

Reversible watermarking using a perceptual model

Mohammad Awrangjeb
Mohan S. Kankanhalli

National University of Singapore

School of Computing

Department of Computer Science

Singapore 117543

E-mail: mohammal@comp.nus.edu.sg

Abstract. During data hiding, distortions are introduced in an original image because of quantization errors, bit replacement, or truncation at the gray-scale limit. These distortions are irreversible and visible, which is unacceptable in some applications such as medical imaging. However, the reversible watermarking technique overcomes this problem by retrieving the original image from the watermarked image. We present a novel reversible watermarking algorithm with a high embedding capacity considering the human visual system (HVS). We use the arithmetic coding technique to compress a part of the original image and store the compressed data together with necessary authentication information as the payload. The payload is then embedded within the original image with consideration of the HVS. Due to this, the watermarked image contains no perceptible artifacts. During the extraction phase, we extract the payload, restore the exact copy of the original image, and verify the authenticity. Experimental results show that our method provides a higher embedding capacity compared to the other algorithms proposed in the literature. © 2005 SPIE and IS&T.

[DOI: 10.1117/1.1877523]

1 Introduction

Content authentication of multimedia data such as images is becoming more and more important in the fields of law enforcement, medical imaging, astrophysics research, etc. Having the original image during analysis and diagnosis to make the right decision is of critical importance. Traditional watermarking techniques cannot provide adequate security and integrity for content authentication because of their irreversible nature. Reversible watermarking enables exact recovery of the original image by extracting the embedded information from the watermarked image; if the watermarked image is deemed to be authentic, that means no single bit of the watermarked image is changed after embedding the payload in the original image. Alternative terms for reversible watermarking are “lossless,” “distortion-free,” “invertible,” and “erasable” watermarking.

Celik *et al.* categorize lossless watermarking techniques in two classes.¹ The first is spread-spectrum based methods, where a spread spectrum signal corresponding to the information payload is superimposed (added) on the host signal

during embedding.^{2,3} The advantage of this class of algorithms is the increased robustness due to the use of the spread spectrum technique. However, they have the drawbacks of lower capacity as well as the presence of salt-and-pepper artifacts.¹ The second class is feature-based methods, where some features (portions) of the original image are replaced or overwritten with the watermark payload.^{1,4,5} The original portions of the image that will be replaced by the watermark payload are compressed and passed as a part of the embedded payload during embedding. The advantages of this class of algorithms are that they do not suffer from salt-and-pepper artifacts and offer high capacity. However, they are not robust like the algorithms of first category.¹

Almost all of the previously proposed algorithms have the common limitation of suffering from visible distortions. Although some of them offer high capacity, distortions become perceptually visible when the payload is increased.

The most important requirement of lossless watermarking is that any difference (distortion) between the original image and the watermarked image should be perceptually invisible. The human visual system (HVS) is the perceptual phenomenon that can be exploited to achieve this requirement. Until now none of the proposed lossless watermarking algorithms utilized the properties of the HVS. If the HVS is used, it basically provides the limit by which each pixel value of an image can be changed without making any perceptible difference to the image quality. This limit is called the just noticeable distortion or JND level. If we change the value of a pixel while keeping the distortion under the limit defined by its JND level, then the degradation of the watermarked image due to the change in the pixel value is imperceptible.

This paper presents an algorithm belonging to the second class (feature based) with a higher embedding capacity and imperceptible distortions. We choose a set of pixels in a suitable pixel range to be replaced by the watermark. We compress the chosen pixels using arithmetic coding.⁶ Experimental results show the proposed algorithm offers a higher capacity on the average than the existing algorithms. The artifacts are perceptually invisible because we consider the HVS while embedding the watermark in the original

Paper 03132 received September 23, 2003; revised manuscript received February 16, 2004; accepted for publication May 14, 2004; published online Mar. 30, 2005.
1017-9909/2005/\$22.00 © 2005 SPIE and IS&T.

image. The rest of the paper is organized as follows: Section 2 presents background, Sec. 3 details⁷ the HVS, and Sec. 4 presents a decimal-to- L -ary and L -ary-to-decimal conversion method. Section 5 presents our proposed algorithm. Section 6 describes the experimental results and comparisons between the proposed algorithm and the existing algorithms. Section 7 concludes with an analysis of the proposed algorithm.

