1998 International Computer Symposium
Workshop on Cryptology and Information Security
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

# DATA HIDING IN IMAGES VIA MULTIPLE-BASED NUMBER CONVERSION AND LOSSY COMPRESSION*

*Da-Chun Wu*[1] *and Wen-Hsiang Tsai*[2]

Department of Computer and Information Science
National Chiao Tung University, Hsinchu, Taiwan 300,
Republic of China
Email: whtsai@cis.nctu.edu.tw

## ABSTRACT

A novel and easy method to embed any form of secret messages into a cover image with controlled distortion is proposed. Any lossy image compressor may be applied first to a cover image to produce a lossily-processed result as the basis for embedding data in the cover image. The stego-image is produced by embedding data in each pixel of a cover image by changing its gray value without excessing the range of the gray value difference of the corresponding pixels of the cover image and its lossily-processed one. The quantity of distortion that is caused by embedding data is never in excess of that is caused by the lossy compressor. A multiple-based number system is proposed to convert the information in the secret bit stream into values to be embedded in the choosing pixels of the cover image. Pseudo-random mechanisms may be used to achieve cryptography. It is found from experiments that the values of the peaks of the signal-to-noise ratio of the stego-images are larger than those yielded by the chosen compressor, which means that the distortion in the embedding result is more imperceptible than that in the compressed one.

## 1. INTRODUCTION

Many kinds of data, such as text, image, audio, and video, are represented in digital form in today's digital world, resulting in various types of digital media. Information can be embedded into digital media by making tiny changes which cause little notice to the human perception. Such data hiding techniques have many applications [1] [2], such as tamper proofing, watermarking, copyright protection, hidden annotations, authentication, secure and invisible communication, etc. In this study, we are interested in developing new methods for hiding secret messages in digital images. The secret message might be a serial number, a copyright logo, a caption data, a covert message, a plain text, another image, or anything that can be represented in bit stream form. An image that holds a secret message is called a cover image, and the output of the hiding process that includes the secret message is called a stego-image [3].

Many techniques for hiding data in images have been proposed. The method of changing least significant bits (LSB's) [4] embeds data by replacing the LSB of each pixel of the cover image, and a random number mechanism [5] is used to accomplish the security work when embedding data into the LSB's of randomly selected words on compact discs. The patchwork method [6] is based on a pseudo-random statistical process which changes the brightness of each selected pixel pairs by increasing one unit to one pixel and decreasing one unit from the other. The texture block coding method [2] hides data in continuous random texture patterns. Data hiding in images can also be applied in frequency or other transform domains. Two of such methods are the use of randomly sequenced pulse position modulated codes [7], and the secure spread spectrum method [8]. Some data hiding techniques exploit the characteristics of the human visual system to guarantee that the modification of the cover image is imperceptible. Two of such techniques are [9] and [10].

To produce stego-images with imperception is the most important goal of data hiding in image. The presence of the embedded message should not be noticed by the observer. Not every pixel in a cover image is suitable for embedding an equal quantity of information by changing its gray value. Although equal amounts of changes of gray values cause equal effects in some types of distortion measurements that are based on gray values differences, such as the root-mean-square-error (RMSE) and the peak of the signal-to-noise ratio (PNSR), the perceptible feelings of different image areas may be quite different even within an image. A mesh area in an image may need a different amount of modification from that needed by a smooth area in the same image to create a noticeable change. For an application that needs to hind a large amount of data in a cover image, finding an easy way to decide a suitable and acceptable gray value change for

[1] Also with the Department of Information Management, Ming Chuan University, Taipei, Taiwan 111, Republic of China.

[2] To whom all correspondence should be sent.

* This work is supported partially by National Science Council, the Republic of China under the Grant NSC86-2213-E009-113.

each individual pixel is a critical issue and is explored in depth in this study.

