INTERNATIONAL JOURNAL OF
INFORMATION AND SYSTEMS SCIENCES
Volume 3, Number 3, Pages 431-448

A CODEBOOK-BASED IMAGE-IN-IMAGE TECHNIQE FOR JPEG FILE STORAGE[1]

SHIH-CHANG HSIA AND I-CHANG JOU

Abstract Image-in-image techniques have been widely used for watermarking, data hiding and multi-windows applications. However, a hidden image produces very serious distortions after JPEG compressing. In this study, a high-performance hiding algorithm for JPEG format storage is presented based on the frequency dispersal concept. The main technologies involve (1) the frequency codebook processing, (2) JPEG domain processing. The privacy key for a hiding image can be found from the codebook scheme. For JPEG compression, the privacy key is further hidden into the bit-stream of JPEG domain without increasing bit rate. Simulations demonstrate that this technique can achieve superb quality for the source image and hiding images after codebook processing scheme. Moreover, the hiding image can be efficiently retrieved from a single JPEG file without any additional information required.

Key Words. JPEG compression, watermark, frequency, codebook, privacy key

1. Introduction

The digital multimedia data, such as image, audio, and video etc. , can be easily accessed and distributed over Internet systems nowadays. The copyright protection for the multimedia data becomes a major concern. Recently, an image-in-image technique [1-5] has been developed for watermarking and data hiding systems to protect the valuable data. Moreover, this technique also can provide the function of multi-window for multimedia systems. In order to reduce the size of image file, JPEG coder has become a common tool for image compression, so any hiding technique proposed must have the capability to resist JPEG washing for real applications. Because of using the spatial domain processing [6-7], the hidden image would be completely discarded after JPEG compression. Instead, the frequency domain processing [8-11] is employed to enhance the robustness for JPEG processing [12]. Intuitively, the high frequency area is a good selection to attain better hiding result. However, the spectrum energy of image is concentrated at the low frequency band. Due to quantization required, the high frequency components would be discarded for the spatial redundancy removal after JPEG compression. Therefore, the hidden image would produce very serious distortions as extracting from a JPEG file.

In order to stand JPEG compression, the image can be embedded into the low frequency band. Nevertheless, this approach would produce visible distortion. Instead, authors [13] suggest to select the middle frequency part of the 8×8 DCT domain as an embedded target. Although this method can provide higher robustness, the middle frequency components are also eliminated a little after JPEG processing and the

[1] This work was supported by the National Science Council, Taiwan ROC, NO. NSC89-2213-E-327-024.
Received by the editors and in revised form December 21, 2006

reconstructed image also produces some distortion. So this approach cannot provide a good rule to trade off the hiding quality and the robustness for JPEG compression. Recently, the bit-stream processing [14-16] is used to enhance the hiding performance. But the extra data is required to compensate the quality drift, so the coding bit-rate shall be increased.

How to accomplish a high performance image-in-image using JPEG compression without knowledge of the original image to extract the hiding information is studied in the paper. At first, the embedding process is based on the frequency codebook scheme. The hiding information is randomly dispersed into the original source image with frequency distribution processing to reduce the spatial correlation. Then the secret key can be obtained to provide higher security for data protection. For JPEG format storage, the bit-stream domain processing is developed to reduce the visible distortion and to enhance the robustness. The paper is organized as follows. In Section II, the proposed image-in-image algorithm about the embedding process and the extracting process is presented with codebook concept. The second hidden approach on JPEG domain is described processing in Section III. The simulated results and comparisons are addressed in Section IV. Finally, some conclusions are draw in Section V.

2. The Codebook-Based Processing

For image-in-image systems, we have a source image (or called an original image) and a destination image (or called a watermarking or hiding image). As the destination image is embedded to the source image, the resulted image is called a composition image (or called a watermarked image). For the purpose of image watermarking or hiding, the destination image that is a secret data must be hidden into the source image. The destination image can be retrieved from the composition image in the decoder. For practical applications, there are three basic requirements: (1) the destination image invisible after embedding; (2) high robustness for the destination image as the composition image suffering from attacks; (3) high security for the destination image.

2.1 The Embedding Process

In the current signal transformations, DCT (discrete cosine transform) [18-19] has become very popular since many coding standards had used. In our approach, both the source image and the destination image are transformed into DCT domain. As the frame size of the source image f_{jk} is $M1 \times N1$, the entire frame transformation is performed. The DCT coefficients can be obtained from

$$(1) \qquad F_{u1v1} = FDCT(f_{j\,k})_{M1\times N1} = \begin{pmatrix} F_{00} & F_{01} & F_{02....} & F_{0(v1-1)} \\ F_{10} & F_{11} & ..... & F_{1(v1-1)} \\ ..... & & & \\ F_{(u1-1)0} & ..... & F_{(u1-1)(v1-1)} & \end{pmatrix}_{u1\times v1}$$

where u1 and v1 are the frequency indexes for DCT coefficients. Also, the full destination image is also transformed using DCT as

$$(2) \qquad D_{u2v2}=FDCT(d_{jk})_{M2\times N2},$$

wherein its image size is $M2 \times N2$. To reduce the visible distortion, the high frequency part of F_{u1v1} is a good choice to insert the destination image since the great variation in high frequency makes the hiding data much more difficult to detect. However, when the destination image is embedded to the high frequency part, most of the information

for the destination image should be discarded after JPEG compression since high frequency components are the spatial redundancy.