2 Background

The aim of lossless watermarking is to develop a method that minimizes visible distortions while keeping the embedding capacity as high as possible. The method essentially should have the ability to recover the original image from the watermarked image if so required. From existing works, we can observe that if the embedding capacity is increased, distortions become large and visible. Celik *et al.* propose a simple reversible data-hiding algorithm having high embedding capacity.¹ It searches the whole image and finds the lower levels of pixel values to obtain enough space after compression. The main problem with this method is that all images do not offer enough features (pixel values) at low levels for lossless compression. Thus, as the embedding level becomes larger, distortions become larger and perceptually visible. Honsinger *et al.*² and Fridrich *et al.*³ use the spread spectrum approach to add the information payload to the host signal. These methods are robust to a wide range of distortions, but due to use of modulo arithmetic salt-and-pepper noises are introduced and the embedding capacity is low. The lossless compression and encryption of the bit plane by Fridrich *et al.* offers high embedding capacity with strength equivalent to the security offered by cryptographic methods.³ But noisy images and high payload force the embedding of the message in higher bit planes, and hence distortions become visible. The invertible authentication for JPEG images is a fast algorithm offering high embedding capacity, but artifacts become visible for JPEG images with a high-quality factor.³⁻⁵ The regular-singular (RS) lossless data embedding method divides the image into disjoint groups (regular, singular, and unusable).^{4,8,9} The whole image is scanned and RS groups are checked for the necessity to apply the flip operation while embedding information. Although this method involves low and invertible distortions, the capacity is not very high. Tian proposed a high-capacity watermarking algorithm based on difference expansion.^{10,11} This method does not involve any compression-decompression of selected image features, but there is a significant degradation of image quality due to bit replacement. Vleeschouwer *et al.* proposed an original circular interpretation of bijective transformations as a solution to fulfill all quality and functionality requirements of lossless watermarking.¹² The additive method proposed in his first work suffers from salt-and-pepper artifacts.¹³ Later Vleeschouwer *et al.* proposed a modification that solves this problem.^{12,14} This method helps to convey embedded message from a lossless environment to a lossy environment and offers high capacity, but the visual quality of the watermarked image is poor. This method offers high embedding capacity but distortions are easily visible if multiple or higher bit planes are used for embedding. Hongmei *et al.* proposed a high-capacity distortion-free data-hiding technique for a palette image.¹⁵ Ni *et al.* proposed a revers-

ible data-hiding technique based on histogram shifting.¹⁶ In this method, although the distortions are low, the embedding capacity is limited by the frequency of the most frequent pixel value in the image. Xuan *et al.* proposed a high-capacity lossless algorithm based on integer wavelet transform.¹⁷

The HVS has been extensively considered for perceptually acceptable compressed images.¹⁸⁻²⁰ Many image compression schemes, such as the JPEG standard,¹⁸ employ a block discrete cosine transform (DCT) followed by a uniform quantization. Acceptable rate-distortion performance depends on proper design of the quantization matrix.¹⁹ Ahumada and Peterson²⁰ provided a mathematical formulation for calculating perceptually lossless quantization matrix. Peterson *et al.*²¹ measured psychophysically the smallest coefficient, yielding a visible signal, as the amplitude-threshold for DCT basis functions. This model is called the image-independent perceptual (IIP) approach by Watson²² because it depends explicitly on detection thresholds for DCT basis functions and a single matrix is computed independent of any image. Later Peterson *et al.*²³ extended the value of the IIP approach by calculating thresholds under various conditions and by providing a formula that enables extrapolation to display luminance and pixel size. Watson²² provided an image-dependent perceptual (IDP) model that computes a quantization matrix optimized for a particular image. The IDP model eliminates the problems of IIP model: luminance masking, contrast masking, error pooling, and selectable quality.²² Podilchuk and Zeng²⁴ used this IDP model for watermarking of still images.

Previous HVS models can work well at a global level, since the quantization matrix cannot be changed within an image, and they do not provide a framework for spatial domain watermarking methods. The use of the dc coefficient of the DCT block in such cases, to cater for the luminance sensitivity, also does not guarantee a true representation as it is only the average of the block. It may work for blocks with low variance, but may deviate substantially in certain image blocks. To overcome these limitations, Kankanhalli *et al.*⁷ proposed a new way of analyzing the noise sensitivity of every pixel based on the local region image content, such as texture, edge, and luminance information. Local region content information was considered according to psychovisual studies by Johnston *et al.*²⁵ and Torres and Kunt.²⁶ This results in a JND mask for the image to be watermarked. When the mean value of the square of the noise is the same as that of the background, the noise tends to be most visible against a midgray background, assuming that the input image includes gamma correction.²⁷ The midgray regions have lower JND values as compared to the other regions.⁷

