(n, k, p)-Gray Code for Image Systems

Yicong Zhou, *Member, IEEE*, Karen Panetta, *Fellow, IEEE*, Sos Agaian, *Senior Member, IEEE*, and C. L. Philip Chen, *Fellow, IEEE*

***Abstract*—This paper introduces a new parametric n-ary Gray code, the (n, k, p)-Gray code, which includes several commonly used codes such as the binary-reflected, ternary, and (n, k)-Gray codes. The new (n, k, p)-Gray code has potential applications in digital communications and signal/image processing systems. This paper focuses on three illustrative applications of the (n, k, p)-Gray code, namely, image bit-plane decomposition, image denoising, and encryption. The computer simulations demonstrate that the (n, k, p)-Gray code shows better performance than other traditional Gray codes for these applications in image systems.**

***Index Terms*—Bit-plane decomposition, image denoising, image encryption, (n, k, p)-Gray code.**

I. Introduction

GRAY CODE, named after a Bell Laboratories researcher, Frank Gray, generally refers to a binary-reflected Gray code (BRGC) in which two successive codes differ in only one bit position [1]. Generalization of Gray codes has been done by arranging some combinatorial objects such that any two consecutive elements in the list differ in some *a priori* properties [2] or by listing subsets of the binary n-tuples in a Gray code manner such that the list has more predefined properties [3]. Balanced Gray code [4] is an example of this generalization. The concept of Gray code has been extended to any single distance code in which each code word differs from the next in only one digit. Based on this, non-Boolean Gray codes such as n-ary Gray code [5], [6], whose code words are non-Boolean values, have been generated. For example, the 3-ary (ternary) Gray code is an n-ary Gray code with sequence elements {0, 1, 2}. The (n, k)-Gray code is a type of n-ary Gray code with the base-n and k digits [7], [8]. A survey of combinatorial Gray codes is discussed in [9].

Manuscript received November 13, 2011; revised May 16, 2012; accepted July 20, 2012. Date of publication August 22, 2012; date of current version April 16, 2013. This work was supported in part by the Research Committee of the University of Macau under Grants SRG007-FST12-ZYC and MYRG113(Y1-L3)-FST12-ZYC. This paper was recommended by Associate Editor D. Goldgof.

Y. Zhou and C. L. P. Chen are with the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: yicongzhou@umac.mo; philipchen@umac.mo).

K. Panetta is with the Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155 USA (e-mail: karen@eecs.tufts.edu).

S. Agaian is with the Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: sagaian@utsa.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2012.2210706

Applications of those existing Gray codes have been found in many different areas, including digital circuit designs [10], digital communication systems [11]–[15], and image processing, including image filtering [16], compression [17], recognition [18], scrambling [19], [20], and watermarking [21], as well as motion estimation for video processing [22] and image stabilization [23].

To offer Gray code more general properties and make it more suitable for applications in image systems, this paper introduces a new type of n-ary Gray code called the (n, k, p)-Gray code. It integrates a new distance parameter p with the concept of the (n, k)-Gray code. The new (n, k, p)-Gray code changes as the values of the base n and the distance parameter p vary. Several examples will be provided in this paper to demonstrate its applications in image systems.

The rest of this paper is organized as follows. Section II introduces the new (n, k, p)-Gray code. Section III introduces the (n, k, p)-Gray-code bit-plane decomposition after reviewing two traditional bit-plane decomposition methods. Section IV introduces a new image denoising algorithm and provides several simulation results to show the application of the (n, k, p)-Gray code in image denoising. Section V introduces several (n, k, p)-Gray-code transforms and then introduces a new image encryption algorithm using these transforms. The comparison and analysis are also provided to show the new algorithm's performance. Section VI draws a conclusion.

II. (n, k, p)-Gray Code

Extending the concept of the (n, k)-Gray code with an additional distance parameter p, this section introduces a new type of Gray code, called the (n, k, p)-Gray code. It is a new type of non-Boolean Gray code when its base is greater than two. It changes as the values of its base n and the distance parameter p vary. The new (n, k, p)-Gray code is defined as follows.

Definition 2.1 (The (n, k, p)-Gray Code): The sequences $(a_{k-1}, \ldots, a_1, a_0)_n$ and $(g_{k-1}, \ldots, g_1, g_0)_n$ are the k-digit base-n representations of the nonnegative integers A and G, respectively, i.e., $A = \sum_{i=0}^{k-1} a_i n^i$ and $G = \sum_{i=0}^{k-1} g_i n^i$. G is called the (n, k, p)-Gray code of A if the sequences are satisfied with

$$g_i = \begin{cases} a_i, & \text{if } i > k-p-2 \\ (a_i + a_{i+p+1}) \bmod n, & \text{if } 0 \leqslant i \leqslant k-p-2 \end{cases} \tag{1}$$

where $0 \leqslant i \leqslant k-1$, $n \geqslant 2$, and $0 \leqslant p \leqslant k-2$.

