Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-18539-8

High capacity reversible data hiding in radiographic images with optimal bit allocation

Davide Cavagnino[1] · Alessandro Druetto[1] · Marco Grangetto[1] · Maurizio Lucenteforte[1]

Received: 7 August 2023 / Revised: 21 December 2023 / Accepted: 29 January 2024

Abstract
This paper extends and improves the performance of a digital reversible watermarking algorithm based on histogram shifting presented in previous works. The considered algorithm exploits the property of image histograms of some kinds of medical images which present many contiguous 0-runs, i.e., a comb structure in the gray level frequencies. In particular, radiographic images exhibit this structure after contrast enhancement during the acquisition process. The previous work suggested performing gray-level histogram shifting according to a local optimization technique. In this paper, we apply combinatorial optimization techniques to entire blocks of contiguous 0-runs using a non-linear objective function transformed to fit a linear optimization algorithm. The obtained results show a meaningful improvement in the payload capacity of the original data-hiding method. A mild Peak Signal-to-Noise Ratio (PSNR) reduction is still acceptable for a qualitative preview of the images, which can be completely restored to their original cover form thanks to the reversibility of the method.

Keywords Authentication · Data hiding · Watermarking · Zero-Run encoding

Abbreviations

- bpp : bits-per-pixel
- DCT : Discrete Cosine Transform
- DPM : difference-pair-mapping

✉ Alessandro Druetto
alessandro.druetto@unito.it

Davide Cavagnino
davide.cavagnino@unito.it

Marco Grangetto
marco.grangetto@unito.it

Maurizio Lucenteforte
maurizio.lucenteforte@unito.it

[1] Dipartimento di Informatica, Università di Torino, Corso Svizzera 185, Torino 10149, Italy

Published online: 28 February 2024

- HS : Histogram Shifting
- NC : Normalized Correlation
- PSNR : Peak Signal-to-Noise Ratio
- RDH : Reversible Data Hiding
- RDH-CE : reversible data hiding with contrast enhancement

1 Introduction

Data hiding in digital objects comprises a set of frameworks, methods and algorithms aimed at embedding information into digital audio, images, 3D models, text, etc. In case of covert communication between entities, the process is called steganography while embedding data for tamper detection, copyright protection, or extra data storing is called watermarking [1].

Data hiding algorithms modify a digital object, called cover object, to embed data and may be reversible or non-reversible: the choice depends on the application context because a reversible algorithm (see, for example, [2]) allows to restore the cover object after data extraction while a non-reversible one cannot [3]. In fact, some classes of applications, like those related to medical, legal or human safety fields, require that the end-user can have access to the cover object after the hidden data has been extracted.

Digital steganography deals with a set of methods and techniques to hide a message into a digital object in a manner that is not detectable in a subjective and objective way. This requirement means neither humans nor specific algorithms can distinguish between cover and stego objects. Good starting points on this topic are [1, 4].

On the other hand, digital watermarking aims at embedding a signal into a digital object, like an image, a video, a 3D model, etc. The reasons for storing data in digital objects are various, like copyright protection, tracking of origin, authentication, integrity protection, or simple data transfer. Depending on the objective of the methods, requirements for watermarking algorithms range from security and payload capability to imperceptibility, to fragility or robustness [1].

A large class of image watermarking algorithms operates on the intensity level histogram exploiting the image redundancy modifying pixel intensities to embed information called payload [5, 6].

Histogram shifting represents a family of methods that operate in the spatial domain by modifying pixel intensity levels in appropriate portions of the image histogram to insert payload data. Ideally, the parts to be used are those characterized by a high peak level next to a zero level. In general, the previous condition may be created by shifting by 1 several levels adjacent to the peak, keeping track of that shifting to recover the original image. In the work [7], a simple histogram shifting approach is applied, exploiting a characteristic of some images having highly populated histogram levels adjacent to one zero frequency level.

A more sophisticated histogram shifting approach was proposed in [8] and applied to a set of radiographic images where only a subset of gray levels is used, giving the histogram a comb structure where single non-zero frequency levels are surrounded by clusters of unused gray levels. This property is generally due to contrast enhancement operations during the acquisition process. In that work, the payload was embedded operating the level shifting considering a *local* optimization approach in order to maximize the payload size. Therefore, the primary goal of our study is to investigate if a meaningful payload improvement can be obtained taking into account the *global* characteristics of such histogram comb structure.

To this aim, in this paper we propose a global optimization procedure that considers all the contiguous zero-runs of levels, as explained in Section 3.

It should be noted that our approach does not modify the pixel levels to reserve space and the payload information is embedded for every shifted pixel value. With this assumption, our approach divides the image histogram into a sequence of sub-histograms (called blocks in the following) and applies to each of them an optimal shifting strategy that maximizes the total payload.

Thus, the contributions of this work can be summarized in the following points:

- develop a reversible data hiding scheme for medical images;
- exploit the intrinsic redundancy of a class of radiographic images, where contrast enhancement operations performed during the medical exam produce a histogram where many contiguous gray levels have a null frequency;
- optimize for the payload capacity using a combinatorial approach.

The obtained results show that the proposed approach has a better performance than the known algorithms in the literature. Hence, the presented work becomes the new reference for data embedding in images having the already mentioned characteristics.

The paper is organized as follows: the next section presents some related works and Section 3 discusses the proposed improvement over the method [8]. Experimental results are shown in Sections 4 and 5 draws some conclusions. The last section contains supporting material.

2 Related works

In [9] the authors revisit the Histogram Shifting (HS) technique and present a general framework to construct HS-based Reversible Data Hiding (RDH). By the proposed framework, one can get an RDH algorithm by simply designing the so-called shifting and embedding functions. One of the algorithms proposed in this paper reaches average PSNRs of 58.13 dB and 54.95 dB for payloads of 0.038 bpp and 0.076 bpp respectively, computed over a set of classical images (namely, Lena, Baboon, F-16, Peppers, Sailboat, Fishing boat; Baboon is not considered at 0.076 bpp given that the method is not able to reach this payload).

The paper [10] extends the concept of mono-dimensional histogram to two dimensions considering pairs of differences between pixels, then embeds data employing an injective difference-pair-mapping (DPM) that is an evolution of the shifting and expansion techniques applied in other RDH methods. By using a two-dimensional approach the pixel's spatial redundancy is used in a wiser manner and the performance of the resulting method is superior to the mono-dimensional based ones reaching payloads of 0.038 bpp with a PSNR of 58.53 dB and 0.076 bpp with a PSNR of 55.67 dB.

The work [11] performs HS in the quantized Discrete Cosine Transform (DCT) domain of image's JPEG coefficients: one bit of information is (reversibly) stored in embeddable pairs of quantized coefficients, thus performing two-dimensional HS in a transformed domain. The data extraction method recovers the original watermark and input image. The shown results, obtained with 6 typical test images (Lena, Sailboat, F-16, Baboon, Peppers, and Splash) indicate a maximum payload rate of 0.076 bpp and a related PSNR of around 47 dB.

Also, the method proposed in [12] modifies the histogram of the quantized DCT coefficients of JPEG encoded images. In particular, 8 × 8 subimages are sorted according to the number of zero-valued quantized AC coefficients and those having the higher number of zeroes are used to store the payload in the quantized AC coefficients having absolute value

1 or 2: in this case the method is non-reversible. The evaluation section, based on a set of 8 test images, reports a maximum payload rate of 0.11 bpp corresponding to a PSNR of 42.5 dB.

