Generic Lossless Visible Watermarking—A New Approach

Tsung-Yuan Liu, *Student Member, IEEE*, and Wen-Hsiang Tsai, *Senior Member, IEEE*

***Abstract*—A novel method for generic visible watermarking with a capability of lossless image recovery is proposed. The method is based on the use of deterministic one-to-one compound mappings of image pixel values for overlaying a variety of visible watermarks of arbitrary sizes on cover images. The compound mappings are proved to be reversible, which allows for lossless recovery of original images from watermarked images. The mappings may be adjusted to yield pixel values close to those of desired visible watermarks. Different types of visible watermarks, including opaque monochrome and translucent full color ones, are embedded as applications of the proposed generic approach. A two-fold monotonically increasing compound mapping is created and proved to yield more distinctive visible watermarks in the watermarked image. Security protection measures by parameter and mapping randomizations have also been proposed to deter attackers from illicit image recoveries. Experimental results demonstrating the effectiveness of the proposed approach are also included.**

***Index Terms*—Lossless reversible visible watermarking, mapping randomization, one-to-one compound mapping, parameter randomization, translucent watermark, two-fold monotonically increasing.**

I. Introduction

THE advance of computer technologies and the proliferation of the Internet have made reproduction and distribution of digital information easier than ever before. Copyright protection of intellectual properties has, therefore, become an important topic. One way for copyright protection is *digital watermarking* [1]–[7], which means embedding of certain specific information about the copyright holder (company logos, ownership descriptions, etc.) into the media to be protected.

Digital watermarking methods for images are usually categorized into two types: *invisible* and *visible*. The first type aims to embed copyright information imperceptibly into host media such that in cases of copyright infringements, the hidden information can be retrieved to identify the ownership of the protected host. It is important for the watermarked image to be resistant to common image operations to ensure that the hidden information is still retrievable after such alterations. Methods of the second type, on the other hand, yield visible watermarks which are generally clearly visible after common image operations are applied. In addition, visible watermarks convey ownership information directly on the media and can deter attempts of copyright violations.

Embedding of watermarks, either visible or invisible, degrade the quality of the host media in general. A group of techniques, named *reversible* watermarking [8]–[19], allow legitimate users to remove the embedded watermark and restore the original content as needed. However, not all reversible watermarking techniques guarantee *lossless image recovery*, which means that the recovered image is identical to the original, pixel by pixel. Lossless recovery is important in many applications where serious concerns about image quality arise. Some examples include forensics, medical image analysis, historical art imaging, or military applications.

Compared with their invisible counterparts, there are relatively few mentions of lossless visible watermarking in the literature. Several lossless invisible watermarking techniques have been proposed in the past. The most common approach is to compress a portion of the original host and then embed the compressed data together with the intended payload into the host [5], [13]–[15]. Another approach is to superimpose the spread-spectrum signal of the payload on the host so that the signal is detectable and removable [3]. A third approach is to manipulate a group of pixels as a unit to embed a bit of information [16], [17]. Although one may use lossless invisible techniques to embed removable visible watermarks [11], [18], the low embedding capacities of these techniques hinder the possibility of implanting large-sized visible watermarks into host media.

As to lossless visible watermarking, the most common approach is to embed a monochrome watermark using deterministic and reversible mappings of pixel values or DCT coefficients in the watermark region [6], [9], [19]. Another approach is to rotate consecutive watermark pixels to embed a visible watermark [19]. One advantage of these approaches is that watermarks of arbitrary sizes can be embedded into any host image. However, only *binary* visible watermarks can be embedded using these approaches, which is too restrictive since most company logos are colorful.

In this paper, a new method for lossless visible watermarking is proposed by using appropriate *compound mappings* that allow mapped values to be controllable. The mappings are proved to be *reversible* for lossless recovery of the original image. The approach is *generic*, leading to the possibility of embedding different types of visible watermarks into cover images. Two applications of the proposed method are demonstrated, where opaque monochrome watermarks and nonuniformly translucent full-color ones are respectively embedded into color images. More specific compound mappings are also created and proved

Manuscript received June 09, 2009; revised December 23, 2009. First published January 19, 2010; current version published April 16, 2010. This work was supported in part by the NSC project 97-2631-H-009-001. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Srdjan Stankovic.

The authors are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: gis91811@cis.nctu.edu.tw; whtsai@cis.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2040757

to be able to yield visually more distinctive visible watermarks in the watermarked image. To the best knowledge of the authors, this is the first method ever proposed for embedding *removable translucent full-color watermarks* which provide better advertising effects than traditional monochrome ones.

In the remainder of this paper, the proposed method for deriving one-to-one compound mappings is described in Section II. Related lemmas and theorems are also proved and security protection measures described. Applications of the proposed method for embedding opaque monochrome and translucent color watermarks into color images are described in Sections III and IV, respectively. In Section V, the specific compound mapping for yielding more distinctive visible watermarks is described. In Section VI, experimental results are presented to demonstrate the effectiveness of the proposed method. Finally, a conclusion with some suggestions for future work is included in Section VII.

II. Proposed New Approach to Lossless Visible Watermarking

In this section, we describe the proposed approach to lossless reversible visible watermarking, based on which appropriate one-to-one compound mappings can be designed for embedding different types of visible watermarks into images. The original image can be recovered losslessly from a resulting watermarked image by using the corresponding reverse mappings.

A. Reversible One-to-One Compound Mapping

First, we propose a generic *one-to-one compound mapping* f for converting a set of numerical values $P = \{p_1, p_2 \ldots, p_M\}$ to another set $Q = \{q_1, q_2 \ldots, q_M\}$, such that the respective mapping from p_i to q_i for all $i = 1, 2, \ldots, M$ is *reversible*. Here, for the copyright protection applications investigated in this study, all the values p_i and q_i are image pixel values (grayscale or color values). The compound mapping f is governed by a one-to-one function F_x with one parameter $x = a$ or b in the following way:

$$q = f(p) = F_b^{-1}(F_a(p)) \tag{1}$$

where F_x^{-1} is the inverse of F_x which, by the one-to-one property, leads to the fact that if $F_a(p) = p'$, then $F_a^{-1}(p') = p$ for all values of a and p. On the other hand, $F_a(p)$ and $F_b(p)$ generally are set to be *unequal* if $a \neq b$.

The compound mapping described by (1) is indeed *reversible*, that is, p can be derived exactly from q using the following formula:

$$p = f^{-1}(q) = F_a^{-1}(F_b(q)) \tag{2}$$

as proved below.

Lemma 1 (Reversibility of Compound Mapping): If $q = F_b^{-1}(F_a(p))$ for any one-to-one function F_x with a parameter x, then $p = F_a^{-1}(F_b(q))$ for any values of a, b, p, and q.

Proof: Substituting (1) into $F_a^{-1}(F_b(q))$, we get

$$F_a^{-1}(F_b(q)) = F_a^{-1}\left(F_b\left(F_b^{-1}(F_a(p))\right)\right).$$

By regarding $F_a(p)$ as a value c, the right-hand side becomes $F_a^{-1}(F_b(F_b^{-1}(c)))$, which, after F_b and F_b^{-1} are cancelled out, becomes $F_a^{-1}(c)$. But $F_a^{-1}(c) = F_a^{-1}(F_a(p))$, which is just p after F_a and F_a^{-1} are cancelled out. That is, we have proved $p = F_a^{-1}(F_b(q))$. □

As an example, if $F_x(p) = xp + d$, then $F_x^{-1}(p') = (p' - d)/x$. Thus

$$\begin{aligned} q &= F_b^{-1}(F_a(p)) = F_b^{-1}(ap + d) \\ &= (ap + d - d)/b = ap/b \end{aligned}$$

and so, we have

$$\begin{aligned} F_a^{-1}(F_b(q)) &= F_a^{-1}(b(ap/b) + d) = F_a^{-1}(ap + d) \\ &= [((ap + d) - d)/a] = (ap/a) = p \end{aligned}$$

as expected by Lemma 1.

