Data-Hiding for Halftone Images Using an Improved CPT scheme

Phan Trung Huy[1], Nguyen Hai Thanh[2], Cheonshik Kim[3], Ching-Nung Yang[4]

[1]Hanoi University of Science and Technology, Hanoi, Vietnam
[e-mail: phanhuy@hn.vnn.vn]
[2]Ministry of Education and Training, Hanoi, Vietnam
[e-mail: nhthanh@moet.gov.vn]
[3]Dept. of Computer Engineering, Sejong University, 98 Gunja-Dong,
Gwangjin-Gu, Seoul 143-747, Korea
[e-mail: mipsan@paran.com, mipsan@sejong.ac.kr]
[4]Department of Computer Science and Information Engineering,
National Dong Hwa University, #1, Sec. 2, Da Hsueh Rd., Hualien, Taiwan
[e-mail: cnyang@mail.ndhu.edu.tw]

**Corresponding author: Cheonshik Kim*

Received October 22, 2012; revised January 2, 2013; accepted January 28, 2013; published February 26, 2013

Abstract

In publishing applications, it is advantageous to embed data in halftone images. The CPT scheme (Chen-Pan-Tseng, 2000) is a steganographic data hiding scheme that was proposed for binary images, e.g., facsimiles. The CPT scheme uses a secret key and weight matrix to protect the hidden data and can hide as many as $r = \lfloor \log_2(m \times n+1) \rfloor$ bits of data in the image by changing at most 2 bits in the image. Our proposed scheme also uses a secret key to protect it from being compromised and a weight matrix to increase the data hiding rate. Our scheme improves the performance of the CPT scheme by using the simple principle of splitting each block into two parts. Our proposed scheme is called improved CPT (ICPT) and has a very high embedding capacity compared to previous schemes. Experimental results demonstrate that our proposed scheme generally has higher performance than previous schemes.

Keywords: CPT, Data Hiding, DHSPT, Error Diffusion, Halftone Images, ICPT

A preliminary version of this paper appeared in ACIIDS 2011, April 20-22, Daegu, Korea.

http://dx.doi.org/10.3837/tiis.2013.02.013

1. Introduction

Data hiding [14-18] can be used for copyrights, annotation, and communication. The data are hidden by altering some nonessential pixels in a cover image. For example, in a color image (or grayscale image), the least-significant bit (LSB) of each pixel can be modified to embed secret data. However, the human visual system (HVS) is very sensitive to such modifications. The most challenging problems in hiding secret data in halftone images are achieving a high embedding capacity and low image distortion. Pixels with a low amount of modification in an image can elude detection by the HVS [1].

Halftone images can be used to print books, newspapers, and magazines, which cannot print continuous tones. A halftoning technique is used to produce two-tone texture patterns as approximations of the original multi-tone images [1-6]. There are two main kinds of halftoning techniques: ordered dithering [5] and error diffusion [5]. Error diffusion is more complicated than ordered dithering, but it can yield a higher visual quality with few blocking effects. This technology compares the sum of the image pixel intensity and error from the past with a mixed threshold to determine the output. Then, the halftoning error is fed forward to its adjacent neighbors using a kernel. Two commonly used kernels are the Jarvis kernel and the Steinberg kernel [4, 5]. The Jarvis kernel has large support and tends to give halftone images with high contrast and a coarse texture. The Steinberg kernel has less support and gives halftone images with a fine texture and good contrast. A few bit-planes, which are composed of halftone images, are changed, which will have a bad effect on the quality of an image. Therefore, it is important to flip a bit in a 4 × 4 block of an image. However, most schemes are not appropriate for hiding a large amount of data in a halftone image because a halftone image is only composed of 0's and 1's.

To compensate for these issues, special techniques have been proposed for halftone images, such as covering code [6, 12]. Alternatively, *Fu* & *Au* [7] proposed the Data Hiding Smart Pair Toggling (DHSPT) method, which has good image quality. *Chao et al*. [6] used the hamming code (7, 4) to hide secret data. *Li et al*. [10] proposed a block-overlapping parity check (BOPC), which can be applied to the existing halftone image data hiding scheme. BOPC reduces the number of pair togglings required in DHSPT. *Yip et al*. [11] proposed Pattern-and-Intensity-Preserve data hiding (PI-Preserve), which hides data without the help of the original grayscale image. The PI-Preserve scheme achieves a considerably high visual quality. *Kim et al*. proposed a data hiding scheme that uses the hamming code (15, 11) [12], and this scheme is very efficient because it only requires a few pixel flips to conceal messages in an image. The first data-hiding scheme based on a block-wise method was CPT (Chen, Pan, and Tseng) [9].

Qin et al. [14] proposed an image hashing scheme using the halftone mechanism, which can be applied in the field for image authentication and retrieval. This scheme has satisfactory robustness performance against perceptual content-preserving manipulations. In [19], a Modified Data Hiding Error Diffusion (MDHED) method was proposed as an effective method to hide a relatively large amount of data while yielding halftone images with good visual quality. Yet, the robustness was lacking for practical print-and-scan applications. In [20], data hiding and authentication schemes were proposed based on halftoning and coordinate projection. These schemes can detect, localize, and repair the tampered area of an image. Any portion of the image can be used to reconstruct the entire image, with a greater reconstruction quality as the portion size increases.

Kim and Afif [21] presented a cryptographically secure authentication watermarking technique for halftone and binary images. It consists of choosing a set of pseudorandom pixels in an image, clearing them, computing the message authentication code (or the digital signature) of the cleared image, and inserting the resulting code into the selected random pixels. The scheme can detect even a single pixel alteration in the host image. It can be used with secret key or public key ciphers. *Pei et al*. [22] proposed a watermarking scheme, which embeds a watermark into the dithered image by switching the positions of the sub-blocks according to the bit distribution of the watermark. This method is called paired sub-image matching ordered dithering (PSMOD), and the decoder is provided with a priori information about the original watermark. The advantage of this method is that it is sufficiently robust to withstand the cropping, tampering, and printed-and-scanned degradation processes. Its shortcoming is its dependence on the prior knowledge of the original watermark in decoding. In [23], *Pei* and *Guo* proposed applying a noise-balanced error diffusion technique (NBEDF) to an original gray-level image. The visual decoding pattern can be perceived when two or more similar error-diffused images are overlaid on each other.

The CPT method is based on a pixel block scheme. Its embedding capacity is $\log_2(q+1)$ secret bits per block by flipping at most two pixels, where the size $q = m \times n$ [8, 9]. The evaluation of data hiding is often based on the visual quality of a stego image and the data-hiding capacity. Overall performance comparisons are summarized in Table 1.

Table 1. Comparisons of various methods (advantages (+), moderate (~), shortcomings (-), unknown (x)).

	Quality	Robustness	Complexity	Capacity	Additional feature (s)
[1]	~	+	~	-	Inverse halftoning
[2]	~	-	~	-	Reversible Data Hiding
[3]	-	+	+	-	Visual cryptography.
[6]	~	-	+	~	Cover codes
[7]	+	~	+	~	Hiding smart pair toggling (DHSPT)
[10]	+	-	-	~	Block-overlapping parity check (BOPC)
[11]	+	-	~	~	Pattern-and-Intensity-Preserve Data Hiding (PI-Preserve)
[12]	+	-	+	~	Matrix Encoding
[14]	x	+	-	x	An image hashing scheme using the halftone - authentication and retrieval.
[19]	x	x	~	~	Modified Data Hiding Ordered Dithering (MDHED)
[20]	x	+	-	x	Detect, localize, and repair the tampered area of the image.
[21]	x	-	+	-	Secure authentication
[22]	~	+	-	~	Paired sub-image matching ordered dithering (PSMOD)
[23]	x	~	-	~	Decoding: overlaid on each other
ICPT	+	-	~	+	

In this paper, we improve the CPT scheme and develop a new scheme (called the ICPT scheme) using a block-wise operation. Our ICPT has a higher embedding capacity—maximal or near-maximal—than any block-wise scheme. The advantages of our proposed scheme are a higher capacity and a better image quality. CPT's embedding rate for an $m \times n$ block is $r = \lfloor \log_2(m{\times}n{+}1) \rfloor$, whereas the ICPT scheme has an embedding rate of approximately $rx = \lfloor \log_2((1{+}(m{\times}n){\times}(m{\times}n{+}1)/\,2) \rfloor$, which is about $2r$ - 1. The ICPT scheme can change at most two bits to conceal the embedding rate in a block, in which case rx is $2r$-1. The *MPSNR* of ICPT is the same that of the BOPC scheme in a specific region.

The remainder of this paper is organized as follows. Section 2 reviews related works. In Section 3, our ICPT scheme is proposed. Section 4 presents our experimental results, and a comparison of the results with those of other previous data hiding schemes is also given. Section 5 is the conclusion.

2. Related Work

2.1 Error Diffusion

Error-diffusion is an alternative dithering technique that has emerged as the standard because of its simplicity and output quality. The error-diffusion algorithm, first proposed by Floyd and Steinberg [4], is schematically shown in Fig. 2 and works as follows. The quantization error at each pixel is filtered and fed back to the input in order to diffuse the error among neighboring grayscale pixels. The error diffusion filter is shown in **Fig. 1**, in which "*x*" denotes the current pixel. **Fig. 2** shows the error diffusion algorithm for creating a cover image. First, the original grayscale image is divided into P × Q blocks. The image is segmented into 4 × 4 pixel blocks, assuming that x_1, x_2, ...,$x_{n\times n}$ are the values of the pixels in a block.

