ELSEVIER

Pattern Recognition Letters 22 (2001) 1051–1058

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

A novel SVD- and VQ-based image hiding scheme

Kuo-Liang Chung *, Chao-Hui Shen, Lung-Chun Chang

Department of Information Management, Institute of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 10672, Taiwan, ROC

Received 1 November 2000; received in revised form 1 March 2001

Abstract

This paper presents a novel singular value decomposition (SVD)- and vector quantization (VQ)-based image hiding scheme to hide image data. Plugging the VQ technique into the SVD-based compression method, the proposed scheme leads to good compression ratio and satisfactory image quality. Experimental results show that the embedding image is visually indistinguishable from the stego-image. © 2001 Published by Elsevier Science B.V.

Keywords: Compression; Data hiding; Image hiding; PSNR; Singular value decomposition; Steganography; Vector quantization

1. Introduction

Image hiding can be used as a secret image deliver. In this paper, the naming conventions used in (Anderson, 1996; Johnson and Jajodia, 1998) are followed, the image to be hidden is named the embedding image; the image that is used to hold the embedding image is named the cover image, and the image combining the cover image and the embedding image makes a stego-image. Recently, Chen et al. (1998) presented a novel virtual image cryptosystem based upon vector quantization (VQ) (Gersho and Gray, 1992; Gray and Neuhoff, 1998). Wu and Tsai (1998) presented an image hiding scheme using a multiple-based number conversion and a lossy compression technique. Both of the two methods in (Chen et al., 1998; Wu and Tsai, 1998) focused on how to hide the compressed embedding image into the cover image, but do not consider how to compress the cover image. There are some other image hiding methods (Chen and Wornell, 1999; Eggers et al., 2000; Kundur, 2000; Moulin et al., 2000; Petitcolas et al., 1999; Wang et al, 2001) that have been developed, but most of them also do not consider the compression of cover images.

In this paper, we present a new singular value decomposition (SVD)- and VQ-based image hiding scheme. Plugging the VQ technique into the SVD-based compression method, the proposed hiding scheme leads to good compression ratio and satisfactory image quality in the cover image and the embedding image. Experimental results show that the embedding image is visually indistinguishable from the stego-image. To the best of our knowledge, this is the first time that such a SVD- and VQ-based image hiding scheme with good compression capability is presented.

The rest of this paper is organized as follows. In Section 2, the background of SVD and VQ are described, respectively. Section 3 introduces the SVD- and VQ-based compression scheme. The

* Corresponding author. Tel.: +886-2-27376771; fax: +886-2-27376777.

E-mail address: klchung@cs.ntust.edu.tw (K.-L. Chung).

0167-8655/01/$ - see front matter © 2001 Published by Elsevier Science B.V.
PII: S0167-8655(01)00044-7

proposed image hiding scheme is presented in Section 4. In Section 5, some experimental results are illustrated to demonstrate the effect of the proposed image hiding scheme. Finally, some conclusions are addressed in Section 6.

2. Preliminary

In this section, we first describe the background of SVD (Golub and Loan, 1989; Leon, 1998) and the application to image compression (Andrews and Patterson, 1976; Knockaert et al., 1999). Then, the background of VQ (Gersho and Gray, 1992; Gray and Neuhoff, 1998; Linde et al., 1980) is described.

2.1. SVD

Suppose an image matrix A is an N by N matrix with rank $= r$, $r \leqslant N$. The matrix A is given by

$$A = UDV^{\mathrm{T}} = [u_1, u_2, \ldots, u_N] \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \sigma_N \end{bmatrix} \times [v_1, v_2, \ldots, v_N]^{\mathrm{T}} = \sum_{i=1}^{r} \sigma_i u_i v_i^{\mathrm{T}}, \quad (1)$$

where U and V are N by N real unitary matrices whose column vectors are u_i's and v_i's, respectively. Here, D is an N by N diagonal matrix with entry σ_i's (singular values) satisfying

$$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > \sigma_{r+1} = \cdots = \sigma_N = 0.$$