Since the method of Kankanhalli *et al.*⁷ can be used in the spatial domain, we consider using this model in our lossless watermarking technique in spatial domain and discuss this model briefly in the next section. An approach similar to that of Kankanhalli *et al.*⁷ was used for adaptive watermarking in the DCT domain.²⁸ HVS properties such as luminance masking, edge masking, and texture masking effects were exploited in this approach. It classifies a block into one of six perceptual classes: edge, uniform with moderate intensity, uniform with either high or low intensity,

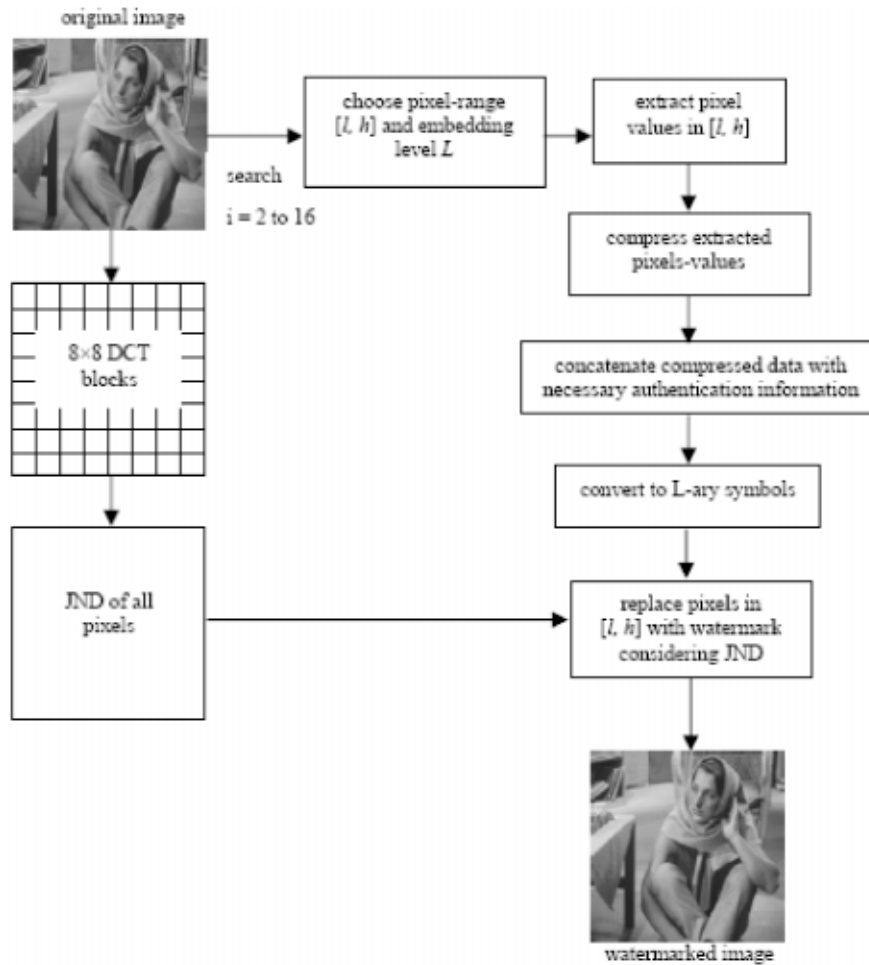


Fig. 1 Embedding process.

moderately busy, busy, and very busy, in descending order of noise sensitivity.²⁸

3 HVS

The HVS defines the limit of maximum allowable distortion for each pixel depending on the image quality and the pixel value. We use the HVS in our proposed algorithm to keep the distortions perceptually invisible. At the same time, we attempt to boost the embedding capacity as much as possible. We divide the original image into 8×8 blocks, and then take the DCT of each 8×8 block. In a block with abrupt changes between adjacent pixels, the signal energy tends to be concentrated in the ac coefficients. Equation (1) gives the energy in the ac coefficients E_{ac} , and Eq. (2) gives the maximum energy E_{max} . The maximum energy is found when the adjacent pixels have maximum and minimum permissible gray values. By using E_{max} as the normalization factor, we measure the roughness level R_b for block b [see Eq. (3)].

