High capacity reversible data hiding and content protection for radiographic images

Published version:

DOI:10.1016/j.sigpro.2015.05.020

(Article begins on next page)

doi: 10.1016/j.sigpro.2015.05.020

High capacity reversible data hiding and content protection for radiographic images

Davide Cavagnino, Maurizio Lucenteforte, Marco Grangetto

Dipartimento di Informatica, Università degli Studi di Torino, Corso Svizzera 185, 10149, Torino, Italia

Abstract

The watermarking of digital images has an important role in the protection of digital content with respect to many aspects. In this paper we present a reversible watermarking algorithm for hiding information into medical images having luminance histograms with particular characteristics. Some radiographic images have the property that not all the gray levels are present; this leads to sequences of 0 values (0-runs) in the corresponding histograms. It is possible to use these 0-runs to encode information by modifying pixels having gray levels contiguous to these runs; by encoding also the run information it is possible to restore the original image after extracting the stored data. In this work we present a novel reversible watermarking technique capable of exploiting all the 0-runs in the image histogram to achieve high capacity. We show that an optimization problem arises for those cases in which two or more non-zero frequency gray levels are contiguous to 0-runs. Part of the watermark information may be devoted to a digital signature of the original image, whose authenticity may be also verified by a user.

Keywords: Biomedical image processing, Data hiding, Watermarking

Email addresses: `davide.cavagnino@unito.it` (Davide Cavagnino), `maurizio.lucenteforte@unito.it` (Maurizio Lucenteforte), `marco.grangetto@unito.it` (Marco Grangetto)

1. Introduction

The diffusion of digital media and the consequent necessity of protection has posed new challenges and problems in the computer science field. At the same time objects of large dimensions (like images, sounds or videos) and containing a large quantity of redundant data allow for compression or, alternatively, storage of extra information. In the field of data hiding many different techniques have been developed; as a good overview of the research area one can refer to [1]. All the proposed approaches have in common the ability to store information into an object (called host), but the motivation and purpose of data embedding depend on applications. For example, steganography [2, 3] requires that the presence of the stored data cannot be revealed (nor statistically proven), whilst robust watermarking, as in [4, 5, 6], inserts a signal into a digital object in such a way that it is difficult to intentionally remove it without degradation. On the other hand, fragile watermarking, e.g. [7, 8], is aimed at revealing if any modification has been made to the digital object, possibly localizing the modified area. When altering a digital object, reversibility is an issue: as compression algorithms are classified as lossless or lossy, in the data hiding context the classification distinguishes between reversible and non-reversible. A reversible algorithm must store data and some additional information to recover the original host object, whilst a non-reversible algorithm deals with the issue of minimizing the embedding impact in terms of some measure, like the Peak Signal-to-Noise Ratio (PSNR). Generally, medical applications do not tolerate (also for legal aspects) any modification that does not allow the recovering of the original host object, so reversible algorithms are required.

In this work we target to improve the state of the art by designing a novel watermarking algorithm specifically tailored to radiographic images. To this end we have discovered that, taking into account the characteristics of such images, one can dramatically improve the watermarking capacity. In particular, we propose an algorithm that hides information into the image levels that are not present in the original image and appear as missing values in the histogram. As a consequence the proposed algorithm is able to store data into a radiographic image by exploiting only the intrinsic redundancy of the image (absence of some gray levels in the pixel values). The major contributions of the paper are:

- exploitation of typical characteristics of radiographic images' histogram for the development of a novel high capacity watermarking technique. In particular, we observed that, during the acquisition phase, radiographic images are usually enhanced so as to fit all the dynamic range. This process, in turn, produces a typical histogram where not all intensity levels are used; this leads to the creation of 0-runs in the histogram bins. The proposed technique exploits such 0-runs or "histogram holes" to hide data;

- the design of a simple iterative procedure to maximize the payload capacity. The proposed technique permits to store a variable number of bits per marked intensity level; therefore, the marked levels can be jointly optimized to achieve maximum capacity. Both analytical and algorithmic approaches have been followed to achieve such goal;

- the possibility to store multiple bits and the capacity optimization stage are very innovative elements of the proposed technique and are shown to provide excellent results, outperforming competing techniques in terms of payload size and visual impact on the original image;

- the development of a fully fledged prototype and extensive experimentation and comparisons on a large set of images; in particular 100 radiographic images acquired with different devices have been used to guarantee a strong statistical validation of the method.

The paper is organized as follows: in Sect. 2 some related works are reviewed; in Sect. 3 our scheme is presented whereas in Sect. 4 we propose our capacity optimization procedure. Sect. 5 presents our experimental results and comparisons whilst the final section draws some conclusions.

2. Related works

In this section we recall some previous works on data hiding in images. First of all we recall some works related to ours that have been proposed for general images, then we focus our attention on the medical imagery.

Many watermarking techniques have been proposed in the recent years exploiting different features of the images. When an algorithm performs computations exclusively using the pixel values, it is said to work in the spatial domain. Some proposed algorithms alter pixels depending on their neighborhood whilst other ones base their computation on the gray level histogram. We recall some of them here, and refer to [7] for a comprehensive survey of other methods. In [9] a reversible data hiding algorithm based on Difference Expansions (DEs) is proposed: pairs of (neighboring) pixels are considered for reversibly bearing one bit of payload by modifying their value's difference; given that not all pairs may satisfy the conditions for embedding, the pairs are classified as expandable, changeable, or none of them, and a location map of the pairs is compressed and saved along with the payload to ensure reversibility. The Tian's method has been extended by Alattar to triplets [10] and quads [11] of pixels by embedding in each group respectively two and three bits, and generalized in [12] to a reversible integer transform applied to vectors of pixels.

Among various techniques that reversibly embed data using characteristics from the histogram, we cite the following because they use an approach similar to ours. The paper by Ni et al. [13] uses histogram zeros and histogram shifting to embed one data bit in every pixel having a gray level in a certain range. The gray levels histogram is scanned to find its maximum and minimum: if the minimum is not zero the pixels having the gray level of the minimum are zeroed and their coordinates saved along with the data payload (for reversibility); then, the gray levels between the histogram maximum and the zero are shifted towards the zero, freeing a gray level contiguous to the maximum: the pixels corresponding to the maximum are used for (reversibly) embedding one bit using also the free adjacent gray level.

In [14] all pixels having the gray level that precedes a value not present in the image histogram are used for embedding one bit. The process is made reversible using some auxiliary information and the introduced distortion turns to be small given that the maximum pixel value modification is limited by 1. Nonetheless, for histograms that have many contiguous missing gray levels this approach is suboptimal in terms of embedding capacity: the present paper aims at filling this gap. Another work similar to our approach but limited to the usage of a single 0-run is [15] where a reversible data hiding method based on histogram shifting is presented; the longest run of zeros in the image histogram is located and shifted to be contiguous to the histogram peak level: calling q the run length, it is determined the optimum number of digits d in the q-ary system to be used to embed a binary string (i.e. performing a binary to q-ary converions), as a function of the frequency of the peak level.

In [16] spatial redundancy is exploited for watermarking insertion using sorting and prediction. The histogram of the prediction error is used to embed information using shifting.

Local histograms (i.e. histograms of blocks of pixels) are used in [17] to reversibly embed one watermark bit per pixels' block. One data bit is embedded shifting the histogram, depending on some properties of the local histogram distribution. Given that this procedure may be impossible in some cases, then an Error Correcting Code is used in the watermark to deal with errors introduced in embedding. The resulting embedding is semi-fragile. A following paper [18] identifies a weakness of the method, namely the impossibility to recover the original image (even if the watermarked image was not altered) due to some incorrect block classifications or a low embedding level. The new proposal introduces the concept of unstable block and the definition of a new embedding level that apparently solve the issues identified in [17]. Our concern remains in the fact that the new proposal needs the unstable blocks map as per-image side information, that in some sense vanishes the idea of watermarking.