The factorization UDV^{T} is called the singular value decomposition of A. In (Leon, 1998), it has been shown that the rank of A equals the number of nonzero singular values and that the magnitudes of the nonzero singular values provide a measure of how close A is to a matrix of lower rank. From Eq. (1), the SVD components also satisfy $Av_i = \sigma_i u_i$ and $u_i^{\mathrm{T}} A = \sigma_i v_i^{\mathrm{T}}$. Generally, the matrix A has many small singular values. Thus, the matrix A can be approximated by a matrix with much lower rank. Let $k \leqslant r$. Then the approximate matrix of A is denoted by $\hat{A}_k = \sum_{i=1}^{k} \sigma_i u_i v_i^{\mathrm{T}}$ and the error matrix is expressed as $E = A - \hat{A}_k$.

2.2. VQ

VQ is an important technique for low-bit-rate image compression. It can be defined as a mapping function Q from an l-dimensional Euclidean space R^l to a finite subset C of R^l:

$$Q: \ R^l \rightarrow C,$$

where $C = \{c_i \mid i = 1, 2, \ldots, N\}$ is the codebook with size N and each possible mapped codeword $c_i = (c_{i1}, c_{i2}, \ldots, c_{il})$ in $C \subset R^l$ is of l-dimension.

The process of VQ can be divided into three phases: (1) codebook generation, (2) encoding and (3) decoding. Given a large amount of vectors, i.e. blocks, the goal of codebook design is to build up a codebook C which contains the most representative codewords, and then this constructed codebook will be used by both encoder and decoder. The most popular codebook design algorithm was developed by Linde et al. (1980) and is referred to as the LBG algorithm.

In the encoding phase, the encoder first divides the image into many square blocks (or vectors) and each vector is with dimension l $(= \sqrt{l} \times \sqrt{l})$. Next, the encoder wants to design a mapping function Q such that given a vector $x = (x_1, x_2, \ldots, x_l)$, the squared Euclidean distance between x and the mapped vector $Q(x) = c_i$ is smallest:

$$d^2(x, c_i) = \min_j \sum_{t=1}^{l} (x_t - c_{jt})^2.$$

That is, the corresponding distortion is minimal. Then, the vector x is replaced by the index i of c_i. Since the number of bits used for representing the index is always smaller than that of the vector x, the encoded image is thus compressed. In the decoding phase, the decoder has the same codebook as the encoder. The decoder has index i as input and merely performs a simple table lookup operation to obtain the decoded codeword c_i and then uses c_i to reconstruct the input vector x approximately. This low complexity decompression in VQ has good computational advantage in comparison with the other compression techniques requiring extra-computation in the decoding phase.

3. SVD- and VQ-based image compression

Although the SVD can find all the singular values and vectors, it requires some amount of computational complexity. In (Yang and Lu, 1995), the iterative estimation procedure for finding the partial SVD, which requires less computational complexity, is given by:

$$\hat{\sigma}_i = u_i^{\mathrm{T}} A_i v_i \quad \text{for } i = 1, 2, \ldots, k, \quad \text{and } A_1 = A,$$
$$A_{i+1} = A_i - \hat{\sigma}_i u_i v_i^{\mathrm{T}} \quad \text{for } i = 1, 2, \ldots, k-1, \tag{2}$$

where A_i is called the residual matrix after removing the first $(i-1)$ principal SVD components, i.e., $\hat{\sigma}_j$, u_j and v_j for $1 \leqslant j \leqslant i-1$. In the above procedure, how to calculate v_i's, u_i's and $\hat{\sigma}_i$'s will be introduced in Step 3, Step 4, and Step 5, respectively, as shown below.

In order to increase the compression ratio for storing all the concerning SVD components, Yang and Lu (1995) plug the VQ technique into the SVD compression method and their method is listed below.