$$E_{ac} = \log \sum (x_{i,j}^2 - x_{0,0}^2) \quad 0 \leq i, j < 8, \quad (1)$$

$$E_{max} = \log \left(\frac{G}{2} \right)^2, \quad (2)$$

$$R_b = \frac{E_{ac}}{E_{max}}, \quad (3)$$

$$J(i, j, b) = \hat{J}_b + \left\lfloor \frac{(G/2 - P_{i,j,b})^2}{\alpha} \right\rfloor, \quad (4)$$

where $x_{i,j}$ is the (i, j) DCT coefficient of a block, G is the maximum permissible gray value (for an 8-bit image it is 255), $P_{i,j,b}$ is the pixel value at position (i, j) in block b , α is a precalculated constant, and we use the log for range compression. The range of R_b is uniformly divided into eight subgroups, and depending on the value of R_b , each block is given a block distortion index I_b , $I_b \in \{1, 2, \dots, 8\}$. Each block based on the corresponding I_b is mapped to a JND value \hat{J}_b , $\hat{J}_b \in \{3, 4, \dots, 10\}$, while $I_b \in \{1, 2, \dots, 8\}$. Now we consider pixel luminance to have the final JND $J(i, j, b)$ at any pixel position (i, j) in each block b [see Eq. (4)]. This

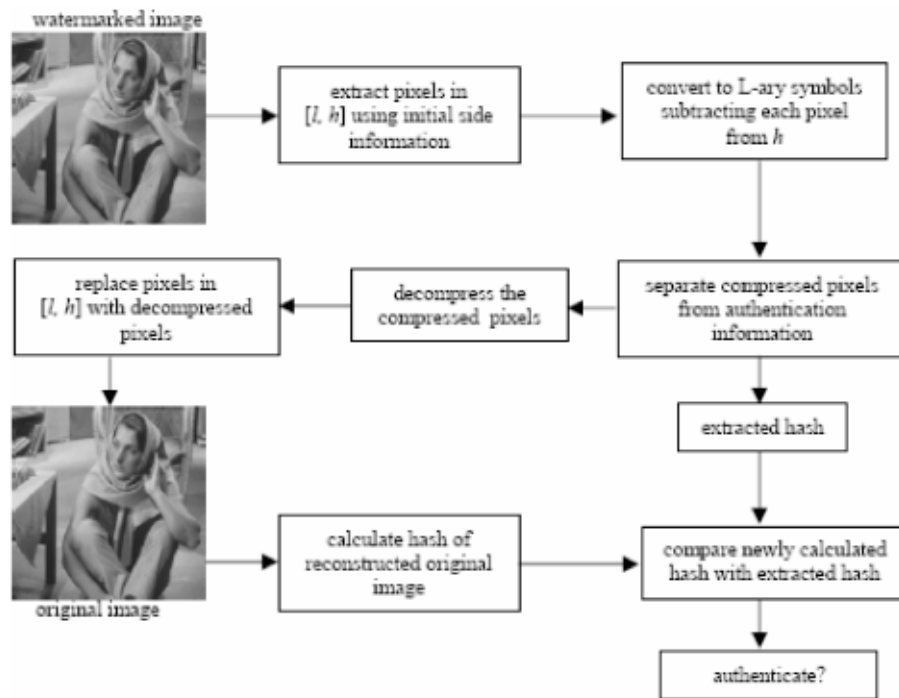


Fig. 2 Extraction process.

actually incorporates the effect of luminance into the JND value, since the distortion in an image is more noticeable in the midgray region. The distortion is less noticeable at the two ends of the gray scale because the sensitivity falls off parabolically toward both the ends.⁷ Note that a pixel at position (i, j) means the element at the $(i + 1)$ 'th row and the $(j + 1)$ 'th column in a 2-D array.

4 Number Conversion

We must perform a number conversion during the embedding and extraction processes. The compressed byte string (of extracted features concatenated with necessary information) is converted into L -ary symbols byte by byte during the embedding process and extracted L -ary symbols are converted into decimal numbers during the extraction process. Note that when numbers are represented as L -ary symbols, all digits or symbols in the representation are in the set $\{0, 1, 2, \dots, L - 1\}$. For example, 2-ary representation, known as binary number, of decimal number 190 is 10111110 and its 8-ary representation, known as octal number, is 276.

Let $C = c_1, c_2, \dots, c_n$ be the byte string (each c_i is an 8-bit unsigned integer) of concatenated information and compressed features. Steps for decimal-to- L -ary conversion are

1. If c_i is the current byte to convert, set $t = 1$.
2. $q_j = c_i / L$ and $r_j = c_i \bmod L$.
3. $w_j = r_j$ is the next watermark symbol, $t = t + 1$.
4. $c_i = q_j$.