Image compression is a well-studied topic that codes pictures into less amounts of data. There are two kinds of image compression approaches: lossless and lossy. Lossless image compression techniques are error-free coding methods. A lossless-compressed image can be decompressed to be one which is identical to the original image. Since lossless compression methods keep detailed information in the image, the sizes of the compressed results are not reduced so much. Lossy image compression techniques instead produce results with smaller sizes and the image obtained from decompressing is not identical to the original one. However, to produce a result as similar to the original one as possible is a common goal of the lossy compression technique. Many lossy compression techniques have been proposed, such as block truncation coding (BTC) [11], vector quantization (VQ) [12], and transform coding like wavelet coding [13] and the cosine transform used by Joint Photographic Experts Group (JPEG) [14]. These methods use the characteristics of the human visual system's low sensitivity to small changes in gray values to reduce the size of compressed data with little distortion. Actually, all lossy image compression techniques look for ways to produce small-sized compressed results under the condition that the distortion in the resulting image is not noticeable by casual viewers. The quantity of the discarded information varies in different regions of the resulting image, that is, the magnitude of changes of the gray values in distinct pixels in the resulting image are not the same.

The aim of a lossy compression technique for getting higher similarity in a compression result to the original image is just like the goal of producing a stego-image as similar to the cover image as possible in data hiding. In this paper, we propose a novel, easy, and efficient technique for data hiding in images with imperception via the lossy image compression technique. We apply any of the existing lossy compression techniques to a cover image to produce a compressed image and then decompress it immediately to obtain a result, which we call in the sequel a lossily-processed image. A good lossy compression technique will produce a lossily-processed result which is quite similar to the original one. The lossily-processed image will be employed as the basis for further processing in the proposed data hiding process.

Also, we define a term, tolerable error range (TER), as follows. If a gray value g of a pixel p in an image is changed to a value in a range of gray values from $g'$ to $g''$ with imperception, we say that the range of gray values from $g'$ to $g''$ is a TER of p. In the proposed data hiding method, the difference range of gray values of every pair of corresponding pixels in the cover image and the lossily-processed one is taken as a TER. A modification of the gray value of each pixel in the cover image in the proposed hiding process is bounded by the TER of the pixel. So, the gray value change of each stego-image pixel, which is created by embedding, is limited to be no greater than that of the corresponding pixel in the lossily-processed image, which is created by the chosen lossy compression technique. That is, the distortion caused by embedding in each pixel of the cover image is no more than the distortion caused by the chosen lossy compression technique. This guarantees that the changes in the resulting stego-image is more imperceptible than those in the lossily-processed result.

The TER's for embedding data vary with each pixel, and so is the amount of information which can be represented by the TER. Since the size of a TER may not be exactly in power of 2, we propose a multiple-based number conversion method to convert the bit stream of the secret message into sets of data for embedding according to the TER's of the pixels.

The remaining part of this paper is organized as follows. In Section 2, the proposed multiple-based number system and embedding method are described. And several experimental results are illustrated in Section 3. Finally, some concluding remarks are stated in Section 4.

## 2. PROPOSED IMAGE HIDING METHOD

The concept of the proposed image hiding method is illustrated in Fig. 1. And an overview of the embedding system is shown in Fig.2. We sequentially present two main parts of the system, the multiple-based number conversion and the embedding process, in the following.

### 2.1 Multiple-Based Number Conversion

Since each pixel in the cover image has an individual TER, the magnitude of information which can be hidden in each pixel varies with its TER. In the following, a multiple-based number system and a corresponding number conversion process are defined to provide a way for embedding any type of information into a set of pixels with unequal sizes of TER's.

