A Feasible Image Hiding Scheme Based on JPEG and VQ

Shinfeng D. Lin, Shih-Chieh Shie, and Chung-Chien Chou
Department of Computer Science and Information Engineering,
National Dong Hwa University, Hualien, Taiwan, R.O.C.
E-mail: david@mail.ndhu.edu.tw

Abstract

A feasible image hiding scheme is proposed in this article. The goal of the proposed scheme is to obscurely deliver some images via a JPEG image file. Secret images are first encoded into binary indexes by vector quantization. Then, these indexes are embedded into JPEG file of cover image by modifying its quantized coefficients and quantization factors in some invertible way. The stego-file keeps the JPEG syntax, which can be displayed by any standard JPEG decoder, without noticeable distortion. A legal receiver with the key can completely extract the secret images and restore the original JPEG image file at the same time. Simulation results demonstrate that the proposed scheme is practicable.

Keywords: Data hiding, JPEG, vector quantization, cryptography, and steganography.

1. Introduction

Since digital technology grows so rapidly in recent years, the storage format of traditional analog signals faces a revolution. Images, videos, audios, and texts are digitalized one by one and saved as multimedia in the computer equipment to achieve the purpose of better and longer preservation. On the other hand, the development of Internet also brings the convenience of our daily lives.

Generally speaking, digital data have their commonality – the fidelity and plasticity among duplications. At the same time, people may receive and transmit data via network communication more easily and quickly. However, the ownership certification problems usually occur when the inventors try to sell or share their products on the web. Moreover, the issues of data integrity and consistency arise after many times deliveries. How to prevent the contents of digital data from revealing is even another question as many ill-affected individuals intercept the transmitted data package in practice. Therefore, the major data security problems include: copyright protection, data integrity, and content confidentiality. Many techniques have been developed to solve these problems. They are mainly classified into two domains: cryptography and steganography.

Cryptographic encryption is a traditional method to protect data against unauthorized usage. This kind of techniques usually scrambles important data, which are usually referred to plain-texts, into meaningless sequences, which are so-called cipher-texts, with a predetermined key. Data are kept in safety if it is impossible to invert cipher-texts to plain-texts without the key. In more detail, it is cryptographically secure if decrypting the scrambled data in a limited time is infeasible for any illegal interceptor, while others can easily extract the important information with a secret key. Conventional encryptions usually manipulate the plain-texts by permutations, substitutions, or mathematical operations with a single key. They are also called symmetric key cryptosystems. In the late few years, asymmetric key cryptosystems, which include a private key and another public key in the scheme, are proposed and widely used. Generally speaking, cryptography has the ability to keep cipher-texts in amazing safety. However, after the data are decrypted, there is no protection to the data any more. Hence, cryptography is not always appropriate to deal with data security problems.

Steganography or data embedding refers to techniques of inserting some information, such as watermarks, signatures, or error correction codes, into other host media. It is usually accomplished by modifying host media themselves, and the modification should not introduce noticeable artifacts. In late years, the issues of steganography in digital images have become more and more popular. However, they can be further classified into two regions, digital watermarking and information hiding, according to their applications. Data embedding (digital watermarking) can achieve the task of copyright protection or data integrity. It depends on what we insert into host data, however. Once we embed trademarks or insignias, for instances, they may be useful for ownership identification. If any error detection codes or something helpful to tamper-proofing are inserted, the consistency of host data can then be ensured. Under this situation, the host data are more valuable and necessary than the watermarks. On the other hand, data embedding (information hiding) can also be employed on covert communication. Data to be secretly transmitted, called embedded-media, may be obscurely inserted into another data, called cover-media, and the stego-media are obtained. If the stego-media look the same with cover-media, secret data can be steganographically securely distributed to anywhere without attracting interceptors. On the contrary, data encrypted by cryptographic algorithms often catch their eyes. Therefore, information hiding is quite working on the premise of nothing susceptible. In general, information hiding makes secret data invisible and safe, while digital watermarking protects host data by hidden information.

A feasible image hiding scheme that based on JPEG compression standard and vector quantization is presented here. The rest of this article is organized as follows. In Section 2, we address the preliminaries and the survey of related works with respect to data embedding. The proposed JPEG and VQ based image hiding scheme is then introduced in Section 3. Section 4 demonstrates the simulation results and experimental analyses. Finally, conclusions are given in Section 5.

