Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Engineering

Elixir Comp. Engg. 116 (2018) 50124-50132

ISSN: 2229-712X

A Novel Joint Data-Hiding and Compression Scheme Based on SMVQ and Image inpainting

Chennakesavarao.M, M.Tejaswini, M. Kumar Chandra, U.Nagamani and M.Mounika
Department of Computer Science Engineering, Tirumala Engineering College.

ARTICLE INFO

Article history:
Received: 12 February 2018;
Received in revised form:
8 March 2018;
Accepted: 19 March 2018;

Keywords
Data Hiding,
Image Compression,
Side Match Vector Quantization (SMVQ),
Image Inpainting.

ABSTRACT
In this paper, we propose a novel joint data-hiding and compression scheme for digital images using side match vector quantization (SMVQ) and image inpainting. The two functions of data hiding and image compression can be integrated into one single module seamlessly. On the sender side, except for the blocks in the leftmost and topmost of the image, each of the other residual blocks in raster-scanning order can be embedded with secret data and compressed simultaneously by SMVQ or image inpainting adaptively according to the current embedding bit. Vector quantization is also utilized for some complex blocks to control the visual distortion and error diffusion caused by the progressive compression. After segmenting the image compressed codes into a series of sections by the indicator bits, the receiver can achieve the extraction of secret bits and image decompression successfully according to the index values in the segmented sections. Experimental results demonstrate the effectiveness of the proposed scheme.

I. Introduction

With the rapid development of Internet technology, people can transmit and share digital content with each other conveniently. In order to guarantee communication efficiency and save network bandwidth, compression techniques can be implemented on digital content to reduce redundancy, and the quality of the decompressed versions should also be preserved. Nowadays, most digital content, especially digital images and videos, are converted into the compressed forms for transmission [1]–[4]. Another important issue in an open network environment is how to transmit secret or private data securely. Even though traditional cryptographic methods can encrypt the plaintext into the ciphertext [5], [6], the meaningless random data of the ciphertext may also arouse the suspicion from the attacker. To solve this problem, information hiding techniques have been widely developed in both academia and industry, which can embed secret data into the cover data imperceptibly [7], [8]. Due to the prevalence of digital images on the Internet, how to compress images and hide secret data into the compressed images efficiently deserves in-depth study.

Recently, many data-hiding schemes for the compressed codes have been reported, which can be applied to various compression techniques of digital images, such as JPEG [9], [10], JPEG2000 [11], and vector quantization (VQ) [12]–[15]. As one of the most popular lossy data compression algorithms, VQ is widely used for digital image compression due to its simplicity and cost effectiveness in implementation [16], [17]. During the VQ compression process, the Euclidean distance is utilized to evaluate the similarity between each image block and the codewords in the codebook. The index of the codeword with the smallest distance is recorded to represent the block. Thus, an index table consisting of the index values for all the blocks is generated as the VQ compression codes.

Instead of pixel values, only the index values are stored, therefore, the compression is achieved effectively. The VQ decompression process can be implemented easily and efficiently because only a simple table lookup operation is required for each received index. In this work, we mainly focus on the data embedding in VQ-related image compressed codes.

In 2003, Du and Hsu proposed an adaptive data hiding method for VQ compressed images [18], which can vary the embedding process according to the amount of hidden data. In this method, the VQ codebook was partitioned into two or more subcodebooks, and the best match in one of the sub-codebooks was found to hide secret data. In order to increase the embedding capacity, a VQ-based data-hiding scheme by a codeword clustering technique was proposed in [19].The secret data were embedded into the VQ index table by codeword-order-cycle permutation. By the cycle technique, more possibilities and flexibility can be offered to improve the performance of this scheme. Inspired by [18], [19], Lin *et al.* adjusted the pre-determined distance threshold according to the required hiding capacity and arranged a number of similar codewords in one group to embed the secret sub-message [20]. The search-order coding (SOC) algorithm was proposed by Hsieh and Tsai, which can be utilized to further compress the VQ index table and achieve better performance of the bit rate through searching nearby identical image blocks following a spiral path [21]. Some steganographic schemes were also proposed to embed secret data into SOC compressed codes [22]–[24].

Side match vector quantization (SMVQ) was designed as an improved version of VQ [25], in which both the codebook and the subcodebooks are used to generate the index values, excluding the blocks in the leftmost column and the topmost row. Recently, many researchers have studied on embedding secret message by SMVQ [26]–[31].

Tele:
E-mail address: chenna1207@gmail.com

In 2010, Chen and Chang proposed an SMVQ-based secret-hiding scheme using adaptive index [27]. The weighted squared Euclidean distance (WSED) was utilized to increase the probability of SMVQ for a high embedding rate. In order to make the secret data imperceptible to the interceptors, Shie and Jiang hided secret data into the SMVQ compressed codes of the image by using a partially sorted codebook [29]. The restoration of the original SMVQ-compressed image can be achieved at the receiver side.

However, in all of the above mentioned schemes, data hiding is always conducted after image compression, which means the image compression process and the data hiding process are two independent modules on the server or sender side. Under this circumstance, the attacker may have the opportunity to intercept the compressed image without the watermark information embedded, and the two independent modules may cause a lower efficiency in applications. Thus, in this work, we not only focus on the high hiding capacity and recovery quality, but also establish a joint data-hiding and compression (JDHC) concept and integrate the data hiding and the image compression into a single module seamlessly, which can avoid the risk of the attack from interceptors and increase the implementation efficiency. The proposed JDHC scheme in this paper is based on SMVQ and image inpainting. On the sender side, except for the blocks in the leftmost and topmost of the image, each of the other residual blocks in raster-scanning order can be embedded with secret data and compressed simultaneously by SMVQ or image inpainting adaptively according to the current embedding bit. VQ is also utilized for some complex residual blocks to control the visual distortion and error diffusion caused by the progressive compression. After receiving the compressed codes, the receiver can segment the compressed codes into a series of sections by the indicator bits. According to the index values in the segmented sections, the embedded secret bits can be extracted correctly, and the decompression for each block can be achieved successfully.

The rest of the paper is organized as follows. Section II describes the proposed joint data-hiding and compression scheme, including the procedure of image compression and secret data embedding and the procedure of image decompression and secret data extraction. Experimental results and analysis are given in Section III, and Section IV concludes the paper.