Kim et al. [13] present an embedding system which uses two estimates for every pixel leading to a couple of histograms that allow to reduce the distortion of the shifted pixels. The obtained skewed histograms are used to identify the pixels in the peak and in the short tails to use them for payload embedding: this allows for a reduction of the noise consequent to data storing. Experimental results were performed using three different image databases (USC-SIPI, Kodak, and BOSS v1-1) obtaining a maximum embedding payload of 0.305 bpp with a PSNR of 52.5 dB.

The method presented in [14] embeds additional data with HS reducing the invalid shifting of pixels, that is lessening the number of modified pixels that do not carry any payload. To this aim, the image is analyzed and only the regions with low-frequency content are selected for embedding. On a test image set composed by Lena, Baboon, Airplane, Boat, Elaine and Man, a maximum PSNR of just over 53 dB is obtained, corresponding to a payload rate of 0.14 bpp.

In the work [15] an optimal rule for pixel value modification is computed through a payload-distortion criterion method; this is obtained through the iterative computation of a transfer matrix fine-tuned on the image data. The resulting algorithm performs reversible data embedding storing in the payload the data to be encoded and the auxiliary information to be used for decoding and recovering the original image. Here, using the standard images Lena, Baboon, Plane and Lake, the results show an embedding capacity of slightly less 1.4 bpp and a PSNR of about 30 dB.

[16] presents a framework for reversible data hiding with contrast enhancement (RDH-CE). The method has two phases: in the first one the histogram bins counting fewer pixels are merged and the produced bins with 0 value are used for data embedding through histogram shifting; in the second phase the payload is increased using mean-square-error based embedding. Both steps iteratively compute a transfer matrix to improve the payload. The evaluation is based on a set of 8 test images, with a maximum payload of 1.11 bpp and a corresponding PSNR of 19.39 dB.

In [17] two-stage reversible data hiding is performed using a dynamic predictor called LASSO: bits are embedded expanding the prediction error and applying HS. Presented results on a standard set of 14 images report a maximum embedding rate of 0.84 bpp with a relative distorsion of about 38 dB. Also [18] aims at reducing distortion by improving the accuracy of the prediction and shrinking the histogram: moreover, areas of lower variance are preferred for embedding. Experimental results on a set of standard images including Baboon, Lena, Peppers, Elaine, Boat, Barbara and 1000 collected natural images show a maximum embedding capacity of about 0.78 bpp and a relative PSNR of about 42 dB.

A recent work [19] presents a reversible data hiding algorithm that, while increasing the contrast in the original image, artificially shifts the histogram to “create” free space that can accommodate the watermark. While the maximum reported capacity is 2 bpp, this procedure introduces a distortion in the original image. The maximum PSNR in the resulting embedded image is, in fact, not bigger than 34 dB. This approach is substantially different from our method: as already stated in Section 1, our approach does not modify the pixel levels to reserve extra space; this leads to a substantially higher PSNR because the distortion is only produced by the data hiding and not by other image pre-processing operations.

The work in [20] proposes a reversible watermarking scheme that is based on interpolation and histogram shift. An adaptive interpolation scheme doubles the horizontal and vertical dimensions of the input image, producing 3 non-seed pixels for each seed (original) pixel.

Payload bits are inserted using only non-seed pixels, thus ensuring the reversibility of the method. The experimental results show an embedding capacity up to 1 bpp on the input image, with a peak signal-to-noise ratio above 50 dB. It should be noted that the method requires handling a watermarked image that is 4 times larger than the original input image. Also, while the embedding is done in the *enlarged* 4× image, the evaluation of payload is done considering the dimension of the *original* image. In fact, the maximal embedding capacity should have been reported in relation to the dimension of the *enlarged* image, that is, the true recipient of the embedding. Compared to our method, the maximum embedding capacity reported above for [20] corresponds to a value of 0.25 bpp.

To the authors' knowledge, the newly proposed "combinatorial optimization" improvement with the development of a mathematical model was still not undertaken by the scientific community, so no related work in this direction can be found. Nevertheless, for some insights and references about the construction of our Mixed-Integer Linear Program, the book by Nemhauser and Wolsey [21] can be useful.

3 Proposed method

This section will recall the method developed in [8] and will present the proposed improvement.

Starting from a gray level image having a depth of n bit per pixel, it is derived the histogram $h(g)$ representing the frequency of each gray level g, with $0 \leq g \leq 2^n - 1$. Due to the processing performed on many X-ray images (for example, intensity transformation for contrast enhancement) the histogram of the resulting image may have many 0-valued bins, i.e. $h(l) = 0$ for a finite number of gray levels l: this characteristic may be exploited to reversibly store extra data in the image.

Following the definitions in [8] we call:

- 0-run $R = [s, t]$, with s and t integers in the range $\left[0, 2^n - 1\right]$: a closed interval of contiguous gray levels l having $h(l) = 0$. More formally, $h(l) = 0$, with $s \leq l \leq t$ AND $(h(s-1) > 0$ OR $s = 0)$ AND $(h(t+1) > 0$ OR $t = 2^n - 1)$;
- lower bound $L(R) = s - 1$ and upper bound $U(R) = t + 1$: the left and right limits of the 0-run $R = [s, t]$;
- markable block B: a maximal sequence of k 0-runs $\{R_0, R_1, \ldots, R_{k-1}\}$, where $U(R_{i-1}) = L(R_i)$, $1 \leq i < k$; maximal refers to the fact that no other 0-runs are contiguous to B, that is, $(h(L(R_0) - 1) > 0$ OR $L(R_0) < 1)$ AND $(h(U(R_{k-1}) + 1) > 0$ OR $U(R_{k-1}) > 2^n$ $-2)$.

It follows that the histogram of a gray-level image may contain zero or more blocks and that any block may be composed of one or more 0-runs.

The histogram bins that make up the lower and upper bounds of the 0-runs are called *markable levels* and may be used for embedding data in a reversible manner by shifting these levels towards the 0-valued bins. For a block B of k 0-runs we will denote the markable levels as $c_i = L(R_i)$, $0 \leq i < k$, and $c_k = U(R_{k-1})$.

For example, suppose that a single 0-run block has, as its bounds, $c_0 = L(R) = 11$ and $c_1 = U(R) = 16$. Through the assignment of the levels 12, 13 and 14 to c_0 it is possible to store 2 bits in every pixel valued 11 shifting its value according to the payload data; also, it is possible to assign the level 15 to 16 and store one bit of payload data in every pixel with value c_1. In this simple case, it is obvious that to maximize the payload capacity three levels must be assigned to the bound having the maximum histogram value, i.e. c_0 if $h(c_0) > h(c_1)$, c_1 otherwise. In Appendix C the computation for this case is presented.

A more complicated situation happens when a block is composed of more than one 0-run: in that case, an optimization procedure involving *all* the markable levels of the block is required.