A recent work for reversible data hiding exploiting two-dimensional histogram modification is presented in [19], where an injective mapping computed on difference-pairs allows to embed a watermarking bit modifying by one gray level a pixel of each pair.

Watermarking techniques have attracted much attention in the medical imaging area for their security aspects such as integrity, verification, tampering. For a complete and recent review of medical watermarking tools the interested reader is referred to [20]. Here we limit our attention to the approaches more relevant to our works in terms of algorithms and medical target image.

In [3] the original algorithm [13] is exploited in conjunction with block based prediction to hide data using the histogram of the prediction error. Another recent paper [21] is based on the concept of Tian's DE: the image is considered as composed by two disjoint areas, one containing the smooth regions and the other the remaining part. The two algorithms proposed therein divide the image into blocks of size 4×4 classifying each one as smooth or not; if the block is of the second kind it is embedded with data using Tian's DE (first algorithm) or Alattar's method [11] (second algorithm): the resulting embedding map is compressed and concatenated with the remaining data and inserted into the smooth blocks using [22]. A reversible spatial method employing dual layer embedding is presented in [23]. In the first layer, contiguous blocks of size 2×2 pixels are embedded with one data bit: the constraint posed by the method is that not all the gray levels can be used. Moreover, not all blocks may be able to carry one bit of information. A second layer is used to embed a 16 bit CRC for all the contiguous 16×16 blocks to enable tamper detection.

3. Multilevel histogram watermarking

Let us consider a $W \times H$ n-bits gray levels image $f(i,j) < 2^n$ with $i = 0, \dots, H-1$, $j = 0, \dots, W-1$ and the corresponding histogram $h(x) = n_x$, with n_x being the number of pixels having gray level x. In the specific case of a radiographic image a high dynamic range is used, e.g. $n = 12$. Moreover, radiographic images are generally contrast enhanced (the actual algorithm depending on the specific acquisition machinery) and, as a consequence of this process, some histogram bins are left empty. It follows that $\exists x : h(x) = 0$. More x values satisfying the previous predicate imply that less information (or equivalently more redundancy) is present in the image; this characteristic can be exploited to reversibly hide data as described in the following.

To this end let us term as 0-run a sequence of adjacent bins that are characterized by zero histogram values. We can assume to identify r 0-runs $R_0, R_1, \ldots, R_{r-1}$ in $h(x)$, where $R_m = [i, j]$ is a closed interval of levels with $h(x) = 0, i \leq x \leq j$. Given R_m we denote its lower and upper bounds as $L(R_m) = i-1$ and $U(R_m) = j+1$, respectively[1]. Now, let us define as *markable block* $B_k = \{R_m, R_{m+1}, \ldots, R_{m+s_k-1}\}$ a maximal sequence of s_k consecutive 0-runs separated by a single non-zero bin; this implies that the 0-runs satisfy the following constraint: $L(R_{l+1}) = U(R_l)$,$m \leq l < m + s_k - 1$. An example of histogram where a markable block can be identified is reported in Fig. 1. In the following we will use the notation $R_i^k = R_{m+i}$ to identify the i-th 0-run in the k-th markable block. Please note that, with a small abuse of notation, we also call block (of size $s_k = 1$) any isolated 0-run, i.e. with non-zero histogram bins preceding and following it (see markable block B_{k+1} in Fig. 1). According to these definitions it is possible to identify a total of b markable blocks B_k, $k = 0, 1, \ldots, b-1$ with $b \leq r$, in the whole histogram $h(x)$.

A markable block B_k is a maximal set of s_k 0-runs separated by isolated non-zero histogram bins. Such bins, termed in the following as *markable levels*, can be used for information hiding. In particular, we propose to encode the message in terms of shifts with respect to the original level. A receiver that is aware of the data hiding mechanism can compute the shift undergone by a certain intensity level, extract the hidden message and recover the original pixel value as well.

Given $B_k = \{R_0^k, R_1^k, \ldots, R_{s_k-1}^k\}$ one can identify $s_k + 1$ markable levels c_i^k, $i = 0, \ldots, s_k$, namely the lower level of each 0-run in the block $L(R_i^k)$, $0 \leq i < s_k$ plus the rightmost upper level $U(R_{s_k-1}^k)$. In the pictorial example

[1] Please note that lower or upper level may be undefined when $i = 0$ or $j = 2^n - 1$

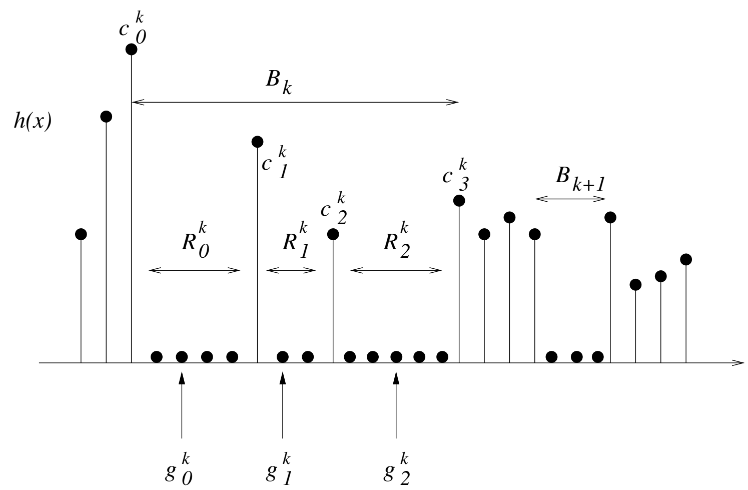

Figure 1: Sample image histogram with 0-runs, markable blocks, levels and thresholds.

shown in Fig. 1 a markable block comprising $s_k = 3$ 0-runs allows one to identify 4 markable levels c_i^k.

The watermark can be encoded shifting the markable levels. Clearly, the possible shifts are mutually constrained by the respective 0-runs ranges. To this end, as graphically shown in Fig. 1, we define a set of thresholds levels g_i^k that limit the possible shift of c_i^k within the interval $[g_{i-1}^k, g_i^k)$, $0 \leq i \leq s_k$. It follows that $g_i^k \in \{R_i^k \cup U(R_i^k)\}$, $0 \leq i < s_k$; moreover, to take the left and right edges of the block into account we also set $g_{-1}^k = L(R_0^k)$ and $g_{s_k}^k = U(R_{s_k-1}^k) + 1$.

Given any binary message d to be hidden, a mapping function can be used to shift a given markable level c_i^k by selecting an originally unused intensity level y computed as

$$y = g_{i-1}^k + z(d) \tag{1}$$

where $z(\cdot)$ is a function mapping the data onto a non-negative shift so that $y < g_i^k$. The proposed construction guarantees that $h(y) = 0$ in the original image (with the obvious exception of $h(c_i^k)$) and therefore assures that the watermarking is reversible. Clearly, the number of bits n_i^k that can be mapped by level c_i^k depends on the amplitude of the allocated shift interval and can be computed as $n_i^k = \lfloor \log_2(g_i^k - g_{i-1}^k) \rfloor$. As an example, using the markable levels and thresholds in Fig. 1 one can encode:

- 1 bit using c_0^k, e.g. by using pixel intensity c_0^k to represent 0 and $(c_0^k + 1)$ to represent 1;
- 2 bits for each of the remaining 3 levels c_1^k, c_2^k, c_3^k and corresponding thresholds; as an example we can map 00, 01, 10 and 11 messages onto pixel intensities $g_0^k = (c_1^k - 3)$, $(g_0^k + 1) = (c_1^k - 2)$, $(g_0^k + 2) = (c_1^k - 1)$ and $(g_0^k + 3) = c_1^k$, respectively.

