Journal of Information Assurance and Security 1 (2006) 33-42

Multipurpose Image Watermarking Method Based on Mean-removed Vector Quantization

Zhe-Ming Lu[1*], Wei-Min Zheng[2], Jeng-Shyang Pan[3] and Zhen Sun[4]

[1] Harbin Institute of Technology Shenzhen Graduate School, Visual Information Analysis and Processing Research Center, Room 417,Building No.4,HIT Campus Shenzhen University Town, Xili,Shenzhen 518055 P.R.China. ([*] *corresponding author*)
zhemingl@yahoo.com

[2] Chinese Academy of Sciences, Institute of Computing Technology, Beijing 100080, P. R. China
zhengwm@ict.ac.cn

[3] National Kaohsiung University of Applied Sciences, Department of Electronic Engineering, Kaohsiung 807, Taiwan
jspan@cc.kuas.edu.tw

[4] Harbin Institute of Technology, Department of Automatic Test and Control, P. O. Box 339, Harbin 150001, P. R. China

***Abstract*:** Digital watermarking technique has been presented and widely researched to solve some important issues in the digital world, such as copyright protection, copy protection, and content authentication. Conventional watermarking algorithms are mostly based on discrete cosine transform (DCT), discrete Fourier transform (DFT), and discrete wavelet transform (DWT). Most of these algorithms are designed for only one purpose. In recent years, some multipurpose digital watermarking methods based on DWT and DFT have been presented to achieve the goal of both content authentication and copyright protection. Lately, several robust watermarking schemes based on vector quantization (VQ) have been presented, but they can be used only for copyright protection. In this paper, we present a novel multipurpose digital image watermarking method based on a mean-removed vector quantizer (MRVQ) structure. In the proposed method, the fragile watermark and the robust watermark are embedded in mean indices and residual indices using different techniques, and both of them can be blindly extracted. Simulation results demonstrate the effectiveness of our algorithm in terms of robustness and fragility.

***Keywords*:** Copyright protection, fragile watermarking, image authentication, mean-removed vector quantization, multipurpose watermarking, robust watermarking**.**

1. Introduction

The explosive growth of digital multimedia techniques, together with the rapid development of digital network communications, has created a pressing demand for techniques that can be used for copy protection, copyright protection, and content authentication. Conventional cryptographic systems permit only valid keyholders access to encrypted data, but once such data are decrypted there is no way to track its reproduction or retransmission. Over the last decade, digital watermarking has been presented to complement cryptographic processes. Digital watermarking is a technique to insert a secret signal (i.e., a watermark) in digital data (namely audio, video or a digital image), which enables one to establish ownership or identify a buyer. In general, there are two types of digital watermarks addressed in the existing literature, visible and invisible watermarks. A visible watermark typically contains a visible message or a company logo indicating the ownership of the image. On the other hand, the invisibly watermarked digital content appears visually very similar to the original.

Most of existing invisible watermarking schemes are designed for either copyright protection or content authentication. Invisible watermarks can be broadly classified into two types, *robust* and *fragile* watermarks. Robust watermarks [1]-[6] are generally used for copyright protection and ownership verification because they are robust to nearly all kinds of image processing operations. In comparison, fragile watermarks [7]-[10] are mainly applied to content authentication and integrity attestation because they are completely fragile to any modifications. To fulfill multipurpose applications, several multipurpose watermarking algorithms based on wavelet transform [11] and fast Fourier transform [12] have been presented. In [11], watermarks are embedded once in the hiding process and can be blindly extracted for different applications in the detection process. In addition to images (gray-scale and color), this method has been extended to audio watermarking [12]. It should be addressed that, unlike multipurpose watermarking, multiple or cocktail watermarking methods [13], [14] are mainly applied to copyright protection by embedding multiple robust watermarks, each one being robust to certain kinds of attacks. Recently, some robust image watermarking techniques based on vector quantization (VQ) [15]-[21] have been presented. References [15]-[18] embed the watermark information into the encoded indices under the constraint that the extra distortion is less than a given threshold. Reference [19] embeds the watermark bit in the dimension information

Received on December 17, 2005.

of the variable dimension reconstruction blocks of the input image. References [20], [21] embed the watermark information by utilizing the properties, such as mean and variance, of neighboring indices. In this paper, we present a novel multipurpose watermarking method based on mean-removed vector quantization. In the proposed algorithm, the robust watermark is embedded in the quantized mean indices by using the embedding method presented in [20], and the fragile watermark is embedded in the residual codeword indices by using a novel index constrained method.

The remainder of this paper is organized as follows. In Section 2, previous VQ-based watermarking algorithms are reviewed. In Section 3, the proposed multipurpose watermarking method is described in detail. The simulation results and conclusions are given in Section 4 and Section 5, respectively.

2. Previous VQ-based Watermarking Algorithms

2.1 Vector quantization

VQ is an efficient block-based lossy image compression technique with a high compression ratio and a simple table lookup decoder. VQ can be defined as a mapping from k-dimensional Euclidean space R^k into a finite codebook $C=\{\boldsymbol{c}_i| i=0, 1,\ldots,N\text{-}1\}$, where $\boldsymbol{c}_i$ is called a codeword and N is the codebook size. Before online encoding, VQ first generates a representative codebook offline from a number of training vectors using the well-known GLA algorithm [22]. In image vector quantization, the image to be encoded is first segmented into vectors and then sequentially encoded vector by vector. In the encoding stage, for each k-dimensional input vector $\boldsymbol{x}=(x_1, x_2,\ldots, x_k)$, we find the nearest neighbor codeword $\boldsymbol{c}_i = (c_{i1},c_{i2},\ldots, c_{ik})$ in the codebook $C=\{\boldsymbol{c}_0,\boldsymbol{c}_1,\ldots\boldsymbol{c}_{N\text{-}1}\}$, which satisfies the following condition:

$$d(\boldsymbol{x},\boldsymbol{c}_i)=\min_{0\le j\le N\text{-}1} d(\boldsymbol{x},\boldsymbol{c}_j) \tag{1}$$

Where $d(\boldsymbol{x}, \boldsymbol{c}_j)$ is the distortion between the input vector $\boldsymbol{x}$ and the codeword $\boldsymbol{c}_j$, which can be defined as follows

$$d(\boldsymbol{x},\boldsymbol{c}_j)=\sum_{l=1}^{k}(x_l-c_{jl})^2 \tag{2}$$

And then the index i of the nearest neighbor codeword assigned to the input vector $\boldsymbol{x}$ is transmitted over the channel to the decoder. The decoder has the same codebook as the encoder. In the decoding phase, for each index i, the decoder merely performs a simple table look-up operation to obtain $\boldsymbol{c}_i$ and then uses $\boldsymbol{c}_i$ to reconstruct the input vector $\boldsymbol{x}$. Compression is achieved by transmitting or storing the index of a codeword rather than the codeword itself.

