A Sequential Multiple Watermarks Embedding Technique

Peter H. W. Wong*, Andy Chang**, Oscar C. Au†
Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Email: peterwong@ieee.org*, eecax@ust.hk**, eeau@ust.hk†

ABSTRACT

In this paper, we propose a novel multiple watermarks embedding scheme. We assume *M* watermarks have already embedded in the image using *M* sets of secret key. With the availability of these *M* sets secret key, another *N* watermarks can be embedded using the proposed technique while the energies of the watermarks are minimized. And the watermarks embedded later will not interfere with the first *M* watermarks. Experimental results show watermarked images have good visual quality and the watermarks are robustness to JPEG compression and noise attacks.

I. INTRODUCTION

Many robust watermarking algorithms were proposed in the past few years. Most of these algorithms focus on increasing the robustness of the watermark against different kinds of attacks such as cropping, rotation, scaling, etc. Usually these methods can be used to embed one watermark only. While most schemes embed only a single watermark, some allow multiple watermarks embedding [1-5]. Cox et al. [1] assumes the multiple watermarks are close to orthogonal and simply extend the single watermark algorithms to embed them together. Some [3-4] embed orthogonal watermarks and extend the single watermark algorithms for multiple watermarks. One difficulty in multiple watermarking is that watermarks may interfere with each other (crosstalk). Although most researchers use noise-like watermarks with small correlation, the small correlation will affect the detection score and cause bit error. In [2], we proposed a method called MWE [2] to embed multiple watermarks simultaneously without crosstalk while minimizing the distortion due to watermarking. However, in some applications such as Digital Rights Management (DRM) systems, it is desirable to embed the multiple watermarks sequentially. A straight forward solution is to embed the additional watermarks using the original image and information of previously embedded watermarks. But it is costly and not feasible in some situations. In this paper, we propose a technique call Sequential Multiple Watermarks Embedding (SMWE) to embed watermarks sequentially, assuming that the keys of previously embedded watermarks are available. The SMWE also minimizes the image distortion due to watermarking.

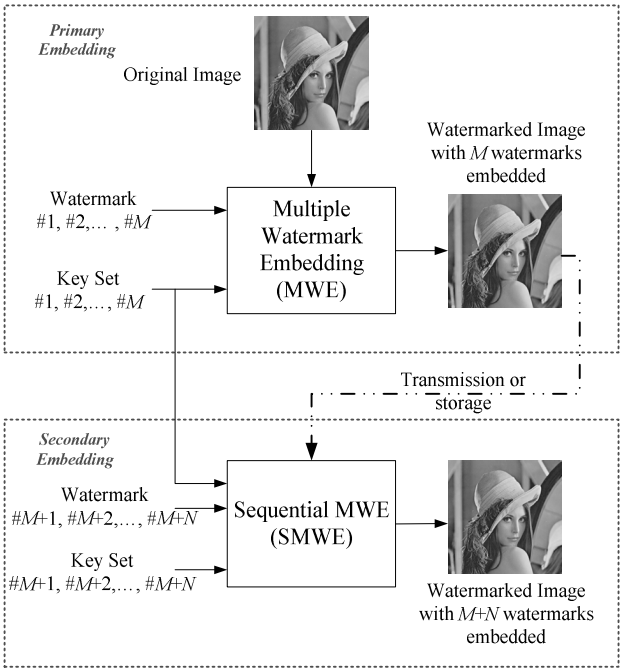

Fig. 1. The proposed Sequential Multiple Watermarks Embedding (SMWE).

II. THE PROPOSED SMWE

A. Primary Embedding

In primary embedding, *M* watermarks (where *M* can be 1) are embedded using the previously proposed MWE scheme [2]. The MWE embeds multiple watermarks simultaneously in the same watermark space while minimizing the watermark (distortion) energy. As shown in Fig. 1, one key set is used to embed each watermark in MWE. These key sets are randomly chosen and need not be orthogonal to each other. The watermark host vector is extracted from the image from some domains and split into sub-vectors. The sub-vector will be used to embed one or more bits and the number if bits embedded in each sub-vector is equal to the number of watermarks. One advantage of the MWE is that it avoids crosstalk among different watermarks. To decode or detect a particular watermark, only the corresponding key set is needed at the decoder and each embedded watermark bit sequence can be decoded independently.

