

Reversible Watermarking Scheme with Image-independent Embedding Capacity

Chang-Tsun Li

Department of Computer Science
University of Warwick
Coventry CV4 7AL, UK
+44 24 7657 3794
ctli@dcs.warwick.ac.uk

Abstract

Permanent distortion is one of the main drawbacks of all the irreversible watermarking schemes. Attempts to recover the original signal after the signal passing the authentication process are being made starting just a few years ago. Some common problems, such as salt-and-pepper artifacts due to intensity wraparound and low embedding capacity, can now be resolved. However, we point out in this work that there are still some significant problems remain unsolved. Firstly, the embedding capacity is signal-dependent, i.e., capacity varies significantly depending on the nature of the host signal. The direct impact of this ill factor is compromised security for signals with low capacity. Some signal may be even non-embeddable. Secondly, while seriously tackled in the irreversible watermarking schemes, the well-recognized problem of block-wise dependence, which opens a security gap for the vector quantization attack and transplantation are not addressed by the researchers of the reversible schemes. It is our intention in this work to propose a reversible watermarking scheme with near-constant signal-independent embedding capacity and immunity to the vector quantization attack and transplantation.

1. Introduction

Significant amount of effort has been put into the research of fragile watermarking methods for multimedia authentication in general and image authentication in particular. A common drawback of all the irreversible watermarking schemes [2, 10, 11, 16, 17] is permanent embedding distortion, which cannot be erased after the signal passing the authentication procedure so as to recover the original signal. Attempts to eliminate this problem have been made in the last few years [1, 4, 7, 9, 12-15]. Some common problems regarding reversible watermarking, such as salt-and-pepper artifacts due to intensity wraparound (e.g., intensity change from 0 to 255 or vice versa for 8-bit images) and low embedding capacity, can now be resolved [1, 7, 13-15]. However, we point out in this work that there are still some significant issues to be addressed.

Firstly, the embedding capacity is signal-dependent, i.e., capacity varies significantly depending on the nature of the host signal. The direct impact of this ill factor is compromised security for signals with low capacity. Some signal may be even non-embeddable. One way to approach this problem is to trade distortion for capacity by allowing more significant components of the signal (e.g., more significant bits of the image) to be watermarked [1, 4, 7, 13-15]. This approach is feasible provided that the distortion does not become noticeable. However, for the schemes with *signal-dependent* embedding capacity, there is no guarantee that the signal will be watermarked to a desired degree of capacity without inflicting noticeable distortion on it. One may argue that noticeable distortion is acceptable because the distortion will eventually be erased by the reversible scheme. However, if we rethink what makes watermarking differ from cryptography, we will realize that the requirement of low distortion should not be compromised

simply because the scheme is reversible. In the applications of copyright protection using digital watermarking, two key superior factors distinguishing watermarking from cryptography are

- Cryptography provides no further protection to the signal after decryption while watermarking does after watermark extraction because has been blended into the raw data.
- Cryptography scrambles the contents and masks the semantics of the signal while watermarking does not.

The first feature is important for applications of copyright protection because the emphases are the *robustness* and the very *existence* of the watermark in the host media in the *future*. On the other hand, in the applications of multimedia authentication and content integrity verification with *fragile* watermarking, which is the theme of this work, rather than being designed to survive the attacks, the fragile watermark is designed to be destroyed if attacked. That is to say that the emphases are the authenticity and integrity at the moment of authentication, not the robustness and existence of the watermark in the host media after the authentication procedure. This is also why we want to make the fragile watermarking scheme reversible so that the existence of the watermark can be *removed* after authentication. Thus, we can say that the first feature has less or no significance for reversible fragile watermarking scheme in the applications of authentication and the second factor is the one characterize the advantage of fragile watermarking over cryptography. However, when the watermark embedding distortion becomes noticeable, this superiority of fragile watermarking over cryptography becomes marginal. Therefore, ensuring balance between capacity and distortion is more important than simply seeking high capacity at the expense of high distortion. Unfortunately, finding this balance is not a trivial task for the schemes with *signal-dependent embedding capacity* [1, 4, 7, 13-15]. It is, thus, desirable to have

a scheme with *signal-independent embedding capacity*, which allows the user to specify a near-constant capacity and distortion in a reasonable range.

Secondly, while seriously tackled in the irreversible watermarking schemes, the well-recognized problem of block-wise dependence, which opens a security gap for vector quantization attack [16] (also known as the Holliman-Memon counterfeiting attack [8], birthday attack, or collage attack [6]) and transplantation attack [2,10,11] are not addressed by the researchers of the reversible schemes. Vector quantization attack is a malicious operation of collecting some image blocks from a large set / database of images watermarked with the same scheme to create a counterfeit or ‘collage’. By involving block-wise dependent or contextual information in the embedding procedure, vector quantization attack cannot succeed because placing watermarked blocks in the wrong context will not pass the authentication. Transplantation attack is another form of malicious operation of collecting blocks with *deterministic* dependence information (i.e., the dependence information is not calculated in a random but a deterministic manner) to create a counterfeit. The reader is referred to [6, 8, 16] and [2, 10, 11] for more information about vector quantization attack and transplantation attack, respectively.