As introduced in [8], considering a generic block of k 0-runs, the assignment of the zero-valued levels to the markable levels may be defined by means of *threshold levels* g_i, $0 \leq i < k$, one for every 0-run; in addition, to simplify the notation, it is convenient to use two more thresholds, $g_{-1} = L(R_0)$ and $g_k = U(R_{k-1}) + 1$, on the left and the right of the block respectively. A markable level c_i, $0 \leq i \leq k$, will be shifted inside the range $[g_{i-1}, g_i)$, i.e. $g_{i-1} \leq c_i < g_i$. By shifting a pixel with gray level c_i it will be possible to reversibly store $\lfloor \log_2(g_i - g_{i-1}) \rfloor$ bits, where $\lfloor x \rfloor$ represents the floor operation of largest integer not greater than x.

The objective is to maximize the number of bits of payload data by choosing optimal values for the threshold levels g_m, with $m = 0, \ldots, k-1$:

$$P = \sum_{i=0}^{k} \lfloor \log_2(g_i - g_{i-1}) \rfloor \, h(c_i)\,. \tag{1}$$

In [8] it was proposed a local approximation working on two 0-runs at a time, iterating many times on all the runs in the block: the formulas used in this local optimization are recalled in Appendix B. Being a local optimization based on continuous values for an integer domain the results obtained are not the absolute optimum.

Algorithm 1 Block payload capacity local optimization [8].
Input: Block of k 0-runs   Output: threshold levels g_i

```
Initialize $g_i$, with $-1 \leq i \leq k$
for Maximum number of iterations do
    for $i = 1, \ldots, k-1$ do
        Optimize threshold levels $g_{i-1}$ and $g_i$ according to Appendix B
    end for
end for
```

The local optimization version published in [8] considers a block of 0-runs (see, for example, Fig. 1) at a time and performs Algorithm 1. In particular, this algorithm cycles through pairs of contiguous 0-runs and optimizes the corresponding threshold levels g_{i-1}

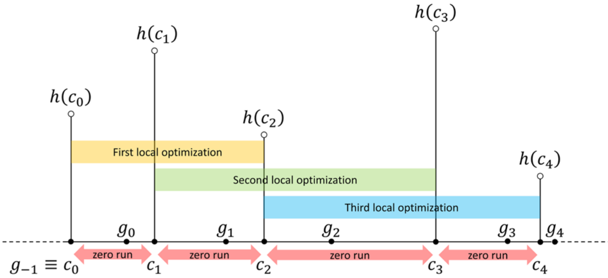


Fig. 1 Example of a block composed of four 0-runs

and g_i according to the formulas in Appendix B: the first pair considered is composed by the first and the second 0-runs, then the next pair processed is made up of the second and the third 0-runs, and so on until the last pair of the block is processed. After that, if a maximum number of iterations has been reached or the corresponding payload doesn't change the process is stopped and the obtained threshold levels are output; otherwise, another scan of 0-run pairs is begun from the first and second 0-runs.

The rationale behind Algorithm 1 is that performing a set of local optimizations that reciprocally influence their parameters will lead to a quasi-optimal solution. As this paper will show, a remarkable improvement in payload capacity may be obtained with an optimization that considers all the 0-runs of a block as a single optimization domain.

As mentioned above, Algorithm 1 is sub-optimal since it tries to find an estimate of the maximum value of the watermark capacity iterating a series of local optimizations on sets of two contiguous threshold levels. The process is repeated until a stable value or a maximum number of iterations is reached. Moreover, the local optimization we performed is based on the following continuous (convex) approximation of the capacity function,

$$\hat{P} = \sum_{i=0}^{k} \log_2 \left(g_i - g_{i-1}\right) \, h\left(c_i\right), \tag{2}$$

on which we can find (one of) the maximum value(s), using the formulas described in Appendix B.

This paper presents an optimization procedure that works simultaneously on all the 0-runs of a block to find the maximum of the payload defined in (1), and it shows the gain achieved on a set of images with respect to the local approach.

3.1 Block payload global optimization

The previous approach, as we stated, potentially gives a *sub-optimal* payload value, since it does a series of local optimizations. In the following, we apply combinatorial optimization techniques [21] to define a mathematical model that, by means of maximization of an objective function, is able to obtain the global and therefore *optimal* value for payload given a block of 0-runs.

As previously stated, pixel values are not modified to create histogram gaps for payload embedding; instead, the proposed algorithm exploits the characteristics of some medical images having large 0-runs in their histogram. As a consequence, for each shifted pixel value payload data is embedded.

The aim is to maximize the total payload value (1); instead of looking at small parts and doing a local optimization, our goal will be to look at the *global optimum* for a block of k consecutive 0-runs, only separated by non-zero singleton frequencies: the optimization will assign all g_i $(-1 \leq i \leq k)$ from c_0 (the location for g_{-1}) to $c_k + 1$ (the location for g_k).

The only constraint on this optimization problem, for now, is that the thresholds g_i must fall in the right places, thus

$$\begin{array}{rcl} & g_{-1} & = c_0 \\ c_0 < & g_0 & \leq c_1 \\ c_1 < & g_1 & \leq c_2 \\ \dots & \dots & \dots \\ c_{k-1} < & g_{k-1} & \leq c_k \\ c_k + 1 = & g_k & \end{array} \qquad \left\{ \begin{array}{rcll} & g_{-1} & = c_0 & \\ c_i < & g_i & \leq c_{i+1} & \forall i \in \{0, \dots, k-1\} \\ & g_k & = c_k + 1 & \end{array} \right. \tag{3}$$

will enforce a correct positioning for all of them.

Since c_i and $h(c_i)$ are part of the data extracted from the image, the only variables in this problem are the g_i positions, and those must obviously be integer numbers satisfying the constraints in (3). This said, the model should present itself as follows.

$$\text{maximize} \sum_i \left\lfloor \log_2(g_i - g_{i-1}) \right\rfloor \cdot h(c_i) \tag{4}$$

subject to

$$\begin{aligned} g_{-1} &= c_0 \\ g_i &> c_i \\ g_i &\leq c_{i+1} \qquad\qquad \forall i \in \{0, \ldots, k-1\} \\ g_k &= c_k + 1 \end{aligned} \tag{5}$$

$$g_i \in \mathbb{Z}_0^+ \tag{6}$$

As it can be easily seen, this model is *linear* in the constraints, but *non-linear* in the objective function (it is, in fact, logarithmic). Written this way, the optimum cannot be easily found by solvers, but luckily, with the usage of some *linearization techniques*, we can modify this problem by means of additional variables and constraints to obtain a fully linear one.

To linearize the model, we start with a consideration: the maximum *consecutive mappable space* assigned to a peak level c_i between thresholds is the maximum possible distance between its neighbor peaks c_{i-1} and c_{i+1} (see Fig. 1), since the maximum possible distance between two consecutive thresholds g_{i-1} and g_i is obtained when $g_{i-1} = c_{i-1} + 1$ and $g_i = c_{i+1}$. From this consideration, we can compute the maximum payload p for a block as the maximum *power of 2* not greater than this distance (the payload associated with a peak is the logarithm of the distance between two consecutive thresholds):

$$p = \left\lfloor \log_2 \max_{i \in \{1, k-1\}} (c_{i+1} - c_{i-1}) \right\rfloor; \tag{7}$$

this quantity is indeed a constant for every processed block of 0-runs, then it will be an optimization parameter of the model.