B. Secondary Embedding

After M watermarks are embedded using MWE, the watermarked image may be stored or transmit to somewhere else. Suppose N (where N can be 1 also) more watermarks need to be embedded in this image. Without the knowledge of the previously embedded watermarks, it is impossible to embed more watermarks without interfering the previously embedded watermarks. Even if the previously embedded watermarks are known, the complexity of simultaneously embedding $M+N$ watermarks is large. Here we propose a technique called Sequential Multiple Watermarks Embedding (SMWE) to handle this problem.

Same as MWE, the watermark host vector in SMWE is extracted from the image from the same domain and splitted into sub-vectors as in MWE. Let the sub-vector be $\boldsymbol{Y} = [y_1, y_2, ..., y_L]$ with length L with $N<<L$. This sub-vector already contains M previously embedded bits, and will be used to embed N additional bits from the N additional watermarks. Denote these N bits as $w_{M+1}, w_{M+2}, ..., w_{M+N}$ with $w_i \in \{0,1\}$. As SMWE uses the same method to embed N watermark bits in every sub-vector, we only describe SMWE for one sub-vector.

SMWE assumes that the M sets of secret keys used to embed the first M watermarks are available. Denote the keys used in Primary Embedding as $\boldsymbol{K_1}$, $\boldsymbol{K_2}$,…, $\boldsymbol{K_M}$. One key set is used to embed one watermark. A key set consists of two keys. The first key, denoted as d, is a pseudo-random positive real number with mean and variance denoted as $\bar{d}$ and σ_d^2 respectively. The second key is $\boldsymbol{K} = [k_1, k_2, ..., k_L]$ with each k_i being zero-mean Gaussian with variance σ_k^2. Both keys are needed to decode or detect the modulated watermark.

As N bits will be embedded in the sub-vector, therefore N sets of keys are needed in the secondary embedding to embed the N additional watermarks. So N sets of secret keys $\boldsymbol{K_{M+1}}$, $\boldsymbol{K_{M+2}}$,…, $\boldsymbol{K_{M+N}}$. are randomly generated. As the embedded watermarks should not interfere with the previously embedded watermarks, the N keys need to be orthogonal to all the keys used in Primary Embedding. We modify the $\boldsymbol{K_{M+i}}$ to $\boldsymbol{K_i'}$ using the Gram-Schmidt procedure [6] such that $\boldsymbol{K_i'} \perp \boldsymbol{K_j}$ for i=1,2, …,N and j=1,2,…,M. An orthogonal set of vector $\boldsymbol{Z_1'}, \boldsymbol{Z_2'}, ..., \boldsymbol{Z_M'}$ is computed first according to (1):

$$\boldsymbol{Z_i} = \boldsymbol{K_i} - \sum_{j=1}^{i-1} \frac{\langle \boldsymbol{K_i}, \boldsymbol{Z_j} \rangle}{\|\boldsymbol{Z_j}\|_2^2} \cdot \boldsymbol{Z_j} \quad i = 1,2,...,M \qquad (1)$$

where, $\|\cdot\|_2$ is and L^2-norm and $\langle \boldsymbol{K_i}, \boldsymbol{Z_j} \rangle$ is the inner product of $\boldsymbol{K_i}$ and $\boldsymbol{Z_i}$. It is easy to verify that $\boldsymbol{Z_i} \perp \boldsymbol{K_j}$ for $i \neq j$. The modified key $\boldsymbol{K_i'}$ can be obtained using (2):

$$\boldsymbol{K_i'} = \boldsymbol{K_{M+i}} - \sum_{j=1}^{M} \frac{\langle \boldsymbol{K_{M+i}}, \boldsymbol{Z_j} \rangle}{\|\boldsymbol{Z_j}\|_2^2} \cdot \boldsymbol{Z_j} \quad i = 1,2,...,N \qquad (2)$$

It should be noted that it is not required that $\boldsymbol{K_i'}$ is orthogonal to $\boldsymbol{K_j'}$ for $i \neq j$ and $1 \le i, j \le N$. The watermarked sub-vector, denoted as $\boldsymbol{Y'}$ is obtained by adding the scaled $\boldsymbol{K_i}$ according to the watermark bits to be embedded.

$$\boldsymbol{Y'} = \boldsymbol{Y} + \alpha_1 \boldsymbol{K_1'} + \alpha_2 \boldsymbol{K_2'} + ... + \alpha_N \boldsymbol{K_N'} \qquad (3)$$

The scaling factors form a row vector $\boldsymbol{A} = [\alpha_1, \alpha_2, ..., \alpha_N]$ with $\alpha_j \in \Re$.