2.2 Watermarking algorithms based on codebook partition

The main idea of the VQ-based digital watermarking schemes presented in [15]-[18] is to carry secret copyright information by codeword indices. The aim of the codebook partition is to classify the neighboring codewords into the same cluster. Given a threshold $D>0$, we denote by $S=\{S_1,S_2,\ldots,S_M\}$ a *standard partition* of the codebook $C=\{\boldsymbol{c}_0,\boldsymbol{c}_1,\ldots,\boldsymbol{c}_{N\text{-}1}\}$ for the threshold D, if S satisfies the following four conditions:

1) $S=\bigcup_{i=1}^{M} S_i$;

2) $\forall\, i,j, 1\le i,j\le M$, if $i\ne j$, then $S_i\bigcap S_j=\Phi$;

3) $\forall\, i,\ 1\le i\le M$, if $\boldsymbol{c}_l\in S_i$ and $\boldsymbol{c}_j\in S_i$ $(0\le l, j\le N\text{-}1)$, then $d(\boldsymbol{c}_l, \boldsymbol{c}_j)\le D$;

4) $\|S_i\|=2^{n(i)}$. Where $\|S_i\|$ denotes the number of codewords in S_i and $n(i)$ is a natural number.

Before the embedding process, the original image is first divided into blocks. For each block, the index of the best match codeword is found. The watermarked codeword index is then obtained by modifying the original codeword index according to the corresponding watermark bits. The modification is under the constraint that the modified index and the original one is in the same partition such that the introduced extra distortion is less than the given distortion threshold. In the decoding phase, not the original but the watermarked codeword is used to represent the input image block. Therefore, the VQ-based digital image watermarking will introduce some extra distortion. Whether the original image is required or not during the watermark extraction is dependent on the embedding method. In these algorithms, the codebook is open for users but the partition is the secret key. Experimental results show that these algorithms are robust to VQ compression with high-performance codebooks, JPEG compression and some spatial image processing operations. However, these algorithms are fragile to rotation operations and VQ compression with low-performance codebooks.

2.3 Watermarking algorithms based on index properties

To enhance the robustness to rotation operations and VQ compression operations, some image watermarking algorithms [20], [21] based on the properties of neighboring indices have been proposed. In [20], the original watermark $\boldsymbol{W}$ with size $A_w\times B_w$ is first permuted by a predetermined key, key_1, to generate the permuted watermark $\boldsymbol{W_P}$ for embedding. The original image $\boldsymbol{X}$ with size $A\times B$ is then divided into vectors $\boldsymbol{x}(h,l)$ with size $(A/A_w)\times(B/B_w)$, where $\boldsymbol{x}(h,l)$ denotes the image block at the position of (h,l). After that, each vector $\boldsymbol{x}(h,l)$ finds its best codeword $\boldsymbol{c}_i$ in the codebook C and the index i is assigned to $\boldsymbol{x}(h,l)$, we can then obtain the indices matrix $\boldsymbol{Y}$ with elements $y(h,l)$, which can be represented by

$$\boldsymbol{Y}=\mathrm{VQ}(\boldsymbol{X})=\bigcup_{h=0}^{\frac{A}{A_w}-1}\bigcup_{l=0}^{\frac{B}{B_w}-1}\mathrm{VQ}(\boldsymbol{x}(h,l))=\bigcup_{h=0}^{\frac{A}{A_w}-1}\bigcup_{l=0}^{\frac{B}{B_w}-1} y(h,l) \tag{3}$$

For natural images, the VQ indices among neighboring blocks tend to be very similar, so we can make use of this property to generate the *polarities* $\boldsymbol{P}$. After calculating the

variances of $y(h,l)$ and the indices of its surrounding blocks with

$$\sigma^2(h,l)=\left(\frac{1}{9}\sum_{i=h-1}^{h+1}\sum_{j=l-1}^{l+1}y^2(i,j)\right)-\left(\frac{1}{9}\sum_{i=h-1}^{h+1}\sum_{j=l-1}^{l+1}y(i,j)\right)^2 \tag{4}$$

We can obtain the polarities $\boldsymbol{P}$ as follows

$$\boldsymbol{P}=\bigcup_{h=0}^{\frac{A}{A_w}-1}\bigcup_{l=0}^{\frac{B}{B_w}-1}p(h,l) \tag{5}$$

Where

$$p(h,l)=\begin{cases}1 & \text{if } \sigma^2(h,l)\ge T\\ 0 & \text{otherwise}\end{cases} \tag{6}$$

For convenience, we set the threshold T to be half of the codebook size, $N/2$. We are then able to generate the final embedded watermark or the secret key, key_2, with the exclusive-or operation as follows

$$key_2=\boldsymbol{W_P}\oplus\boldsymbol{P} \tag{7}$$

After the inverse-VQ operation, both the reconstructed image $\boldsymbol{X'}$ and the secret key, key_2, work together to protect the ownership of the original image.

In the extraction process, we first calculate the estimated polarities $\boldsymbol{P'}$ from $\boldsymbol{X'}$, and then obtain an estimate of the permuted watermark as follows

$$\boldsymbol{W'_P}=key_2\oplus\boldsymbol{P'} \tag{8}$$

Finally, we can perform the inverse permutation operation with key_1 to obtain the extracted watermark $\boldsymbol{W'}$.