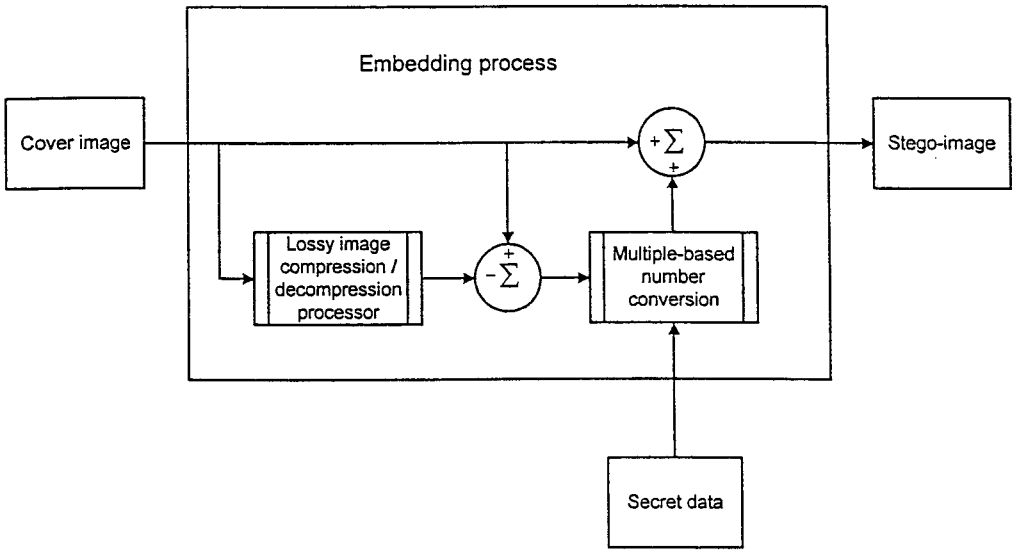



Fig. 1: Proposed data hiding method.

*Definition 1. Multiple-based number system.*

An integer number $d_{n-1}d_{n-2}..d_0$ is called a multiple-based number if every digit $d_i$ in the number has base $b_i > 0$ where i = 0, 1, ..., n-1. The value of $d_i$ is in the range from 0 to $b_i - 1$. The multiple-based number may also be denoted by $d_{n-1(b_{n-1})}d_{n-2(b_{n-2})}...d_{2(b_2)}d_{1(b_1)}d_{0(b_0)}$, and its decimal value is calculated by the equation

$$\begin{aligned} & d_{n-1(b_{n-1})}d_{n-2(b_{n-2})}...d_{2(b_2)}d_{1(b_1)}d_{0(b_0)} \\ &= d_{n-1} * (b_{n-2} * b_{n-3} * ... * b_0) \\ &\quad + d_{n-2} * (b_{n-3} * b_{n-4} * ... * b_0) + ... \\ &\quad + d_2 * (b_1 * b_0) + d_1 * b_0 + d_0 \\ &= ((...((d_{n-1} * b_{n-2} + d_{n-2}) * b_{n-3} \\ &\quad + d_{n-3}) * ... + d_2) * b_1 + d_1) * b_0 + d_0. \end{aligned}$$

*Property 1. Multiple-based number conversion.*

An integer I can be converted into a multiple-based number $d_{n-1(b_{n-1})}d_{n-2(b_{n-2})}...d_{2(b_2)}d_{1(b_1)}d_{0(b_0)}$ by computing the coefficients $d_0$, $d_1$, ..., $d_{n-1}$ as the remainders of iterative integer divisions of I by $b_0$, $b_1$, ..., $b_{n-1}$ progressively.

This property is similar to the process of computing the bits for a binary number from a given decimal value except variable bases instead of 2, 2, 2, ... are used in the iterative integer division steps.

*Property 2.*

An n-digit multiple-based number system with the chosen bases $b_{n-1}$, $b_{n-2}$, ..., $b_0$ and the number form of $d_{n-1(b_{n-1})}d_{n-2(b_{n-2})}...d_{2(b_2)}d_{1(b_1)}d_{0(b_0)}$ may represent any m-bit binary number if the chosen bases $b_{n-1}$, $b_{n-2}$, ..., $b_0$ satisfy the equation $m \le \left\lfloor \log_2 \left( \prod_{i=0}^{n-1} b_i \right) \right\rfloor$.

This property may be verified in the following way. Since an m-digit binary number can represent $2^m$ different information and an n-digit multiple-based number with base $b_{n-1}$, $b_{n-2}$, ..., $b_0$ can represent $\prod_{i=0}^{n-1} b_i$ different information, we can use an n-digit multiple-based number with base $b_{n-1}$, $b_{n-2}$, ..., $b_0$ to represent an m-bit binary number, if the condition $m \le \left\lfloor \log_2 \left( \prod_{i=0}^{n-1} b_i \right) \right\rfloor$ is met.

