JOINT BI-WATERMARKING AND HALFTONING TECHNIQUE CAPABILITY FOR BOTH TAMPERED AREAS LOCALIZATION AND RECOVERY OF STILL IMAGE

Soo-Chang Pei[1] and *Yi-Chong Zeng*[2]

[1]Department of Electrical Engineering, National Taiwan University
[2]Institute of Information Science, Academia Sinica
[1]pei@cc.ee.ntu.edu.tw, [2]yichongzeng@iis.sinica.edu.tw

ABSTRACT

This paper presents a method to localize and to recover the tampered areas in the watermarked image simultaneously. First, the proposed watermarking algorithm aims to embed a bi-watermark, which is the binary halftone of the downscaled host image, into the host image. While the watermarked image is tampered unauthorizedly, the extracted bi-watermark can be exploited to localize the tampered areas automatically without initial bi-watermark. Subsequently, Gaussian lowpass filter and quadratic programming are used to restore the bi-watermark to gray-scale image for inverse halftoning procedure. Eventually, the restored gray-scale image is used to recover the tampered areas.

Index Terms**—** image recovery, watermark, inverse halftoning, quadratic programming

1. INTRODUCTION

The pirates illegally copy, tamper and edit the image, therefore, the image authentication becomes an important topic to solve above problems. The researchers invest a lot of efforts to develop various watermarking methods in order to solve the problem of image authentication. Kundur and Hatzinakos [1] presented the fragile watermarking approach, which modifies the wavelet coefficients for watermark embedding. Ko and Part [2] developed a semi-fragile watermarking method in wavelet domain to detect the attacks inflicted on the content Kostopoulos et al. [3] suggested a self-authentication scheme to aim at the authentication of a specific region in the scanned paintings. The proposed solutions utilize the encryption approach and watermark technique. R. Caldelli et al. [4] presented a system to joint the image watermarking and JPEG lossless compression for image authentication and tamper localization. The bi-watermarking algorithm [5] will embed the bi-watermark, which is also the halftones of the host image, into the host image. The halftone image consists of 0- and 1- values, and it is very suitable to be treated as watermark. In our opinion, the halftone image is not considered as the watermark, but also it can be exploited to recover the corrupted image by inverse halftoning techniques.

In the past, the researchers proposed many halftoning approaches, including error diffusion, ordered differ, blue noise mask, dot diffusion, green noise halftoning, direct binary search, etc [6]. The goal of inverse halftoning is to transform the halftone image to the gray-scale or color image. For existing various inverse halftoning approaches, they can be classified into filtering-based and training-based algorithms. The filtering-based algorithms include the Gaussian lowpass filtering approach [7], optimized linear filter [8], the wavelet approach [9], etc. For training-based algorithms, they include the vector quantization technique [10], the lookup table-based approach [11, 12], etc.

This paper is organized as follows. Sections 2 and 3 describe the bi-watermarking algorithm and the proposed inverse halftoning, respectively. Section 4 introduces the tampered areas localization and recovery. Section 5 shows the experimental results. Conclusion is made in Sections 6.

2. BI-WATERMARKING ALGORITHM

2.1. Embedding Procedure

The bi-watermarking algorithm is one kind of quantization index modulation. Assume that an $A\times B$ gray-scale is divided into several $\lfloor A/2\rfloor\times\lfloor B/2\rfloor$ non-overlapping blocks of size 2×2, and then every block is embedded with a 2-bit watermark symbol. The sample $q_{i,j}$ of every block is measured by,

$$q_{i,j} = p_{i,j}(0,0) + p_{i,j}(0,1) + p_{i,j}(1,0) + p_{i,j}(1,1), \tag{1}$$

where $p_{i,j}(u,v)$ denotes the (u,v)-th pixel value of (i,j)-th block. The sample $(q_{i,j})$ is quantized to the proper reconstructed value $(r_{i,j})$ according to the watermark symbol $\mathbf{W}_{\rm p}(i,j)$. Then, the pixels' values of every block are adjusted such that the new sample $q_{i,j}$ will be equal to $r_{i,j}$. The reconstructed value is defined as,