The goal of the watermark embedding process is to derive a set of scaling factors (or vector $\boldsymbol{A}$) which satisfies two conditions. The first condition is that the projection of $\boldsymbol{Y'}$ onto the direction of $\boldsymbol{K_i}$ corresponds to the correct watermark bit as

$$\left[Round\left(\frac{\langle \boldsymbol{Y'}, \boldsymbol{K_i} \rangle}{d_i} \right) \right] \%2 = w_i, \quad M+1 \le i \le M+N \qquad (4)$$

where $Round(\,)$ and % are rounding and modulo 2 respectively. This is similar to quantization-based embedding [7]. Recall that d_i is the first secret key which is a pseudo-random positive real number with the mean $\bar{d}$ and $\bar{d}$ can be used to controls the robustness an the energy of the embedded watermark bit.

The second condition is that the distortion or the Euclidean distance E_w between $\boldsymbol{Y}$ and $\boldsymbol{Y'}$ is minimized. The Euclidean distance E_w is also the energy of the watermark bits and is equal to

$$E_w = \|\boldsymbol{Y'} - \boldsymbol{Y}\|_2^2 = \boldsymbol{ACA^T} \qquad (5)$$

where

$$\boldsymbol{C} = \begin{pmatrix} \langle \boldsymbol{K_1'}, \boldsymbol{K_{M+1}} \rangle & \langle \boldsymbol{K_1'}, \boldsymbol{K_{M+2}} \rangle & \cdots & \langle \boldsymbol{K_1'}, \boldsymbol{K_{M+N}} \rangle \\ \langle \boldsymbol{K_2'}, \boldsymbol{K_{M+1}} \rangle & \langle \boldsymbol{K_2'}, \boldsymbol{K_{M+2}} \rangle & \cdots & \langle \boldsymbol{K_2'}, \boldsymbol{K_{M+N}} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \boldsymbol{K_N'}, K_{M+1} \rangle & \langle \boldsymbol{K_N'}, \boldsymbol{K_{M+2}} \rangle & \cdots & \langle \boldsymbol{K_N'}, \boldsymbol{K_{M+N}} \rangle \end{pmatrix} \qquad (6)$$

Substituting (3) into (4), N simultaneous equations can be obtained. They are, in matrix form,

$$\boldsymbol{AC} = \boldsymbol{B} = [b_1, b_2, ..., b_N] \tag{7}$$

with

$$b_i = d_i \cdot (2 \cdot r_i + w_i) - \langle \boldsymbol{Y}, \boldsymbol{K_{M+i}} \rangle, \quad 1 \le i \le N \tag{8}$$

where r_i is an integer for any i. If all the r_i are determined, the row vector $\boldsymbol{B}$ can be computed using (8) and the scaling vector $\boldsymbol{A}$ can then be obtained as $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{C}^{-1}$ from (7). It is important to choose the integers r_i such that the E_w is as small as possible. By substituting (7) into (5), the E_w can be rewritten in terms of $\boldsymbol{C}$ and $\boldsymbol{B}$ as

$$E_w = \boldsymbol{B}\boldsymbol{C}^{-1}\boldsymbol{B}^{\boldsymbol{T}} \tag{9}$$

We use the IA-R approach in [2] to determine the r_i and we will not to mention the IA-R due to the limitation of space.