TABLE I
EXAMPLES OF (n,k,p)-GRAY CODE WITH BINARY AND NONBINARY BASES FOR INTEGERS FROM 0 TO 20

A	(n,k,p)-Gray code of A			
	n=2,p=0	n=2,p=2	n=3,p=0	n=3,p=1
1	00001	00001	001	001
2	00011	00010	002	002
3	00010	00011	011	010
4	00110	00100	012	011
5	00111	00101	010	012
6	00101	00110	022	020
7	00100	00111	020	021
8	01100	01001	021	022
9	01101	01000	110	101
10	01111	01011	111	102
11	01110	01010	112	100
12	01010	01101	121	111
13	01011	01100	122	112
14	01001	01111	120	110
15	01000	01110	102	121
16	11000	10010	100	122
17	11001	10011	101	120
18	11011	10000	220	202
19	11010	10001	221	200
20	11110	10110	222	201

The definition of the (n,k,p)-Gray code in (1) can be represented in the matrix format. For example, if $p=0$, it can be written as

$$\begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{k-2} \\ g_{k-1} \end{pmatrix} = \left(\begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 1 & \cdots & 0 \\ \vdots & & & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & 1 \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{k-2} \\ a_{k-1} \end{pmatrix} \right) \bmod n. \tag{2}$$

Moreover, if $p=2$, it will be

$$\begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-2} \\ g_{k-1} \end{pmatrix} = \left(\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & & 1 & 0 \\ 0 & 0 & \cdots & 0 & & 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-2} \\ a_{k-1} \end{pmatrix} \right) \bmod n. \tag{3}$$

By selecting values of the base n and distance parameter p, the (n,k,p)-Gray code generates different Gray codes, including several traditional Gray codes in [1], such as the following.

1) For $p=0$, the (n,k,p)-Gray code reverts to the (n,k)-Gray code.
2) For $n=2$ and $p=0$, the (n,k,p)-Gray code becomes the BRGC.
3) For $n=3$ and $p=0$, the (n,k,p)-Gray code is back to the traditional ternary Gray code.
4) If n is another value, the (n,k,p)-Gray code will be a new type of Gray codes.

Table I gives some examples for the (n,k,p)-Gray code of the integers from 0 to 20 with different n and p values.

It can be observed in Table I that the (n,k,p)-Gray code with a base equal to three demonstrates that the new code differs from the BRGC in that it does not satisfy either the unit distance property or the adjacency property of two adjacent code words that differ by exactly one element, as is the case in the BRGC [8].

The presented (n,k,p)-Gray code has several potential applications in digital communication, and signal and image processing. In this paper, we focus on its applications in image processing systems.

III. (n,k,p)-GRAY CODE FOR BIT-PLANE DECOMPOSITION

There are two traditional methods for the image bit-plane decomposition: binary bit-plane decomposition and Gray-code bit-plane decomposition [24]. Both methods intend to decompose an image into several binary bit planes. The higher order bit planes consist of the higher significant bits of each image pixel, which contain almost all the significantly visual data. The lower order bit planes collect the less significant bits of image pixels, which describe more of the image details. Binary bit-plane decomposition has been used for image systems such as edge detection [25] and image coding and compression [26]–[28], as well as for security applications such as image encryption [29]–[31], data hiding [32], [33], watermarking [34], and steganography [35], [36]. As an alternative method, Gray-code bit-plane decomposition can reduce the effect of small gray-level changes due to the fact that two successive Gray codes differ in only one bit position. For example, the Gray codes corresponding to the decimal number 98 and 99 are 01010011 and 01010010, respectively. It has been used for motion estimation in video processing [22] and image stabilization [23].

However, these two traditional methods decompose an image into only a certain number of bit planes, and the content of each bit plane is predicable. Moreover, all the bit planes are binary. Those limitations significantly affect their applications (such as security applications) and motivate us to develop a new approach for the image bit-plane decomposition. In this section, we introduce a new image bit-plane decomposition method using the (n,k,p)-Gray code presented in Section II.

Definition 3.1 (The (n,k,p)-Gray-Code Bit-Plane Decomposition): From Definition 2.1, a grayscale image can be decomposed into k (n,k,p)-Gray-code bit planes, where the pixel values in the ith bit plane are the ith bit g_i of those pixels that have the same locations in the grayscale image. This is called the (n,k,p)-Gray-code bit-plane decomposition.

As a parameter-dependent bit-plane decomposition method, the new (n,k,p)-Gray-code bit-plane decomposition includes the traditional binary bit-plane decomposition and Gray-code bit-plane decomposition as special instances and also extends the concept of the bit-plane decomposition from the binary base (base 2) to the arbitrary base.

When the base n is greater than two, the values in the (n,k,p)-Gray-code bit planes are no longer binary. For example, the (3, 6, 1)-Gray code for a pixel with decimal value 10 is 000102. The least significant bit plane for a pixel of this value will have a "2" stored in the respective bit plane.