$$r_{i,j} = \begin{cases} r_1 & \text{, if } q_{i,j} < r_2 \text{ and } \mathbf{W}_{\rm p}(i,j) = \mathrm{S}_0 \\ r_3 & \text{, if } r_2 \le q_{i,j} < r_5 \text{ and } \mathbf{W}_{\rm p}(i,j) = \mathrm{S}_0 \\ r_1 + 2Q_1 & \text{, if } q_{i,j} \ge r_5 \text{ and } \mathbf{W}_{\rm p}(i,j) = \mathrm{S}_0 \\ r_6 - 2Q_1 & \text{, if } q_{i,j} < r_2 \text{ and } \mathbf{W}_{\rm p}(i,j) = \mathrm{S}_1 \\ r_4 & \text{, if } r_2 \le q_{i,j} < r_5 \text{ and } \mathbf{W}_{\rm p}(i,j) = \mathrm{S}_1 \\ r_6 & \text{, if } q_{i,j} \ge r_5 \text{ and } \mathbf{W}_{\rm p}(i,j) = \mathrm{S}_1 \\ r_5 - 2Q_1 & \text{, if } q_{i,j} < r_2 \text{ and } \mathbf{W}_{\rm p}(i,j) = \mathrm{U}_0 \\ r_5 & \text{, if } q_{i,j} \ge r_2 \text{ and } \mathbf{W}_{\rm p}(i,j) = \mathrm{U}_0 \\ r_2 & \text{, if } q_{i,j} < r_5 \text{ and } \mathbf{W}_{\rm p}(i,j) = \mathrm{U}_1 \\ r_2 + 2Q_1 & \text{, if } q_{i,j} \ge r_5 \text{ and } \mathbf{W}_{\rm p}(i,j) = \mathrm{U}_1 \end{cases}, \tag{2}$$

where Q_1 and Q_2 are quantization stepsizes. Those two stepsizes yield the non-uniform quantization intervals. Every interval corresponds to an initial reconstructed value and a watermark symbol, and the watermark symbols include stable zero (S_0=<u>00</u>), unstable zero (U_0=<u>01</u>), stable one (S_1=<u>11</u>) and unstable one (U_1=<u>10</u>). Six kinds of the initial reconstructed values are given by,

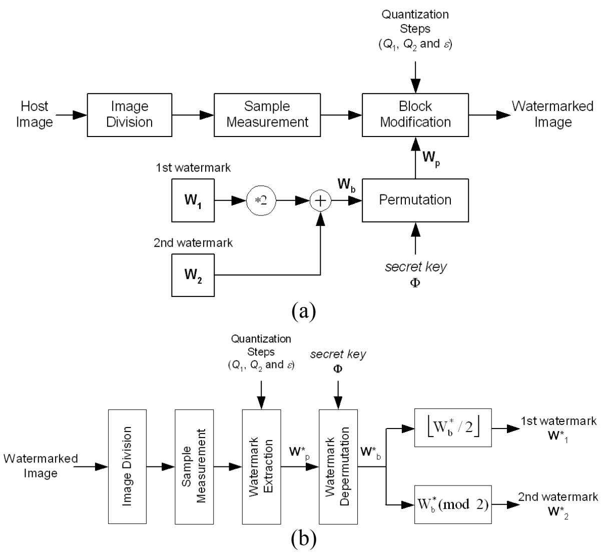

Fig.1. The block diagrams of (a) bi-watermark embedding procedure, and (b) bi-watermarking extraction procedure.