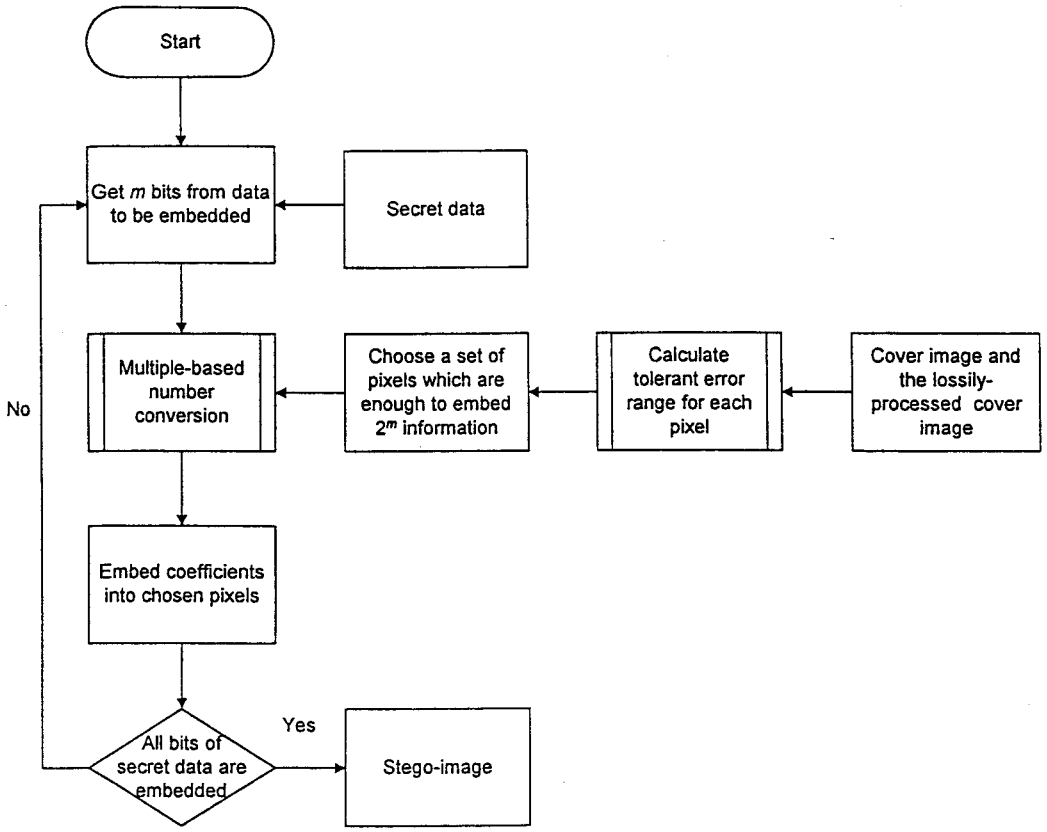



Fig. 2: Overview of proposed embedding system.

### 2.2 Data Embedding Process

In the proposed data embedding process, given a cover image C, we apply a lossy compression technique T to it and subsequently decompress the compression result to produce a lossily-processed image $C'$. For the gray value $g_i$ of each pixel $p_i$ in C and the gray value $g_i'$ of the corresponding pixel in $C'$, where $i = 0, 1, \ldots, n-1$, the change of gray values from $g_i$ to $g_i'$ is considered as an imperceptible noise if T is well-defined. The range from $g_i$ to $g_i'$ or from $g_i'$ to $g_i$ is the TER of $p_i$. We then change $g_i$ to a new pixel value $\hat{g}_i = g_i + d_i$, where $\hat{g}_i$ is in the TER and $d_i$ is a digit to be embedded. So, the amount of change made in the embedding is guaranteed not to excess that caused by the chosen lossy compression and decompression process, and the value of $d_i$ is in the range from 0 to $g_i' - g_i$ or $g_i - g_i'$ to 0. The number of all possible values of $d_i$ is just $|g_i' - g_i| + 1$ i. e., the size of the TER of $p_i$. We then take $|d_i|$ as a digit of a multiple-based number with base $b_i$ where the value of $b_i$ is taken to be $|g_i' - g_i| + 1$. The overall effect is that we use $p_i$ to embed a digit $d_i$ with base $b_i$ by changing the gray value of $p_i$ from $g_i$ to $g_i + d_i$.

