2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing

Dual-watermarking by QR-code Applications in Image Processing

Fu-Hau Hsu
Department of Computer Science and Information Engineering,
National Central University
Taoyuan 320, Taiwan
hsufh@csie.ncu.edu.tw

Min-Hao Wu
Department of Computer Science and Information Engineering,
National Central University
Taoyuan 320, Taiwan
mhwu@csie.ncu.edu.tw

Shiuh-Jeng WANG*
Department of Information Management, Central Police University
Taoyuan 320, Taiwan
sjwang@mail.cpu.edu.tw

***Abstract*—Digital watermarking has recently emerged as a solution to the problem of providing guarantees about copyright protection of digital images. However, several problems related to the robustness of invisible watermarking techniques from malicious or non-malicious attacks still remain unsolved. Visible watermarking is an effective technique for preventing unauthorized use of an image, based on the insertion of a translucent mark, which provides immediate claim of ownership. Digital watermarking technology primarily joins the rightful owner of totem to the protected media. Once the media are suspected to be illegally used, an open algorithm can be used to extract the digital watermark, for the purpose of showing the media's ownership. A reversible visible watermarking scheme is proposed to satisfy the applications, in which the visible watermark is expected to combat copyright piracy but can be removed to recover the original image without loss. In this paper, we propose a reversible visible watermark method, which embeds QR code into gray-scale images to create a visible watermark. Not using complex calculations, this paper tries to simply change the pixel value to achieve the digital watermark. Furthermore, a reversible steganographic method is used to embed the watermarking information, which can be used to recover the original images, into the watermarking images.**

Keywords-reversible; watermarking; histogram; QR cord; data hiding

I. INTRODUCTION

QR Code (Quick Response Code) is the trademark for a type of matrix barcode (or two-dimensional code). More recently, the system has become popular outside of the industry due to its fast readability and large storage capacity compared to standard UPC barcodes. The code consists of black modules arranged in a square pattern on a white background. The information encoded can be made up of four standardized types of modes of data that are similar to numeric, alphanumeric, byte or binary, Kanji, or through supported extensions, or virtually any kind of data. Therefore, the QR code has become a focus of advertising strategy, since it provides quick and effortless access to the brand's website. Beyond mere convenience to the consumer, the importance of this capability is that it increases the conversion rate, by coaxing qualified prospects further down the conversion funnel without any delay or effort, bringing the viewer to the advertiser's site immediately, where a longer and more targeted sales pitch may continue. As users who take the mobile phone camera to the bar code and via the mobile phone software decoder, we can directly obtain information such as URLs, text data, images, and significant savings in the time of input. In addition to text information, picture information (such as maps) can also be directly entered the mobile phone through this function. People have made linking to web pages easy by using PC, PDA, iPAD or smart phones over the wireless networks. Users have become accustomed to browsing web pages or Apps (Application) with a smart phone, bringing more conveniences to daily life. The mobile device has also input a Uniform Resource Locator (URL), bringing more difficulties. In 2008, Tan [1] presented a method that uses the quick response code (QR code) to solve difficulties of the input problem. The QR code can be used as a visual watermark embedded into the gray-image, and using reversible data hiding provides the ability to restore the original image.

Data hiding imperceptibly embeds data into a cover media so that messages can be delivered secretly. Digital images are often used as carriers to deliver such messages. A cover image is an image that is used to carry data, and a stego-image is the image that carried data. Embedding distortion occurs when data are embedded into a cover image. Many approaches to information hiding have been proposed for different attributes, such as capacity, imperceptibility, undetectability, robustness, and reversibility. These attributes are used for various applications, such as secret communication, copyright protection, tampering detection, and other human-centered approaches. Although traditional data hiding techniques have been able to make secret data imperceptible to attackers, distortions of cover media, caused by the hiding process, have often been inevitable and irreversible. For this reason, researchers have focused their attention on developing reversible data hiding methods. Reversibility allows original media to be completely recovered from the stego-media after the embedded message is extracted. Many reversible data hiding approaches have been proposed [2-19]. According to where the data are embedded, these approaches can be classified into two categories: the spatial domain [2, 3-14], and other compression types, such as vector quantization (VQ) [15-19].