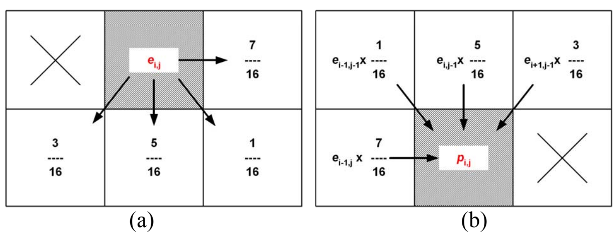

Fig.2. (a) The Floyd-Steinberg kernel of error diffusion, and (b) the relationship between pixel value and four neighbor errors

$$r_1 = r_2 - \tfrac{1}{2}Q_2 - \varepsilon, \quad r_2 = \tfrac{1}{2}Q_1 + 2Q_1 \cdot \lfloor q_{i,j}/2Q_1 \rfloor, \quad r_3 = r_2 + \tfrac{1}{2}Q_2 + \varepsilon, \tag{3}$$
$$r_4 = r_5 - \tfrac{1}{2}Q_2 - \varepsilon, \quad r_5 = \tfrac{3}{2}Q_1 + 2Q_1 \cdot \lfloor q_{i,j}/2Q_1 \rfloor, \quad r_6 = r_5 + \tfrac{1}{2}Q_2 + \varepsilon.$$

where the stepsize ε is a scalar, and $0<\varepsilon<(Q_1-Q_2)/4$.

The bi-watermark ($\mathbf{W}_b$) consists of two $M\times N$ binary watermarks ($\mathbf{W}_1$ and $\mathbf{W}_2$), which is given by $\mathbf{W}_b=2\mathbf{W}_1+\mathbf{W}_2$, and the bi-watermark is permuted to $\mathbf{W}_p$ with the secret key **Φ**. The goal of watermark permutation is to provide the perceptual invisibility. Fig.1(a) shows the block diagram of the bi-watermark embedding procedure.

2.2. Extraction Procedure

During the bi-watermark extraction procedure, the gray-scale image is also divided into several $M\times N$ blocks, and the sample $q_{i,j}$ of every block is measured by (1). Subsequently, the watermark symbol $\mathbf{W}^*_p(i,j)$ is extracted according to,

$$\mathbf{W}^*_p(i,j) = \begin{cases} \mathrm{U}_0 = \underline{01}_b = 1 & \text{, if } r_5 - \tfrac{1}{2}Q_2 \le q_{i,j} < r_5 + \tfrac{1}{2}Q_2, \\ \mathrm{U}_1 = \underline{10}_b = 2 & \text{, if } r_2 - \tfrac{1}{2}Q_2 \le q_{i,j} < r_2 + \tfrac{1}{2}Q_2, \\ \mathrm{S}_0 = \underline{00}_b = 0 & \text{, if } q_{i,j} < r_2 - \tfrac{1}{2}Q_2 \text{ or } r_2 + \tfrac{1}{2}Q_2 \le q_{i,j} < r_2 + \tfrac{1}{2}Q_1, \\ \mathrm{S}_1 = \underline{11}_b = 3 & \text{, elsewhere.} \end{cases} \tag{4}$$

$\mathbf{W}^*_p$ is de-permuted to bi-watermark $\mathbf{W}^*_b$ with the secret key **Φ**, and then $\mathbf{W}^*_b$ is separated into two binary watermarks ($\mathbf{W}^*_1$ and $\mathbf{W}^*_2$), which are calculated by $\mathbf{W}^*_1=\lfloor \mathbf{W}^*_b/2 \rfloor$ and $\mathbf{W}^*_2=\mathbf{W}^*_b(\text{mod } 2)$. Fig.1(b) illustrates the block diagram of bi-watermark extraction procedure.

In the proposed method, the bi-watermark is a binary halftone of downscaled host image. For instance, an $A\times B$ gray-scale image $\mathbf{X}$ is downscaled to $a\times b$ image $\mathbf{X}_{ds}$, and $a=\lfloor A\times 2^{-2.5} \rfloor \times 2^2$ and $b=\lfloor B\times 2^{-2.5} \rfloor \times 2^2$. The error diffusion scheme transforms $\mathbf{X}_{ds}$ to an $a\times b$ halftone ($\mathbf{Y}_{ds}$), and it is partitioned to two watermarks ($\mathbf{W}_1$ and $\mathbf{W}_2$) with size $\lfloor A/2 \rfloor \times \lfloor B/2 \rfloor$. Finally, the bi-watermark is obtained by $\mathbf{W}_b=2\mathbf{W}_1+\mathbf{W}_2$.