For a given peak c_i, we can say that it is associated with the space between g_{i-1} and g_i; then, c_i has $g_i - g_{i-1}$ gray levels mapped, $\forall i \in \{0, \ldots, k\}$ (see Fig. 1). It is worth noting that we care only about space as powers of 2, as it participates in the objective function as a logarithm, and that if we explicitly say that the i-th threshold has a space of 2^t levels mapped it cannot have a space of 2^{t-1}, or 2^{t+1} levels.

With these considerations in mind, the definition of a new auxiliary variable is required: we need a binary variable $x_{ij} \in \{0, 1\}$ that is equal to 1 if and only if the model assigns 2^j gray levels to threshold g_i and 0 otherwise, $\forall i \in \{0, \ldots, k\}$ and $\forall j \in \{0, \ldots, p\}$. Here comes at hand the value computed in (7): there will be only a limited number of variables spanning the *required* powers of 2. The following constraint, then, enforces that only one power of 2 will be assigned to every threshold:

$$\sum_j x_{ij} = 1 \qquad \forall i \in \{0, \ldots, k\}. \tag{8}$$

Now a link between the two variables g_i and x_{ij} must be made for the model. Remember that usable space 2^j must always fit between two consecutive thresholds g_{i-1} and g_i if the model decides to assign j bits to the i-th threshold (that is, if $x_{ij} = 1$). The following constraint enforces this requirement:

$$\sum_j x_{ij} \cdot 2^j \le g_i - g_{i-1} \qquad \forall i \in \{0, \dots, k\}. \tag{9}$$

The last thing that needs adjustment is the previously non-linear objective function: since we know which $x_{ij} = 1$, then there is no need to extract the logarithm from the mapped space; we have this information already encoded inside j. Thus, the only required computation is the product between j and $h(c_i)$ for all $x_{ij} = 1$:

$$\sum_{i,j} x_{ij} \cdot j \cdot h(c_i). \tag{10}$$

Constraints in (3) that enforce a correct positioning for the thresholds can be inherited from the previous model (4), (5), and (6) . We can now write the complete new model (11), (12), (13), and (14), obtained after the *linearization* of a non-linear objective function and used for the actual optimization with the solver.

$$\text{maximize} \sum_{i,j} x_{ij} \cdot j \cdot h(c_i) \tag{11}$$

subject to

$$\left.\begin{array}{l} \sum_j x_{ij} = 1 \\ \sum_j x_{ij} \cdot 2^j \le g_i - g_{i-1} \end{array}\right. \qquad \forall i \in \{0, \dots, k\} \tag{12}$$

$$\begin{array}{l} g_{-1} = c_0 \\ g_i > c_i \\ g_i \le c_{i+1} \\ g_k = c_k + 1 \end{array} \qquad \forall i \in \{0, \dots, k-1\} \tag{13}$$

$$x_{ij} \in \{0, 1\}, \quad g_i \in \mathbb{Z}_0^+ \tag{14}$$

The model has $\mathcal{O}(k)$ constraints and, after the linearization, $\mathcal{O}(kp)$ variables. Recall that k is the number of consecutive 0-runs present in the optimized block and p is the maximum power of 2 not greater than the distance between two consecutive thresholds.

Even if those quantities are pseudo-polynomial in the input size, we know the following. First, k is bounded to be less than *half* the length of the optimized block. Second, the absolute maximum value for p is 12 because of the image characteristics; in fact, all tested images have 4096 gray levels, hence are encoded with $n = 12$ bit depth.

Also, despite solving such a model results in an NP-hard problem, the optimal solution can be found very quickly in practice. This is true, especially thanks to the structure and the reduced size of the model itself.

3.2 Range assignment and encoding/decoding

After having determined the threshold levels g_i, to minimize the distortion due to payload embedding the shifts of every gray level c_i should be as near as possible to it: this requires that the effective range of size $2^{\lfloor \log_2(g_i - g_{i-1}) \rfloor}$ is centered on c_i, obviously with the constraint of being contained in the right open interval $[g_{i-1}, g_i)$. Algorithm 2 in pseudo-code shows how to define the range $[l_i, h_i]$ of possible shifts around c_i minimizing the mean squared error.

Algorithm 2 Compute range of possible shifts around c_i.
Input: c_i, g_{i-1}, g_i Output: l_i, h_i

```
numbits = $\lfloor \log_2 (g_i - g_{i-1}) \rfloor$
r = $2^{\text{numbits}}$
d = r / 2
$h_i$ = $c_i$ + d – 1
$l_i$ = $c_i$ – d
if $h_i$ >= $g_i$ then
    $h_i$ = $g_i$ – 1
    $l_i$ = $h_i$ – (r – 1)
else if $l_i$ < $g_{i-1}$ then
    $l_i$ = $g_{i-1}$
    $h_i$ = $l_i$ + (r – 1)
end if
```

A bit string of $\lfloor \log_2 (g_i - g_{i-1}) \rfloor$ bits with value v, $0 \leq v < 2^{\lfloor \log_2 (g_i - g_{i-1}) \rfloor}$, will be encoded by substituting the pixel value c_i with $l_i + v$.

On the decoding side a receiver aware of the sequence of markable levels and threshold levels can compute the range $[l_i, h_i]$ for every c_i: if a pixel in the encoded image has value q in the range $[l_i, h_i]$ then it can be restored to its original value c_i and $\lfloor \log_2 (g_i - g_{i-1}) \rfloor$ bits representing $v = q - l_i$ can be extracted as payload.

4 Experimental results

This section presents a comparison between the proposed method and [8] on which it builds upon. In particular, the objective metrics Peak Signal-to-Noise Ratio (PSNR in dB) and payload in bits-per-pixel (bpp) are evaluated on a set of 100 radiographic images. Moreover, we also compared with two works in the literature, namely [10] and [13].

All the experiments are worked out on a dataset 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, respectively. In particular, the dataset consists of 86 images acquired with Kodak DirectView DR 5100 and DR 3000 systems, and 14 images captured by Kodak CR 260, CR 975 and ELITE systems.

The software implementing the tested algorithms is written in MATLAB® and is run on a machine with processor Intel® Core™ i9 CPU @ 2.80GHz and 32 GByte RAM. The optimization procedure, handled by IBM® ILOG® CPLEX® solver, is directly called from the MATLAB® script.

From the results presented in Table 1 it may be observed that the developed method has a gain of 14% in payload with respect to [8]: this is reasonable because the proposed optimization obtains the best payload using histogram shifting in the 0-runs. Also, the payload is one order of magnitude greater than those of the other two methods used for comparison. One limitation of our method is a slight reduction in PSNR: this is a direct and well paid back consequence of the increased payload (even if, being the method reversible, the PSNR metric is not so important, and it just grants that the watermarked image is useful as a visual preview). As a side note, the values of PSNR above 70 dB indicate that the processed images are visually indistinguishable by a human from the original ones, thus we do not show the watermarked images. Moreover, the proposed algorithm has an increased complexity with

Table 1 Comparison of quality (PSNR) and performance (pixel payload) parameters for various data hiding methods applied to a set of 100 images, average (avg) and standard deviation (std) values

Method	PSNR [dB] (avg ± std)	Payload [bpp] (avg ± std)
Proposed	71.60 ± 3.13	1.14 ± 0.27
Cavagnino et al. [8]	72.72 ± 3.44	1.00 ± 0.24
Li et al. [10]	75.54 ± 0.55	0.04 ± 0.04
Kim et al. [13]	76.59 ± 0.66	0.08 ± 0.09

respect to [8] due to the optimization that finds the global optimum bit allocation; nonetheless, note that the decoding and extraction time is the same as [8] and is negligible.