III. WATERMARK DECODING AND DETECTION

It should be noted that after applying the MWE and SMWE, total M+N watermarks are embedded in the image. To decode a particular watermark out of the M+N embedded watermarks, only the corresponding key set are needed. The feature vector is extracted from the testing image and segmented to sub-vectors similar to the watermark embedding process. As the decoding process is the same for all sub-vectors, for a sub-vector $\boldsymbol{X}$, the decoded bit for the i^{th} watermark are computed according to (10):

$$\hat{w}_i = \left[Round\left(\frac{\langle \boldsymbol{X}, \boldsymbol{K_i} \rangle}{d_i} \right) \right] \%2 \tag{10}$$

To detect the existence of a particular watermark in a testing image, more information is needed: both the watermark and corresponding key set. After all the bits of watermark is decoded, a normalized cross-correlation score S between the original watermark and the decoded watermark is computed. If the detection score is higher than a threshold, the watermark is said to be detected in the testing image.

IV .EXPERIMANTAL RESULTS AND DISCUSSION

The 512x512 image 'Lena' is used in our experiments. Only the luminance component is used. The whole original image is transformed to the (512x512) DCT domain and scanned in a zigzag order. We use the first 10% of the DCT AC coefficients to form the host vector to embed the watermark. Each watermark is a random bit sequence with 1320 bits. The key d_i is generated from a Gaussian distribution with $\overline{d} = 250$ and $\sigma_d^{\,2} = 4$. The value of $\overline{d}$ controls the watermarks strength and the robustness of the watermarks. The other keys $\boldsymbol{K_1}$, $\boldsymbol{K_2}$, …, $\boldsymbol{K_{M+N}}$ are generated from a Gaussian distribution with zero mean and $\sigma_k^{\,2} = 16$. These values are chosen in an ad-hoc way to achieve a PSNR of about 45dB for watermarked images when M+N=5. The experiments are simulated in 10 trial and the average results are report.

In the first experiment, multiple watermarks are embedded in 3 ways. For the first way, multiple watermarks are embedded simultaneously using MWE. For the second way, the watermarks are embedded one-by-one using SMWE, we denote this way as SMWE1. For the third way, about half of the watermarks are embedded using MWE first, then the rest watermarks are embedded using SMWE, we denote this way as SMWEn. The average PSNR of the watermarked images are shown in Fig. 2. The PSNRs decrease as the number of watermarks increases. The MWE has the highest PSNR because the watermarks are embedded simultaneously and the best possible joint optimization is done. The SMWE1 has the lowest PSNR because the watermarks are embedded one-by-one and no joint optimization can be applied. However, the SMWE1 has the lowest complexity among these three ways.

In the second experiment, we test the robustness of the watermarks against JPEG compression attacks. We choose to embed 5 watermarks (totally 6600 bits embedded) into the images in three ways mentioned above. The mean PSNR of the watermarked images are 44.92dB, 44.73dB and 44.77dB for MWE, SMWE1 and SMWEn respectively. The watermarked images are then compressed using JPEG with different Scaling Factors (SF) and the SF is related to the Quality Factor (QF) as:

$$\mathrm{SF} = \begin{cases} \dfrac{50}{\mathrm{QF}}, & \mathrm{QF} \le 50 \\ 2 - \dfrac{\mathrm{QF}}{50}, & 50 < \mathrm{QF} < 100 \end{cases} \tag{11}$$

The average detection scores (average of 5 watermarks, 10 trials) against JPEG attacks are shown in Fig. 3. The detection scores decrease with the increase in SF. The detection scores of MWE, SMWE1 and SMWEn are almost the same. This suggests that MWE is the best among the three since MWE gives the highest PSNR. Experiment results (not shown in this paper) show that a detection threshold of 0.2 is enough to have zero false positive detection. Refer to Fig. 3, we can conclude that the watermarks are robust against JPEG attacks up to

SF=2.5. It should be noted that in our experiments, when the detection score is 0.2, 3960 bits out of 6600 bits are decoded correctly.

In the third experiment, the watermarks are tested under noise attacks. Gaussian noise with different variances are added to watermarked image before watermark decoding. The results are shown in Fig. 4. Again, the detection scores for MWE, SMWE1 and SMWEn are almost the same. However, MWE is not feasible for some situation as the watermarks should be embedded simultaneously.

V. CONCLUSION

In this paper, we proposed a sequential multiple watermark embedding technique. The watermarks can be embedded sequentially provided that the keys of the previously embedded watermarks are available. Experimental results show that the proposed approach can give good quality watermarked images and the watermarks are robust against JPEG and noise attack.