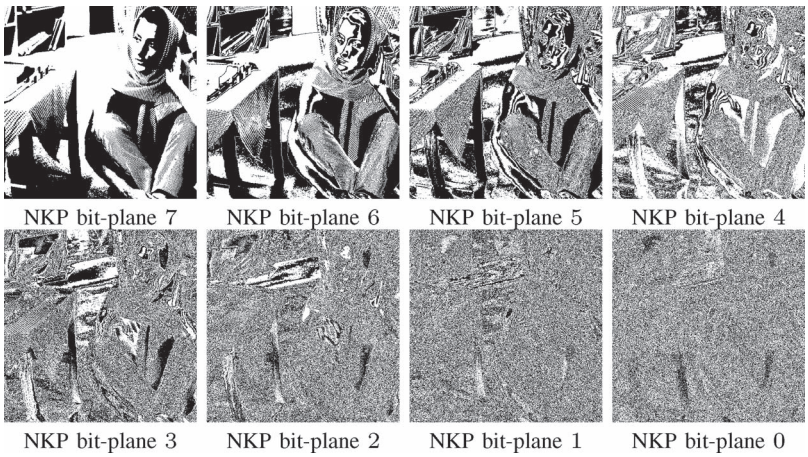

Fig. 1. (n, k, p)-Gray-code bit-plane decomposition of a grayscale image; $n = 2$ and $p = 2$.

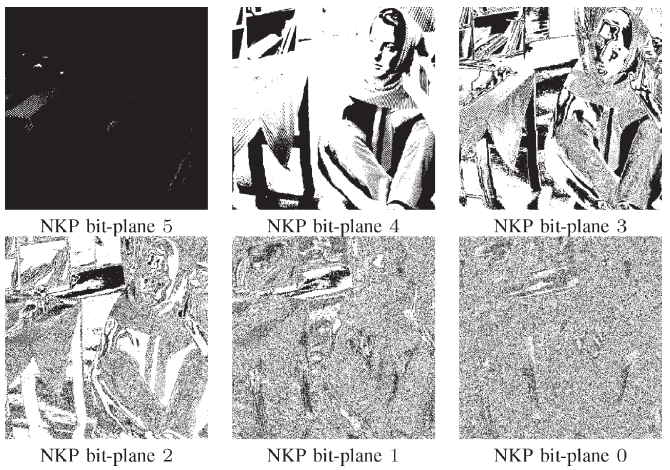

Fig. 2. (n, k, p)-Gray-code bit-plane decomposition of a grayscale image; $n = 3$ and $p = 0$. This is also an example of the (n, k)-Gray code.

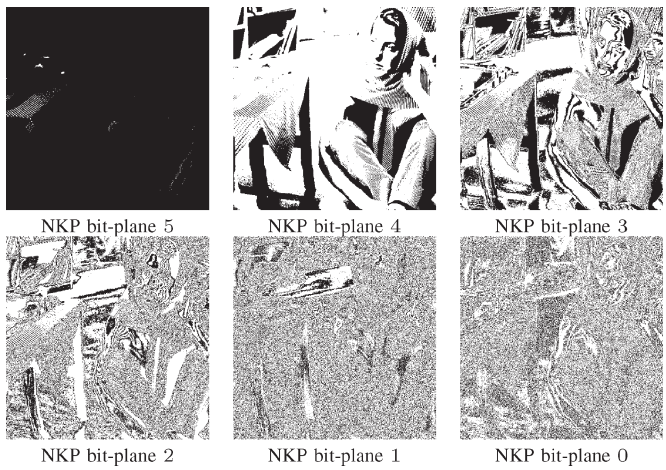

Fig. 3. (n, k, p)-Gray-code bit-plane decomposition of a grayscale image; $n = 3$ and $p = 4$.

The novelty of this decomposition method is that it can decompose an image not only into binary bit planes (for base $n = 2$) but also into nonbinary bit planes (for base $n > 2$). The (n, k, p)-Gray-code bit planes will change as the values of base n and distance parameter p change. For a specific image, the number of bit planes k is determined by the base-n value. For example, a grayscale image with gray levels between 0 and 255 can be decomposed into 8 $(k = \lceil \log_2 255 \rceil = 8)$ binary bit planes for $n = 2$. Fig. 1 shows an example of this.

When the base n is greater than two, the decomposed bit planes of a grayscale image that has gray levels between 0 and 255 will no longer be binary, and the number of decomposed bit planes will be less than eight. As a special case of the (n, k, p)-Gray code, the (n, k)-Gray code can also achieve this. Fig. 2 shows an example of this, where $n = 3$ and $p = 0$. However, for a given base n, the content of the decomposition results of an image that uses the (n, k)-Gray code will always be the same, whereas the content of the (n, k, p)-Gray code bit planes will change with the value of p. This is one of the advantages of the new decomposition method that uses the (n, k, p)-Gray code. Fig. 3 shows another example with $n = 3$ and $p = 4$.

Analysis of Figs. 2 and 3 demonstrates that the most significant bit plane does not change while the content of several of the least significant bit planes differs as the p values change. This is because, according to Definition 2.1, the (n, k, p)-Gray code keeps the most significant bit unchangeable.

In summary, the (n, k, p)-Gray-code bit-plane decomposition can decompose an image into binary and nonbinary bit planes. Both the decomposed results and the number of the (n, k, p)-Gray-code bit planes are parameter dependent. Those allow it to be used for many applications in image systems. The basic idea is shown in Fig. 4. The image is decomposed into the (n, k, p)-Gray-code bit planes. They are then manipulated by different image processing technologies. Based on this concept, we provide two illustrative examples in image denoising and encryption in the following two sections.

IV. (n, k, p)-Gray Code for Image Denoising

Image denoising is a fundamental process in image systems with a goal of removing different types of noises presented in images. Many effects have discussed noise reduction or removal in recent years [37]–[41].