II. Joint Data-Hiding And Compression Scheme

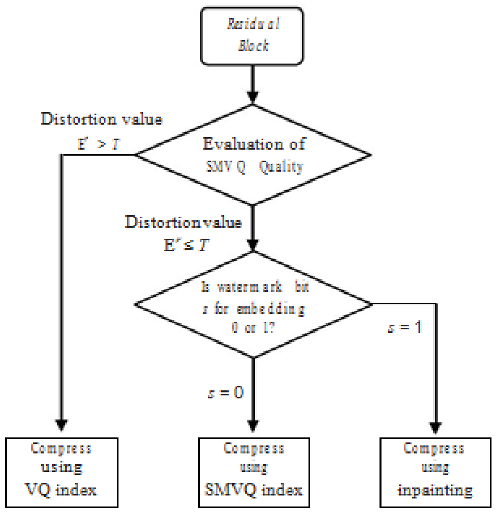

Residual Block
Distortion value
$E' > T$
Evaluation of SMVQ Quality
Distortion value
$E' \leq T$
Is watermark bit s for embedding 0 or 1?
$s = 1$
$s = 0$
Compress using VQ index
Compress using SMVQ index
Compress using inpainting

Fig. 1. Flowchart of compression and secret data embedding for each residual block.

In the proposed scheme, rather than two separate modules, only a single module is used to realize the two functions, i.e., image compression and secret data embedding, simultaneously. The image compression in our JDHC scheme is based mainly on the SMVQ mechanism. According to the secret bits for embedding, the image compression based on SMVQ is adjusted adaptively by incorporating the image inpainting technique. After receiving the secret embedded and compressed codes of the image, one can extract the embedded secret bits successfully during the image decompression.

A. Image Compression and Secret Data Embedding

As an extension of VQ, SMVQ was developed to alleviate the block artifact of the decompressed image and increase the compression ratio, because the correlation of neighboring blocks is considered and the indices of the subcodebooks are stored. In our scheme, the standard algorithm of SMVQ is modified to further achieve better decompression quality and to make it suitable for embedding secret bits. The detailed procedure is described as follows.

In our scheme, the sender and the receiver both have the same codebook with W code words, and each code word length is n^2. Denote the original uncompressed image sized $M \times N$ as $\mathbf{I}$, and it is divided into the non-overlapping $n \times n$ blocks. For simplicity, we assume that M and N can be divided by n with no remainder. Denote all k divided blocks in raster-scanning order as $\mathbf{B}_{i,j}$, where $k = M \times N / n^2$, $i = 1, 2, \ldots, M/n$, and $j = 1, 2, \ldots, N/n$. Before being embedded, the secret bits are scrambled by a secret key to ensure security. The blocks in the leftmost and topmost of the image $\mathbf{I}$, i.e., $\mathbf{B}_{i,1}(i = 1, 2, \ldots, M/n)$ and $\mathbf{B}_{1,j}$ $(j = 2, 3, \ldots, N/n)$, are encoded by VQ directly and are not used to embed secret bits. The residual blocks are encoded progressively in raster-scanning order, and their encoded methods are related to the secret bits for embedding and the correlation between their neighboring blocks. The flowchart of the processing for each residual block is illustrated in Figure 1.

Denote the current processing block as $\mathbf{B}_{x,y}$ $(2 \leq x \leq M/n,$ $2 \leq y \leq N/n)$, and its left and up blocks are $\mathbf{B}_{x,y-1}$ and

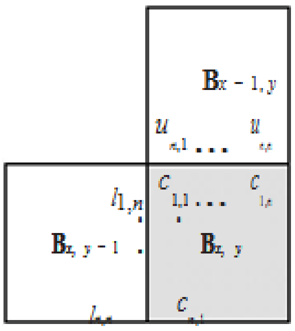


Fig 2. Illustration of the prediction based on left and up neighboring pixels.

$\mathbf{B}_{x-1,y}$, respectively. As shown in Figure 2, $c_{p,1}$ $(1 \leq p \leq n)$ and $c_{1,q}$ $(2 \leq q \leq n)$ represent the $2n - 1$ pixels in the left and upper borders of $\mathbf{B}_{x,y}$. The n pixels in the right border of $\mathbf{B}_{x,y-1}$ and the n pixels in the bottom border of $\mathbf{B}_{x-1,y}$ are denoted as $l_{p,n}$ $(1 \leq p \leq n)$ and $u_{n,q}$ $(1 \leq q \leq n)$, respectively. Similar with SMVQ, the $2n - 1$ pixels in the left and upper borders of $\mathbf{B}_{x,y}$ are predicted by the neighboring pixels in $\mathbf{B}_{x,y-1}$ and $\mathbf{B}_{x-1,y}$: $c_{1,1} = (l_{1,n} + u_{n,1})/2$, $c_{p,1} = l_{p,n}$ $(2 \leq p \leq n)$, and $c_{1,q} = u_{n,q}$ $(2 \leq q \leq n)$. Instead of all n^2 pixels in $\mathbf{B}_{x,y}$, only these $2n - 1$ predicted pixels are used to search the codebook . After transforming all W codewords in the codebook into the $n \times n$ matrices, the mean square error (MSE) E^w is calculated between the $2n - 1$ predicted pixels in $\mathbf{B}_{x,y}$ with the corresponding values of each transformed codeword C^w sized $n \times n$.

$$E^W = \sum_{p=1}^{n} (c_{p,1} - c^W_{p,1})^2 + \sum_{q=2}^{n} (c_{1,q} - c^W_{1,q})^2 , \quad (1)$$

where $c^w_{p,q}$ are the elements of each codeword C^w in code-book . The R codewords with the smallest MSEs, i.e., E^w , are selected to generate one subcodebook $_{x,y}$ for the block $\mathbf{B}_{x,y}$ ($R < W$). Suppose that, among the R codewords in $_{x,y}$, the codeword indexed λ has the smallest MSE, i.e., E^r , with all n^2 pixels in $\mathbf{B}_{x,y}$ ($0 \le \lambda \le R - 1$). If the value of E^r is greater than a pre-determined threshold T for distortion control, it implies that the current residual block $\mathbf{B}_{x,y}$ locates in a relatively complex region and it has lower correlation with its neighboring blocks. Under this circumstance, in order to achieve better decompression quality, the standard, block-independent VQ with codebook is used to compress the block $\mathbf{B}_{x,y}$, and no secret bits are embedded. Otherwise, if $E^r \le T$, it implies that the current residual block $\mathbf{B}_{x,y}$ locates in a relatively smooth region and it has higher correlation with its neighboring blocks. Thus, in this condition, SMVQ or image inpainting is adaptively utilized to compress the block $\mathbf{B}_{x,y}$ according to the secret bit s for embedding, which results in the shorter index length and the success of secret data hiding. Note that, if VQ is adopted, an indicator bit, i.e., 0, should be added before the compressed code of the VQ index for $\mathbf{B}_{x,y}$. If not, the indicator bit, i.e., 1, is added as the prefix of the compressed code for $\mathbf{B}_{x,y}$.