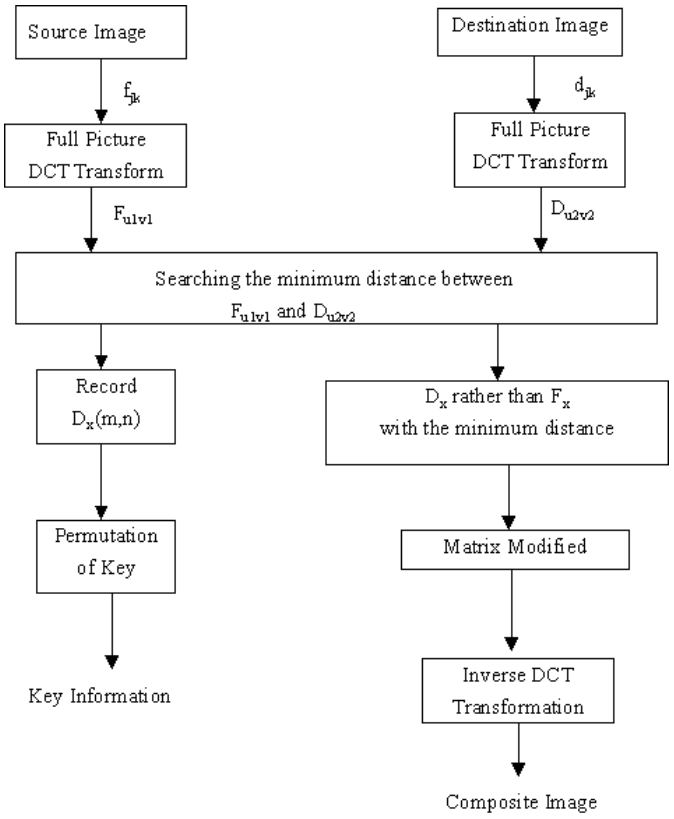

Fig. 1 The proposed embedding flowchart with the codebook scheme

The embedding procedure of the proposed algorithm is illustrated in Fig. 1. A frequency codebook scheme is proposed to strengthen the robustness for JPEG processing. The matrix (1) can be treated as one codebook content. As one destination coefficient is embedded to the source image, we search the minimum differential value between the coefficient and the codebook matrix by

(3) $(Diff)_{min} = Min| D_x - F_{u1v1} |, \quad for\ u1,v1=0\ to\ n1,m1,$

where D_x is the one of D_{u2v2} coefficient. After $n1 \times m1$ points comparing, we can find that D_x has the best match in the codebook matrix F_{u1v1}. Then the D_x coefficients can be embedded into (1) according to the minimum difference location with

(4) $(F_x)_{\in Diff(min)} = D_x,$

where F_x is the one of F_{u1v1} coefficient. Simultaneously, the matrix coordinate in (1) is recorded for the coefficient D_x. One-by-one coefficient embedding, until all D_x coefficients are completely processed. It is noted that as the coefficient of matrix (1) has been replaced with the previous D_x coefficient, the coefficient is not changed again even if the next coefficient is the best match with the current coordinate, where only the position is recorded for the next coefficient. Based on this codebook concept, each coefficient of the destination image can be mapped to the relative coordinate of the

matrix (1). The codebook vector can be presented by using a set of the coordinate position. We use the vector $D_x(m,n)$ to denote the coefficient D_x located at the $(m,n)^{th}$ coordinate. The $D_x(m,n)$ is an important information to restore the destination image for the decoder and it can be treated as a privacy key (*PK*) [20] that is defined by

(5) $$PK=\{M_1,M_2 \ldots M_{n2\times m2}\}, \quad Mx \subset D_x(m,n),\ x=1 \text{ to } n2\times m2,$$

where *PK* is a set of coordinate. Furthermore, these privacy keys can be re-permuted by using a random function to disperse the spatial correlation and to increase the security level. Now the new key becomes

(6) $$P(p,q)=Permute(PK),$$

where *permute( )* is the function of permutation [20]. After the embedding process, the new matrix can be presented by mixing the coefficients of the source image and the destination image, where an example is given as

(7) $$\tilde{F}_{u1v1}=\begin{pmatrix} F_{00}\ D_{31}F_{02}\ F_{03}\ D_{12}\ F_{05}\ \ldots\ldots \\ D_{21}\ D_{35}F_{12}\ F_{13}\ D_{14}\ F_{15}\ \ldots\ldots \\ F_{20}\ F_{21}D_{62}\ F_{23}\ F_{24}\ D_{02}\ \ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots \end{pmatrix}_{u1\times v1}$$

where some D_x coefficients are instead of some F_x coefficients. Finally, the DCT coefficients are converted into the spatial domain using inverse DCT transformation with

(8) $$\tilde{f}_{jk}=IDCT(\tilde{F}_{u1v1}),$$

to attain a composite image.

2.2. The Destination Image Extraction

In order to restore the destination image, the system key uses the same parameter in the encoder and the decoder. In our approach, the system key contains two important parameters. One is the privacy key, and the other is the permutation function. The permutation function could be fixed in the extracting program, so only the privacy key is used as an input parameter.

The extracting process is the inverse operation of embedding one. Fig. 2 indicates the processing steps. When the composite image and key parameters are received, the image should be transformed by DCT from

(9) $$\tilde{F}_{u1v1}=DCT(\tilde{f}_{jk})_{n1\times m1}.$$

Because the key order is permuted in (6), the privacy key must be re-permuted from

(10) $$PK=re\text{-}Permute(P(p,q)).$$

According to the privacy key, the coefficient could be found from the matrix (9). The coefficients of the destination image are sequentially extracted according to the privacy key and the composition image. Then the coefficient matrix $\hat{D}_{uv}$ can be restored. Finally, the destination image could be reconstructed from inverse transform of $\hat{D}_{uv}$.