In order to embed multiple watermarks, reference [21] also uses the mean of indices to generate another polarities $\boldsymbol{P}_1$ for embedding. Experimental results show that these algorithms are robust to many kinds of attacks, including JPEG, VQ, filtering, blurring and rotation. However, these algorithms have the following problems:

- We can also extract the watermark from the original image that has no watermark embedded in it at all.
- The codebook should be used as a key, because if the user possesses the same codebook, he can also embed his own watermark in the watermarked image without any modification.

3. Proposed Multipurpose Watermarking Algorithm

3.1 Mean-removed vector quantization

Usually we deal with vectors that have zero statistical mean in the sense that the expected value of each component is zero. Nevertheless, many vectors such as sampled image intensity rasters have only nonnegative components and hence have nonzero means. The local means over small blocks can vary quite widely over an image. Furthermore, this mean of an image vector can often be regarded as statistically independent of the variation of the vector, that is, of the way the components vary about this average. The term *mean* of a vector is used in this section specifically to refer to the *sample mean*, i.e., the average of the components of a vector. Thus the mean, m, of vector $\boldsymbol{x}$ is itself a scalar random variable given by

$$m=\frac{1}{k}\sum_{i=1}^{k}x_i=\frac{1}{k}\mathbf{1}^t\boldsymbol{x} \tag{9}$$

Where $\mathbf{1}=(1,1,\ldots,1)^t$, the k-dimensional vector with all components equal to unity. The mean-removed residual, $\boldsymbol{r}$, of the random variable $\boldsymbol{x}$ is defined as

$$\boldsymbol{r}=\boldsymbol{x}-\frac{1}{k}(\mathbf{1}^t\boldsymbol{x})\mathbf{1}=\boldsymbol{x}-m\mathbf{1} \tag{10}$$

Hence, $\boldsymbol{x}$ can be represented as the sum of a mean vector $m\mathbf{1}$ and the residual vector $\boldsymbol{r}$ according to:

$$\boldsymbol{x}=\boldsymbol{r}+m\mathbf{1} \tag{11}$$

The residual $\boldsymbol{r}$ is the "mean-removed" version of the vector $\boldsymbol{x}$ and has zero mean. Thus we have a natural decomposition of the original vector into separate features, a mean (representing a general background level) and a residual (representing the shape of the vector about its mean). Quantizing these features using separate codebooks is referred to as MRVQ for "mean-removed VQ" or "mean-residual VQ."

In this paper, we use the encoding structure depicted in Fig. 1(a). The mean of $\boldsymbol{x}$ is first computed and quantized with a mean codebook C_m, and the quantized mean $\hat{m}$ is then subtracted from each component of $\boldsymbol{x}$ to obtain the residual vector $\boldsymbol{r}$. Note that here the residual is computed with respect to the decoder's reproduction of the mean rather than with respect to the true mean. The residual vector $\boldsymbol{r}$ is then quantized with a residual codebook C_r. The output of the encoder includes two indices for the mean and residual, respectively. The corresponding decoder structure is shown in Fig. 1(b). The representation of $\boldsymbol{x}$ offers a simple, and very valuable product code. The reconstructed vector after quantization of the mean and the residual is given by

$$\hat{\boldsymbol{x}}=\hat{\boldsymbol{r}}+\hat{m}\mathbf{1} \tag{12}$$

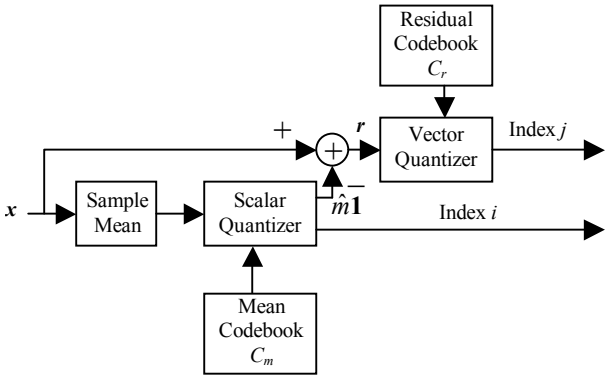
Where $\hat{m}$ is a quantization level from a scalar codebook C_m of size N_m for the mean code levels, and $\hat{\boldsymbol{r}}$ is a codeword chosen from a codebook of size N_r for the residual vectors. In fact, the overall codeword or index is the concatenation of codewords or indices chosen from each of two codebooks. That is to say, this is a product code where the composition function g of the decoder is simply a summation of the reproductions from the different two quantizers. Thus, the equivalent codebook for $\boldsymbol{x}$ is the product codebook C that can be generated from the Cartesian product $C_m\times C_r$. Compared to the full search VQ with the product codebook C, the mean-removed VQ can reduce the complexity from $N=N_m\times N_r$ to $N_m/k+N_r$.

3.2 The embedding process

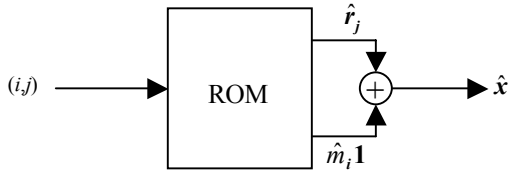
Before describing the proposed algorithm, we make some assumptions. Let $\boldsymbol{X}$ be the original image with size $A\times B$, let

$\boldsymbol{W_R}$ and $\boldsymbol{W_F}$ be the binary robust and fragile watermarks with size $A_w \times B_w$, respectively. Here, a small visually meaningful binary image $\boldsymbol{V}$ with size $a \times b$ is replicated periodically to obtain the binary fragile watermark $\boldsymbol{W_F}$ with size $A_w \times B_w$ that is large enough for embedding. In the proposed algorithm, only one bit is embedded in the mean or residual index of each image block (or vector), so the dimension of each input vector or codeword is $k=(A/A_w)\times(B/B_w)$. Assume that the mean codebook is $C_m=\{\hat{m}_0, \hat{m}_1, \ldots, \hat{m}_{N_m-1}\}$ with size $N_m=2^{n_m}$ and the residual codebook is $C_r=\{\hat{\boldsymbol{r}}_0, \hat{\boldsymbol{r}}_1, \ldots, \hat{\boldsymbol{r}}_{N_r-1}\}$ with size $N_r=2^{n_r}$, where n_m and n_r are natural numbers. Thus a binary number with n_m+n_r bits, in which the first n_m bits stand for the mean index and the last n_r bits denote the residual index, can represent the overall index. The overall codeword can be selected from the equivalent product codebook $C=\{\boldsymbol{c}_0,\boldsymbol{c}_1,\ldots,\boldsymbol{c}_{N-1}\}$ with size $N=N_m\times N_r$. In other words, if the index in codebook C_m is i and the index in codebook C_r is j, then the equivalent overall index in the product codebook C is $j+i\times N_r$.

