Digital Watermarking Techniques

Hsien-Chu Wu
National Taichung Institute of Technology, Taiwan

Hei-Chuan Lin
National Taichung Institute of Technology, Taiwan

INTRODUCTION

In recent years, services on the Internet have greatly improved and are more reliable than before. However, the easy downloads and duplications on the Internet have created a rush of illicit reproductions. Undoubtedly, the rights of ownership are violated and vulnerable to the predators that stalk the Internet. Therefore, protection against these illegal acts has become a mind-boggling issue.

Previously, artists and publishers painstakingly signed or marked their products to prevent illegal use. However with the invention of digital products, protecting rightful ownership has become difficult. Currently, there are two schemes to protect data on the Internet. The first scheme is the traditional **cryptography** where the important data or secret is to be encrypted by a special process before being transmitted on the Internet. This scheme requires much computational process and time to encrypt or decrypt. On the other hand, the second scheme is **steganography** where the important message or secret is hidden in the digital media. The hidden data is not perceptible by the **human visual system** (**HVS**). The digital **watermarking** technique is an application of **steganography** (Chang, Huang, & Chen, 2000; Chen, Chang, & Huang 2001). In order to safeguard copyrights and rightful ownerships, a representative logo or watermark could be hidden in the image or media that is to be protected. The hidden data can be recovered and used as proof of rightful ownership.

The **watermarking** schemes can be grouped into three kinds, largely, dependent on its application. They use the fragile watermark, semi-fragile watermark, and robust watermark, respectively (Fabien, Ross, & Markus, 1999). Fragile watermarks are easily corrupted when the watermarked image is compressed or tampered with. Semi-fragile watermarks can sustain attacks from normal image processing, but are not robust against malicious tampering. Fragile and semi-fragile watermarks are restricted in its use for image authentication and integrity attestation (Fridrich,2002; Fridrich, Memon, & Goljan, 2000). For the **robust watermarking**, it is always applied in ownership verification and copyright protection (Fridrich, Baldoza, & Simard, 1998; Huang, Wang, & Pan, 2002; Lu, Xu, & Sun, 2005; Solanki, Jacobsoen, Madhow, Manjunath, & Chandrasekaran, 2004). Some basic conditions must be followed: (1) Invisibility: the watermarked image must look similar to its original and any difference invisible to the **human visual system**. (2) Undetectable: the watermark embedded in the image must not be easily detectable by computing processes or statistical methods. (3) Safety: watermark is encrypted and if accessed by a hacker; cannot be removed or tampered with. (4) Robustness: the watermark is able to withstand normal and/or illegal manipulations, such as compression, blurring, sharpening, cropping, rotations and more. The retrieved watermark is perceptible even after these processes. (5) Independence: the watermark can be retrieved without the original image. Last but not the least, (6) Efficiency: the watermarked image should not require large storage and must also allow for a comparable-sized watermark to be hidden in the media.

The proposed method is a **VQ-based watermark** technique that depends on the structure of a **tree growth** for grouping the codebook. The scheme is robust. That is, the watermark is irremovable and also can withstand normal compression process, tampering by compression or other malicious attacks. After these attacks, the watermark must be recovered with comparable perceptibility and useful in providing proof of rightful ownerships.

BACKGROUND

The **watermarking** schemes are classified into the methods of the **spatial domain** and the **frequency domain**, respectively (Chang, Huang, & Chen, 2000; Chen, Chang, & Huang, 2001; Zhao, Campisi, & Kundur, 2004). To hide a watermark in the **frequency domain**, an image has to be transformed from a **spatial domain** into its **frequency domain**. This scheme requires many computations and time to embed/retrieve the watermarks. Meanwhile in the **spatial domain**, the watermark can be directly embedded into the pixels values. The algorithms for embedding and recovering are simple. Traditionally, the scheme involves hiding the watermark bits in the **least significant bits** (**LSB**). More literature on the **LSB** technique can be found in Chan and Cheng (2004) and Chang, Hsiao, and Chan (2003). This scheme is not robust. The watermark is easily corrupted after compression.