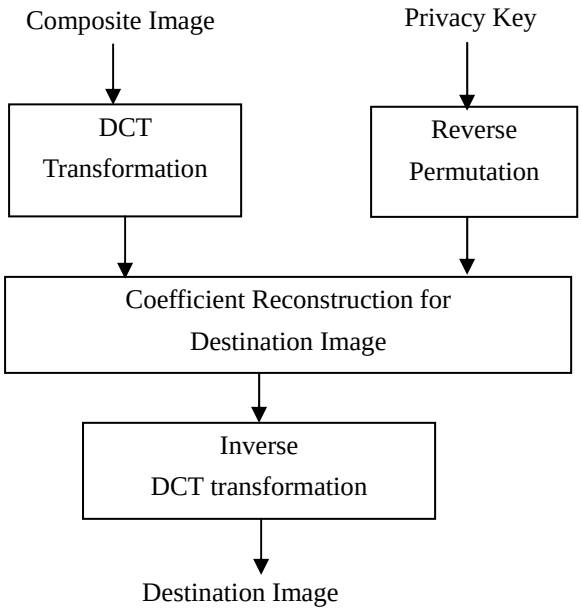

Fig. 2 The processing flow for the destination image extraction

3. Composition image storage with JPEG format

Currently, JPEG compressor has been widely used to reduce the image size. However, most of the destination image is discarded after JPEG processing as well. In order to overcome this drawback, we study a high robustness algorithm for JPEG processing.

3.1 Privacy key hidden on JPEG domain

In our proposed method, the codebook scheme is used in the first stage. The composite image and the privacy key are required to send to the receiver. Because the privacy key is a secret data, in order to keep high secrecy, the privacy key can be further hidden on JPEG file in the second stage. As the composite image is compressed by JPEG, the privacy key can be added to the attachment of JPEG file. Fig. 3 shows the output format, where the privacy key becomes the appendix data of JPEG bit stream. For image compressing, most of the high frequency components would be discarded in order to reduce the spatial redundancy. As the destination image is embedded into the high frequency parts of the source image, most of information should be lost forever. Fortunately, our approach uses a random distribution for coefficients embedding based on the codebook procedure, so high robustness for the destination image is achieved to resist JPEG washing.

When the JPEG file contains the privacy key, the file size would become

$$Size_{JPEG} = \frac{M1 \times N1}{CR} + PK \,, \tag{11}$$

where the *CR* is the JPEG compression ratio. The size of JPEG file becomes large while the privacy key is inserted into the JPEG file. In order to avoid the coding bit-rate increased, the second hidden approach is presented.

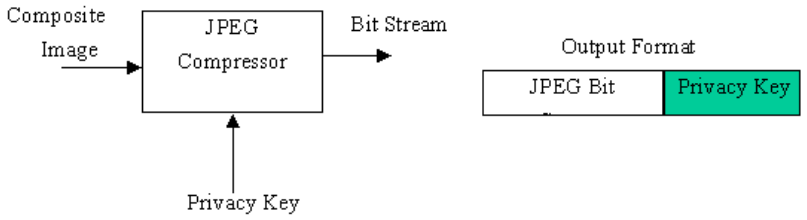

Fig. 3 Privacy key hidden into the appendix of JPEG bit

3.2 The second layer hidden On JPEG domain

In the JPEG compressor [12], the image is partitioned into 8×8, and then is transformed by DCT processing. These DCT coefficients are quantized using a particular quantization table, which can be expressed as

$$\hat{F}_{uv} = \frac{DCT(\tilde{f}_{jk})_{8\times 8}}{Q_{uv}}, \tag{12}$$

where Q_{uv} is the quantization level that is dependent on the u and v components, $\tilde{f}_{jk}$ is a composite image that is attained from (8), and $\hat{F}_{uv}$ is the final quantized coefficient. For RLC/VLC package, the coefficient $\hat{F}_{uv}$ is truncated into an integer. In fact, the LSB bit of each coefficient is not exact after the truncation processing. Thus the compression performance is not changed too much as the privacy key information is embedded into LSBs of non-zero coefficients. With this concept, we first search all non-zero coefficients. Then the LSB of non-zero coefficient is modified by one-bit of the privacy key, which is given by

$$\begin{cases} If\ (\hat{F}_{uv})\ \ is\ odd, \text{then}\ (\hat{F}_{uv})_{LSB} = 0, as\ (Key)_{bit} = 0 \\ If\ (\hat{F}_{uv})\ \ is\ even, \text{then}\ (\hat{F}_{uv})_{LSB} = 1, as\ (Key)_{one\ bit} = 1 \\ Else\quad (\hat{F}_{uv})_{LSB}\ No\ Change \end{cases}, \qquad \text{as } \hat{F}_{uv} \neq 0 \tag{13}$$

The information of the privacy key is sequentially inserted into the LSB location of each non-zero coefficient with bit-by-bit approach. Since only LSBs of non-zero DCT coefficients are modified, the compression ratio is not affected.