As for the block $\mathbf{B}_{x,y}$, if its E^r is not greater than the threshold T and the current secret bit s for embedding is 0, SMVQ is utilized to conduct compression, which means that the index value λ occupying $\log_2 R$ bits is used to represent the block $\mathbf{B}_{x,y}$ in the compressed code. Because the codeword number R in subcodebook $_{x,y}$ is less than the codeword number W of the original codebook , the length of the compressed code for $\mathbf{B}_{x,y}$ using SMVQ must be shorter than using VQ. On the other hand, if $E^r \le T$ and the current secret bit s for embedding is 1, the image inpainting technique is used.

The concept of image inpainting is inherited from the ancient technique of manually repairing valuable artworks in an undetectable manner [34]. Inpainting for digital images has found applications in such areas as repairing of damaged photographs, filling in or removing chosen areas, and wiping off visible watermarks. Image inpainting can generate or create image regions that initially do not exist at all, based on the useful information in the close neighborhood. Currently, there are mainly three classes of the image inpainting methods, i.e., partial differential equation (PDE) based methods [34]–[36], interpolation-based methods [37], and patch-based methods [38].

The PDE-based inpainting methods often propagate the available information of gray values automatically from surrounding areas into region to be inpainted along a specific direction. There are several mathematical physics models that can be used for PDE-based inpainting, such as the fluid dynamics model [34] and the heat transfer model [35]. Different PDE models correspond to the different methods of information propagation. Image inpainting can recover the image structural information effectively when the processed region is not too large. Evidently, if $E^r \le T$, it implies that $\mathbf{B}_{x,y}$ locates in a relatively smooth region.

Thus, it is suitable to conduct image inpainting in the compression for $\mathbf{B}_{x,y}$ under this condition.

In our scheme, a PDE-based image inpainting method using the fluid dynamics model is adopted [34]. Denote $\mathbf{B}_\chi$ as the region including the current block $\mathbf{B}_{x,y}$ that needs compression by inpainting and the available neighboring region of $\mathbf{B}_{x,y}$. Let $B_\chi(\xi, \eta)$ be the gray value of $\mathbf{B}_\chi$ in the coordinate (ξ , η). The Laplacian $B_\chi(\xi, \eta)$ is used as a smoothness measure of the region $\mathbf{B}_\chi$. By analogizing the inpainting process as the fluid flowing and imitating the practice of a traditional art professional in the manual retouching, details in the unknown region may be created through propagating the available information in the surrounding areas into the unknown region along isophote directions. The field of isophote is defined as:

$$\nabla^{\perp} B_\chi(\xi, \eta) = -\frac{\partial}{\partial \xi}\mathbf{i} + \frac{\partial}{\partial \eta}\mathbf{j}\, B_\chi(\xi, \eta) , \quad (2)$$

where i and j are unit directional vectors. Clearly, variations in image gray values are minimal along the isophote directions. Having finished the inpainting process, $\nabla^{\perp} B_\chi(\xi, \eta)$ should be normal to the gradient of the smoothness $B_\chi(\xi, \eta)$:

$$\nabla [B_\chi(\xi, \eta)] \cdot \nabla^{\perp} B_\chi(\xi, \eta) = 0 . \quad (3)$$

The scalar product in the above equation indicates projection of the smoothness change onto the direction of isophote. If we let the projection value be equal to the change of image gray values with respect to time t , the following PDE can be required

$$\frac{\partial}{\partial t} B_\chi(\xi, \eta) = \nabla [B_\chi(\xi, \eta)] \cdot \nabla^{\perp} B_\chi(\xi, \eta), \quad \forall (\xi, \eta) \in \mathbf{B}_\chi \quad (4)$$

By using the finite difference method, we can obtain a discretized iteration algorithm to solve the PDE in Eq. (4). Information propagation of this inpainting model finishes until the gray values in $\mathbf{B}_{x,y}$ reach stable state. Consequently, the structural and geometric information of the block $\mathbf{B}_{x,y}$ can be recovered effectively without serious blurring on edges.

Consequently, when $s = 1$, in order to indicate that block is processed by inpainting and differentiate from the index λ produced by SMVQ, the index value R occupying $\log_2(R + 1)$ bits is used as the compressed code of $\mathbf{B}_{x,y}$ ($R > \lambda$). For simplicity, we assume that $\log_2(R + 1)$ is an integer and $\log_2 R \equiv \log_2(R + 1)$.

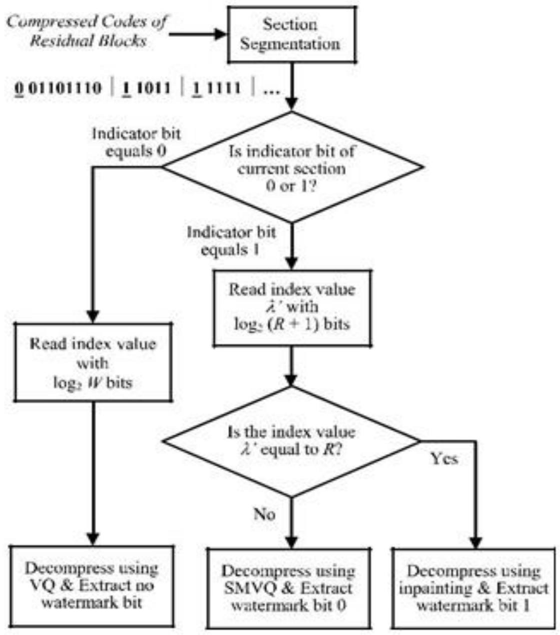
After the current block $\mathbf{B}_{x,y}$ is processed, the following block in raster-scanning order repeats the above procedure. Note that each processed block should be substituted with its corresponding decompressed result, i.e., VQ codeword, SMVQ codeword, or inpainting result, for the success of progressive mechanism. The used image inpainting technique is described in the next subsection detailedly. The whole procedure of image compression and secret data embedding finishes until all residual blocks are processed. Then, the compressed codes of all image blocks are concatenated and transmitted to the receiver side.

