A robust associative watermarking technique based on vector quantization

Jau-Ji Shen\textsuperscript{a,}\textsuperscript{*}, Jia-Min Ren\textsuperscript{b}

\textsuperscript{a} Department of Management Information Systems, National Chung Hsing University, 250, Kuo Kuang Road, Taichung, Taiwan 402, ROC
\textsuperscript{b} Department of Computer Science, National Tsing Hua University, No. 101, Section 2, Kuang-Fu Road, Hsinchu, Taiwan 30013, ROC

\textbf{ARTICLE INFO}

Article history:
Available online 24 October 2009

Keywords:
Association rules
Digital image
Digital watermarking
Vector quantization (VQ)

\textbf{ABSTRACT}

In this paper, the concepts of vector quantization (VQ) and association rules in data mining are employed to propose a robust watermarking technique. Unlike ordinary or traditional watermarking techniques, our approach hides association rules of the watermark, instead of the whole watermark; in other words, the embedded information is the association rules of the watermark. First, VQ encoding is performed on the original image and watermark to generate the index tables, and from which association rules are further mined. Subsequently, by embedding the association rules of the watermark into the association rules of the original image, the purpose for watermarking is accomplished. Finally, VQ decoding technique is applied to reconstruct the watermarked image from the watermarked index table. Experimental results show that our proposed method achieves effective resistance against several image processings such as blurring, sharpening, adding in Gaussian noise, cropping, and JPEG lossy compression. Moreover, the embedding capacity is also significantly increased, so any a complex watermark image is still acceptable in this method.

\textcopyright{} 2009 Elsevier Inc. All rights reserved.

1. Introduction

In the era of information explosion, any data exposed on the Internet is easily accessible to anyone. Image, among others, is one of the most frequently used media for message conveyance. However, due to the convenient access to images via the Internet, anyone can easily download, reproduce, modify, or even widely distribute images. All the illegal actions indirectly affect the copyright owners. Hence, copyright protection for digital images has become a critical issue. From the earlier image steganographic techniques \cite{1} to digital watermarking techniques \cite{2–4} provide an effective option to protect intellectual property rights. A digital watermark plays a role as the protector for the image's copyright information. In most cases, a digital watermark can be any trademark, symbol, or random sequence. After embedding a watermark into the image, the watermarked image with the ownership protected is generated. Subsequently, this kind of image can be broadcasted over the Internet. If any dispute over the copyright of the digital image arises, the ownership of the legitimate image owner can be certified with the previous embedded watermark.

In general, applications of watermarking techniques can be roughly split into two categories: spatial domain and frequency domain. Watermarking techniques applied in the frequency domain mainly transform images via particular equations so as to express images as frequency spectrums. Subsequently, watermarking information is further embedded into the coefficients of the frequency domain \cite{5–9}. On the other hand, watermarking techniques used in the spatial domain directly modify coefficients of images to achieve the purpose for watermarking \cite{10–12}. The watermarking technique proposed in...
this paper is based on vector quantization (VQ) [13–17]. Strictly speaking, VQ cannot be categorized into either of the above types; rather, it is a kind of image compression technique that uses indices of codewords in the codebook to represent the input image and finally refers the index table as the original image. In the future, the image is reconstructed and restored according to the index table. In this paper, the concepts of vector quantization (VQ) and association rules in data mining are employed to propose a robust watermarking technique. First, VQ technique was performed on the original image and the watermark in order to obtain their index tables, from which association rules were further mined. The association rules of the watermark were then embedded into the association rules of the original image. Eventually, via the VQ technique, the watermarked index table that had been embedded into the association rules of the watermark was used to reconstruct the watermarked image. Experimental results show that this technique excels similar watermarking techniques in terms of both robustness and information capacity.

The remainder of this paper is organized as follows. Section 2 briefly reviews the techniques applied in this paper, including VQ, the characteristics in VQ index table, association rule, and edge block detection method. Section 3 is divided into three subsections. The first explains the mining process of association rules from the original image and watermark; the second touches upon the hiding procedure of the whole watermarking association rules; and the last elucidates the extraction of the watermarking. Section 4 mainly presents relevant experiments and results, and conclusion of this paper is made in the last section.

2. Background

The following subsections introduce the techniques to be applied in this paper.

2.1. Association rules

Data mining [18–21] refers to the extraction of hidden predictive information from a profusion of data, and association rules have been developed as one of the frequently applied techniques. With association rules obtained from pieces of data in large databases, the underlying relationships among these data can be discovered such that decision-makers can utilize the information to achieve their purposes.