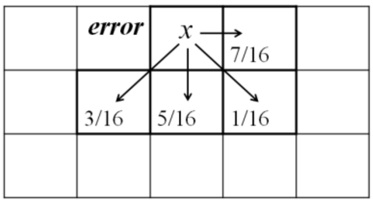


Fig. 1. Floyd-Steinberg kernel.

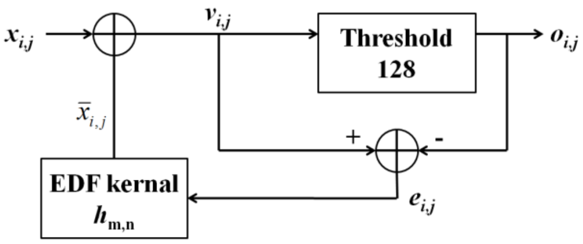


Fig. 2. Diagram of error diffusion algorithm.

The variable $x_{i,j}$ denotes the current input pixel value, and $\bar{x}_{i,j}$ denotes the diffused error sum added from the neighboring processed pixels to the kernel. The variable $o_{i,j}$ denotes the binary output in position (i, j). To achieve the effect of a continuous-tone illusion without the diagonal visual artifacts, the error kernel $h_{\mathrm{m,n}}$ is used to diffuse the error caused by the difference $e_{i,j}$ between the output binary value and input grey level value, according to the matrix shown graphically in Fig. 1. The variable $v_{i,j}$ denotes the modified value. The kernel is shown in **Fig. 1**, where "*x*" denotes the position of the currently processed pixel. The variables can be calculated as follows using **Eq. (1)** and **Eq. (2)**.

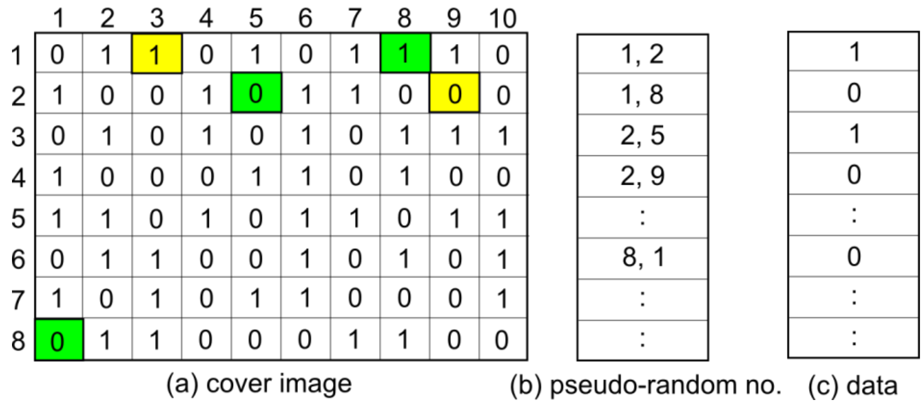
$$v_{i,j} = x_{i,j} + \bar{x}_{i,j},\;\; where\;\; \bar{x}_{i,j} = \sum_{m=0}^{2}\sum_{n=-2}^{2} e_{i+m,j+n} \times h_{m,n} \tag{1}$$

$$e_{i,j} = v_{i,j} - o_{i,j},\;\; where\;\; o_{i,j} = \begin{cases} 0, & if \quad v_{i,j} < 128 \\ 255, & if \quad v_{i,j} \geq 128 \end{cases} \tag{2}$$

In **Eq. (1)**, $e_{i,j}$ is the difference between $v_{i,j}$ and $o_{i,j}$. The error diffusion algorithm scans the image from left to right and top to bottom, quantizing pixel values one by one. Each time, the quantization error is transferred to the neighboring pixels, while not affecting the pixels that have already been quantized.

2.2 Data Hiding Self Toggling

DHST (Data Hiding Self Toggling) [7] was designed to select the position where data (**Fig. 3**(c)) can be concealed using a pseudo-random number (**Fig. 3**(b)). Therefore, one bit can be hidden by flipping a pixel at that position. In **Fig. 3**(a), the green boxes must be flipped to conceal bits. Thus, DHST can encode an image using only a few computations based on a random number generator. As a result, one major disadvantage of DHST is a low perceptual quality.



	1	2	3	4	5	6	7	8	9	10
1	0	1	1	0	1	0	1	1	1	0
2	1	0	0	1	0	1	1	0	0	0
3	0	1	0	1	0	1	0	1	1	1
4	1	0	0	0	1	1	0	1	0	0
5	1	1	0	1	0	1	1	0	1	1
6	0	1	1	0	0	1	0	1	0	1
7	1	0	1	0	1	1	0	0	0	1
8	0	1	1	0	0	0	1	1	0	0

(a) cover image

1, 2
1, 8
2, 5
2, 9
:
8, 1
:
:

(b) pseudo-random no.

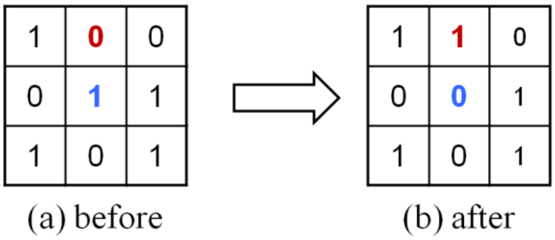
1
0
1
0
:
0
:
:

(c) data

Fig. 3. Example diagram of Data Hiding Self Toggling.

2.3 Data Hiding Pair Toggling

DHPT [7] is a method to improve DHST by alleviating the problem of low perceptual quality. Suppose a master pixel (center pixel in **Fig. 4**(a)) at a pseudo-random location needs to self-toggle and there are M pixels with the opposite color in the 3 × 3 neighborhood. In DHPT, one of the pixels (called the slave pixel (**Fig. 4**(b)) is also randomly chosen (a red pixel) to self-toggle. In this case, two errors have occurred; however, the average intensity of one positive and one negative can be preserved. If M (3 × 3 in (a)) is equal to zero (i.e., all of the pixels within the 3 × 3 neighborhood have the same color), then there is no flipping because of the generation of salt-and-pepper in the image.



1	0	0
0	1	1
1	0	1

(a) before

1	1	0
0	0	1
1	0	1

(b) after

Fig. 4. Example diagram of Data Hiding Pair Toggling.

2.4 CPT Scheme

The CPT scheme is a pixel block-based method for hiding data. A given halftone image, G, is partitioned into disjoint blocks F_i, where $1 \le i \le N$ for some N, each of which is a binary matrix of size $m \times n$. Each entry in each block is considered to be a pixel of an image and has a binary value. Two matrices, K and W, also of size $m \times n$, are combined with these image blocks. The matrix K is a binary secret key, and W is a weight matrix of integers that are shared by the sender and receiver. The variable r is the number of bits to be embedded in each $m \times n$ block of F by changing the values of, at most, two pixels. The embedding procedure of the CPT scheme is organized as follows.

CPT (Embedding Method)

```
Input: Block $F$ of original image $G$, secret key $K$, secret weight matrix $W$, number $r=\lfloor \log_2(m\times n+1)\rfloor$ of bits to be embedded in a block, secret $r$-bit stream $b = b_1b_2...b_r$.
Output: Block $F'$ of stego image $G'$.
Step 1. Compute $T=F \oplus K$, where $\oplus$ is the bit-wise exclusive OR (XOR) operation.
Step 2. Compute SUM($T\otimes W$) mod $2^r$, where $\otimes$ is the bit-wise multiplication operation.
Step 3. From the matrix T, compute for each $w = 1..2^r-1$ the following set:
        $S_w = \{(j,k)|\ ([W]_{j,k}=w\wedge[T]_{j,k}=0) \vee ([W]_{j,k}=2^r-w\wedge[T]_{j,k}=1)\}$.
        // every matrix index $(j,k)$ such that if we complement $[F_i]_{j,k}$, increase the sum by $w$
        Compute $d=b_1b_2\ldots b_r$ – SUM($T\otimes W$) mod $2^r$.
        if $d = 0$, then $F$ is not modified;
        else {
            a) Randomly pick an $h\in\{0, 1, ..., 2^r-1\}$ such that $S_{hd} \neq 0$ and $S_{-(h-1)d}\neq 0$.
            b) Randomly pick a $(j, k) \in S_{hd}$ and complement the bit $[F_i]_{j,k}$;
            c) Randomly pick a $(j, k) \in S_{-(h-1)d}$ and complement the bit $[F_i]_{j,k}$;
        }
```

Example 1. Assume that the size of K and W is 3×3 (see **Fig. 5**). We consider a 3×3 block, F, (see **Fig. 5**) of a host image, G, and show how to embed $r = 2$ bits, b_1b_2, of data in F, where b_1b_2 is 11_2. Compute $SUM\ ((F \oplus K) \otimes W) = 1+2+3+1+2 = 9 \bmod 2^2$ to obtain 1. Next, compute $d = b_1b_2 - SUM\ ((F \oplus K) \otimes W) = 11_2 - 1 = 2 \bmod 4$. If $d = 0$, then there is no change in F; otherwise we have to increase or decrease F by (d mod 4). Observe that if we complement bit F, then $[F\oplus K]$ will be complemented. If $[F\oplus K]$ is swapped from 0 to 1, then the modular sum will be increased by w; otherwise, the sum will be decreased by w. In this case, $d = 2$, so we have to increase the weight by 2. Since $[F\oplus K]_{2,2} = 0$ and $[W]_{2,2} = 2$, we can complement $[F]_{2,2}$. Therefore, we obtain the new F' from F. In the extraction phase, we compute $SUM\ ((F' \oplus K) \otimes W) = 1+2+3+2+1+2 = 11 \bmod 4$ or $3 = 11_2 = b_1b_2$, which is the extracted secret data.