In Fig. 2 are reported two graphs: the first one plots the payload (in bpp) of the compared algorithms for each of the 100 images; the second graph shows for each image the corresponding PSNR (in dB). As it can be seen the proposed algorithm outperforms the others, in particular the payload is always not less than the one of [8] and larger than those of [10] and [13]. The disadvantage of the present proposal is shown in the PSNR graph where a lower quality of the embedded images is a consequence of the higher payload: as already said, the reversibility of the method makes this defect negligible.

The overall performance of the proposed algorithm can be qualitatively appreciated by observing Fig. 3 where the scatter plot of payload versus PSNR for the four compared

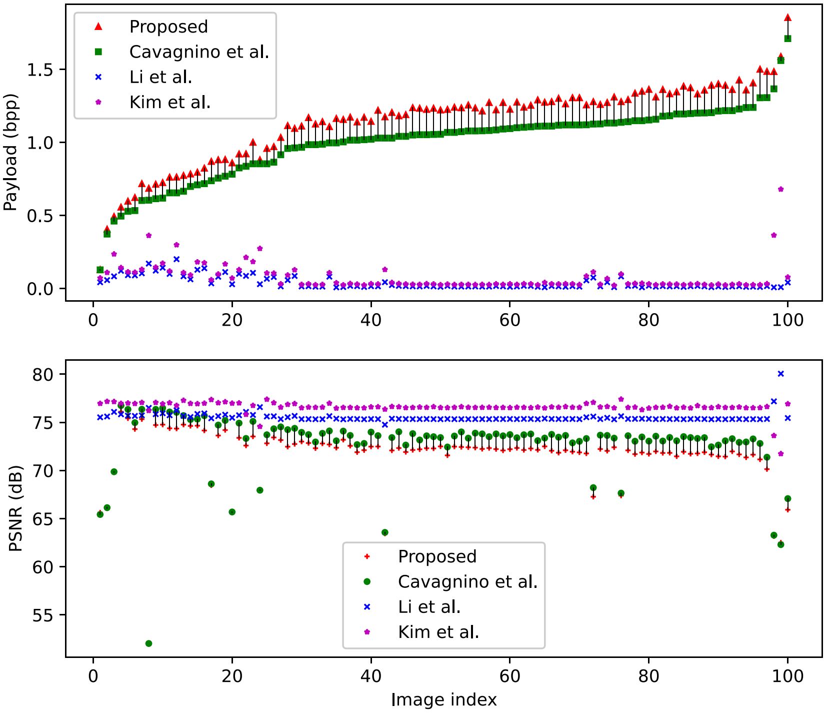


Fig. 2 Plot of payload and PSNR for Cavagnino et al. [8], Li et al. [10], Kim et al. [13] and the proposed method on a set of 100 images

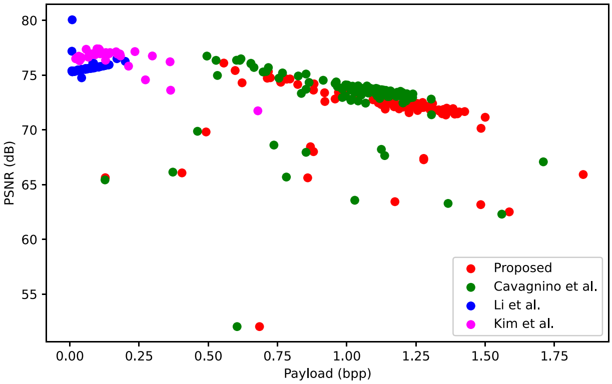


Fig. 3 Scatter plot of payload versus PSNR for Cavagnino et al. [8], Li et al. [10], Kim et al. [13] and the proposed method on a set of 100 images

algorithms applied to the image set is presented: the high concentration of points for the proposed algorithm in the upper-right part of the graph witnesses a high payload with an average good objective quality.

A visual example of the result of data embedding in a medical image is depicted in Fig. 4: Fig. 4a shows the cover image with a detail highlighted by a black box, which is magnified in Fig. 4b. The corresponding marked images with method [8] and with the proposed algorithm are not shown because differences are non-visually perceptible: in fact, the corresponding PSNRs are over 71 dB, with payloads of 0.83 bpp and 1.13 bpp, respectively.

Visual results of embedding with [8] and with the proposed algorithm are shown respectively in Fig. 4c and d: since the reduced error of embedding, to make it visually noticeable we have magnified the error respect to the original image by a factor 500. Analogous results are reported in Fig. 5 for another image belonging to the dataset we tested, where we obtained a PSNR of more than 52 dB with payloads of 0.604 and 0.685 bpp respectively.

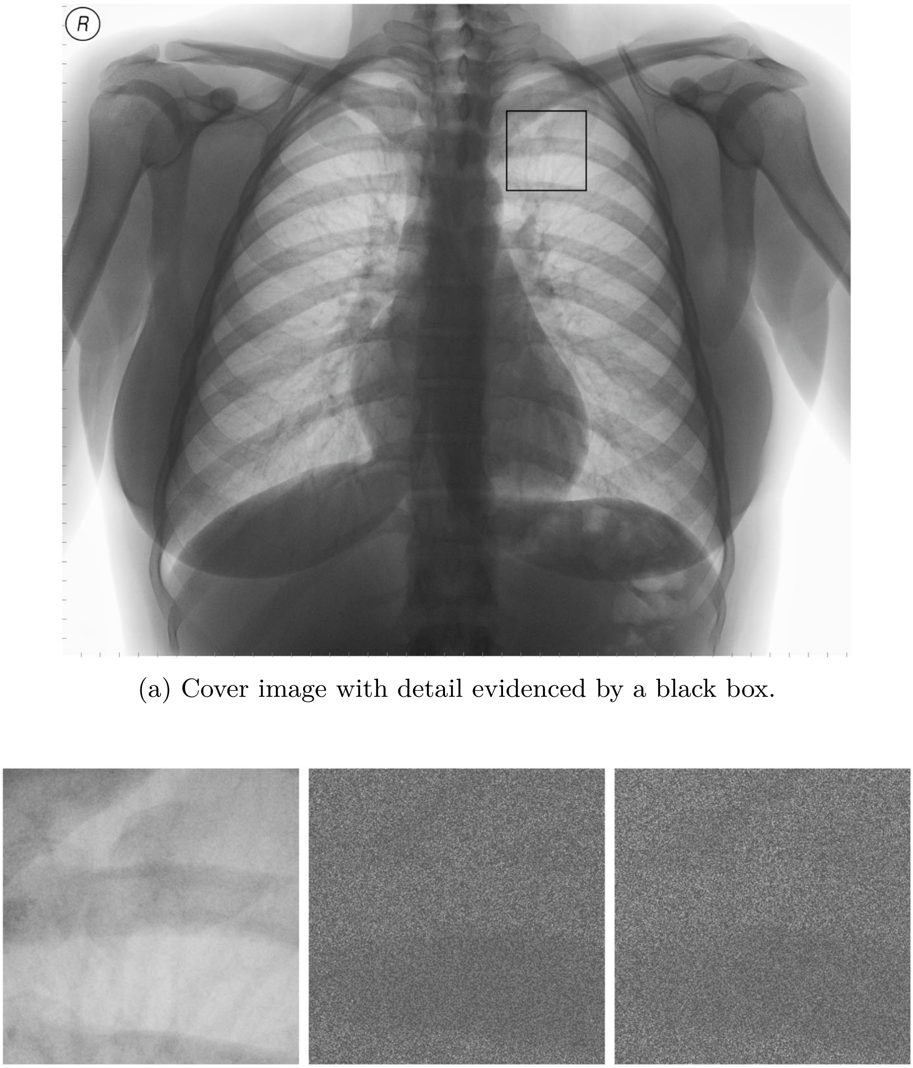
As another metric for the quality of the watermarked image, we also evaluated the Normalized Correlation (NC) as defined in [22]. Between the cover image and the watermarked image, the NC value is always very close to 1, about $1 - 1 \cdot 10^{-7}$.