In those developed reversible data hiding methods, four main technologies have been widely applied: the compression-based technology [2-5], the difference-expansion-based technology [6-7], the histogram-based technology [8-14] and visible watermarking [19-24]. For compression-based technology, Fridrich et al. compressed the least significant bit (LSB) plane to obtain additional space for embedding secret data [2, 3] in 2002. In 2004, Awrangjeb and Kankanhalli [4, 5] presented a scheme that detects the textured blocks, extracts the LSBs of the pixel-values from these textured blocks based on the Human Visual System (HVS), and concatenates the authentication information with the compressed bit-string. In 2003, for difference-expansion-based technology, Tian [6] proposed a reversible data hiding method based on difference expansion. They partitioned the cover image into pixel pairs and embedded a message bit into the LSB of the expanded difference. Tian's method provides high payload; however, the distortion caused by difference expansion is significant. In 2004, Alattar [7] used the difference expansion of vectors of adjacent pixels to obtain more embedding space. In 2007, Weng et al. [8, 9] used the correlations among four pixels in a quad and embedded data by expanding the differences between one pixel and each of its three neighboring pixels. For histogram-based technology, Ni et al. [10] presented a reversible data hiding method based on the histogram in 2006. In their method, histogram bins are shifted to vacate an empty bin for data embedment. Ni et al.'s method provides excellent image quality; however, the payload is limited by the peak height of the image histogram. Their method used the pixel values in the original image to create the histogram, but the peak points of the histogram are not high enough. Some methods used predictive concepts to increase the peak highs [11-14]. In 2009, Tsai et al. [11] proposed a reversible data hiding method based on histogram shifting. Their method partitioned the cover image into blocks of n × n pixels, and the differences between the center pixel and other pixels are calculated, respectively. Data are then embedded by modifying the differences. They employed the histogram of the pixel difference sequence to increase the embedding capacity. For vector quantization technology, the modified fast correlation VQ (MFCVQ) was proposed by Yang et al. [15] in 2005. In 2009, Lu et al. [16] provided a VQ reversible data hiding method with the MFCVQ-based approach. Recently, many scholars used the four adjacent blocks of the current block to embed secret data and achieve reversibility in the VQ domain [15, 17, 18].

For visible-watermarking-based technology, the existence of the watermark can be judged by human eyes [19-22]. Embedding watermarks visibly will degrade the quality of the host images, of course. However, reversible techniques can be applied to allow legitimate users to remove the embedded visible watermark and restore the original images. Several reversible visible watermarking techniques have been proposed [20, 24]. One common kind of approach is to compress a portion of the original image and then embed the compressed data together with the intended payload into the image [20]. Another kind of approach 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 [24].

In this paper, a new reversible visible watermarking method is proposed, in which binary images are used as the watermark. The embedding approach for the visible watermark is based on the pixel modifications. After being embedded, the watermark can be clearly distinguished by human eyes. Moreover, some information, including the watermark, is imperceptibly embedded into the stego-image by a reversible steganographic approach. The information can be extracted and used to completely recover the original image. The remainder of this paper is organized as follows. Some related works are reviewed in Section 2. Our proposed approach is shown in Section 3. Section 4 gives some experimental results. Finally, we draw some conclusions in Section 5.

II. RELATED WORKS

In this section, we review Hu and Jeon's [20] data hiding and visible watermark method and Yang and Tsai's [14] histogram-based reversible data hiding methods.