Step 1 (*Generating codebook*). Given a sequence of training images, each training image with size $2^{n_1} \times 2^{n_2}$ is first partitioned into $2^{n_1-m} \times 2^{n_2-m}$ blocks, each block with size $2^m \times 2^m$. For each block B_i, $1 \leqslant i \leqslant 2^{n_1-m} \times 2^{n_2-m}$, using the SVD algorithm (Golub and Loan, 1989; Leon, 1998), we obtain its singular vectors u_j^i and v_j^i for $1 \leqslant j \leqslant 2^m$. After obtaining all the singular vectors, u_j^i's and v_j^i's, we apply the codebook design method (Gersho and Gray, 1992; Gray and Neuhoff, 1998; Linde et al., 1980), and then the codebook $C_U()$ ($C_V()$) with a specific size, say 2^s for u_j^i (v_j^i), $1 \leqslant i \leqslant 2^{n_1-m} \times 2^{n_2-m}$ and $1 \leqslant j \leqslant 2^m$ is obtained. The two codebooks $C_U()$ and $C_V()$ will be used later to obtain the approximated singular vector in terms of index.

Step 2 (*Partitioning input image*). Given an input image with size $2^{n_1} \times 2^{n_2}$ to be encoded, the image is first partitioned into $2^{n_1-m} \times 2^{n_2-m}$ blocks. Each block B_i, $1 \leqslant i \leqslant 2^{n_1-m} \times 2^{n_2-m}$ is of size $2^m \times 2^m$. For each partitioned block B_i, we perform Steps 3–5 k times. Let $j = 1$.

Step 3 (*Encoding* v_j). Using the compatible property of the matrix norm (Golub and Loan, 1989; Leon, 1998), for any codeword $C_V(p)$, $1 \leqslant p \leqslant 2^s$, in the codebook C_V, we compute

$$\max_{1 \leqslant p \leqslant 2^s} \|A_j C_V(p)\|_2 \leqslant \|A_j\|_2 \|C_V(p)\|_2 = \|A_j\|_2 = \sigma_j,$$

where A_j denotes the residual matrix after removing the first $(j-1)$ principal SVD components of each block B_i. The found index p is denoted by p^j which is the representative of the closest singular vector of v_j.

Step 4 (*Encoding* u_j). From the obtained closest vector $C_V(p^j)$ for any codeword $C_U(q)$, $1 \leqslant q \leqslant 2^s$, in the codebook C_U, we compute

$$\max_{1 \leqslant q \leqslant 2^s} C_U^{\mathrm{T}}(q) A_j C_V(p^j) \leqslant \|C_U^{\mathrm{T}}(q)\|_2 \|A_j C_V(p^j)\|_2$$
$$= \|A_j C_V(p^j)\|_2 \leqslant \sigma_j.$$

Equivalently, the found index q is denoted by q^j which is the representative of the closest singular vector of u_j.

Step 5 (*Calculating closest* σ_j). By Eq. (2), from the encoded v_j and u_j, the singular value $\hat{\sigma}_j$ $(= u_j^{\mathrm{T}} A_j v_j)$ is calculated. If $j < k$, then $j = j + 1$ and go to Step 3; otherwise we stop.

For each block, performing Steps 3–5 k times for calculating the k principal SVD components of that block is called the partial SVD procedure.

4. The proposed image hiding scheme

Suppose the cover image, say CI and the embedding image, say EI, are of size $2^{n_1} \times 2^{n_2}$. CI (EI) is first partitioned into $2^{n_1-m} \times 2^{n_2-m}$ blocks too. The partitioned CI (EI) can be represented by

$$CI = \{CI_i,\ 1 \leqslant i \leqslant 2^{n_1-m} \times 2^{n_2-m}\}$$
$$(EI = \{EI_i,\ 1 \leqslant i \leqslant 2^{n_1-m} \times 2^{n_2-m}\}),$$

where CI_i (EI_i) denotes the ith block with size $2^m \times 2^m$ in CI (EI). Our proposed image hiding scheme is shown below:

$$CI_i = \begin{bmatrix} 162 & 162 & 159 & 157 & 159 & 163 & 157 & 161 \\ 162 & 162 & 163 & 159 & 160 & 162 & 159 & 161 \\ 162 & 162 & 159 & 157 & 159 & 163 & 157 & 161 \\ 162 & 162 & 163 & 159 & 160 & 162 & 159 & 161 \\ 162 & 162 & 159 & 157 & 159 & 163 & 157 & 161 \\ 162 & 161 & 158 & 158 & 163 & 162 & 155 & 158 \\ 162 & 162 & 159 & 157 & 159 & 163 & 157 & 161 \\ 161 & 160 & 155 & 156 & 158 & 161 & 157 & 159 \end{bmatrix} = \underbrace{\begin{bmatrix} -.355 & -.271 & .015 & .011 & .894 & .001 & -.002 & .002 \\ -.356 & .127 & -.332 & -.184 & -.844 & 0 & 0 & 0 \\ -.355 & -.271 & .015 & .011 & -.222 & -.480 & -.131 & -.710 \\ -.356 & .127 & -.332 & -.184 & .340 & -.711 & .101 & .284 \\ -.355 & -.271 & .015 & .011 & -.222 & .345 & .794 & -.029 \\ -.354 & -.145 & .646 & -.512 & -.015 & .053 & -.287 & .298 \\ -.355 & -.271 & .015 & .011 & -.226 & .633 & -.589 & .033 \\ -.351 & -.296 & .282 & .764 & -.174 & -.278 & -.113 & .070 \end{bmatrix}}_{U}$$

$$\underbrace{\begin{bmatrix} 1283.3 & 0 & & \cdot & \cdot & \cdot & & 0 \\ 0 & 9.0 & 0 & & & & & \\ & 0 & 6.3 & 0 & & & & \cdot \\ \cdot & & 0 & 3.6 & 0 & & & \cdot \\ \cdot & & & 0 & 0 & 0 & & \cdot \\ \cdot & & & & 0 & 0 & 0 & \\ & & & & & 0 & 0 & 0 \\ 0 & & \cdot & \cdot & \cdot & & 0 & 0 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} -.355 & -.355 & -.352 & -.348 & -.352 & -.355 & -.347 & -.352 \\ -.271 & -.015 & .405 & .808 & .491 & -.416 & .478 & -.181 \\ .015 & -.101 & -.634 & .248 & .748 & .053 & -.191 & -.456 \\ .011 & -.091 & -.557 & .210 & -.273 & -.126 & .784 & .136 \\ -.225 & .207 & 0 & .126 & 0 & .094 & 0 & .271 \\ -.500 & -.031 & 0 & .234 & 0 & .167 & 0 & .575 \\ -.072 & -.449 & 0 & .223 & 0 & .766 & 0 & -.240 \\ .703 & -.781 & 0 & -.018 & 0 & -.244 & 0 & .394 \end{bmatrix}}_{V^T}$$

(a)

$$EI_i = \begin{bmatrix} 131 & 131 & 130 & 131 & 129 & 129 & 130 & 131 \\ 131 & 131 & 129 & 132 & 133 & 130 & 137 & 129 \\ 131 & 131 & 130 & 131 & 129 & 129 & 130 & 131 \\ 127 & 128 & 127 & 130 & 131 & 127 & 131 & 126 \\ 131 & 131 & 130 & 131 & 129 & 129 & 130 & 131 \\ 131 & 131 & 128 & 131 & 129 & 129 & 131 & 131 \\ 131 & 131 & 130 & 131 & 129 & 129 & 130 & 131 \\ 131 & 131 & 131 & 129 & 126 & 130 & 131 & 132 \end{bmatrix} = \underbrace{\begin{bmatrix} -.354 & -.187 & -.048 & .194 & .470 & .761 & .001 & 0 \\ -.358 & .566 & -.449 & .056 & -.589 & 0 & 0 & 0 \\ -.354 & -.187 & -.048 & .194 & .600 & -.633 & .199 & -.004 \\ -.348 & .511 & .133 & -.332 & .570 & .200 & .302 & -.183 \\ -.354 & -.187 & -.048 & .194 & -.198 & -.139 & -.850 & -.135 \\ -.354 & -.090 & -.246 & .808 & .392 & 0 & 0 & 0 \\ -.354 & -.187 & -.048 & .194 & -.410 & -.010 & .211 & .767 \\ -.354 & -.402 & -.481 & -.422 & .156 & .107 & -.030 & .517 \end{bmatrix}}_{U}$$