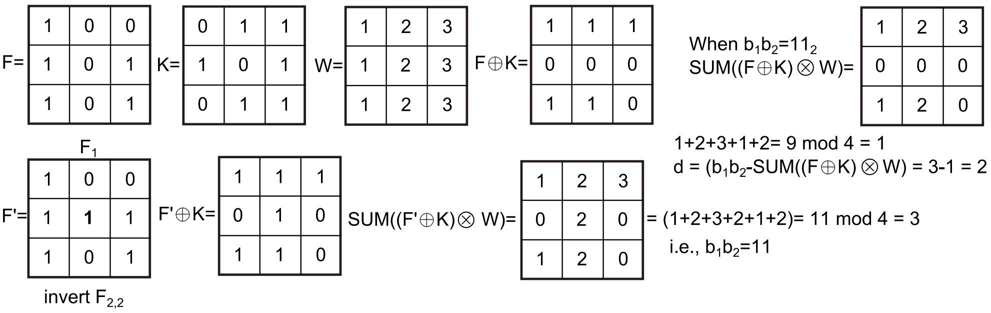


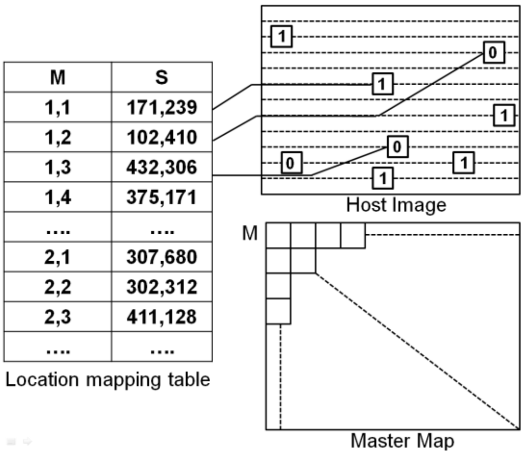
Fig. 5. Example of embedding with block *F*, key *K*, and weight matrix *W*.

2.5 Bock-Overlapping Parity Check (BOPC) Method

In BOPC, a digital logo *L* is embedded into a host image *S* with all the same size X × Y to create a watermarked image *W*. BOPC uses a data structure called the bubble formation, which is applied on the toggle map to choose the locations where smart pair toggling will be applied. In BOPC, *S* is divided into two groups, master pixels and slave pixels. Master pixels will be toggled, if necessary, to store the embedded logo data. When a master pixel is toggled, a neighboring slave pixel is typically toggled in a complementary way to preserve the local intensity. A pseudo-random number generator with a known seed, *K*, is used to generate a set of (2X+1)×(2Y+1) pseudo-random locations on *S*. These are master pixels that are used to form a master map (**Fig. 6**). The elements in the master map *M* are divided into overlapping blocks with the size of 3 × 3 (**Fig. 7**).

$$I_{ij} = \left[\sum_{X=2i-1}^{2i+1} \sum_{Y=2j-1}^{2j+1} M_{xy} \right] \mathrm{mod}\ 2 \tag{3}$$

During the watermark extraction process, the master map is built using a pseudo-random number generator with the same seed *K*. A parity map is formed using **Eq. (3)**.



M	S
1,1	171,239
1,2	102,410
1,3	432,306
1,4	375,171
….	….
2,1	307,680
2,2	302,312
2,3	411,128
….	….

Fig. 6. Master map formation with corresponding location mapping table.

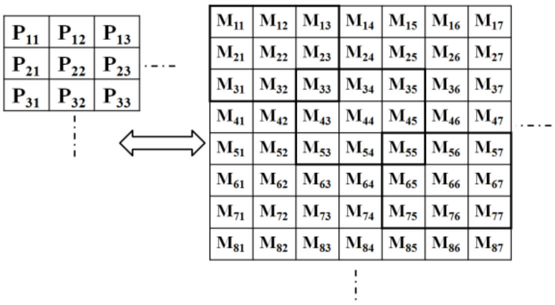

P11 P12 P13
P21 P22 P23
P31 P32 P33
M11 M12 M13 M14 M15 M16 M17
M21 M22 M23 M24 M25 M26 M27
M31 M32 M33 M34 M35 M36 M37
M41 M42 M43 M44 M45 M46 M47
M51 M52 M53 M54 M55 M56 M57
M61 M62 M63 M64 M65 M66 M67
M71 M72 M73 M74 M75 M76 M77
M81 M82 M83 M84 M85 M86 M87

Fig. 7. Corresponding locations of element in *P* and block in *M*.

3. Proposed ICPT Scheme

In this section, we introduce the ICPT scheme, which has an embedding rate that is maximal and about twice the embedding rate of the CPT scheme.

3.1 Embedding Algorithm for ICPT Scheme

In this section, we propose an embedding algorithm that can improve that in [8] for ICPT. The basic idea is to use an abelian group under an XOR ⊕ operation.

The inputs and notation to our scheme are as follows:

- *F*: a host bitmap, which is to be modified to embed data. We will partition *F* into blocks of size $m \times n$. For simplicity, we assume that the size of *F* is a multiple of $m \times n$.

- *K*: a secret key shared by the sender and receiver. It is a randomly selected bitmap of size $m \times n$.

- *W*: a secret weight matrix shared by the sender and receiver. It is an integer matrix of size $m \times n$.

- *b*: some critical information consisting of bits to be embedded in *F*.

- *r*: the number of bits to be embedded in each $m \times n$ block of *F*. The value of *r* satisfies 2^r-1 $\leq mn$.

- α, β: $p = m \times n + 2$,
$$\alpha,\beta = \begin{cases} \beta = \log_2 p \text{ and } \alpha = \log_2 p, & \text{If } (\log_2 p = \log_2 (p/3)+1) \\ \beta = \log_2 p - 1 = \alpha, & \text{If } (\log 2\, p > \log 2\, (p/3) + 1) \end{cases} \tag{4}$$

- $m(p)$: the size of the bits to be concealed in *F*, where $p = m \times n + 2$.
$$m(\mathrm{p}) = \alpha + \beta, \text{ where } \mathrm{p} \geq 2^{\alpha} + 2^{\beta} \tag{5}$$

- *M*: an abelian group under an XOR ⊕ operation on integers represented as $M = \{\check{n} \in \mathbb{Z}: 0 \leq \check{n}^{\leq} 2^{m(p)}-1\}$ with the scalar multiplication • defined by $\check{n} \bullet 0 = 0$ and $\check{n} \bullet 1 = \check{n}$ for every $\check{n} \in M$, where 0 is the unit for the ⊕ operation, and $0,1 \in \mathbb{Z}_2$, where $\mathbb{Z}_2$ is a field of characteristic 2. *M* is a $\mathbb{Z}_2$-module with this multiplication. We define two subsets of *M*: $M_1 = \{ \check{n}: 1 \leq \check{n} \leq 2^{\alpha} -1\}$, $M_2 = \{ \check{n} : 2^{\alpha} \leq \check{n} \leq 2^{\alpha+\beta} -1\}$. M_1 and M_2 are the set of *W*.

- L_1, L_2: a set of *F*. Namely, $|L_1| \geq 2^{\alpha}$-1, $|L_2| \geq 2^{\beta}$-1, and *W*, L_1, L_2 are required to satisfy $\{W_{ij}: F_{ij} \in L_1\} = M_1$, $\{W_{ij}: F_{ij} \in L_2\} = M_2$.

- *S*: weighted sum of block *F*. $S = [W \bullet T]$, where $T = F \oplus K$. $S = x_1 \oplus x_2$ with $x_1 = S \wedge ((2^{\alpha} -1) \times 2^{\beta})$ and $x_2 = S \wedge (2^{\beta} -1)$, where ∧ is the bit-wise AND operation and ⊕ is the bit-wise XOR operation. That is, x_1 has the β leftmost bits set to "0" and x_2 has the α rightmost bits set to "0."

The following algorithm of the proposed scheme is described as follows.

ICPT (Embedding Method)

Input: Block F of image G, secret key K, secret weight matrix W, secret message $b = b_{\alpha+\beta}b_{\alpha+\beta-1}...b_{\beta+1}b_{\beta}...b_2b_1$.
Output: Stego-block image F' (of stego image G').

Step 1. Compute $T = F \oplus K$.

Step 2. Compute $S = [W \bullet T]$, and represent S as $x_1 \oplus x_2$.

Step 3. Find $u = b_{\alpha+\beta}b_{\alpha+\beta-1}...b_{\beta+1}0...0$ by computing $u = b \wedge ((2^{\alpha} -1) \times 2^{\beta})$

Find $v = 0...0b_{\beta}...b_2b_1$ by computing $v = b \wedge (2^{\beta} -1)$, so that $b = u \oplus v$.
Consider x_1 and x_2.