VQ-Based Watermarking

VQ is a low bit-rate image compression technique. It is simple and easy to encode and decode. Suppose an original image I is partitioned into small non-overlapped blocks with $m \times m$ pixels. Each block contains m^2 pixels. Before **VQ** encoding, block vectors are trained dispersedly and uniformly from several images. The trained set of the block vectors is called codebook. Each block vector in the codebook is called a codeword. In the **VQ** encoding phase, the codeword closest to the encoded block is chosen and stored as an index value in a table. The procedure is repeated for all the blocks in the image. In **VQ** decoding, indexes from the index table will be used to find the corresponding codeword in the codebook and to recover the image (Chang, Huang, & Chen, 2000; Poggi & Ragozini, 2001; Wu & Shih, 2004).

VQ-based watermarking is an effective **steganography** scheme in **spatial domain** and several algorithms have been proposed in different literatures (Wu & Chang, 2004; Huang, Wang, & Pan, 2002). Many of them rely on modifying codewords to achieve the purpose of watermark embedding. But it makes more distortion of images. Lu et al. (2005) used another scheme in illustrated as follows:

1. Input an image X with size $M \times N$, watermark W with size $M_W \times N_W$.
2. In **VQ** encoding process, X is divided into vectors x's with size

   $$\frac{M}{M_W} \times \frac{N}{N_W},$$

   then find the closest to encoded block vectors from the codebook, yield and record index $y(m, n)$ to table Y. Equation is shown as follows:

   $$Y = VQ(X),\ y(m, n) = VQ(x(m, n). \quad (1)$$

3. Compute the variances of $y(m, n)$ of Y after **VQ** encoding vectors by this equation.

   $$\sigma^2(m,n) = \left(\frac{1}{9}\sum_{i=m-1}^{m+1}\sum_{j=n-1}^{n+1} y^2(i,j)\right) - \left(\frac{1}{9}\sum_{i=m-1}^{m+1}\sum_{j=n-1}^{n+1} y(i,j)\right)^2. \quad (2)$$

   And obtain polarites P:

   $$P = \bigcup_{m=0}^{(M/M_W)-1} \bigcup_{n=0}^{(N/N_w)-1} p(m,n), \text{ where}$$

   $$p(m,n) = \begin{cases} 1, & if\ \sigma^2(m,n) \geq Threshold \\ 0, & otherwise. \end{cases} \quad (3)$$

   W_p will be the watermark W reordered by random function with $key1$.

   $$key2 = W_p \oplus P \quad (4)$$

   After the **VQ** decoding, the reconstructed image and the secret key will be the protection of the original image.

In Lu et al. (2005), the authors also point to the problems of the algorithm: (1) The watermarks are not really embedded into the image. A secret key is only produced by composing the **VQ** index table and watermark messages, so users can retrieve the watermark from the original image where no other watermark had been embedded in it. (2) If the codebook is public then the user can also embed another watermark in the watermarked image without any modification.

THE PROPOSED SCHEME

The proposed method in this article embeds the watermark bit stream into the **VQ** encoding codes directly without modifying the **VQ** codewords anymore. Meanwhile, the watermarked image quality is controlled by **VQ** compression technique. We can use the existing codebook or produce a new one. In the next section, we demonstrate the proposed codewords grouping method by applying the tree growing structure. The proposed watermark embedding and retrieving processes are illustrated in the following sections.

Grouping Codewords by Proposed Tree Growing Structure

The scheme for grouping the codewords is based on a tree growing structure. First, codewords from a codebook are classified into groups by tree growing structure as described in Algorithm 1. Assume $X(x_1, x_2, ..., x_{m \times m})$, the centroid of the codebook, is the root of tree. Let $Y(y_1, y_2, ..., y_{m \times m})$ and $Z(z_1, z_2, ..., z_{m \times m})$ individually be two nodes of two branches such that $y_i = x_i + k$, $z_i = x_i - k$ and k is a constant. Let each codword belong to the nearest branch node and separate all the codewords into two groups. Computation is repeated to the new centroids Y' and Z', and let Y' and Z' be the new nodes in the sub-tree which will grow the same way as before until members of each sub-tree are equal or less than 2. Each group of codewords should be close to each other and their members are distributed to subcodebook 1 and subcodebook 2. The
steps are described in detail in Algorithm 1.