Accordingly, a cover image can be used to represent $\prod_{i=0}^{S-1} \left(|g_i' - g_i| + 1\right)$ different information, where S denotes the size of the cover image. And so the cover image can be employed to embed any binary bit stream with length no larger than $\left\lfloor \log_2 \prod_{i=0}^{S-1} \left(|g_i' - g_i| + 1\right) \right\rfloor$. If we regard a given secret bit stream totally as a long binary number and apply the proposed multiple-based number conversion to it by using a set of pixels which have the largest sizes of TER's in the cover image, the number of the pixels which are

needed for converting the secret bit stream can be reduced to a minimum. But it is an effort to do computation with such a long binary number. In the proposed approach, we divide the bit stream into non-overlapping sub-streams whose sizes may be mutually unequal. For every sub-stream, say m bits long, we collect a set of pixels in C as a subimage $S_i$ for embedding the m-bit sub-stream. The size of the subimage is decided by Property 2, i. e., the product of all the sizes of TER's of the collected pixels in the subimage is no smaller than the value of $2^m$. For the gray value $g_{s_i}$ of each pixel $p_{s_i}$ in the subimage $S_i$ of C, let $g'_{s_i}$ be the gray value of the corresponding pixel of C', where $i = 0, 1, \ldots, n-1$, and let $s_{b_i} = |g_{s_i} - g'_{s_i}| + 1$. We then apply the multiple-based number conversion to the value of the m-bit sub-stream with multiple bases $b_{s_{n-1}}$, $b_{s_{n-2}}, \ldots, b_{s_0}$. Assume that the resulting digits from the conversion are $c_{n-1}, c_{n-2}, \ldots, c_0$. Then we embed the digits of the multiple-based number $(c_{n-1}c_{n-2}\ldots c_0)$ in each pixel $p_{s_i}$ of the subimage $S_i$ of C by the following equation

$$\hat{g}_{s_i} = \begin{cases} g_{s_i} + c_i & g_{s_i} \le g'_{s_i}; \\ g_{s_i} - c_i & g_{s_i} > g'_{s_i}, \end{cases}$$

where $\hat{g}_{s_i}$ is the gray value of the embedding result of $p_{s_i}$. In the above calculation, the new gray value $\hat{g}_{s_i}$ of $p_{s_i}$ must be in the range from $g_{s_i}$ to $g'_{s_i}$ or from $g'_{s_i}$ to $g_{s_i}$. So the distortion caused by embedding data in $p_{s_i}$ is no more than that caused by the lossy compression/decompression process. It is thus shown that our proposed method can guarantee to produce a more imperceptible result than that of the chosen lossy compression method. Furthermore, any secret digital data, such as plain text, images, files, etc., are treated identically in the proposed method as a bit stream and can be embedded without special preprocessing.

### 2.3 Security

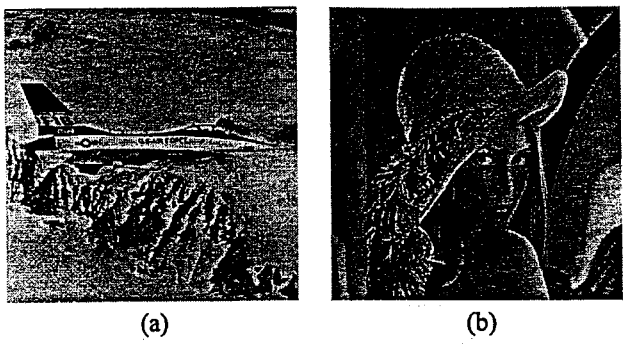
Many alternatives can be used in the embedding process to avoid easy illicit extraction of the embedded data from a stego-image. A pseudo-random mechanism can be used in many steps in the embedding process to achieve cryptography, for example, in the selection of the parameters of the chosen compression method, in the selection of the sizes of the sub-streams of the secret data stream, in the choice of the bits from the secret bit stream which compose the sub-stream, and in the way of choosing the subimage from the cover image in which the secret sub-stream is embedded. The pseudo-random mechanism may use a single seed or multiple seeds to generate a sequence of numbers in the embedding steps. Without the seed(s), an observer cannot reconstruct the embedded data correctly.