2. Preliminaries and Related Works

First, the JPEG compression standard and the concept of vector quantization are briefly introduced in this section. Second, some related works about image hiding techniques are presented. And finally, the criterion for measuring the quality of image is given.

2.1 Brief Introduction of JPEG

In JPEG specification, four operation modes have been designed for adapting to various compression requirements [1]. These modes include sequential DCT-based mode, progressive DCT-based mode, lossless mode, and hierarchical mode. In this section, we only introduce the baseline mode, i.e. the sequential DCT-based mode, of JPEG compression standard. First, the source image is divided into non-overlapping blocks with size 8×8. Next, each pixel of image block is subtracted with a value of 128. Then, the DCT transformation is applied on each image block. After quantized by the recommended quantization table, as shown in Fig. 1, the DCT coefficients are zigzag-scanned and entropy-encoded by some specified tables one by one. Finally, the compressed data are generated.

Discrete Cosine Transform (DCT) is a kind of decomposition that converts images from spatial domain into frequency domain. DCT contributes to separation and positioning of various energy of an image. Here are the equations of forward and inverse DCT regarding an image block with size of $N \times N$:

$$DCT(i,j) = C(i)C(j)\sum_{r=0}^{N-1}\sum_{c=0}^{N-1} pixel(r,c) * T \quad (1)$$

$$pixel(r,c) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} C(i)C(j)DCT(i,j) * T \quad (2)$$

where

$$C(i), C(j) = \begin{cases} \sqrt{\dfrac{1}{N}} & \text{for } i, j = 0 \\ \sqrt{\dfrac{2}{N}} & \text{for } i, j = 1, 2, ..., N-1 \end{cases}$$

and $T = \cos\left[\dfrac{(2r+1)i\pi}{2N}\right]\cos\left[\dfrac{(2c+1)j\pi}{2N}\right]$

The elements of quantization table are called quantization factors, which denote the scalar quantization steps. The quantization equation in JPEG is:

$$DCT^Q[i,j] = IntegerRound(\frac{DCT[i,j]}{QT[i,j]}) \quad (3)$$

$DCT[i, j]$ and $DCT^Q[i, j]$ denote the coefficients at index $[i, j]$ of a block before and after quantization, respectively. $QT[i, j]$ denotes the quantization factor at index $[i, j]$ of the quantization table. However, most of image processing tools usually define another mapping equation from a given quality factor to certain quantization factors. It is listed below:

$$SF = \begin{cases} 5000/QF & ,if\ QF \le 50 \\ 200 - QF * 2 & ,if\ QF > 50 \end{cases} \quad (4)$$

$$QT[i,j] = (StdQT[i,j] * SF + 50)/100 \quad (5)$$

QF is assigned a value from 1 to 99, which denotes the predetermined image quality. $StdQT[i, j]$ denotes the quantization factor at index $[i, j]$ of the recommended table.

2.2 The Concept of VQ

In general, vector quantization (VQ) is a lossy compression technique for image. First of all, it extracts some dominant data, referred as codewords, from a training set. Afterward, it uses the indexes of the closest codewords to represent the data instead of the data themselves. Since the size of indexes is much smaller than that of data, compression can thus be achieved.

In the encoder, source images are first divided into non-overlapping blocks. For each block, the encoder will search the codebook for the most similar codeword, and the corresponding index will be kept and transmitted for later decoding. Once the decoder has to recover the compressed data, it just looks up the codebook with the index and pastes the corresponding codeword back. Then, a decoded image is reconstructed. Note that the encoder and decoder must equipped with the same codebook.

As for codebook generation, most researchers apply the famous LBG algorithm [2] to train codebook. The codewords of codebook really affect the quality of decompressed images. If a similar codeword could not always be found for each image block, the quality of decompressed images may not satisfy people. However, if the image to be VQ encoded had been included in the training set, the gaps between image blocks and codewords can be effectively reduced, and this is so-called inside-training. On the contrary, encoding images by an outside-trained codebook will lead to bottleneck quality. Hence, it is a key point for VQ whether the source images are included in the training set.