A. Hu and Jeon's method

In this section, Hu and Jeon [20] first proposed a reversible visible watermarking scheme by modifying one significant bit plane of the pixels of the host image. They achieved reversibility via hiding the compressed version of the altered bit plane without loss into the non-watermarked image region. However, the embedded visible watermark with this method appears to be somewhat blurred, and the visual quality of the original image is significantly distorted. There are two procedures in this algorithm: data hiding and visible watermark embedding. Fig. 1 illustrates the framework of the proposed algorithm. W and R denote the visible watermark and the image region to be protected, respectively. W has the same size as R. To achieve lossless recovery of the image I, the bit plane of R must be preserved in the non-watermarked image area $I-R$ before W is embedded into a bit plane of R. In Stage 1, the bit plane of R constitutes the pixel set D. A bit-plane data usually has a statistical structure, so D is further compressed into Dc in Stage 2 using the open C code of JBIG-KIT.

In Stages 3 and 4, a payload-adaptive scheme is performed to constitute a subset S of the lowest bit plane in $\mathrm{I}-R$. S satisfies the constraint $|Dc| = |S| - |Sc|$, where Sc is the compressed result of S and the operator$|.|$ represents the length of a sequence or the cardinality of a set. In Stages 5 and 6, $H = Sc \cup Dc$ is hidden into image area $I-R$ by directly replacing S with H. Finally, in Stage 7, W is embedded into R by replacing the bit plane directly. To summarize, the above preprocessing and data hiding processes are described in the following steps.

Input: cover-image I, watermarking W.

Step 1: To determine the watermarking region R and denote R_D, which is the bit plane of R to be altered by W.
Step 2: To compress D by JBIG and generate D_C, which will be embedded in $I-R$.
Step 3: To choose the compression tools version of the CACM. To determine the key bit plane I_K by each bit plane from the LSB to the MSB plane of the $I-R$.

Step 4: To find out S via Stem, which is composed of the one-bit pixel on the I_K. The constitution of S_{tem} starts at $|S_{tem}| = 1$.
Step 5: If S is found, then compress S_{tem} before each addition and stratify $|DC| = |S_{tem}| - |S_{tem,c}|$.
If not, repeat step 5.
Step 6: Construct the payload bit stream as $H = S_C \cup D_C$. Replace S with H to create $(I - R)_m$.

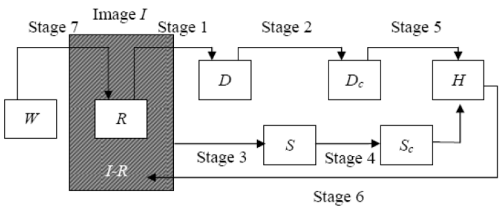


Fig. 1. The framework of Hu and Jeon's algorithm

B. *Yang and Tsai's method*

In 2010, Yang and Tsai [14] provided interleaving prediction methods to enhance the histogram-based reversible hiding approach on gray images. The predictive difference values are transformed into histograms to create higher peak values. For one level data hiding, the change of each pixel between the original image and the stego-image remains within ±1. Fig. 2(a) shows their column interleaving prediction. Pixels in odd columns will be predicted by pixels in even columns, then pixels in even columns are predicted by pixels in odd columns, and vice versa. In the embedding process, predictive difference values of odd columns are used to generate a histogram to embed secret data. Then predictive difference values of even columns are used. In the extracting and reversing process, predictive difference values of even columns are processed first. Then predictive difference values of odd columns are processed. Fig. 2(b) shows their chessboard interleaving prediction. An image is considered as a chessboard containing white pixels and black pixels. In the first stage, white pixels are processed and predicted by their neighboring black pixels. Then, in the second stage, black pixels are processed and predicted by their neighboring white pixels. The predictive difference value is calculated by the difference between a pixel value and the averaged pixel value of its neighboring pixels.

(a) (b)

Fig. 2. Two interleaving-prediction strategies: (a) Column-based; (b) Chessboard-based

Suppose that the original image is a 512 × 512 gray image. $I_{i,j}$ and $I'_{i,j}$ are the pixel values at location (i, j) before and after being processed, respectively. $D_{i,j}$ is the predictive error value at location (i, j), and $D'_{i,j}$ is the embedded predictive error value at location (i, j). The embedding algorithm is as follows.