5 Conclusions

In this paper, we presented a global optimization procedure with the aim to increase the payload size of a reversible data hiding algorithm for images having histograms containing many contiguous runs of zeros, commonly found in a specific class of medical images that are optimized for visual inspection purposes.

The proposed approach is a substantial improvement of the works [7, 8]. In particular, the most recent one performs a set of local optimization steps on pairs of contiguous runs. Implementing a global optimization involving all contiguous runs, as we present in this work, led to an average increase of 14% of payload size.

On the other side, the increased payload leads to a reduction of PSNR of 1 dB with respect to [8], which on the image set used in the experiments leads to 71 dB. This reduction is acceptable since the images are used for diagnosis purposes only after recovering the cover images thanks to the reversibility of our method.



(a) Cover image with detail evidenced by a black box.

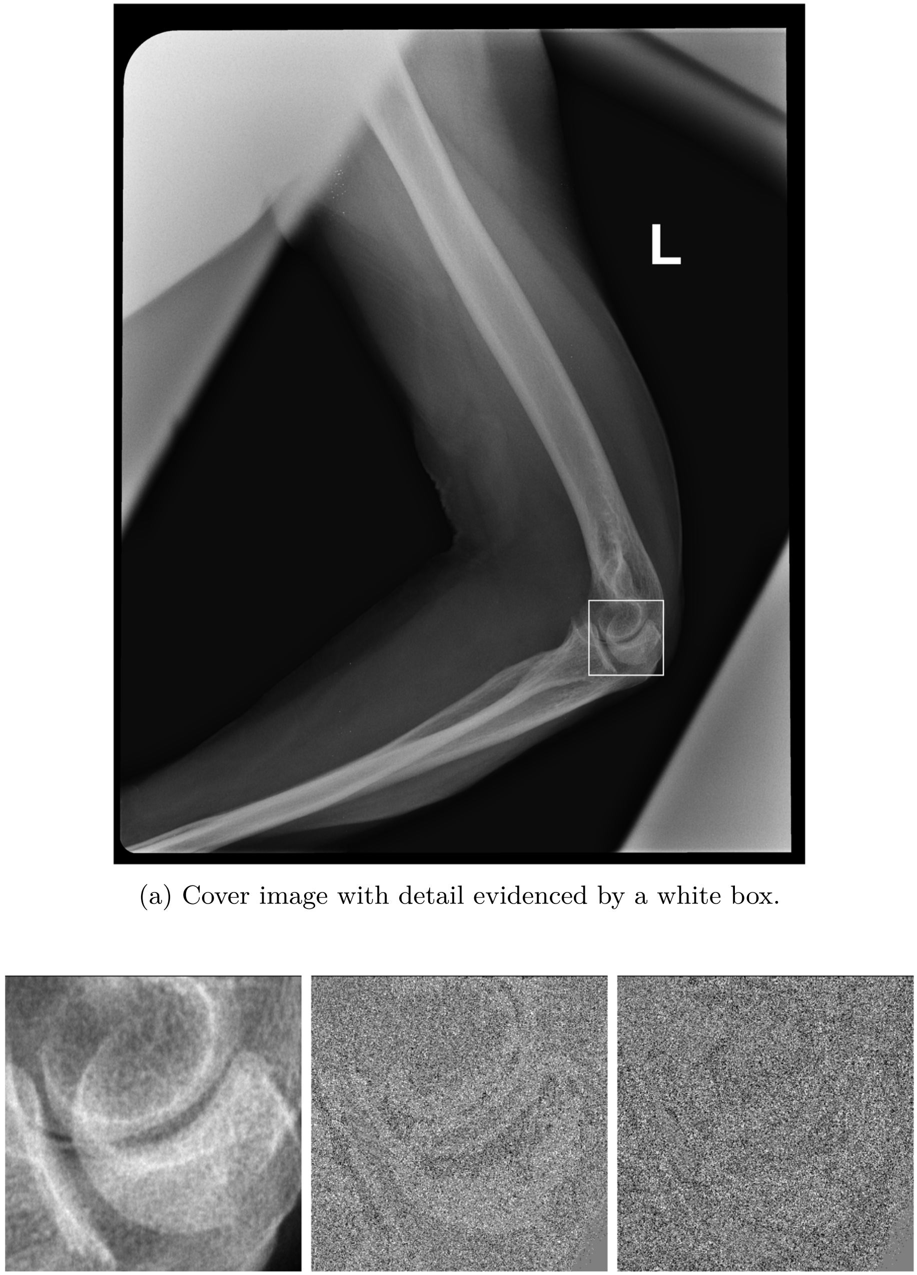
(b) Cover image detail.

(c) Detail of difference between cover and image marked by [8].

(d) Detail of difference between cover and image marked by the proposed algorithm.

Fig. 4 Visual results of embedding (difference values of Fig. 4c and d are magnified by a factor 500)

A limitation of this approach is its applicability restricted to images that exhibit a histogram having a comb structure, where single non-zero frequency levels are surrounded by clusters of unused gray levels. This typically happens in radiographic images, and it is generally due to contrast enhancement operations during the acquisition process.



(a) Cover image with detail evidenced by a white box.

(b) Cover image detail.

(c) Detail of difference between cover and image marked by [8].

(d) Detail of difference between cover and image marked by the proposed algorithm.

Fig. 5 Visual results of embedding (difference values of Fig. 5c and d are magnified by a factor 500)

We plan to study if and how this method can be extended to cope with histograms with different recognizable patterns of 0-runs, for example when a single non-zero frequency gray level is replaced with a cluster of several non-zero levels.

In conclusion, the presented results prove that the suggested approach has better performances than existing algorithms found in the literature. Therefore, the proposed work provides a new benchmark for hiding data in images exhibiting the previously mentioned histogram structure.

Appendix

A Payload overhead computation

This subsection is devoted to evaluating the bit overhead required to encode the subsidiary information expressing the positions of the various blocks and runs.