To investigate the application of the (n, k, p)-Gray code for image denoising, this section introduces a new image denoising algorithm based on the (n, k, p)-Gray-code bit-plane decomposition.

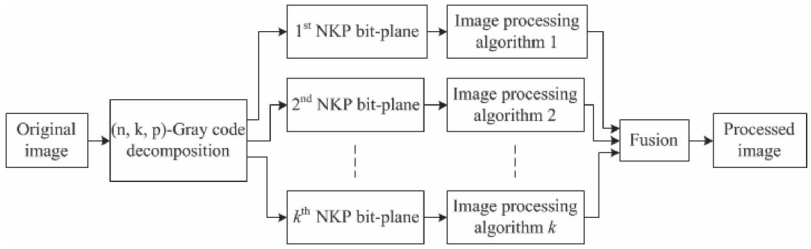

Fig. 4. (n, k, p)-Gray code in image systems.

Fig. 5. Image denoising using the α-trimmed mean filter and the (n, k, p)-Gray-code bit-plane decomposition. (a) Noise image with Gaussian noise and salt-and-pepper noise. (b) Denoised image by the (n, k, p)-Gray code with $n = 2$ and $p = 3$. (c) Denoised image by the (n, k)-Gray code with $n = 3$. (d) Denoised image by the AMF with windows size varying from 3 to 11.

A. New Image Denoising Algorithm

The proposed image denoising algorithm combines the filtering technique with the (n, k, p)-Gray-code bit-plane decomposition. Its block diagram can be obtained by replacing the image processing algorithm in Fig. 4 with filtering. The algorithm first decomposes the input image into several (n, k, p)-Gray-code bit planes and then performs a filtering operation for each bit plane. The resulting image with noise reduction is obtained by combining these filtered bit planes.

Users have the flexibility to utilize any new or existing filter in this denoising algorithm for removing different types of noises in images. They can filter only several selective (n, k, p)-Gray-code bit planes and keep others unchangeable to achieve better denoising performance.

B. Alpha-Trimmed Mean Filter

Because the alpha-trimmed mean filter shows a good compromise between median and moving average filters [42], we select it as an example of existing filters for simulation in this paper.

Definition 4.1 (The α-Trimmed Mean Filter): Let $x(m, n)$ be the gray level of the pixel with location (m, n) in an original image I, a small $N \times N$ $(N = 2k + 1)$ observation window centered by $x(m, n)$, and $\{x_1(m, n), x_2(m, n), \ldots, x_{N^2}(m, n)\}$ be a set of sorted gray levels of all pixels within the observation window, where $x_1(m, n) \leqslant x_2(m, n) \leqslant \cdots \leqslant x_{N^2}(m, n)$ and the $x_1(m, n)$ and $x_{N^2}(m, n)$ are the minimum and maximum values, respectively. The output of the α-trimmed mean filter is

$$y(m, n) = \frac{1}{N^2 - 2M} \sum_{i=M+1}^{N^2-M} x_i(m, n) \tag{4}$$

where $M = \lceil \alpha N^2 \rceil$ is the nearest integer greater than or equal to αN^2 and $0 \leqslant \alpha \leqslant 0.5$.

C. Simulation Results

Here, we not only compare the performance of the (n, k, p)-Gray code and traditional (n, k)-Gray code for image denoising but also compare the presented denoising algorithm with other denoising methods such as the adaptive median filter (AMF) [24], [43], [44]. Fig. 5 shows the experimental results of image denoising. The noise image in Fig. 5(a) is obtained by adding 0.3% Gaussian noise and 15% salt-and-pepper noise into the original clean logo image of the University of Macau. The proposed algorithm first decomposes the noise image [Fig. 5(a)] into several (n, k, p)-Gray-code bit planes using selected base-n and parameter p values. Each bit plane is then filtered by the α-trimmed mean filter with a specific α value. The denoised image is obtained by combining all (n, k, p)-Gray-code bit planes.

Fig. 5(b) and (c) shows denoised images using the (n, k, p)-Gray code and traditional (n, k)-Gray code, respectively. Fig. 5(d) shows the denoised result using the AMF. The denoising result using the (n, k, p)-Gray code shows the best visual quality.

To quantitatively evaluate their denoising performance, we measure the results shown in Fig. 5 using two different measures: the structural similarity (SSIM) index [45] and the root-mean-square error (RMSE). The measure results are shown in Table II. The SSIM is used to measure the similarity between the noised/denoised image and the original clean image. The SSIM values approaching to one indicate that the measured images are more similar. The images are identical if their SSIM values are equal to one. On the other hand, the RMSE is to quantitatively identify the difference between the noised/denoised image and the original clean image. The lower RMSE values mean the less difference between the measured images. From the measure results in Table II, the image obtained by the (n, k, p)-Gray code has the highest SSIM and least RMSE values. This means that the denoised result by the (n, k, p)-Gray code is the closest one to the original clean image.

TABLE II
MEASURE RESULTS OF IMAGE DENOISING

	SSIM	RMSE
Noise image	0.18697	8.1718
(n, k, p)-Gray code	0.90356	3.1882
(n, k)-Gray code	0.49541	4.8255
Adaptive median filter	0.3827	7.7722

Fig. 5 and Table II demonstrate that the presented denoising algorithm shows better performance than the AMF for reducing both Gaussian noise and salt-and-pepper noise. The (n, k, p)-Gray code outperforms the traditional (n, k)-Gray code in image denoising.