Fig. 4(a) illustrates the privacy key embedded into non-zero coefficients on the JPEG encoder. Fig. 4(b) illustrates the JPEG decoder system block for the destination image extracting. From the JPEG file, the composite image is reconstructed after JPEG decoder. Simultaneously, the privacy key is extracted from each non-zero coefficient. Then the composite image is transformed into DCT domain. According to the privacy key, the coefficients of the destination image can be extracted. Finally, the destination image can be restored from inverse DCT processing.

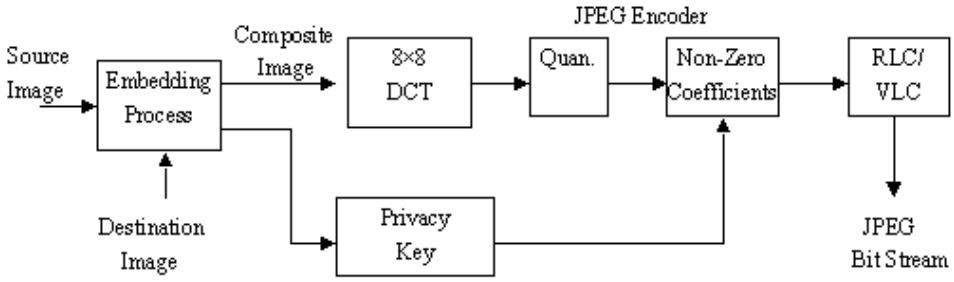

Fig. 4(a) Embedding the privacy key into JPEG domain.

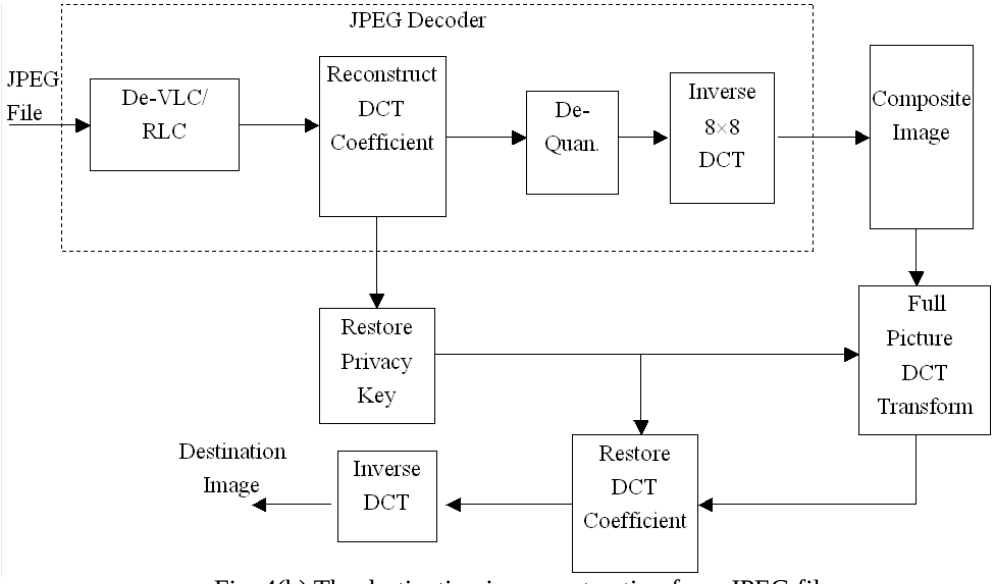

Fig. 4(b) The destination image extracting from JPEG file

From the above mentions, how much the key information embedded has to be evaluated. The total bit of the key information is $n2 \times m2 \times \log_2(n1 \times m1)$ since there are n2×m2 keys and each key with $\log_2(n1 \times m1)$ bits. After JPEG processing, the number of non-zero coefficients (NNZC) in the full frame is computed by

(14) $If\ \hat{F}_{uv}^{i} \neq 0, \text{ then } NNZC = NNZC + 1, \quad i = 1\,to\,(n1/8) \times (m1/8),\ u, v = 0\,to\,7$

, where $\hat{F}_{uv}^{i}$ denote the quantized coefficient at the (u,v) location for the i^{th} block, and the composite image is split into (n1/8)×(m1/8) blocks as a block size is 8×8. To embed the key information completely, the following equation

(15) $NNZC \geq n2 \times m2 \times \log_2(n1 \times m1)$,

needs to be satisfied. Otherwise, some key information would be discarded, and the extracting quality will degrade accordingly. In fact, the *NNZC* becomes smaller after quantization. Eq. (15) is seldom satisfied in practical case, except the destination image

size is further reduced or the source image size is further enlarged, but the hiding system shall become no efficiency. For practical applications, we present three scanning approaches as below.

3.2.1 Sequential procedure

Since DCT transformation has a progressive feature, each coefficient implies one spatial resolution. The destination image can be restored using only a few DCT coefficients. The sequential procedure is shown in Fig. 5(a), where the key information is scanned to insert JPEG bit-stream with block-by-block processing. Until all non-zero coefficients are completely processed, the inserting procedure is stopped. Because the amount of the key information is much larger than that of non-zero coefficients, the extracting quality is not good generally. Intuitively, we can increase the number of non-zero coefficients to improve the image quality, but this scheme will reduce the compression ratio. Instead, the high-efficiency scanning styles are presented.

3.2.2 Zigzag style for LL band Scan

Because most of the image energy is compacted into the low frequency band in the DCT domain, the key information located at the low frequency area is first considered to embed. Based on a sub-band concept [21], the coefficients can be split into LL(low-low), LH(low-high), HL(high-low) and HH (high-high) bands. In order to keep a regular processing flow, only LL band information is imbedded by using the zigzag scan as shown in Fig. 5(b). The zigzag scan looks like as the run length coding within JPEG compressor, the scanning direction is from left-top corner to right-bottom for the LL band only.