It is our intention in this work to propose a reversible watermarking scheme with near-constant signal-independent embedding capacity and immunity to the vector quantization attack and transplantation attack.

2. Related Work

Barton [3] proposed one of the earliest reversible data embedding schemes, which compresses the bits to be affected by the embedding operation for two purposes: firstly preserving the original data and secondly creating space for the payload – the secret information to be hidden. The compressed data and the payload are then embedded into the host media. This practice of compressing original data for reversibility purpose has been widely adopted [1, 7, 13]. Honsinger et al. [9] employed reversible embedding for authentication application, which uses addition modulo 256 to overcome the problems of overflow and underflow due to embedding operation. However, apart from the embedding distortion, this modulo operation introduces salt-and-pepper artifacts because intensity close to zero are flipped / wraparound to 255 and the intensities close to 255 are mapped to 0. Another ill effect of this is that the scheme may not be able to extract the payload if the number of flipped pixels is too significant. The salt-and-pepper artifacts can also be found in Macq's method [12]. Schemes with high embedding capacity and without the salt-and-pepper artifacts have been reported in [1, 7, 12-15]. The main difference among these methods is that the methods of [1, 7, 13] employ compression technique for reserving the original data while the method of [14] 'clip' the intensities of some pixels before embedding payload in order to create intensity gaps for the payload. This is justifiable because of the fact that images captured by the acquisition systems are not 'perfect' representation of the real scene. The deviation of the clipped version from the 'perfect' version is not necessarily greater than the deviation of the captured one from the 'perfect' version.

Although steady progress in terms of high embedding capacity is being made, the capacity of all the afore-mentioned methods is highly sensitive to the nature of the images. Images with larger low-frequency areas tend to have higher capacity while the images with more

high-frequency areas tend to have lower capacity. Another common limitation of these methods is that the well-recognized requirement of establishing block-wise dependence for resisting vector quantisation attack and transplantation attack is not met.

3. Proposed Scheme

To eliminate those two common limitations of the afore-reviewed work, we propose a reversible watermarking scheme with *near-constant* and *image-independent* embedding capacity and immunity to the vector quantization attack and transplantation. Let us define some symbols as follows.

f : the original image with the grayscale of its i th pixel denoted as $f(i)$ and bit j of $f(i)$ denoted as

$f_j(i)$

f' : the image received by the watermark detector

w : the secret-key-generated watermark image of the same size as the original image f

w' : the extracted watermark image by the decoder

$h(f(i), w(i))$: the Hamming code of pixel i generated by performing Exclusive-OR operation on $f(i)$ and $w(i)$.

$D(f(i), w(i))$: the Hamming distance between $f(i)$ and $w(i)$

$N(i)$: the square dependence neighbourhood centered at pixel i of an image

$s(i)$: the secret non-deterministic dependence information of pixel i extracted from $N(i)$. This secret information is intended to counter the vector quantization attack [6, 8, 16] and transplantation attack [2,10,11]. This information can be the hash output, the sum of the

intensity, or some measure calculated in a random / non-deterministic manner. Specific design of $s(i)$ in this work will be detailed later. Now let us assume that $s(i)$ is available.

The basic idea behind the proposed work is to assign the pixels into a finite number of states characterized by some conditions so that only one-to-one transition from one state to another can be made. In the context of digital image watermarking, the state pixel i is in is determined by the intensity $f(i)$ or the transformed form of the intensity such as hamming code, the corresponding watermark pixel $w(i)$, and the secret information $s(i)$. The action of *forward* state transition is the operation of watermark *embedding* and the action of *backward* state transition is the operation of watermark *extraction*.

3.1. Observations on Hamming Code

In this work, we will map / transform the intensity of each pixel in an image into a Hamming code by performing an Exclusive-OR (XOR) operation on the intensity of the pixel $f(i)$ and a watermark pixel $w(i)$. Although the proposed scheme can be employed for watermarking color and grayscale images with arbitrary number of bits per pixel, without loss of generality, we will assume that we are working with 8-bit grayscale images throughout the rest of this work. Since there are 8 bits per pixel, if we allow one value of Hamming distance to represent one main state, then there will be 9 states, which can be denoted as D_k , $k \in [0,8]$ with D_0 standing for Hamming distance 0, D_1 for distance 1, and so on. Therefore, after the mapping, each pixel will be in one of the 9 main states. Taking state D_1 as an example, with 8 bit positions, there are 8 possible Hamming codes with distance 1, e.g., 00000001 and 00000010. So state D_1 can be partitioned into 8 sub-states, each corresponding to one hamming code. Each sub-state can be further divided into two sub-sub-states, *watermarkable* and *non-watermarkable*, depending on