B. Lossless Visible Watermarking Scheme

Based on Lemma 1, we will now derive the proposed generic lossless visible watermarking scheme in the form of a class of one-to-one compound mappings, which can be used to embed a variety of *visible watermarks* into images. The embedding is reversible, that is, the watermark can be removed to recover the original image losslessly. For this aim, a preliminary lemma is first described as follows.

Lemma 2 (Preference of Compound-Mapped Value q): It is possible to use the compound mapping $q = F_b^{-1}(F_a(p))$ to convert a numerical value p to another value close to a *preferred* value l.

Proof: Let $F_x(p) = p - x$ where x is the parameter for F. Then $F_x^{-1}(p') = p' + x$. Also, let $a = p - \varepsilon$ and $b = l$ where ε is a small value. Then, the compound mapping $F_b^{-1}(F_a(p))$ of p yields q as

$$\begin{aligned} q &= F_b^{-1}(F_a(p)) = F_b^{-1}(p - a) = F_b^{-1}(\varepsilon) \\ &= \varepsilon + b = \varepsilon + l \end{aligned}$$

which means that the value q is close to the preferred value l. □

The above lemma relies on two assumptions. The first is that a is close to p, or equivalently, that $a = p - \varepsilon$. The reason why we derive the above lemma for $a = p - \varepsilon$ instead of for $a = p$, is that in the reverse mapping we want to recover p from q *without knowing* p, which is a requirement in the applications of reversible visible watermarking investigated in this study. Although the value of p cannot be known in advance for such applications, it can usually be estimated, and we will describe some techniques for such estimations in the subsequent sections.

The second assumption is that $F_x(p)$ yields a small value if x and p are close. Though the basic difference function $F_x(p) = p - x$ used in the above proof satisfies this requirement for most cases, there is a possible problem where the mapped value may exceed the range of valid pixel values for some values of a, b, and p. For example, when $a = 255, b = 255$, and $p = 253$, we have $q = 255 - 253 + 255 = 257 > 255$. It is possible to use the standard modulo technique (i.e., taking $q = 257_{\text{mod } 256} = 1$) to solve this issue; however, such a technique will make q far from the desired target value of b, which is 255. Nevertheless, we will show in Section 3 that using such a standard modulo function, $F_x(p) = (p - x)_{\mathrm{mod}\ 256}$, can still yield reasonable experimental results. Furthermore, we show in Section 5 a more sophisticated one-to-one function that is free from such a wrap-around problem.

By satisfying the above two requirements, the compound mapping yields a value q that is close to the desired value l. We now prove a theorem about the desired lossless reversible visible watermarking in the following.

Theorem 1 (Lossless Reversible Visible Watermarking): There exist one-to-one compound mappings for use to embed into a given image I a visible watermark Q whose pixel values are close to those of a given watermark L, such that the original image I can be recovered from Q losslessly.

Proof: This is a consequence of Lemmas 1 and 2 after regarding the individual pixel values in I, L, and Q respectively as those of p, l, and q mentioned in Lemma 2. And it is clear by Lemma 1 that the value p can be recovered losslessly from the mapped value q which is derived in Lemma 2. □

The above discussions are valid for embedding a watermark in a grayscale image. If color images are used both as the cover image and the watermark, we can apply the mappings to each of the color channels to get multiple independent results. The resulting visible watermark is the composite result of the color channels.

Based on Theorem 1, the proposed generic lossless reversible visible watermarking scheme with a given image I and a watermark L as input is described as an algorithm as follows.

Algorithm 1: Generic Visible Watermark Embedding

```
Input: an image $I$ and a watermark $L$.

Output: watermarked image $W$.

Steps:
  1) Select a set $P$ of pixels from $I$ where $L$ is to be
     embedded, and call $P$ a watermarking area.
  2) Denote the set of pixels corresponding to $P$ in $W$ by $Q$.
  3) For each pixel $X$ with value $p$ in $P$, denote the
     corresponding pixel in $Q$ as $Z$ and the value of the
     corresponding pixel $Y$ in $L$ as $l$, and conduct the
     following steps.
       a) Apply an estimation technique to derive $a$ to
          be a value close to $p$, using the values of the
          neighboring pixels of $X$ (excluding $X$ itself).
       b) Set $b$ to be the value $l$.
       c) Map $p$ to a new value $q = F_b^{-1}(F_a(p))$.
       d) Set the value of $Z$ to be $q$.
  4) Set the value of each remaining pixel in $W$, which
     is outside the region $P$, to be equal to that of the
     corresponding pixel in $I$.
```

Note that we do *not* use the information of the *original* image pixel value of X itself for computing the parameters a and b for X. This ensures that identical parameter values can be calculated by the receiver of a watermarked image for the purpose of lossless image recovery.

As an example, the process performed by Step 3 of the above algorithm for a pixel is illustrated by Fig. 1, where the north and west pixels are used to estimate the color of the center pixel. Note that the east and south pixels are not used because these pixels are covered by the watermark and unknown to the receiver. It is important to allow as many neighbors of a pixel as possible to be known by the receiver to ensure that a good estimate can be calculated for that pixel. We will describe in Section 4 techniques for processing pixels, which can ensure that sufficiently many neighbor colors are known by a receiver for each pixel in the watermarking area.

The corresponding watermark removal process for a watermarked image W generated by Algorithm 1 is described as an algorithm as follows.

Algorithm 2: Generic Watermark Removal for Lossless Image Recovery

```
Input: a watermarked image $W$ and a watermark $L$.

Output: the original image $R$ recovered from $W$.

Steps:
  1) Select the same watermarking area $Q$ in $W$ as that
     selected in Algorithm 1.
  2) Set the value of each pixel in $R$, which is outside the
     region $Q$, to be equal to that of the corresponding pixel
     in $W$.
  3) For each pixel $Z$ with value $q$ in $Q$, denote the
     corresponding pixel in the recovered image $R$ as $X$
     and the value of the corresponding pixel $Y$ in $L$ as $l$,
     and conduct the following steps.
       a) Obtain the same value $a$ as that derived in Step 3a
          of Algorithm 1 by applying the same estimation
          technique used there.
       b) Set $b$ to be the value $l$.
       c) Restore $p$ from $q$ by setting $p = F_a^{-1}(F_b(q))$.
       d) Set the value of $X$ to be $p$.
```

C. Security Considerations

As mentioned previously, although we want legitimate users to be able to recover the original image from a watermarked one, we do not want an attacker to be able to do the same. Herein, we propose some security protection measures against illicit recoveries of original images.

First, we make the parameters a and b in the above algorithms to be dependent on certain secret keys that are known only by the creator of the watermarked image and the intended receivers. One simple technique to achieve this is to use a secret key to generate a pseudo-random sequence of numerical values and add them to either or both of a and b for the pixels in the watermarking area. This technique is hereinafter referred to as *parameter randomization*.