In our algorithm, the robust watermark $\boldsymbol{W_R}$ and the fragile watermark $\boldsymbol{W_F}$ are embedded in two quantizers respectively. Here, we can adopt two embedding orders. One is to embed the fragile watermark in the mean quantizer and the robust in the residual quantizer; the other is to embed the robust one first and then the fragile. However, experimental results show that we'd better embed the robust watermark in the mean quantizer and the fragile one in the residual quantizer to enhance the robustness of the proposed algorithm. In what follows, we describe the two embedding processes separately.



(a) The encoder



(b) The decoder

Figure 1. Mean-removed VQ

3.2.1 The robust watermark embedding process

In the proposed algorithm, we adopt the method [20] based on index properties to embed the robust watermark in the mean codeword indices as shown in Fig. 2. For convenience of description, the mean scalar quantization here is looked upon as the mean vector quantization (VQ_m), where all components of a mean vector (or codeword) are equal to its mean value. The original watermark $\boldsymbol{W_R}$ is first permuted by a predetermined key, key_1, to generate the permuted watermark $\boldsymbol{W_{RP}}$ for embedding. The polarities $\boldsymbol{P}$ can then be calculated with (3)-(6). Finally, we generate the final embedded watermark or the secret key, key_2, with the exclusive-or operation (7). After the robust embedding, we can obtain the reconstructed image $\boldsymbol{X'}$ and the residual image $\boldsymbol{X}_r$ as follows

$$\boldsymbol{X'} = \mathrm{VQ}_m^{-1}[\mathrm{VQ}_m[\boldsymbol{X}]] \tag{13}$$

$$\boldsymbol{X}_r = \boldsymbol{X} - \boldsymbol{X'} \tag{14}$$

According to Section 2.3, we know that this method has two problems. However, in our algorithm, these two problems can be automatically solved, which will be discussed later in the extraction process.

3.2.2 The fragile watermark embedding process

To embed one bit in each residual index, we can adopt an index constrained vector quantization (ICVQ) encoding scheme as shown in Fig. 3. Because each index has n_r bits, we can select an embedding position from n_r candidate positions. Assume that we select Position key_3, which is considered as a key, to embed the watermark bit, where $0 \le key_3 \le n_r-1$. Unlike the normal VQ encoder, the embedding process for each watermark bit can be performed by searching the best match codeword $\hat{\boldsymbol{r}}_p$ for each input residual vector under the constraints that the key_3-th bit of index p is equal to the watermark bit to be embedded. After the normal VQ decoder, we can obtain the reconstructed residual image $\boldsymbol{X}'_r$ as follows

$$\boldsymbol{X}'_r = \mathrm{VQ}_r^{-1}[\mathrm{ICVQ}_r[\boldsymbol{X}_r]] \tag{15}$$

And then we can obtain the final watermarked image $\boldsymbol{X_W}$ as follows

$$\boldsymbol{X_W} = \boldsymbol{X'} + \boldsymbol{X}'_r \tag{16}$$

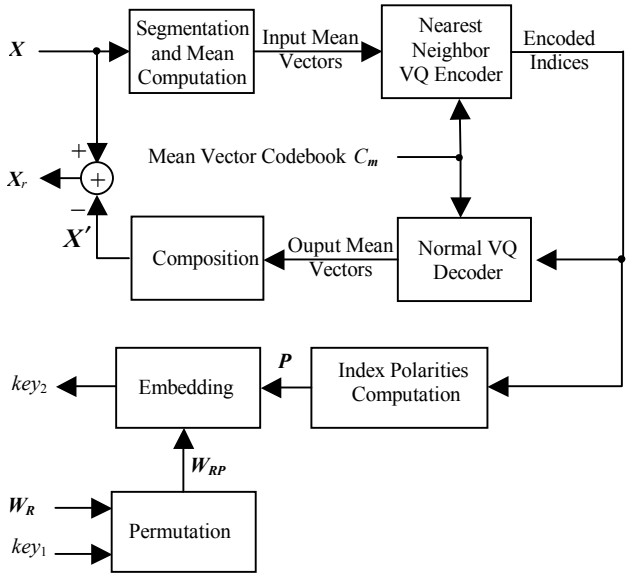


Figure 2. The robust watermark embedding process

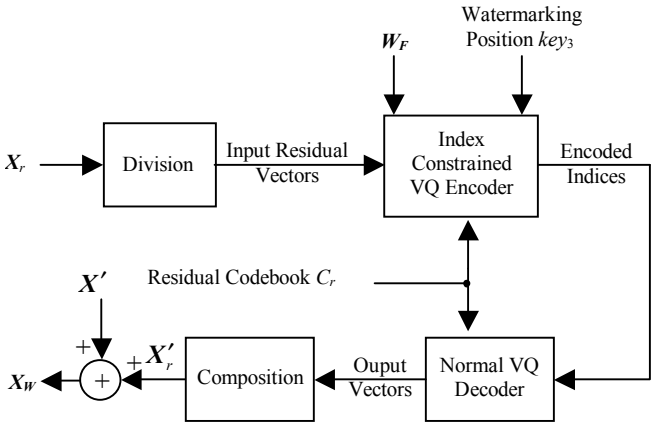


Figure 3. The fragile watermark embedding process

3.3 The extraction process

To enhance the security of our embedding process, we use the equivalent product codebook C in the extraction process as shown in Fig. 4, that is to say, the mean and residual codebooks are used as secret keys while the product codebook is open for user. In addition, because the users don't know the mean and residual codebook sizes used in mean-removed VQ either, how to segment the overall index into the mean index and the residual index is also a secret key, key_5, to users. In order to make the embedding algorithm more secretly, we can also permute the product codebook and then publicize the permuted codebook C_u for users.