$w(i)$ and $s(i)$. For the pixels of each watermarkable sub-sub-state, only one specific bit is taken as watermarkable *bit*. Reversible embedding is carried out by negating the watermarkable bit of the pixel to make a transition / mapping from the sub-sub-state of D_k to another of D_{k-1} if $\forall k, 0 < k \leq 4$ or from one of D_k to another of D_{k+1} if $\forall k, 4 \leq k < 8$. Note that because of the symmetry, which will become clear later after Table 1 is explained, pixels in sub-sub-states of D_4 are allowed to transit to the sub-sub-state of D_3 or D_5 depending on their Hamming codes. Since the state transition is one-to-one and the watermarkable bit of each sub-sub-state is specifically defined, at the verifier's side, when a pixel is detected as watermarked and passes the authentication, the scheme will be able to negate the watermarked bit to its original value, making a backward state transition to the original sub-sub-state. From now on, we will use the words 'state', 'sub-state', and 'sub-sub-state' interchangeably.

Note also that to make the reversible embedding possible, empty states must be in existence initially. This can be achieved by changing the grayscale of an insignificant proportion of pixels and use this pre-processed image as the original. This operation is similar to the intensity clipping frequently adopted by reversible watermarking schemes [5, 14]. Giving the fact that image acquisition systems are not perfect (e.g., a captured image is by no means a perfect representation of the real scene), sensible minute changes would make the pre-processed version close to the captured image or possibly even closer to the 'perfect' version. Moreover, since the key concern of the reversible watermarking scheme is the reversibility to the image before it is watermarked, not to any prior version(s), thus, provided that the effect of the pre-processing is insignificant in terms of the number of pixels affected and the amount of intensity changes, sensible pre-processing would be acceptable for the users.

For two 8-bit numbers, the total number of possible Hamming codes can be obtained is 256. The number of codes (or pixels) belonging to D_i is the number of combination of “choosing i from 8” denoted as $C(8,i)$. In this context, $C(8, 0) = C(8, 8) = 1$, $C(8, 1) = C(8, 7) = 8$, $C(8, 2) = C(8, 6) = 28$, $C(8, 3) = C(8, 5) = 56$, and $C(8, 4) = 70$. For any image, except the random noise images highly similar to the secret-key-generated random watermark image w which should not be deemed as images, the hamming codes created of the image f and w have the same statistical property, i.e. the number of pixels in state D_i is close the afore-mentioned figures. Thus, a natural step toward creating empty states would be negating bit 0 of the pixels with their Hamming distance equal to 0 or 8 because these pixels account statistically for only ($2/256 = 0.78\%$) of the total population. Another benefit of the proposed pre-processing, which will become clear later, is that *half* of those pixels mapped into the new states of D_1 and D_7 are watermarkable, thus, contributes to higher embedding capacity.

3.2 Algorithm Design

Based on the above framework, the proposed algorithm can be described as follows. First, a secret-key shared by the embedder and the verifier is used to generate a random 8-bit watermark image w of the same dimension as the host image, f . Secondly, an image, h , of the Hamming code for each pixel is created by performing an Exclusive-OR operation on the corresponding pixels of the host image f and the watermark image w . Pre-processing is then carried out by negating bit 0 of the pixels $f(i)$ with a Hamming distance of 0 or 8 so as to create empty states D_0 and D_8 . To maintain low embedding distortion, we do not watermark any bit of the pixels more significant than bit 3. (Note that the indices of the bits are in $[0, 7]$.) So a symbol system denoted as

$D_{k,h_3h_2h_1h_0}$ & $s_j \sim s_0 = w_j \sim w_0$ can be adopted for identifying the states of D_k , with $h_3h_2h_1h_0$ standing for the 4 least significant bits of a hamming code and & $s_j \sim s_0 = w_j \sim w_0$ specifying the condition that the $j+1$ least significant bits of the secret information $s(i)$ and watermark pixel $w(i)$ must be the same ($j < 8$). The underscored bit of $h_3h_2\underline{h_1}h_0$, is the *watermarkable bit*. Note that position of watermarkable bit varies from state to state. w_j , the most significant bit specified in the condition & $s_j \sim s_0 = w_j \sim w_0$, is the watermark bit to be embedded. Note that s_j and w_j are bit j of $s(i)$ and $w(i)$, respectively.