Input: Original image, secret message.
Output: Stego-image, two pairs of peak and zero points.

1. Input the original image I.
2. The predictive difference value $D_{i,j}$ will use pixels in even columns in the odd columns.
   if j=1 then $D_{i,j} = I_{i,j} - I_{i,j} + 1$
   if j mod 2 = 1 then $D_{i,j} = I_{i,j} - \left\lfloor \frac{I_{i,j-1} + I_{i,j+1}}{2} \right\rfloor$
3. Create the histogram $H(x)$ with $x \in [-255, 255]$ from all prediction values.
4. Find two pairs of peak and zero points.
   (a) Select the two highest peak points from $H(x)$, where $P_1 > P_2$.
   (b) Select two zero points Z_1 and Z_2 from $H(x)$ with $x \in [P_1+1, 255]$ and $x \in [-255, P_2 - 1]$.
5. Shift the histogram
   (a) if $x > P_1$ and $x < Z_1$ then move the whole part of the histogram $H(x)$ to the right by one unit.
   (b) if $x < P_2$ and $x > Z_2$ then move the whole part of the histogram $H(x)$ to the left by one unit.
6. Embed a secret bit if the predictive difference value $D_{i,j}$ is equal to P_1 and P_2 and then put embedded bit in the $D'_{i,j}$.
7. To convert the predictive difference value $D_{i,j}$, use pixels in even columns in the odd columns.
   if j=1 then $I'_{i,j} = D'_{i,j} + I_{i,j}$
   if j mod 2 = 0 then $I'_{i,j} = D_{i,j} + \left\lfloor \frac{I_{i,j-1} + I_{i,j+1}}{2} \right\rfloor$
8. Similar to Steps 2−6, embed the message by these predictive difference values in even columns.
9. Output the stego-image and two pairs of peak and zero points.

III. OUR PROPOSED VISUAL WATERMARK METHOD

This study proposes a visual watermark method of embedding black-and-white binary images into the target gray-scale images. The embedding method changes pixel values by adding positive random values to them, such that the changed results are visible. After that, the watermark information is hidden into the gray image by a reversible steganographic method. The watermark information can be extracted to completely recover the watermarked images. Our watermark embedding and removing procedures are introduced in the following subsections.

A. *Visible watermark embedding*

This study proposes a visual watermark method by embedding QR code into the target gray-scale images. The embedding method changes pixel values by giving minus positive values to them, such that the changed results are visible in mobile phone scanning. After that, the QR code information is hidden into the gray image by a reversible steganographic method. The QR code information can be extracted to completely recover the watermarked images. Our watermark embedding and removing procedures are introduced in the following subsections.

The original image and digital watermark are defined as $I = \{I(i, j) \mid 0 \le i < M_1, 0 \le j < M_2\}$, and $Q = \{Q(i, j) \mid 0 \le i < N_1, 0 \le j < N_2\}$,where I denotes the original image, Q denotes the QR code image, and (i, j) denotes the pixel coordinate. The original image I is a $M_1 \times M_2$ gray-scale image, the watermark image Q is a $N_1 \times N_2$ QR code image with white background pixels denoted as 1 and black logo pixels denoted as 0, and $N_i \le M_i$ for i = 1, 2. Fig. 3 shows the flowchart of our embedding algorithm. In Stage 1, the embedding region is selected and is simply represented by a coordinate where watermark Q begins to be embedded. In Stage 2, the watermark Q is visibly embedded into image I to form watermarked image I'. Then, in Stage 3, bit-plane data usually has a statistical structure, so Q is further compressed into Q' using the open C code of JBIG-KIT so that the watermark Q' and the region information are invisibly embedded into image I' to form the reversible watermarked image I''. For this QR code to be successfully read out of the mobile, we will set a threshold T and a threshold value equal to 32 and record them to 3-bit flag string F. The detailed embedding algorithm is as follows:

Input: Original image I, and QR code Q.
Output: Reversible watermarked image I''
Step 1: Select a coordinate L of image I where watermark Q begins to be embedded.
Step 2: Read the pixel value $Q(i, j)$ of QR code Q. Let $I(k, l)$ be the pixel into which watermark bit $Q(i, j)$ will be embedded.
    if $Q(i, j) = 0$
        if ($I(k, l) \geqq T$ and $I(k, l) < 2T$)
            flag string $F = F \| 001_{(2)}$
            $I'(k, l) = I(k, l) - T$
        else if ($I(k, l) \geqq 2T$ and $I(k, l) < 3T$)
            flag string $F = F \| 010_{(2)}$
            $I'(k, l) = I(k, l) - 2T$
        else if ($I(k, l) \geqq 3T$ and $I(k, l) < 4T$)
            flag string $F = F \| 011_{(2)}$
            $I'(k, l) = I(k, l) - 3T$
        else if ($I(k, l) \geqq 4T$ and $I(k, l) < 5T$)
            flag string $F = F \| 100_{(2)}$
            $I'(k, l) = I(k, l) - 4T$
        else if ($I(k, l) \geqq 5T$ and $I(k, l) < 6T$)
            flag string $F = F \| 101_{(2)}$
            $I'(k, l) = I(k, l) - 5T$
            flag string $F = F \| 101_{(2)}$
        else if ($I(k, l) \geqq 6T$ and $I(k, l) < 7T$)
            flag string $F = F \| 110_{(2)}$
            $I'(k, l) = I(k, l) - 6T$
        else if ($I(k, l) \geqq 7T$ and $I(k, l) < 8T-1$)
            flag string $F = F \| 111_{(2)}$
            $I'(k, l) = I(k, l) - 7T$
        else
            $I'(k, l) = I(k, l)$;
    else if ($Q(i, j) = 1$)
        if ($I(k, l) \leqq 128$)
            flag string $F = F \| 000_{(2)}$
            $I'(k, l) = I(k, l) + T$

Step 3: Use BIG-KIT to compress Q into Q'. For Q', L and F, use a Yang and Tsai [14] method to embed them into the image I to construct image I''.
Step 4: Output image I''.

Fig. 3. The flowchart of our embedding algorithm.

B. Watermark removing and image reconstruction

Fig. 4 shows the flowchart of our watermark removing algorithm. In Stage 1, the embedded information Q', L and F in image I'' is extracted and image I'' is recovered into image I'. Then, in Stage 2, the information Q, L and F is used to recover image I from image I'. The detailed algorithm is as follows.

Input: Image I''
Output: Original image I and QR code image Q.
Step 1: Use the reversible steganographic method to extract compressed watermark Q', binary string F, and the beginning coordinate L and recover image I' from the image I''. Use BIG-KIT to decompress Q' into Q.
Step 2: Remove each watermark bit $Q(i, j)$ from image I' to get image I as follows:
Let it be the pixel into which watermark bit $Q(i, j)$ has been embedded and f be the last three bits of F corresponding to $I'(k, l)$.
if $Q(i, j) = 0$, then
    if $f = 001_{(2)}$, then $I(k, l) = I'(k, l) + T$
    else if $f = 010_{(2)}$, then $I(k, l) = I'(k, l) + 2T$
    else if $f = 011_{(2)}$, then $I(k, l) = I'(k, l) + 3T$
    else if $f = 100_{(2)}$, then $I(k, l) = I'(k, l) + 4T$
    else if $f = 101_{(2)}$ then $I(k, l) = I'(k, l) + 5T$
    else if $f = 110_{(2)}$ then $I(k, l) = I'(k, l) + 6T$
    else if $f = 111_{(2)}$ then $I(k, l) = I'(k, l) + 7T$
    else $I(k, l) = I'(k, l)$
else if $Q(i, j) = 1$ and $f = 000_{(2)}$ then $I(k, l) = I'(k, l) - T$
Step 3: Output watermark Q and original image I.