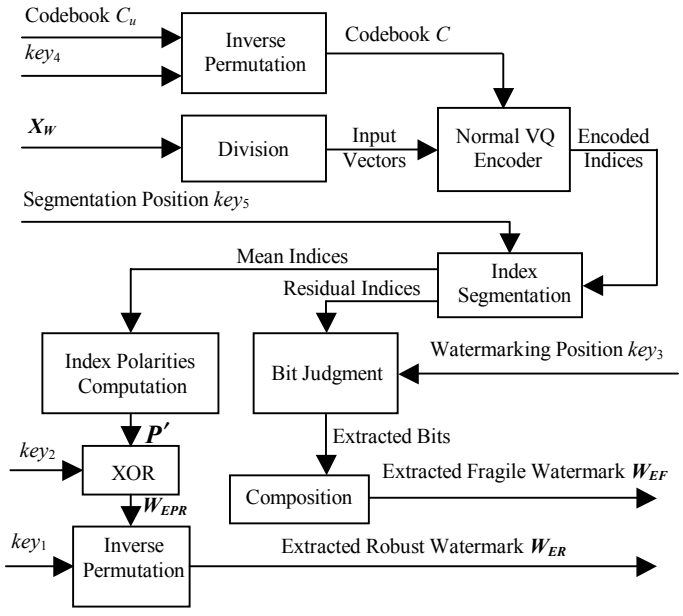


Figure 4. The watermark extraction process

The extraction process can be performed without the original image and can be described as follows: Firstly, perform the inverse permutation operation with key_4 on Codebook C_u to obtain the product codebook C. Secondly, the watermarked image X_W is divided into blocks or vectors. Thirdly, the normal VQ encoder performs the nearest neighbor codeword search on all input vectors to obtain the encoded overall indices. Fourthly, according to the two codebook sizes, each overall index is segmented into two indices. One is for robust watermark extraction; the other is for fragile watermark extraction. Finally, the robust and fragile watermarks are extracted independently. For the robust watermark extraction, we first compute the polarities P from the mean indices, and then perform XOR operation between P and key_2 to obtain the extracted permuted robust watermark W_{EPR}, and finally perform inverse permutation operation with key_1 to obtain the extracted robust watermark W_{ER}. For the fragile watermark extraction, we can simply check the key_3-th bit of each residual index to obtain the extracted watermark bit, where key_3 is just the watermarking position, and then piece all extracted bits together to form the extracted fragile watermark W_{EF}.

In Section 2.3, we point out two problems of the robust embedding technique [20]. However, in our algorithm, these two problems can be automatically solved. Detecting the inexistence of the fragile watermark in the original image can solve the first problem. Using not the mean and residual codebooks but the equivalent product codebook to extract the watermarks can solve the second one. From Fig. 4 we can see that the extraction time is determined by the codebook size of C. If N is very large, then the full search VQ encoding is rather a time-consuming process, so fast codeword search algorithm [23] is used in the proposed algorithm.And then we can obtain the final watermarked image X_W as follows

$$X_W = X' + X'_r \tag{16}$$

4. Experimental Results

To evaluate the performance of the proposed method, the 512×512 Lena image with 8bits/pixel resolution is used for multipurpose watermarking. The Lena image is divided into 16384 blocks of size 4×4 for VQ encoding. A binary image of size 32×32 is replicated for 16 times to obtain a binary watermark W_F with size 128×128 for fragile watermarking. Another binary watermark W_R with size 128×128 is used for robust watermarking. The original Lena image and two watermarks are shown in Fig. 5 (a)-(c). The mean codebook C_m with size 16 and the residual codebook C_r with size 256 are obtained by the well-known LBG algorithm [22], which corresponds to 4+8=12 bits per overall index. If we embed the fragile watermark in the residual index, then we can randomly select the watermarking position key_3 ranged from 0 to 7 for the fragile watermarking. Before extraction, the equivalent product codebook C with size 16×256=4096 can be generated by the Cartesian product $C_m \times C_r$.

(a) (b) (c)

Figure 5. Original image and watermarks. (a) Original Lena image. (b) Original robust watermark. (c) Original fragile watermark.

First, we make an experiment upon the order of embedding robust and fragile watermarks to show why we should embed the robust watermark in the mean index and the fragile one in the residual index. Fig. 6(a) shows the watermarked image with PSNR=30.40dB obtained by the proposed method, and Fig. 6(b) shows the watermarked image with PSNR=26.97dB obtained by the algorithm in the reverse embedding order. From these results, we can see that the proposed embedding order can obtain higher image quality than the alternative one. The first reason is that the mean codebook is small. The second reason is that the robust embedding algorithm [20] doesn't modify the encoded indices at all while the fragile watermarking method does. If we embed the fragile watermark in the mean index, then the reconstructed residual image may be very poor, which affects the whole reconstructed image very much.

In this paper, we employ the normalized Hamming distance, NHD, to evaluate the effectiveness of the proposed algorithm. The NHD between the embedded binary watermark W and the extracted one $\boldsymbol{W}'$ is defined as

$$\mathrm{NHD} = \frac{\mathrm{HD}(\boldsymbol{W}, \boldsymbol{W}')}{A_w \times B_w} \tag{17}$$

Where HD($\cdot$, $\cdot$) denotes the Hamming distance between two binary strings, i.e., the number of bits different in the two binary strings. We can easily prove that NHD$\in$ [0,1]. If we acquire the higher NHD values, the embedded watermark is more similar to the extracted one. Fig.7 shows the robust and fragile watermarks extracted from the watermarked image without any attack. Both NHD values are equal to 1.0, which means that the proposed algorithm is able to extract the watermarks perfectly because the embedded watermarks and the extracted ones are identical.

(a) (b)

Figure 6. Watermarked images obtained by algorithms in different embedding orders. (a) Watermarked image with PSNR=30.40dB obtained by the proposed method. (b) Watermarked image with PSNR= 26.97dB obtained by the algorithm in the reverse embedding order.

(a) (b)

Figure 7. Extracted watermarks under no attacks. (a) Extracted robust watermark, NHS=1.0. (b) Extracted fragile watermark, NHS=1.0.