The *watermarkable* states characterized by the Hamming code, Hamming distance, and conditions are listed in Table 1. For example, an image pixel $f(i)$ is said to be in *watermarkable* state $D_{2,010\underline{1}}$ & $s_3 \sim s_0 = w_3 \sim w_0$ (the third column of the third row in Table 1) if their Hamming distance is 2, the 4 least significant bits of the Hamming code are 0101, and bit 0 to bit 3 of $s(i)$ and $w(i)$ are the same. The bit of $f(i)$ corresponding to the bit position underscored in the state symbol $D_{2,010\underline{1}}$ & $s_3 \sim s_0 = w_3 \sim w_0$ is the *watermarkable bit*. Watermarking a pixel is simply done by negating the watermarkable bit. This operation results in a forward state transition. For example, for any image pixel $f(i)$ in state $D_{2,010\underline{1}}$ & $s_3 \sim s_0 = w_3 \sim w_0$, to watermark it, bit **0** of $f(i)$ is negated (because $\underline{h_0}$ is underscored), resulting in a transition from state $D_{2,010\underline{1}}$ & $s_3 \sim s_0 = w_3 \sim w_0$ to state $D_{1,0\underline{1}00}$ & $s_2 \sim s_0 = w_2 \sim w_0$. Now it is clear that the arrow in each entry of Table 1 points to the *forward* state transition directions during watermark *embedding* process. To verify the received image at the verifier's side, when any *received* image pixel $f(i)$ in, for example, state $D_{1,0\underline{1}00}$ & $s_2 \sim s_0 = w_2 \sim w_0$ is encountered, the scheme will take s_3 as the *extracted* watermark bit w'_3 (i.e., let $w'_3 = s_3$.) Then if w'_3 equals the original watermark bit w_3 , $f(i)$ is deemed authentic and the original image pixel $f(i)$ can be recovered by simply negating bit **0** of $f(i)$, i.e., making a *backward* transition from state $D_{1,0\underline{1}00}$ & $s_2 \sim s_0 = w_2 \sim w_0$ to state $D_{2,010\underline{1}}$ & $s_3 \sim s_0 = w_3 \sim w_0$. If the image

is attacked, $s(i)$ would be different from its counterpart at the embedding side. In this case, assigning wrong value of s_j to w'_j results in a mismatch between w'_j and w_j (i.e., an alarm of attack).

One of the key features of the proposed scheme is its property of *near-constant* and *image-independent* embedding capacity, which is to be explained as follows. From the entry in the first column of the second row of Table 1 (i.e., $\mathbf{D}_{1,000\underline{1}}$ & $s_0=\mathbf{w}_0$), we know that 1/2 of the pixels with Hamming distance 1 and the 4 least significant bits equal to 0001 are watermarkable because only 1/2 of the pixels satisfy the condition $s_0=\mathbf{w}_0$. By the same token, only 1/4 of the pixels associated with $\mathbf{D}_{2,00\underline{11}}$ & $s_1s_0=\mathbf{w}_1\mathbf{w}_0$ are watermarkable because of condition $s_1s_0=\mathbf{w}_1\mathbf{w}_0$. The condition, $s_2\sim s_0=\mathbf{w}_2\sim\mathbf{w}_0$, in the first column of the fourth row indicates that only 1/8 of the pixels associated with $\mathbf{D}_{3,0\underline{111}}$ & $s_1s_0=\mathbf{w}_1\mathbf{w}_0$ are watermarkable, and so on. Since Table 1 is symmetrical about the **bold** line. The same property can be found in the lower part of the table. The benefit of the pre-processing becomes clear now. By negating bit 0 of the pixels with their Hamming distance equal to 0 or 8, those pixels transit to states $\mathbf{D}_{1,000\underline{1}}$ & $s_0=\mathbf{w}_0$ or $\mathbf{D}_{7,111\underline{0}}$ & $s_0=\mathbf{w}_0$, respectively, and, as just mentioned, 1/2 of them are watermarkable. The proportions of the pixels associated with $\mathbf{D}_{k,h_3h_2h_1h_0}$ & $s_j\sim s_0=\mathbf{w}_j\sim\mathbf{w}_0$, which are watermarkable, are listed in Table 2. The two values of 1/2 in parentheses in the first column remind us that half of the pixels in the two states are mapped from states of \mathbf{D}_0 and \mathbf{D}_8 by the pre-processing operation. The symmetry and regularity of Table 1 imply the simplicity for implementing the proposed scheme while the symmetry and regularity of Table 2 imply the *near-constancy* of embedding capacity. The sum of the proportions including the two ‘1/2’ in the parentheses equals 4.5156. With 8 bits, the number of possible Hamming codes (states) is 256, each having a probability of 1/256. Therefore, the *predicted embedding capacity* (PEC) of the proposed scheme is

$$PEC = 4.5156 \cdot \frac{1}{256} = 0.0176 \quad \text{bits/pixel} \quad (1)$$

Note from Table 2, we can see that the number of watermarkable pixels with more significant watermarkable bit is smaller. This is helpful in keeping distortion down.