Another way of security protection is to make the choices of the *positions* for the pixels to be dependent on a secret key. Specifically, we propose to process *two* randomly chosen pixels (based on the security key) in P simultaneously as follows. Let the two pixels be denoted as X_1 and X_2 with values p_1 and p_2,

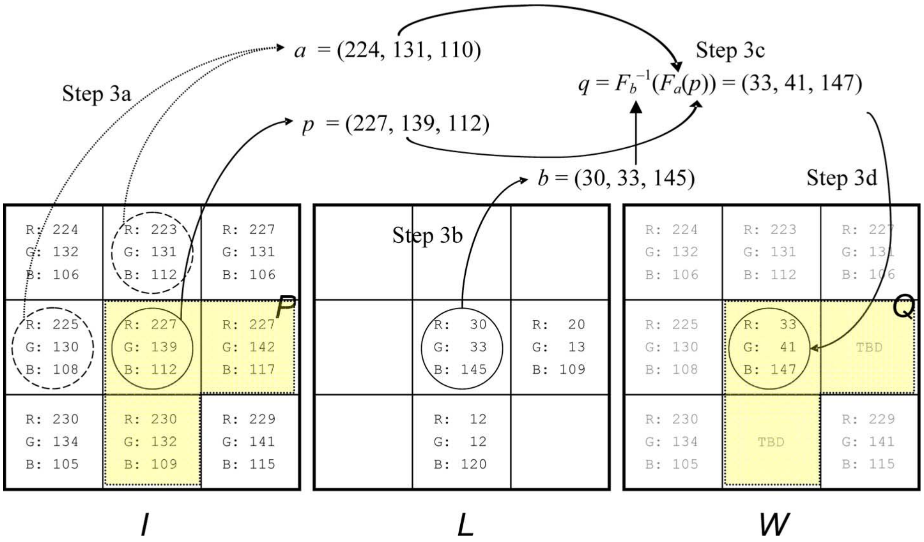
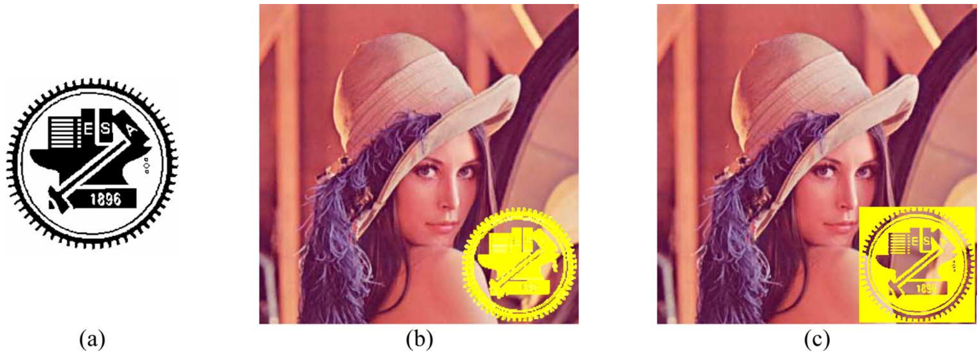


Fig. 1. Illustration of mapping the center pixel of a 3 × 3 image using Algorithm 1. Only the mapping of the center pixel is shown for clarity; the east and south pixels are depicted as TBD (to be determined) in W.



(a) (b) (c)

Fig. 2. Illustration of pixels in a watermark. (a) A monochrome watermark. (b) Area of P (yellow pixels). (c) Area of P' (yellow pixels).

respectively. The color estimates a_1 and a_2 corresponding to X_1 and X_2, respectively, are individually derived as before using their respective neighbors. The parameters b_1 and b_2 are set to be the values l_1 and l_2 of the respective watermark pixels Y_1 and Y_2. Then, instead of setting the values of the watermarked pixels Z_1 and Z_2 to be $q_1 = F_{b_1}^{-1}(F_{a_1}(p_1))$ and $q_2 = F_{b_2}^{-1}(F_{a_2}(p_2))$ as before, we *swap* the parameters and set

$$q_1 = F_{b_1}^{-1}(F_{a_2}(p_2)) \quad \text{and} \quad q_2 = F_{b_2}^{-1}(F_{a_1}(p_1)).$$

This parameter exchange does not affect the effectiveness of lossless recoverability, because we can now recover the original pixel values by the following compound mappings:

$$p_1 = F_{a_1}^{-1}(F_{b_2}(q_2)) \quad \text{and} \quad p_2 = F_{a_2}^{-1}(F_{b_1}(q_1)).$$

We will refer to this technique in the sequel as *mapping randomization*. We may also combine this technique with the above-mentioned parameter randomization technique to enhance the security further.

Last, the position in the image where a watermark is embedded affects the resilience of the watermarked image against illicit image recovery attempts. In more detail, if the watermark is embedded in a smooth region of the image, an attacker can simply fill the region with the background color to remove the watermark irrespective of the watermarking technique used. To counter this problem, an appropriate position should be chosen, using, for example, the adaptive positioning technique [20] when embedding a watermark. However, for ease of discussions and comparisons, we always embed a watermark in the lower right-hand corner of an image in this study.

III. Lossless Visible Watermarking of Opaque Monochrome Watermarks

As an application of the proposed generic approach to lossless visible watermarking, we describe now how we embed a losslessly-removable opaque monochrome watermark L into a color image I such that the watermark is *visually distinctive* in the watermarked image W.

First, we denote the sets of those pixels in I corresponding spatially to the *black* and *white* pixels in L by P and P', respectively. An illustration of such areas of P and P' is shown in Fig. 2. We define Q and Q' in a similar way for the watermarked image W, which correspond to P and P', respectively.

Then, we adopt the simple one-to-one function $F_a(p) = p - a$, and use the same pair of parameters a and b for *all* mappings

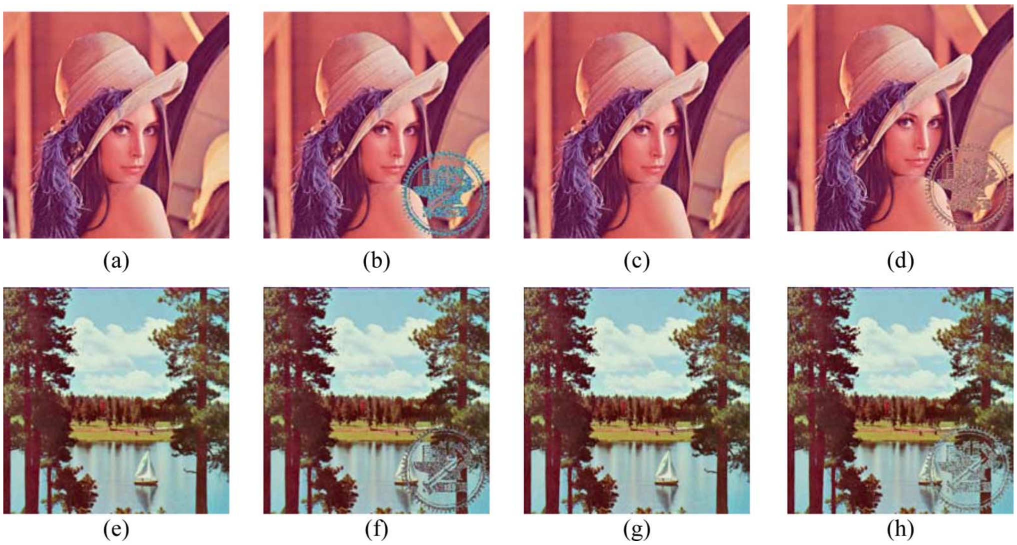


Fig. 3. Experimental results of monochrome watermark embedding and removal. (a) Image Lena. (e) Image Sailboat. (b), (f) Watermarked images of (a) and (e), respectively. (c), (g) Images losslessly recovered from (b) and (f), respectively, with correct keys. (d), (h) Images recovered from (b) and (f) with incorrect keys.

of pixels in P. Also, we apply the "modulo-256" operation to the results of all computations so that they are within the valid range of color values. Our experiments show that this method still yields reasonable results.