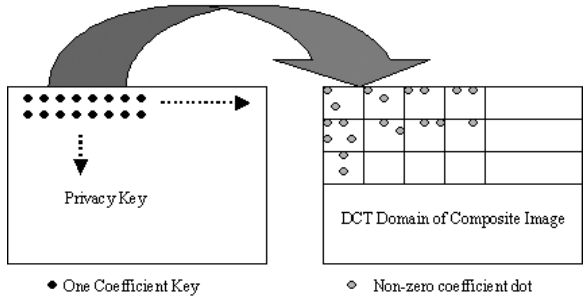

Fig. 5(a) Privacy keys sequentially insert into non-zero coefficients

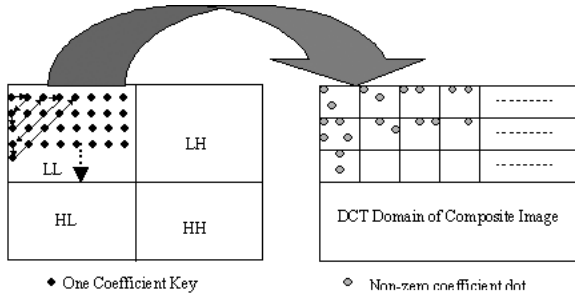

Fig. 5(b) Key insertion with the zigzag scan for LL band

3.2.3 Maximum weight approach for LL band

Since the DCT coefficients are independent, the image can be approximately reconstructed from inverse transformation with coefficient-by-coefficient approach. As the coefficient value is larger, its weight is larger to affect the reconstructed result. To improve the image quality, the key position corresponding to a larger coefficient is first inserted. Because the coefficients are random distribution on the codebook, one-bit flag is required to record the current coefficient whether or not to be selected. For LL band processing, there are n2/2×m2/2 bits used to mark the available coordinate. The number of maximum weight coefficient (NMWC) recorded for the key processing can be computed as

$$NMWC = \frac{NNZC - (n2/2 \times m2/2)}{\log_2(n1 + m1)} \quad , \tag{16}$$

The processing procedures using the maximum weight scan are:

1. Pre-computing *NNZC* and *NMWC*.
2. Find the maximum coefficient and record its corresponding key position.
3. The one-bit flag is set to high, and counter=counter+1.
4. The next maximum coefficient is processed with repeating (b)-(c), until the counter = *NMWC*.

Although we sacrifice one bit space to record the location of maximum weight coefficients, the larger weight coefficients used can improve the inverse transformation result.

As previously mentioned, we had used two-layer DCT domain hidden. First the DCT coefficients are hidden on the frequency codebook. The codebook vector is treated as a privacy key. Then the privacy key is further hidden into DCT coefficients in the bit stream of JPEG domain. This approach can meet three requirements for the image hiding systems: invisible, robustness and security. The destination image becomes invisible since its coefficients are embedded with the best match from the codebook content. Our approach can provide a high robustness for JPEG compressing because of the random coefficient distribution over the entire image. With double DCT layer hidden, a high security for the destination image protection is achieved.

4. Simulations and performance evaluations

To evaluate the performance of the proposed algorithm, some pictures are employed as test patterns for our simulations. To meet the practical applications, the frame size of the source image and the destination image adopts 256×256 and 64×64 respectively.

4.1 Simulations with codebook scheme

Fig.6 (a) and 6(b) show the source image “scenery” and the destination image “baloon” respectively. Fig. 7(a) shows the embedded result from a codebook embedding scheme, where PSNR achieves 51dB. The embedding quality is very high, it is very difficult to recognize the difference between Fig. 6(a) and 7(a). Table I shows the evaluation results using five pictures. The average PSNR and MSE value is about 51dB and 0.5 respectively, where

$$PSNR = 10\log\frac{\sum_{j=0}^{M-1}\sum_{k=o}^{N-1} 255^2}{MSE} \quad and \quad MSE = \frac{\sum_{j=0}^{M-1}\sum_{k=o}^{N-1} (f_{jk} - \hat{f}_{jk})^2}{M \times N} \quad , \tag{17}$$

f_{jk} and $\hat{f}_{jk}$ is the source image and the composite image respectively, and the

frame resolution is $M \times N$. From Table I, we could find that the embedded results are almost independent on the source image. Hence the proposed algorithm is capable to provide a high quality for data hiding.

Fig. 6(a) Source Image

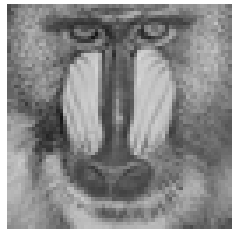

Fig. 6(b) Destination Image.

Fig. 7(a) The result with codebook scheme.

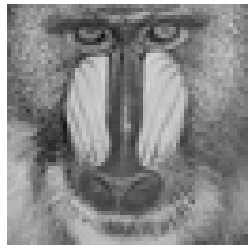

Fig. 7(b) Extracting from Fig.7(a).

When users obtain the composite image and the privacy key, the destination image can be restored from the decoding program. The extracted image is shown in Fig. 7(b) that is extracted from Fig. 7(a). The image quality is superb, where its PSNR achieves about 48dB. Moreover, the five pictures are simulated, where the results are shown in the right side of Table I. The average PSNR and MSE is about 46dB and 1.6 respectively, so we can provide a reliable destination image for users.