Another key feature of the proposed work is the involvement of a secret contextual dependence information, $s(i)$, for countering vector quantization attack and transplantation attack.

The missing definition of $s(i)$ can now be defined as

$$S(i) = \left(\sum_{j \in N(i)} f(j) \right) \bmod 256, \quad \forall f(i) \text{ whose } D_{k,h_3,h_2,h_1,h_0} \text{ does not match any one in Table1.} \quad (2)$$

Since D_{k,h_3,h_2,h_1,h_0} is unknown to the third party without the secret key, $s(i)$ is secret. *mod* is the modular arithmetic operator. ‘*mod 256*’ operation confines the range of $s(i)$ in $[0, 255]$, the same range as $f(i)$ and $w(i)$. If the watermarked image is attacked, $s(i)$ changes accordingly. Consequently, the states are disturbed and the correct watermark bits cannot be extracted.

The watermark embedding and detecting algorithms of proposed scheme are summarized as follows.

Watermark embedding algorithm

Step_e 1: Generate an 8-bit watermark image w with the secret key shared with the detector

Step_e 2: For each pixel i ,

Step_e 2.1: Calculate the Hamming code $h(f(i), w(i))$ and distance $D(f(i), w(i))$

Step_e 2.2: Negate the LSB of $f(i)$ if $D(f(i), w(i)) = 0$ or 8 (pre-processing).

Step_e 3: Identify watermarkable pixels

Step_e 4: For each watermarkable pixel i ,

Step_e 4.1: Calculate the secret dependence information $s(i)$ according to Eq. (2)

Step_e 4.2: Negate the watermarkable bit of $f(i)$ depending on the state of the pixel.

Watermark Detecting Algorithm

Step_d 1: Generate an 8-bit watermark image w with the secret key shared with the embedder

Step_d 2: Initialise the extracted watermark w' by letting $w' = w$.

Step_d 3: For each pixel i , calculate the Hamming code $h(f(i), w(i))$ and distance $D(f(i), w(i))$

Step_d 4: Identify watermarkable pixels

Step_d 5: For each watermarkable pixel i ,

Step_d 5.1: Calculate the secret dependence information $s(i)$ according to Eq. (2)

Step_d 5.2: Extract watermark bit by setting $w'_f(i) = s_f(i)$

Step_d 5.3: Recover original image pixel $f(i)$ by negating the watermarkable bit of $f(i)$
if $w'(i) = w(i)$.

4. Algorithm Analyses

A general expression of the embedding capacity of the proposed scheme can be derived as follow. Suppose we have an image of b bits per pixel and we do not want to watermark any bit of the pixels more significant than bit k , $k \in [0, b-1]$. Table 2 can be expanded into a $2 \cdot \lfloor \frac{b}{2} \rfloor \times (k+1)$ matrix, where $\lfloor \cdot \rfloor$ is the *floor* function that returns the greatest integer less than or equal to its argument. Taking the upper half of Table 2 without the '1/2' in the parentheses into consideration, the value of each element decreases monotonically toward the lower-right corner of the matrix.

The value at the upper-left corner equals $1/2$ while the value at the upper-right corner equals $1/2^{\lfloor \frac{b}{2} \rfloor \times (k+1)}$. If we take the two halves of the matrix and the two '1/2' in the parentheses into consideration, because of the symmetrical characteristic of the matrix, an expression of the *predictable embedding capacity (PEC)* can be formulated as

$$PEC = 2 \cdot \left(\frac{1}{2} + \sum_{i=0}^{\lfloor b/2 \rfloor} \sum_{j=0}^k \frac{1}{2^{i+j+1}} \right) \cdot \frac{1}{2^b}$$

If we take the special case described in Section 3 as an example, i.e., $b = 8$ and $k = 3$, then $PEC = 0.0176$.

This model offers the user some degree of freedom in specifying the performance of the scheme. The factor $1/2^b$ in the above expression indicates that the more bits per pixel, the lower the capacity. Therefore, instead of involving all the b bits in the creation of Hamming code and the definition of the states, the user can choose to use fewer bits per pixel. Now, with a b -bit image, if we only allow the b_1 least significant bits to be involved in the definition of the states and the watermarkable bit to be as high as bit k , $k \in [0, b_1-1]$, a more general expression of the *predictable embedding capacity (PEC)* can be formulated as

$$PEC = 2 \cdot \left(\frac{1}{2} + \sum_{i=0}^{\lfloor b_1/2 \rfloor} \sum_{j=0}^k \frac{1}{2^{i+j+1}} \right) \cdot \frac{1}{2^{b_1}} \quad (3)$$

However, the capacity is increased at the expense of having to involve more pixels in the pre-processing stage in order to make empty initial states. For example if b_1 equals 4, the number of possible Hamming codes is 16, and the pixels associated with Hamming distance 0 and 4 will have to be involved in the pre-processing stage, which account for 2/16 of the total pixel population. Although as we mentioned in the previous section that due to the imperfection of the

image acquisition system, involving small proportion of pixels in the pre-processing stage is acceptable. However, large-scale involvement still needs to be avoided in practice.