REFERENCE

[1] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1673-1687, Dec. 1997.

[2] P.H.W. Wong, Oscar C. Au and Gene Y.M. Yeung, "A Novel Blind Multiple Watermarking Technique for Images," *IEEE Trans on Circuits and Syst. for Video Technol.*, vol. 13, no. 8, pp. 813-830, Aug. 2003.

[3] C. T. Hsu and J. L. Wu, "Hidden Digital Watermarks in Images," *IEEE Trans. Image Processing*, vol. 8, no. 1, pp. 55-68, Jan. 1999.

[4] S. Stankovic, I. Djurovic and I. Pitas, "Watermarking in the Space/Spatial-Frequency Domain Using Two-Dimensional Radon-Wigner Distribution," *IEEE Trans. Image Processing*, vol. 10, no. 4, pp. 650-658, April 2001.

[5] W. N. Lie, G. S. Lin, C. L. Wu and T. C. Wang, "Robust Image Watermarking on the DCT Domain," in *Proc. of IEEE Int. Sym. on Circuits and Systems*, vol. 1, pp. 228-231, May 2000.

[6] D.G. Luenberger, *Optimization by Vector Space Method*, John Wiley & Sons, Inc., 1969.

[7] M. Wu, "Joint Security and Robustness Enhancement for Quantization Based Embedding," *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 13, no. 8, pp. 831-841, Aug. 2003.

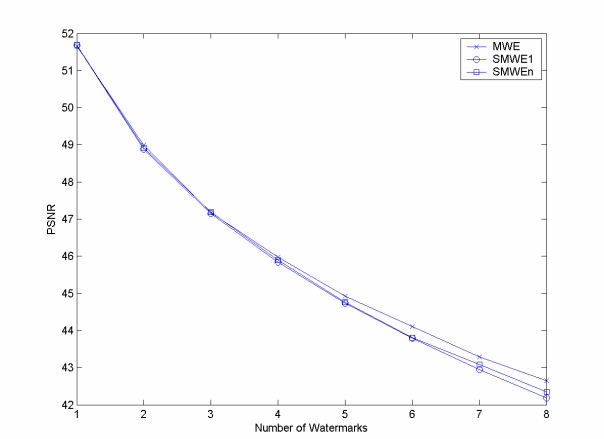

Fig. 2. Average PSNR of watermarked images Vs number of watermarks embedded

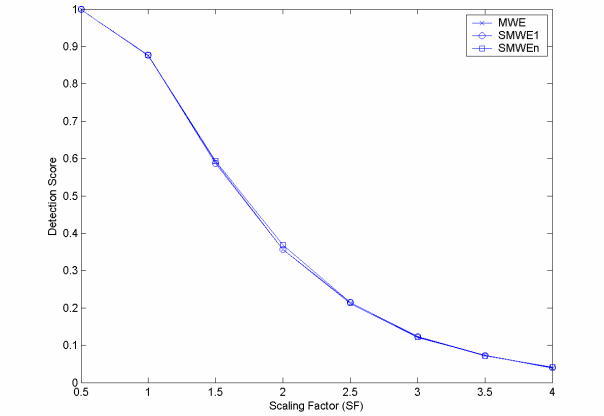

Fig. 3. Average detection scores against JPEG attacks

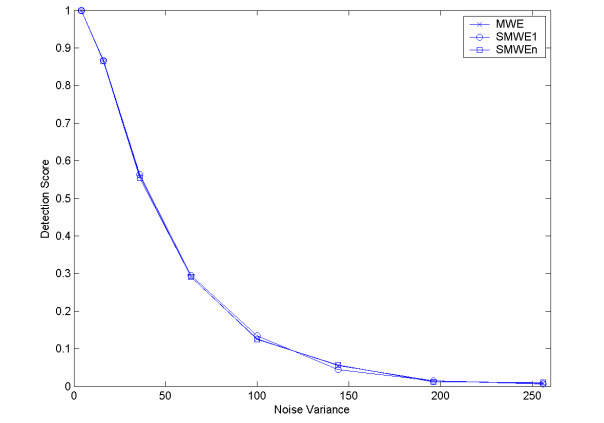

Fig. 4. Average detection scores against noise attacks