Compression and Data Hiding Based on SVD and Integer Wavelet Transform

T. Radhika¹, Anitha Swanidas², G. Kiranayé³

¹PG Scholar, Dept of ECE, Gurunanak Institutions Technical Campus, Hyderabad, India, E-mail: tradhika6@gmail.com.
²Professor & HOD, Dept of ECE, Gurunanak Institutions Technical Campus, Hyderabad, India.
³Associate Professor, Dept of ECE, Gurunanak Institutions Technical Campus, Hyderabad, India.

Abstract: Today Reversible data hiding (RDH) is latest research area in field of data concealment scheme. In RDH, the data to be cover up is embedded in the cover image and at the origin side both conceal secret and cover image is reconstructed without any misrepresentation. Data hiding are a group of techniques used to put a secure data in a host media (like images) with small deterioration in host and the means to extract the secure data afterwards. For example, steganography can be named. Steganography is one such pro-security innovation in which secret data is embedded in a cover. Reversible data-hidings insert information bits by modifying the host signal, but enable the exact (lossless) restoration of the original host signal after extracting the embedded information. Proposal of a novel joint data-hiding and compression scheme for digital images using side match vector quantization (SMVQ) and image in painting. 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 in painting adaptively according to the current embedding bit. The concept of hiding data can be further extended by integer wavelet transform where the data dwt is implemented in the spatial domain and the data can be hided in subbands of the wavelet the performance is fast and reliable. PSNR parameters show the proposed scheme and extended scheme accuracy results.

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

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 code words 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 subcode books, and the best match in one of the subcode books 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 code word-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

Copyright © 2015 IJIT. All rights reserved.
threshold according to the required hiding capacity and arranged a number of similar code words 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 code book and the subcode books are used to generate the index values, excluding the blocks in the leftmost column and the top most row. Recently, many researchers have studied on embedding secret message by SMVQ [26]–[31]. 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 adaptive index [27]. Chang proposed an SMVQ scheme in this paper is based on SMVQ and image in painting adaptively according to current image blocks, the receiver can segment the compressed codes 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 in painting. 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 in painting 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.

II. JOINT DATA-HIDING AND COMPRESSION SCHEME

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.

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 in painting. 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 in painting 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.

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 subcode books are stored. In our scheme, the standard algorithm of SMVQ is modified to further achieve better data compression rate 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 subcodebooks with W code words, and each code word length is n2. Denote the original uncompressed image sized M × N as I, and it is divided into the non-overlapping n × 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 Bk, where k = M × N/n2, i = 1, 2, . . . , M/n, and j = 1, 2, . . . , 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 I, i.e., B1, i (i = 1, 2, . . . , M/n) and B1, j (j = 2, 3, . . . , 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 Bk,x,y (2 ≤ x ≤ M/n, 2 ≤ y ≤ N/n), and its left and up blocks are Bk,x−1,y and Bk−1,y, respectively. As shown in Fig.2, cp,1 (1 ≤ p ≤ n) and c1,q (2 ≤ q ≤ n) represent the 2n−1 pixels in the left and upper borders of Bk,y. The n pixels in the right border of Bk,x−1 and the n pixels in the bottom border of

International Journal of Innovative Technologies
Volume.03, IssueNo.02, June-2015, Pages: 0131-0136
Compression and Data Hiding Based on SVD and Integer Wavelet Transform

B_{x,y-1} are denoted as $B_{p,n}$ (1 ≤ p ≤ n) and $u_{n,q}$ (1 ≤ q ≤ n), respectively. Similar with SMVQ, the 2n - 1 pixels in the left and upper borders of $B_{x,y}$ are predicted by the neighboring pixels in $B_{x,y-1}$ and $B_{x-1,y}$; $c_{1,1} = (l_{1,0} + u_{0,1})/2$, $c_{p,1} = 1_{p,0}(2 ≤ p ≤ n)$, and $c_{1,q} = u_{n,q}$ (2 ≤ q ≤ n). Instead of all $n^2$ pixels in $B_{x,y}$, only these 2n - 1 predicted pixels are used to search the codebook. After transforming all W code words into the codebook, in order to conduct compression, which means the elements of each codeword $C_w$ sized n × n.

$$E^D = \sum_{p=1}^{n} (c_{p,1} - c_{p,1}^{0})^2 + \sum_{q=2}^{n} (c_{1,q} - c_{1,q}^{0})^2$$

(1)

where $c_{p,q}$ are the elements of each codeword $C_w$ in codebook.