5. Experiments

The proposed scheme has been tested on six common images as shown in Figure 1. We involve all the 8 bits of a pixel to create Hamming code and allow the scheme to mark up to bit 3 only. The size of the tested images and the performance of the scheme in terms of embedding capacity (bits per pixel) and PSNR (dB) are listed in Table 3. From the table, we can clearly see that capacity and distortion for Lena's Face and Lena are nearly the same. We can also see from the same table that even the nature of the images varies significantly, the embedding capacity is still nearly constant, i.e., the capacity is independent of the images. These figures are closely consistent with the *predicted embedding capacity (PEC)*, 0.0176 bits per pixel, as calculated in Eq (1). It is interesting to see that the distortion in terms of PSNR inflicted on the images by the scheme is also near-constant, with the lowest one equal to 55.719 dB. The original and watermarked versions of Lena are illustrated in Figure 2 for comparison. Note higher capacity can be achieved by changing the parameters b_1 and k in Eq. (3).

Embedding capacity of Fridrich et al's scheme with 'amplitude' equal to 1 reported in [7] is converted into bits per pixels and listed in Table 4. The embedding capacity of our proposed scheme for the first three images is listed alongside for comparison. From the first two entries of Table 4, we can see that, with Fridrich et al's scheme, the embedding capacities of Lena's Face and Lena are 1.5 times different, while the capacities with our scheme are nearly the same. This difference is more prominent when the capacities of Mandrill and Desert are compared. While

Desert has very high capacity, Mandrill's capacity is very close to zero. These figures indicate that, Fridrich et al's scheme is highly sensitive to the nature of the image. Images with more low-frequency contents tend to have higher capacity while images with more high-frequency contents tend to have relatively lower capacity. Actually this is a common characteristic, which can be found in Tan's [13], Alattar's [1], and van Leest et al's [14] schemes.

6. Conclusions

We pointed out, in this work, that seeking high embedding capacity at the expense of high distortion to some extent may marginalize the advantage of fragile watermarking over cryptography and emphasized the importance of finding the balance between embedding capacity and embedding distortion. We also observed that finding this balance is not a trivial task for the schemes with *signal-dependent embedding capacity* and proposed a new scheme with near-constant embedding capacity, which is independent of the host signal. We also addressed the issue of leaving a security gap open to the vector quantization attack and transplantation attack due to the lack of non-deterministic contextual dependence information in the embedding process and proposed a simple method for establishing the dependence information.

REFERENCES

- [1] A. M. Alattar, "Reversible watermark using difference expansion of triplets," in *Proc. IEEE Intl. Conf. Image Processing*, Sep. 2003.

- [2] U P. S. L. M. Barreto, H. Y. Kim, and V. Rijmen, "Toward secure public-key blockwise fragile authentication watermarking," in *IEE Proceedings - Vision, Image and Signal Processing*, vol. 148, no. 2, pp. 57 – 62, April 2002.
- [3] J. M. Barton, "Method and apparatus for embedding authentication information within digital data," U. S. Patent 5 646 997, 1997.
- [4] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible data hiding," in *Proc. Int. Conf. Image Proceesing*, vol. II, pp. 157-160, Sep. 2002.
- [5] I. J. Cox, M. Miller, and B. Jeffrey, *Digital Watermarking: Principles and Practice*, Morgan Kaufmann, 2002.
- [6] J. J. Fridrich, M. Goljan, and N. Memom, "Cryptanalysis of the Yeung–Mintzer fragile watermarking technique," *Journal of Electronic Imaging*, vol. 11, no 2, pp. 262-274, 2002.
- [7] M. Goljan, J. J. Fridrich, and R. Du, "Distortion-free data embedding for images," *Proceeding of the 4th Information Hiding Workshop*, 2001.
- [8] M. Holliman and N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," *IEEE Trans. Image Processing*, vol. 9, no. 3, pp. 432-441, March 2000.
- [9] C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," *US patent*, 6 278 791, 2001.
- [10] C.-T. Li, "Digital fragile watermarking scheme for authentication of JPEG images image authenticity," *IEE Proceedings – Vision, Image, and Signal Processing* (Accepted).