B. Image Decompression and Secret Data Extraction

After receiving the compressed codes, the receiver conducts the decompression process to obtain the decoded image that is visually similar to the original uncompressed image, and the embedded secret bits can be extracted either before or during the decompression.

Because the $(M + N - n)/n$ blocks in the leftmost and topmost of the image need to be used in the decompression for other residual blocks, they should be first decompressed by their VQ indices retrieved from the image compressed codes. Each VQ index of these pre-decompressed blocks occupies $\log_2 W$ bits. Then, the $k - (M + N - n)/n$ residual blocks are processed block by block in raster-scanning order. Figure 3 shows the flowchart of decompression and secret bit extraction for each residual block.

To conduct the decompression and secret bit extraction of each residual block, the compressed codes are segmented into a series of sections adaptively according to the indicator bits. Explicitly, if the current indicator bit in the compressed codes is 0, this indicator bit and the following $\log_2 W$ bits are segmented as a section, which means this section corresponds to a VQ compressed block with no embedded secret bit. The decimal value of the last $\log_2 W$ bits in this section is exactly the VQ index that can be used directly to recover the block. Otherwise, if the current indicator bit is 1, this indicator bit and the following $\log_2(R + 1)$ bits are then segmented as a section, which means this section corresponds to an SMVQ or inpainting compressed block. Denote the decimal value of the last $\log_2(R + 1)$ bits in this section as λ'. Under this circumstance, if λ' is equal to R, it implies that the residual block corresponding to this section was compressed by inpainting and that the embedded secret bit in this block is 1. Otherwise, if $\lambda' \in [0, R - 1]$, it implies that the block corresponding to this section was compressed by SMVQ and that the embedded secret bit is 0.



Compressed Codes of Residual Blocks
Section Segmentation
0 01101110 | 1 1011 | 1 1111 | ...
Indicator bit equals 0
Is indicator bit of current section 0 or 1?
Indicator bit equals 1
Read index value λ' with log₂ (R + 1) bits
Read index value with log₂ W bits
Is the index value λ' equal to R?
Yes
No
Decompress using VQ & Extract no watermark bit
Decompress using SMVQ & Extract watermark bit 0
Decompress using inpainting & Extract watermark bit 1

Fig 3. Flowchart of decompression and secret data extraction for each residual block.

If the current segmented section corresponds to an inpainting compressed block Bi np , the available information of its neighboring decompressed blocks are utilized to conduct recovery by the same inpainting technique used in the compression process. If the current segmented section corresponds to an SMVQ compressed block Bs mq , SMVQ index value, i.e., λ', is used to recover this block with the assistance of its left and upper decompressed blocks. Using the same prediction method described in Subsection A, the 2n − 1 pixels in the left and upper borders of Bs mq are estimated by the neighboring pixels in its left and upper decompressed blocks. Similarly, the MSEs are calculated between these 2n − 1 predicted pixels in Bs mq with the corresponding values of all W codewords in the codebook. Then, the R codewords in with the smallest MSEs are chosen to generate a subcodebook. Finally, the codeword indexed λ' in the generated subcodebook is used to recover the block Bsmq.

After all the segmented sections in the compressed codes complete the above described procedure, the embedded secret bits can be extracted correctly, and the decompressed image $\mathbf{I}_d$ can be obtained successfully. Due to the decoding of the compressed codes, the decompressed image $\mathbf{I}_d$ doesn't contain the embedded secret bits any longer. Note that the process of secret bit extraction can also be conducted independently, which means that the receiver can obtain all embedded bits by simply segmenting and analyzing the compressed codes without the decoding.

Fig 4. Six standard test images.

Therefore, besides the image compression, the proposed scheme can achieve the function of data hiding that can be used for covert communication of secret data. The sender can transmit the secret data securely through the image compressed codes, and the receiver can extract the hidden secret data effectively from the received compressed codes to complete the process of covert communication. Additionally, because the secret data extraction in our scheme can be conducted independently with the decompression process, the receiver can obtain the secret bits at any time if he or she preserves the compressed codes. The proposed scheme can also be used for the integrity authentication of the images, in which the secret bits for embedding can be regarded as the hash of the image principle contents. The receiver can calculate the hash of the principle contents for the decompressed image, and then compare this calculated hash with the extracted secret bits (embedded hash) to judge the integrity of the received compressed codes and the corresponding decompressed image. If the two hashes are equal, it means the image is authenticated. Otherwise, the received compressed codes must be tampered.

III. Experimental Results and Analysis

Experiments were conducted on a group of gray-level images to verify the effectiveness of the proposed scheme. In the experiment, the sizes of the divided non-overlapping image blocks were 4 × 4, i.e., $n = 4$. Accordingly, the length of each codeword in the used VQ codebooks was 16. The parameter R was set to 15. Six standard, 512 × 512 test images, i.e., Lena, Airplane, Lake, Peppers, Sailboat, and Tiffany, are shown in Figure 4. Besides these six standard images, the uncompressed color image database (UCID) that contains 1338 various color images with sizes of 512 × 384 was also adopted [39]. The luminance components of the color images in this database were extracted and used in the experiments. The performances of compression ratio, decompression quality, and hiding capacity for the proposed scheme were evaluated.

All experiments were implemented on a computer with a 3.00 GHz AMD Phenom II processor,
2.00 GB memory, and Windows 7 operating system, and the programming environment was Matlab 7.

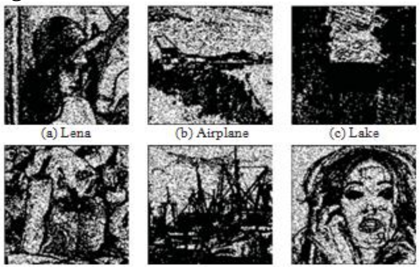


Fig 5. Labels of image blocks with different types (*T* = 16). The black block, gray block, and white block in Figures 5(a)–(f) correspond to the blocks compressed by VQ, SMVQ, and image inpainting, respectively.