Given a transaction database, where every product is also defined as an item in the item database $I$, and each transaction associated with a unique identifier called TID consists of at least one item. Here, an itemset is a set of items, and a transaction is said to contain an itemset if this itemset is contained in the transaction. In this way, association rules defined upon a transaction database can be described as follows. Let $A$ and $B$ be two itemsets, if $A \subseteq I$, $B \subseteq I$, $A \cap B = \emptyset$, and $|A| + |B| = K$, then $A \Rightarrow B$ is referred as an association rule defined upon $K$-itemset.

This can be briefly explained with the transaction database in Table 1. The set $\{D, G, H\}$ can constitute a 3-itemset association rule (i.e., $\{(D, G) \Rightarrow (H)\}$), which simultaneously appears in T1 and T2. Some kind of a trading habit may be reflected from this association rule, and that is also given as an explanation why association rules are pertinent to data mining.

2.2. Vector quantization (VQ)

A compression technique, vector quantization (VQ), has been widely applied in image coding and speech. This technique also facilitates progressive transmission of images, especially in the circumstances of one-to-many network communication. Briefly, VQ can be defined as a mapping function that maps any vector $v = \{v_1, v_2, \ldots, v_k\}$ in the $k$-dimensional space $R^k$ to a finite subset $C = \{c_i\}$, where $c_i$ is the $i$th codeword in $C$, and $L$ is the codebook size. First, VQ repeats an iterative clustering algorithm (i.e., the well-known LGB algorithm [22]) among vectors so as to generate a representative codebook. In the encoding procedure, the image is segmented into non-overlapping vectors with the same dimension $k$ and then sequentially compared to the codewords in the codebook such that the closest codeword for each $k$-dimensional input vector $v$ can be found. The closest codeword for $v$ is determined by the Euclidean distance $d(v, c_i)$ as the following equation, where $v_j$ and $c_{ij}$ are the $j$th elements of the vector $v$ and $c_i$, respectively.

$$d(v, c_i) = \|v - c_i\| = \left[\sum_{j=0}^{k-1} (v_j - c_{ij})^2\right]^{1/2}$$  \hspace{1cm} (1)
The closest codeword or the nearest codeword, as implied by its name, means that while calculating the Euclidean distance of \( v \) and each codeword in the codebook, the distance between \( v \) and this codeword is the smallest. Once the closest codeword for \( v \) is found, the index \( i \) of the best matching codeword is assigned to the input vector \( v \) for the basis of future VQ decoding. VQ facilitates compression by means of transmitting or storing the index of the codeword instead of the codeword itself. In the decoding phase, the decoder also utilizes the same codebook to perform a simple table look-up operation for each index \( i \), so as to obtain the corresponding codeword \( c_i \). Finally, \( c_i \) is applied to reconstruct the input vector \( v \).

2.3. The characteristics in VQ index table

Some watermarking approaches [13,14] based on index properties have been proposed. These methods first perform VQ encoding on the original image to obtain the index table and then make use of the properties of neighboring indices to embed watermarks.

Given the original image \( X \) with size \( A_X \times B_X \) and a watermark \( W \) with size \( A_W \times B_W \), and then divide \( X \) into vectors \( x_{mn} \) with size \( (A_X/A_W) \times (B_X/B_W) \). Here, \( x_{mn} \) indicates the image block located at \((m,n)\) of the coordinate. Then, each vector \( x_{mn} \) is searching its nearest codeword from a codebook, in which the size of codewords is equal to \((A_X/A_W) \times (B_X/B_W)\). By assigning the matched index for \( x_{mn} \), an index matrix \( Y \) composed of elements \( y_{mn} \) can be obtained, as shown in the following equation.

\[
Y = \text{VQ}(X) = \left[ \text{VQ}(x_{mn}) \right] = [y_{mn}]. \quad 0 \leq m \leq A_X/A_W - 1, \quad 0 \leq n \leq B_X/B_W - 1
\] (2)

For natural images, since VQ indices among neighboring blocks are inclined to be very similar, we can take advantages of this characteristic to calculate the mean of \( y_{mn} \) and the indices of its surrounding blocks with

\[
\mu_{mn} = \frac{1}{9} \sum_{i=m-1}^{m+1} \sum_{j=n-1}^{n+1} y_{ij}
\] (3)

Similarly, the variance of \( y_{mn} \) and the indices of its surrounding blocks is also another index property, and this can be figured out via the following equation.

\[
\sigma^2_{mn} = \frac{1}{9} \sum_{i=m-1}^{m+1} \sum_{j=n-1}^{n+1} y_{ij}^2 - \mu^2_{mn}
\] (4)

Finally, given an appropriate threshold (i.e., the half of the codebook size), if \( \mu_{mn} \) equals or exceeds this threshold, then the polarity of \( \mu_{mn} \) equals to 1; otherwise, it will be set to 0. In the same way, if \( \sigma^2_{mn} \) is no less than this threshold, the polarity of \( \sigma^2_{mn} \) equals to 1; otherwise, it will be 0. Through the above calculations, characteristics of VQ indices can be obtained on the basis of the mean and variance.