$$\underbrace{\begin{bmatrix} 1041.8 & 0 & & \cdot & \cdot & \cdot & & 0 \\ 0 & 8.8 & 0 & & & & & \\ & 0 & 5.5 & 0 & & & & \cdot \\ \cdot & & 0 & 2.4 & 0 & & & \cdot \\ \cdot & & & 0 & 0 & 0 & & \cdot \\ \cdot & & & & 0 & 0 & 0 & \\ & & & & & 0 & 0 & 0 \\ 0 & & \cdot & \cdot & \cdot & & 0 & 0 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} -.354 & -.355 & -.352 & -.356 & -.353 & -.350 & -.355 & -.354 \\ -.187 & -.112 & -.285 & .103 & .629 & -.115 & .590 & -.460 \\ -.048 & .262 & .249 & .633 & .680 & .012 & -.681 & .139 \\ .194 & -.042 & -.856 & .124 & .134 & -.198 & -.249 & -.002 \\ -.461 & -.095 & 0 & -.156 & 0 & .074 & 0 & -.334 \\ .021 & .509 & 0 & -.191 & 0 & -.792 & 0 & .161 \\ .439 & -.683 & 0 & .068 & 0 & -.259 & 0 & .608 \\ -.628 & -.238 & 0 & .617 & 0 & -.353 & 0 & -.370 \end{bmatrix}}_{V^T}$$

(b)

$$\underbrace{\begin{bmatrix} -.355 & -.271 & .015 & -.354 & -.187 & -.048 & 0 & 0 \\ -.356 & .127 & -.332 & -.358 & .566 & -.449 & 0 & 0 \\ -.355 & -.271 & .015 & -.354 & -.187 & -.048 & 0 & 0 \\ -.356 & .127 & -.332 & -.348 & .511 & .133 & 0 & 0 \\ -.355 & -.271 & .015 & -.354 & -.187 & -.048 & 0 & 0 \\ -.354 & -.145 & .646 & -.354 & -.090 & -.246 & 0 & 0 \\ -.355 & -.271 & .015 & -.354 & -.187 & -.048 & 0 & 0 \\ -.351 & -.296 & .282 & -.354 & -.402 & -.481 & 0 & 0 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 1283 & 0 & & \cdot & \cdot & \cdot & & 0 \\ 0 & 9 & 0 & & & & & \\ & 0 & 6 & 0 & & & & \cdot \\ \cdot & & 0 & 4.1875 & 0 & & & \cdot \\ \cdot & & & 0 & .1875 & 0 & & \cdot \\ \cdot & & & & 0 & .125 & 0 & \\ & & & & & 0 & 0 & 0 \\ 0 & & \cdot & \cdot & \cdot & & 0 & 0 \end{bmatrix}}_{D}$$

$$\underbrace{\begin{bmatrix} -.355 & -.355 & -.352 & -.348 & -.352 & -.355 & -.347 & -.352 \\ -.271 & -.015 & .405 & .808 & .491 & -.416 & .478 & -.181 \\ .015 & -.101 & -.634 & .248 & .748 & .053 & -.191 & -.456 \\ -.354 & -.355 & -.352 & -.356 & -.353 & -.350 & -.355 & -.354 \\ -.187 & -.112 & -.285 & .103 & .629 & -.115 & .590 & -.460 \\ -.048 & .262 & .249 & .633 & .680 & .012 & -.681 & .139 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{V^T} = \begin{bmatrix} 163 & 162 & 160 & 157 & 160 & 163 & 157 & 161 \\ 162 & 163 & 163 & 160 & 160 & 162 & 160 & 162 \\ 163 & 162 & 160 & 157 & 160 & 163 & 157 & 161 \\ 162 & 163 & 163 & 160 & 160 & 162 & 160 & 162 \\ 163 & 162 & 160 & 157 & 160 & 163 & 157 & 161 \\ 162 & 161 & 157 & 158 & 163 & 163 & 157 & 159 \\ 163 & 162 & 160 & 157 & 160 & 163 & 157 & 161 \\ 161 & 160 & 157 & 155 & 159 & 162 & 155 & 159 \end{bmatrix} = S$$