5. If $c_i \neq 0$, go to step 2.
6. If $c_i = 0$ and $t < \lceil \log_L G \rceil$, next watermark symbol $w_j = L$ (to keep track for ease of extraction later).
7. If $c_i = 0$ go to step 1 for the next byte.

To convert L -ary symbols, $W = w_1, w_2, w_3, \dots, w_m$ to a decimal unsigned byte, these steps are followed:

1. If $c_i = 0$, $\text{exp} = 0$.
2. w_j is the current watermark symbol.
3. If $w_j \neq L$ and $\text{exp} < \lceil \log_L G \rceil$, do $c_i = c_i + w_j L^{\text{exp}}$, $\text{exp} = \text{exp} + 1$, go to step 2 for next watermark symbol.
4. Go to step 1 for next watermark symbol to decode next byte.

During encoding we convert C to W and during extraction we convert W to C . Note that G is the maximum permissible gray value; for an 8-bit image it is 255.

5 Proposed Algorithm

The proposed algorithm searches the image for a pixel range that offers the highest capacity with the lowest possible distortions. Lossless compression is performed on the pixel values in the chosen pixel range (say for embedding level $L = 4$ pixels, with values 224 to 227 at range $[224, 227]$) using arithmetic coding.⁶ The compressed bit string is concatenated with the authentication information and converted into L -ary symbols (each symbol value is in the range $[0, L - 1]$). We also calculate the JND values for all



Fig. 3 (a) “Airplane” (512×512), (b) “Lena” (512×512), (c) “Tiffany” (512×512), (d) “Jet” (512×512), (e) “Peppers” (512×512), (f) “Sailboat” (512×512), (g) “House” (512×512), (h) “CAT” (374×452), (i) “Man” (512×512), (j) “Lena2” (252×252), (k) “Boat” (512×512), (l) “Elaine” (512×512), (m) “Bridge” (512×512), and (n) “Barbara” (512×512).

pixels in the original image. The pixels in the chosen pixel range are then replaced considering the HVS. If the current watermark symbol is greater than the limit defined by the JND value of the current pixel (being considered to be replaced), the pixel is kept unchanged. Thus, when we complete the embedding process, no pixel is left having a distortion that is perceptually visible. For the sake of ease in the extraction process, we add some side information (unchanged pixel indices) with the watermark. Also we embed L -ary symbols of initial side information (embedding level, chosen pixel range, L -ary symbol length) with the first few pixels of the first block (from which pixels to be replaced are added to watermark) to initiate the extraction process.

5.1 Embedding Process

Figure 1 shows the embedding process. The steps involved are (1) calculate JND values of all pixels by dividing the image into 8×8 blocks, set initial embedding level value

$i=2$ that varies from 2 to 16, and (2) considering all values of i , determine the pixel range (by searching the whole image) for which the pixels at chosen range give largest space after compression. We do compression using arithmetic coding.⁶ While scanning the whole image we compress pixel values at the chosen pixel range depending on the value of i , and calculate the available capacity. Depending on the required capacity we keep the embedding level i as low as possible. This possible low value of i is the embedding level L and $[l, h]$ is the chosen pixel range, where $L = h - l + 1$. The third step is to calculate the hash (MD5 or SHA-1) of the original image,²⁹ concatenate it (including other necessary information as required) with the compressed bit string of pixels in the chosen pixel range, and convert the resultant bit string into L -ary symbols using the method given in Sec. 4. For example, for $L=2$ and an 8-bit gray scale image, 8-bit decimal numbers are converted to 8-bit binary numbers. In the fourth step, we consider the

HVS while replacing the pixels at chosen range: Let x_1 = decimal value of current pixel p , x_2 = decimal value of current L -ary symbol, and x_3 = JND value of pixel p . If $[x_1 - (h - x_2) \leq x_3]$, then replace the pixel p with $(h - x_2)$, else keep the pixel unchanged.

5.2 Extraction Process

The extraction process is just the inverse of the embedding process shown in Fig. 1, but we do not calculate the JND. We extract the watermark and retrieve the original image from the watermarked image as shown in Fig. 2. The embedded information is extracted from this chosen pixel range using the initial side information (such as pixel range $[l, h]$). The L -ary symbols are extracted by subtracting each pixel value in range $[l, h]$ from h . The resultant L -ary symbols are then converted into their original decimal values using the method given in Sec. 4. The compressed bit string of pixels is separated from authentication information and decompressed to replace the pixels at their original position.