### 2.4 Extraction process

The secret data extraction process is accomplished by using the stego-image, the original cover image, and the seed(s) used by the pseudo-random generation mechanism in the embedding steps. Firstly, apply the same lossy compression method used in embedding process to the cover image and then decompress it to produce a lossily-processed cover image. Select the subimages from the stego-image with the same sequences as in the embedding process, and then compare the corresponding pixels in both the cover image and the lossily-processed cover image to determine the TER's and the bases which are used in the multiple-based number conversion. Then, extract out the embedded value in each pixel, group the values to form a multiple-based number, and convert the number into a bit stream using the bases to which the values belong. Collect the resulting bit streams to recover the entire embedded data.

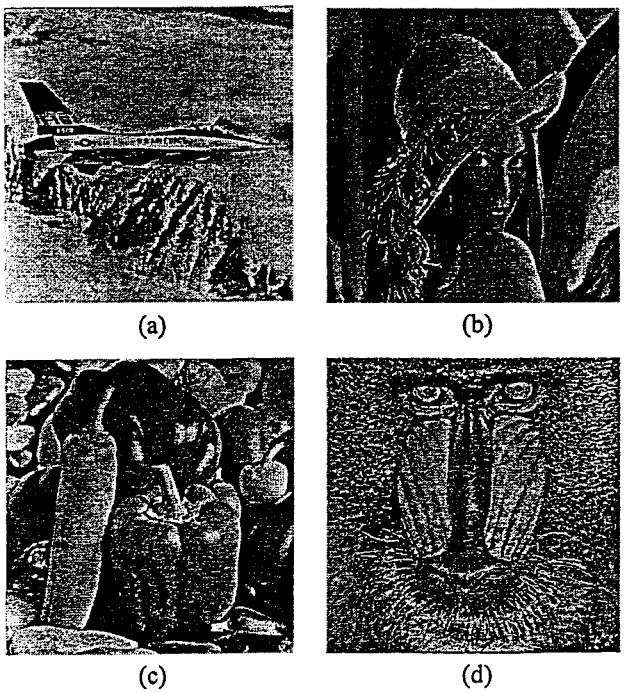
## 3. EXPERIMENTAL RESULTS

In our experiments, four cover images as shown in Fig. 3, each with size $512 \times 512$, were used. A lossy JPEG compressor was applied to the cover images. The lossily-processed results are shown in Fig. 4. The values of the compression ratios, the root-mean-square errors (RMSE's), and the peaks of the signal-to-noise ratio (PNSR's) of the compression results are shown in Table 1, which shows that an image consisting of large smooth areas yields a more condensed size and smaller gray value difference than an image consisting of large edged areas.



(a) (b) (c) (d)

Figure 3: Cover images used in experiments with size 512×512. (a) "F-16", (b) "Lena", (c) "Peppers", and (d) "Baboon".