Suppose to have an image with l gray levels. The maximum possible number of blocks happens in the case of contiguous blocks, each one made up of a single 0-run of single 0 frequency level; considering the possibilities of having blocks beginning and ending with a 0 frequency level at gray levels 0 and $2^l - 1$ the maximum number of blocks is $\left\lfloor \frac{l-4}{3} \right\rfloor + 2$. The number of bits required to represent this number is $\log_2 \left(\left\lfloor \frac{l-4}{3} \right\rfloor + 2 \right)$ (for coding simplicity do not consider the fact that 0 blocks is not possible because in that case the image would be useless for the proposed algorithm).

Thus, the number of bits n_b required to represent the number of blocks is upper limited by $\log_2 l$.

The maximum number r of 0-runs in a block can be $l/2$ (consider an image having alternate 0 and non-zero gray level frequencies, i.e., a single block for an image): the maximum number of bits required to represent this number is $\log_2 (l/2) = \log_2 l - 1$ which is upper limited by $n_s = \log_2 l$.

For every block must be encoded the initial spike position, c_0, requiring $\log_2 l$ bits and the following pairs (g_i, c_i), each requiring $2 \log_2 l$ bits (in case of fixed length encoding).

Summing up, the side information required to code a block composed of r 0-runs is $n_s + \log_2 l + 2\,r \log_2 l = (2\,r + 2) \log_2 l$ bits.

B Expression derivation for local optimization

Recalling Appendix A from [8], the payload carried by a triple of levels $\left[c_{i-1}, c_i, c_{i+1}\right]$ is

$$P = \sum_{j=i-1}^{i+1} \left\lfloor \log_2 \left(g_j - g_{j-1} \right) \right\rfloor h \left(c_j \right) \tag{15}$$

where $[g_{i-1}, g_i)$ is the right open interval of c_i, i.e., $g_{i-1} \leq c_i < g_i$.

This payload can be maximized for g_{i-1}, g_i with the following constraints:

$$g_{i-1} > c_{i-1} \tag{16}$$

$$g_i \leq c_{i+1}. \tag{17}$$

Introducing a continuous function approximating (15)

$$\tilde{P} = \sum_{j=i-1}^{i+1} \log_2 \left(g_j - g_{j-1}\right) h\left(c_j\right) \tag{18}$$

the optimal values for the limits g_{i-1}, g_i may be found equating to 0 the first derivatives of $\tilde{P}$ with respect to g_{i-1} and g_i.

$$\frac{\partial \tilde{P}}{\partial g_{i-1}} = \frac{1}{\ln 2}\left(\frac{h\left(c_{i-1}\right)}{g_{i-1} - g_{i-2}} - \frac{h\left(c_i\right)}{g_i - g_{i-1}}\right) \tag{19}$$

$$\frac{\partial \tilde{P}}{\partial g_i} = \frac{1}{\ln 2}\left(\frac{h\left(c_i\right)}{g_i - g_{i-1}} - \frac{h\left(c_{i+1}\right)}{g_{i+1} - g_i}\right) \tag{20}$$

$$\frac{\partial \tilde{P}}{\partial g_{i-1}} = \frac{\partial \tilde{P}}{\partial g_i} = 0$$

These equations lead to

$$\frac{h\left(c_{i-1}\right)}{g_{i-1} - g_{i-2}} = \frac{h\left(c_i\right)}{g_i - g_{i-1}} \tag{21}$$

and

$$\frac{h\left(c_i\right)}{g_i - g_{i-1}} = \frac{h\left(c_{i+1}\right)}{g_{i+1} - g_i}. \tag{22}$$

Starting from (21) it is possible to write

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

obtaining

$$g_i = \left(1 + \frac{h\left(c_i\right)}{h\left(c_{i-1}\right)}\right) g_{i-1} - \frac{h\left(c_i\right)}{h\left(c_{i-1}\right)} g_{i-2}. \tag{23}$$

Observing that the right hand side of (21) is equal to the left hand side of (22) it is possible to write

$$h\left(c_{i-1}\right)\left(g_{i+1} - g_i\right) = h\left(c_{i+1}\right)\left(g_{i-1} - g_{i-2}\right) \tag{24}$$

$$g_i = g_{i+1} - \frac{h\left(c_{i+1}\right)}{h\left(c_{i-1}\right)}\left(g_{i-1} - g_{i-2}\right). \tag{25}$$

Substituting g_i from (23) in (25)

$$\begin{aligned}\left(1+\frac{h\left(c_i\right)}{h\left(c_{i-1}\right)}\right) g_{i-1}-\frac{h\left(c_i\right)}{h\left(c_{i-1}\right)} g_{i-2} &= g_{i+1}-\frac{h\left(c_{i+1}\right)}{h\left(c_{i-1}\right)}\left(g_{i-1}-g_{i-2}\right)\\ \left(1+\frac{h\left(c_i\right)}{h\left(c_{i-1}\right)}+\frac{h\left(c_{i+1}\right)}{h\left(c_{i-1}\right)}\right) g_{i-1} &= g_{i+1}+\left(\frac{h\left(c_i\right)}{h\left(c_{i-1}\right)}+\frac{h\left(c_{i+1}\right)}{h\left(c_{i-1}\right)}\right) g_{i-2}\\ \frac{h\left(c_{i-1}\right)+h\left(c_i\right)+h\left(c_{i+1}\right)}{h\left(c_{i-1}\right)} g_{i-1} &= \frac{h\left(c_{i-1}\right) g_{i+1}+\left(h\left(c_i\right)+h\left(c_{i+1}\right)\right) g_{i-2}}{h\left(c_{i-1}\right)}\\ g_{i-1} &= \frac{h\left(c_{i-1}\right) g_{i+1}+\left(h\left(c_i\right)+h\left(c_{i+1}\right)\right) g_{i-2}}{h\left(c_{i-1}\right)+h\left(c_i\right)+h\left(c_{i+1}\right)}. \qquad (26)\end{aligned}$$

Equations (23) and (26) report the values that maximize $\tilde{P}$.

When operating on the two boundaries of the histogram there may be gray level distributions where the algorithm has to assign the value 0 to $h\left(c_{i-1}\right)$. To deal with this case other expressions of g_{i-1} and g_i should be used.

Using (22) it is possible to write

$$\begin{gathered}h\left(c_i\right)\left(g_{i+1}-g_i\right)=h\left(c_{i+1}\right)\left(g_i-g_{i-1}\right)\\ h\left(c_{i+1}\right) g_{i-1}=h\left(c_{i+1}\right) g_i-h\left(c_i\right)\left(g_{i+1}-g_i\right)\end{gathered}$$

determining

$$g_{i-1}=\left(1+\frac{h\left(c_i\right)}{h\left(c_{i+1}\right)}\right) g_i-\frac{h\left(c_i\right)}{h\left(c_{i+1}\right)} g_{i+1}. \qquad (27)$$

From (24) it is also possible to compute g_{i-1}

$$g_{i-1}=\frac{h\left(c_{i-1}\right)}{h\left(c_{i+1}\right)}\left(g_{i+1}-g_i\right)+g_{i-2}. \qquad (28)$$

Equating (27) to (28) leads to

$$\begin{aligned}\left(1+\frac{h\left(c_i\right)}{h\left(c_{i+1}\right)}\right) g_i-\frac{h\left(c_i\right)}{h\left(c_{i+1}\right)} g_{i+1} &= \frac{h\left(c_{i-1}\right)}{h\left(c_{i+1}\right)}\left(g_{i+1}-g_i\right)+g_{i-2}\\ \left(1+\frac{h\left(c_i\right)}{h\left(c_{i+1}\right)}+\frac{h\left(c_{i-1}\right)}{h\left(c_{i+1}\right)}\right) g_i &= \left(\frac{h\left(c_{i-1}\right)}{h\left(c_{i+1}\right)}+\frac{h\left(c_i\right)}{h\left(c_{i+1}\right)}\right) g_{i+1}+g_{i-2}\\ g_i &= \frac{\left(h\left(c_{i-1}\right)+h\left(c_i\right)\right) g_{i+1}+h\left(c_{i+1}\right) g_{i-2}}{h\left(c_{i-1}\right)+h\left(c_i\right)+h\left(c_{i+1}\right)}. \qquad (29)\end{aligned}$$

Equations (27) and (29) allow the computation of g_{i-1} and g_i in case $h\left(c_{i-1}\right)$ is equal to 0.