3.1. Watermark insertion

The proposed watermarking technique is based on the mapping of the input message onto shifts of pixel intensity values taking into account the presence of holes, the 0-runs, in the original histogram. The watermark insertion procedure works as follows:

1. compute and analyze $h(x)$ to identify the markable levels c_i^k, with $k = 0, \ldots, b-1$, $i = 0, \ldots, s_k$;
2. select shift level thresholds g_i^k with $k = 0, \ldots, b-1$, $i = 0, \ldots, s_k - 1$;
3. store as side information the values c_i^k and g_i^k;
4. scan image to find pixels whose intensity x is equal to a markable level c_i^k and hide n_i^k payload bits using (1) to compute the new intensity y.

The side information c_i^k and g_i^k that need to be provided to the decoder, being an ordered set of integers, can be compressed efficiently. As an example, we observed that simple approaches based on differential coding of the markable levels and thresholds yield significantly compact representations.

In Fig. 2 we propose a possible format to represent the side information as a sequence of b markable blocks data. First of all, the number of blocks b is encoded using n_b bits. Each block B_k is described by providing the number s_k of 0-runs it contains (using n_s bits) and the leftmost markable level c_0^k (n bits intensity level); then, a sequence of differences pointing to the subsequent

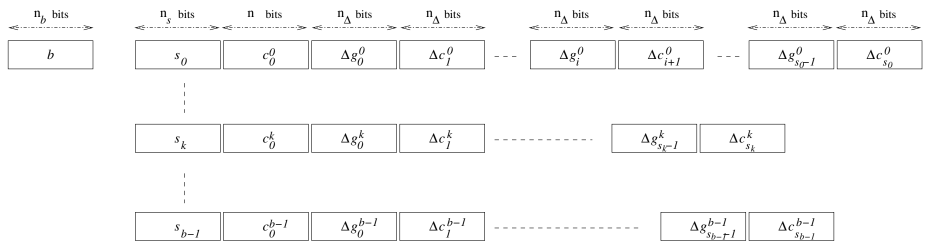

Figure 2: Bit allocation for side information.

g_i^k, c_i^k values is encoded. In particular, by defining $\Delta g_i^k = g_i^k - c_i^k$ and $\Delta c_i^k = c_i^k - g_{i-1}^k$ we can signal all the necessary levels as a list of pairs $(\Delta g_{i-1}^k, \Delta c_i^k)$, for $i = 1, \ldots, s_k$. Since all the levels within a block are generally close to each other, the number of bits n_Δ required to represent differential values can be kept limited. The overall cost required to signal all the necessary levels of the k-th markable block turns out to be $n(B_k) = n_s + n + s_k \cdot n_\Delta$ bits.

In Sect. 5 we will show that using this differential approach one gets a very compact representation that amounts to a negligible percentage of the payload that can be inserted. Moreover, it is worth pointing out that using the same technique presented in [14] it is possible to embed such information in the watermark itself avoiding to send ancillary data to the receiver. In this case, one can signal a single intensity level by using the least significant bits (LSBs) of certain pixels, e.g. the ones in the top-left corner. Such selected level is used to hide the removed LSBs and all the compressed ancillary information. The removed LSBs are included in the watermak so as to guarantee reversibility.

3.2. Watermark extraction

A decoder that knows the values c_i^k, $k = 0, \ldots, b-1$, $i = 0, \ldots, s_k$ and g_i^k, $k = 0, \ldots, b-1$, $i = 0, \ldots, s_k - 1$ is able to extract the watermark and restore the original image using the following procedure:

1. scan image pixel with intensity level y;
2. if $y \in [g_{i-1}^k, g_i^k)$ get the next n_i^k watermark bits d computing

$$d = z^{-1}(y - g_{i-1}^k) \tag{2}$$

and restore the original pixel value setting it to c_i^k.

3. if there are more pixels to scan go to 1.

4. Payload capacity optimization

In Sect. 3 we have shown that, analyzing the histogram of a radiographic image, one identifies a set of markable levels c_i^k that can host the watermark using shifts. To this end, we assumed to select proper threshold levels g_i^k that limit the maximum shifts and guarantee the reversibility of the process. In this section our goal is to maximize the watermark capacity by optimizing the selection of the threshold levels. The overall watermark capacity can be computed as:

$$C = \sum_{k=0}^{b-1} \sum_{i=0}^{s_k} n_i^k h(c_i^k) = \sum_{k=0}^{b-1} \sum_{i=0}^{s_k} \lfloor \log_2(g_i^k - g_{i-1}^k) \rfloor h(c_i^k) \tag{3}$$

Clearly, the values g_i^k must be jointly optimized taking into account both the position and histogram value of the markable levels. Since the markable levels of different markable blocks are not adjacent, the global optimization can be worked out independently on each block. In other words, we can write:

$$\max C = \sum_{k=0}^{b-1} \max \sum_{i=0}^{s_k} \lfloor \log_2(g_i^k - g_{i-1}^k) \rfloor h(c_i^k) \tag{4}$$

Therefore, from now on, we can focus on the maximization of the bits that can be embedded in a given block B_k. This amounts to the selection of s_k optimal thresholds g_i^k, $i = 0, s_k - 1$ per markable block.

The maximization problem may be solved with different strategies, each having varying complexities. To simplify this step, in this paper we propose to iterate a series of local optimizations on subsets of markable levels; the process is repeated until a steady state and/or a maximum number of iterations is reached. As shown in Fig. 3, it can be noted that the choice of the threshold levels g_{i-1}^k, g_i^k around a certain markable level c_i^k, i.e. the amplitude of the shift allocated to the i-th markable level, impacts the payload carried by 3 markable levels, namely $c_{i-1}^k, c_i^k, c_{i+1}^k$. Our local optimization approach aims at determining the optimal

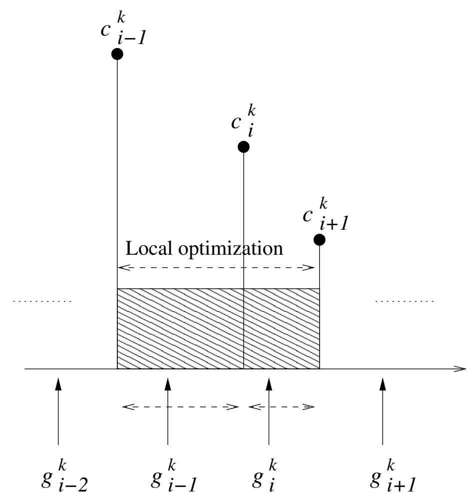

Figure 3: Local optimization approach.

values of the pair $\{g^k_{i-1},\ g^k_i\}$, given $g^k_{i-2},\ g^k_{i+1}$ and $h(c^k_{i-1})$, $h(c^k_i)$ and $h(c^k_{i+1})$. To this end we compute the payload, measured in bits, carried by the triple of levels $c^k_{i-1}, c^k_i, c^k_{i+1}$ as

$$P^k_i = \sum_{j=i-1}^{i+1} \lfloor \log_2(g^k_j - g^k_{j-1}) \rfloor h(c^k_j) \tag{5}$$

in order to maximize it. In particular, the previous equation can be used to set up the following constrained maximization problem:

$$\begin{cases} g^k_{i-1}, g^k_i & = \arg\max P^k_i \\ \text{subject to} & g^k_{i-1} > c^k_{i-1} \\ & g^k_i \leq c^k_{i+1} \end{cases} \tag{6}$$