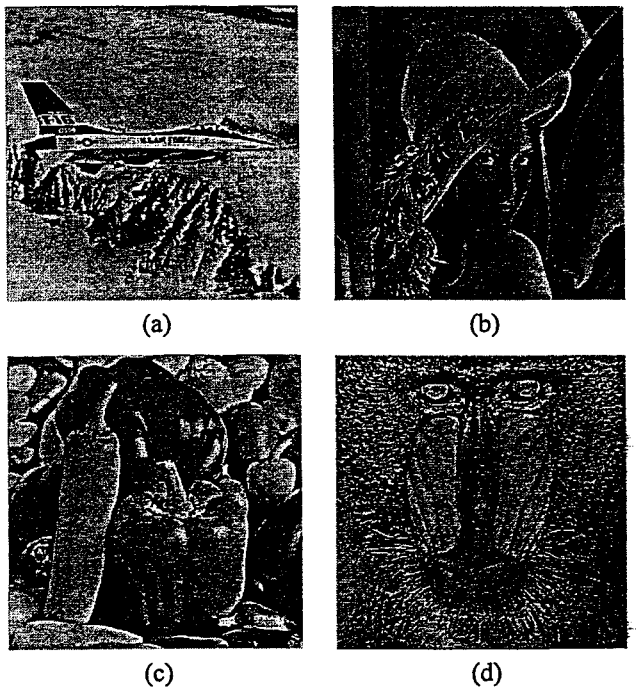
(a) (b) (c) (d)

Figure 4: JPEG-compressed images with size 512×512. (a) "F-16", (b) "Lena", (c) "Peppers", and (d) "Baboon".

We then used a file which consists of the text of this article excluding Section II as the secret data in the experiments. The stego-images resulting from embedding the file are shown in Fig. 5 and Fig. 6. The results of Fig. 5 were produced by embedding data in each pixel of the cover image in a random traversing order generated by a

Table 1
Values of compression ratios (CR's), RMSE's, and PSNR's of compressed images.

| Images | After JPEG-processed | | |
|---|---|---|---|
| | CR | RMSE | PSNR |
| F-16 | 9.99 | 4.00 | 36.08 |
| Lena | 11.30 | 3.89 | 36.34 |
| Peppers | 10.84 | 4.60 | 34.87 |
| Baboon | 4.97 | 8.32 | 29.73 |



(a) (b) (c) (d)

Figure 5: Resulting stego-images of (a) "F-16", (b) "Lena", (c) "Peppers", and (d) "Baboon" that embed a file into the pixels of the cover image by a random traversing order. The file consists of the text of this article excluding Section II.

pseudo-random scheme, which walks through all pixel in the cover image and visit each pixel only once. In Fig. 6, the results were produced by embedding data in each pixel of the cover image in a traversing order from left-top to right-bottom. So, embedding the secret file changes the pixel values of the upper part of the cover image, but no apparent distortion can be noticed in the upper part of the resulting stego-image. It is thus shown that the proposed hiding method can embed data without noticeable changes. The values of the PSNR and RMSE of the embedding results are shown in Table 2. It is seen from the table that the RMSE values become smaller and the PSNR values become larger, which means that the distortion caused by embedding data in the stego-images are not more than those of the JPEG-processed results, so the resulting stego-images are more imperceptible than the lossily-processed one.

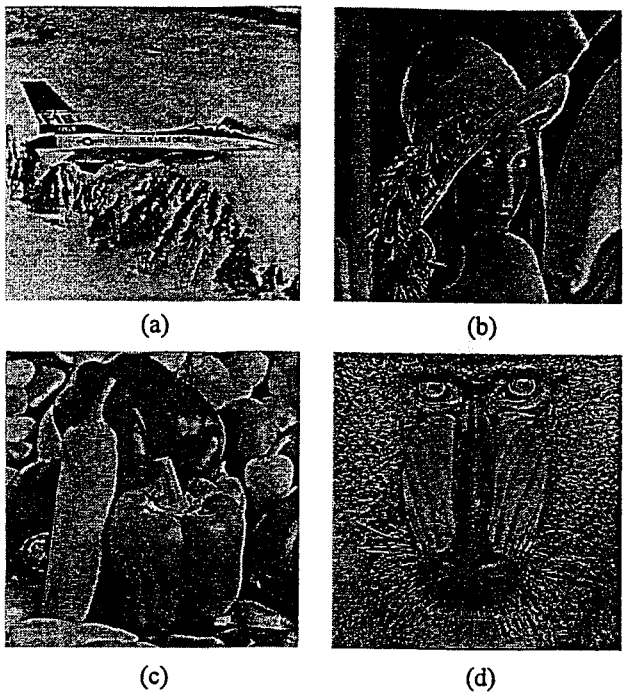


Figure 6: Resulting stego-images of (a) "F-16", (b) "Lena", (c) "Peppers", and (d) "Baboon" that embed a file into the pixels of the cover image by a traversing order from the left-top to right-bottom. The file consists of the text of this article excluding Section II.

