A SUITABLE IMAGE HIDING SCHEME FOR JPEG IMAGES

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

An image hiding scheme suitable for JPEG compressed images is proposed in this article. The goal of this scheme is to obscurely deliver some secret images via a JPEG image file. Secret images are first encoded into binary indexes by vector quantization. Then, these indexes are embedded into the JPEG file of a cover image by modifying its quantized coefficients and quantization factors in an 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.

1. INTRODUCTION

Digital data have their commonality – the fidelity and plasticity among duplications. 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. 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.

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. The techniques of steganography have been significantly studied recently. These schemes can further be derived into two branches: digital watermarking and information hiding [1]. The former usually provides the protection of intellectual property, whereas the latter concerns the privacy of data contents.

In general, a watermarking scheme embeds recognizable signatures into the host media that have to be protected by directly modifying the media themselves. Besides, digital watermarking can also be used for tamper-proofing. No matter what applications they are, it is necessary for them to survive JPEG compression [2]. Therefore, most of the watermarking schemes will discuss their experimental results under JPEG compression. Moreover, there are 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 [3]. 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.

Information hiding can be used for covert communication. In the prior researches, raw images without any compression are considered as cover media. To solve the problems of inefficient capacity, Chen et al. proposed a solution that secret images should be compressed by VQ [4] and then encrypted before the embedding process, which is called virtual image cryptosystem [5]. Hu also proposes a revised algorithm of virtual image cryptosystem [6]. 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.

An image hiding scheme that based on JPEG compression standard and vector quantization is presented in this article. The details of the proposed image hiding scheme are introduced in Section 2. Section 3 demonstrates the simulation results and experimental analyses. Finally, conclusions are given in Section 4.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

2. THE PROPOSED IMAGE HIDING SCHEME

The details of the proposed scheme are 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. As for invertible property, we refer to the scheme proposed by Fridrich et al. [3].

In this 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. First, secret images are 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 flowchart of secret image hiding is shown in Fig. 1.

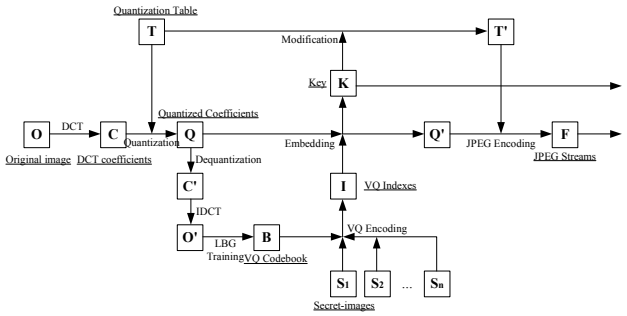


Figure 1. Detailed flowchart of secret image hiding.

2.1. 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 be determined. This is a characteristic of the proposed scheme since all information is hidden in the positions of even numbers among all quantization factors. The more quantization factors of factor 2, the more information we can embed.

2.2. Codebook Generation and VQ Encoding

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 is the standard JPEG encoding process, 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. In this scheme, we adopt the famous LBG algorithm [4] 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.

In the encoding step, secret images are VQ encoded by the codebook obtained 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 [7] 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.

2.3. 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. Fridrich et al. have proposed a feasible method [3] that modifies quantized coefficients and quantization factors concurrently. However, we only choose those positions where the quantization factor is even. Furthermore, we propose two strategies for space allocation. One is based on the zigzag scan order, and the other is decided in order of the magnitude of all quantization factors.

2.3.1. 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.

2.3.2. Minimal-Value-First Strategy (MVF)

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 the quantization factor is, the more distortion it introduces. If one emphasizes on the image quality of stego-media, he should adopts this strategy hypothetically.

2.4. 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.

2.5. 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 image size.
(d). Flag of embedding strategy.

2.6. Secret Image Hiding Algorithm

The summarized steps of the proposed image hiding scheme are listed below.