- [11] C.-T. Li and F.-M. Yang, "One-dimensional neighbourhood forming strategy for fragile watermarking," *Journal of Electronic Imaging*, vol. 12, no 2, pp. 284-291, 2003.
- [12] Macq, "Lossless multiresolution transform for image authenticating watermarking," in *Proc. EUSIPCO 2000*, Sept. 2000.
- [13] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, Aug. 2003
- [14] van Leest, A., van der Veen, M., & Bruekers, F. (2003). Reversible image watermarking. *Proceedings of the IEEE International Conference on Image Processing, II*, 731-734.
- [15] C. De Vleeschouwer, J. F. Delaigle, and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Trans. on Multimedia*, vol. 5, pp. 87-105, March 2003.
- [16] P. W. Wong and N. Memom, "Secret and public key authentication watermarking schemes that resist vector quantization attack," in *Proc. SPIE Security and Watermarking of Multimedia Contents II*, vol. 3971, no. 40, Jan. 2000.
- [17] M. Yeung and F. Minzter, "Invisible watermarking for image verification," *Journal of Electronic Imaging*, vol. 7, no. 2, pp. 578-591, July 1998.

Table 1. Table of watermarkable states and forward state transition directions specified with the arrows.

D_0	$D_0 \& s_0=w_0$	$D_0 \& s_1s_0=w_1w_0$	$D_0 \& s_2\sim s_0=w_2\sim w_0$
$D_{1,000\underline{1}} \& s_0=w_0$	$\uparrow D_{1,00\underline{1}0} \& s_1s_0=w_1w_0$	$\uparrow D_{1,0\underline{1}00} \& s_2\sim s_0=w_2\sim w_0$	$\uparrow D_{1,\underline{1}000} \& s_3\sim s_0=w_3\sim w_0$
$D_{2,00\underline{1}1} \& s_1s_0=w_1w_0$	$\uparrow D_{2,0\underline{1}10} \& s_2\sim s_0=w_2\sim w_0$	$\uparrow D_{2,010\underline{1}} \& s_3\sim s_0=w_3\sim w_0$	$\uparrow D_{2,100\underline{1}} \& s_4\sim s_0=w_4\sim w_0$
$D_{3,0\underline{1}11} \& s_2\sim s_0=w_2\sim w_0$	$\uparrow D_{3,\underline{1}110} \& s_3\sim s_0=w_3\sim w_0$	$\uparrow D_{3,\underline{1}101} \& s_4\sim s_0=w_4\sim w_0$	$\uparrow D_{3,10\underline{1}1} \& s_5\sim s_0=w_5\sim w_0$
$D_{4,\underline{1}111} \& s_3\sim s_0=w_3\sim w_0$	$\uparrow D_{4,1\underline{1}11} \& s_4\sim s_0=w_4\sim w_0$	$\uparrow D_{4,11\underline{1}1} \& s_5\sim s_0=w_5\sim w_0$	$\uparrow D_{4,11\underline{1}1} \& s_6\sim s_0=w_6\sim w_0$
$D_{4,\underline{0}000} \& s_3\sim s_0=w_3\sim w_0$	$\downarrow D_{4,000\underline{0}} \& s_4\sim s_0=w_4\sim w_0$	$\downarrow D_{4,000\underline{0}} \& s_5\sim s_0=w_5\sim w_0$	$\downarrow D_{4,000\underline{0}} \& s_6\sim s_0=w_6\sim w_0$
$D_{5,1\underline{0}00} \& s_2\sim s_0=w_2\sim w_0$	$\downarrow D_{5,\underline{0}001} \& s_3\sim s_0=w_3\sim w_0$	$\downarrow D_{5,\underline{0}010} \& s_4\sim s_0=w_4\sim w_0$	$\downarrow D_{5,01\underline{0}0} \& s_5\sim s_0=w_5\sim w_0$
$D_{6,11\underline{0}0} \& s_1s_0=w_1w_0$	$\downarrow D_{6,1\underline{0}01} \& s_2\sim s_0=w_2\sim w_0$	$\downarrow D_{6,101\underline{0}} \& s_3\sim s_0=w_3\sim w_0$	$\downarrow D_{6,011\underline{0}} \& s_4\sim s_0=w_4\sim w_0$
$D_{7,111\underline{0}} \& s_0=w_0$	$\downarrow D_{7,11\underline{0}1} \& s_1s_0=w_1w_0$	$\downarrow D_{7,1\underline{0}11} \& s_2\sim s_0=w_2\sim w_0$	$\downarrow D_{7,\underline{0}111} \& s_3\sim s_0=w_3\sim w_0$
D_8	$D_8 \& s_0=w_0$	$D_8 \& s_1s_0=w_1w_0$	$D_8 \& s_2\sim s_0=w_2\sim w_0$

Table 2. The proportions of the pixels, which are watermarkable.