V. (n, k, p)-GRAY CODE FOR IMAGE ENCRYPTION

There is a need for multimedia applications including content distribution, archiving, search, and retrieval. These services bring new challenges for ensuring multimedia content confidentiality [46]. Image encryption is an effective tool to provide the security of images or videos by transforming them into a completely different format. Many effects have addressed this issue [46]–[48].

There are several image encryption algorithms based on binary bit-plane decomposition, such as the bit-plane encryption algorithm using exclusive-OR operations (BPE-XOR) [29], the selective bit-plane encryption algorithm using the Advanced Encryption Standard algorithm (SBE-AES) [31], and the selective bit-plane encryption algorithm using the least significant bit plane of images (SBE-LBP) [30]. These encryption methods have particular contributions in their specific applications. However, all of them are subject to security limitations due to the following facts: 1) Their decomposition results are sometimes predictable, and 2) the XOR operation and selective bit-plane encryption schemes have been shown to be vulnerable to a low-computational-cost attack [49].

In this section, we investigate the applications of the (n, k, p)-Gray code in image encryption. To enhance the security of the bit-plane-decomposition-based encryption methods, we introduce a new image encryption algorithm using the proposed (n, k, p)-Gray-code transforms and the parameter-dependent (n, k, p)-Gray-code bit-plane decomposition.

A. (n, k, p)-Gray-Code Transforms

This section introduces several (n, k, p)-Gray-code transforms which are going to be used for shuffling the order of bit planes and changing the pixel locations in applications of image encryption in Section V-B.

Definition 5.1 (The (n, k, p)-Gray-Code Transform): If two nonnegative integer sequences $\mathbb{A} = (A_1, A_2, \ldots, A_m)$ and $\mathbb{G} = (G_1, G_2, \ldots, G_m)$ are represented in the k-digital base-n matrices, namely,

$$\mathbb{A} = (A_1, A_2, \ldots, A_m) = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1k} & a_{2k} & \cdots & a_{mk} \end{pmatrix} \quad (5)$$

$$\mathbb{G} = (G_1, G_2, \ldots, G_m) = \begin{pmatrix} g_{11} & g_{21} & \cdots & g_{m1} \\ g_{12} & g_{22} & \cdots & g_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ g_{1k} & g_{2k} & \cdots & g_{mk} \end{pmatrix} \quad (6)$$

where $A_i = \sum_{j=1}^{k} a_{ij} n^{j-1}$ and $G_i = \sum_{j=1}^{k} g_{ij} n^{j-1}$, $0 < i \leqslant m$, the following transformation is called the (n, k, p)-Gray-code transform

$$\mathbb{G} = (C_p \mathbb{A}) \bmod n \quad (7)$$

where the coefficient matrix

$$C_p = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1k} \\ c_{21} & c_{22} & \cdots & c_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kk} \end{pmatrix} \quad (8)$$

where

$$c_{xy} = \begin{cases} 1, & \text{if } x = y \\ 1, & \text{if } y = x + p + 1 \leqslant k \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$m,\ k,\ i,\ j,\ p,\ x,$ and y are integers, $1 \leqslant x, y \leqslant k$, and $0 \leqslant p \leqslant k - 1$.

From Definition 5.1, the k-digital base-n representation matrices of the input sequence $\mathbb{A}$ and the output sequence $\mathbb{G}$ change with different base-n values. The $k \times k$ coefficient matrix C_p changes with different values of the base n and parameter p. For example, if $p = 0$, the (n, k, p)-Gray-code transform is

$$\mathbb{G} = \left(\begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \mathbb{A} \right) \bmod n. \quad (10)$$

If $p = 1$, the (n, k, p)-Gray-code transform will change to

$$\mathbb{G} = \left(\begin{pmatrix} 1 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \mathbb{A} \right) \bmod n. \quad (11)$$

The output integer sequence $\mathbb{G}$ in the (n, k, p)-Gray-code transform is the permutation of the input sequence $\mathbb{A}$. It changes with alterations of the base n and parameter p. For example, if the input sequence of the (n, k, p)-Gray-code transform is (1, 2, 3, 4, 5, 6, 7, 8), its output sequence will be (1, 2, 4, 5, 3, 8, 6, 7) for $n = 3$ and $p = 0$. The output sequence will be (1, 2, 3, 5, 4, 7, 6, 8) when $n = 2$ and $p = 1$. In this way, we obtain different permutations of the input sequence. For a given permutated sequence, we define its inverse transform in Definition 5.2.

Definition 5.2 (The Inverse (n, k, p)-Gray-Code Transform): If an integer sequence $\mathbb{G} = (G_1, G_2, \ldots, G_m)$ is the (n, k, p)-Gray-code representation of a nonnegative integer sequence

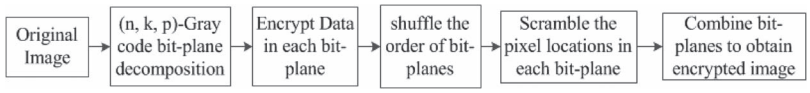

Fig. 6. Image encryption algorithm using the (n, k, p)-Gray-code bit-plane decomposition.