To check the robustness and fragility of our algorithm, we perform several attacks on the watermarked image, including JPEG compression, VQ compression, spatial image processing and rotation. In addition, we also do the experiment of watermark extraction from the original image in which no watermark is embedded at all. In what follows, we give the experimental results in five subsections.

4.1 Watermark extraction from the original image

The experimental results of watermark extraction from the original image are shown in Fig. 8. In this experiment, we use not the product codebook but the mean and residual codebooks in the extraction process. From the results, we can see that, although we can extract the robust watermark with NHD=1.0 from the original image, we cannot extract the fragile watermark from the original image. Thus, we can decide that there are no watermarks embedded in the original image. The first problem of algorithm [20] described in Section 2.3 is therefore solved.

4.2 JPEG compression attacks

In this experiment, we perform JPEG compression with different quality factors (QF) on the watermarked image as shown in Fig. 9 with QF=100%, 80%, 50% and 30%, respectively. The extracted watermarks and NHD values are depicted in Fig. 10. From these results, we can see that the proposed algorithm is robust to JPEG compression. For the case that QF is larger than 80%, the extracted watermarks, both robust and fragile, are similar to the embedded ones. For all cases, the extracted robust watermarks are with relatively high NHD values. From these results, we can see that the VQ indices can, to some extent, tolerate the incidental distortions induced by high-quality JPEG compression.

(a) (b)

Figure 8. The watermarks extracted from the original image using the mean and residual codebooks. (a) Extracted robust watermark, NHS=1.0. (b) Extracted fragile watermark, NHS=0.002.

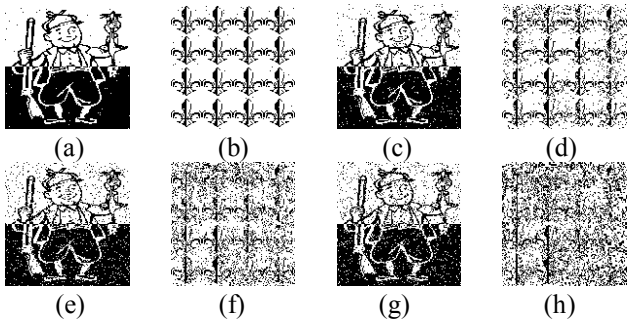
(a) (b)

(c) (d)

Figure 9. The JPEG compressed watermarked images. (a) QF=100% . (b) QF=80%. (c) QF=50%. (d) QF=30%.

4.3 VQ compression attacks

Here, we use four different codebooks to compress the watermarked image. Codebook 1 is the product codebook used in our method. Codebook 2 with size 8192 and Codebook 3 with size 256 are both trained from the Lena image. Codebook 4 with size 4096 is trained from the Pepper image. Fig. 11 shows the four VQ-attacked watermarked images, and Fig. 12 shows the watermarks extracted from these images. From these results, we can see that the proposed algorithm can extract the same watermarks as the embedded ones from the VQ compressed watermarked image with the product codebook. The reason is that the watermarked image isn't modified under the VQ compression with the product codebook. For other cases, the robust watermark can tolerate the VQ compression, while the fragile watermark cannot. The higher the codebook performance is, the larger the NHD value of the fragile watermark is.



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

(e) (f) (g) (h)

Figure 10. The watermarks extracted from JPEG compressed watermarked images. (a) The robust watermark extracted from Fig. 9(a), NHS=0.99. (b) The fragile watermark extracted from Fig. 9(a), NHS=0.99. (c) The robust watermark extracted from Fig. 9(b), NHS=0.94. (d) The fragile watermark extracted from Fig. 9(b), NHS=0.83. (e) The robust watermark extracted from Fig. 9(c), NHS=0.88. (f) The fragile watermark extracted from Fig. 9(c), NHS=0.61. (g) The robust watermark extracted from Fig. 9(d), NHS=0.85. (h) The fragile watermark extracted from Fig. 9(d), NHS=0.42.

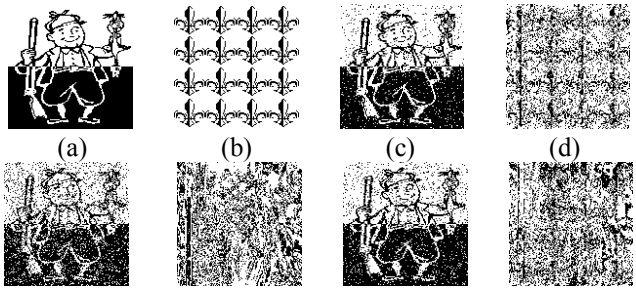
(a) (b)

(c) (d)

Figure 11. The VQ compressed watermarked images. (a) By Codebook 1. (b) By Codebook 2. (c) By Codebook 3. (d) By Codebook 4.

4.4 Spatial-domain image processing attacks

Several spatial-domain image processing techniques, including image cropping, median filtering, blurring, sharpening, contrast enhancement, adding Guassian noise are performed on the watermarked image as shown in Fig. 13. The extracted watermarks are depicted in Fig. 14. For each case, the robust watermark can successfully survive with NHD>0.77. For the case of image cropping in the upper-left corner, the extracted fragile watermark can locate the cropping position. For each case, the fragile watermark can be used to verify the authenticity of the watermarked image.



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

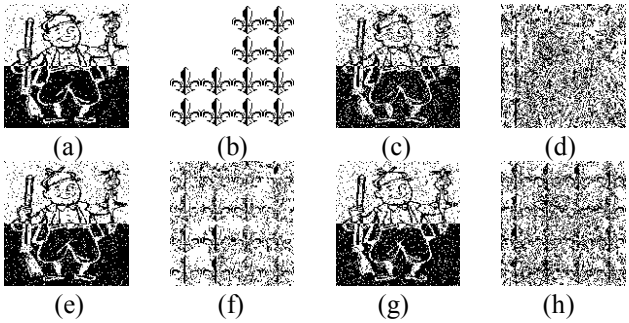
(e) (f) (g) (h)

Figure 12. The watermarks extracted from VQ compressed watermarked images. (a) The robust watermark extracted from Fig. 11(a), NHS=1.0. (b) The fragile watermark extracted from Fig. 11(a), NHS=1.0. (c) The robust watermark extracted from Fig. 11(b), NHS=0.89. (d) The fragile watermark extracted from Fig. 11(b), NHS=0.57. (e) The robust watermark extracted from Fig. 11(c), NHS=0.72. (f) The fragile watermark extracted from Fig. 11(c), NHS=0.17. (g) The robust watermark extracted from Fig. 11(d), NHS=0.83. (h) The fragile watermark extracted from Fig. 11(d), NHS=0.29.