This latter can be solved using a numerical procedure if the search space is small, i.e. the amplitude of zero runs (R^k_{i-1}, R^k_i) is limited. This amounts to compute the capacity given by a limited set of possible thresholds $\{g^k_{i-1}, g^k_i\}$ around c^k_i in order to select the optimal ones. If the numerical solution is not viable for complexity reasons we can introduce a continuous linear approximation $\widetilde{P}^k_i$ of the payload carried by the triple

$$\widetilde{P}_i^k = \sum_{j=i-1}^{i+1} \log_2(g_j^k - g_{j-1}^k) h(c_j^k) \tag{7}$$

where the rounding operation has been removed. If we consider unconstrained continuous optimization of $\widetilde{P}_i^k$, the optimal values $\{g_{i-1}^k, g_i^k\}$ can be found by equating to zero the gradients $\partial\widetilde{P}_i^k/\partial g_{j-1}^k = 0$ and $\partial\widetilde{P}_i^k/\partial g_j^k = 0$. With simple algebraic manipulations (see Appendix A) it can be found that $\widetilde{P}_i^k$, being a convex function, has a unique maximum given by:

$$\begin{aligned} g_i^k &= \frac{h(c_{i+1}^k) g_{i-2}^k + g_{i+1}^k \left[h(c_{i-1}^k) + h(c_i^k)\right]}{h(c_{i-1}^k) + h(c_i^k) + h(c_{i+1}^k)} \\ g_{i-1}^k &= g_i^k \left[1 + \frac{h(c_i^k)}{h(c_{i+1}^k)}\right] - g_{i+1}^k \frac{h(c_i^k)}{h(c_{i+1}^k)} \end{aligned} \tag{8}$$

Please note that, with a small abuse of notation, we considered the thresholds as continuous variables. Clearly, the optimal values given in (8) can be used to speed up the optimization process by providing an initial approximation for the constrained maximization in (6).

The local maximization can be performed iteratively on different adjacent levels to progressively refine the payload allocated to the block. In our implementation we apply the local optimization sequentially to all the threshold levels in B_k. As an example, for the block B_k shown in Fig. 1, one starts by optimizing the pair $\{g_0^k, g_1^k\}$, then moves to $\{g_1^k, g_2^k\}$ assuming g_0^k as given. Then the process is iterated for a maximum number of times M_I in order to reach a steady allocation. The whole procedure is summarized using pseudocode in Algorithm 1. For each block B_k, plausible thresholds are first initialized, then local optimization is repeated for a maximum number M_I of iterations. By iteratively applying the local optimization, one progressively refines the overall allocation taking into account the adjacent levels.

One can also take into account the bit cost $n(B_k)$, defined in Sect. 3, to signal the markable levels and thresholds of the k-th block. Clearly, it is worth using B_k only if the optimization is able to allocate a payload larger than its

Algorithm 1 `Optimize capacity`

for All markable blocks B_k **do**

  Initialize allocation g_i^k,$i = 0, \ldots, s_k - 1$

  for Maximum number of iterations M_I **do**

    for $i = 1, \ldots, s_k - 1$ **do**

      Perform local maximization on P_i^k to get $\{g_{i-1}^k, g_i^k\}$

    end for

  end for

  Mark B_k as unused if $\sum_{i=0}^{s_k} n_i^k h(c_i^k) \leq n(B_k)$

end for

signalling cost, i.e. only if

$$\sum_{i=0}^{s_k} n_i^k h(c_i^k) > n(B_k)$$

It follows that markable blocks not satisfying the previous constraint can be removed as a last step of the optimization process as shown in Algorithm 1.

5. Experimental results

In this section several aspects of the proposed data hiding technique are discussed ranging from implementation issues to performance evaluation in terms of watermarking capacity and imperceptibility, to security attacks. All the experiments are worked out on a data set comprising 100 high resolution radiographic images with $n = 12$ bit depth. The dataset includes images acquired by digital radiography (DR) systems and cassette based computed radiography (CR) systems [24], respectively. Since the proposed method depends on histogram characteristics, we keep the two image classes separate to better analyze the algorithm performance on different sensors and digitalization methods. In particular, in all the following experiments we will refer to the datasets "Set 1", comprising 86 images acquired with Kodak DirectView DR 5100 and DR 3000 systems, and "Set 2", containing 14 images captured by KODAK CR 260, CR 975 and ELITE systems.

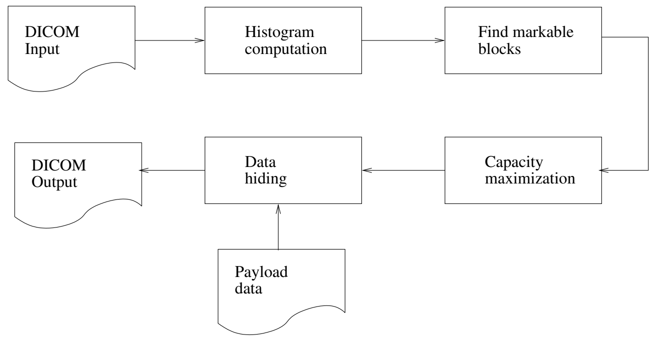

Figure 4: Block diagram of the implemented algorithm.

5.1. Algorithm implementation

The proposed technique has been implemented using $C++$ language and Imebra DICOM SDK library [25] for medical image input/output. The block diagram of the software implementation of the algorithm is shown in Fig. 4. The input DICOM image is first analyzed to compute the histogram and to locate the set of markable blocks and levels. Then the optimization procedure presented in Algorithm 1 is used to compute the thresholds maximizing the watermarking capacity. The optimizer also creates a header payload used to signal the markable levels using the compressed format described in Sect. 3; the header and the actual payload data are then hidden in the image. It is possible to make the stored data a fragile watermark appending to it a digital signature of the original host image and the payload itself. Such a signature can be used at the receiver side to check the authenticity of both the imaging data and additional information carried in the payload. Finally, if privacy is to be enforced on some data, these may be encrypted before embedding.

Different strategies can be used to decide which markable blocks/levels to use; as an example, depending on the size of the watermark, one may use with higher priority the markable levels that carry less bits, e.g. $n_i^k = 1$, so as to minimize the shift undergone by marked pixels to limit the distortion. A simpler solution is to scan the image pixels in raster scan order looking for markable

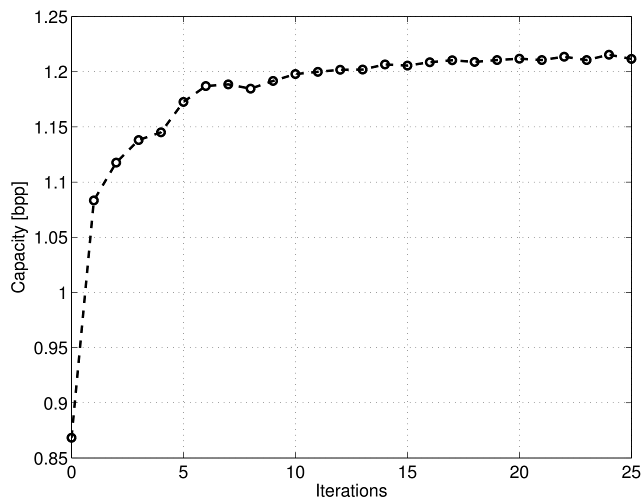

Figure 5: Watermark capacity as a function of optimization iterations.

levels; the pixels whose intensity x is equal to a markable level c_i^k are selected to store n_i^k bits of the payload using (1) to compute the new intensity value y of the marked pixel. On the available dataset we have observed that in most cases $n_i^k \leq 2$, i.e. the maximum shift error is bounded by ± 3. This feature along with fact that markable levels are usually a limited subset of the image dynamic, allows the simpler solution to achieve quite low *Mean Square Error* (MSE), as reported in the following section. On the contrary, the optimization of the levels to be used, given a desired payload size, would further increase the complexity of the technique whilst providing negligible MSE improvement. Therefore, all the following results are worked out using raster scan order data hiding.