As to the values of parameters a and b, we set a to be the *average* of the color component values of the pixels in P'. This average value presumably is close to the value p of pixel X in P, fulfilling the condition $a = p - \varepsilon$ mentioned previously. To ensure that the watermark is distinctive in W, we do not simply embed black values for pixels in watermarking area P (that is, we do not embed $l = 0$ for P), but set l to be a value which is distinctive with respect to the pixel colors in the surrounding region P'. To achieve this, we set $b = l = a + 128$, which is a value *distinctive* with respect to a. As a result, the value of a pixel in Q, according to Lemma 2, becomes $q = F_b^{-1}(F_a(p)) = b + \varepsilon = a + 128 + \varepsilon$, meaning that the pixel values of Q are also distinctive with respect to those of the surrounding pixels in Q' as desired.

On the other hand, since both a and b are derived from P' during watermark embedding, the exact same values of a and b can be derived during watermark removal because Q' is identical to P'. The original image can, therefore, be recovered losslessly using Algorithm 2.

To demonstrate the effectiveness of the proposed method, in one of our experiments we embedded the watermark of Fig. 2(a) into the images Lena and Sailboat, respectively, and the results are shown in Fig. 3. For security protection, we applied both the mapping randomization and the parameter randomization techniques described in Section 2. Specifically, for the latter technique we added random integer values in the range of -12 to $+12$ to the parameter b.

The images recovered by using correct keys for the parameter and mapping randomization processes are shown in Fig. 3(c) and (g), and those recovered with incorrect keys are shown in Fig. 3(d) and (h). We observe from these figures that the embedded opaque watermarks are distinctive with respect to their surroundings and can be removed completely when the input key is correct. On the contrary, when the key was incorrect, the inserted watermark cannot be removed cleanly, with noise remaining in the watermarking area.

IV. Lossless Visible Watermarking of Translucent Color Watermarks

As another application of the proposed approach, we describe now how we embed more complicated *translucent color watermarks*. A translucent color watermark used in this study is an arbitrary RGB image with each pixel being associated with an *alpha component value* defining its *opacity*. The extreme alpha values of 0 and 255 mean that the watermark pixel is *completely transparent* and *totally opaque*, respectively. A translucent full-color watermark is visually more attractive and distinctive in a watermarked image than a traditional transparent monochrome watermark, as mentioned previously. Such a kind of watermark can better represent trademarks, emblems, logos, etc., and thus is more suitable for the purpose of advertising or copyright declaration.

If recoverability is not an issue, we can overlay the translucent watermark over the original image with an application package like Photoshop using the standard *alpha blending* operation to obtain a watermarked image, as illustrated in Fig. 4. Such an image will be called a *nonrecoverable watermarked image* in the sequel, and will be used as a *benchmark* in our experiments.

The proposed algorithm for embedding a translucent color watermark is similar to Algorithm 1 and is described below. To ensure that the parameter a is close to p for each pixel, we keep track of the pixels that have been *processed* throughout the embedding process. The pixels outside region P need not be processed and are regarded as having been processed in the following discussion.

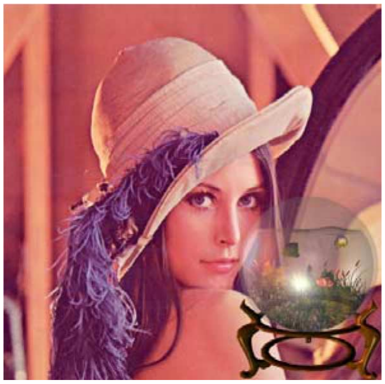


Fig. 4. Watermarked image of Lena with a translucent image of “Globe” superimposed using alpha blending.

Algorithm 3: Watermark Embedding of a Translucent Color Watermark

Input: an image I and a translucent watermark L.

Output: a watermarked image W.

Steps:

1) Select the watermarking area P in I to be the set of pixels corresponding spatially to those in L which are nontransparent (with alpha values larger than zero).
2) Denote the set of pixels corresponding to P in W as Q.
3) For each pixel X with value p in P, denote the corresponding pixel in Q as Z and the value of the corresponding pixel Y in L as l, and conduct the following steps.
    a) Set the parameter a to be a *neighbor-based color estimate* value that is *close* to p by using the colors of the neighboring pixels of X that *have already been processed* (see discussion below).
    b) Perform alpha blending with l over a to get the parameter b according to the formula $b = l \times \alpha + a \times (255 - \alpha)$ where α is the opacity of Y.
    c) Map p to a new value $q = F_b^{-1}(F_a(p))$.
    d) Set the value of Z to be q.
4) Set the value of each remaining pixel in W, which is outside the region P, to be equal to that of the corresponding pixel in I.

For Step 3a above, there are several ways to determine the color estimate of a pixel using the colors of its neighbors that have already been processed, such as simply averaging the colors of the processed 4-neighbors of the pixel, or averaging those of the processed 8-neighbors with more weights on the horizontal and vertical members. We may also use more sophisticated techniques such as edge-directed prediction [21] for this purpose, as long as we use only processed pixels.

The reason for using *only* processed pixels is that these pixels are the ones that a receiver can reliably recover during watermark removal. This is to ensure that the same color estimates can be computed for lossless recovery. Specifically, the value q of the first processed pixel is computed from the neighboring pixels outside the region P. Since the values of these pixels outside P are unchanged, a receiver can, therefore, reliably recover the first pixel using a reverse mapping using q and the values of neighboring pixels outside P. Each of the other unprocessed pixels is handled by using the processed pixels in a similar way.

To ensure that there always exists processed neighbors for accurate color estimates, we limit the pixels to be selected and processed next to be those with at least two already-processed neighbors in a four-pixel neighborhood. A consequence of this is that pixels around the outer edges of the watermark region are processed before those in the center. This can be clearly seen in Fig. 5, where some of the intermediate outputs yielded during watermark embedding and removing are shown [the most obvious outer edges are seen in Fig. 5(a)].

V. Two-Fold Monotonically Increasing Compound Mapping

In Section 2, we mapped a pixel value to a preferred value by using a simple one-to-one function $F_x(p) = (p - x)_{\mathrm{mod}\ 256}$. A problem of this mapping is that for certain values of a, b, and p, the mapped value will wrap around and deviate from the intended value. To solve this problem, we propose an alternative one-to-one function F_x such that the compound mapping $q = F_b^{-1}(F_a(p))$ does not exhibit the wrap-around phenomenon. Specifically, the mapping always yields a value *close* to b if a and p are *close* to each other for all values of a, b, and p. We will call this a *two-fold monotonically increasing* property, and will prove by a theorem that such a property holds if the one-to-one function F_x has a *one-fold monotonically increasing* property. The definitions of both of these properties and the detail of the theorem are described in the following.

Definition 1 (One-Fold Monotonically Increasing One-to-One Function): A one-to-one function F_a is one-fold monotonically increasing if for all values of a, p_1, and $p_2, F_a(p_1) < F_a(p_2)$ implies $|a - p_1| \le |a - p_2|$.

Lemma 3 (Inverse Monotonicity): The inverse of a one-fold monotonically increasing function F_x exhibits the following characteristic of *inverse monotonicity*:

for all values of b, p'_1, and $p'_2, p'_1 < p'_2$ implies $|b - F_b^{-1}(p'_1)| \le |b - F_b^{-1}(p'_2)|$.

Proof: Let $p'_1 = F_b(p_1)$ and $p'_2 = F_b(p_2)$ for some b, p_1 and p_2. Then

$$|b - p_1| \le |b - p_2|$$

by Definition 1. Also, we have $F_b^{-1}(p'_1) = F_b^{-1}(F_b(p_1)) = p_1$, and $F_b^{-1}(p'_2) = F_b^{-1}(F_b(p_2)) = p_2$, similarly. Substituting p_1 and p_2 into the above inequality, we get $|b - F_b^{-1}(p'_1)| \le |b - F_b^{-1}(p'_2)|$. This completes the proof. □

Definition 2 (Two-Fold Monotonically Increasing): The compound mapping $q = F_b^{-1}(F_a(p))$ is two-fold monotonically increasing if for all values of a, b, p_1 and p_2, $|a - p_1| < |a - p_2|$ (i.e., if a is closer to p_1 than p_2) implies $|b - q_1| \le |b - q_2|$ (i.e.,

b is at least as close to q_1 as q_2), where $q_1 = F_b^{-1}(F_a(p_1))$ and $q_2 = F_b^{-1}(F_a(p_2))$.