For x_1, there are two cases:

a) if $x_1 = u$, then keep L_1 intact;

b) if $x_1 \neq u$, then compute $e = u \oplus x_1 \in M_1$, e $\neq 0$,

find a pixel F_{ij} in L_1 such that $e = W_{ij}$ and flip F_{ij}.

For x_2, there are two cases:

a) if $x_2 = v$, then keep L_2 intact;

b) if $x_2 \neq v$, then compute $e = v \oplus x_2 \in M_2$, $e \neq 0$,

find a pixel F_{kl} in L_2 such that $e = W_{kl}$ and flip F_{kl}.

Example 2 (ICPT: Embedding): Let F, K, and W be a binary block, key matrix, and weight matrix, respectively, all having the size 3 × 3. We can embed a 4-bit stream, $b = b_4b_3b_2b_1$, into F. For this example, let b be 1011_2. We can find $p = 9+2 = 11 = 4+4+3 > 2^2 + 2^2$; therefore, $\beta = \alpha = 2$ and $m(p) = 4$. Let $L_1 = \{F_{11},F_{12},F_{13},F_{21},F_{22}\}$, $L_2 = \{F_{23},F_{31},F_{32},F_{33}\}$, $M_1 = \{12,8,4\}$, and $M_2 = \{3,2,1\}$, or, in a binary presentation, $M_1 = \{1100,1000,0100\}$ and $M_2 = \{0011,0010,0001\}$.

F_1

$$F=\begin{array}{|c|c|c|}\hline 1&0&1\\\hline 0&0&1\\\hline 1&0&1\\\hline\end{array}\quad K=\begin{array}{|c|c|c|}\hline 0&1&1\\\hline 1&0&1\\\hline 0&1&1\\\hline\end{array}\quad W=\begin{array}{|c|c|c|}\hline 8&12&4\\\hline 4&8&3\\\hline 1&2&3\\\hline\end{array}\quad T=F\oplus K=\begin{array}{|c|c|c|}\hline 1&1&0\\\hline 1&0&0\\\hline 1&1&0\\\hline\end{array}\quad S=[W\bullet T]=\begin{array}{|c|c|c|}\hline 8&12&0\\\hline 4&0&0\\\hline 1&2&0\\\hline\end{array}$$

$S = [W\bullet T] = 8\bullet 1+12\bullet 1+4\bullet 1+1\bullet 1+2\bullet 1 = 1000\oplus 1100\oplus 0100\oplus 0001\oplus 0010=0011$

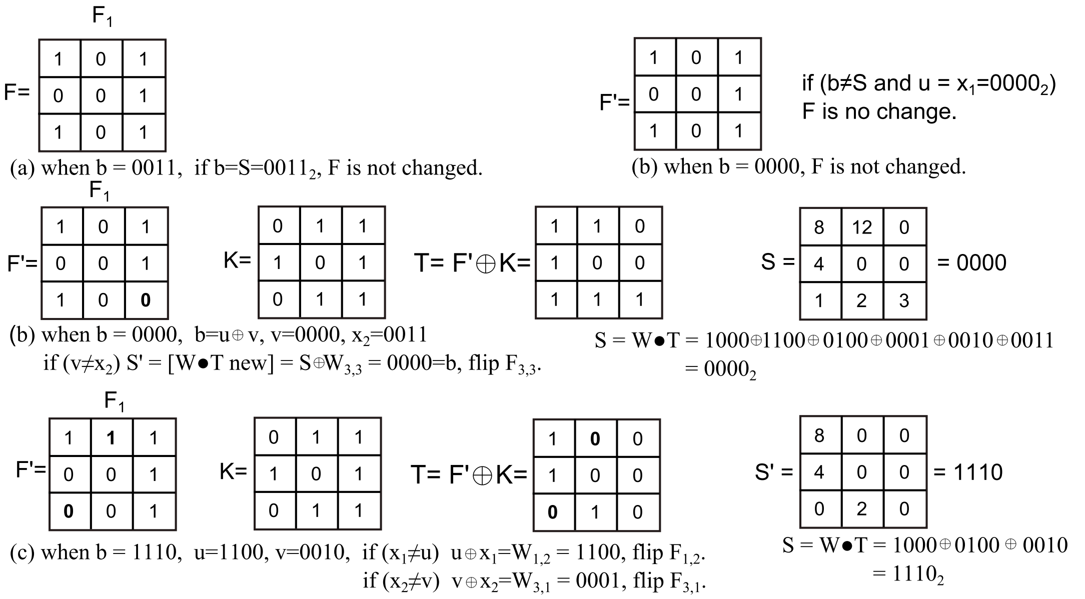
$m(p) = \alpha+\beta = 4$, $S = x_1\oplus x_2$, $x_1=0000$, $x_2 = 0011$

Fig. 8. Example of embedding procedure with block F, key K, and weight matrix W

We will determine the value of S based on F, K, and W by computing $S = [W\bullet T]$,
$S = 8\bullet 1+12\bullet 1+4\bullet 1+1\bullet 1+2\bullet 1 = 1000\oplus 1100\oplus 0100\oplus 0001\oplus 0010 = 0011$, where $T = F\oplus K$.

It is possible to express S as $x_1 \oplus x_2$, where $x_1 = S \wedge ((2^2 - 1) \times 2^2) = 0011 \wedge 1100 = 0000$ and $x_2 = S \wedge (2^2 - 1) = 0011 \wedge 0011 = 0011$.

1) Assuming that $b = 0011$: if $b = S$, F is not changed (see **Fig. 9**(a)).
2) Assuming that $b = 0000$: by Step 3, compute $b = u \oplus v$, $u = 0000$, and $v = 0000$. Because $u = x_1$, we keep L_1 intact. Because $v \neq x_2$, we need to change one entry in L_2. To do this, we find any pixel $F_{i,j}$ in L_2 such that $e = W_{i,j}$, where $e = v \oplus x_2 = 0000 \oplus 0011 = 0011$; in this case we flip $F_{3,3}$ (see **Fig. 9**(b)**)**. Therefore, $T_{3,3}$ is changed to $T_{3,3} \oplus 1 = 1$, and the new sum $S' = [W \bullet T'] = S \oplus W_{3,3} = 0000 = b$.
3) Assuming that $b = 1110$, compute $b = u \oplus v$, $u = 1100$, and $v = 0010$. Because $u \neq x_1$, and $v \neq x_2$, we need to change one entry in L_1 and another in L_2. In L_1, compute $u \oplus x_1 = 1100 = W_{1,2}$ and then flip $F_{1,2}$. In L_2, compute $v \oplus x_2 = 0001 = W_{3,1}$ and then flip $F_{3,1}$ (see **Fig. 9**(c)**)**. Hence, T has two new entries, $T'_{1,2} = T_{1,2} \oplus 1 = 0$; $T'_{3,1} = T_{3,1} \oplus 1 = 0$, and the new sum $S' = [W \bullet T'] = S \oplus W_{1,2} \oplus W_{3,1} = 0011 \oplus 1100 \oplus 0001 = 1110 = b$, as claimed.



F_1

$F = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$

(a) when b = 0011, if b=S=0011_2, F is not changed.

$F' = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ if (b≠S and u = x_1=0000_2) F is no change.

(b) when b = 0000, F is not changed.

F_1

$F' = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & \mathbf{0} \end{bmatrix}$ $K = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ $T = F' \oplus K = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ $S = \begin{bmatrix} 8 & 12 & 0 \\ 4 & 0 & 0 \\ 1 & 2 & 3 \end{bmatrix} = 0000$

(b) when b = 0000, b=u⊕v, v=0000, x_2=0011
if ($v \neq x_2$) S' = [W●T new] = S⊕$W_{3,3}$ = 0000=b, flip $F_{3,3}$.

S = W●T = 1000⊕1100⊕0100⊕0001⊕0010⊕0011 = 0000_2

F_1

$F' = \begin{bmatrix} 1 & \mathbf{1} & 1 \\ 0 & 0 & 1 \\ \mathbf{0} & 0 & 1 \end{bmatrix}$ $K = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ $T = F' \oplus K = \begin{bmatrix} 1 & \mathbf{0} & 0 \\ 1 & 0 & 0 \\ \mathbf{0} & 1 & 0 \end{bmatrix}$ $S' = \begin{bmatrix} 8 & 0 & 0 \\ 4 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} = 1110$

(c) when b = 1110, u=1100, v=0010, if ($x_1 \neq u$) u⊕x_1=$W_{1,2}$ = 1100, flip $F_{1,2}$.
if ($x_2 \neq v$) v⊕x_2=$W_{3,1}$ = 0001, flip $F_{3,1}$.

S = W●T = 1000⊕0100⊕0010 = 1110_2

Fig. 9. Embedding procedure for three cases: (a) when $b = 0011$, (b) when $b = 0000$, and (c) when $b = 1110$.

3.2 Extraction Algorithm for ICPT Scheme

In this section, we will suggest an extraction algorithm for ICPT and explain it using pseudocode and an example.

ICPT (Extraction Method)

Input: Block F' in stego-image G', secret key K, weight matrix W, all having the same size, $m \times n$.

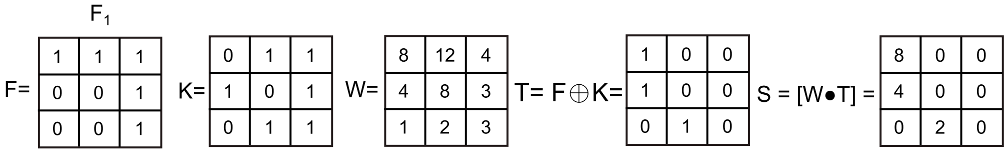
Output: Secret message $b = b_r b_{r-1} \ldots b_1$.