Table I. The Embedding Results and Extracting Results using Codebook Method

	Embedding Results		**Extracting Results**	
	MSE	**PSNR**	**MSE**	**PSNR**
Lena	**0.501**	**51.133**	**2.083**	**44.943**
Lady	**0.498**	**51.162**	**0.830**	**48.938**
Miss	**0.495**	**51.185**	**2.163**	**44.781**

Bloom	**0.501**	**51.136**	**2.135**	**44.837**
Scenery	**0.497**	**51.166**	**1.054**	**47.901**
Average	**0.498**	**51.156**	**1.653**	**46.280**

Table II. The Codebook Simulation Results after JPEG Processing

	Embedding Results		**Extracting Results**	
	MSE	**PSNR**	**MSE**	**PSNR**
Lena	**37.469**	**32.394**	**52.345**	**30.942**
Lady	**23.632**	**34.396**	**40.998**	**32.003**
Miss	**20.503**	**35.013**	**40.195**	**32.089**
Bloom	**16.399**	**35.983**	**35.021**	**32.687**
Scenery	**78.138**	**29.202**	**88.292**	**28.672**
Average	**35.228**	**33.398**	**51.370**	**31.279**

4.2 Simulations for JPEG processing

Fig.8 (a) shows the composite image after JPEG encoding and decoding in our approach. The image produces some blocky effect [24-25], where PSNR value is about 29dB as using the default JPEG quantization table (the compression ratio is about 15). From the JPEG file, the privacy key can be extracted from the appendix data. Simultaneously, the composite image can be reconstructed from JPEG de-compression. According to the privacy key, the destination image can be restored. The result is shown in Fig. 8(b) that PSNR is about 29dB. Table II shows the embedded and extracted results from five pictures, the averaged PSNR achieved about 33 dB and 31 dB for the composite image and the destination image respectively.

Fig. 8(a) After JPEG compression

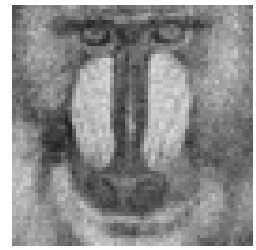

Fig. 8(b) Extracting from Fig.8(a)

4.3 Simulations with double DCT hidden for JPEG file storage

The information for the privacy key has 64×64×16 =65536 bits because the destination image size is 64×64. The number of non-zero coefficients is computed from the composite image after JPEG compression in the experiment, and the results are shown in Table III. The averaged NNZC is about 8795. The ratio of the privacy key insertion (RPKI) is estimated by NNZC/65536. The averaged ratio is only 13% as using the current picture formats. Three kinds of scanning modes are simulated as below.

Table III. The Number of Non-Zero Coefficients Evaluations

	NNZC	RPKI *
Lena	**8961**	**13.67%**
Lady	**6733**	**10.27%**
Miss	**7973**	**12.17%**
Bloom	**7057**	**10.77%**
Scenery	**13252**	**20.22%**
Average	**8795**	**13.42%**

*** defined by (NNC/65536)×100%**

4.3.1. Sequential scan : Fig. 9 (a) and (b) respectively show the experimental results for the reconstructed source image and the extracted destination image from a JPEG file. The extracting quality is not good because only 10%~20% keys can be restored. Table IV shows the results of the embedding and the extracting processes. The averaged PSNR can achieve about 29dB for the composite image. But the extracted destination image is only about 23dB in average.

Fig. 9(a) With sequential scan result

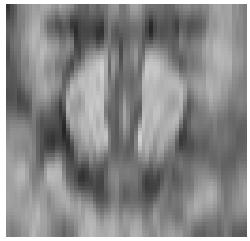

Fig. 9(b) Extracting from Fig.9(a)

Table IV. Simulation Results of using Sequential Mode

	Embedding Results		Extracting Results	
	MSE	**PSNR**	**MSE**	**PSNR**
Lena	**50.232**	**31.121**	**260.796**	**23.968**
Lady	**100.955**	**28.090**	**352.282**	**22.662**
Miss	**29.674**	**33.407**	**273.511**	**23.761**
Bloom	**166.532**	**25.916**	**511.650**	**21.041**
Scenery	**125.052**	**27.160**	**230.567**	**24.503**
Average	**94.489**	**29.139**	**325.761**	**23.187**

4.3.2. Zigzag scan for LL band : the embedded and the extracted images are shown in Fig. 10(a) and (b) respectively. And Table V shows the averaged results with five images. The extracting quality can be improved a little, where PSNR is about 25dB in average.

Fig. 10(a) With Zigzag scan result

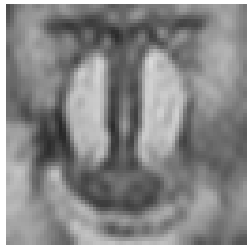

Fig. 10(b) Extracting from Fig. 10(a)

Table V. Performance Improvement with Zigzag Scan for LL Band

	Embedding Results		Extracting Results	
	MSE	**PSNR**	**MSE**	**PSNR**
Lena	**48.817**	**31.245**	**185.927**	**25.927**
Lady	**80.341**	**29.081**	**257.251**	**24.027**
Miss	**28.868**	**33.527**	**204.507**	**25.024**
Bloom	**171.084**	**25.799**	**271.921**	**23.786**
Scenery	**123.409**	**27.217**	**174.747**	**25.707**
Average	**90.504**	**29.374**	**218.871**	**24.894**

4.3.3. Maximum weight scan: Table VI shows the results using the maximum weight approach, the PSNR can achieve about 26dB in average. The performance is better than the sequential mode and the zigzag mode. The embedded image and the extracted image are shown in Fig. 11 (a) and (b) respectively. The extracted image resolution is higher than Fig. 9(b) and 10(b).