2.3 Related Works of Data Embedding

The techniques of steganography, as known as data embedding, have been significantly studied recently. These schemes can be derived into two branches: digital watermarking and information hiding [3]. The former usually provides the protection of intellectual property, whereas the latter concerns the privacy of data contents. Since copyright protection problems arise in this digital world, there are plenty of papers discussing how to exploit the watermarking scheme as well. Especially, invisible robust watermarking technique is among the most outstanding keepers against unauthorized usages [4]. In general, a watermarking scheme embeds recognizable signatures or something helpful into the host media that have to be protected by directly modifying the media themselves [5, 6]. These marks can be easily inserted via substituting pixel values in the spatial domain [7]. Expectably, the insertion or substitution can also be applied on coefficients in the frequency domain [8, 9]. Besides, digital watermarking can be used for not only copyright protection but also tamper-proofing. No matter what applications they are, it is necessary for them to survive JPEG compression. Therefore, most of the watermarking schemes will discuss their experimental results under JPEG compression [10]. Moreover, there are even some techniques combining themselves with JPEG encoder and decoder. Fridrich et al. proposed an invertible algorithm for JPEG images to embed some authentication bits for tampering detection [11]. Since host images of most watermarking scheme are somewhat distorted, Fridrich's invertible algorithm, which can recover the distorted images to the original ones, is an important contribution.

Although the basic theorems and targets of steganography differ from those of cryptography, there are still some researchers trying to combine these two schemes. Visual cryptography is a successful example that utilizes Human Visual System (HVS) to decrypt the hidden information [12]. The base of visual cryptography is built on (t, n) threshold secret sharing algorithm [13]. The secret sharing scheme splits a secret image into n different shares at the beginning. Nevertheless, each share looks independent from the original. The secret image can be revealed only by super-imposing t out of n shares. Hence, visual cryptography can play a role of image hiding [14]. However, there are some researches applying visual cryptography on their watermarking algorithm [15].

Data embedding can also be used for covert communication. In the prior researches, raw images without any compression are considered as cover media [16]. In order to solve the problems of inefficient capacity, Chen et al. proposed a solution that secret images should be compressed by VQ and then encrypted before the embedding process, which is called virtual image cryptosystem [17]. Hu et al. also proposes a revised algorithm of virtual image cryptosystem [18]. They split the pixel value into two parts. The significant one is used for codebook training, and the insignificant one is used for information hiding by greedy substitution. Their schemes have been simulated and proven that multiple images can be hidden concurrently while the quality of decrypted images is acceptable.

There are more and more researchers participating in establishing practicable image hiding scheme. However, a scheme will be really working only if it is combined with the popular compression standard since there are few people transmitting uncompressed data. Hence, hiding images during the JPEG encoding process may be feasible. Moreover, most data embedding strategies insert data by modifying the host data directly, which means the host data will never be restored to the original and their values will thus be lost.

2.4 Measuring the Quality of Image

In the image processing area, people need to determine whether the image quality is good enough. Hence, most researchers use an objective and standardized criterion, named Peak Signal-to-Noise Ratio (*PSNR*), to compute the distortion between two given images. Larger *PSNR* value refers to higher similarity. For 256 gray level images, *PSNR* is defined as:

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \qquad (6)$$

MSE is the Mean-Squared-Error between two images with the size of $N_1 \times N_2$. That is:

$$MSE = \frac{1}{N_1 \times N_2} \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} (x(i,j) - \tilde{x}(i,j))^2 \quad (7)$$

3. The Proposed Image Hiding Scheme

The details of the proposed image hiding scheme was presented in this section. This work claims two main points: practical purpose and invertible property. As for practical purpose, we have to combine the image hiding scheme with popular image compression standard. Therefore, the most widely used image coding algorithm, JPEG, was selected. As for invertible property, we refer to the scheme proposed by Fridrich et al. [11]. However, Hu's algorithm [18] indeed impacts us a lot, because the idea of embedding VQ indexes is quite interesting and full of efficiency. Therefore, an invertible image hiding scheme seems feasible.

In the proposed scheme, the cover medium is a standard JPEG image file, and the embedded information includes one or more images with the same size as cover image. Figure 2 briefly shows the concept of the proposed scheme. Secret images are first VQ encoded with the codebook generated from the covering JPEG image. Then, these indexes are subsequently embedded into covering JPEG file based on an invertible algorithm. After being transmitted over the net and received by the receiver, the secret images can be extracted and the covering JPEG file can also be recovered from the stego-JPEG image file. The detailed flowchart of secret image embedding is shown in Fig. 3.