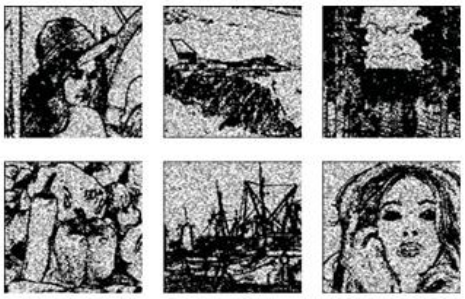


Fig. 6. Labels of image blocks with different types (*T* = 28). The black block, gray block, and white block in Figures 6(a)–(f) correspond to the blocks compressed by VQ, SMVQ, and image inpainting, respectively.

Because the threshold *T* used in the procedure of image compression and secret data embedding is closely related to the compression method for each residual block and also influences on the performance of the proposed scheme (see Figure 1), the testing for different values of *T* was first conducted in the compression and secret embedding procedure. Figures 5 and 6 show the labeling results of image blocks with different compression methods, in which the black block, gray block, and white block correspond to the blocks compressed by VQ, SMVQ, and image inpainting, respectively. The VQ codebook size *W* used in Figures 5 and 6 was 256, and secret bits for embedding were generated by
A pseudo-random number generator (PRNG). Note that the blocks in the topmost row and the leftmost column must be black, i.e., compressed by VQ. It can be obviously found that the number of SMVQ blocks and inpainting blocks increases with *T* , while the number of VQ blocks decreases. As described in Section 2, the blocks compressed by VQ are not used for secret bit embedding, and the secret bits are only embedded in the SMVQ and inpainting blocks. Thus, the hiding capacity of the proposed scheme is equal to the sum of the numbers of SMVQ and inpainting blocks.

Figures 7(a)–(d) show the relationships between hiding capacity and threshold T for the six test images in Figure 4 with codebook sizes *W* = 128, 256, 512, and 1024, respectively. We can observe that, with the same codebook, the hiding capacity increases with the threshold *T* . It can also be seen from Figure 7 that, under the same threshold *T* ,the hiding capacity also increases with the codebook size *W* .

We compared the hiding capacity of the proposed scheme with three typical schemes [31]–[33]. The schemes in [31], [32] and the proposed scheme all conduct the secret data hiding in the image compressed codes, while the scheme in [33] embedded secret data directly in the uncompressed form of the cover image, and its hiding capacity was based on the pixel number corresponding to the two highest peaks of the image histogram. Similar with our scheme, Tsai *et al.*'s scheme [31] was also based on SMVQ, but it isn't a kind of joint data-hiding and compression scheme, which means it embeds secret bits after compression. Qian *et al.*'s scheme [32] utilized the JPEG compressed form of cover image, i.e.,

JPEG bitstream, to embed secret bits. In this experiment, the threshold *T* in the proposed scheme and the JPEG quality factor in [32] were set to 36 and 70, respectively. As we described before, the hiding capacity of the proposed scheme is equal to the number of residual blocks that satisfy the evaluation condition of SMVQ quality, i.e., distortion value $E^r \leq T$. Table I gives the comparison results of the hiding capacity for the six standard images in Figure 4 and the 1338 images in UCID database. The last three rows of Table I are the minimums, maximums, and mean values of the hiding capacities for the images in UCID database, respectively. We can find from the results that the proposed scheme has greater hiding capacity than the two schemes in the compressed domain, i.e., Tsai *et al.*'s scheme [31] and Qian *et al.*'s scheme [32], and is comparable to the scheme in [33] that embedded secret information by histogram shifting in the uncompressed domain. The comparison results also demonstrate that the image compressed codes based on SMVQ can be used to carry more secret bits than the JPEG compressed codes.

We also compared the compression ratio and decompression quality of the proposed scheme with some VQ/SMVQ related schemes, i.e., the standard VQ method, SMVQ method, Chen *et al.*'s scheme [27], and Tsai *et al.*'s scheme [31]. Denote the length of the compressed codes for the image as L_c . The compression ratio C_R can be calculated according to Eq. (5).

Peak signal-to-noise ratio (PSNR) was utilized to measure the visual quality of the decompressed images $\mathbf{I}_d$, see Eq. (6).

$$C_R = \frac{8 \times M \times N}{L_c} \qquad (5)$$

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2 \times M \times N}{\sum_{x=1}^{M} \sum_{y=1}^{N} [I(x,y) - I_d(x,y)]^2} \qquad (6)$$

where *M* and *N* are the height and the width of the images, respectively; $I(x,y)$ and $I_d(x,y)$ are the pixel values at coordinate (x,y) of the original uncompressed image **I** and the decompressed image $\mathbf{I}_d$, respectively Besides PSNR, the structural similarity (SSIM) was also used to assess the visual quality of the decompressed image. The mea-sure of SSIM was developed based on the characteristics of the human visual system (HVS), which integrated the information of structure, luminance and contrast synthetically for the image quality assessment [40].Figure 8 shows the decompressed images of Lena by the proposed scheme with different thresholds *T* , and where the codebook sized 256 was used. Obviously, with the increase of *T* , more blocks are processed by SMVQ and image inpainting, therefore, the compression ratio C_R increases accordingly, and the decompression quality also becomes better because image inpainting has superior recovery capability for smooth blocks compared with VQ.