The adopted design choices allowed us to develop a very fast prototype. As a representative case, our software is able to embed 1 MB payload into a 2456×2968 radiography in about 0.5 s on a standard PC with Intel® Core™ i5 CPU at 2.80 GHz. It is worth pointing out that the payload maximization algorithm can be configured to take less than 20 ms, whereas most of the computing time is due to input/output operations.

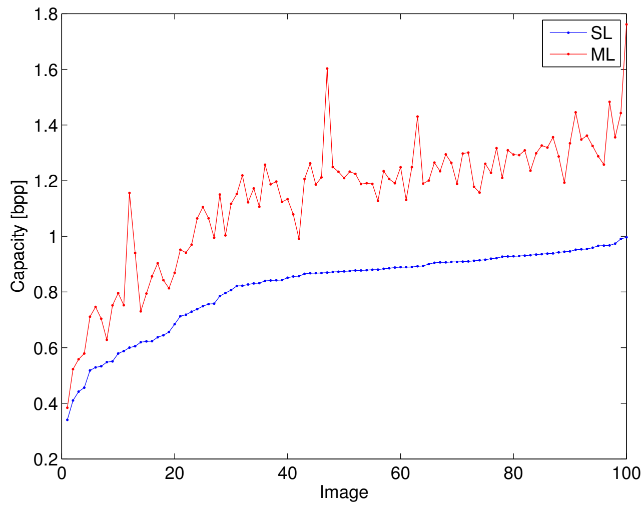

Figure 6: Capacity before (blue) and after (red) optimization.

5.2. Data hiding performance

In this section we analyze the data hiding performance in terms of embedding capacity, measured in bit per pixel (bpp), and visual perceptibility. As common practice in the literature, we compute MSE and Peak Signal-to-Noise Ratio (PSNR) between the original and watermarked images, even if it is well known that such metric may not correspond to the actual visual quality. It is worth noticing that anyway the proposed method guarantees perfect reversibility but visual quality represents an important feature for quick human viewing of the watermarked images.

As a first experiment we show the effectiveness of the proposed payload maximization algorithm on a sample image in Set 1 (DR type). In Fig. 5 the embedding capacity obtained on a 2456×2968 DR image is shown as a function of the maximum number of iterations M_I used in the optimization procedure. It can be observed that the iterative local optimization approach progressively refines the shifts allocation increasing the watermarking capacity from the initial 0.87 bpp up to about 1.2 bpp, with a gain of about 38%. The reported experiments also show that $M_I = 15$ iterations are enough to get an

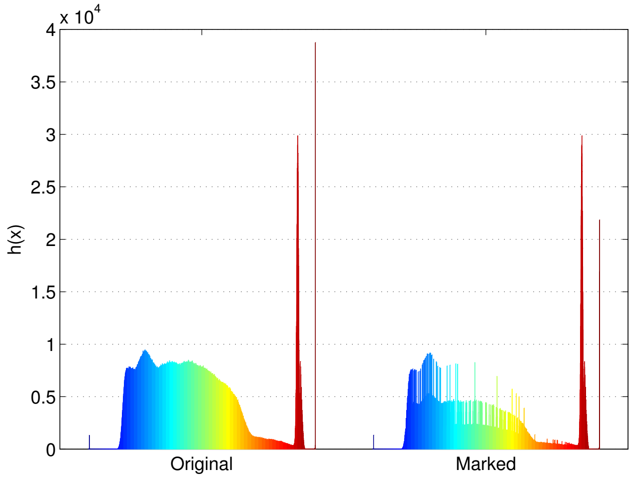

Figure 7: Image histogram before (left) and after (right) watermarking.

optimal allocation. It is worth noticing that the lowest capacity of 0.87 bpp is obtained using a static allocation where all markable levels are used to embed a single bit; in such a case, we obtain a watermarking algorithm that is similar to the one proposed in [14]. Previous experiments have also allowed us to measure the computational cost in terms of CPU time (on Intel® Core™ i5 CPU at 2.80 GHz); we observed that CPU time scales linearly in the number of iterations, taking from 2 ms up to 40 ms in the iteration range used in Fig. 5. We recall that our implementation also embeds the header information storing the list of markable blocks and the capacity is computed taking such bit cost into account. Nonetheless, the header size represents a negligible percentage of the available hiding space. As an example, in our experiments the header can be compressed using the format presented in Sect. 3 with $n_b = 8$, $n_s = 10$ and $n_\Delta = 2$; this amounts to a signalling cost that is about 0.1% of the embedding capacity.

In Fig. 6 the capacity achieved by the optimized multilevel (ML) algorithm and the starting capacity offered by the simple single bit per level (SL) allocation [14] are compared for all the images in our two datasets. For display purposes the images have been sorted in ascending order of SL capacity. It can

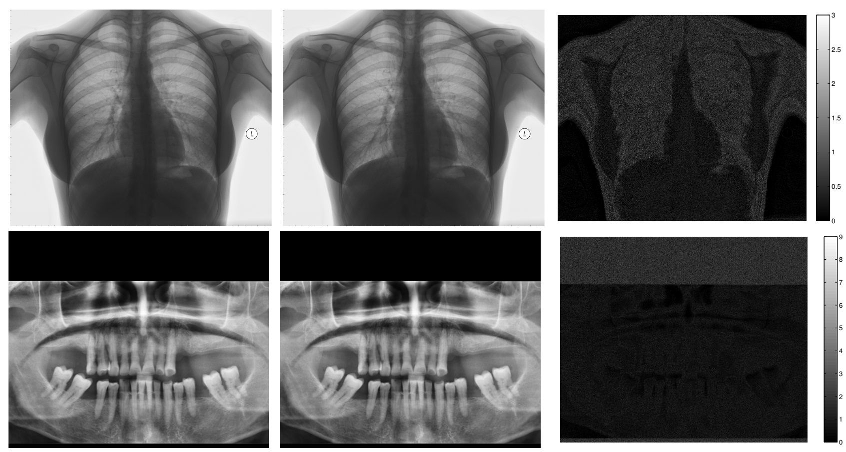

Figure 8: Examples: original image (left), watermarked image (middle) and corresponding absolute error (right).

be clearly noted that the proposed optimization procedure increases significantly the embedding capacity for all the tested images. These results confirm that the presence of holes in the histogram of radiographic images can be efficiently exploited to get high embedding capacity.

Now we focus on the analysis of the visual quality of the watermarked image. To this end we always use a pseudo-random binary payload. To better understand the effect of the watermark on the intensity levels, in Fig. 7 we compare the image histogram before (left) and after (right) data insertion at maximum capacity (1.2 bpp) on a certain image. The markable blocks used by our algorithm can be recognized in the histogram as intervals where the histogram counts have been distributed on a set of adjacent levels; therefore, one can note sections of the marked histogram (right) that have been flattened. One can point out that an attacker could recognize some features of the marked histogram and infer the presence of the watermark. It is worth pointing out that this event does not represent an issue in the scope of this work where we target integrity verification as detailed in the following Sect. 5.4.