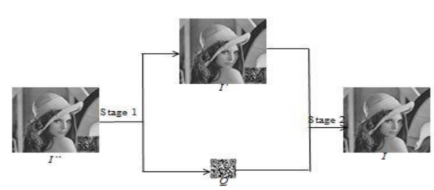


Fig. 4. The flowchart of our watermark removing algorithm

IV. Experiments

In this section we perform some experiments for our method. Fig. 7 shows an experiment of two QR code images of size 135 × 135. Four gray-scale images of size 512 × 512 are shown in Fig. 8. We choose four test images, namely, (a) airplane, (b) baboon, (c) Lena, and (d) pepper, and the image quality of the stego-image is evaluated by the peak signal to noise ratio (PSNR) which is defined as:

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2}{\mathrm{MSE}} (\mathrm{dB}),$$

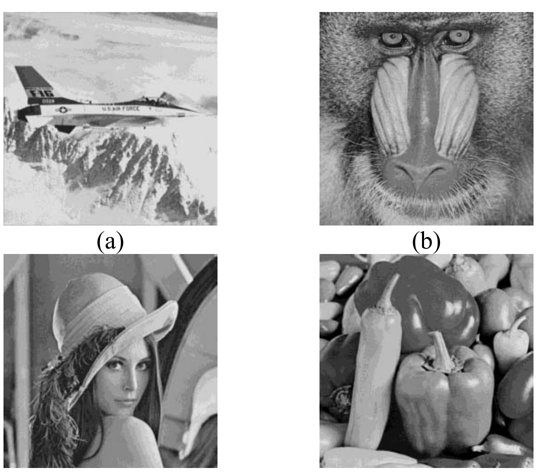
where MSE is the mean square error between the original image and the stego-image. For a 512×512 gray image, the MSE is defined as:

$$\mathrm{MSE} = \frac{1}{512 \times 512} \sum_{i=0}^{511} \sum_{j=0}^{511} (I_{i,j} - I'_{i,j})^2,$$

where $I_{i,j}$ and $I''_{i,j}$ are the pixel values of the original image and the stego-image, respectively.

(a) (b)

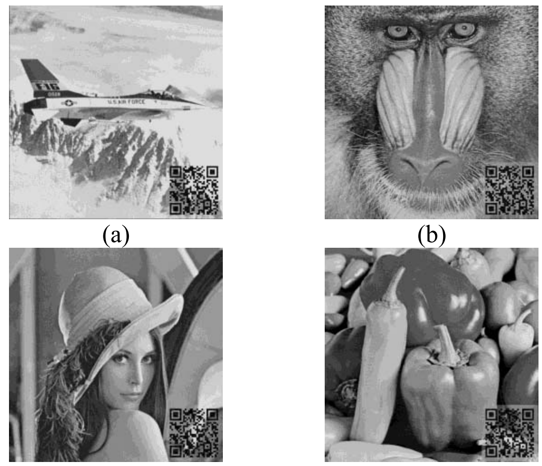
Fig. 7. The experimental QR code URL information of the website of (a) http://in2.csie.ncu.edu.tw/~hsufh/, and (b) http://hera.im.cpu.edu.tw/sjw_2006/



(a) (b) (c) (d)

Fig. 8. The experimental cover images: (a) Airplane; (b) Baboon; (c) Lena; (d) Pepper

Fig. 9 shows the reversible image watermarked with QR code information URL into http://in2.csie.ncu.edu.tw/~hsufh/ with the results clearly visible. Yang and Tsai's reversible data hiding method is used to embed the information of Q and F. In Table 1, PSNR (Peak Signal to Noise Rations) values between two of the images I, I', and I'' are used to assess the similarity between them where there is QR code information of (i) http://in2.csie.ncu.edu.tw/~hsufh/ and (ii) http://hera.im.cpu.edu.tw/sjw_2006/. The QR code is a complex binary image that, through our algorithm embedded in the QR code in the smooth image, has a lower PSNR than a complex image.