3.1 Quantization Table Selection

The first step of the proposed scheme is to determine the quality of cover image. In the standard JPEG encoding process, quantization table is the only factor that affects the quality of images. If the factors of quantization table are universally large, the compressed image will be with smaller file size but lower quality. Oppositely, if the quantization factors are small, the compressed image will be with larger file size but better quality.

Selecting a proper quantization table is very important in the proposed scheme. Once all quantization factors are decided, the embedding capacity of the JPEG image will also be determined. This is a characteristic of the proposed image hiding scheme since all information is hidden in the positions of even numbers among all quantization factors. This process will be stated in the following subsection in more detail. Anyhow, the more quantization factors of factor 2, the more information we can embed. Figure 4 shows a quantization table and its corresponding table for data hiding. The number in the corresponding table reveals the embedding capacity of such position.

3.2 Codebook Generation

After DCT transformation of cover image, all coefficients are scalar quantized by a predetermined quantization table. However, there are two processes in the next step. One progresses like the standard JPEG encoder, and the other passes de-quantization and inverse DCT operations. The latter process will reconstruct a decoded image in the encoder, and this image is used for codebook training process. In this scheme, we adopt the famous LBG algorithm [2] with splitting initialization algorithm. Note that some parameters used for codebook training, such as stopping thresholds, should be recorded in the key for codebook generation in the decoder.

3.3 VQ Encoding

In this step, secret images are VQ encoded by the codebook generated in the previous process. For any given secret image block, one may search the entire codebook for a closest codeword, just before saving its index. There are many techniques equivalent to full search algorithm but spending less time; here we choose the L_2-norm algorithm [19] as our speeding method. Then, these VQ indexes can be multiplexed or further encrypted optionally. Finally, information to be hidden into the cover medium is generated.

3.4 Embedding Space Allocation

This process will subsequently ask the JPEG file for legal space to embed the secret information. It is very important that the allocation operations should not violate the syntax rules of a standard JPEG decoder. In addition, we have to ask for the space in an invertible way. Once we do not need the space any more, we can completely free them. As mentioned in Section 2, Fridrich et al. have proposed a feasible method that modifies quantized coefficients and quantization factors concurrently [11]. However, we only choose those positions where the quantization factor is even. For the odd number, we leave it unchanged. After this step, some extra embedding positions are allocated perfectly. Furthermore, we propose two strategies for space allocation here. One is according to the zigzag scan order, and the other is decided in order of the magnitude of all quantization factors.

3.4.1 Scan Order Strategy (SO)

In general, quantization factors in the front part of the zigzag scan order are smaller than the others. It is because the leading coefficients, which are called low frequency coefficients, are somehow important than the others. However, it forms a sequence with lots of zero at the ending quantized coefficients. After applying run-length-coding, these zero terms are massively erased and the compression ratio is thus kept high. On the premise to increase the JPEG file size as slight as possible, we should not embed information into the end of sequence, which may properly lead to a large increment of file size. Therefore, embedding these indexes according to scan order is hypothetically efficient regarding the size of cover media. Using this strategy, we have to ask the embedding space from the leading coefficients in order of zigzag scan until there is no space available any more.

3.4.2 Minimal-Value-First Strategy (MVF)

By contrast with SO strategy, the quality of stego-media should be another consideration. Quantization can be considered as a kind of importance normalization of all coefficients. After quantization, the importance of all coefficients can be viewed as the same. However, embedding a bit of 1 in our method is equivalent to adding half of the quantization factors to the coefficient. Hence, the lager quantization factor is, the more distortion it introduces. If one emphasizes on the image quality of stego-media, he should adopts this strategy hypothetically.

3.5 Entropy Encoding

The last process of the proposed scheme is to encode the modified coefficients based on the entropy encoder of JPEG. This will form a standard JPEG file stream. And finally the predefined quantization table has to be appended into the header of the JPEG file.

3.6 Content of Key

Some information has to be kept for future use in data extraction process. Note that the key should be carefully kept and transmitted to receiver. Here briefly lists the content of key.

(a). Codebook training parameters inclusive of stopping threshold, codebook size, and codeword dimension.

(b). Hiding position table with the size of 8×8 counters.

(c). Embedded secret image number and size.

(d). Flag of embedding strategy.

The summarized steps of the proposed data hiding scheme are listed below:

Step 1. Predefine the quantization table.