Step 1. Compute $T = F \oplus K$.

Step 2. Compute $S = [W \bullet T]$.

Step 3. Return S as the secret r-bit stream that was embedded in F.

Example 3 (ICPT: Extraction): Let F, K, and W be an image block, key matrix, and weight matrix, respectively. We can extract a 4-bit stream ($b = b_4b_3b_2b_1$) from a 3 × 3 block. We compute $T = F \oplus K$, as shown in **Fig. 10**. Next, we compute $S = [W \bullet T]$. Therefore, the hidden bits in F are "1110."



$$F_1 \quad F=\begin{bmatrix}1&1&1\\0&0&1\\0&0&1\end{bmatrix} \; K=\begin{bmatrix}0&1&1\\1&0&1\\0&1&1\end{bmatrix} \; W=\begin{bmatrix}8&12&4\\4&8&3\\1&2&3\end{bmatrix} \; T=F\oplus K=\begin{bmatrix}1&0&0\\1&0&0\\0&1&0\end{bmatrix} \; S=[W\bullet T]=\begin{bmatrix}8&0&0\\4&0&0\\0&2&0\end{bmatrix}$$

S = [W•T] = 8•1+4•1+2•1 = 1000⊕0100⊕0010 = 1110,
i.e., hidden bits are '1110'

Fig. 10 Extraction procedure for ICPT scheme example.

3.3 Computing Intensity of Texture

Most data hiding schemes for halftone images tend to generate many unpleasant clusters of pixels that resemble "salt-and-pepper" because these methods destroy the image texture, and the texture is related to the quality of an image. In this paper, we attempt to maintain textures during the data embedding process; when a pixel is flipped in a block, if it is composed of all white or all black pixels, then the block is not considered appropriate to conceal a bit because of the "salt-and-pepper" effect. To find good a place to hide information, we use the following equation.

$$con(m,n) = \sum_{i=0}^{7} w(i)h(x_0, x_i) \tag{6}$$

$$f(x_0, x_i) = \begin{cases} 1 & x = x_i \\ 0 & x \neq x_i \end{cases} \tag{7}$$

$$w(i) = \begin{cases} 1, & for\ i = 1,3,6,8 \\ 2, & for\ i = 2,4,5,7 \end{cases} \tag{8}$$

In **Eq. (6)**, $con(m,n)$ [1,7] computes the intensity and texture of a block in an error diffused halftone image. Variable w denotes the texture intensity, according to value i. When i is equal to 2, 4, 5, or 7, corresponding to above, left, right, or below a pixel in the block, respectively, then w is assigned to 2. Therefore, pixels that include 2, 3, 5, and 7 have many effects on the human visual system in comparison to the 1, 3, 6, and 8 position pixels in a block. A function $f(x_0, x_i)$ denotes whether there is a connection between x_0 and x_i. If there is a connection between them, then $f(x_0, x_i) = 1$, otherwise $f(x_0, x_i) = 0$.

If $con(m,n) = 0$ or $con(m,n) = 12$, then a block is black or white, respectively. Therefore, these blocks are not appropriate for concealing bits because of noise. In this case, we need a threshold to embed bit streams into a cover image without noise. To maintain the high quality of a halftone image, we do not conceal bit streams when $con(m,n) < 2$ and $con(m,n) > 10$. Even though this may maintain the quality of the stego image, it is not easy to conceal large bit streams. From a steganography point of view, **Eq. (6)** is required to control the quality of stego images when hiding data.

3.4 Data-Hiding Capacity of ICPT Scheme

For the ICPT scheme, we assume that matrix F is a block of size $m \times n$, $q = m \times n$, and $k > 1$, which is the number of colors. Secret bits can be embedded in F by changing at most two entries to obtain a new matrix F', which we call a "*configuration*." Because each entry can change in k-1 ways, the number of configurations that change one entry is at most $(k\text{-}1) \times q$. If two pixels are changed in F, there are at most $(k\text{-}1)^2 \times q \times (q\text{-}1)/2$ ways to change two entries in F. Therefore, in the ICPT scheme, there are at most $1 + (k\text{-}1) \times q + (k\text{-}1)^2 \times q \times (q\text{-}1)/2$ configurations. This means that we can hide at most $R = \lfloor log_2(1+(k\text{-}1) \times q + (k\text{-}1)^2 \times q \times (q\text{-}1)/2)\rfloor$ secret bits in F. We call R the Maximal Secret Data Ratio (MSDR) for the ICPT scheme. In particular, in the case of a halftone image (with $k = 2$), $MSDR = \lfloor log_2(1+q^2/2)\rfloor$ secret bits can be embedded in F.

Proposition 1. The number of bits embedded in F by ICPT is $m(p)$ and approximates the MSDR.

Proof. Consider $q = m \times n$, $p = m \times n + 2 = q + 2$.

From **Eqs. 4-5**, we deduce $m(p) \geqslant \log_2\lfloor((p/2)^2)\rfloor$ because, if p has a binary presentation $p = b_t b_{t-1}..b_1 b_0$ with $b_{t-1} = 1$, then $m(p) = 2t\text{-}1$ and $\lfloor\log_2((p/2)^2)\rfloor = 2t\text{-}2$, and with $b_{t-1} = 1$, $m(p) = \lfloor\log_2((p/2)^2)\rfloor = 2t\text{-}2$. By maximality, $MSDR \geqslant m(p)$. Hence, $MSDR = \lfloor\log_2(1+q(q+1)/2)\rfloor \geqslant m(p) \geqslant \lfloor\log_2((q+2)/2)^2\rfloor$. From the obvious inequation $(1+q(q+1)/2) < 2((q+2)/2)^2$ for any $q > 0$, $\log_2(1+q(q+1)/2) - \log_2(p/2)^2 = \log_2[(1+q(q+1)/2)/(p/2)^2] < \log 2^2 = 1$. Based on these inequalities, with some simple computations it is found that $MSDR = \lfloor\log_2(1+q(q+1)/2\rfloor \geq m(p) \geq \lfloor\log_2(1+q(q+1)/2\rfloor$ -1, which implies that $MSDR$ - m(p) ≤ 1. It follows that $MSDR - m(p)$ can be equal to 0, for example, with $q \geq 8$ satisfying $q + 1 = 2^t$, $p = 2^t + 1$ or $q + 1 = 2^t\text{-}1$, $p = 2^t$, then $MSDR = \lfloor\log_2(1+q(q+1)/2\rfloor = 2t\text{-}2 = m(p)$.

The proof is complete.

4. Experimental Results

The DHST, DHPT, CPT, DHSPT, *Kim et al.*, PI-Preserve, BOPC, and ICPT methods are applied to halftone images generated with error diffusion. The DHST, DHPT, CPT, DHSPT, *Kim et al.*, and ICPT methods are also applied to halftone images generated with ordered dithering. To reasonably evaluate the halftone images, we apply a low-pass filter (LPF) (a Gaussian LPF with a 7 × 7 square matrix and a standard deviation of 2.0) simulating HVS to measure the visual quality. The test images are 512 × 512 halftone images (see **Fig. 11**), which were obtained using Steinberg-kernel error diffused dithering [2] from 8-bit gray scale images. In the experiments, the objective quality measure used is the modified peak signal-to-noise ratio (MPSNR) (**Eq. (11)**), which is the PSNR (**Eqs. (9)** and **(10)**) between the original multi-tone image and the lowpass filtered halftone images.

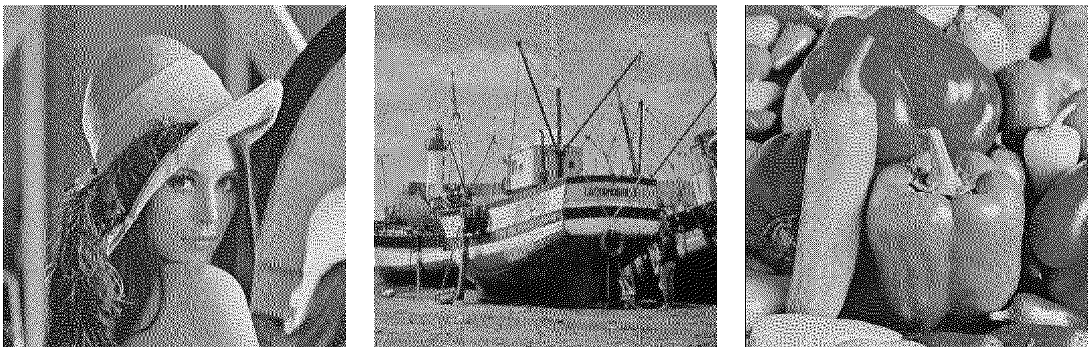


Fig. 11. Original *Error Diffusion* halftone images used in experiment: (a) Lena, (b) Boat, and (c) Peppers

PSNR is the most popular criterion for measuring the distortion between the original image and shadow images. It is defined as follows:

$$PSNR(I, I') = 20 \bullet \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right), \tag{9}$$

where *MSE* is the mean square error between the original grayscale image and the shadow image:

$$MSE = \frac{1}{mn}\sum_{i=1}^{m-1}\sum_{j=1}^{n-1}[I(i, j) - I'(i, j)]^2, \tag{10}$$

The symbols *I*(*i*, *j*) and *I*'(*i*, *j*) represent the pixel values of the original grayscale image and the stego image at position (*i*, *j*), respectively, and *m* and *n* are the width and height of the original image.

