A VQ-Based Image-in-Image Data Hiding Scheme

Feng-Hsing Wang¹, Jeng-Shyang Pan², Lakhmi C. Jain¹ and Hsiang-Cheh Huang³

¹ School of Electrical and Information Engineering, University of South Australia, Australia
² Dept. of Electronic Eng., Nat’l Kaohsiung U. of Applied Sciences, Kaohsiung, Taiwan, R.O.C.
³ Dept. of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

E-mail: wanfy002@students.unisa.edu.au

Abstract

Vector quantization (VQ)-based data hiding and watermarking techniques play a newly developed branch in digital watermarking research fields recently. In this paper, we propose a novel watermarking scheme for image-in-image data hiding based on VQ. Unlike other existing schemes in literature to embed the binary watermark into the original image with VQ, we present an efficient and secure algorithm for embedding a gray-level watermark into the original. The offered scheme is robust to attacks, such as VQ compression and JPEG compression. Moreover, it greatly expands the capacity of the watermarking system. Experimental results demonstrate the superiority of our scheme in watermark robustness and watermark capacity. This also proves the effectiveness of our proposed schemes in VQ-based image-in-image data hiding.

1. Introduction

With the swift growth of computer industry and the popularization and widespread use of Internet, people nowadays easily retrieve digital multimedia items, especially the digital images, through the World Wide Webs. Ironically, because of the digital nature of multimedia, such as easy transportation and flexible editing, they suffer from infringing upon the intellectual properties of original owners of such digital contents. Consequently, issues relating to Digital Rights Management (DRM) are getting more and more important in both academic researches and industrial applications. Therefore, in addition to conventional solutions such as encryption and cryptographic methods, digital watermarking offers another useful solution for copyright protection and DRM.

In this paper, we focus the multimedia formats on digital images. There are three fundamental requirements in designing watermarking algorithms, namely, imperceptibility of the watermarked image, robustness of the watermarking algorithm, and capacity of the watermark, represented by the number of bits to be embedded. Among watermarking researches in literature, vector quantization (VQ) based watermarking [1][2][3][4][5][6] is a newly developed branch that can be further explored for researches and applications. Watermarking schemes based on VQ are getting more attention since they are able to enhance the traditional VQ system with the extra ability of watermarking. Based on the VQ-based schemes for digital still images, Lu et al. [1] proposed the scheme to embed binary watermark bits by modifying VQ indices. For better performances, Huang et al. [2] presented hiding watermark bits into secret keys by referring to neighboring VQ indices. Jo and Kim [3] suggested a codebook partition method to achieve embedding of binary watermark. However, the type of embedded watermarks in literature stated above relating to VQ-based watermarking are binary watermarks, hence the watermarking capacities of these schemes are limited. To improve this deficiency in addition to retaining both the imperceptibility and the robustness requirements, we present a novel watermarking scheme for hiding the 8-bit per pixel, gray-level watermark, with much larger capacities than those in literature.

This paper is organized as follows. In Section 2, we briefly describe the backgrounds and concepts for vector quantization. In Section 3, we discuss the watermarking schemes for hiding and extracting gray-level watermark with VQ. Experimental results are depicted in Section 4 with VQ and JPEG compression attacks, and we also compare those with the results from existing algorithms in literature. And we conclude this paper in Section 5.
2. Backgrounds and Concepts for Vector Quantization

To reduce the space requirement for storage and the bandwidth requirement for communication, a wide variety of compression techniques had been developed [7]. For multimedia applications, less significant information can be sacrificed for higher compression rate, since human sensory system is less sensitive to detailed information. In this kind of applications, vector quantization [8] had received considerable attention for its high compression rate and its essential role in various compression applications. As an extension to scalar quantization, vector quantization works on vectors of raw data. A vector can be fixed numbers of consecutive samples of audio data or a small block of image pixels, for example, the gray-level values of a 4×4 pixel image block forms a 16-dimensional vector. Figure 1 gives a block diagram illustration of the operation of vector quantization compression.

![Figure 1. A block diagram for vector quantization.](image)

As we can see from Figure 1, the codebook plays an essential role in vector quantization. The codebook size, or the number of codewords in a codebook, is a trade-off between the reconstructed image quality and the compression rate. The codewords in the codebook decide the resultant compression distortion. A dedicated procedure is required for the generation of appropriate codebook. Among other alternatives, LBG algorithm [9] is widely used in various applications.

3. Watermarking Algorithm

We proposed the procedures for embedding and extracting the gray-level watermark in the following sub-sections. Figures 2 and Figure 3 also depict the flow charts for the embedding and extraction processes.

3.1. Sub-codebook Extracting and Main Codebook Partition

Given a codebook \( C = \{c_i | 0 \leq i < N\} \) with \( N \) codewords, a sub-codebook \( C_s \) is chosen from some codewords in \( C \) first, by referring to one user-key \( S_1 \). Hence, \( C_s \in C \).

Next, \( C \) is partitioned into \( p \) sub-codebooks \( \{C_0, C_1, ..., C_{p-1}\} \) by referring to another user-key \( S_2 \). Here \( C_0 \cup C_1 \cup ... \cup C_{p-1} = C \), \( C_0 \cap C_1 \cap ... \cap C_{p-1} = \phi \), and \( S_2 \) defines allocations of the codewords in \( C \) to the \( p \) sub-codebooks.

In our watermarking scheme, \( C_s \) is employed to encode the gray watermark \( W \). In splitting the codebook into \( \{C_0, C_1, ..., C_{p-1}\} \), we suggest using \( p = 2^m \) to hide \( m \) watermark bits into each VQ index.