Figures 8 and 9 allow one to inspect the visual quality obtained on three

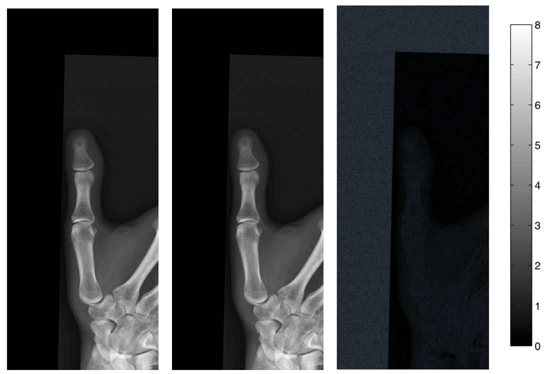

Figure 9: Example: original image (left), watermarked image (middle) and corresponding absolute error (right).

sample images. In particular we compare the original image (left) with the watermarked image at maximum capacity (middle). Each figure also shows the absolute error between the original and the watermarked data (right); please note that the absolute error image plot has been scaled to allow appreciating small differences (the color bar on the right side can be used to map plot intensity to the actual values). In Fig. 8 the visual quality obtained on a torso and a dental panoramic image, acquired with a Kodak DirectView DR 5100 and a Kodak ELITE system respectively can be appreciated. The first watermarked image hosts 1.2 bpp with a PSNR of 72.16 dB, whereas the second one hides 1.48 bpp with a PSNR of 63.20 dB. In both cases the inserted noise is visually imperceptible as witnessed by the limited dynamic of the absolute error, shown in the leftmost images. Fig. 9 shows an analogous example in the case of a thumb image acquired with a Kodac CR 975 equipment. This latter image can embed as much as 1.6 bpp with a PSNR of 62.65 dB. In this case it is also worth pointing out that most of the watermarking noise is inserted in the background area, as evident from the absolute error image.

In Fig. 10 the PSNR obtained on the previous image is shown as a function of the payload size in the rage from 0.1 bpp up to the maximum embedding capacity. As expected, the image quality remains quite high in the whole range

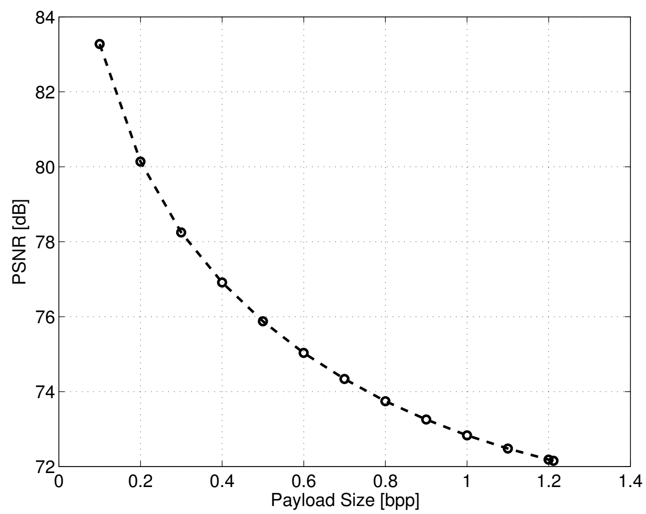

Figure 10: Marked image PSNR [dB] as a function of the payload size.

and shows that the proposed solution exhibits a scalable behavior.

In Fig. 11 we show a scatter plot of the maximum capacity and corresponding PSNR obtained using all the 100 images in our dataset; cross and circle marks refer to Set 1 and Set 2, respectively. It can be observed that Set 1 yields better embedding capacity/PSNR trade-off with respect to images in Set 2. This is likely to be due to different histogram characteristic induced by the acquisition device. Nonetheless the algorithm performance is quite good for all image kinds. In fact, it can be noted that 76% of the tested images can host more than 1 bpp. Moreover, most images exhibit PSNR values larger than 70 dB when marked at maximum capacity.

5.3. Performance comparison

The purpose of the following analysis is to compare our algorithm with other histogram based approaches. To this end we selected two schemes whose software has been made available by the authors, allowing us to reproduce the performance of all techniques on the same dataset. In particular, we have selected [19], that is based on the histogram of differences between pixel pairs, and

[16], that exploits the histogram of errors after spatial prediction. Both these schemes are designed to work with general images; therefore, our goal here is to show that the proposed algorithm achieves better performance exploiting the characteristics of the radiographic images histogram. We have used the software provided by [19] and [16] to estimate the maximum payload size that can be hosted in each of the 100 images and corresponding PSNR. In Tab. 1 we compare the performance of the proposed algorithm with multilevel (ML) and without multilevel (SL) optimization with that of the other techniques in terms of capacity (in bpp) and PSNR (in dB). In particular, we show minimum, average and maximum of the capacity and the PSNR obtained on the whole dataset. It can be observed that [19] is characterized by limited capacity (below 0.06 bpp on average) whereas it can guarantee high image quality (more than 75 dB in terms of PSNR). Using [16] one can obtain higher capacity of about 0.8 bpp, that in turn is paid with a significant amount of distortion, e.g. around 40 dB in PSNR. The results in Tab. 1 clearly show that the proposed algorithm is able to strike an optimal balance between capacity and quality. Indeed, the SL solution already yields much better capacity/PSNR trade-off with respect to [19] achieving about 0.8 bpp and almost 90 dB of PSNR. It is worth pointing out that this performance is similar to [16] in terms of capacity but we get much better image quality. Finally, the proposed multilevel optimization allows us to further increase the maximum capacity of about 40% while keeping the PSNR significantly high, achieving payload sizes larger than 1 bpp with PSNR beyond 65 dB. To better notice the different trade-offs yielded by the 3 compared techniques in Fig. 12 we represent as a scatter plot each pair PSNR versus maximum capacity obtained on every image. It can be easily noted that the proposed solution pushes the performance toward the top/right corner yielding better capacity and image quality. Finally, in Fig. 13 we also show the PSNR comparison obtained on all the marked images when the proposed method and [19] are used to embed the same amount of data; in particular, we set the embedding capacity equal to the maximum provided by [19], since this latter method can host a smaller payload. It is worth noticing that our solution gets much

Table 1: Minimum, average and maximum of the capacity (bpp) and PSNR (dB) obtained on the data sets using [19], [16] and the proposed algorithm with multilevel (ML) and without multilevel (SL) optimization.

		Set 1				Set 2			
		[19]	[16]	SL	ML	[19]	[16]	SL	ML
capacity	min.	0.008	0.50	0.44	0.55	0.007	0.50	0.34	0.38
	ave.	0.04	0.65	0.83	1.14	0.06	0.80	0.75	1.11
	max.	0.21	0.90	0.99	1.45	0.17	0.95	1.00	1.76
PSNR	min.	75.32	37.69	82.47	69.92	75.42	37.67	82.94	52.01
	ave.	75.45	39.59	91.02	72.78	76.16	42.73	87.97	66.10
	max.	76.26	43.97	94.91	76.92	80.05	52.31	93.22	73.11

higher PSNR when embedding the same amount of data.

We conclude the comparison with the state of the art by targeting other works that apply reversible watermarking to biomedical and, in particular radiographic, images. In this area there is not yet a reference image dataset and therefore it is not straightforward to reproduce and compare the results if the software is not made available. In the following we provide some comparisons using previous published results.

One first work that is certainly important to mention and compare to is the one in [3], where block prediction and histogram shifting are jointly used to reversibly hide information in biomedical images. Therefore we believe that this histogram based technique is the most similar to ours. In [3] it is found that the average quality of the resulting image is greater than 49 dB while the average capacity is more than 42 thousands bits for images of resolution of about 500×500, i.e. about 0.17 bpp. The best capacity of 0.43 bpp is achieved on a 512×512 X-ray image. These results, even if worked out on different image data, support the conclusion that our algorithm exhibits very competitive performance in terms of both capacity and image quality.