Table 2
Values of RMSE's and PSNR's of stego-images that embed a file consisting of the text of this article excluding Section II.

| Images | Embedding using random traversing order | | Embedding using scanning order | |
|---|---|---|---|---|
| | RMSE | PSNR | RMSE | PSNR |
| F-16 | 1.33 | 45.64 | 1.17 | 46.38 |
| Lena | 1.29 | 45.94 | 1.21 | 46.64 |
| Peppers | 1.43 | 45.00 | 1.39 | 45.29 |
| Baboon | 2.16 | 41.46 | 2.45 | 40.36 |

## 4. CONCLUDING AND REMARKS

We have proposed a novel and easy method for embedding any form of digital data into an image with controlled distortion. Any lossy image compressor may be applied first to a cover image to produce a lossily-processed result as the basis for embedding data in the cover image. A multiple-based number conversion is also used to embed a bit stream into a group of pixels and any pseudo random mechanism may be applied to achieve cryptography. It is guaranteed the error caused by embedding data in each pixel of the cover image does not excess the error caused by the chosen lossy image compressor. The proposed method can be easily applied to embed data in a color cover image. The method also can be modified for information hiding if one can provide two similar information sources and embed data into the difference of the two sources by using the proposed multiple-based number conversion. This provides a way to foresee the worst distorted stego-result before embedding steps begin.

## REFERENCES

[1] J. Zhao, E. Koch, and C. Luo, "Digital Watermarking In business Today and Tomorrow," Communication of the ACM, vol. 41, No. 7, pp. 67-74, 1998.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for Data Hiding," IBM System Journal, vol. 35(3/4), pp. 313-336, 1996.

[3] B. Pfitzmann, "Information Hiding Terminology," Proc. First Int'l Workshop Information Hiding, Lecture Notes in Computer Science No. 1174, Springer-Verlag, Berlin, 1996, pp. 347-349.

[4] S. Walton, "Image Authentication for a Slippery New Age," *Dr. Dobb's Journal*, vol. 20, no. 4, pp. 18-26, April 1995.

[5] L. F. Turner, "*Digital Data Security System*," Patent IPN, WO 89/08915, 1989.

[6] W. Bender, N. Morimoto, and D. Gruhl, "Method and Apparatus for Data Hiding in Images," U. S. Patent No. 5689587, 1997.

[7] E. Koch and J. Zhao, "Towards Robust and Hidden Image Copyright Labeling," *Proc. IEEE Nonlinear signal and image processing*, pp. 452-455, June 1995.

[8] Ingemar J. Cox, Joe Kilian, F. Thomson Leighton and Talal Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673-1687, Dec. 1997.

[9] M. D. Swanson, B. Zin, and A. H. Tewfik, "Multiresolution Scene-Based Video Watermarking Using Perceptual Models," *IEEE Journal on Selected Areas in Communication*, vol. 16, No. 4, pp. 540-550, 1998.

[10] C. I. Podilchuk, and W. Zeng, "Image-Adaptive Watermarking Using Visual Models," *IEEE Journal on Selected Areas in Communication*, vol. 16, No. 4, pp. 525-539, 1998.

[11] E.J. Delp, and O. R. Mitchell, "Image Compression Using Block truncation Coding," *IEEE Transaction on. Communication*, COM-27, No. 9, pp. 1335-1342, 1979.

[12] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.

[13] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image Compression Through Wavelet Transform Coding," *IEEE Transaction on. Information Theory*, vol. 38, no. 2, pp. 719-746, 1992.

[14] ISO/IEC JTC1/SC2/WG8, *Coding of Still Picture*, JPEG Committee Technical Draft R8, Aug. 14, 1990.