Step 2. Quantize and de-quantize DCT coefficients of cover image.

Step 3. Train the codebook from the reconstructed cover image.

Step 4. Encode secret images based on the codebook by VQ.

Step 5. Modify DCT coefficients of cover image and quantization table.

Step 6. Embed the VQ indexes of secret images into cover medium.

Step 7. Encode the final DCT coefficients into JPEG file streams by entropy encoder.

3.7 Secret Image Extracting Algorithm

The secret image extracting process is simple and similar to the inverse of hiding algorithm. Figure 5 shows the detailed extracting flowchart, and the summarized image extracting steps are listed below.

Step 1. Decode the JPEG streams of received stego-JPEG image file by entropy decoder, and extract the quantization table from file header.

Step 2. Extract VQ indexes of secret images with the key.

Step 3. Transform the DCT coefficients based on the inverse DCT transformation and restore the quantization table with the key.

Step 4. Train the codebook based on the restored cover image.

Step 5. Decode secret images by VQ.

4. Simulation Results

To measure the feasibility of the proposed image hiding scheme, we have conducted a series of experiments. In our experiments, the test image “*Airplane*” is adopted as the cover JPEG image. Three secret images are "*Lena*", "*Pepper*" and "*Toys*". All of these test images are with 256 gray levels and with size 512×512 pixels.

Table I shows the experimental results of the proposed scheme under different quality factors of cover JPEG image. The codeword dimension is 4×4, and the codebook sizes are 64 and 256, respectively. The results show that quality factor only affects the quality and file size of stego-image. This is one characteristic of JPEG image. Considering the quality of extracted secret images with the same codebook size, however, we can find that quality factor does not affect secret images very much. The reason can be imputed to codebook training sets. Because the codebook is trained from the covering JPEG image, which is not related to the secret images at all. Moreover, this table also reveals that larger codebook size results in higher quality of extracted secret image. This is one property of VQ. However, larger size of codebook implies more bits have to be embedded into the cover image. And this further implies lower quality and larger file size of stego-image.

Table II shows the experimental results of the proposed scheme where 2 and 3 secret images are embedded, respectively. There are some blanks left in this table just because the embedding capacity of quantization table generated from quality factor is not enough. However, we can directly modify quantization factors to the nearest even number in order to increase the capacity if needed. Once the quality factor, codebook size, codeword dimension are chosen, the quality of extracted secret-images will be determined. Nevertheless, the image quality and file size of stego-image still vary due to the embedded indexes. The architecture of the proposed scheme is similar with Hu et al. [18]. However, our scheme is designed for JPEG encoder and decoder. This improves the practicability due to smaller file size and popular format of cover image, while Hu's method applies to raw image. Another advantage of the proposed scheme is that the cover image can be perfectly restored back as the original one, while Hu's method modifies and hurts the cover image. Nevertheless, we still list the results of Hu's scheme for reference, even though the platform completely differs. The *PSNRs* of the extracted secret images "*Lena*", "*Pepper*", and "*Toys*" are 28.51 dB, 28.08 dB, and 24.11 dB, respectively, with codebook size 256 and codeword dimension 4×4. And the file size of cover image is 262144 bytes. As shown in Table II, the proposed scheme achieves almost the same result with codebook size 64 and quality factor 80. Furthermore, the file size of cover image by our scheme is 121163 bytes, and the quality of cover image can be restored back to 40.04 dB.

Table III gives the experimental results of the proposed scheme by different embedding strategies. In this table, SO denotes "Scan-Order" strategy and MVF denotes "Minimal-Value-First" strategy. It clearly reveals that, for both SO and MVF strategies, the larger the codebook size is, the larger the file size of stego-image is. The experimental data indirectly supports our hypothesis. As compared with SO strategy, MVF strategy gains 0.69% improvement of image quality but 1.81% increment of file size on average. Hence, there exists a trade-off between image quality and file size of cover image.

5. Conclusion

A feasible image hiding scheme that is suitable for current JPEG image file has been introduced in this article. This scheme is developed on the basis of JPEG compression standard and vector quantization. During the JPEG encoding process, several secret images are embedded into the cover JPEG image by an invertible algorithm. After extracting the embedded images, one can completely remove the extra information and restore the original JPEG image. The proposed scheme may be useful for secret communication, such as military map delivery. It can also be considered as a brand-new compression method that multiplexes several images into one JPEG image. Simulation results demonstrate the practicability of the proposed scheme.