2.4. Edge block detection method

In the pattern-based side match vector quantization (PSMVQ) method proposed by Chang et al. [23], each image block also called a sub-image can be categorized as either “edge block” or “smooth block”. The former means that the image block includes more complex contents while the later refers to that the variation in such block is not so distinct. In this approach, an image is first divided into non-overlapping sub-images with size \( u \times u \) (i.e., \( u = 2, 3, 4, \ldots \)). Subsequently, in order to decide whether an image block is an edge block or not, four \( u \times u \) edge masks are applied to recognize its block type. These masks are generated according to the following principles. For the horizontal direction mask, the elements of the top half are all set to \(-1\), and the others elements are assigned 1. In the vertical direction mask, the elements of the left half and the others are set to \(-1\) and 1, respectively. For the \(+45^\circ\) direction mask, except for the elements of the left-bottom to the right-top diagonal are assigned 0, the elements in the left-top half and right-bottom are set to \(-1\) and 1, respectively. In the \(+135^\circ\) direction mask, the elements along the right-bottom to the left-top diagonal are set to 0, and the elements of the right-top half and the left-bottom half are assigned \(-1\) and 1, respectively. Finally, by applying these four masks to do convolution on the image block so as to obtain four edge intensities. Once any one absolute value of the four values is equal to or larger than a given threshold, this image block is considered to be an edge block; otherwise, it is a smooth block. After all image blocks have been processed, an edge map based on image blocks is created.

The following example demonstrates how this approach works. Here, Fig. 1 shows a \( 4 \times 4 \) image block, where every element is a gray value. Fig. 2 illustrates the four \( 4 \times 4 \) masks used to detect edge block. Each mask stands for different direction occurring in the edge block. Afterwards, those masks are applied to do convolution on the \( 4 \times 4 \) image block. If one absolute value of above values is no less than a given threshold, then this \( 4 \times 4 \) image block is an edge block.
3. Proposed method

The proposed method embeds the association rules generated from the watermark into the original image. In other words, this approach hides the association rules of the watermark, rather than the whole watermark. Before proceeding into detailed explanation of the hiding process, we would like to define the original image $X$ with size $A_X \times B_X$ and the watermark $W$ with size $A_W \times B_W$. The original image and watermark are gray level in the range $[0, 255]$. In general, the watermark should be smaller than the original image. However, our approach goes beyond such limitation; that is, the watermark can be larger than the original image.

3.1. Association rules of the original image and watermark

For each block’s index generated by VQ from the original image and watermark, our proposed method will firstly define an association rule on it, and then mines the association rule from each block’s index. The mining process is depicted in Fig. 3. First, both the original image $X$ and the watermark $W$ are divided into non-overlapping blocks with size $k \times k$, and for each block, the codebook $C$ (including $L k^2$-dimensional codewords) is utilized to find the closest codeword so as to obtain the index tables of the original image and watermark, $X_T$ and $W_T$, respectively. The size of $X_T$ and $W_T$ are $(A_X/k \times B_X/k)$ and $(A_W/k \times B_W/k)$. Subsequently, association rules defined upon 4-itemset are exploited to mine association rules from $X_T$ and $W_T$, respectively.
The 4-itemset association rule for each index in the index tables of original and watermark images can be illustrated as \(((1\text{st item}, 2\text{nd item}, 3\text{rd item}) \Rightarrow (4\text{th item}))\). The first three items are utilized to find the nearest original image rules for the watermark rules, and by changing the fourth item’s value of the rule, which is derived from some selected original image blocks, such that the watermark can be embedded. For each block’s rule, its four items are defined below.

1st item: To calculate the mean of its index and the indices of its neighboring eight blocks, if the value is no less than a given threshold \(T_1\), then the 1st item value of this rule is assigned to 1; otherwise, it is set to 0.

2nd item: To calculate the variance of its index and the indices of its neighboring eight blocks, once the calculated value is no less than a given threshold \(T_2\), then the 2nd item value of this rule is set to 1; otherwise, it is set to 0.

3rd item: To determine whether its corresponding codeword is an edge block or not. Here, four \(k \times k\) masks as mentioned above are applied to do convolution on this block. If any one absolute value of those four computed values is no less than a given threshold \(T_3\), this block is considered as an edge block, and the 3rd item value of this rule is set to 1; otherwise, it is set to 0.

4th item: The block corresponding index value indicated in the index table.

After each block in \(X_T\) and \(W_T\) has been mined an association rule, we can get the sets of association rules \(X_R\) and \(W_R\), respectively. Note that the same association rule may be derived from different blocks of \(X_T\), and this also may happen in mining the association rules from \(W_T\).