Theorem 2 (Two-Fold Monotonically Increasing): If F_x is a one-fold monotonically increasing one-to-one function with a parameter x, then the compound mapping $q = F_b^{-1}(F_a(p))$ is two-fold monotonically increasing.

The proof of the above theorem is included in the Appendix. We now show the existence of a one-fold monotonically increasing function $F_a(p)$ and how it works for any pixel value a and p in the range of 0 to 255, by way of an algorithm below.

Algorithm 4: One-to-One Mapping Exhibiting One-Fold Monotonically Increasing Property

***Input*:** a parameter a and an input value p, each in the range of 0 to 255.

***Output*:** a mapped output p' in the range from 0 to 255.

Steps:

```
1) Initialize $p'$ to be zero.
2) Create a set $S$ with initial elements being the 256
   values of 0 through 255.
3) Find a value $r$ in $S$ such that $|a - r|$ is the minimum,
   preferring a smaller $r$ in case of ties.
4) If $r$ is not equal to $p$, then remove $r$ from $S$, increment
   $p'$ by one, and go to Step 3; otherwise, take the final
   $p'$ as the output.
```

As an example, if we want to determine the function value $F_a(p)$ for $a = 3$ and $p = 1$ by the above algorithm, then we will find $r = 3$ in Step 3 of the above algorithm. But $r = 3 \neq 1 = p$, so 3 is removed from S with p' being incremented from 0 to 1. The subsequent iterations will compute r to be 2, 4, and finally 1 which is equal to p, with the final value of p' being taken to be 3 as the output.

The inverse of the one-to-one function described by Algorithm 4 is described below.

Algorithm 5: Inverse of the Mapping Function Described by Algorithm 4

***Input*:** a parameter b and an input value p', each in the range of 0 to 255.

***Output*:** an output value p that is in the range from 0 to 255.

Steps:

```
1) Create a set $S$ with the initial elements being the 256
   values of 0 through 255.
2) Find a value $p$ in $S$ such that $|b - p|$ is the minimum,
   preferring a smaller $p$ in case of ties.
3) If $p'$ is larger than zero, then remove $p$ from $S$,
   decrement $p'$ by one, and go to Step 2; otherwise, take
   the final $p$ as the output.
```

As an example, if we want to compute $F_b^{-1}(p')$ for $b = 3$ and $p' = 3$ by the above algorithm, then we will find in Step 2 the sequence of 3, 2, 4, and 1 for the values of p, with p' decreasing from 3, 2, 1, and then 0. The output is hence $p = 1$.

Note that in practice, we can precompute all 256 × 256 possible one-to-one mappings in both Algorithms 4 and 5 beforehand, so that the mapping F_x and its inverse F_x^{-1} can be implemented by efficient lookup-table operations of constant-time complexity. As proved by Theorem 2 and two extra lemmas (Lemmas 4 and 5) included in the Appendix, we can use the mapping and its inverse described in Algorithms 4 and 5, respectively, to map the pixel values of an image to the desired values of a watermarked image, such that the watermark is visually clear if a is close to p. It is guaranteed that the original image can be recovered losslessly from the watermarked image, as proved by Theorem 1.

VI. Experimental Results

A series of experiments implementing the proposed methods were conducted using the Java SE platform.[1] To quantitatively measure the effectiveness of the proposed method, we define a set of performance metrics here. First, the quality of a watermarked image W is measured by the peak signal-to-noise ratio (PSNR) of W with respect to the nonrecoverable watermarked image B in the following way:

$$\mathrm{PSNR}_W = 20 \times \log_{10}\left(255 \Bigg/ \sqrt{\frac{1}{w \times h}\sum_{y=1}^{h}\sum_{x=1}^{w}[W(x,y) - B(x,y)]^2}\right).$$

Also, the quality of a *recovered* image R is measured by the PSNR of R with respect to the original image I in a similar way

$$\mathrm{PSNR}_R = 20 \times \log_{10}\left(255 \Bigg/ \sqrt{\frac{1}{w \times h}\sum_{y=1}^{h}\sum_{x=1}^{w}[R(x,y) - I(x,y)]^2}\right).$$

It is desired to have the value of the PSNR_W to be as high as possible, so that the watermarked image can be visually as close to the benchmark image as possible. For illicit recoveries, the PSNR_R should be as low as possible to make the recovered image visually intolerable (e.g., very noisy). In particular, we want the region obscured by the watermark to be as noisy as possible in an illicitly recovered image. For this purpose, we introduce an additional quality metric for an illicitly recovered image that only takes into account the region Q covered by the watermark. Specifically, we measure the quality of the recovered image R by the following PSNR measure:

$$\mathrm{PSNR}_Q = 20 \times \log_{10}\left(255 \Bigg/ \sqrt{\frac{1}{|Q|}\sum_{y=1}^{h}\sum_{x=1}^{w} SE_Q(x,y)}\right)$$

[1] The source code of the implementation is available at http://sites.google.com/site/ktyliu/lossless-visible-watermarking.

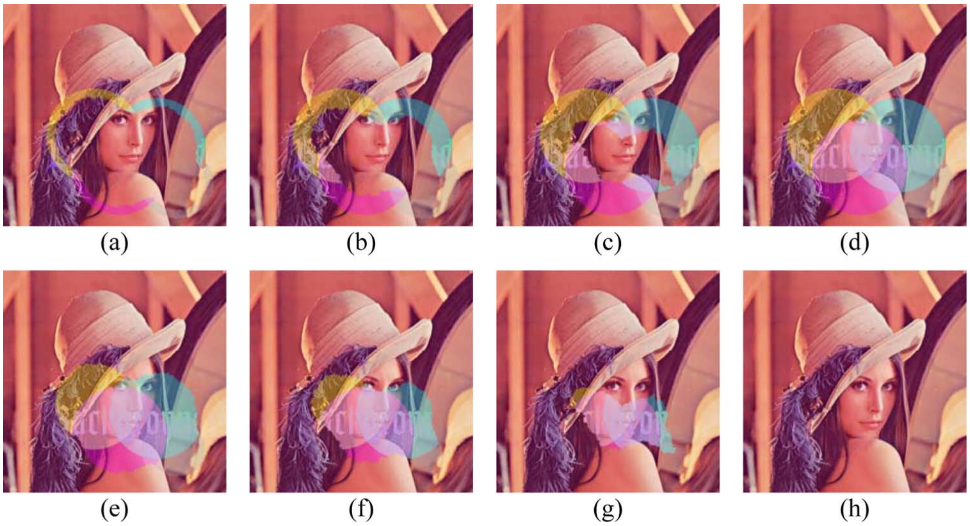


Fig. 5. Illustration of pixel processing order in watermark embedding and removal. (a)–(d) Intermediate results of image watermarking when 25%, 50%, 75%, and 100% of the watermark pixels have been processed, respectively. (e)–(h) Intermediate results of image recovery when 25%, 50%, 75%, and 100% of the watermark pixels have been recovered, respectively.