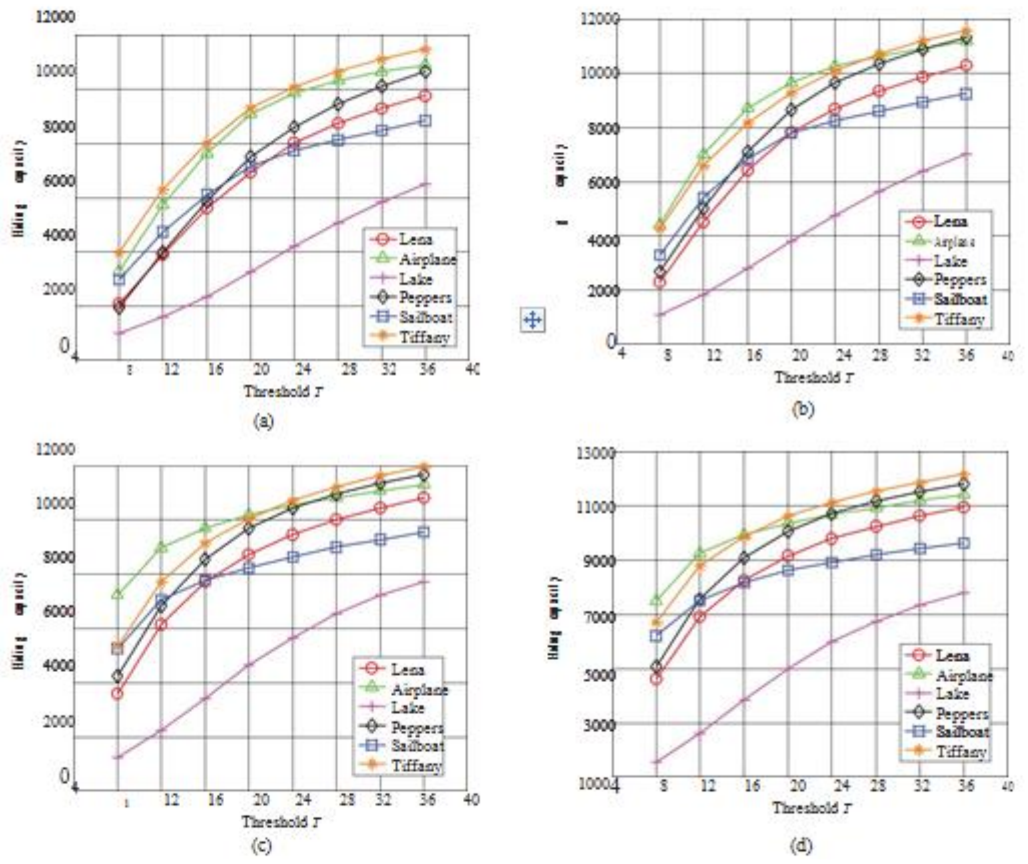


Fig 7. Relationship between hiding capacity and threshold *T* with different codebook sizes *W* . (a) *W* = 128, (b) *W* = 256, (c) *W* = 512, (d) *W* = 1024.

Table I. Comparisons 0f Hiding Capacity Between the Prop Os Ed Scheme And [31]– [33] (Unit: Bits).

Images	Tsai *et al.*'s Scheme [31]	Qian *et al.*'s Scheme [32]	Ni *et al.*'s Scheme [33]	Proposed Scheme
Lena	5237	248	4604	10292
Airplane	7920	526	15384	11182
Lake	4983	457	7320	7247
Peppers	5332	372	5624	11295
Sailboat	6228	513	10546	9301
Tiffany	7522	625	7692	11566
Min. of	2055	124	2139	2246
UCID Max. of	7126	609	91178	11832
UCID Mean of	4852	437	10076	9153

Table II. Comparisons of Compression Performance With Different Code Book Sizes For Lena.

Schemes	^{C}R	*W* = 128			*W* = 256			*W* = 512			*W* = 1024	
		PSNR	SSIM	^{C}R	PSNR	SSIM	^{C}R	PSNR	SSIM	^{C}R	PSNR	SSIM
VQ	18.29	29.70	0.8871	16.00	30.40	0.9101	14.22	31.13	0.9287	12.80	31.67	0.9412
SMVQ	20.66	29.70	0.8871	19.80	30.40	0.9100	19.16	31.12	0.9269	18.32	31.64	0.9413
Scheme in [31]	20.66	28.86	0.8853	19.80	29.75	0.9037	19.16	30.49	0.9254	18.32	30.93	0.9347
Scheme in [27]	21.98	26.62	0.8754	20.20	28.11	0.8986	17.73	28.54	0.9038	15.65	28.94	0.9105
Proposed Scheme	20.66	29.85	0.9086	19.80	30.57	0.9353	19.16	31.27	0.9447	18.32	31.78	0.9526

Table III. Comparisons Of Compression Performance With Different Code Book Sizes For Airplane

Schemes	^{C}R	*W* = 128			*W* = 256			*W* = 512			*W* = 1024	
		PSNR	SSIM	^{C}R	PSNR	SSIM	^{C}R	PSNR	SSIM	^{C}R	PSNR	SSIM
VQ	18.29	29.16	0.9188	16.00	29.81	0.9305	14.22	30.62	0.9479	12.80	31.14	0.9570
SMVQ	21.37	29.16	0.9188	20.49	29.81	0.9305	19.61	30.62	0.9481	18.80	31.12	0.9559
Scheme in [31]	21.37	28.20	0.9001	20.49	28.75	0.9284	19.61	29.97	0.9330	18.80	30.56	0.9416
Scheme in [27]	25.51	26.29	0.8914	21.23	27.38	0.8995	17.70	27.90	0.9026	16.42	28.35	0.9157
Proposed Scheme	21.37	29.31	0.9273	20.49	29.95	0.9367	19.61	30.72	0.9528	18.80	31.24	0.9602

Table IV. Comparisons Of Compression Performance With Different Code Book Sizes For Lake.

Schemes	C_R	***W* = 128**			***W* = 256**			***W* = 512**			***W* = 1024**	
		PSNR	**SSIM**	C_R	**PSNR**	**SSIM**	C_R	**PSNR**	**SSIM**	C_R	**PSNR**	**SSIM**
VQ	18.29	26.63	0.9073	16.00	27.15	0.9189	14.22	27.69	0.9387	12.80	28.11	0.9496
SMVQ	18.83	26.63	0.9073	17.61	27.15	0.9192	16.81	27.69	0.9387	15.75	28.10	0.9513
Scheme in [31]	18.83	26.04	0.9008	17.61	26.58	0.9124	16.81	27.16	0.9325	15.75	27.76	0.9486
Scheme in [27]	20.70	25.23	0.8986	19.72	26.40	0.9042	16.95	26.78	0.9250	15.13	27.24	0.9393
Proposed Scheme	18.83	26.67	0.9164	17.61	27.19	0.9287	16.81	27.74	0.9430	15.75	28.14	0.9524

Table V. Comparisons of Compression Performance With Different Codebook Sizes For Peppers.