3. INVERSE HALFTONE TECHNIQUE

The error diffusion is a well-known halftoning scheme, and it transforms the gray-scale image to the binary image namely halftone. The Floyd-Steinberg kernel is utilized in error diffusion as shown in Fig.2(a). Let $e_{i,j}$ represent the error between the resultant pixel value $p'_{i,j}$ and $255\times q_{i,j}$ at (i,j)-th pixel, and $q_{i,j}$ be binary value (0 or 1). The error diffuses to neighbor four pixels with various weighted values. In other words, every pixel $p_{i,j}$ is added four errors derived from the neighbor pixels, the relationship (showing in Fig.2(b)) is given by,

$$p'_{i,j} = p_{i,j} + \tfrac{1}{16}e_{i-1,j-1} + \tfrac{5}{16}e_{i,j-1} + \tfrac{3}{16}e_{i+1,j-1} + \tfrac{7}{16}e_{i-1,j}. \tag{5}$$
$$e_{i,j} = p'_{i,j} - 255\times q_{i,j}$$
$$q_{i,j} = \begin{cases} 0 & \text{, if } p'_{i,j} < 128 \\ 1 & \text{, elsewhere} \end{cases}$$

For the conventional inverse halftoning, the Gaussian lowpass filter is utilized to restore the halftone image to gray-scale image. The advantage of lowpass filtering is the lower computational time, however, the quality of resultant gray-scale image is adequate. For this reason, the resultant image is treated as reference image, and it is refined using quadratic programming. The general form of quadratic programming is defined as,

$$\begin{aligned} \text{minimize} \quad & \mathbf{f} = \tfrac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{h}\mathbf{x} + \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} < \mathbf{b}, \\ & \mathbf{A}_{\mathbf{eq}}\mathbf{x} = \mathbf{b}_{\mathbf{eq}}, \\ & \mathbf{B}^{\mathrm{L}} \le \mathbf{x} \le \mathbf{B}^{\mathrm{U}}, \end{aligned} \tag{6}$$

where **f** is the object function, **h** and **c** are the cost matrix and cost vector, respectively. The variable **x** is the unknown variable vector, and two variables **A** and $\mathbf{A}_{eq}$ are matrices for equality and inequality, respectively. $\mathbf{B}^{L}$ and $\mathbf{B}^{U}$ denote the lower and upper bounds, respectively. Equation (5) is rewritten to the general form of quadratic programming,

$$\begin{aligned} & \text{minimum} \quad \sum_i \sum_j d_{i,j}^2 \\ & p_{i,j} + \tfrac{1}{16}e_{i-1,j-1} + \tfrac{5}{16}e_{i,j-1} + \tfrac{3}{16}e_{i+1,j-1} + \tfrac{7}{16}e_{i-1,j} - p'_{i,j} = 0, \\ & p'_{i,j} - e_{i,j} - 255q_{i,j} = 0, \\ & p_{i,j} + d_{i,j} = r_{i,j}, \\ & \mathbf{B}^{\mathrm{L}} \le p_{i,j} \le \mathbf{B}^{\mathrm{U}}, \\ & -\infty \le p'_{i,j}, d_{i,j}, e_{i,j} \le \infty. \end{aligned} \tag{7}$$

where $r_{i,j}$ is the (i,j)-th pixel value of reference image, and $d_{i,j}$ denotes the difference between $p_{i,j}$ and $r_{i,j}$. The unknown variable vector is $\mathbf{x}=[p_{i,j}\ p'_{i,j}\ e_{i-1,j-1}\ e_{i,j-1}\ e_{i+1,j-1}\ e_{i-1,j}\ e_{i,j}\ q_{i,j}\ d_{i,j}]$, and $p_{i,j}$ is estimated by solving the quadratic functions.