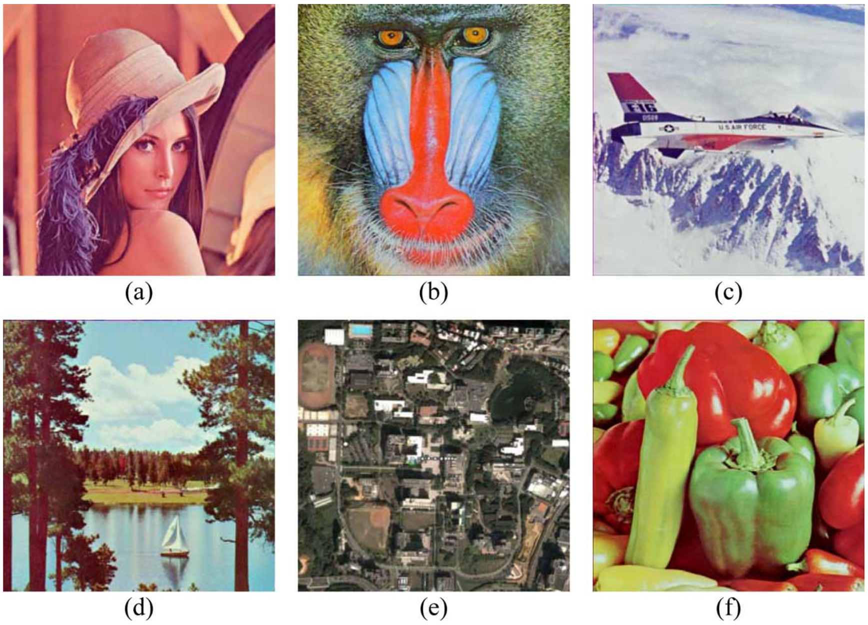


Fig. 6. Test images used in experiments: (a) Lena. (b) Baboon. (c) Jet. (d) Sailboat. (e) A satellite image of NCTU campus. (f) Pepper.

where

$$SE_Q\,(x,y) = \begin{cases} [R(x,y) - I(x,y)]^2, & \text{if } (x,y) \in Q \\ 0, & \text{if } (x,y) \notin Q. \end{cases}$$

Six test images, each of dimensions 512 × 512, were used in the experiments. They are shown in Fig. 6, referred to as “Lena,” “baboon,” “jet,” “boat,” “satellite,” and “pepper,” respectively, in the sequel. And seven test watermarks were used in the experiments as shown in Fig. 7, hereinafter referred to as watermarks A, B, C, D, E, F, and G, respectively. The width and height of each watermark are shown in Table I, along with the number of nontransparent pixels in each watermark ($|P|$) and several other properties described next. The *average opacity*, as shown in the fourth column, is the average of the opacities of the pixels in the watermark, and the *coverage*, as shown in the last column, is the size of the watermark over the original image, which is computed as $|P|/(w \times h)$. The watermarks are listed in an increasing order of $|P|$, and the pixels in watermarks A, B, C, and E are either totally opaque or totally transparent, while watermarks D, F, and G contain semi-transparent pixels.

Each of the seven test watermarks was embedded in the six test images using the method described in Section 4 with the one-to-one compound mapping described in Section 5. The color estimate of a pixel was derived by averaging the available four-neighbors of that pixel. Such an experiment was conducted twice to test the effectiveness of the two proposed security protection measures: the mapping and the parameter randomization techniques. For the latter, both the parameters

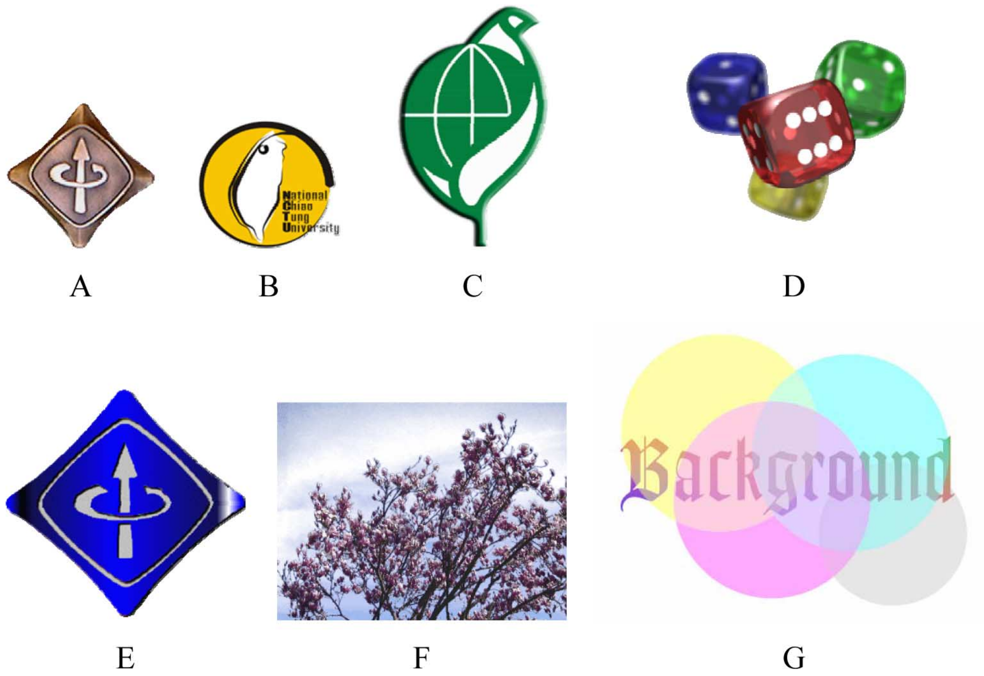


Fig. 7. Watermarks A through G used in experiments.

TABLE I
CHARACTERISTICS OF WATERMARKS A THROUGH G USED IN EXPERIMENTS

Watermark	Watermark Dimension	Non-transparent Pixels	Average Opacity	Watermark Coverage
A	162×160	12,226	255	4.7%
B	160×143	17,453	255	6.7%
C	168×268	28,001	255	10.7%
D	320×240	30,317	233	11.6%
E	274×263	32,995	255	12.6%
F	320×240	72,564	151	27.7%
G	440×330	88,424	125	33.7%

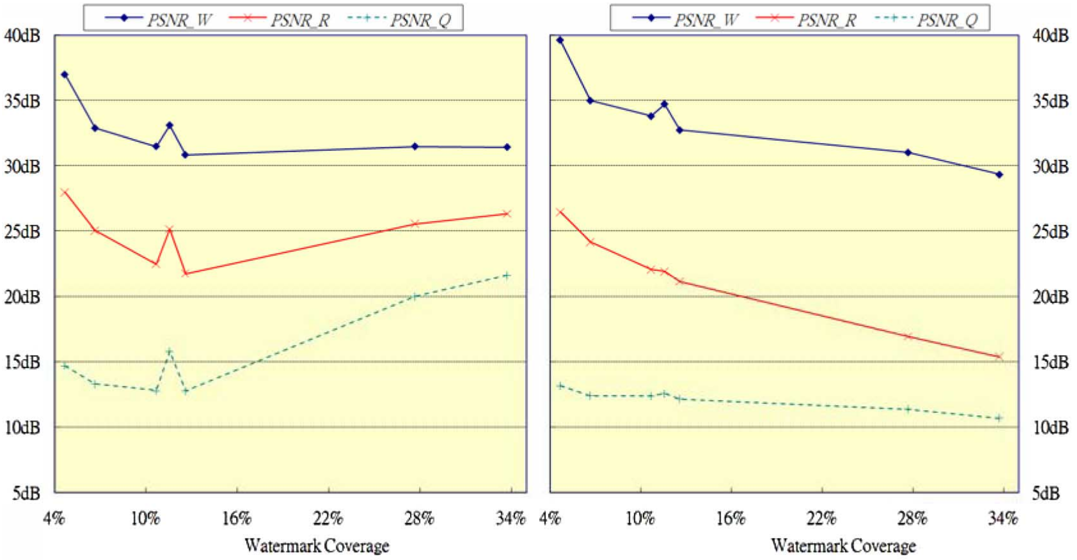
a and b of the compound mapping were adjusted randomly within a range of 25 with a uniform probability distribution.