Acknowledgments

The authors are supported by National Dong Hwa University (Taiwan, R. O. C.) and National Science Council (Taiwan, R. O. C.) with project no. NSC 91-2213-E-259-016.

References

[1] CCITT Recommendation T.81, "Digital Compression and Coding of Continuous-tone Still Images," 1992.

[2] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transaction on Communications*, vol. 28, no. 1, pp. 84-95, Jan. 1980.

[3] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information Hiding - A Survey," *Proceedings of the IEEE*, vol. 87, no. 7, July 1999.

[4] F. Mintzer, G. W. Braudaway, and M. M. Yeung, "Digital Watermarking for High-quality Imaging," *IEEE First Workshop on Multimedia*

Signal Processing, pp. 357-362, 1997.

[5] H. Berghel and L. O'Gorman, "Protecting Ownership Rights through Digital Watermarking," *IEEE Computer Mag.*, pp. 101-103, July 1996.

[6] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies," *Proceedings of the IEEE*, vol. 86, pp. 1064-1087, June 1998.

[7] N. Nikolaidis and I. Pitas, "Robust Image Watermarking in the Spatial Domain," *Signal Processing*, vol. 66, pp. 385-403, 1998.

[8] R. Wolfgang and E. J. Delp, "A Watermark for Digital Image," *in Porc. of IEEE Int. Conf. on Image Processing*, vol. 3, pp. 219-222, Sep. 1996.

[9] C. T. Hsu and J. L. Wu, "Multiresolution Watermarking for Digital Images," *IEEE Trans. on Circuits and System II*, vol. 45, no. 8, pp. 1097-1101, Aug. 1998.

[10] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Robust Data Hiding for Images," *in Proc. of 1996 Digital Signal Processing Workshop*, pp. 37-40, Sep. 1996.

[11] J. Fridrich, M. Goljan, and D. Rui, "Invertible Authentication Watermark for JPEG Images," *in Proc. of 2001 International Conference on Information Technology: Coding and Computing*, pp. 223-227, Apr 2001.

[12] M. Noar, and A. Shamir, "Visual Cryptography," *in Porc. of Advances in Cryptology – Eurocrypt'94*, vol. 950, pp. 1-12, 1995.

[13] A. Shamir, "How to Share a Secret," *Communications of ACM*, vol. 22, pp. 612-613, 1979.

[14] C. C. Chang and I. C. Lin, "A New (t, n) Threshold Image Hiding Scheme for Sharing a Secret Color Image," *in Proc. of International Conference on Communication Technology*, vol. 1, pp. 196-202, April 2003.

[15] Y. C. Hou and P. M. Chen, "An Asymmetric Watermarking Scheme Based on Visual Cryptography," *in Proc. of 5th International Conference on Signal Processing*, vol. 2, pp. 992-995, Aug. 2000.

[16] D. C. Wu and W. H. Tsai, "Spatial-Domain Image Hiding Using Image Differencing," *IEE Proceedings on Vision, Image and Signal Processing*, vol. 147, no. 1, pp. 29-37, Feb. 2000.

[17] T. S. Chen, C. C. Chang, and M. S. Hwang, "A Virtual Image Cryptosystem Based upon Vector Quantization," *IEEE Transactions on Image Processing*, vol. 7, no. 10, pp. 1485-1488, Oct. 1998.

[18] Y. C. Hu, "Grey-level Image Hiding Scheme Based on Vector Quantisation," *IEE Electronics Letters*, vol. 39, no. 2, pp. 202-203, Jan. 2003.

[19] H. Q. Cao and W. Li, "A Fast Search Algorithm for Vector Quantization Using L_2-Norm Pyramid of Codewords," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 4, pp. 585-593, June 2000.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figure 1. The JPEG recommended quantization table.