Schemes	C_R	***W* = 128**			***W* = 256**			***W* = 512**			***W* = 1024**	
		PSNR	**SSIM**	C_R	**PSNR**	**SSIM**	C_R	**PSNR**	**SSIM**	C_R	**PSNR**	**SSIM**
VQ	18.29	30.15	0.8821	16.00	30.94	0.9009	14.22	31.68	0.9236	12.80	32.25	0.9351
SMVQ	21.23	30.15	0.8821	20.58	30.94	0.9009	19.88	31.67	0.9228	19.16	32.21	0.9346
Scheme in [31]	21.23	27.84	0.8766	20.58	28.36	0.8973	19.88	29.10	0.9153	19.16	29.76	0.9217
Scheme in [27]	21.93	25.28	0.8718	20.45	27.50	0.8894	17.73	27.86	0.9009	15.72	28.48	0.9132
Proposed Scheme	21.23	30.35	0.8999	20.58	31.16	0.9181	19.88	31.87	0.9259	19.16	32.41	0.9388

Table VI. Comparisons of Compression Performance With Different Codebook Sizes For Sailboat.

Schemes	C_R	***W* = 128**			***W* = 256**			***W* = 512**			***W* = 1024**	
		PSNR	**SSIM**	C_R	**PSNR**	**SSIM**	C_R	**PSNR**	**SSIM**	C_R	**PSNR**	**SSIM**
VQ	18.29	28.33	0.9068	16.00	28.85	0.9200	14.22	29.52	0.9398	12.80	30.09	0.9510
SMVQ	20.12	28.33	0.9068	19.05	28.85	0.9201	18.08	29.51	0.9412	17.15	30.08	0.9567
Scheme in [31]	20.12	27.18	0.8992	19.05	27.69	0.9186	18.08	28.34	0.9327	17.15	28.96	0.9455
Scheme in [27]	20.53	25.34	0.8957	19.92	26.79	0.9009	17.18	27.20	0.9105	15.36	27.73	0.9271
Proposed Scheme	20.12	28.42	0.9171	19.05	28.94	0.9325	18.08	29.59	0.9486	17.15	30.15	0.9622

Table VII. Comparisons of Compression Performance With Different Codebook Sizes For Tiffany.

Schemes	C_R	*W* = 128			*W* = 256			*W* = 512			*W* = 1024	
		PSNR	SSIM	C_R	PSNR	SSIM	C_R	PSNR	SSIM	C_R	PSNR	SSIM
VQ	18.29	30.38	0.8982	16.00	30.92	0.9160	14.22	31.46	0.9353	12.80	32.16	0.9453
SMVQ	21.74	30.38	0.8982	20.79	30.92	0.9160	20.20	31.45	0.9349	19.57	32.13	0.9446
Scheme in [31]	21.74	28.35	0.8958	20.79	28.82	0.9027	20.20	29.47	0.9264	19.57	30.18	0.9321
Scheme in [27]	26.46	26.14	0.8746	22.68	27.89	0.8989	18.83	28.05	0.9018	17.04	28.30	0.9152
Proposed Scheme	21.74	30.54	0.8996	20.79	31.09	0.9175	20.20	31.63	0.9348	19.57	32.29	0.9454

Table VIII. Comparisons of Compression Performance With Different Codebook Sizes For UCID.

Schemes	C_R	***W* = 128**			***W* = 256**			***W* = 512**			***W* = 1024**	
		PSNR	**SSIM**	C_R	**PSNR**	**SSIM**	C_R	**PSNR**	**SSIM**	C_R	**PSNR**	**SSIM**
VQ	18.29	29.05	0.9001	16.00	29.69	0.9267	14.22	30.34	0.9423	12.80	30.92	0.9536
SMVQ	20.67	29.04	0.8998	19.74	29.69	0.9273	18.96	30.33	0.9465	18.13	30.90	0.9532
Scheme in [31]	20.67	27.73	0.8994	19.74	28.34	0.9095	18.96	29.09	0.9228	18.13	29.72	0.9301
Scheme in [27]	22.82	25.83	0.8923	20.68	27.36	0.9016	17.68	27.74	0.9102	15.99	28.19	0.9247
Proposed Scheme	20.67	29.19	0.9012	19.74	29.85	0.9294	18.96	30.73	0.9488	18.13	31.69	0.9623

Tables II–VIII show the performance comparisons of the compression ratio and the decompression quality with different codebook sizes, and where T was set to 36. Note that the results of C_R, PSNR and SSIM in Table VIII are the mean values for all the images in the UCID database. We can find from Tables II–VIII that, compared with the standard VQ method, the proposed scheme can achieve the comparable visual quality of decompressed images and obtain greater compression ratios. The standard SMVQ method has the exactly same compression ratio with the proposed scheme and the scheme in [31]. However, the standard SMVQ method cannot carry secret information within its compressed codes. Our scheme not only can carry a large amount of secret bits within the compressed codes as shown in Table I, but also achieves higher decompression quality than the standard SMVQ method and [31] due to the satisfactory recovery property of image inpainting. Although the scheme in [27] can embed more secret bits than our scheme, our scheme has better performance of compression ratio and significantly higher decompression quality than [27]. Furthermore, the proposed scheme can realize data-hiding and image compression simultaneously in a single module, i.e., joint data hiding and compression.

IV. Conclusion

In this paper, we proposed a joint data-hiding and compression scheme by using SMVQ and PDE-based image inpainting. The blocks, except for those in the leftmost and topmost of the image, can be embedded with secret data and compressed simultaneously, and the adopted compression method switches between SMVQ and image inpainting adaptively according to the embedding bits. VQ is also utilized for some complex blocks to control the visual distortion and error diffusion. On the receiver side, after segmenting the compressed codes into a series of sections by the indicator bits, the embedded secret bits can be easily extracted according to the index values in the segmented sections, and the decompression for all blocks can also be achieved successfully by VQ, SMVQ, and image inpainting. The experimental results show that our scheme has the satisfactory performances for hiding capacity, compression ratio, and decompression quality. Furthermore, the proposed scheme can integrate the two functions of data hiding and image compression into a single module seamlessly.

REFERENCES

[1] W. B. Pennebaker and J. L. Mitchell, The JPEG Still Image Data Compression Standard. New York, NY, USA: Reinhold, 1993.

[2] D. S. Taubman and M. W. Marcellin, JPEG2000: Image Compression Fundamentals Standards and Practice. Norwell, MA, USA: Kluwer, 2002.

[3] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression. Norwell, MA, USA: Kluwer, 1992.

[4] N. M. Nasrabadi and R. King, "Image coding using vector quantiza-tion: A review," IEEE Trans. Commun., vol. 36, no. 8, pp. 957–971, Aug. 1988.