### 3.2. Embedding process

After the sets of association rules \(X_R\) and \(W_R\) being derived from \(X_T\) and \(W_T\), the hiding procedure of our proposed method can be designed as the following five steps.

**Step 1:** For the association rule sets \(X_R\) and \(W_R\), the first three items play a role as the matching pattern. (For instance, if a rule \(w_r = \{(0, 0, 1) \Rightarrow (9)\}\) in \(W_R\), the matched rules of \(w_r\) being selected from \(X_R\) should have the pattern as \(\{(0, 0, 1) \Rightarrow (\ast)\}\), where \(\ast\) indicates any value in the interval \([0, L - 1]\).) This step is repeated until each rule in \(W_R\) has found at least one matched rule in \(X_R\). In this matching process, if there is any rule in \(W_R\) unable to find a matched rule in \(X_R\) then go to Step 2; otherwise, go to Step 3.

**Step 2:** For any rule in \(W_R\) has no matched rule in \(X_R\), then any of the first three item values of this rule must be changed, and go to Step 1 to find its matched rule again. (For instance, if the rule \(\{(0, 1, 0) \Rightarrow (20)\}\) in \(X_R\) then change it into \(\{(1, 1, 0) \Rightarrow (20)\}\) in order to search the matched rules as \(\{(1, 1, 0) \Rightarrow (\ast)\}\) in \(X_R\) again. By repeating this process, each non-matched rule in \(W_R\) will finally find at least one matched rule in \(X_R\).)

**Step 3:** Since \(w_r\) may have more than one matched rules, say \(s\) rules in general \((i.e., x_{r1}, x_{r2}, \ldots, x_{rs})\), an appropriate rule should be selected from these \(s\) rules \(x_{r1}, x_{r2}, \ldots, x_{rs}\) in order to embed \(w_r\). The selected rule \(x_t\) is so-called the most similar rule to \(w_r\). The similarity is defined as follows.

For each \(x_{rt}\), and \(1 \leq t \leq s\)

\[
\text{MSE} = \frac{1}{k \times k} \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} (c_{w_r}(i, j) - c_{x_t}(i, j))^2
\]

\[
\text{similarity} = \frac{1}{|\text{MSE} - TM|}, \quad \text{TM is a given threshold}
\]

where \(c_{w_r}\) means the codeword denoted by the index taken from the 4th item value of \(w_r\), and \(c_{x_t}\) refers to the codeword denoted by the index taken from the 4th item value of rule \(x_{rt}\) which represents each rule of \(x_{r1}, x_{r2}, \ldots, x_{rs}\).

**Step 4:** For any rule \(x_t\) in \(\{x_{r1}, x_{r2}, \ldots, x_{rs}\}\), it is possible that exists more than one \(X_T\)’s block which derived the same rule \(x_t\). If it is the case, one of these \(x_t\)’s relevant blocks is randomly selected for the watermark information (i.e., \(w_r\)’s) embedding. Subsequently, by replacing the index of this block with the 4th item value of \(w_r\), the purpose for watermarking is successfully achieved.

**Step 5:** Finally, VQ decoding is performed on the watermarked index table, which has been embedded with all the rules in \(W_R\), to reconstruct the watermarked image.

After performing the embedding process in row-major, each \(W_T\)’s block has been assigned by one suitable \(X_T\)’s block to embed its rule. By the same sequence, let the abbreviation \(L_X\) stand for the set of all selected \(X_T\)’s block’s locations, and \(I_B\) be the set of original indices of these blocks, and \(I_A\) be the indices of these blocks after embedding. In order to extract the watermark back in the future, the following information should be recorded. (1) key\(_2\): \(L_X\); (2) key\(_3\): \(I_B\); (3) key\(_4\): the MSE values between the codewords with indices as \(I_B(z)\) and \(I_A(z)\), where \(z\) means the \(z\)th element in \(I_B\) and \(I_A\); and (4) key\(_5\): for each element \(I_B(z)\), record all of its corresponding blocks in \(W_T\). In general, memory size required to keep the above information would be about 10% volume of the watermark.
3.3. Extracting process

For a given test image $Y$, the recorded information about the embedded watermark (i.e., $key_1 \sim key_4$) should be provided for the watermark extraction from this image. The detailed procedure is described as follows:

Step 1: Perform VQ encoding on the test image $Y$ to obtain the index table $Y_T$.
Step 2: Use $key_1$ to pick out the watermarked blocks $Y_{TW}$ from $Y_T$.
Step 3: Calculate the MSE values between the codewords indexed by all the elements in $Y_{TW}$ and $key_2$. The method is below:

$$mse(l) = \text{MSE}(c_{Y_{TW}(l)}, c_{key_2(l)})$$

(6)

Let $Q$ be the set comprised of all the $\{mse(l)\}$, in which $l$ is the $l$th element in $Y_{TW}$ and $key_2$; $c_{Y_{TW}(l)}$ and $c_{key_2(l)}$ stand for the codewords with indices $Y_{TW}(l)$ and $key_2(l)$, respectively; and $\text{MSE}(c_{Y_{TW}(l)}, c_{key_2(l)})$ is the MSE value calculated from these two codewords.