The *MPSNR* is a quality metric that attempts to model the human visual system. First, a simple inverse halftone I'_{low} is generated with a low pass filter. For our purposes, a 7 × 7 Gaussian low-pass filter worked well [13]. The matrix I'_{low} is then fed into the *PSNR* function to produce *MPSNR* (see **Eq. (11)**). This function allows for automated algorithm testing. Note that *MPSNR*, like *PSNR*, measures the relative visual quality, meaning that *MPSNR* measurements can only be used for comparisons of variations of the same image.

$$MPSNR(I, I') = PSNR(I, I'_{low}) \tag{11}$$

Table 2 shows an *MPSNR* comparison between DHST, DHPT, DHSPT, *Kim et al*. [12], PI-Preserve, BOPC, and our proposed scheme (ICPT) for an error diffusion halftone. ICPT had a better *MPSNR* than DHST, DHPT, CPT, DHSPT, *Kim et al*., PI-Preserve, and BOPC. These results show that the visual quality values of DHPT, CPT, DHSPT, and *Kim et al*. are lower than those of PI-Preserve, BOPC, and ICPT. However, the *MPSNR* of ICPT was similar to those of PI-Preserve and BOPC. In the ICPT scheme, if the size of the block increased, then the embedding bits also increased, whereas the error bit rates decreased because the error bit was 2 without the size of a block. Therefore, ICPT shows a high *MPSNR* and a higher subjective quality with few "salt-and-pepper" effects. The PI-preserve scheme preserved the local pattern and intensity, so it is possible to preserve the good quality. BOPC achieves better visual quality than DHSPT by reducing the number of smart pair toggles under the same payload situation.

Table 2. *MPSNR* values of various algorithms based on error diffusion halftone image. (Steinberg kernel)

Method	MPSNR (*dB*)				
	Payload (bit)	Lena	Boat	Peppers	Ave. MPSNR
Original		30.74	30.32	30.56	30.54
DHST	4096	29.50	29.18	29.39	29.36
DHPT	4096	29.54	29.02	29.46	29.34
CPT	4096	29.40	29.12	29.28	29.26
DHSPT	4096	30.04	29.52	29.96	29.84
Kim et al.	4096	30.36	30.07	30.29	30.24
PI-Preserve	4096	30.54	30.14	30.39	30.36
BOPC	4096	30.55	30.14	30.38	30.36
ICPT	4096	30.55	30.16	30.39	30.37

We also experimented with data hiding using the order dithering halftone images in **Fig. 12**. To generate an ordered dithering halftone image, we use a grayscale image with the bayer-5 screen method, which is very commonly used and produces a cross-hatch pattern in the resulting image. However, the ordered dithering technique is very fast and powerful. Ordered dithering images show high frequency characteristics. As a result, the error diffusion is more complicated than with ordered dithering, but it can yield a higher visual quality.

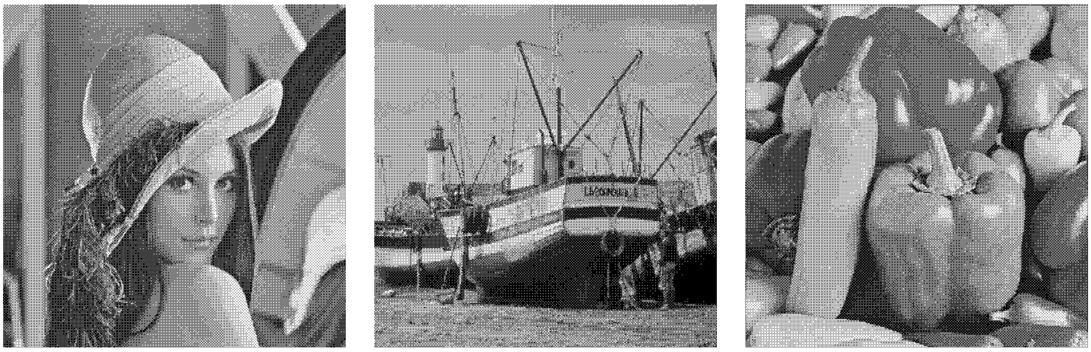


Fig. 12. Original Ordered Dither halftone images used in the experiment: (a) Lena, (b) Boat, and (c) Peppers.

Table 3 shows a comparison of the *MPSNR* values between ICPT and the previous schemes, including DHST, DHPT, CPT, DHSPT, and *Kim et al*. This comparison shows that ICPT has a slightly higher *MPSNR* than the previous schemes such as DHSPT and *Kim et al*. However, most schemes such as DHSPT, *Kim et al*., and ICPT have some “salt-and-pepper” pieces. DHSPT has a better quality than DHPT, CPT, and DHST because of the smart slave pixel choices. ICPT can control the quality as the block size increases because the number of flips is always two at most, regardless of the size of the block.

Table 4 compares the embedding capacities of the proposed and previous schemes such as

DHST, DHPT, DHSPT, and *Kim et al*. When the size of *F* is increased from 6 to 30 and 63, ICPT's embedding capacity is almost the same as the *MSDR*. DHSPT can conceal 1 bit in a 3 × 3 block, so it is possible to embed 10 bits when *F* is 94, but the number of flipping pixels is also increased. *Kim et al*.'s scheme shows a high embedding capacity, i.e., when the size of *F* is 94, the hidden bits will be 24.

Table 3. *MPSNR* values of various algorithms based on Ordered Dither halftone image.

Method	MPSNR (*dB*)				
	Payload (bit)	Lena	Boat	Peppers	Ave. MPSNR
Original		27.22	26.87	27.01	27.03
DHST	4096	26.50	26.15	26.48	26.37
DHPT	4096	26.72	26.37	26.70	26.59
CPT	4096	26.49	26.20	26.29	26.32
DHSPT	4096	26.79	26.57	26.77	26.71
Kim et al.	4096	26.83	26.63	26.84	26.76
ICPT	4096	26.87	26.66	26.91	27.81

ICPT flips no more than 2 pixels regardless of the size of *F*. Furthermore, ICPT shows a high embedding rate with a weighted block. To embed 24 bits of secret bit messages in a block using *Kim et al*.'s scheme, the size of *F* should be 94, whereas ICPT needs 4 blocks, each with an *F* of 12 pixels. Therefore, ICPT provides a better performance than the other schemes. *MSDR* is the Maximal Secret Data Ratio (see **Section 3.4**); it is the mathematically determined maximum data-hiding capacity.

Table 4. Comparison of embedding capacities of DHSTP, *Kim et al*., CPT, and ICPT (block *F* from a halftone image).

Number of pixels in F	MSDR	DHSPT		*Kim et al*.'s		CPT		ICPT	
		bits	flipping	bits	flipping	bits	flipping	bits	flipping
5	4	0	0	0	0	2	2	3	2
6	4	0	0	0	0	2	2	4	2
9	5	1	2	0	0	3	2	4	2
12	6	1	2	0	0	3	2	6	2
30	8	3	6	8	2	4	2	8	2
46	10	5	10	12	3	5	2	9	2
63	10	7	14	16	4	6	2	10	2
94	12	10	20	24	6	6	2	11	2

When *F* is 5 pixels, *MSDR* is 4 and the embedding capacity of ICPT is 3. That is, ICPT's embedding capacity is higher than any of the other methods. Therefore, ICPT has a good performance with respect to its embedding capacity.

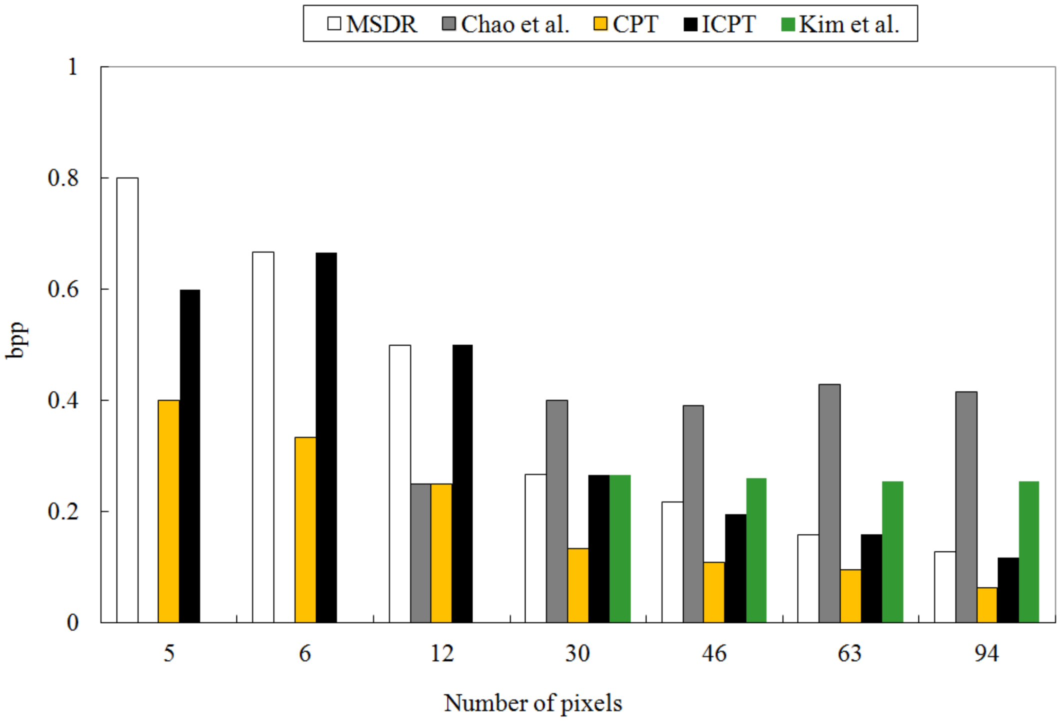


Fig. 13. Comparison of embedding rates among MSDR, *Chao et al*., CPT, ICPT, and *Kim et al*., when block size is increased.

