jose, CA, January 2006, pp. 727–738.

[8] M. Holliman, N. Memon, Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes, IEEE Trans. Image Process. 9 (3) (March 2000) 432–441.

[9] ISO/IEC FCD 14495-1, Information technology—lossless and near-lossless compression of continuous-tone still images—part 1: baseline [JTC 1/SC 29/WG 1 N 575], in: ISO/IEC JTC 1/SC 29, July 1997.

[10] A. Krivoulets, Progressive near-lossless coding of medical images, in: Proceedings of the Third International Symposium on Image and Signal Processing and Analysis, 2003, ISPA 2003, vol. 1, 18–20 September 2003, pp. 202-207.

[11] A. Piva, R. Caldelli, F. Bartolini, M. Barni, Semi-fragile watermarking for still images authentication and content recovery, in: CD-ROM Proceedings of the WIAMIS 2004, Fifth International Workshop on Image Analysis for Multimedia Interactive Services, Lisboa, Portugal, 21–23 April 2004.

[12] A.M. Tekalp, Digital Video Processing, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[13] M. Weinberger, G. Seroussi, G. Sapiro, The LOCO-I Lossless Image Compression Algorithm: principles and standardization into JPEG-LS, IEEE Trans. Image Process. 9 (8) (August 2000) 1309–1324.

[14] P.W. Wong, A public key watermark for image verification and authentication, in: Proceedings of the ICIP98, IEEE International Conference on Image Processing, vol. I, Chicago, IL, October 1998, pp. 455–459.

[15] M.M. Yeung, F. Mintzer, An invisible watermarking technique for image verification, in: Proceedings of the ICIP97, IEEE International Conference on Image Processing, vol. 2, Santa Barbara, CA, October 1997, pp. 680–683.