(c)

$$E_1 = \begin{bmatrix} -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & -1 & -1 \\ -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & -1 & -1 \\ -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & -2 & -1 \\ -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -2 & 1 & -1 & -1 & 2 & 0 \end{bmatrix} \qquad E_2 = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 0 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 0 & -1 & -1 & 0 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 0 & -1 & -1 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

(d)

Fig. 1. A hiding example. (a) The SVD of cover image; (b) the SVD of embedding image; (c) the stego-image; (d) the error matrices for $E1 = CI_i - S$ and $E_2 = EI_i - \hat{EI}_i$.

Step 1 (*Applying partial SVD procedure*). For each block in *CI* or *EI*, applying the partial SVD procedure, the first k principal SVD components, i.e., $\hat{\sigma}_j$, q^j and p^j for $1 \leqslant j \leqslant k$ and $k \leqslant 2^m/2$ are obtained.

Step 2. (*Embedding*). Based on the row-major manner, the k SVD components, i.e., $\widehat{\sigma_j}$, u_j and v_j for $k+1 \leqslant j \leqslant 2k$ of block CI_i in *CI* are replaced by the first k principal SVD components of block EI_i in *EI*. Note that the number of the SVD components of each block in the stego-image is $2k$. Since each block is of size $2^m \times 2^m$, the inequality $2k \leqslant 2^m$ should be held and it explains why $k \leqslant 2^m/2$. In addition, the remaining SVD components of the block CI_i in *CI*, i.e., $\hat{\sigma}_j$, u_j and v_j for $2k+1 \leqslant j \leqslant 2^m$ are set to zeros and zero vectors, respectively.

Step 3 (*Quantization*). In the embedded CI_i, the newly constructed 2^m singular values along the main diagonal may not be in a decreasing manner. Later, those k embedded singular values in CI_i are quantized by the quantization sequence, $\beta_1, \beta_2, \ldots, \beta_k$, respectively. That is, we perform $\hat{\sigma}_{k+j}/\beta_j$, $1 \leqslant j \leqslant k$. According to our experiments, β_1 is selected by 250 and the other quantized factors are selected by 50. This quantization adaptation can make those 2^m singular values along the main diagonal decreasing.

Step 4 (*Encoding singular values*). Since the first k original singular values in the CI_i are some large, among them, each one is encoded by the closest integer. However, the next k quantized singular

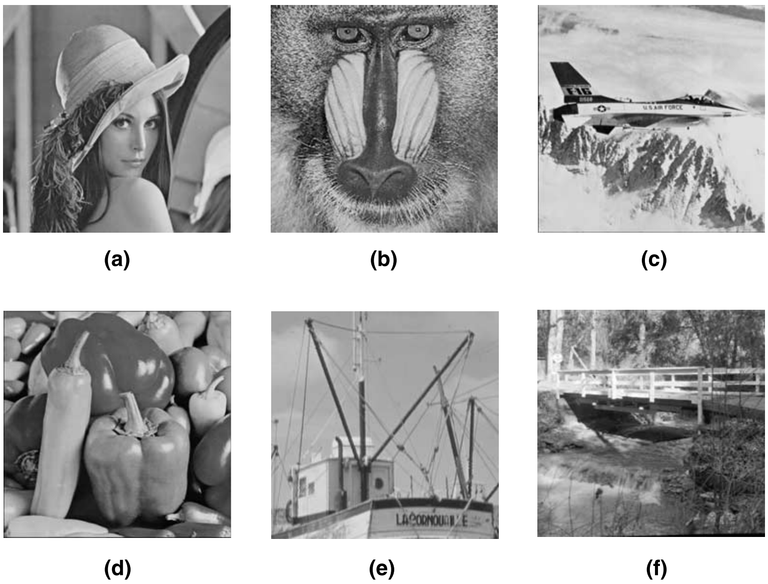

Fig. 2. The six original images: (a) Lena; (b) Baboon; (c) F16; (d) Pepper; (e) Boat; (f) Bridge