Figure 13. The spatial-domain-attacked watermarked images. (a) Image cropping in the upper-left corner. (b) Median filtering with the radius of 2 pixels. (c) Blurring with radius=1.0 and threshold=10.0. (d) Sharpening. (e) Contrast Enhancement by 10%. (f) Adding Guassian noise by the amount of 4%.

4.5 Rotation attacks

With StirMark, we can perform the geometric attack by rotating the watermarked image with some angles. We rotate the watermarked image by 0.5^{o} and 1^{o} in clockwise and counter- clockwise directions as shown in Fig. 15, and the extracted watermarks are shown in Fig. 16. Although the NHD value of the extracted robust watermark in each case is somewhat smaller in our algorithm, the information conveyed therein is still recognizable. From these results, we can demonstrate the robustness of the robust watermark and the fragility of the fragile watermark to rotation operations.



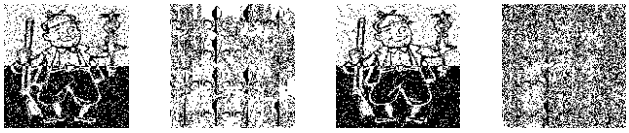


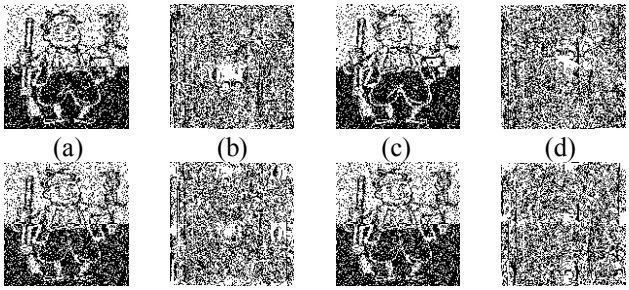
Figure 14. The watermarks extracted from spatial-domain-attacked watermarked images. (a) The robust watermark extracted from Fig. 13(a), NHS=0.90. (b) The fragile watermark extracted from Fig. 13(a), NHS=0.89. (c) The robust watermark extracted from Fig. 13(b), NHS=0.78. (d) The fragile watermark extracted from Fig. 13(b), NHS=0.25. (e) The robust watermark extracted from Fig. 13(c), NHS=0.86. (f) The fragile watermark extracted from Fig. 13(c), NHS=0.63. (g) The robust watermark extracted from Fig. 13(d), NHS=0.83. (h) The fragile watermark extracted from Fig. 13(d), NHS=0.51. (i) The robust watermark extracted from Fig. 13(e), NHS=0.80. (j) The fragile watermark extracted from Fig. 13(e), NHS=0.25. (k) The robust watermark extracted from Fig. 13(f), NHS=0.83. (l) The fragile watermark extracted from Fig. 13(f), NHS=0.22.

5. Conclusions

An efficient multipurpose watermarking algorithm based on mean-removed VQ has been presented. In the proposed algorithm, the robust watermark is embedded in the mean index using the robust watermarking method based on index properties [20] and the fragile watermark is embedded in the residual index using a simple index constrained method. Although the encoded indices of the attacked watermarked image may be very different from the original ones, the variance of neighboring mean indices doesn't vary too much. This watermarking method is therefore robust. On the other hand, the residual watermarking method is based on an index constrained codeword search procedure, in which the index is modified according to the bit to be embedded. Any change in the encoded residual indices may introduce the change in the extracted watermark bit. In other words, this watermarking method can tolerate few modifications, so it is fragile to most intentional attacks. Experimental results demonstrate that the proposed method can be used for copyright protection by extracting the robust watermark, and it can also be used for image authentication by extracting the fragile watermark.

(c) (d)

Figure 15. The rotated watermarked images. (a) Rotation by 0.5o in the clockwise direction. (b) Rotation by 0.5o in the counter-clockwise direction. (c) Rotation by 1o in the clockwise direction. (d) Rotation by 1o in the counter-clockwise direction.



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

(e) (f) (g) (h)

Figure 16. The watermarks extracted from rotated watermarked images. (a) The robust watermark extracted from Fig. 15(a), NHS=0.69. (b) The fragile watermark extracted from Fig. 15(a), NHS=0.13. (c) The robust watermark extracted from Fig. 15(b), NHS=0.69. (d) The fragile watermark extracted from Fig. 15(b), NHS=0.14. (e) The robust watermark extracted from Fig. 15(c), NHS=0.61. (f) The fragile watermark extracted from Fig. 15(c), NHS=0.15. (g) The robust watermark extracted from Fig. 15(d), NHS=0.61. (h) The fragile watermark extracted from Fig. 15(d), NHS=0.13.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under grant 60272074 and Program for New Century Excellent Talents in University of China under grant NCET-04-0329 and Foundation for the Author of National Excellent Doctoral Dissertation of P. R China (No. 2003027).

References

[1] J. J. K. O'Ruanaidh, W. J. Dowling and F. M. Boland. "Watermarking Digital Images for Copyright Protection", *IEE Proceedings-Vision, Image and Signal Processing*, 143 (4), pp. 250-256, 1996.

[2] I. J. Cox, J. Kilian, F. T. Leighton and T. Shamoon. "Secure Spread Spectrum Watermarking for Multimedia", *IEEE Transactions on Image Processing*, 6 (12), pp. 1673-1687, 1997.

[3] M. D. Swanson, Z. Bin and A. H. Tewfik. "Multiresolution Scene-based Video Watermarking Using Perceptual Models", *IEEE Journal on Selected Areas in Communication*s, 16 (4), pp. 540-550, 1998.