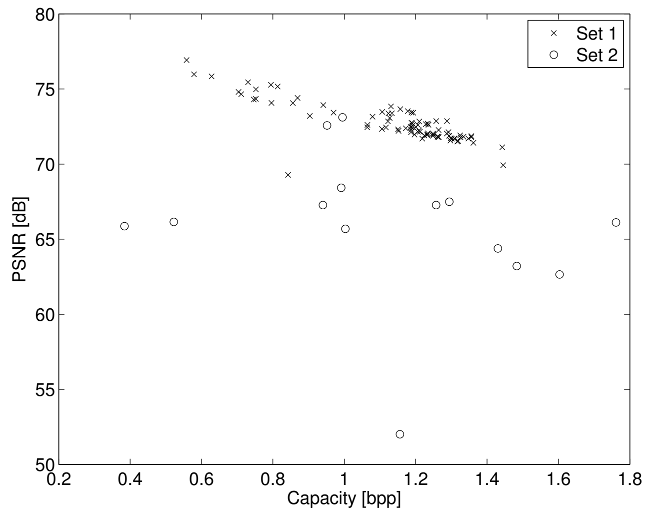

Figure 11: PSNR vs. capacity scatter plot on whole dataset.

Finally, we compare our performance with other recent biomedical image watermarking methods that use different approaches. In [21] a set of DE techniques is tested on 4 X-ray images with resolution and bit depth (12 bits) equal to those included in our dataset; therefore, we believe the comparison with those results is possible and quite meaningful. The maximum capacity obtained on an image in [21] is about 0.78 bpp; the quality of the watermaked images ranges from about 41 to 54 dB when the payload size is 0.1 and 0.6 bpp, respectively. Even if we cannot assure a direct comparison on the same image we can certainly state that our proposed algorithm yields high capacity and quality. Another work that provide experiments on a single 8 bit X-ray image is [23] where a capacity of 0.56 bpp is reported. In this latter case the comparison with our results is less significant given the lower dynamic of the used image.

5.4. Security considerations

In this subsection the content integrity aspects of the proposed watermarking algorithm are discussed. Integrity protection may be achieved appending a Message Authentication Code (MAC) to the payload string and inserting it as

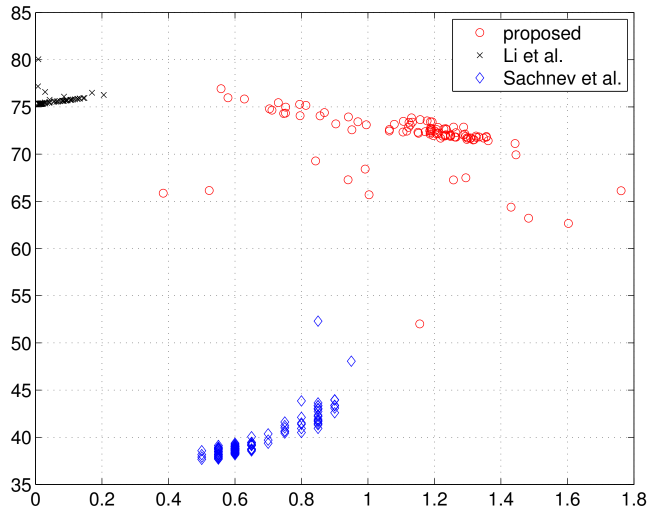

Figure 12: PSNR vs. capacity scatter plot for proposed algorithm, Li et. al [19] and Sachnev et al. [16].

part of the watermark. If proof of origin is also necessary, a Digital Signature (DS) is used in place of the MAC. In the following we will refer to one of these methods as integrity algorithm.

The integrity algorithm is applied to the ordered sequence of image pixels concatenated to the watermark: the result is appended as the final part of the watermark. We consider an implementation where all the required side information about markable levels is included in the watermak following the LSB substitution approach discussed at the end of Sect. 3. As a consequence, the watermark message is composed by the removed LSBs, the side information on marked levels and the actual payload data; in this case the integrity algorithm can be used to guarantee their correctness.

Now let us analyze how the whole image content is protected. We may distinguish pixels in two sets: those bearing watermark bits (call them of type A) and the remaining ones (call them of type B). Let us consider the modification attack of one pixel only, by one gray level; such an attack can results in one of the following cases:

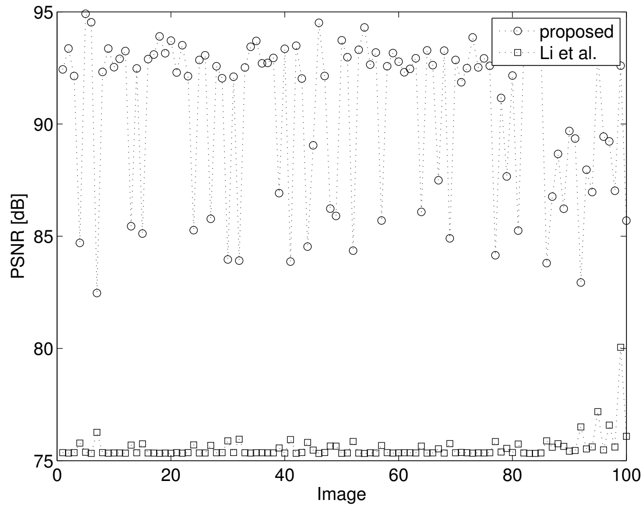

Figure 13: PSNR comparison between the proposed algorithm and Li et al. [19] when setting the embedding capacity equal to the maximum of the latter method.

- alteration from type A to type A: happens when a pixel containing the watermark is modified to another value, encoding a different watermark bit sequence: given that the integrity algorithm protects the watermark, this modification is detected during the MAC (or DS) verification;

- alteration from type A to type B: the resulting pixel value does not bear any information, so it will be left as is after watermark extraction, producing an altered watermark and also a modified pixel: this will be detected by the integrity verification algorithm;

- alteration from type B to type A: some bit(s) will be added to the watermark, allowing detection by the integrity verification algorithm (longer watermark and wrong restored pixel);

- alteration from type B to type B: an altered pixel will be detected by the integrity algorithm.

Detailing an attack to some sensible information we may say that:

- if one of the LSBs signaling the gray level encoding the compressed ancillary information will lead to a wrong set of c_i^ks and g_i^ks, thus a wrong authentication information will reveal wrongly decoded data;
- if one of the pixels encoding the c_i^ks and g_i^ks, or the initial LSBs, is altered, the integrity algorithm will detect the modification.

In all cases an alteration will be detected by the integrity algorithm, signaling that the image and the payload should not be trusted.

6. Conclusions

In this paper we have shown that exploiting a typical characteristic of radiographic images, in particular the presence of "holes" in the histogram, a novel high capacity reversible watermarking technique can be designed. The achievable embedding capacity can be increased by setting up a proper optimization problem. In this paper we have proposed an iterative method representing an efficient solution while keeping the computational complexity very limited. Our experiments show that the proposed technique yields excellent results both in terms of the maximum embedding capacity and marked image quality. It is worth pointing out that, given the reversibility of the data hiding process, the quality of marked image is important only from the point of view of visual inspection because the original host image can always be restored and its authenticity verified by means of a digital signature (part of the watermark).

Future works include evaluating the use of the [15] method in the proposed algorithm and the usage of the proposed algorithm in a security framework providing, apart from image authentication, also tamper localization.

Acknowledgment

The authors are grateful to Xiaolong Li, Weiming Zhang, Xinlu Gui and Bin Yang for providing the source code of their algorithm.