$\mathbb{A} = (A_1, A_2, \ldots, A_m)$, the following transformation is called the inverse (n, k, p)-Gray-code transform

$$\mathbb{A} = \left(C_p^{-1}\mathbb{G}\right) \bmod n \tag{12}$$

where all matrices and m, n, p, k are given in Definition 5.1 and C_p^{-1} is the inverse matrix of C_p.

The (n, k, p)-Gray-code transform can be used to scramble 1-D sequences or data streams such as string, password, and audio/speech signals. In this paper, we use this transform to shuffle the order of bit planes in the image encryption algorithm presented in Section V-B.

Although the (n, k, p)-Gray-code transform works optimally for 1-D data streams, the same transform can be used to scramble 2-D images line by line. However, the computational costs of this are extremely high. To overcome the high overhead of using the (n, k, p)-Gray-code transform for 2-D cases, we introduce a more efficient 2-D transform to create the permutations for the 2-D cases.

Definition 5.3 (The 2-D (n, k, p)-Gray-Code Transform): Letting D be an $M \times N$ image and T_r and T_c be the row and column coefficient matrices, respectively, the 2-D (n, k, p)-Gray-code transform is defined as

$$E = T_r D T_c \tag{13}$$

where E is the encrypted image and

$$T_r(x, y) = \begin{cases} 1, & \text{if } (x, G_x) \\ 0, & \text{otherwise} \end{cases} \qquad T_c(i, j) = \begin{cases} 1, & \text{if } (G_j, j) \\ 0, & \text{otherwise} \end{cases}$$

where $1 \leqslant x, y \leqslant M$, $1 \leqslant i, j \leqslant N$, G_x, and G_i can be generated from Definition 5.1 when the base n and parameter p are specified.

For example, for an 8×8 input image, if $n = 2$ and $p = 0$, the row and column coefficient matrices of the 2-D (n, k, p)-Gray-code transform can be represented as

$$T_r = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_c = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Definition 5.4 (The Inverse 2-D (n, k, p)-Gray-Code Transform): Letting E be the encrypted images, T_r^{-1} and T_c^{-1} be inverse matrices of the row and column coefficient matrices T_r and T_c defined in Definition 5.3, respectively, the following transformation is called the inverse 2-D (n, k, p)-Gray-code transform:

$$R = T_r^{-1} E T_c^{-1} \tag{14}$$

where R is the reconstructed image.

The 2-D (n, k, p)-Gray-code transform is a more efficient process to encrypt 2-D images than the (n, k, p)-Gray-code transform in Definition 5.1. It can encrypt 2-D images by applying the 2-D (n, k, p)-Gray-code transform only one time. Furthermore, the users simply apply the inverse 2-D (n, k, p)-Gray-code transform one time to reconstruct the original 2-D image. In this paper, we use the 2-D (n, k, p)-Gray-code transform to encrypt bit planes of images.

B. New Image Encryption Algorithm

The basic idea of the new image encryption algorithm is to decompose the image into the (n, k, p)-Gray-code bit planes, change the image pixel values of each bit plane using the mod operation, shuffle the order of all (n, k, p)-Gray-code bit planes and pixel locations within each bit plane, and combine all (n, k, p)-Gray-code bit planes to obtain the encrypted image. Fig. 6 shows the new image encryption algorithm.

The new encryption algorithm contains four processes: image decomposition, data encryption, bit-plane shuffling, and pixel scrambling. First, it decomposes the original image with a size of $M \times N$ into several (n, k, p)-Gray-code bit planes using parameters n_D and p_D. Second, it encrypts pixel data in each (n, k, p)-Gray-code bit plane using a mod operation defined by

$$E(i, j) = (I(i, j) + Y(i, j)) \bmod n_D \tag{15}$$

where $I(i, j)$ and $E(i, j)$ are the pixel intensity values with location (i, j) in the original and encrypted (n, k, p)-Gray-code bit plane, respectively. n_D is the base of the (n, k, p)-Gray code in the image decomposition process. $Y(i, j)$ is the security key plane generated from the chaotic logistic map defined by

$$\begin{cases} Y(i, j) = x\left[N(i-1) + j\right] \\ x(m) = rx(m-1)\left[1 - x(m-1)\right] \end{cases} \tag{16}$$

where $1 \leqslant i \leqslant M$, $1 \leqslant j \leqslant N$, parameters in the chaotic logistic map $1 \leqslant m \leqslant MN$, $0 < x_0 < 1$, $3.5699456 \leqslant r \leqslant 4$.

Third, the order of all bit planes is shuffled by means of a shuffling method. Fourth, the locations of all pixels in each bit plane are scrambled using an image scrambling algorithm. The final encrypted image is obtained by combining all the scrambled (n, k, p)-Gray-code bit planes.