Step 1. Predefine a quantization table for cover image.
Step 2. Quantize and de-quantize the DCT coefficients of cover image by this quantization table.
Step 3. Train a codebook from the JPEG compressed cover image.
Step 4. Encode secret images with this codebook by VQ.
Step 5. Modify the DCT coefficients of cover image and the factors in the quantization table.
Step 6. Embed the VQ indexes of secret images into the cover image.
Step 7. Encode the final DCT coefficients into JPEG file streams by entropy encoder.

2.7. Secret Image Extracting Algorithm

The secret image extracting process is simple and similar to the inverse of image hiding algorithm. 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 codebook from the restored cover image.
Step 5. Decode secret images by VQ.

3. SIMULATION RESULTS

To measure the feasibility of proposed image hiding scheme, we have conducted a series of experiments. In our experiments, *F16* is adopted as cover image. Three secret images are *Lena*, *Pepper* and *Toys*. All of the images are with 256-gray-levels and 512×512 pixels.

Table I gives the experimental results of the proposed scheme under different JPEG quality factors for cover image and different codebook sizes for secret image. This table shows that the quality factor of cover image only affects the image quality and file size of stego-image. Considering the image quality of extracted secret images with the same codebook size, however, we can find that the JPEG 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. However, larger size of codebook implies more bits have to be embedded into cover image. And this further implies lower quality and larger file size of stego-image.

Codebook Size	64		256	
Quality Factor	50	80	50	80
Stego-*F16* (dB)	34.19	37.26	32.25	36.19
File Size (byte)	45411	58871	52708	68319
Secret Lena (dB)	28.49	28.48	30.21	30.29

Table I. Experimental results under different JPEG quality factors and different codebook sizes.

Table II shows the results of the proposed scheme where 2 and 3 secret images are embedded, respectively. There are some blanks left in this table because the embedding capacity is not enough in the corresponding condition (i.e. the 3 secret images are coded with a codebook of size 256 and have to be embedded into the cover image *F16*). However, we can directly modify quantization factors to the nearest even number to increase the embedding capacity if needed. The architecture of the proposed scheme is similar with Hu's scheme [6]. 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.

Codebook Size	64		256	
Quality Factor	50	80	50	80
Stego-*F16* (dB)	27.60	32.58	24.59	29.58
File Size (byte)	81,012	89,694	101577	110495
Secret Lena (dB)	28.49	28.48	30.21	30.29
Secret Pepper (dB)	28.10	28.09	29.78	29.82
Stego-*F16* (dB)	22.61	27.36		
File Size (byte)	112301	121163		
Secret Lena (dB)	28.49	28.48		
Secret Pepper (dB)	28.10	28.09		
Secret Toys (dB)	24.07	24.10		

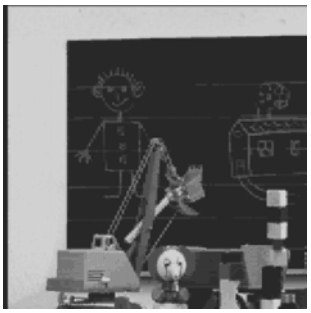
Table II. Results of embedding several secret images into a JPEG compressed cover image.

To illustrate, Fig. 2 shows the carrier image after JPEG compression and the embedded images after recovery. Note that, the quality factor of JPEG compressed cover image is 80 and the size of codebook is 64. It demonstrates that the visual quality of extracted secret images is good while the existence of them is imperceptible to human visual system.

(a)

(b)

(c)



(d)

Figure 2. (a) The stego-*F16*; (b)-(d) the extracted secret images *Lena*, *Pepper* and *Toys*.

4. CONCLUSION

An image hiding scheme for 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 secret images, one can completely remove the extra information and restore the original JPEG image. The proposed scheme may be useful for secret communication. It can also be considered as a brand-new compression method that multiplexes several images into one JPEG image.

5. ACKNOWLEDGEMENTS

The authors are supported by NDHU-ROC and NSC-ROC with project no. NSC 92-2213-E-259-018.

6. REFERENCES

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

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

[3] J. Fridrich, M. Goljan, and D. Rui, "Invertible Authentication Watermark for JPEG Images," *Proc. of 2001ICIT: Coding and Computing*, pp. 223-227, 2001.

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

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

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

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