A total of $7 \times 6 \times 2 = 84$ watermarked images were generated and for each watermarked image, recoveries using correct as well as incorrect keys were conducted. It was verified that the original images can be recovered *losslessly* from the watermarked ones for all the 84 test cases if correct keys were used. In Fig. 8, we plot the average values of the PSNR_W obtained after embedding a particular watermark in the six test images, as well as the average corresponding values of the PSNR_R and PSNR_Q obtained when incorrect keys were used for image recoveries.

Fig. 9 shows three sets of the results, where Fig. 9(c), (f), and (i) shows the results where the parameter randomization technique was applied, while the other six images show the results where mapping randomization was applied. As can be seen from Fig. 9(a)–(c), the watermarked images are visually close to the respective benchmark images, and the translucent color watermarks are distinctive in the watermarked images. There is some noise in the watermarking area of the watermarked images (yielded by large values of $|b - q|$) due to bad color estimations (with large values of $|a - p|$), which happen at edges in the images. The noise is scattered in the watermarking area when the mapping randomization technique was used, and coincides with the edges in the images when parameter randomization was applied.

The images recovered with correct keys are shown in Fig. 9(d)–(f). As expected, the pixels of the recovered images are exactly identical to those of the original images.

The robustness of the mapping randomization technique against illicit recoveries is evident as shown by the low PSNR_Q in Fig. 8. This comes from the fact that the incorrect recovery of one pixel value affects subsequent color estimations around that pixel. This *error avalanche* can be visually seen as patches of blurry noise in illicitly recovered images, as shown in Fig. 9(g)–(h). On the other hand, the parameter randomization technique is weaker against illicit recoveries, especially in regions where the watermark has low opacity.



(a)

(b)

Fig. 8. Average values of PSNR_W obtained after watermark embedding and average values of PSNR_R and PSNR_Q obtained after illicit image recoveries. (a) Results yielded by parameter randomization. (b) Results yielded by mapping randomization.

A comparison of the capabilities of the proposed reversible visible watermarking method with those of four recently-published techniques is shown in Table II. All but Hu [10] allows lossless recovery of the original image. Only Hu [10] and this study reported the *PSNR* for attempted recoveries using incorrect keys, and our results are better. In more detail, we embedded binary transparent watermarks similar to those used in Hu [10] using the proposed method, and obtained much better results (very low values of *PSNR* in the range of 12–14 dB) than Hu's (37–39 dB). More importantly, the proposed approach allows embedding of arbitrary-sized watermarks and has wider applicability than all four methods.

VII. Conclusions and Suggestions for Future Work

In this paper, a new method for reversible visible watermarking with lossless image recovery capability has been proposed. The method uses one-to-one compound mappings that can map image pixel values to those of the desired visible watermarks. Relevant lemmas and theorems are described and proved to demonstrate the reversibility of the compound mappings for lossless reversible visible watermarking. The compound mappings allow different types of visible watermarks to be embedded, and two applications have been described for embedding opaque monochrome watermarks as well as translucent full-color ones. A translucent watermark is clearly visible and visually appealing, thus more appropriate than traditional transparent binary watermarks in terms of advertising effect and copyright declaration. The two-fold monotonically increasing property of compound mappings was defined and an implementation proposed that can provably allow mapped values to always be close to the desired watermark if color estimates are accurate. Also described are parameter randomization and mapping randomization techniques, which can prevent illicit recoveries of original images without correct input keys. Experimental results have demonstrated the feasibility of the proposed method and the effectiveness of the proposed security protection measures.

Future research may be guided to more applications of the proposed method and extensions of the method to other data types other than bitmap images, like DCT coefficients in JPEG images and MPEG videos.

Appendix A
Proof of Theorem 2

Theorem 2 (Two-Fold Monotonically Increasing): If F_x is a one-fold monotonically increasing one-to-one function with a parameter x, then the compound mapping $q = F_b^{-1}(F_a(p))$ is two-fold monotonically increasing.

Proof: In the beginning, we prove that for all values of a, p_1, and p_2, $F_a(p_1) < F_a(p_2)$ if $|a-p_1| < |a-p_2|$ by showing that both the inequality (i) $F_a(p_1) > F_a(p_2)$ and the equality (ii) $F_a(p_1) = F_a(p_2)$ are impossible if $|a-p_1| < |a-p_2|$. First, (i) is impossible by the definition of one-fold monotonically increasing function (Definition 1), since if not so, it will then imply that $|a-p_2| \leq |a-p_1|$, which is a contradiction. Next, (ii) is also impossible because F_x is a one-to-one function, implying $p_1 = p_2$, which contradicts the condition $|a-p_1| < |a-p_2|$. This completes the first part of the proof that

$$F_a(p_1) < F_a(p_2) \quad \text{if } |a-p_1| < |a-p_2|.$$

In the second part of proof, by regarding $F_a(p_1)$ as p_1' and $F_a(p_2)$ as p_2', and substituting them into the inequalities of Lemma 3, we reach the fact that for all values of a, b, p_1, and p_2

$$\left|b - F_b^{-1}(F_a(p_1))\right| \leq \left|b - F_b^{-1}(F_a(p_2))\right| \quad \text{if } F_a(p_1) < F_a(p_2).$$

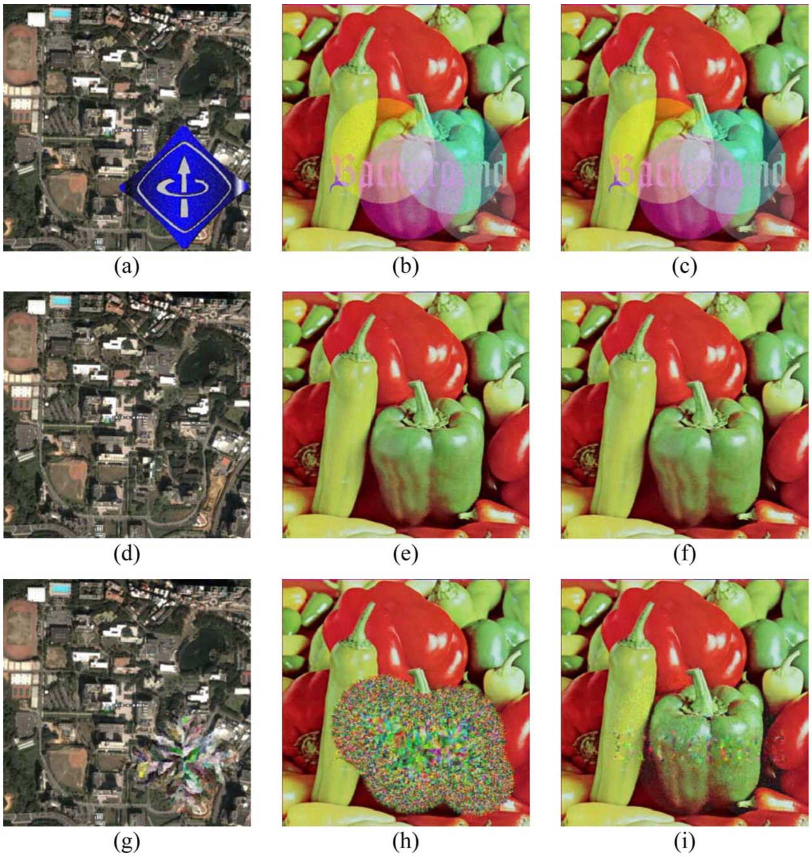


Fig. 9. Watermarked images, licitly recovered images, and illicitly recovered images. (a)–(c) Watermarked images. (d)–(f) Licitly recovered images from images (a)–(c), respectively. (g)–(i) Illicitly recovered images from images (a)–(c), respectively.