Appendix A. Analytical optimization details

In the following we provide some details of the analytical optimization of the payload that can be carried by a triple of levels by maximizing (7). To avoid cluttering the notation we drop the indexes that are not useful in this section and we redefine the capacity of the triple as:

$$P_i = \sum_{j=i-1}^{i+1} \log_2(g_j - g_{j-1})h(c_j)$$

This function is a summation of 3 logarithmic terms, it is convex and it has a unique maximum that can be determined by setting its gradient to zero. To this end we compute:

$$\begin{aligned}\frac{\partial P_i}{\partial g_{i-1}} &= \frac{1}{\ln 2}\left(\frac{h(c_{i-1})}{g_{i-1}-g_{i-2}} - \frac{h(c_i)}{g_i - g_{i-1}}\right)\\ \frac{\partial P_i}{\partial g_i} &= \frac{1}{\ln 2}\left(\frac{h(c_i)}{g_i-g_{i-1}} - \frac{h(c_{i+1})}{g_{i+1} - g_i}\right).\end{aligned}$$

Setting these partial derivatives to zero one gets:

$$\frac{h(c_{i-1})}{g_{i-1}-g_{i-2}} = \frac{h(c_i)}{g_i - g_{i-1}} \tag{A.1}$$

$$\frac{h(c_i)}{g_i - g_{i-1}} = \frac{h(c_{i+1})}{g_{i+1} - g_i} \tag{A.2}$$

By simplifying (A.2) we get:

$$\begin{aligned}h(c_i)(g_{i+1}-g_i) &= h(c_{i+1})(g_i - g_{i-1})\\ h(c_{i+1})g_{i-1} &= h(c_{i+1})g_i - h(c_i)(g_{i+1}-g_i)\end{aligned}$$

It turns out that the optimal threshold g_{i-1} is given by:

$$g_{i-1} = g_i\left(1 + \frac{h(c_i)}{h(c_{i+1})}\right) - g_{i+1}\frac{h(c_i)}{h(c_{i+1})} \tag{A.3}$$

Then, by equating (A.1) and (A.2) we get:

$$\begin{aligned}\frac{h(c_{i-1})}{g_{i-1}-g_{i-2}} &= \frac{h(c_{i+1})}{g_{i+1}-g_i}\\ h(c_{i-1})(g_{i+1}-g_i) &= h(c_{i+1})(g_{i-1}-g_{i-2})\end{aligned}$$

The optimal g_i can be computed as follows:

$$g_{i-1} = \frac{h(c_{i-1})}{h(c_{i+1})}(g_{i+1} - g_i) + g_{i-2}$$

$$g_i \left(1 + \frac{h(c_i)}{h(c_{i+1})}\right) - g_{i+1}\frac{h(c_i)}{h(c_{i+1})} = \frac{h(c_{i-1})}{h(c_{i+1})} + (g_{i+1} - g_i) + g_{i-2}$$

$$g_i \left(1 + \frac{h(c_i)}{h(c_{i+1})} + \frac{h(c_{i-1})}{h(c_{i+1})}\right) = \left(\frac{h(c_{i-1})}{h(c_{i+1})} + \frac{h(c_i)}{h(c_{i+1})}\right) g_{i+1} + g_{i-2}$$

$$g_i = \frac{(h(c_{i-1}) + h(c_i))\, g_{i+1} + h(c_{i+1})g_{i-2}}{h(c_{i-1}) + h(c_i) + h(c_{i+1})}. \tag{A.4}$$

Therefore, the optimal pair can be found by first computing g_i using (A.4) and then substituting it in (A.3) to calculate g_{i-1}.

References

[1] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Steganography (Second Edition), Morgan Kaufmann, Burlington, 2008.

[2] N. Provos, P. Honeyman, Hide and seek: An introduction to steganography, IEEE Security & Privacy 1 (3) (2003) 32–44.

[3] P. Tsai, Y.-C. Hu, H.-L. Yeh, Reversible image hiding scheme using predictive coding and histogram shifting, Signal Processing 89 (6) (2009) 1129 – 1143.

[4] M. Barni, F. Bartolini, V. Cappellini, A. Piva, A dct-domain system for robust image watermarking, Signal processing 66 (3) (1998) 357–372.

[5] P. Tsai, Y.-C. Hu, C.-C. Chang, A color image watermarking scheme based on color quantization, Signal Processing 84 (1) (2004) 95–106.

[6] A. A. Mohammad, A. Alhaj, S. Shaltaf, An improved svd-based watermarking scheme for protecting rightful ownership, Signal Processing 88 (9) (2008) 2158 – 2180.

[7] A. Khan, A. Siddiqa, S. Munib, S. A. Malik, A recent survey of reversible watermarking techniques, Information Sciences 279 (2014) 251–272.

[8] S. Bravo-Solorio, A. K. Nandi, Secure fragile watermarking method for image authentication with improved tampering localisation and self-recovery capabilities, Signal Processing 91 (4) (2011) 728–739.

[9] J. Tian, Reversible data embedding using a difference expansion, Circuits and Systems for Video Technology, IEEE Transactions on 13 (8) (2003) 890–896.

[10] A. M. Alattar, Reversible watermark using difference expansion of triplets, in: Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, Vol. 1, 2003, pp. 501–4.

[11] A. M. Alattar, Reversible watermark using difference expansion of quads, in: Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, Vol. 3, 2004, pp. 377–80.

[12] A. M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, Image Processing, IEEE Transactions on 13 (8) (2004) 1147–1156.

[13] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Techn. 16 (3) (2006) 354–362.

[14] N. Balossino, D. Cavagnino, M. Grangetto, M. Lucenteforte, S. Rabellino, A high capacity reversible data hiding scheme for radiographic images, Communications in Applied and Industrial Mathematics.

[15] M. Fujiyoshi, H. Kiya, Generalized histogram shifting-based reversible data hiding with an adaptive binary-to-q-ary converter, in: Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, 2012, pp. 1–4.

[16] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, Y. Q. Shi, Reversible watermarking algorithm using sorting and prediction, Circuits and Systems for Video Technology, IEEE Transactions on 19 (7) (2009) 989–999.

[17] Z. Ni, Y. Shi, N. Ansari, W. Su, Q. Sun, X. Lin, Robust lossless image data hiding designed for semi-fragile image authentication, Circuits and Systems for Video Technology, IEEE Transactions on 18 (4) (2008) 497–509.

[18] X. Gao, L. An, X. Li, D. Tao, Reversibility improved lossless data hiding, Signal Processing 89 (10) (2009) 2053–2065.

[19] X. Li, W. Zhang, X. Gui, B. Yang, A novel reversible data hiding scheme based on two-dimensional difference-histogram modification, Information Forensics and Security, IEEE Transactions on 8 (7) (2013) 1091–1100.

[20] S. Mousavi, A. Naghsh, S. Abu-Bakar, Watermarking techniques used in medical images: a survey, Journal of Digital Imaging 27 (6) (2014) 714–729.

[21] O. M. Al-Qershi, B. E. Khoo, High capacity data hiding schemes for medical images based on difference expansion, J. Syst. Softw. 84 (1) (2011) 105–112.

[22] K.-H. Chiang, K.-C. Chang-Chien, R.-F. Chang, H.-Y. Yen, Tamper detection and restoring system for medical images using wavelet-based reversible data embedding, Journal of Digital Imaging 21 (1) (2008) 77–90.

[23] C. Tan, J. Ng, X. Xu, C. Poh, Y. Guan, K. Sheah, Security protection of dicom medical images using dual-layer reversible watermarking with tamper detection capability, Journal of Digital Imaging 24 (3) (2011) 528–540.

[24] J. A. Rowlands, The physics of computed radiography, Physics in Medicine and Biology 47 (23) (2002) 123–166.

[25] Imebra C++ DICOM SDK library, http://imebra.com.