3.2. Encoding Procedure for the Gray Watermark

For a gray watermark \( W \), we employ the traditional VQ encoding procedure with codebook \( C_s \), and we collect the VQ indices as \( I_w \). In order to embed \( I_w \) into the host image \( X \) in the VQ-domain, \( I_w \) has to be split into \( T \) parts. Here \( T \) is equal to the number of non-overlapping blocks consisting of the host image. The following steps illustrate the details of how to split \( I_w \) into the considered number of parts.

- **Step (i)** Convert \( I_w \) into a binary bitstream \( B \).
- **Step (ii)** Decompose \( B \) into \( T \) parts, where the length of each part is \( m \) bits. We use \( B = \{b_0, b_1, ..., b_{T-1}\} \) to denote the decomposed result.
- **Step (iii)** Translate each binary element in \( B \) into a decimal integer, and present the collection of integers by \( J \). The rule is according to their translations from binary formats to decimal formats. We denote \( J = \{j_0, j_1, ..., j_{T-1}\} \), where \( 0 \leq j_i < p \), \( 0 \leq i < T \), to represent the translated result. For example, in the case of \( m = 2 \), if \( b_i = \{1, 0\} \), then the translated result is \( j_i = 2 \); in the case of \( m = 3 \), if \( b_i = \{1, 1, 0\} \), then the translated result is \( j_i = 6 \).

3.3. The Watermark Embedding Procedure

The host image \( X \) is decomposed into \( T \) vectors \( \{x_0, x_1, ..., x_{T-1}\} \) with dimension \( k \). Then, by employing the encoded result \( J \) from Section 3.2, the
following steps illustrate how to hide \( m \) bits into each of the vectors.

**Step (i)** By referring to the \( i \)th integer \( j_i \) of \( J \), the \( j_i \)th sub-codebook \( C_{j_i} \) is selected from \( \{ C_0, C_1, ..., C_{p-1} \} \) as the default codebook for the VQ encoding procedure.

**Step (ii)** Find the nearest codeword from \( C_{j_i} \) and use it to replace \( x_i \).

After dealing with all the vectors, the gray watermark can be embedded into the host image. Figure 2 illustrates the flow chart of the embedding procedures.

**Step (iii)** Convert the index \( j_i \) of \( C_{j_i} \) into a binary stream \( b_i \) with length \( m \); e.g., if \( m = 4 \) and \( j_i = 5 \), then \( b_i = \{0, 1, 0, 1\} \).

After obtaining the embedded bits from the watermarked image, the hidden gray watermark can be recovered by reversing the steps in Section 3.2.

**4. Experimental Results**

In our experiments, the host image is 8-bit per pixel gray image Lena, with size \( 512 \times 512 \). The image in Figure 4(a) represents the 8-bit per pixel, gray level watermark, with size \( 256 \times 256 \). The codebook of size \( 256 \) was trained from Lena with the LBG algorithm [9]. We divide the host image and the watermark into \( 4 \times 4 \) blocks, and construct 16384 and 4096 vectors, respectively. The indices obtained from the watermark are converted into a binary bitstream, which is then split into 16384 parts with dimension \( m = 2 \) bits. To describe the performance, the peak-signal-to-noise ratio (PSNR) is employed. The PSNR value between the host image and the watermarked one is 28.83 dB, and the PSNR value between the embedded watermark and the extracted one under no attack is 32.04 dB. Figure 4 depicts the original watermark and the watermarks extracted from the attacked images. Table I summarizes the capacity of the VQ-based watermarking methods in [1]–[3] in addition to the watermarked image quality. It is reasonable to see that if we embed more bits into the host image, the PSNR values in the watermarked images become lower. The results demonstrate the proposed scheme has superior capacity, and it can survive under the VQ compression and the JPEG compression with different quality factors (QF) successfully.
Figure 4. The original watermark and the extracted ones (256×256 pixels, 8-bit per pixel gray-scale). (a) Original watermark. (b) Extracted watermark under VQ compression with codebook size 256, PSNR = 32.04 dB. (c) Extracted watermark under JPEG compression with QF = 80%, PSNR = 29.36 dB. (d) Extracted watermark under JPEG compression with QF = 60%, PSNR = 20.00 dB.

Table I: Comparison of watermark capacity (in bits) and PSNR (in dB) of the watermarked image among different methods in literature.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Watermark capacity</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method in [1]</td>
<td>33199 bits</td>
<td>30.59 dB</td>
</tr>
<tr>
<td>Proposed</td>
<td>524288 bits 256×256</td>
<td>28.83 dB</td>
</tr>
</tbody>
</table>

5. Conclusion

In this paper, we propose a novel VQ-based watermarking scheme for hiding gray watermark. It does not require the original host image in the watermark extraction procedure. Moreover, no extra codebook is needed during the watermark embedding procedure. To compare with other VQ-based watermarking schemes in literature, the major difference is that the method proposed in this paper is to embed a gray-level watermark into the host image, while others embed binary watermark. Experimental results have shown that the proposed technique can hide much more information bits, and it can also survive under VQ compression and JPEG compression. All the extracted watermarks are easily recognizable to protect the copyright of the original owner. Summing up, the proposed scheme in this paper shows a new direction to greatly expand the capacity of the watermarking scheme, while retaining the robustness requirement in watermarking. Better schemes for optimizing the trade-off among the imperceptibility, robustness, and capacity requirements will be developed in the future.

6. Acknowledgement

This work was supported by National Science Council (Taiwan, R.O.C.) under Grant No. NSC 92–2213–E–151–022.

7. References