Table VI. Simulation Results of Using Maximum Weight Coefficients

	Embedding Results		Extracting Results	
	MSE	**PSNR**	**MSE**	**PSNR**
Lena	**50.218**	**31.122**	**143.397**	**26.565**
Lady	**92.672**	**28.461**	**198.875**	**25.145**
Miss	**28.906**	**33.521**	**148.047**	**26.427**
Bloom	**161.468**	**26.050**	**211.188**	**24.884**
Scenery	**100.215**	**28.121**	**142.843**	**26.582**
Average	**86.696**	**29.455**	**168.870**	**25.921**

Fig. 11(a) With maximum weight scan

Fig. 11(b) Extracting from Fig. 11(a)

4.4 Comparisons with other methods

Table VII lists the results for comparisons of the proposed and other methods. The embedding ratio (*ER*) is an important parameter to evaluate the system performance, which is defined by *ER=MD/MS*, where *MD* and *MS* is an amount of the destination image and the source image respectively. When *ER* is low, any method will obtain a good performance since the source image has been modified only slightly. For this purpose, most algorithms choose to use a “binary logo” or a “text number”, rarely a gray-level image since its high information content shall raise the ER and hence the hiding quality will be degraded accordingly. In our simulations, the gray-level image uses 8-bit and the ER is 0.0625 that is the highest as comparisons with other methods. As for the destination image extraction, the one-layer processing (the codebook scheme) requires the extra key information to send the decoder. Also we proposed two layer

processing (with the maximum weight mode) to achieve higher security. We can efficiently restore the destination image from one JPEG file without requiring the source image, where the PSNR individually achieved 31dB and 26dB using one-layer and two-layer processing. But other approaches mostly fail from the JPEG file for the destination image extracting.

In our approach, the destination image is transformed by using the entire image, and the coefficients are randomly dispersed into the composite image according to the best match. Due to the fact that DCT coefficients have the progressive feature where one coefficient represents one spatial resolution, the resolution of the image shall become better and better as the computing coefficients are successively accumulated into the transformed kernel. With the progressive transformation, the resulting watermark only suffers different degree of blurring but no serious distortions. Therefore, our algorithm can resist any data compression as long as the fundamental frequency feature of the composite image does not changed drastically after processing.

Table VII Comparisons of the Proposed Method and Existed Algorithms

	Hsu [13]	Hsu [22]	Kim[23]	Kim[24]	Proposed
Source Image Size	256×256 (8bit)	256×256 (8bit)	256×256 (24bit)	256×256 (8bit)	256×256 (8bit)
Destination Image Type	Binary Logo	Binary Logo	Binary Logo	Number	Gray Image
The number of hiding data (bit based)	16384	16384	16384	9000	32768
Embedding Ratio	0.0313	0.0313	0.0104	0.017	0.0625
Extraction with Extra Keys	Yes	Yes	Yes	No	Yes (one layer) No (two layer)
Extraction with Source Image	Yes	Yes	No	Yes	No
Processing Domain	DCT	Wavelet	FFT	Wavelet	DCT
Security Level	Middle	Low	Middle	Low	High
Extracting Quality From JPEG File	14dB	11dB	10dB	17dB	31dB(one-layer) 26dB(two-layer)

4.5. Discussions

The papers [11,22-23] use an original image as reference for the hiding data extraction. From comparisons of the source image and the composite image, their performance should become better than that of without using a source image. However, the extra disk is required to store the source image in the decoder. Moreover, the hackers can extract the destination image using a statistical computation from the difference of the source image and the composite image. So the security maybe becomes poor using this approach. In order to avoid this drawback, our scheme employs the privacy key rather than the source image in the extracting process.

In summary, there are some features of the proposed algorithm. (1) Two-layer DCT domain processing is used. At first, the DCT coefficient is hidden using the codebook scheme. Then the privacy key is further hidden on the JPEG domain. Since the hiding quality is very high, the attacker is hardly to detect whether the image embeds a destination image. Due to double DCT domain processing, our approach can provide higher security. (2) The destination image can be completely extracted from the composite image without using the source image. Because we do not require pre-storing the source image in the decoder, the system file size is smaller in our approach. (3) Three scanning modes are proposed to solve the problem of bit-stream domain hidden for JPEG format storage. Although NNZC is not enough for the key inserting, the maximum weight scheme can provide the best quality for the destination image extracting. Thus the composition image can be stored with a JPEG file to reduce the file size. The decoder can restore the destination image from a single JPEG file without using the extra information required.

5. Conclusions

In this paper, an efficient image-in-image technique is proposed using the codebook concept consisting of JPEG domain processing. In the first layer, the codebook scheme is used to hide the destination image into the source image. For user convenience, the key information can be further hidden on the bit-stream of JPEG domain, so we can extract the destination for a single JPEG file without extra information required. With the double DCT hidden technology, our approach can achieve high security. It is very hard to extract the destination image using a non-normal extracting process. Simulations demonstrate that a high-quality composite image is achieved, and the destination image can be completely extracted with various approaches, where the resulting image only suffers different degree of blurring but no serious distortions. Therefore, the proposed algorithm can provide high security, strong robustness and superior hiding quality for the image data protection.