6 Experimental Results and Comparisons

We first present here our experimental results in Sec. 6.1, and then in Sec. 6.2, we compare the results with those of other algorithms so far proposed.

6.1 Experimental Results

We implemented our algorithm and applied it to several images of different types: medical, astronomical, and general images. The images whose data are included here are shown in Figs. 3(a) to 3(n). We have used arithmetic coding for compression.⁶ Table 1 shows the experimental results for our proposed algorithm. All images are 8-bit gray scale. Average 1 shows the average capacity and peak SNR

Table 1 Experimental results of the proposed algorithm.

No	Image ^a	PSNR (dB)	Capacity (byte)
1	"Airplane"	54.64	6565
2	"Lenal"	58.30	2047
3	"Tiffany"	51.16	1906
4	"Jet"	48.24	25 218
5	"Peppers"	25.75	911
6	"Sailboat"	43.70	1761
7	"House"	52.84	3331
8	"CAT"	49.51	9650
9	"Man"	57.91	2239
10	"Lena2"	44.19	546
11	"Boat"	64.83	2714
12	"Elaine"	57.70	1689
13	"Bridge"	59.28	1733
14	"Barbara"	54.79	2261
Average 1 ^b		51.89	2873
Average 2		51.63	4469

^aSee Figs. 3(a) to 3(n).

^bExcluding the data of image "Jet" from the computation.

(PSNR) values excluding the data of image "Jet" [Fig. 3(d)]. Average 2 shows the average values including the data of image "Jet." The results in Table 1 show that the proposed algorithm offers reversible data hiding with a high capacity and imperceptible artifacts. The image "Jet" is a smooth image. Almost all lossless watermarking algorithms can embed information to such smooth image. We include the data here to compare it with other algorithms in the next section.

6.2 Comparisons with Other Algorithms

First we reveal the limitations of some of the recently proposed lossless watermarking algorithms. Lossless compression and encryption of bit plane method given by Fridrich *et al.*³ has these limitations:³ (1) capacity is limited and (2) the higher payload forces the method to embed watermark at a higher bit plane, and hence the artifacts are easily visible. Reversible data hiding method by Celik *et al.* gives high capacity.¹ But in this method, some images (especially small and textured ones) do not provide enough data for lossless compression. As a result, a larger embedding level is required, which leads to visible artifacts. For example, this method does not give enough capacity (at least 128 bit) for images 4, 5, 7, 8, and 12 in Table 1, while we use compression algorithms like arithmetic coding or Lempel-Ziv-Welch²⁹ (LZW), even if we increase the embedding level up to 50. Celik *et al.*¹ suggest using the context-based, adaptive, lossless image codec (CALIC) algorithm.³⁰ But if we increase the embedding level up to 50 for the image "Tiffany" [Fig. 3(c)] we get only 256 bytes of data. It is impossible to compress such a small amount of data to have enough space (at least 128 bits) to embed information. Moreover, the amount of maximum distortion is high (at most 50 for $L=50$), so artifacts are easily visible. Ni *et al.* proposed a method by shifting the histogram.¹⁶ In this method, the capacity is limited by the frequency of the most frequent gray level in the image. We include the experimental data provided in that paper for comparisons with results from our scheme (Table 2). This method offers a fixed PSNR 48.13 for all images because of its embedding criteria (shifting of histogram). The claim to

Table 2 Comparing capacity and PSNR offered by the proposed algorithm and Ni *et al.*¹⁶

No.	Image ^a (512×512)	Capacity (byte)		PSNR (dB)	
		Ni <i>et al.</i> ^b	Proposed	Ni <i>et al.</i> ^b	Proposed
1	"Airplane"	2022	6565	48.30	54.64
2	"Lenal"	683	2047	48.20	58.30
3	"Tiffany"	1098	1906	48.20	51.16
4	"Jet"	7498	25 218	48.70	48.24
5	"Peppers"	681	911	48.20	25.75
6	"Sailboat"	913	1761	48.20	43.70
7	"House"	1789	3331	48.30	52.84
Average 1 ^b		1198	2753	48.23	47.73
Average 2		2098	5962	48.30	48.80

^aSee Figs. 3(a) to 3(g).

^bExcluding the data of image "Jet" in the computation.

Table 3 Comparing capacity and PSNR offered by previous methods and our method.