(a) (b) (c) (d)

Fig. 8. The experimental stego images: (a) Airplane; (b) Baboon; (c) Lena; (d) Pepper with QR code information URL of the website of http://in2.csie.ncu.edu.tw/~hsufh/

Table 1. The PSNR values between two of the images I, I', and I''

Images	Watermark sizes	QR code information	PSNR		
			I' to I	I'' to I'	I'' to I
Airplane	135×135	(i)	20.18	49.69	20.15
		(ii)	20.72	49.67	20.68
Baboon	135×135	(i)	21.96	48.51	21.91
		(ii)	23.54	48.51	23.47
Lena	135×135	(i)	21.77	49.56	21.73
		(ii)	22.11	49.55	22.06
Pepper	135×135	(i)	23.06	49.34	23.01
		(ii)	24.94	49.34	24.87

Table 2 shows a comparison between our and Huang et al.'s [26] methods. In our scheme, three gray-scale images of size 512 × 512 are used and the QR code size of the binary image with the size of 135 × 135 is produced. The QR code can be inserted into any corner or the center of the original image. Table 2 shows the results inserting the QR code at the lower-right. In Huang et al.'s method, an image size of 1024 × 1024 was used and a binary image of the QR code with size 135 × 135 is produced. The QR code is inserted into the corner of the original image by directly replacing the pixel values at the lower-right portion of the original image. The PSNR value of our scheme is close to that of Huang et al.'s method. Nevertheless, our test image size is smaller than that of Huang et al.

Table 2. The comparison between ours and Huang et al.'s [26] methods

Test Image	Lena	Baboon	Pepper
QR-code Image Size	135×135	135×135	135×135
QR-code Information	http://www.lenna.org	http://www.google.com	http://www.yahoo.com.tw
Huang et al.[26] of PSNR	22.81	21.71	21.59
Our scheme of PSNR	21.24	21.61	22.59

While we should be hiding the QR-code and additional data with reversible data hiding, the fact that our algorithm is capable of hiding such an amount of data allows the test image size to be 512 × 512. Furthermore, we hid the data with an average utility of capacity of 0.27 bpp (bits per pixel), which is higher than that of Huang et al.'s method of the DEQ scheme with 0.18 bpp. The hiding done with the reversible data hiding of our scheme's average bits is 21,539 bits lower than Huang et al.'s method, which is 54,675 (135×135×3) bits.

V. CONCLUSIONS

In this paper, a visible watermarking technology is proposed to embed QR code images into gray-scale images. In addition, embedding the information of the watermark into the watermarked image by a reversible steganographic method can successfully recover the original image. The image quality of the reversible watermarked image is similar to the watermarked image. Experimental results show that our method can clearly display a watermark and can completely remove the watermark.

ACKNOWLEDGMENT

This research was partially supported by the National Science Council of the Republic of China under the Grants NSC 100-2221-E-008-077-, NSC 100-2218-E-008-013-MY3, NSC 98-2221-E-015-001-MY3, and NSC 100-2221-E-015 - 001 -MY2.

REFERENCES

[1] J. S. Tan, "QR code," *Synthesis Journal*, Section 3, 2008, pp. 59-78.

[2] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in *Proceedings of SPIE Security and Watermarking of Multimedia Contents*, 2001, pp. 197-208.

[3] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding-new paradigm in digital watermarking," *EURASIP Journal on Applied Signal Processing*, Vol. 2, 2002, pp. 185-196.

[4] M. Awrangjeb and M. S. Kankanhalli, "Lossless watermarking considering the human visual system," *Lecture Notes in Computer Science*, Vol. 2939, 2004, pp. 329-336.