C Single zero-run bit assignment

If a block is made of a single 0-run there is no optimization step to be performed because it is sufficient to assign all the possible bits to the level with the highest frequency and, if any, the remaining bits to the other level.

Let's call c_0 and c_1 ($c_0 < c_1$) the two levels delimiting the 0-run, h_{c_0} and h_{c_1} the respective histogram frequencies and let $g_{-1} = c_0$, $g_1 = c_1 + 1$.

The threshold g_0 dividing the levels assigned to c_0 from those for c_1 should be computed from the maximum number of bits $\left\lfloor \log_2\left(c_1 - c_0\right)\right\rfloor$ for the gray level, among c_0 and c_1, with the higher frequency: this requires $2^{\left\lfloor \log_2\left(c_1-c_0\right)\right\rfloor}$ gray levels leading to

- $g_0 = g_{-1} + 2^{\lfloor \log_2(c_1 - c_0) \rfloor}$ if $h_{c_0} > h_{c_1}$,
- $g_0 = g_1 - 2^{\lfloor \log_2(c_1 - c_0) \rfloor}$ otherwise (i.e., $h_{c_0} \leq h_{c_1}$).

The gray level c_0 will have assigned $\lfloor \log_2 (g_0 - c_0) \rfloor$ bits and c_1 will use $\lfloor \log_2 (c_1 - g_0) \rfloor$ bits.

Acknowledgements Not applicable.

Author Contributions All the authors of this paper equally contributed to its development and writing.

Funding Open access funding provided by Università degli Studi di Torino within the CRUI-CARE Agreement.

Availability of data and materials The anonymized datasets analyzed during the current study are not publicly available due to privacy reasons.

Declarations

Competing interests Not applicable.

References

1. Cox IJ, Miller ML, Bloom JA, Fridrich J, Kalker T (2007) Digital Watermarking and Steganography, Second Edition. Morgan Kaufmann
2. Zhang H, Sun S (2022) Meng F A high-capacity and reversible patient data hiding scheme for telemedicine. Biomed Signal Process Control 76:103706. https://doi.org/10.1016/j.bspc.2022.103706
3. Tang M, Zhou F (2022) A robust and secure watermarking algorithm based on DWT and SVD in the fractional order fourier transform domain. Array 15:100230. https://doi.org/10.1016/j.array.2022.100230
4. Abdulla AA (2015) Exploiting similarities between secret and cover images for improved embedding efficiency and security in digital steganography. Ph.D. thesis, University of Buckingham
5. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol 13(8):890–896. https://doi.org/10.1109/TCSVT.2003.815962
6. Ni Z, Shi Y-Q, Ansari N, Su W (2006) Reversible data hiding. IEEE Trans Circuits Syst Video Technol 16(3):354–362. https://doi.org/10.1109/TCSVT.2006.869964
7. Balossino N, Cavagnino D, Grangetto M, Lucenteforte M, Rabellino S (2013) A high capacity reversible data hiding scheme for radiographic images. Communications in Applied and Industrial Mathematics 4:1–14 https://hdl.handle.net/2318/140213
8. Cavagnino D, Lucenteforte M, Grangetto M (2015) High capacity reversible data hiding and content protection for radiographic images. Signal Process 117:258–269. https://doi.org/10.1016/j.sigpro.2015.05.020
9. Li X, Li B, Yang B, Zeng T (2013) General framework to histogram-shifting-based reversible data hiding. IEEE Trans Image Process 22(6):2181–2191. https://doi.org/10.1109/TIP.2013.2246179
10. Li X, Zhang W, Gui X, Yang B (2013) A novel reversible data hiding scheme based on two-dimensional difference-histogram modification. IEEE Trans Inf Forensics Secur 8(7):1091–1100. https://doi.org/10.1109/TIFS.2013.2261062

11. He B, Chen Y, Zhou Y, Wang Y, Chen Y (2022) A novel two-dimensional reversible data hiding scheme based on high-efficiency histogram shifting for JPEG images. Int J Distrib Sens Netw 18(3):1–14. https://doi.org/10.1177/15501329221084226
12. Li Y, Yao S, Yang K, Tan Y-A, Zhang Q (2019) A high-imperceptibility and histogram-shifting data hiding scheme for JPEG images. IEEE Access 7:73573–73582. https://doi.org/10.1109/ACCESS.2019.2920178
13. Kim S, Qu X, Sachnev V, Kim HJ (2019) Skewed histogram shifting for reversible data hiding using a pair of extreme predictions. IEEE Trans Circuits Syst Video Technol 29(11):3236–3246. https://doi.org/10.1109/TCSVT.2018.2878932
14. Jia Y, Yin Z, Zhang X, Luo Y (2019) Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting. Signal Process 163:238–246. https://doi.org/10.1016/j.sigpro.2019.05.020
15. Zhang X (2013) Reversible data hiding with optimal value transfer. IEEE Trans Multimedia 15(2):316–325. https://doi.org/10.1109/TMM.2012.2229262
16. Ying Q, Qian Z, Zhang X, Ye D (2019) Reversible data hiding with image enhancement using histogram shifting. IEEE Access 7:46506–46521. https://doi.org/10.1109/ACCESS.2019.2909560
17. Hwang HJ, Kim S, Kim HJ (2016) Reversible data hiding using least square predictor via the LASSO. EURASIP Journal on Image and Video Processing 2016:1–12. https://doi.org/10.1186/s13640-016-0144-3
18. Hung K-M, Yih C-H, Yeh C-H, Chen L-M (2020) A high capacity reversible data hiding through multi-directional gradient prediction, non-linear regression analysis and embedding selection. EURASIP Journal on Image and Video Processing 2020(1):1–20
19. Gao G et al (2021) Reversible data hiding with automatic contrast enhancement for medical images. Signal Process 178:107817
20. Ren F, Liu Y, Zhang X, Li Q (2023) Reversible information hiding scheme based on interpolation and histogram shift for medical images. Multimed Tools Appl 1–27
21. Nemhauser GL, Wolsey LA (1988) Integer and combinatorial optimization. John Wiley & Sons Inc
22. Alafandy K et al (2016) A comparative study for color systems used in the dct-dwt watermarking algorithm. Adv Sci Technol Eng Syst J 1:42–49 https://doi.org/10.25046/aj010508

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.