Table 1
PSNR of Lena with and without hidden image data

	The size of the VQ codebook				
	64	128	256	512	1024
Lena	33.423	34.266	35.326	36.251	37.002
Lena (Baboon)	30.807	31.415	32.535	33.371	34.013
Lena (F16)	30.784	31.391	32.501	33.330	33.965
Lena (Pepper)	30.814	31.425	32.550	33.388	34.036
Lena (Boat)	30.800	31.410	32.525	33.145	34.000
Lena (Bridge)	30.816	31.427	32.553	33.172	34.038

values are some small, the integer part of each quantized singular value is retained, but its decimal fraction is represented by four bits approximately. Consequently, the encoded CI_i is the so called stego-image.

The above proposed image hiding scheme can hide an image easily and the hidden image, i.e., the embedding image, can be extracted easily in a reverse manner.

We now take an example to illustrate how the above proposed image hiding scheme works. For exposition, the demonstration does not include the VQ technique (Yang and Lu, 1995). However, in the experiments as shown in Section 5, the VQ technique (Yang and Lu, 1995) will be covered. The cover (embedding) image CI_i (EI_i), which is obtained from one block in Lena (Pepper) image, with size $2^3 \times 2^3$ and the SVD of CI_i (EI_i) are shown in Fig. 1(a) (Fig. 1(b)). Here, let $k = 3$. Then the first three principal SVD components of CI_i and EI_i are retained and the stego-image S combining CI_i and EI_i is shown in Fig. 1(c). The error matrices E_1 $(= CI_i - S)$ and E_2 $(= EI_i - \hat{E}I_i)$ are shown in Fig. 1(d), where $\hat{E}I_i$ denotes the extracted embedding image from the stego-image S. It is observed that each entry in E_1 or E_2 is infinitesimal. It implies that the embedding image is visually indistinguishable from the stego-image and it achieves the image hiding effect.

Fig. 3. The original Lena image and five stego-images: (a) original image; (b) Lena embedded by Baboon; (c) Lena embedded by F16; (d) Lena embedded by Pepper; (e) Lena embedded by Boat; (f) Lena embedded by Bridge.

5. Experimental results

In this section, some experiments are carried out to demonstrate the effect of the proposed image hiding scheme. The 512×512 Lena image is selected as the cover image and shown in Fig. 2(a), and the 512×512 Baboon, F16, Pepper, Boat, and Bridge images are selected as the five embedding images and shown in Figs. 2(b)–(f), respectively. Each block is of size 8×8 and the three principal

Table 2
PSNR for the extracted embedding image with different VQ codebook size

	The size of the VQ codebook				
	64	128	256	512	1024
Baboon	27.945	28.002	28.089	29.080	29.987
F16	29.587	29.760	30.011	31.319	33.880
Pepper	28.435	28.829	30.138	31.655	33.981
Boat	32.477	32.777	33.783	34.679	35.554
Bridge	30.170	30.756	31.292	32.059	32.970

Table 3
The bpp required for different VQ codebook size

	The size of the VQ codebook				
	64	128	256	512	1024
bpp	1.922	2.109	2.297	2.484	2.672

SVD components of each block are calculated. The VQ codebooks with different size, namely, 64, 128, 256, 512, and 1024, are created by using LBG algorithm (Linde et al., 1980) from the training images, Baboon, F16, Pepper, and Lena. Peak signal-to-noise ratio (PSNR) is used to measure the effect of the proposed scheme.

In the first row of Table 1, the PSNR denotes the image quality of the compressed Lena when applying the SVD- and VQ-based compression method (Yang and Lu, 1995) as described in Section 3. In the second, third, fourth, fifth and sixth row, the PSNR denotes the image quality of the stego-image Lena when embedding the original Lena by Baboon, F16, Pepper, Boat and Bridge, respectively. It is observed that the change of the PSNR after hiding the cover image is some small. On the other hand, each embedding image is visually indistinguishable from the stego-image and the image hiding effect is quite good (See Figs. 3(b)–(f) for 1024 codebook size).