TABLE II
COMPARISON OF REVERSIBLE VISIBLE WATERMARKING TECHNIQUES

Method	Legitimate recovery	Illegitimate recovery	Watermark size	Binary transparent watermark	Binary opaque watermark	Color translucent watermark
Hu [10]	43~44 dB	37~39dB	Unlimited	Yes	–	–
Hu [11]	Lossless	Not reported	Limited	Yes	–	–
Tsai [18]	Lossless	Not reported	Limited	Yes	–	–
Yip [19]	Lossless	Not reported	Unlimited	Yes	Yes	–
Proposed	Lossless	12~14dB	Unlimited	Yes	Yes	Yes

Combining the results of the two parts of proof above, we have

$$\left|b - F_b^{-1}(F_a(p_1))\right| \leq \left|b - F_b^{-1}(F_a(p_2))\right| \quad \text{if } |a-p_1| < |a-p_2|$$

or equivalently

$$|b-q_1| \leq |b-q_2| \quad \text{if } |a-p_1| < |a-p_2|$$

where $q_1 = F_b^{-1}(F_a(p_1))$, and $q_2 = F_b^{-1}(F_a(p_2))$. That is, the two-fold monotonically increasing property holds. This completes the proof. □

APPENDIX B
PROOF OF MONOTONICITY PROPERTY OF ALGORITHM 4 AND CORRECTNESS OF ALGORITHM 5

Lemma 4: The function described by Algorithm 4 is one-to-one and one-fold monotonically increasing.

Proof: In Step 4 of Algorithm 4, we always remove a unique element from the set S and in turn increment p', and so each of the 256 possible input values of p will yield its own unique output p' value. Thus, Algorithm 4 indeed describes a one-to-one function for all values of a.

Furthermore, since we remove values of r from S in an increasing order of $|a-r|$, a larger value of p' means that r is farther away from a. This means that the value of $p' = F_a(p)$ yielded by Algorithm 4 satisfies the one-fold monotonically increasing property: $F_a(p) < F_a(p')$ implies $|a-p| \leq |a-p'|$. □

Lemma 5: The function described in Algorithm 5 is the inverse of the function described in Algorithm 4.

Proof: If we set the value of input b in Algorithm 5 to be the input a in Algorithm 4, then the set S in Algorithms 4 and 5 will always contain exactly the same elements for each iteration. This is because in each iteration the value r picked by Step 3 of Algorithm 4 will be the same as the value p picked by Step 2 of Algorithm 5, and this same value is removed respectively in Step 4 of Algorithm 4 and Step 3 of Algorithm 5.

For all values of input p', Algorithm 5 will pick the $(p'+1)$th item in the sequence of $p's$ computed in Step 2 as the final output p, which we denote as $p*$. Since the sequence of p's selected in Step 2 of Algorithm 5 is exactly identical to the sequence of r's picked in Step 3 of Algorithm 4, if the value $p*$ is used as the input p to Algorithm 4, then r will match $p*$ exactly after $(p'+1)$ iterations. Algorithm 4 will, therefore, output the same p' value, demonstrating that the function it described is the inverse of that described by Algorithm 5. Since the functions described by the two algorithms are one-to-one, the function described by Algorithm 5 is the inverse of that described by Algorithm 4. This completes the proof. □

Acknowledgment

The authors would like to thank the reviewers for their very helpful comments and suggestions which have helped improve the overall organization and clarity of the paper.

References

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[2] N. F. Johnson, Z. Duric, and S. Jajodia*, Information Hiding. Steganography and Watermarking—Attacks and Countermeasures*. Boston, MA: Kluwer, 2001.

[3] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Jun. 1997.

[4] M. S. Kankanhalli, Rajmohan, and K. R. Ramakrishnan, "Adaptive visible watermarking of images," in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, 1999, vol. 1, pp. 568–573.

[5] Y. Hu and S. Kwong, "Wavelet domain adaptive visible watermarking," *Electron. Lett.*, vol. 37, no. 20, pp. 1219–1220, Sep. 2001.

[6] S. P. Mohanty, K. R. Ramakrishnan, and M. S. Kankanhalli, "A DCT domain visible watermarking technique for images," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Jul. 2000, vol. 2, pp. 1029–1032.

[7] G. Braudaway, K. A. Magerlein, and F. Mintzer, "Protecting publicly available images with a visible image watermark," in *Proc. SPIE Int. Conf. Electronic Imaging*, Feb. 1996, vol. 2659, pp. 126–133.

[8] Y. J. Cheng and W. H. Tsai, "A new method for copyright and integrity protection for bitmap images by removable visible watermarks and irremovable invisible watermarks," presented at the Int. Computer Symp.—Workshop on Cryptology and Information Security, Hualien, Taiwan, R.O.C., Dec. 2002.

[9] P. M. Huang and W. H. Tsai, "Copyright protection and authentication of grayscale images by removable visible watermarking and invisible signal embedding techniques: A new approach," presented at the Conf. Computer Vision, Graphics and Image Processing, Kinmen, Taiwan, R.O.C., Aug. 2003.

[10] Y. Hu, S. Kwong, and J. Huang, "An algorithm for removable visible watermarking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 129–133, Jan. 2006.

[11] Y. Hu and B. Jeon, "Reversible visible watermarking and lossless recovery of original images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 11, pp. 1423–1429, Nov. 2006.

[12] B. Macq, "Lossless multiresolution transform for image authenticating watermarking," presented at the European Signal Processing Conf., Tampere, Finland, Sep. 2000.

[13] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding—New paradigm in digital watermarking," *J. Appl. Signal Process.*, vol. 2002, no. 2, pp. 185–196, Feb. 2002.

[14] M. Awrangjeb and M. S. Kankanhalli, "Lossless watermarking considering the human visual system," presented at the Int. Workshop on Digital Watermarking, Seoul, Korea, Oct. 2003.

[15] M. Awrangjeb and M. S. Kankanhalli, "Reversible watermarking using a perceptual model," *J. Electron. Imag.*, vol. 14, no. 013014, Mar. 2005.

[16] C. de Vleeschouwer, J. F. Delaigle, and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 97–105, Mar. 2003.

[17] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[18] H. M. Tsai and L. W. Chang, "A high secure reversible visible watermarking scheme," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Beijing, China, Jul. 2007, pp. 2106–2109.

[19] S. K. Yip, O. C. Au, C. W. Ho, and H. M. Wong, "Lossless visible watermarking," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Jul. 2006, pp. 853–856.

[20] A. Lumini and D. Maio, "Adaptive positioning of a visible watermark in a digital image," in *Proc. Int. Conf. Multimedia and Expo*, Taipei, Taiwan, R.O.C., Jun. 2004, pp. 967–970.

[21] X. Li and M. T. Orchard, "Edge-directed prediction for lossless compression of natural images," *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 813–817, Jun. 2001.

Tsung-Yuan Liu (S'04) received the B.S. degree in electrical engineering from the University of the Witwatersrand, Johannesburg, South Africa, and the M.B.A. degree from the National Taiwan University, Taipei, Taiwan, R.O.C. He is currently pursuing the Ph.D. degree at the College of Computer Science, National Chiao Tung University, Hsinchu, Taiwan.

He is a Software Engineer with Google, Taipei. His research interests include information hiding, image processing, web search, data mining, and artificial intelligence.

Wen-Hsiang Tsai (SM'91) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., the M.S. degree from Brown University, Providence, RI, and the Ph.D. degree from Purdue University, West Lafayette, IN.

Currently, he is a Chair Professor with the Department of Computer Science, National Chiao-Tung University, Hsinchu, Taiwan, and was the President of Asia University, Taichung, Taiwan. So far, he has published 135 journal papers and 220 conference papers. His research interests include image processing, computer vision, information security, and autonomous vehicle applications.