$D_{1,000\underline{1}}$	1/2 (1/2)	$D_{1,00\underline{1}0}$	1/4	$D_{1,0\underline{1}00}$	1/8	$D_{1,\underline{1}000}$	1/16
$D_{2,00\underline{1}1}$	1/4	$D_{2,0\underline{1}10}$	1/8	$D_{2,010\underline{1}}$	1/16	$D_{2,100\underline{1}}$	1/32
$D_{3,0\underline{1}11}$	1/8	$D_{3,\underline{1}110}$	1/16	$D_{3,\underline{1}101}$	1/32	$D_{3,10\underline{1}1}$	1/64
$D_{4,\underline{1}111}$	1/16	$D_{4,1\underline{1}11}$	1/32	$D_{4,11\underline{1}1}$	1/64	$D_{4,11\underline{1}1}$	1/128
$D_{4,\underline{0}000}$	1/16	$D_{4,000\underline{0}}$	1/32	$D_{4,000\underline{0}}$	1/64	$D_{4,000\underline{0}}$	1/128
$D_{5,1\underline{0}00}$	1/8	$D_{5,\underline{0}001}$	1/16	$D_{5,\underline{0}010}$	1/32	$D_{5,01\underline{0}0}$	1/64
$D_{6,11\underline{0}0}$	1/4	$D_{6,1\underline{0}01}$	1/8	$D_{6,101\underline{0}}$	1/16	$D_{6,011\underline{0}}$	1/32
$D_{7,111\underline{0}}$	1/2 (1/2)	$D_{7,11\underline{0}1}$	1/4	$D_{7,1\underline{0}11}$	1/8	$D_{7,\underline{0}111}$	1/16

Table 3. Performance of the proposed scheme.

Images Performance	Lena face (128 × 128)	Lena (256 × 256)	Mandrill (512 × 512)	Boat (200 × 200)	Cameraman (256 × 256)	F16 (256 × 256)	Average
<i>Embedding Capacity (bits/pixel)</i>	0.0168	0.0169	0.0168	0.0162	0.0167	0.0172	0.0168
<i>PSNR (dB)</i>	56.145	55.844	56.092	56.335	56.405	55.719	56.09

Table 4. Performance comparison

Image Scheme	Fridrich et al	Proposed
Lena Face (128×128)	0.0104	0.0168
Lena (256× 256)	0.0158	0.0169
Mandrill (512 ×512)	0.0007	0.0168
Desert (400×268)	0.0665	N/A
PalmTrees (400×268)	0.0085	N/A
<i>GoldenGate</i> (400×268)	0.0403	N/A
Mountains (400×268)	0.0154	N/A
Elcaptan(592 ×800)	0.0053	N/A
NYC (1024 ×1536)	0.0086	N/A
Girl (1024 ×1536)	0.0160	N/A
Average Capacity (bits/pixel)	0.019	0.0168
Average PSNR (dB)	53.12	56.027

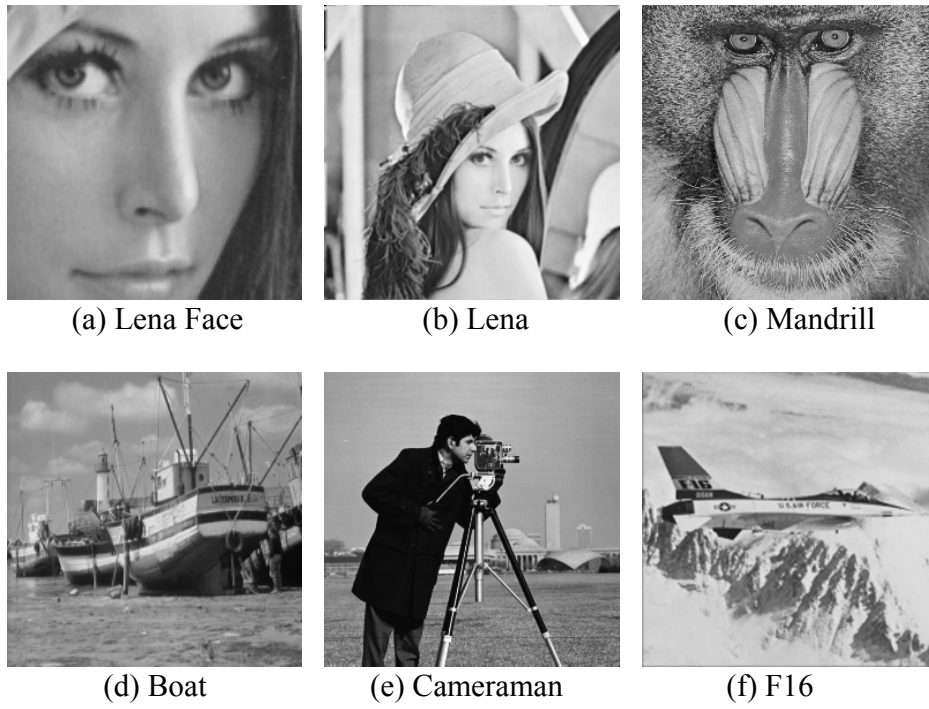


Figure 1. Six images used in the experiment.



Figure 2. (a) Original image of Lena (b) Watermarked image of Lena.