Step 4: Given a threshold $T_S$ and $key_3$, set $R = \{mse(l) \mid mse(l) \leq key_3(l) \times T_S\}$, and then calculate

$$P = \frac{|R|}{|Q|}$$

(7)

where $|R|$ and $|Q|$ denote the total quantities of elements in $R$ and $Q$, respectively.

Step 5: Given a threshold $T_S$ to determine whether the value $P$ equals or exceeds $T_S$. If this is the case, $Y$ is treated as a watermarked image and go to Step 6, otherwise, $Y$ is not watermarked, and the extraction is terminated.

Step 6: According to $key_4$, restore each element in $key_2$ into its corresponding locations on the watermark index table.
Step 7: Do VQ decoding with the above results to reconstruct the extracted watermark.

4. Experimental results

The approach proposed in this paper utilizes the VQ encoded index tables of the watermark and original images to generate association rules. Association rules of the watermark are embedded into the association rules of the original image. In this section, various attacks (including cropping, blurring, sharpening, JPEG lossy compression, and adding in Gaussian noise) are launched on the image to examine the robustness of our proposed watermarking technique. Although several VQ-based robust watermarking approaches have been proposed [13–16], we just compare our experimental results with the method proposed by Wu and Chang [17]. This is because that in [13,14], the authors hided watermark information into the secret keys, and then these keys are registered to the third part to preserve the ownership of the original source; in other words, they did not hide watermark information into the original image directly, however, our approach is in such case. In [15,16], random sequences treated as watermark information were embedded into the protected image, however, our approach embeds an image (watermark) into another image (protected image). Besides, [24] also proposed a robust associative watermarking technique, however, the kind of watermark applied in that approach was random sequence, thus, we do not compare our approach with this technique here.

In the experiments of this paper, PSNR (Peak Signal-to-Noise Ratio) is used to evaluate the difference between the watermarked image and the original image; NC (Normalized Correlation) is applied to determine the degree of similarity between the original watermark and the extracted watermark, as shown in Eqs. (8) and (9). Here, $A_X$ and $B_X$ represent the height and width of the original image $X$, respectively. $A_W$ and $B_W$ are the height and width of watermark $W$. $X_W$ and $X'_W$ separately denote the watermarked image and the extracted watermark.

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right) \text{ dB}$$

(8)

where

$$\text{MSE} = \frac{1}{A_X \times B_X} \sum_{i=0}^{A_X-1} \sum_{j=0}^{B_X-1} (X(i, j) - X_W(i, j))^2$$

$$\text{NC} = \frac{\sum_{i=0}^{A_W-1} \sum_{j=0}^{B_W-1} W(i, j) \times X'_W(i, j)}{\sqrt{\sum_{i=0}^{A_W-1} \sum_{j=0}^{B_W-1} [W(i, j)]^2} \sqrt{\sum_{i=0}^{A_W-1} \sum_{j=0}^{B_W-1} [X'_W(i, j)]^2}}$$

(9)

The codebook used in this study adopts the traditional LBG method [22] to train a grayscale Lena image with size $512 \times 512$, and the threshold for the termination of training is set as 0.05. There are 256 codewords in this codebook, and the size of each codeword is $4 \times 4$ pixels. The original image used in the experiments is a grayscale Lena image with size $512 \times 512$ (as shown in Fig. 4(a)). Although our approach can embed grayscale watermark into Lena image, however, since [17] is a binary watermarking technique, in order to make the comparison of different methods objectively, binary watermark is applied in both approaches. Here, watermarks with different sizes are considered in the experiments (as illustrated in Figs. 4(b)–4(d)).
In addition, since the codebook applied in this paper is in grayscale, the whiteness of conventional binary watermark should be altered to 255. In other words, the image pixel of the binary watermark in our method is either 0 or 255. In order to minimize the distortion of the extracted watermark in this approach, the codeword with all values only 0 ($c_0$) and the codeword with all values only 255 ($c_{255}$) will be produced, and then replace these two codewords nearest to $c_0$ and $c_{255}$, respectively. MSE (mean-square error) as mentioned above is utilized to decide two codewords will be modified to $c_0$ or $c_{255}$. Thus, the codeword with the minimum MSE value between $c_0$ and the codeword itself will be changed to $c_0$. Similarly, the codeword with the minimum MSE value between $c_{255}$ and the codeword itself will be changed to $c_{255}$. Besides, since the codebook applied in our method is grayscale, the extracted watermark will be in grayscale form. However, in order to make objective comparisons between our approach and [17], the extracted grayscale watermark will be polarized. That is, the pixels with values larger than or equivalent to 128 will be reset to 255, and the pixels with values smaller than 128 will be reset to 0, respectively.