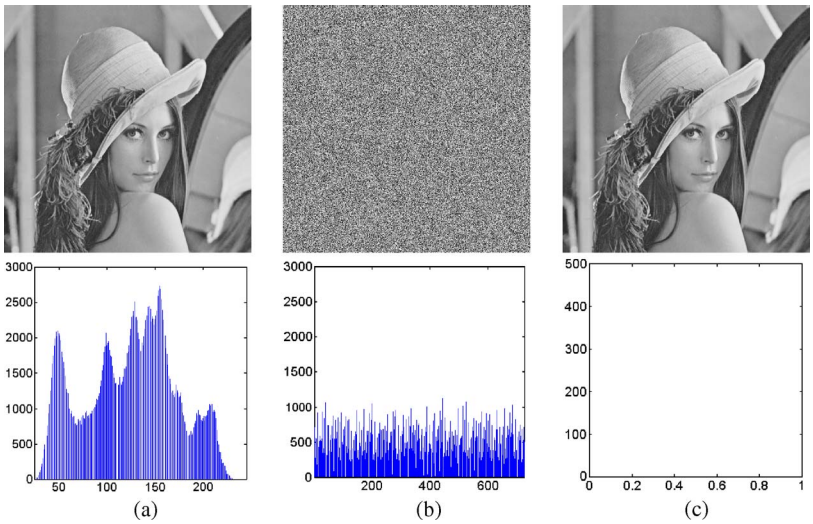

Fig. 7. Image encryption using Case #1. (a) Original image and its histogram. (b) Encrypted image and its histogram; $n = 3$, $k = 6$, and $p = 1$. (c) Reconstructed image and the histogram of the difference between the reconstructed image and the original image.

The security keys of the new image encryption algorithm consist of the parameters in its four processes: 1) n_D and p_D for the image decomposition process; 2) x_0 and r in the logistic map for the data encryption process; 3) parameters for the bit-plane shuffling process; and 4) parameters for the pixel scrambling process. Users have the flexibility to choose different security keys for each bit plane in the data encryption and pixel scrambling processes, achieving a higher level of security.

To reconstruct the original image, authorized users should be provided with the correct combination of the security keys. The decryption process will first decompose the encrypted image into the (n, k, p)-Gray-code bit planes using n_D and p_D, then unscramble pixels in each bit plane, then revert the order of the bit planes back into its original, then apply a mod operation to each bit plane using the security key plane obtained from the logistic map with parameters x_0 and r, and, finally, combine all the (n, k, p)-Gray-code bit planes to obtain the resulting reconstructed image.

In the decomposition process, the decomposition results and the number of the (n, k, p)-Gray-code bit planes are parameter dependent and will change based on different parameters n_D and p_D. The attackers have difficulty to predict the decomposed results. Furthermore, the decomposed results will directly affect the performance of other processes in the running of the new encryption algorithm. The original images will not be completely reconstructed if the user does not correctly decompose the encrypted image into their (n, k, p)-Gray-code bit planes, thereby achieving a higher level of security.

The data encryption process is designed to change image data using a mod operation similar to the XOR operation in the binary number system. The advantage of the mod operation is that it works on the arbitrary base and is able to change pixel values in each bit plane without changing the pixel data range. Furthermore, the security key plane is parameter dependent and changes as the parameters x_0 and r change in the chaotic logistic map.

By rearranging the order of the (n, k, p)-Gray-code bit planes, the bit-plane shuffling process is set to change the image pixel values. The pixel scrambling process is used to change the pixel locations in each (n, k, p)-Gray-code bit plane. Users have the flexibility to select any method in order to perform the following: 1) shuffle the order of the bit planes and 2) change the pixel locations. Moreover, when each bit plane is applied with different scrambling algorithms or different security keys, the scrambling process simultaneously changes image pixel locations and values.

C. Simulation Results and Analysis

Because the new image encryption algorithm offers users the flexibility to choose any existing or new method for shuffling the order of all the (n, k, p)-Gray-code bit planes and for scrambling pixel locations in each bit plane, we study two different cases in this section. *Case* #1 will demonstrate the use of the new (n, k, p)-Gray-code transforms for the bit-plane shuffling process and for performing pixel scrambling. *Case* #2 will demonstrate an existing approach to the shuffling and pixel scrambling processes. This should reveal how the new encryption algorithm is adaptable to a variety of approaches. Note that we use $x_0 = 0.32$ and $r = 3.65$ for the chaotic logistic map in all simulations in this section.

***Case* #1:**

1) For the bit-plane shuffling process, we first reverse the order of all the (n, k, p)-Gray-code bit planes and then shuffle their order using the (n, k, p)-Gray-code transform provided in Definition 5.1. The parameters of the (n, k, p)-Gray-code transform are called n_S and p_S.
2) The 2-D (n, k, p)-Gray-code transform from Definition 5.3 is used to scramble pixels in each bit plane. Its parameters are called n_E and p_E.

For simplicity, we choose the same base-n and parameter p values for image decomposition and for the bit-plane shuffling processes as well as for each bit plane in the pixel scrambling process in our simulations, i.e., $n_D = n_S = n_E$ and $p_D = p_S = p_E$.