[4] G. Voyatzis and I. Pitas. "The Use of Watermarks in the Protection of Digital Multimedia Products", *Proceedings of the IEEE*, 87 (7), pp. 1197–1207, 1999.

[5] S. Pereira and T. Pun. "An Iterative Template Matching Algorithm Using the Chirp-Z Transform for Digital Image Watermarking", *Pattern Recognition*, 33(1), pp. 173-175, 2000.

[6] Y. Wang, J. F. Doherty and R. E. Van Dyck. "A Wavelet-based Watermarking Algorithm for Ownership Verification of Digital Images, *IEEE Transactions on Image Processing*, 11(2),pp. 77-88, 2002.

[7] P. S. L. M. Barreto, H. Y. Kim and V. Rijmen. "Toward Secure Public-key Blockwise Fragile Authentication Watermarking", *IEE Proceedings- Vision, Image and Signal Processing*, 149(2), pp. 57-62, 2002.

[8] M. U. Celik, G. Sharma, E. Saber and A. M. Tekalp. "Hierarchical Watermarking for Secure Image Authentication with Localization", *IEEE Transactions on Image Processing*, 11 (6), pp. 585-595, 2002.

[9] L. Jaejin and S. W. Chee. "A Watermarking Sequence Using Parities of Error Control Coding for Image Authentication and Correction", *IEEE Transactions on Consumer Electronics*, 46(2), pp. 313-317, 2000.

[10] D. Kundur and D. Hatzinakos. "Digital Watermarking for Telltale Tamper Proofing and Authentication", *Proceedings of the IEEE*, 87 (7), pp. 1167-1180, 1999.

[11] C. S. Lu and H.Y.M. Liao. "Multipurpose Watermarking for Image Authentication and Protection", *IEEE Transactions on Image Processing*, 10 (10), pp. 1579-1592, 2001.

[12] C. S. Lu, H. Y. M. Liao and L. H. Chen. "Multipurpose Audio Watermarking". In *Proc. 15th Int. Conf. Pattern Recognition*, vol. 3, pp. 282-285, 2000.

[13] C. S. Lu, S. K. Huang, C. J. Sze and H. Y. M. Liao. "Cocktail Watermarking for Digital Image Protection", *IEEE Transactions on Multimedia*, 2 (4), pp. 209-224, 2000.

[14] C. Busch and S. D. Wolthusen. "Tracing Data Diffusion in Industrial Research with Robust Watermarking". In 2001 *IEEE Fourth Workshop on Multimedia Signal Processing*, pp. 207-212, 2001.

[15] Z. M. Lu and S. H. Sun. "Digital Image Watermarking Technique Based on Vector Quantisation", *Electronics Letters*, 36 (4), pp. 303-305, 2000.

[16] Z. M. Lu, J. S. Pan and S. H. Sun. "VQ-based Digital Image Watermarking Method", *Electronics Letters*, 36 (14), pp. 1201-1202, 2000.

[17] Z. M. Lu, C. H. Liu and S. H. Sun. "Digital Image Watermarking Technique Based on Block Truncation Coding with Vector Quantization", *Chinese Journal of Electronics*, 11 (2), pp. 152-157, 2002.

[18] J. Minho and K. HyoungDo. "A Digital Image Watermarking Scheme Based on Vector Quantisation", *IEICE Trans. Information and Systems*, E85-D (6), pp. 1054-1056, 2002.

[19] A. Makur and S. S. Selvi. "Variable Dimension Vector Quantization Based Image Watermarking", *Signal Processing*, 81 (4), pp. 889-893, 2001.

[20] H. C. Huang, F. H. Wang and J. S. Pan. "A VQ-based Robust Multi-watermarking Algorithm", *IEICE Transactions on Fundamentals*, E85-A (7), pp. 1719-1726, 2002.

[21] H. C. Huang, F. H. Wang and J. S. Pan. "Efficient and Robust Watermarking Algorithm with Vector Quantisation", *Electronics Letters*, 37 (13), pp. 826-828, 2001.

[22] Y. Linde, A. Buzo and R. M. Gray. "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications*, 28 (1), 84-95, 1980.

[23] Z. M. Lu, J. S. Pan and S. H. Sun. "Efficient Codeword Search Algorithm Based on Hadamard Transform", *Electronics Letters*, 36 (16), pp. 1364-1365, 2000.

Author Biographies

Zhe-Ming Lu was born in Zhejiang Province, China, in 1974. He received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in measurement technology and instrumentation from the Harbin Institute of Technology (HIT), Harbin, China, in 1995, 1997, and 2001, respectively. He became a Lecturer with HIT in 1999. Since 2003, he has been a Professor with the Department of Automatic Test and Control, HIT. He has published more than 110 papers and two books (in Chinese). He also participated in a chapter entitled "Watermarking Based on Vector Quantization" in the book Intelligent Watermarking Techniques by J. S. Pan, H.-C. Huang, and L. C. Jain (editors) (Singapore: World Scientific, 2004). His current research interests include speech coding, image processing, and information security.

Wei-Min Zheng was born in Zhejiang Province, China, in 1971. He received the B. S. degree in Electrical & Electronic Engineering in Harbin University of Science and Technology, Harbin, China, in 1993. He received the Ph.D. degree in Testing and Metering Technology and Instrumentation in Harbin Institute of Technology, Harbin, P. R. China, in 2001. Since 2001, he has been engaged in the research and development of Computer Hardware (Godson CPU and its Motherboard) and System-On-Chip Design. He has more than 2 years Practical Experiences on Computer Hardware Architecture and more than 5 years Practical Experiences on Testing and Metering Technology and Instrumentation, and more than 10 years Practical Experiences on Hardware system PCB design and debugging. His current research interests include computer architectures and digital watermarking.

Jeng-Shyang Pan received the B. S. degree in Electronic Engineering from the National Taiwan University of Science and Technology, Taiwan in 1986, the M. S. degree in Communication Engineering from the National Chiao Tung University, Taiwan in 1988, and the Ph.D. degree in Electronic Engineering from the University of Edinburgh, U.K. in 1996. He became a Member (M) of IEEE in 2003. Currently, he is a Professor in the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan. Professor Pan has published more than 45 international journal papers and 80 conference papers. His current research interests include data mining, information security and image processing.