Experimental parameters of the association rules adopted in this paper are described below. First, for convenience, the thresholds utilized in the 1st and 2nd items, $T_1$ and $T_2$, are set as the half of the codebook size, which is 128. The threshold $T_3$ applied to determine whether an image block is an “edge block” is set to 70, and this value is chosen from many experiments (as indicated in Fig. 5). Under the conditions that $T_1 = 128$, $T_2 = 128$, and $T_3 = 70$, various $T_M$ can be set. Different $T_M$ will affect the quality of the watermarked image directly (as shown in Fig. 6). Experimental results reveal that larger $T_M$ facilitates an embedded image to extract the watermark, therefore $T_M = 80$ is selected since it ensures the better PSNR value. Moreover, thresholds $T_R$ and $T_S$ for the judgment of whether an image is embedded with a watermark or not are set as 0.1 and 0.35, respectively. Experiments aimed in this aspect would be further described in the end of this section.

In the method proposed by Wu and Chang, the setting of experimental parameters is based on the same codebook as employed in our approach, except that these two codewords are not changed into $c_0$ and $c_{255}$. Besides, in the codebook editing stage of their method, setting the parameter as 120 will lead to the best PSNR value of the watermarked image. Related experiments about parameter settings are presented in Fig. 7 and Table 2.

All the watermarks as shown in Fig. 4 are considered to facilitate the comparison between our approach with Wu and Chang’s method. Here, we just show some of those watermarked images (as depicted in Figs. 8 and 9). In Fig. 8, the
Fig. 6. The quality curve between different watermarked images is derived from different $T_M$.

Fig. 7. (a) 64 × 64 watermark; (b) watermarked image without codebook editing, PSNR = 29.956 (dB); (c) watermarked image with codebook editing (threshold = 120), PSNR = 30.234 (dB).

Table 2
Different thresholds for codebook editing vs. watermarked image quality and robustness against attacks.

<table>
<thead>
<tr>
<th>Experiments</th>
<th>Thresholds</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
<th>140</th>
<th>160</th>
<th>180</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blurring once (NC)</td>
<td></td>
<td>0.9547</td>
<td>0.9714</td>
<td>0.9753</td>
<td>0.9783</td>
<td>0.9789</td>
<td>0.9850</td>
<td>0.9860</td>
<td>0.9859</td>
<td>0.9844</td>
</tr>
<tr>
<td>Sharpening once (NC)</td>
<td></td>
<td>0.9417</td>
<td>0.9437</td>
<td>0.9456</td>
<td>0.9496</td>
<td>0.9605</td>
<td>0.9622</td>
<td>0.9621</td>
<td>0.9614</td>
<td>0.9652</td>
</tr>
</tbody>
</table>
Fig. 8. (a), (b) and (c) watermarked images (the proposed method), and the corresponding extracted watermarks (d) \(64 \times 64\), (e) \(128 \times 128\) and (f) \(256 \times 128\).

Fig. 9. (a), (b) and (c) watermarked images (Wu and Chang's method) and the corresponding extracted watermarks: (c) \(64 \times 64\) and (d) \(128 \times 128\). Note that the size of Fig. 4(d), \(256 \times 128\), exceeds the watermarking capacity of their method.

extracted watermarks are listed below each watermarked image. The experiments of Wu and Chang's method are illustrated in Fig. 9.

Furthermore, the well-known image processing software, Photoshop, is applied to perform 32 image attacks (including JPEG lossy compression (quality level 0, 1, 2, ..., 12), sharpening (1–3 times), adding in Gaussian noise (\(\sigma = 2, 4, 6, 8, 10, 15\) and 20), blurring (1–3 times) and 6 kinds of cropping methods) on the watermark images. Then NC is used to compare the difference between the original watermark and the extracted watermark. Figs. 10(a)–10(f) show these 6 kinds of cropping attacks to the watermarked images. After applying these cropping attacks to watermarked images of the proposed method and Wu and Chang's method, we can obtain the corresponding extracted watermarks, as shown in Figs. 11(a)–11(f) and Figs. 11(g)–11(l), respectively. Although some shapes of edges look like saw teeth, the extracted watermarks are still recognizable. These jagged edges appeared in the extracted watermarks are brought about by the distortion during VQ operation. It is clear that no matter which kinds of attacks the watermarked images suffer from, extracted watermarks are still sim-
Fig. 10. (a), (b) and (c) the quarter-cropped, watermarked image; (d), (e) and (f) the half-cropped, watermarked image. In this figure, the considered watermark is Fig. 4(c).