Fig. 13 compares the embedding rates of the previous schemes and the proposed scheme. ICPT can conceal 0.67 bpp, when the number of pixels in a block is 6. In this case, ICPT has a high performance embedding rate. In the ICPT scheme, if the size of a block is increased, the embedding bits are increased, whereas the embedding rate for an image and the error bit rate will decrease, because the error bits are always 2 regardless of the size of a block. On the other hand, the CPT scheme shows a high embedding rate when the block size is 5. That is, the embedding rate is 0.4 bpp. *Kim et al.'s* scheme can hide 4 bits in a block composed of 15 pixels. Therefore, the embedding rate is always 0.26 bpp without considering the size of a block. *Chao et al*.'s scheme shows an embedding rate of 0.42 bpp when the block size is 63.

Fig. 14 shows the *MPSNR* values of the test image "Lena" based on an error diffused halftone image (Steinburg kernel) with hidden bits embedded by DHSPT, BOPC, and ICPT. It is clear that BOPC and ICPT show better visual quality than DHSPT. ICPT and BOPC show the same *MPSNR* when embedding 4096 bits. In a case of embedding 10,000 bits, ICPT is slightly lower than BOPC, with a difference of 0.05 dB. Therefore, there are few differences between the *MPSNR* values of ICPT and BOPC. However, the embedding rate of ICPT is higher than any other scheme. ICPT will be a relatively useful scheme if the embedding rate is more important than the image quality.

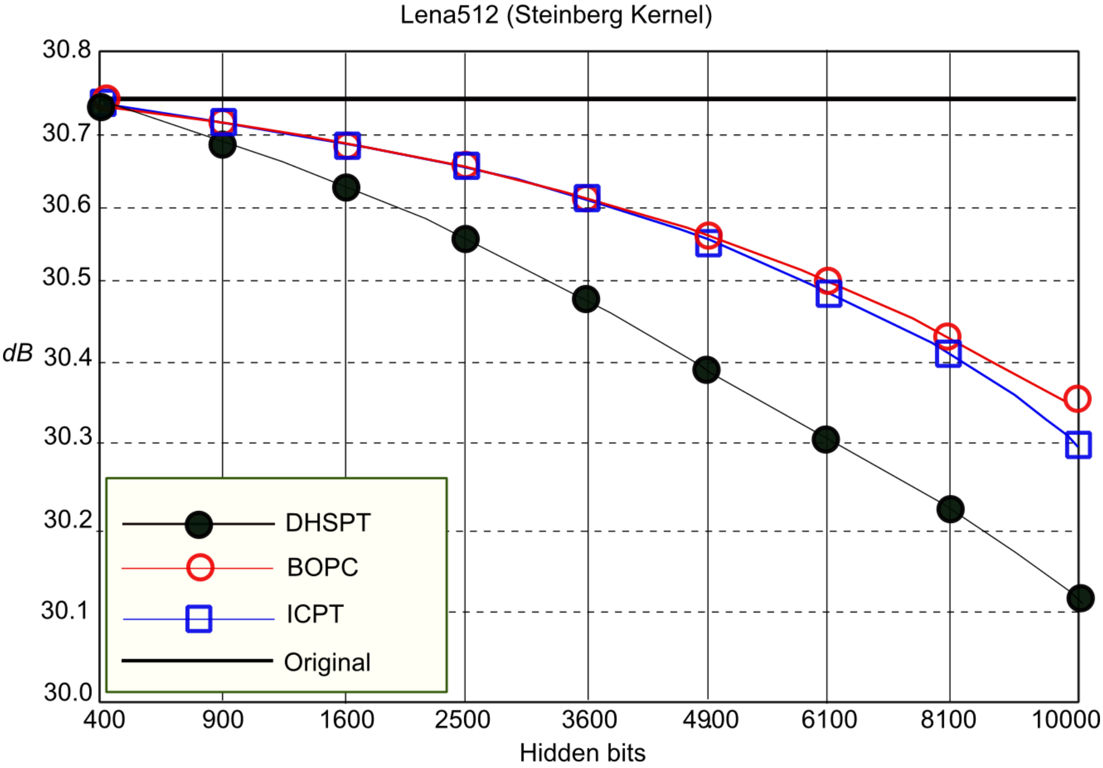


Fig. 14. Comparison of DHSPT, BOPC and ICPT for *Lena* (Steinberg kernel).

Fig. 15 shows the Steinberg-kernel error diffused "Lena," "Boat," and "Pepper" images with 4096 bits embedded using our proposed method. The size of the test images is 512 × 512. ICPT has a better quality with some "salt-and-pepper" pieces.

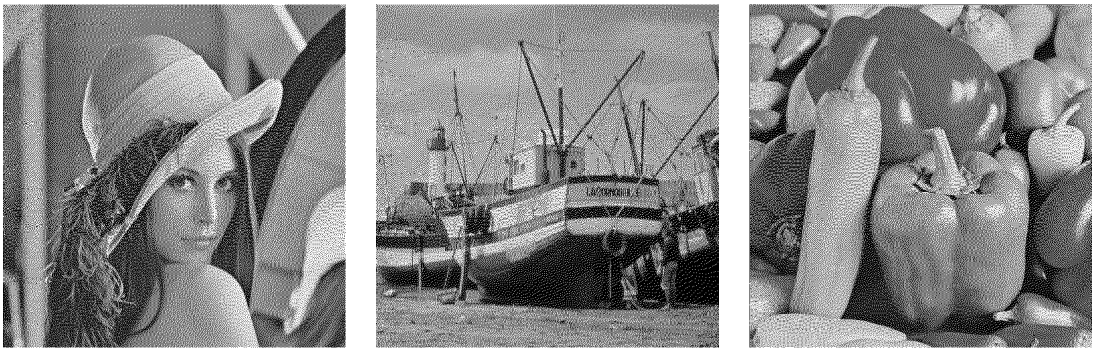


Fig. 15. Experiment results: Stego images (Steinberg kernel): (a) Lena, (b) Boat, and (c) Pepper.

Fig. 16 shows the ordered dithered (using Bayer-5 screen) "Lena," "Boat," and "Pepper" images with 4096 bits embedded. These results show that the visual quality is reasonable with some "salt-and-pepper" pieces. Ordered dither is the name given to any dither process that uses a period deterministic dither array. It is the "ordered" nature of the elements in the array that contrast it with the random nature of white noise dithering. These experiments show that when our proposed scheme and Eq. (6) are used, the “salt-and-pepper” pieces can be controlled and removed effectively. Finally, we overcome the weakness of the previous schemes, i.e., the small number of bits that can be hidden in a halftone image. Moreover, ICPT has the advantage of being able to control the visual quality.

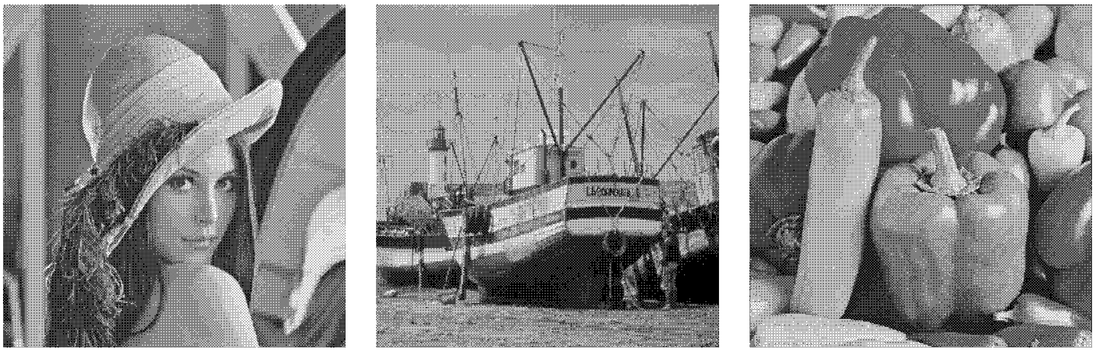


Fig. 16. Experiment results: Stego images (Ordered dithering–*Bayer-5 Screen*): (a) Lena, (b) Boat, and (c) Pepper.