However, the problem of quadratic programming is the heavy computational complexity with the numerous unknown variables. For example, there are 589824 unknown variables to be solved by the quadratic programming for a 256 × 256 image (9 variables/pixel×256×256 pixels). Therefore, the original gray-scale image is divided into several non-overlapped blocks of size 4×4, every block is transformed to halftone block, and then the halftone

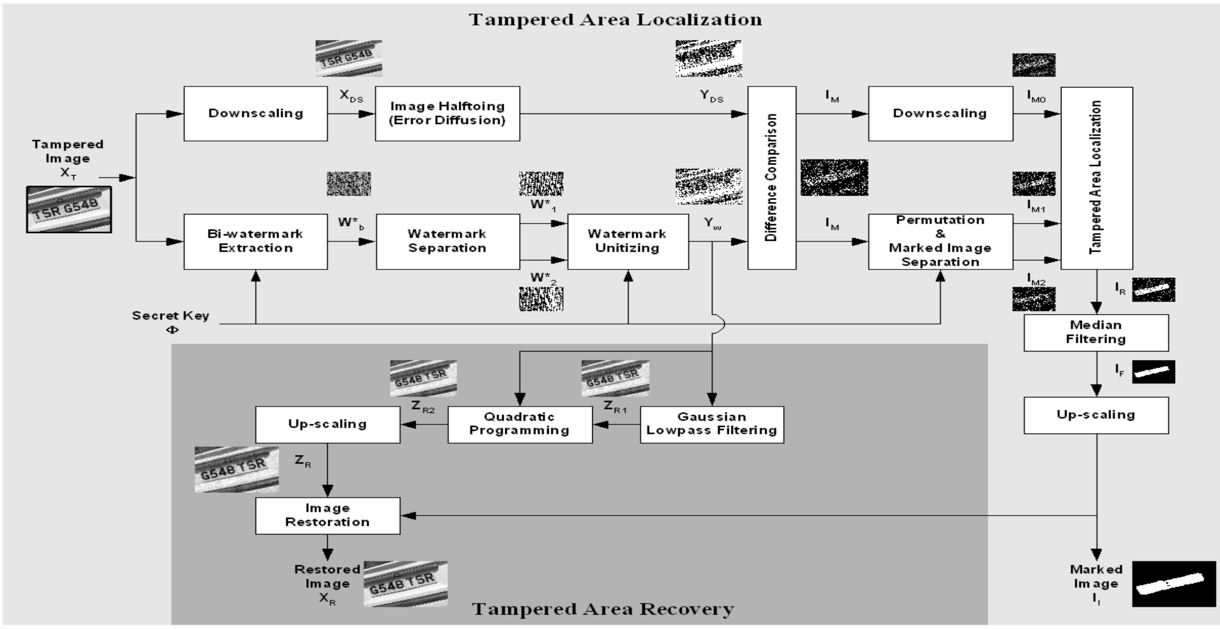

Fig.3. The block diagram of tampered area localization and recovery

image is obtained. Hence, there are only 144 unknown variables to be solved in every halftone block for quadratic programming of proposed inverse halftoning (9 variables/pixel × 4×4 pixels).

4. TAMPERED AREA LOCALIZATION AND RECOVERY

4.1. Tampered Area Localization

The bi-watermark is also a halftone of original host image. In our opinion, the proposed method will localize the tampered areas by the extracted bi-watermark without original image and watermarks. The steps of tampered area localization are described below:

1) The $\lfloor A/2\rfloor\times\lfloor B/2\rfloor$ watermark $\mathbf{W}^*_p$ is extracted from the $A\times B$ tampered image $\mathbf{X}_T$, and then $\mathbf{W}^*_p$ is de-permuted to bi-watermark $\mathbf{W}^*_b$ with secret key $\mathbf{\Phi}$.
2) $\mathbf{W}^*_b$ is separated into two watermarks ($\mathbf{W}^*_1$ and $\mathbf{W}^*_2$) of size $\lfloor A/2\rfloor\times\lfloor B/2\rfloor$, and those two watermarks are calculated by $\mathbf{W}^*_1=\lfloor \mathbf{W}^*_b/2\rfloor$ and $\mathbf{W}^*_2=\mathbf{W}^*_b(\text{mod } 2)$.
3) Two separated watermarks are reshaped to the vectors ($\mathbf{V}_1$ and $\mathbf{V}_2$) of size $1\times(\lfloor A/2\rfloor\times\lfloor B/2\rfloor)$, and those vectors are united as $\mathbf{V}_{12}$ of size $1\times(2\times\lfloor A/2\rfloor\times\lfloor B/2\rfloor)$. Then, the united vector $\mathbf{V}_{12}$ is padded with several 0-bites as $\mathbf{V}'_{12}$ of size $1\times(a\times b)$, and $\mathbf{V}'_{12}$ is reshaped to the array $\mathbf{Y}_w$ of size $a\times b$, where $a=\lfloor A\times 2^{-2.5}\rfloor\times 2^2$ and $b=\lfloor B\times 2^{-2.5}\rfloor\times 2^2$. $\mathbf{Y}_w$ is termed as the corrupted halftone.
4) Simultaneously, the tampered image $\mathbf{X}_T$ is downscaled as an $a\times b$ image $\mathbf{X}_{DS}$, and $\mathbf{X}_{DS}$ is transformed to the halftone image $\mathbf{Y}_{DS}$ using error diffusion of halftoning scheme.
5) The differences between $\mathbf{Y}_w$ and $\mathbf{Y}_{DS}$ are marked as $\mathbf{I}_M$.
6) $\mathbf{I}_M$ is separated to two sub-images of size $\lfloor A/2\rfloor\times\lfloor B/2\rfloor$, and those sub-images are permuted with $\mathbf{\Phi}$ to be the marked sub-images $\mathbf{I}_{M1}$ and $\mathbf{I}_{M2}$.
7) Downscale the marked image $\mathbf{I}_M$ to the marked sub-image $\mathbf{I}_{M0}$ of size $\lfloor A/2\rfloor\times\lfloor B/2\rfloor$, the tamped area is roughly localized by,

$$\mathbf{I}_R(u,v)=\begin{cases}1 & \text{, if } \mathbf{I}_{M0}(u,v)+\mathbf{I}_{M1}(u,v)+\mathbf{I}_{M2}(u,v)\ge\tau \\ 0 & \text{, otherwise}\end{cases}, \quad (8)$$

where $\mathbf{I}_R$ represents the rough marked sub-image, and τ is the threshold.

8) $\mathbf{I}_R$ is processed with median filter to obtain the fine marked sub-image $\mathbf{I}_F$, and $\mathbf{I}_F$ is up-scaled to the marked image $\mathbf{I}_I$ of size $A\times B$ ultimately.

4.2. Tampered Area Recovery

While the corrupted halftone $\mathbf{Y}_w$ is derived from the extracted bi-watermark, the Gaussian lowpass filter is performed on $\mathbf{Y}_w$ to result the 1st reference image $\mathbf{Z}_{R1}$ of size $a\times b$. The quadratic programming refines $\mathbf{Z}_{R1}$, and then we obtain the 2nd reference image $\mathbf{Z}_{R2}$. To up-scale $\mathbf{Z}_{R2}$ to the final reference image $\mathbf{Z}_R$ of size $A\times B$, the recovered image $\mathbf{X}_R$ is estimated by,

$$\mathbf{X}_R(u,v)=\begin{cases}\mathbf{Z}_R(u,v) & \text{, if } \mathbf{I}_I(u,v)=1 \\ \mathbf{X}_T(u,v) & \text{, otherwise}\end{cases}, \quad (9)$$

where $\mathbf{X}_T(u,v)$ and $\mathbf{Z}_R(u,v)$ denote the (u,v)-th pixels of the tampered image $\mathbf{X}_T$ and reference image $\mathbf{Z}_R$, respectively. Fig.3 illustrates the block diagram of tampered area localization and recovery.