Fig. 11. (a), (b), (c), (d), (e) and (f) watermarks extracted from Figs. 10(a)–10(f) (the proposed method). (g), (h), (i), (j), (k) and (l) watermarks extracted from attacked watermarked images (Wu and Chang’s method).

Fig. 12 illustrates these four attacked images watermarked by using the proposed method and Fig. 13 shows the extracted watermarks of both approaches. NC values calculated from extracted watermarks using the proposed method outperform Wu and Chang’s method. Besides, the

Since the number of processed images is too large for listing, four attacked watermarked images with the worst image quality are selected: (1) JPEG lossy compression (quality level = 0); (2) sharpening three times; (3) adding in Gaussian noise (σ = 20) and (4) blurring three times. Here, Fig. 4(c) is used as the watermark. Fig. 12 illustrates these four attacked images watermarked by using the proposed method and Fig. 13 shows the extracted watermarks of both approaches. NC values calculated from extracted watermarks using the proposed method outperform Wu and Chang’s method. Besides, the
extracted watermarks of the proposed method are recognizable. However, in contrast with the original watermark, Fig. 13(g) is non-recognizable.

For the determination of whether an image is embedded with a watermark, a 512 \times 512 grayscale Lena image and 300 128 \times 128 watermarks are applied so as to generate the appropriate thresholds, \( T_R \) and \( T_S \). The above-mentioned 32 kinds of attacks are also performed on the original image and the watermarked image. Here, “false-positive errors” signifies that an image embedded with a watermark is regarded otherwise, and “false-negative errors” means that a non-watermarked image is regarded as a watermarked image. Experimental results show as Tables 3 and 4.

Generally speaking, when a non-watermarked image is falsely judged as embedded with a watermark (i.e., false-negative errors), the cost to be paid would be relatively higher. Therefore, parameters set for the thresholds that ensure “false-negative errors” near 0 and the minimum “false-negative errors” would be preferred; that is, \( T_R = 0.1 \) and \( T_S = 0.35 \).
Table 3
Rate of false judgment with $T_R = 0.05$ and $0.1$, and various $T_S$.

<table>
<thead>
<tr>
<th>$T_R$</th>
<th>$T_S$</th>
<th>False-negative errors</th>
<th>False-positive errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.3</td>
<td>0.325</td>
<td>0.35</td>
</tr>
<tr>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>91/9600</td>
<td>151/9600</td>
<td>216/9600</td>
</tr>
<tr>
<td>(0.94%)</td>
<td>(1.57%)</td>
<td>(2.25%)</td>
<td>(3.56%)</td>
</tr>
<tr>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5/9600</td>
<td>1/9600</td>
<td>0/9600</td>
</tr>
<tr>
<td>(0.05%)</td>
<td>(0.01%)</td>
<td>(0%)</td>
<td>(0%)</td>
</tr>
</tbody>
</table>

Table 4
Rate of false judgment with $T_R = 0.15$ and $0.2$, and various $T_S$.

<table>
<thead>
<tr>
<th>$T_R$</th>
<th>$T_S$</th>
<th>False-negative errors</th>
<th>False-positive errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.3</td>
<td>0.325</td>
<td>0.35</td>
</tr>
<tr>
<td>0.15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>29/9600</td>
<td>69/9600</td>
<td>110/9600</td>
</tr>
<tr>
<td>(0.3%)</td>
<td>(0.72%)</td>
<td>(1.14%)</td>
<td>(2.44%)</td>
</tr>
<tr>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>55/9600</td>
<td>12/9600</td>
<td>6/9600</td>
</tr>
<tr>
<td>(0.57%)</td>
<td>(0.13%)</td>
<td>(0.06%)</td>
<td>(0.04%)</td>
</tr>
</tbody>
</table>

Fig. 14. Embedding 1024 × 1024 watermarks into the 512 × 512 original images. (a) and (b) 1024 × 1024 binary watermarks; (c) and (d) 512 × 512 watermarked images.

The following experiments are considered to show the superiority in embedding capacity of the proposed method by using two 1024 × 1024 binary and grayscale watermarks (see Figs. 14(a) and 14(b)). The resolution of the watermarked Lena images and extracted watermarks are also shown in Figs. 14(c)–14(d) and Figs. 15(a)–15(b), respectively.

Besides, by applying the same image processings, JPEG lossy compression (quality level = 0), sharpening three times, adding in Gaussian noise ($\sigma = 20$) and blurring three times, to Figs. 14(c) and 14(d), the corresponding extracted binary watermarks are shown as Figs. 16 and 17. It is clear that no matter what kind of attacks the embedded image suffers from, the extracted watermark is still almost the same as the original one. Specifically, NC value is larger than 0.9580 at least.
Fig. 15. Extracted binary watermarks.