REFERENCES

[1] Cox, I. J., Kilian, J., Leighton, F. T. ,and Shamoon, T. (1997) ” Secure spread spectrum watermarking for median”, *IEEE Trans on Image Processing*,Vol.6, No.12, pp.1673-687, Dec..

[2]. Bender, W., Gruhl, D., and Morimoto, N.(1995)” Techniques for data hiding “, in *Proc. SPIE*, Vol.2420, Feb. p.40~43,

[3]. Podilchuk, C. I. ,and Zeng, E. (1998) ” Image adaptive watermarking visual model”, *IEEE Journal on Selected Areas in Communications*, Vol.16, No.4, pp.525-539, May.

[4]. Sapwater, E., and Wood, K. (1994) ” Electronic copyright protection”, *Photo Electronic Imaging*, Vol.37, No.6, pp.16-21.

[5]. Mintzer, F., Braudaway, G.W., and Yeung, M. (1997) ” Effective and ineffective digital watermarks”, *Proc. of the IEEE International Conference on Image Processing*, pp.9-12.

[6]. Cox, I.J., and Miller, M. L. (1997) ” A review of watermarking and the importance of

perceptual modeling", *Proc. of the SPIE International Conference on Human Vision and Electronic Image II*, USA, pp.92-99.

[7] Wolfgang, R. B., and Delp, E. J. (1996) " A watermark for digital images", *Proc. of the IEEE International Conference on Image Processing*, pp219-222.

[8]. Piva, A., Barni, M., Bartolini, F., and Cappellini, V. (1997) " DCT –based watermark recovering without resorting to the uncorrupted original image", *Proc. of the IEEE International Conference on Image Processing*, pp.520-523.

[9]. Bors, A. G., and Pitas, I. (1997) " Image watermarking using DCT domain constraints", *Proc. of the IEEE International Conference on Image Processing*, pp231-234.

[10] Zhu, W., Xiong, Z., and Zhang,Y. Q. (1999) " Multi-resolution watermarking for images and video", *IEEE Trans. Circuits Syst. Video Technol*, Vol. 9, No. 4, pp. 545-550, June.

[11] Wong, K. K, Tse, C.H., Ng, K.S., Lee, T.H., and Cheng, L.M., (1997) "Adaptive watermarking", *IEEE Trans on Consumer Electronic*, Vol.43, No.4, pp.1003-1009, Nov.

[12] Wallace, G. K., (1991) "The JPEG still picture compression standard," *Commun. ACM*, Vol. 34, No. 4, pp.30-44, April.

[13] Hsu, C.T. and Wu, J.L. (1999) "Hidden digital watermarks in images", *IEEE Trans on Image Processing*,Vol.8, No.1, pp.58-68, Jan..

[14] Kunder, D. and Hatzinakos, D. (1997) " A robust digital image watermarking method using wavelet-based fusion", *Proc. of the IEEE International Conference on Image Processing*, pp.544-547.

[15]. Swanson, M.D., Zhu, B., and Tewfiw, A.H. (1997) " Data hinting for video-in-video", *Proc. of the IEEE International Conference on Image Processing, pp.676-679*

[16]. Hartung, F. ,and Girod, B.(1997) " Watermarking of MPEG-2 coded video in the bitstream domain ", *Proc. ICASSP*.

[17]. Kim, J.W., and Lee, S.U. (1992) "A transform domain classified vector quantizer for image coding", *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 2, No.1, pp.3-14, Mar.

[18] Cho, N. I., and Lee, S. U., (1992) "A fast 4x4 DCT algorithm for the recursive 2-D DCT," *IEEE Trans. Signal Process.*, Vol. 40, pp.2166-2173, Sept.

[19] Feig, E., and Winograd,S.(1992) "Fast algorithm for the discrete cosine transform," *IEEE Trans. Signal Process.*, vol. 40, pp. 2174-2193, Sept.

[20].Braudaway, G.W. (1997)" Protecting publicly-available image with an invisible image watermask", *Proc. of the IEEE International Conference on Image Processing*, pp.524-527.

[21] Chou, C.H., and Li,Y.C. (1995) " A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile", *Trans. on Circuits and Systems for Video*

Technology, Vol.5, No.6, pp.467-476, Dec.

[22] Hsu, C.T., and Wu, J.L. (1998) ” Multi-resolution watermarking for digital images”, *IEEE Trans. Circuits Syst. Part-II* , Vol. 45, No. 8, pp. 1097-1101, Aug.

[23] Kim, W.G., Lee, J. C., and Lee,W. D. (1999) ” An image watermarking scheme with hidden signatures”, *International Conference of Image Processing*, pp. 206-210.

[24] Kim, J. R., and Moon,Y. S. (1999) ” A robust wavelet-Based digital watermarking using level adaptive thresholding”, *International Conference of Image Processing*, pp.226-230.

Shih-Chang Hsia : He received the Ph.D. degrees from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, ROC, in 1997. During 1986-1989, he was an engineer in the R&D department of Microtek International Inc., Hsin-Chu. He was an instructor and associate professor in the Department of Electronic Engineering, Chung Chou Institute of Technology during 1991-1997. Currently, he is a professor in Department of Computer and Communication Engineering, National Kaohsiung First University of Science and Technology Kaohsiung. His research interests include VLSI designs, video coding and processing, data hiding system