5. EXPERIMENTAL RESULTS

In the experiments, the host image is first separated into several blocks of size 4×4, and these blocks are transformed to halftone blocks to obtain the halftone image as bi-watermark. Subsequently, the host image is divided into several blocks of size 2×2 with bi-watermark symbols embedding. The parameters of bi-watermarking algorithm are set Q_1=24, Q_2=1 and ε=1.5.

Fig.4(a) is a watermarked image of size 500×360 with PSNR=37.44dB [13], and its three color channels have been embedded with the self-halftone images of size 352×252 as shown in Fig.4(b). The pirate alters the numbers of license plate in the watermarked image, the tampered image is displayed in Fig.4(c) with PSNR=18.76dB. The proposed algorithm extracts the bi-watermark from each color channel, the bit error rate (BER) of bi-watermark is 0.0784 bit-per-pixel (bpp), and then the bi-watermark is exploited to localize the tampered areas as illustrated in Fig.4(d). Moreover, we also obtain the recovered image in Fig.4(e) with PSNR=28.46dB. If only Gaussian lowpass filer is used in the inverse halftoning scheme, and the recovered image quality is PSNR=27.80dB. Therefore, the quadratic programming can improve the image quality with 0.66dB.

Fig.5 shows the bi-watermarks extracted from the tampered images of Fig.4(a), which are imposed with four attacks respectively, such as, salt and pepper noise addition (SP, density=5%), Gaussian noise addition (GN, mean μ=0 and variance σ^2=7), lowpass filtering (LP, 3×3 window) and median filtering (MD, 3×3 window). To compare the tampered image with those bi-watermarks, although those attacks seriously damage the image content, the user can catch sight of the number in license plate is ‘G548-TSR’ not ‘TSR-G548’ from the extracted bi-watermarks. The BERs between original bi-watermark and those extracted bi-watermarks are 0.1557bpp, 0.2232bpp, 0.3199bpp and 0.2681bpp for SP, GN, LP and MD attacks, respectively.

6. CONCLUSION

The proposed scheme successfully localizes the tampered areas via the extracted bi-watermark, and the extracted bi-watermark is also a halftone exploited to recovery the corrupted image. Because the proposed bi-watermarking algorithm is a semi-fragile approach, it is sensitive to the image editing. For inverse halftoning approach, the quadratic programming indeed improves the quality of the recovered image.

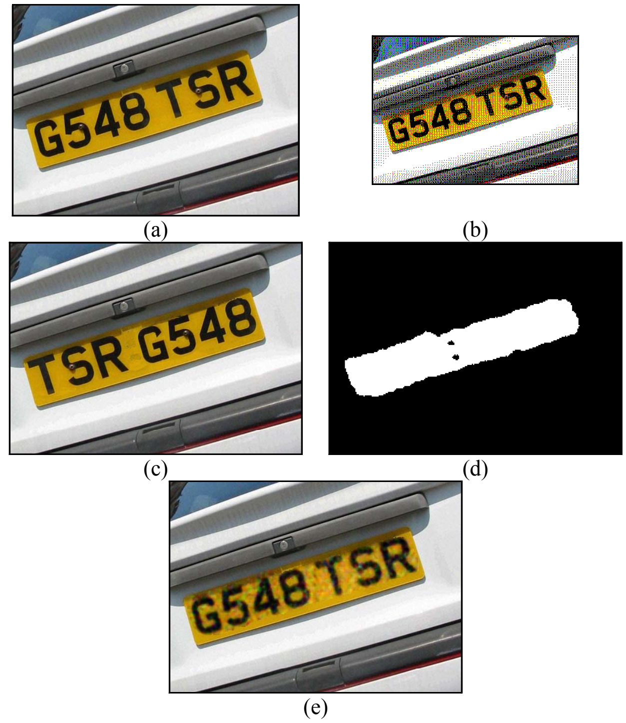

Fig.4. Tampered areas localization and restoration. (a) Watermarked image of size 500×360 (PSNR=37.44dB), (b) original bi-watermarks (self-halftone image) of size 352×252 for three color channel, (c) tampered image (PSNR=18.76 and BER=0.0784bpp), (d) the marked image $\mathbf{I}_I$, and (e) the recovered image (PSNR=28.46dB). We find the number in license plate is ‘G548-TSR’ not ‘TSR-G548’ via the extracted bi-watermarks.