Table 2 shows the image quality, i.e., PSNR, for the extracted embedding image with different VQ codebook size. From Table 2, it is observed that the extracted embedding image can preserve the satisfactory image quality. Finally, with different VQ codebook size, the compression ratios are shown in Table 3. Each entry in Table 3 denotes how many bits required for one pixel, i.e., bpp, of the stego-image. As can be seen from Tables 1 and 3, it is observed that the proposed image hiding scheme can achieve good compression ratio and satisfactory image quality of the stego-image. Finally, the larger the codebook size is, the better the value of PSNR is and the worse the compression ratio is.

6. Conclusion

A new SVD- and VQ-based image hiding scheme to hide image data has been presented in this paper. The combination of the SVD compression method and the VQ technique leads to good compression ratio and satisfactory image quality. Experimental results show that the embedding image is visually indistinguishable from the stego-image.

References

Anderson, R., 1996. Strentching the limits of steganography. In: Proc. First Internat. Workshop Information Hiding, Cambridge, UK. Lecture Notes, Vol. 1174. Springer, Berlin, pp. 39–48.

Andrews, H.C., Patterson, C.L., 1976. Singular value decomposition (SVD) image coding. IEEE Trans. Comm. 24 (4), 425–432.

Chen, B., Wornell, G.W., 1999. Achievable performance digital watermarking systems. Proc. IEEE Internat. Conf. Multimedia Comput. Systems 1, 13–18.

Chen, T.S., Chang, C.C., Hwang, M.S., 1998. A virtual image cryptosystem based upon vector quantization. IEEE Trans. Image Process. 7 (10), 1485–1488.

Eggers, J., Su, J., Girod, B., 2000. Robustness of a blind image watermarking scheme. In: Proc. IEEE Internat. Conf. Image Process. Vancouver, Canada.

Gersho, A., Gray, R.M., 1992. Vector Quantization Signal and Compression. Kluwer Academic Publisher, Boston.

Golub, G.H., Loan, C.F.V., 1989. Matrix Computations. Johns Hopkins University Press, Baltimore, MD (Chapter 8).

Gray, R.M., Neuhoff, D.L., 1998. Quantization. IEEE Trans. Information Theory 44 (6), 2325–2383.

Johnson, N.F., Jajodia, S., 1998. Exploring steganography: seeing the unseen. IEEE Comput. Magazine 31 (2), 26–34.

Knockaert, L., Backer, B.D., Zutter, D.D., 1999. SVD compression, unitary transforms, and computational complexity. IEEE Trans. Signal Process. 47 (10), 2724–2729.

Kundur, D., 2000. Energy allocation for high-capacity watermarking in the presence of compression. Proc. IEEE Internat. Conf. Image Process. Vancouver, Canada.

Leon, S.J., 1998. Linear Algebra with Applications. Prentice-Hall, New Jersey (Chapter 7).

Linde, Y., Buzo, A., Gray, R.M., 1980. An algorithm for vector quantizer design. IEEE Trans. Comm. 28 (1), 84–95.

Moulin, P., Mihcak, M., Lin, G.I., 2000. An information-theoretic model for image watermarking and data hiding.

Proc. IEEE Internat. Conf. Image Process. Vancouver, Canada.

Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G., 1999. Information hiding – a survey. Proc. IEEE 87 (7), 1062–1078.

Wang, R.Z., Lin, C.F., Lin, J.C., 2001. Image hiding by optimal LSB substitution and genetic algorithm. Pattern Recognition 34 (3), 671–683.

Wu, D.C., Tsai, W.H., 1998. Data hiding in images via multiple-based number conversion and lossy compression. IEEE Trans. Consumer Electron. 44 (4), 1406–1412.

Yang, J.F., Lu, C.L., 1995. Combined techniques of singular value decomposition and vector quantization for image coding. IEEE Trans. Image Process. 4 (8), 1141–1146.