[5] Announcing the Advanced Encryption Standard (AES), National Institute of Standards & Technology, Gaithersburg, MD, USA, Nov. 2001.

[6] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, vol. 21, no. 2, pp. 120–126, 1978.

[7] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding- a survey," Proc. IEEE, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[8] C. D. Vleeschouwer, J. F. Delaigle, and B Macq, "Invisibility and application functionalities in perceptual watermarking: An overview," Proc. IEEE, vol. 90, no. 1, pp. 64–77, Jan. 2002.

[9] C. C. Chang, T. S. Chen, and L. Z. Chung, "A steganographic method based upon JPEG and quantization table modification," Inf. Sci., vol. 141, no. 1, pp. 123–138, 2002.

[10] H. W. Tseng and C. C. Chang, "High capacity data hiding in JPEG-compressed images," Informatica, vol. 15, no. 1, pp. 127–142, 2004.

[11] P. C. Su and C. C. Kuo, "Steganography in JPEG2000 compressed images," IEEE Trans. Consum. Electron., vol. 49, no. 4, pp. 824–832, Nov. 2003.

[12] W. J. Wang, C. T. Huang, and S. J. Wang, "VQ applications in steganographic data hiding upon multimedia images," IEEE Syst. J., vol. 5, no. 4, pp. 528–537, Dec. 2011.

[13] Y. C. Hu, "High-capacity image hiding scheme based on vector quanti-zation," Pattern Recognit., vol. 39, no. 9, pp. 1715–1724, 2006.

[14] Y. P. Hsieh, C. C. Chang, and L. J. Liu, "A two-codebook combination and three-phase block matching based image-hiding scheme with high embedding capacity," Pattern Recognit., vol. 41, no. 10, pp. 3104–3113, 2008.

[15] C. H. Yang and Y. C. Lin, "Fractal curves to improve the reversible data embedding for VQ-indexes based on locally adaptive coding," J. Vis. Commun. Image Represent., vol. 21, no. 4, pp. 334–342, 2010.

[16] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantiza-tion design," IEEE Trans. Commun., vol. 28, no. 1, pp. 84–95, Jan. 1980.

[17] C. C. Chang and W. C. Wu, "Fast planar-oriented ripple search algorithm for hyperspace VQ codebook," IEEE Trans. Image Process., vol. 16, no. 6, pp. 1538–1547, Jun. 2007.

[18] W. C. Du and W. J. Hsu, "Adaptive data hiding based on VQ com-pressed images," IEE Proc. Vis., Image Signal Process., vol. 150, no. 4, pp.233–238, Aug. 2003.

[19] C. C. Chang and W. C. Wu, "Hiding secret data adaptively in vector quantisation index tables," IEE Proc. Vis., Image Signal Process., vol. 153, no. 5, pp. 589–597, Oct. 2006.

[20] C. C. Lin, S. C. Chen, and N. L. Hsueh, "Adaptive embedding techniques for VQ-compressed images," Inf. Sci., vol. 179, no. 3, pp. 140–149, 2009.

[21] C. H. Hsieh and J. C. Tsai, "Lossless compression of VQ index with search-order coding," IEEE Trans. Image Process., vol. 5, no. 11,pp. 1579–1582, Nov. 1996.

[22] C. C. Lee, W. H. Ku, and S. Y. Huang, "A new steganographic scheme based on vector quantisation and search-order coding," IET Image Process., vol. 3, no. 4, pp. 243–248, 2009.

[23] S. C. Shie and S. D. Lin, "Data hiding based on compressed VQ indices of images," Comput. Standards Inter., vol. 31, no. 6, pp. 1143–1149, 2009.

[24] C. C. Chang, G. M. Chen, and M. H. Lin, "Information hiding based on search-order coding for VQ indices," Pattern Recognit. Lett., vol. 25, no. 11, pp. 1253–1261, 2004.

[25] T. Kim, "Side match and overlap match vector quantizers for images," IEEE Trans. Image Process., vol. 1, no. 2, pp. 170–185, Apr. 1992.

[26] C. C. Chang, W. L. Tai, and C. C. Lin, "A reversible data hiding scheme based on side match vector quantization," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 10, pp. 1301–1308, Oct. 2006.

[27] C. C. Chen and C. C. Chang, "High capacity SMVQ-based hid-ing scheme using adaptive index," Signal Process., vol. 90, no. 7, pp. 2141–2149, 2010.

[28] L. S. Chen and J. C. Lin, "Steganography scheme based on side match vector quantization," Opt. Eng., vol. 49, no. 3, pp. 0370081–0370087, 2010.

[29] S. C. Shie and J. H. Jiang, "Reversible and high-payload image steganographic scheme based on side-match vector quantization," Signal Process., vol. 92, no. 9, pp. 2332–2338, 2012.

[30] C. F. Lee, H. L. Chen, and S. H. Lai, "An adaptive data hiding scheme with high embedding capacity and visual image quality based on SMVQ prediction through classification codebooks," Image Vis. Comput., vol. 28, no. 8, pp. 1293–1302, 2010.

[31] P. Tsai, "Histogram-based reversible data hiding for vector quantisation-compressed images," IET Image Process., vol. 3, no. 2, pp. 100–114, 2009.

[32] Z. X. Qian and X. P. Zhang, "Lossless data hiding in JPEG bitstream," J. Syst. Softw., vol. 85, no. 2, pp. 309–313, 2012.

[33] Z. C. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar. 2006.

[34] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpaint-ing," in Proc. 27th Int. Conf. Comput. Graph. Interact. Tech., New Orleans, LA, USA, Jul. 2000, pp. 417–424.

[35] C. Qin, S. Wang, and X. Zhang, "Simultaneous inpainting for image structure and texture using anisotropic heat transfer model," Multimedia Tools Appl., vol. 56, no. 3, pp. 469–483, 2012.

[36] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," J. Vis. Commun. Image Represent., vol. 12, no. 4,pp.436–449, 2001.

[37] C. Qin, F. Cao, and X. Zhang, "Efficient image inpainting using adaptive edge-preserving propagation," Imag. Sci. J., vol. 59, no. 4, pp. 211–218, 2011.

[38] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," IEEE Trans. Image Process., vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[39] G. Schaefer and M. Stich, "UCID—An uncompressed color image database," Proc. SPIE, vol. 5307, pp. 472–480, Jan. 2004.

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.