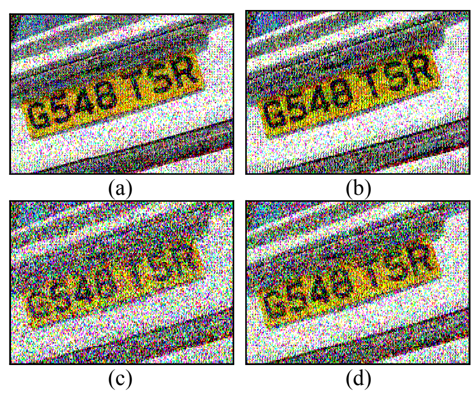

Fig.5. The bi-watermarks are extracted from the tampered images of Fig.4(a), which are imposed with four attacks respectively, they are (a) salt and pepper noise addition (BER= 0.1557bpp), (b) Gaussian noise addition (BER=0.2232bpp), (c) lowpass filtering (BER=0.3199bpp), and (d) median filtering (BER=0.2681).

REFERENCES

[1] D. Kundur and D. Hatzinakos, “Digital watermarking for telltale tamper proofing and authentication”, *Proc. of the IEEE*, vol.87, no.7, pp.1167-1180, July 1999.

[2] H. H. Ko and S. J. Park, “Semi-fragile watermarking for telltale tamper proofing and authenticating,” *ITC-CSCC*, pp.328-339, 2002.

[3] I. Kostopoulos, A. N. Skodras and D. Christodoulakis, “Self-authentication of color images,” *Proc. of The European Conf. on Electronic Imaging & Visual Arts*, March 2001.

[4] R. Caldelli, F. Filippini and M. Barni, “Joint near-lossless compression and watermarking of still images for authentication and tamper localization,” *Signal Processing: Image Communication*, vol.21, pp.890-903, 2006.

[5] S.-C. Pei and Y.-C. Zeng, “Tamper proofing and attack identification of corrupted image by using semi-fragile multiple-watermarking algorithm,” *ASIACCS’06*, pp.166-174, 21-24, March 2006.

[6] M. Mese and P. P. Vaidyanathan, “Recent Advances in Digital Halftoning and Inverse Halftoning Methods”, *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol.49, no.6, pp.790-805, 2002.

[7] P. W. Wong, “Inverse halftoning and kernel estimation for error diffusion,” *IEEE Trans. IP*, vol.4, pp.486-498, 1995.

[8] L.-M. Chen and H.-M. Hang, “An adaptive inverse halftoning algorithm,” *IEEE Trans. IP*, vol.6, no.8, pp.1202-1209, 1997.

[9] N. Damera-Venkata, T. D. Kite and B. L. Evans, “Fast blind inverse halftoning,” *ICIP’98*, vol.2, pp.64-69, 1998.

[10] Z. C. Lai and J. Y. Yen, “Inverse error diffusion using classified vector quantization,” *IEEE Trans. IP*, vol.7, pp.1753-1758, Dec. 1998.

[11] M. Mese and P. P. Vaidyanathan, “Look up table (LUT) method for inverse halftoning,” *IEEE Trans. IP*, vol.10, pp.1566-1578, Oct. 2001.

[12] —, “Tree-structured method for LUT inverse halftoning and for image halftoning,” *IEEE Trans. IP*, vol.11, pp.644-655, Sept. 2002.

[13] http://gallery.tonywhitmore.co.uk/car_plate_game_1_2005/img_2676?full=1