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

The R code words with the smallest MSEs, i.e., $E_{w}$, are selected to generate one subcode book $B_{x,y}$ for the block $B_{x,y}$ (R < W). Suppose that, among the R code words in x,y, the codeword indexed $\lambda$ has the smallest MSE, i.e., $E_{\lambda}$, with all $n^2$ pixels in $B_{x,y}$ (0 ≤ $\lambda$ ≤ R - 1). If the value of $E_{\lambda}$ is greater than a pre-determined threshold T for distortion control, it implies that the current residual block $B_{x,y}$ 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 code book is used to compress the block $B_{x,y}$, and no secret bits are embedded. Otherwise, if $E_{\lambda} \leq T$, it implies that the current residual block $B_{x,y}$ 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 $B_{x,y}$ 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 $B_{x,y}$. If not, the indicator bit, i.e., 1, is added as the prefix of the compressed code for $B_{x,y}$.

As for the block $B_{x,y}$, if its $E_{\lambda}$ 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 $\lambda$ occupying $\log_2 R$ bits is used to represent the block $B_{x,y}$ in the compressed code. Because the code word number R in subcode book $B_{x,y}$ is less than the code word number W of the original codebook, the length of the compressed code for $B_{x,y}$ using SMVQ must be shorter than using VQ. On the other hand, if $E_{\lambda} \leq T$ 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 as 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 $E_{\lambda} \leq T$, it implies that $B_{x,y}$ locates in a relatively smooth region. Thus, it is suitable to conduct image inpainting in the compression for $B_{x,y}$ under this condition. In our scheme, a PDE-based image inpainting method using the fluid dynamics model is adopted [34]. Denote $B_{x}$ as the region including the current block $B_{x,y}$ that needs compression by inpainting and the available neighboring region of $B_{x,y}$. Let $B_{x} (\xi, \eta)$ be the gray value of $B_{x}$ in the coordinate (\xi, \eta). The Laplacian $B_{x} (\xi, \eta)$ is used as a smoothness measure of the region $B_{x}$. 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:

$$\nabla \perp B_{x}(\xi, \eta) = \left(-\frac{\partial}{\partial \xi} \mathbf{i} - \frac{\partial}{\partial \eta} \mathbf{j}\right)B_{x}(\xi, \eta),$$

(2)

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

$$\nabla \left[ \Delta B_{x}(\xi, \eta) \right] \cdot \nabla \perp B_{x}(\xi, \eta) = 0.$$  

(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 acquired:

$$\frac{\partial}{\partial t} B_{x}(\xi, \eta) = \nabla \left[ \Delta B_{x}(\xi, \eta) \right] \nabla \perp B_{x}(\xi, \eta), \quad \forall (\xi, \eta) \in B_{x,y}. $$

(4)

International Journal of Innovative Technologies
Volume 03, IssueNo.02, June-2015, Pages: 0131-0136
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 $B_{x,y}$ reach stable state. Consequently, the structural and geometric information of the block $B_{x,y}$ can be recovered effectively without serious blurring on edges. Consequently, when $s = 1$, in order to indicate that block $B_{x,y}$ is processed by inpainting and differentiate from the index $\lambda$ produced by SMVQ, the index value $R$ occupying $\log_2(R+1)$ bits is used as the compressed code of $B_{x,y}(R > \lambda)$. For simplicity, we assume that $\log_2(R+1)$ is an integer and $\log_2(R) = \log_2(R+1)$. After the current block $B_{x,y}$ is processed, the following block in raster-scan 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 top most 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-scan order. Fig. 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 Fig. 3. Flowchart of decompression and secret data extraction for each residual block. decimal value of the last $\log_2 (R+1)$ bits in this section as $\lambda'$. Under this circumstance, if $\lambda' = R$, 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. If the current segmented section corresponds to an inpainting compressed block $B_{\text{inp}}$, 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 $B_{\text{smq}}$, SMVQ index value, i.e., $\lambda'$, 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 $2^{n-1}$ pixels in the left and upper borders of $B_{\text{smq}}$ are estimated by the neighboring pixels in its left and upper decompressed blocks. Similarly, the MSEs are calculated between these $2^{n-1}$ predicted pixels in $B_{\text{smq}}$ with the corresponding values of all W code words in the codebook. Then, the R code words in with the smallest MSEs are chosen to generate a subcode book.

![Fig. 3. Six standard test images.](image)

Finally, the codeword indexed $\lambda'$ in the generated subcode book is used to recover the block $B_{\text{smq}}$. 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 $I_d$ can be obtained successfully. Due to the decoding of the compressed codes, the decompressed image $I_d$ 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. 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
Compression and Data Hiding Based on SVD and Integer Wavelet Transform

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 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

T. RADIKA, ANITHA SWANDAS, G. KIRANNAYE


Author Profile:

T RADIKA, PG Scholar, Dept of ECE, Gurunanak Institutions Technical Campus, Hyderabad, India, E-mail: tradhika6@gmail.com.