No.	Methods	Capacity Embedded in 8-bit gray-scale Image of Size 512×512 (kbits)	PSNR (dB)
1	Macq and Dewey ³¹	less than 2.046	not available
2	Goljan <i>et al.</i> ⁹	3 to 24	35.00
3	Xuan <i>et al.</i> ¹⁷	15 to 94	24.00 to 36.00
4	Ni <i>et al.</i> ¹⁶	5 to 60	48.13
5	Ours ^a	7 to 202	48.80

^aAverage data from Table 2.

have 60k bits of embedding capacity using this method is not always achieved; it is only found for the image “Jet,” which is a particular case due to the peculiar characteristic of the image (being smooth). If we do not consider the “Jet” image, Table 2 shows that their method offers an average of 9.5k bits of embedding capacity with an average (constant) PSNR of 48.23 dB, whereas our proposed method provides an average embedding capacity 22k bits with average PSNR of 47.73 dB. If we included the “Jet” image, the average capacity offered by their method is 16.7k bits and that of our proposed algorithm it is 47.7k bits. All capacities including the average values shown in Tables 1 and 2 are in bytes. We show the comparison with methods by Goljan *et al.*,⁹ Ni *et al.*,¹⁶ Xuan *et al.*,¹⁷ and Macq and Deweyand³¹ in Table 3 where all capacities are given in kilobits (kbits).

The experimental results show that our proposed algorithm is better, since it enables us to achieve a higher capacity for almost all images and better PSNR compared to the other methods proposed so far. The most important feature of our proposed algorithm is the consideration of the HVS during embedding, and hence the artifacts are completely imperceptible. Thus, the proposed algorithm fulfills the requirements (invisible distortions and higher capacity) of lossless watermarking techniques and is superior to the other algorithms in this respect, as indicated by the experimental data.

7 Conclusion

We presented a lossless watermarking technique with a higher embedding capacity. By considering the HVS we leave no pixel in the watermarked image with a distortion greater than the maximum limit determined by the JND value of the pixel. Hence, there is no visible distortion at all in the watermarked image using our proposed algorithm. The experimental results prove that the proposed algorithm is better than the existing algorithms. Additional security against active attacks can be achieved by encrypting the watermark bit string with a secret key K before embedding. The proposed method also can be used for color images. It also works on both smooth and textured images.