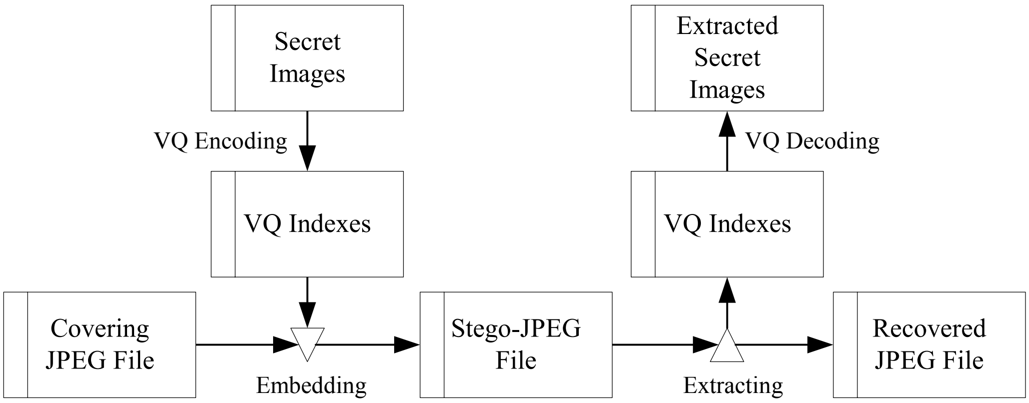

Figure 2. The concept of the proposed scheme.

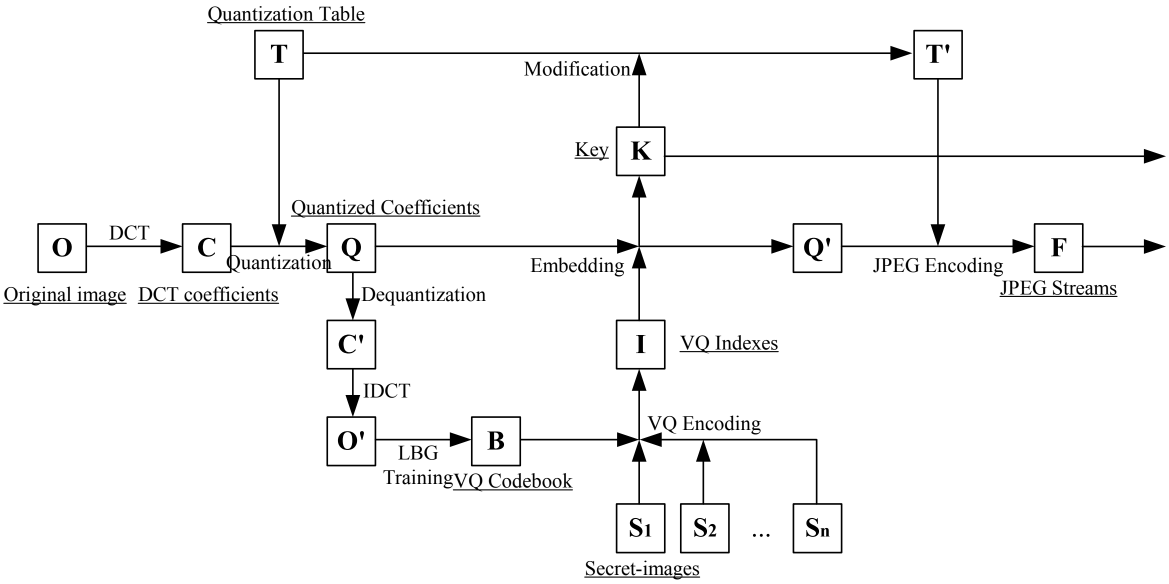

Figure 3. Detailed flowchart of secret image embedding.

80	55	50	80	120	200	255	255
60	60	70	95	130	255	255	255
70	65	80	120	200	255	255	255
70	85	110	145	255	255	255	255
90	110	185	255	255	255	255	255
120	175	255	255	255	255	255	255
245	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

(a)

4	0	1	4	3	3	0	0
2	2	1	0	1	0	0	0
1	0	4	3	3	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)

Figure 4. (a) An arbitrary quantization Table. (b) The corresponding table for data hiding. Each number indicates the embedding capacity of such position.