(a) NC = 0.9966. (b) NC = 0.9524.

Fig. 16. Extracted binary watermarks from attacked watermarked images: (a) JPEG lossy compression (quality level = 0); (b) sharpening three times; (c) adding in Gaussian noise (σ = 20) and (d) blurring three times.

Finally, in order to demonstrate the proposed method can be easily applied in different kinds of protected images, we also utilize another two 512 × 512 images, boat and pepper, to evaluate the performance. Figs. 18(a) and 18(b) are the two original images, and here the size of watermark is 128 × 128 (Fig. 4(c)). Figs. 18(c) and 18(d) are the watermarked images. Note that PSNR values of these two images are smaller than watermarked image, Lena, since the codebook was trained using Lena only. Figs. 18(e) and 18(f) show the extracted watermarks. Then, we also apply the same image processings, JPEG lossy compression (quality level = 0), sharpening three times, adding in Gaussian noise (σ = 20) and blurring three times, to these two watermarked images, we obtain eight corresponding extracted watermarks, as shown in Figs. 19(a)–19(d) and Figs. 19(e)–19(h), respectively.
(a) NC = 0.9681. (b) NC = 0.9700. (c) NC = 0.9580. (d) NC = 0.9685.

Fig. 17. Extracted binary watermarks from attacked watermarked images: (a) JPEG lossy compression (quality level = 0); (b) sharpening three times; (c) adding in Gaussian noise ($\sigma = 20$) and (d) blurring three times.

Table 5

<table>
<thead>
<tr>
<th>Watermark size</th>
<th>Our method/Wu and Chang’s method</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 × 64</td>
<td>Yes/Yes</td>
</tr>
<tr>
<td>128 × 128</td>
<td>Yes/Yes</td>
</tr>
<tr>
<td>256 × 128</td>
<td>Yes/NA</td>
</tr>
<tr>
<td>1024 × 1024</td>
<td>Yes/NA</td>
</tr>
</tbody>
</table>

5. Conclusions

In this paper, the concepts of vector quantization (VQ) and association rules in data mining are employed to propose a robust watermarking technique. With association rules mined from the original image and the watermark, we embed the association rules of the watermark into the association rules of the original image; in other words, our approach embeds the association rules of the watermark, rather than the watermark itself. The comparison with the method proposed by Wu and Chang has revealed that our approach enables the hiding of a watermark in a size twice or even larger than the original image, as shown in Table 5.

However, in our method, four necessary keys should be recorded after hiding a watermark such that the watermark can be successfully extracted. In general, the extra memory size required to record these data is about 10% size of the watermark. Therefore, strictly speaking, the proposed approach is a “semi-blind” watermarking technique. Here the term “semi-blind” means that some side information, e.g., where the watermark blocks were embedded into the original image, are needed to be stored in the embedding procedure such that the watermark can be obtained during the extracting process [25]. Besides, the proposed approach can also be easily extended to applications of grayscale and color watermarks, while Wu and Chang’s method is a binary watermarking technique. Moreover, in the determination stage of whether an image is watermarked, various threshold settings may affect the rate of false judgment (false-positive errors and false-negative errors). Hence, the two aims of further researches are to eliminate the recorded information and reduce the false judgment rate.
Fig. 18. (a) and (b) 512 × 512 original images; (c) and (d) 512 × 512 watermarked images; (e) and (f) 128 × 128 extracted watermarks.

Fig. 19. Binary watermarks extracted from eight attacked watermarked images: (a) JPEG lossy compression (quality level = 0); (b) sharpening three times; (c) adding in Gaussian noise (σ = 20); (d) blurring three times and (e)–(h) the same as (a)–(d).

References


**Jau-Ji Shen** received the B.S. in Mathematics from Fu-Jen University, Taipei County, Taiwan, Republic of China, in 1982. Two years later, he received M.S. in Information Science Program of Applied Mathematics from National Chung-Hsing University, Taichung, Taiwan. In 1988, he received Ph.D. in Information Engineering and Computer Science from National Taiwan University, Taipei, Taiwan. From 1988 to 1994, he was the leader of the software group in Institute of Aeronautic, Chung-Sung Institute of Science and Technology, ROC. He is currently an Associate Professor of the Department of Management Information Systems, National Chung-Hsing University, Taiwan, ROC. His current research interests include digital images, data mining, and database techniques.

**Jia-Min Ren** received the M.S. in Institute of Information Management from National Formosa University, Yunlin County, Taiwan, Republic of China, in 2007. He is now a Ph.D. candidate in Department of Computer Science, National Tsing Hua University, Taiwan. His research interests include digital image processing, digital watermarking, semantic analysis of musical signals and music information retrieval.