References

1. M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, “Reversible data hiding,” in *Proc. IEEE Int. Conf. on Image Processing 2*, 157–160 (2002).
2. C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel, “Lossless recovery of an original image containing embedded data,” U.S. Patent No. 6278791 (2001).
3. J. Fridrich, M. Goljan, and D. Rui, “Invertible authentication,” in *Security and Watermarking of Multimedia Contents III, Proc. SPIE 3971*, 197–208 (2001).
4. J. Fridrich, M. Goljan, and D. Rui, “Lossless data embedding—new paradigm in digital watermarking,” *J. Appl. Signal Process.* **2**, 185–196 (2002).
5. J. Fridrich, M. Goljan, and D. Rui, “Invertible authentication watermark for JPEG images,” in *Proc. IEEE Information Technology: Coding and Computing*, pp. 223–227 (2001).
6. K. Sayood, *Introduction to Data Compression*, pp. 77–104, Morgan Kaufmann, San Francisco (2000).
7. M. S. Kankanhalli, Rajmohan, and K. R. Ramakrishnan, “Content-based watermarking of images,” in *Proc. 6th ACM Int. Multimedia Conf.*, pp. 61–70 (1998).
8. J. Fridrich, M. Goljan, and D. Rui, “Lossless data embedding for all image formats,” in *Security and Watermarking of Multimedia Contents IV, Proc. SPIE 4675*, 572–583 (2002).
9. M. Goljan, J. Fridrich, and D. Rui, “Distortion-free data embedding,” in *4th Information Hiding Workshop, Proc. LNCS 2137*, 27–41 (2001).
10. J. Tian, “Wavelet-based reversible watermarking for authentication,” in *Security and Watermarking of Multimedia Contents IV, Electronic Imaging, Proc. SPIE 4675*, 679–690 (2002).
11. J. Tian, “Reversible watermarking by difference expansion,” in *Proc. ACM Workshop on Multimedia and Security*, 19–22 (2002).
12. C. D. Vleeschouwer, J. F. Delaigle, and B. Macq, “Circular interpretation of histogram for reversible watermarking,” in *Proc. IEEE 4th Int. Workshop Multimedia Signal Processing*, 345–350 (2001).
13. B. Macq, “Lossless multiresolution transform for image authenticating watermarking,” in *Proc. European Association for Signal Processing Conference*, Tampere, Finland (2000).
14. C. D. Vleeschouwer, J. F. Delaigle, and B. Macq, “Circular interpretation of bijective transformations in lossless watermarking for media asset management,” *IEEE Trans. Multimed.* **5**, 97–105 (2003).
15. L. Hongmei, Z. Zhifeng, H. Jiwu, H. Xialing, and Y. Q. Shi, “A high capacity distortion-free data hiding algorithm for palette image,” in *Proc. IEEE Int. Symp. on Circuits and Systems 2*, 916–919 (2003).
16. Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” in *Proc. IEEE Int. Symp. on Circuits and Systems 2*, 912–915 (2003).
17. G. Xuan, J. Chen, J. Zhu, Y. Q. Shi, Z. Ni, and W. Su, “Lossless data hiding based on integer wavelet transform,” in *Proc. IEEE Int. Workshop on Multimedia Signal Processing*, 312–315 (2002).
18. G. Wallace, “The Joint Photographic Experts Group—still picture compression standard,” *IEEE Trans. Consum. Electron.* **38**(1), 18–34 (1992).
19. A. B. Watson, “Perceptual optimization of DCT color quantization matrices,” in *Proc. IEEE Int. Conf. on Image Processing 1*, 100–104 (1994).
20. A. J. Ahumada and H. R. Peterson, “Luminance-model-based DCT quantization for color image compression,” in *Human Vision, Visual Processing, and Digital Display III, Proc. SPIE 1666*, 365–374 (1992).
21. H. A. Peterson, H. Peng, J. H. Morgan, and W. B. Pennebaker, “Quantization of color image components in the DCT domain,” in *Human Vision, Visual Processing, and Digital Display II, Proc. SPIE 1453*, 210–222 (1991).
22. A. B. Watson, “DCT quantization matrices visually optimized for individual images,” in *Human Vision, Visual Processing, and Digital Display IV, Proc. SPIE 1913*, (1993).
23. H. A. Peterson, A. J. Ahumada, and A. B. Watson, “An improved detection model for DCT coefficient quantization,” in *Human Vision, Visual Processing, and Digital Display IV, Proc. SPIE 1913*, (1993).
24. C. I. Podilchuk and W. Zeng, “Perceptual watermarking of still images,” in *Proc. IEEE Multimedia Signal Processing*, pp. 363–368 (1997).
25. J. Johnston, N. Jayant, and R. Safranek, “Signal compression based on models of human perception,” *Proc. IEEE* **81**(10), 1385–1422 (1993).
26. L. Torres and M. Kunt, *Video Coding: The Second Generation Approach*, Kluwer, Boston, MA (1995).
27. A. N. Netravali and J. O. Limb, *Digital Pictures: Representation and Compression*, Plenum, New York (1998).
28. B. Tao and B. Dickinson, “Adaptive watermarking in the DCT domain,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing 4*, 2985 (1997).
29. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ (2002).

30. X. Wu, "Lossless compression of continuous-tone images via context selection, quantization, and modeling," *IEEE Trans. Image Process.* **6**(5), 656–664 (1997).
31. B. Macq and F. Deweyand, "Trusted headers for medical images," in *Proc. Deutsche Forschungsgemeinschaft Verteilte Verarbeitung und Vermittlung Digitaler Dokumente Watermarking Workshop*, Erlangen, Germany (1999).



Mohammad Awrangjeb received his BSc degree in computer science and engineering from Bangladesh University of Engineering and Technology (BUET) in 2002. He is currently pursuing his MS degree in computer science with the Department of Computer Science of the School of Computing at the National University of Singapore (NUS), where he is currently a research scholar. His research interests include watermarking, multimedia security,

and network security.



Mohan S. Kankanhalli received his BTech degree in electrical engineering from the Indian Institute of Technology, Kharagpur, in 1986 and his MS and PhD degrees in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, New York, in 1988 and 1990, respectively. He subsequently joined the Institute of Systems Science (ISS, now the Institute for Infocomm Research) in October 1990. He left ISS in June 1997 and spent the 1997 to 1998 academic year with the Department of Electrical Engineering of the Indian Institute of Science, Bangalore. He has been with the Department of Computer Science of the School of Computing at the National University of Singapore since May 1998. His research interests include multimedia information systems, information security, and digital video processing.