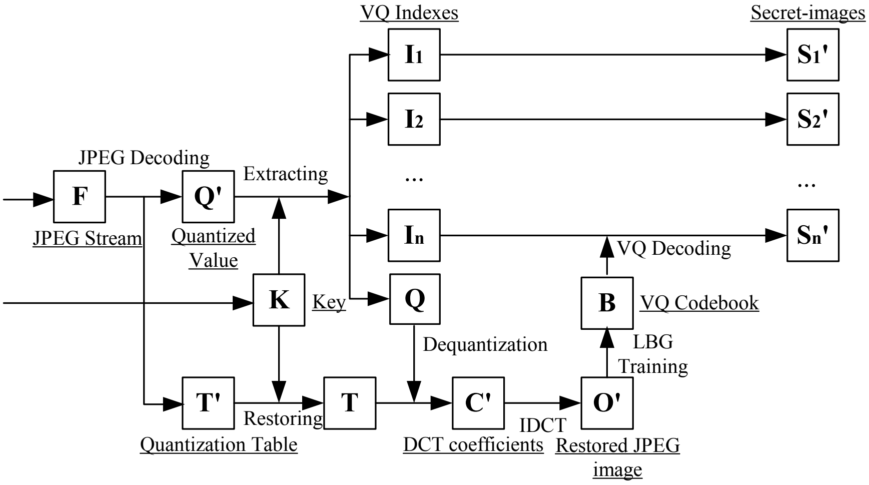

Figure 5. Detailed flowchart of secret image extracting.

Table I. Experimental results under different quality factors of cover image.

Codebook Size		64			256		
Quality Factor		20	50	80	20	50	80
Lena	Stego-image PSNR (dB)	26.75	34.19	37.26	23.31	32.25	36.19
	Stego-image Size (Byte)	39,614	45,411	58,871	54,061	52,708	68,319
	Secret *Lena* PSNR (dB)	28.58	28.49	28.48	30.36	30.21	30.29
Pepper	Stego-image PSNR (dB)	26.31	34.07	37.26	23.10	32.07	36.01
	Stego-image Size (Byte)	40,331	45,322	59,045	54,203	52,760	68,171
	Secret *Pepper* PSNR (dB)	28.14	28.10	28.09	29.90	29.78	29.82
Toys	Stego-image PSNR (dB)	25.24	33.40	36.62	22.20	31.31	35.22
	Stego-image Size (Byte)	40,261	44,501	57,488	54,613	52,857	68,064
	Secret *Toys* PSNR (dB)	24.12	24.07	24.10	26.54	26.18	26.24

Table II. Experimental results of embedding several secret images by the proposed scheme.

Codebook Size		64			256		
Quality Factor		20	50	80	20	50	80
Lena + *Pepper*	Stego-image PSNR (dB)	20.75	27.60	32.58	17.35	24.59	29.58
	Stego-image Size (Byte)	78,857	81,012	89,694	99,998	101,577	110,495
	Secret *Lena* PSNR (dB)	28.58	28.49	28.48	30.36	30.21	30.29
	Secret *Pepper* PSNR (dB)	28.14	28.10	28.09	29.90	29.78	29.82
Lena + *Pepper* + *Toys*	Stego-image PSNR (dB)	15.31	22.61	27.36	☒	☒	☒
	Stego-image Size (Byte)	111,240	112,301	121,163	☒	☒	☒
	Secret *Lena* PSNR (dB)	28.58	28.49	28.48	☒	☒	☒
	Secret *Pepper* PSNR (dB)	28.14	28.10	28.09	☒	☒	☒
	Secret *Toys* PSNR (dB)	24.12	24.07	24.10	☒	☒	☒

Table III. Experimental results of the proposed scheme based on "Scan-Order" and "Minimal-Value-First" strategies, respectively.

Quality Factor			50			
Codeword Dimension			4×4			
Codebook Size			64	256	1024	4096
Secre Lena	SO	Stego-image PSNR (dB)	34.19	32.25	30.30	27.67
		Stego-image Size (Byte)	45,411	52,708	65,632	80,517
	MVF	Stego-image PSNR (dB)	34.07	32.83	30.37	27.86
		Stego-image Size (Byte)	45,297	55,547	67,986	85,298
Secret *Pepper*	SO	Stego-image PSNR (dB)	34.07	32.07	30.27	27.55
		Stego-image Size (Byte)	45,322	52,760	65,622	80,860
	MVF	Stego-image PSNR (dB)	33.98	32.76	30.30	27.77
		Stego-image Size (Byte)	45,674	55,755	67,831	86,522
Secret *Toys*	SO	Stego-image PSNR (dB)	33.40	31.31	29.45	26.85
		Stego-image Size (Byte)	44,501	52,857	66,567	81,045
	MVF	Stego-image PSNR (dB)	33.45	32.00	29.47	27.02
		Stego-image Size (Byte)	44,747	55,638	68,928	87,213