5. Conclusion

Halftoning is a technique for changing multi-tone images into two-tone binary images, which look like the original multi-tone images when viewed from a distance. Halftone images are commonly used in print books, newspapers, magazines, and fax documents. The proposed technique can be applied to mobile phone systems for image authentication or to convey secret messages because mobile phones have computational limits and the proposed scheme uses simple computations. In this paper, we proposed good methods for hiding data in halftone images. When the original grayscale or color image is not available, the proposed ICPT can hide a large amount of data in halftone images using a matrix encoding technique and a smart choice for the pair-toggling candidates. Our experimental results suggest that the resulting image quality can be good. When the original multi-tone image is available and the halftoning method is error diffusion, then the proposed ICPT method can hide data in halftone images with its distortion diffused to the surrounding pixels. The main idea is to use a weight matrix to increase the data hiding ratio. Experiments showed that our scheme is more efficient than the existing schemes. In addition, our ICPT scheme provides reasonably good image quality.

Acknowledgement

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) by the Ministry of Education, Science and Technology (20120192).

References

[1] J.M. Guo, “Watermarking in dithered halftone images with embeddable cells selection and inverse halftoning,” *Signal Processing*, vol.88, pp.1496-1510, 2008. Article (CrossRef Link).

[2] J.S. Pan, H. Luo, & Z.H. Lu, "Look-up Table Based Reversible Data Hiding for Error Diffused Halftone Images," *INFORMATICA*, vol.18, pp.615-628, 2007.
http://130.203.133.150/viewdoc/summary?doi=10.1.1.137.9784

[3] H.W. Tseng & C.C. Chang, "Hiding data in halftone images," *INFORMATICA*, vol.16, pp.419-430. 2005. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.112.9075

[4] R.W. Floyd & L. Steinberg, "An adaptive algorithm for spatial grey scale," *Proceedings of the Society of Information Display*, vol.17, pp.75-77, 1976.
http://en.wikipedia.org/wiki/Floyd%E2%80%93Steinberg_dithering

[5] R. Ulichney, "A Review of Halftoning Techniques," in *Proc. of SPIE – The International Society for Optical Engineering*, vol.3963, 2000.
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.6726

[6] R.M. Chao, Y.A. Ho, & Y.P. Chu, "Data Hiding Scheme Using Covering Codes in Halftone Images Based on Error Diffusion," in *Proc. of Asia-Pacific Services Computing Conference*, APSCC, vol.1, pp. 1483-1488, 2008. Article (CrossRef Link)

[7] M.S. Fu & O.C. Au, "Data Hiding Watermarking for Halftone Images," *IEEE Transaction on Image Processing*, vol.11, pp.477-484, 2002. Article (CrossRef Link)

[8] Y.-C. Tseng & H.-K. Pan, "Secure and Invisible Data Hiding in 2-Color Images," in *Proc. of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2001.*, vol.3, pp. 887-896, 2001. Article (CrossRef Link)

[9] Y. Chen, H. Pan, & Y. Tseng, "A secure data hiding scheme for two-color images," in *Proc. of the Fifth IEEE Symposium on Computers and Communications (ISCC 2000)*, vol. 1, pp. 750-755, 2000. Article (CrossRef Link)

[10] R.Y.M. Li, O.C. Au, C.K.M. Yuk, S.K. Yip, & S.Y. Lam, "Halftone Image Data Hiding with Block-Overlapping Parity Check," *Acoustics, Speech and Signal Processing*, 2007. ICASSP 2007, vol.2, pp.193-196, 2007. Article (CrossRef Link)

[11] S.K. Yip, O.O. Au, C.W. Ho, & H.M. Wong, "PI-preserve data hiding for halftone image," *Intelligent Signal Processing and Communication Systems*, 2005. ISPACS 2005., vol.1 pp.125-128, 2005. Article (CrossRef Link)

[12] C. Kim, D.K. Shin, & D.I. Shin, "Data Hiding in a Halftone Image Using Hamming Code (15, 11)," *Lecture Notes in Computer Science*, vol.6592, pp.372-381, 2011. Article (CrossRef Link)

[13] R. C. Gonzalez & R E. Woods, "Digital Image Processing, 2nd Edition," *Prentice-Hall*, Inc. 2002. http://www.amazon.com/Digital-Image-Processing-2nd-Edition/dp/0201180758

[14] C. Qin, C. C. Chang & P. L. Tsou, "Perceptual Image Hashing Based on the Error Diffusion Halftone Mechanism," *International Journal of Innovative Computing, Information and Control*, vol.8, no. 9, pp. 6161-6172, 2012. http://www.ijicic.org/ijicic-11-05009.pdf

[15] C. Qin, Z. H. Wang, C. C. Chang, & K. N. Chen, "Reversible Data Hiding Scheme Based on Image Inpainting," *Fundamenta Informaticae*, vol.120, no. 1, pp. 59-70, 2012. Article (CrossRef Link)

[16] Z. Zhao, H. Luo, Z.M. Lu, & J.S. Pan, "Reversible Data Hiding Based on Multilevel Histogram Modification and Sequential Recovery," *International Journal of Electronics and Communications*, vol.65, no. 10, pp. 814-826, 2011. Article (CrossRef Link)

[17] C.Y. Yang, C.H. Lin, & W.C. Hu, "Reversible Data Hiding by Adaptive IWT-coefficient Adjustment," *Journal of Information Hiding and Multimedia Signal Processing*, vol.2, no. 1, pp. 24-32, 2011. http://bit.kuas.edu.tw/~jihmsp/2011/vol2/JIH-MSP-2011-02-002.pdf

[18] P. Tsai, Y. C. Hu, & H.L. Yeh, "Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting," *Signal Processing*, vol.89, no. 6, pp.1129-1143, 2009. Article (CrossRef Link).

[19] M.S. Fu & O.C. Au, "Hiding data in halftone image using modified data hiding error diffusion," in *Proc. SPIE Visual Communications and Image Processing*, vol.4067, pp.1671–1680, 2000. Article (CrossRef Link).

[20] C.W. Wu, "Multimedia data hiding and authentication via halftoning and coordinate projection," *EURASIP Journal on Advances in Signal Processing*, vol.2, pp. 143–151, 2002. Article (CrossRef Link).

[21] H.Y. Kim & A. Afif, "A secure authentication watermarking for halftone and binary images," *International Journal of Imaging Systems and Technology*, vol.14, pp. 147–152, 2004. Article (CrossRef Link).

[22] S.C. Pei, J.M. Guo, & H. Lee, "Novel robust watermarking techniques in dithering halftone images," *IEEE Signal Processing Letters*, vol.12, no.4, pp. 333–336, 2005. Article (CrossRef Link).

[23] S.C. Pei & J.M. Guo, "Data hiding in halftone images with noise-balanced error diffusion," *IEEE Signal Processing Letters*, vol.10, no.2, pp. 349–351, 2003. Article (CrossRef Link).

Phan Trung Huy received B.Sc. degree in Mathematics 1976 from Hanoi Pedagogical University No.1, Ph.D. Degree in Mathematics 1992 from Vietnam Institute of Mathematics. Since 2004 he is an Associate Professor at Institute of Applied Mathematics and Informatics (the old name: Faculty of Mathematics and Informatics), Hanoi University of Science and Technology. His research interests include Formal Languages and Automata, Theory of Varieties of Finite Monoids and Varieties of Regular Languages and ω-Regular Languages, Theory of Codes, Combinatorics on Words, Graph Theory, Cryptography, Data Hiding, Quantum Algorithms and Simulation, Pattern Matching. **Email:** huy.phantrung@hust.edu.vn

Nguyen Hai Thanh received B.Sc. Degree 1996 and M.Sc. Degree 1999 in Computer Science from Hanoi National University, Ph.D. Degree 2012 in Mathematical Foundation of Computer Science. Now he is an officer of the Science, Technology and Environment Department - Vietnam Ministry of Education and Traning. His reseach interests include Formal Languages and Automata, Pattern Matching, Theory of Codes, Data Hiding, Cryptography, Digital Image Processing. **Email**: thanhk37@yahoo.com

Cheonshik Kim received his B.S. degree in Computer Engineering from Anyang University, Korea, in 1995; his M.S. degree in Computer Engineering from Hankuk University of Foreign Studies (HUFS), Korea, in 1997; and his Ph.D. degree in Computer Engineering from HUFS in 2003. From March 2010, he was a professor of Department of Computer Science, Sejong University, Korea. He won a research award from the IEEK in 2012. He has served as the Editor for ICACT Transcation on Advanced Communications Technology (TACT) since 2012. He has program chair of International conference, GPC 2013, IEEK Computer Society in varous capacities, including Vice-President since 2010. He is a member of IEEE. His research interests include Multimedia Systems, Data Hiding, and Watermarking. His research is supported by NRF.

Ching-Nung Yang received the B.S. degree and the M.S. degree, both from Department of Telecommunication Engineering at National Chiao Tung University. He received Ph.D. degree in Electrical Engineering from National Cheng Kung University. He is presently a professor in the Department of Computer Science and Information Engineering at National Dong Hwa University, and is also an IEEE senior member. He has published a number of journal and conference papers in the areas of information security, multimedia security and coding theory. He is the guest editor of a special issue on "Visual Cryptography Scheme" for 2 Communication of CCISA, and a coauthor of series of articles on "Image Secret Sharing" for the Encyclopedia of Mutimedia. He is the coeditor of the book "Visual Cryptography and Secret Image Sharing" published by CRC Press/Taylor & Francis. He serves as a technical reviewer for over 30 major scientific journals in the areas of his expertise, and serves as editorial boards of some journals. Also, has served member of program committees of various international conferences committees. He is the recipient of the 2000, 2006, 2010, and 2012 Fine Advising Award in the Thesis of Master of Science awarded by Institute of Information & Computer Machinery.