[5] M. Awrangjeb and M. S. Kankanhalli, "Reversible watermarking using a perceptual model," *Journal of Electronic Imaging*, Vol. 14, 2005, pp. 1-8.

[6] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, 2003, pp. 890-896.

[7] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, Vol. 13, 2004, pp. 1147- 1156.

[8] S. Weng, Y. Zhao, J. S. Pan, and R. Ni, "A novel reversible watermarking based on an integer transform," in *Proceedings of IEEE International Conference on Image Processing*, Vol. 3, 2007, pp. 241-244.

[9] S. Weng, Y. Zhao, J. S. Pan, and R. Ni, "Reversible watermarking based on invariability and adjustment on pixel pairs," *IEEE Signal Processing Letters*, Vol. 15, 2008, pp. 721-724.

[10] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," Circuits and Systems for Video Technology, *IEEE Transactions on*, Vol. 16, no. 3, 2006, pp. 354-362.

[11] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, Vol. 89, no. 6, 2009, pp. 1129-1143.

[12] H. W. Tseng and C. P. Hsieh, "Prediction-based reversible data hiding," *Information Sciences*, Vol. 179, no. 14, pp. 2460-2469, 2009.

[13] X. T. Zeng, L. Ping, and Z. Li, "Lossless data hiding scheme using adjacent pixel difference based on scan path," *Journal of Multimedia*, Vol. 4, no. 3, 2009, pp. 145-152.

[14] C. H. Yang and M. H. Tsai, "Improving histogram-based reversible data hiding by interleaving predictions," *Image Processing, IET*, Vol. 4, no. 4, 2010, pp. 223-234.

[15] B. Yang, Z. M. Lu, and S. H. Sun, "Reversible Watermarking in the VQ-compressed Domain," *Proceedings of the Fifth IASTED International Conference on Visualization, Imaging, and Image Processing(VIIP'2005)*, Benidorm, Spain, 2005, pp. 298-303.

[16] Z. M. Lu, J. X. Wang, and B. B. Liu, "An improved lossless data hiding scheme based on image VQ-index residual value coding," *The Journal of Systems and Software*, Vol. 82, pp. 1016-1024, 2009.

[17] C.C. Chang, T.D. Kieu, and W.C. Wu, "A lossless data embedding technique by joint neighboring coding," *Pattern Recognition*, Vol. 42, no. 7, July 2009, pp. 1597-1603.

[18] Z. M. Lu, J. X. Wang, and B. B. Liu, "An improved lossless data hiding scheme based on image VQ-index residual value coding," *The Journal of Systems and Software*, Vol. 82, pp. 1016-1024, 2009.

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

[20] Y. Hu and B. Jeon, "Reversible visible watermarking and lossless recovery of original images," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, no. 11, Nov. 2006, pp. 1423-1429.

[21] S.D. Lin and S.C. Shie. "Improving robustness of visible watermarking schemes for images," *the 2004 IEEE International Symposium on Consumer Electronics*, 2004, pp.011 -014.

[22] H.M. Tsai and L.W. Chang, "A high secure reversible visible watermarking scheme," *IEEE International Conference on Multimedia and Expo*, 2007, pp. 2106-2109.

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

[24] T.Y. Liu and W.H. Tsai, "Generic Lossless Visible Watermarking—A New Approach," *IEEE Transactions on Image Processing*, vol. 19, May 2010, no. 5.

[25] M.J. Tsai , "A visible watermarking algorithm based on the content and contrast aware (COCOA) technique", *JVCIR*(20), no. 5, July 2009, pp. 323-338.

[26] H.C. Huang, F. C. Chang, and W. C. Fang, "Reversible Data Hiding with Histogram-Based Difference Expansion for QR Code Applications," *IEEE Trans. Consumer Electronics*, Vol. 57, no. 2, May 2011, pp. 779-787.