Fig. 7 shows an example of image encryption based on Case #1 with security keys: $n_D = n_S = n_E = 3$ and $p_D = p_S = p_E = 6$ for image decomposition, for the bit-plane

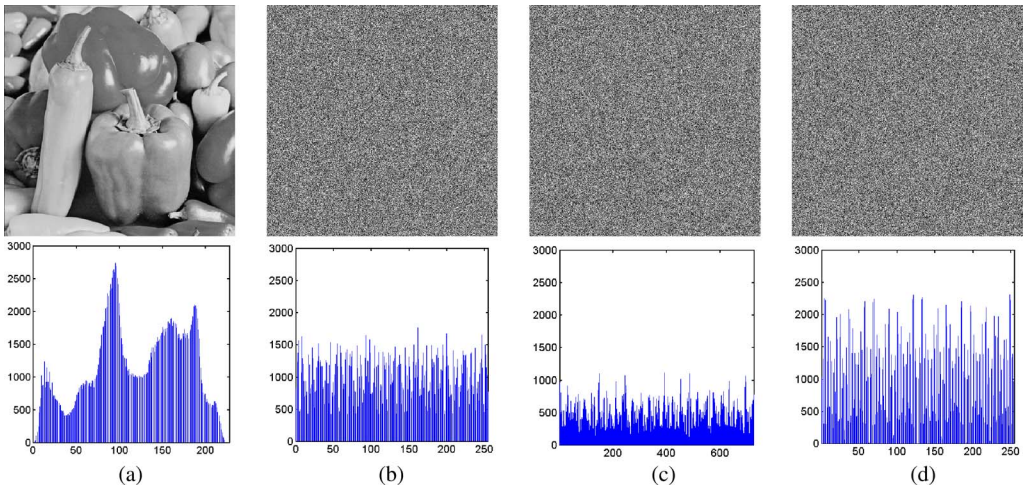

Fig. 8. Images encryption using Case #1. (a) is the original image and its histogram. (b)–(f) are the encrypted images and their corresponding histograms. (b) BRGC; $n = 2$ and $k = 8$. (c) Ternary Gray code; $n = 3$ and $k = 6$. (d) Presented (n, k, p)-Gray code; $n = 2$, $k = 8$, and $p = 7$.

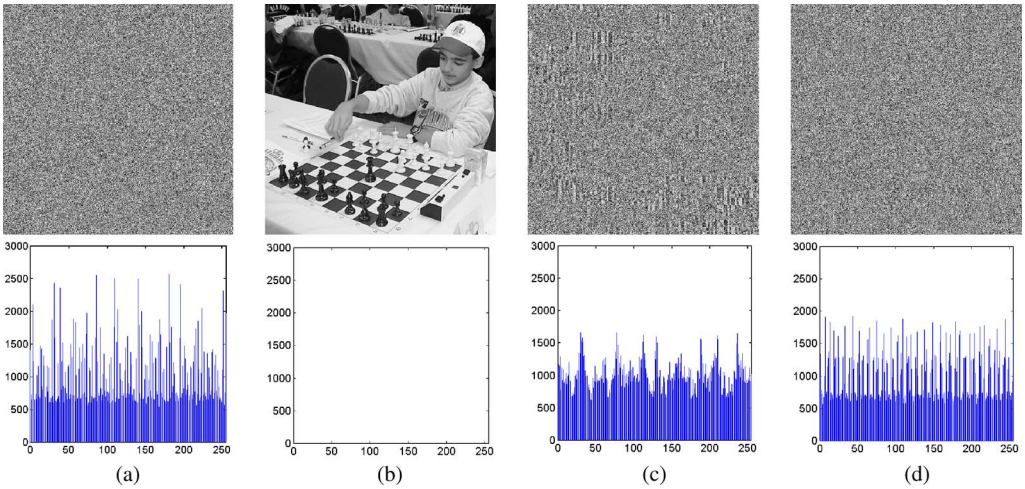

Fig. 9. Case #1 image reconstruction using different parameter p values. (a) Encrypted image and its histogram; $n = 2$, $k = 8$, and $p = 2$. (b) Reconstructed image and the histogram of the difference between the reconstructed image and the original image; $n = 2$, $k = 8$, and $p = 2$. (c) Reconstructed image and its histogram; $n = 2$, $k = 8$, and $p = 1$. (d) Reconstructed image and its histogram; $n = 2$, $k = 8$, and $p = 4$.

shuffling process, and for all the (n, k, p)-Gray-code bit planes in the pixel scrambling process. The original image is fully encrypted [as shown in Fig. 7(b)] and completely reconstructed [as shown in Fig. 7(c)]. The recovered image is visually the same as the original.

To quantitatively and graphically show the difference between the reconstructed image and its original, we subtract the reconstructed image from the original image pixel by pixel to generate a difference image. The histogram of this difference image, as shown in Fig. 7(c), demonstrates that all the pixels in the image are zeros, which proves that the reconstructed image and the original are identical.

Fig. 8 shows several encrypted results using Case #1 with different base-n and parameter p values. The original grayscale image [Fig. 8(a)] is encrypted by the BRGC [Fig. 8(b)] and the traditional ternary Gray code [Fig. 8(c)], both of which are examples of the traditional (n, k)-Gray code. Fig. 8(d) shows the encrypted result obtained by the new (n, k, p)-Gray code. As can be seen, the encrypted images are completely unrecognizable when compared to the original image. Visually, they look like noise images, and their histograms are close to a uniform distribution. This ensures that unauthorized users have difficulty to decode the encrypted images. The encrypted images change with different base-n and parameter p values, a fact that can be verified by their corresponding histograms. These encryption results and their histograms demonstrate that good encryption results can be obtained when the base-n and p values change. The new (n, k, p)-Gray code demonstrates a superior performance when it comes to image encryption compared to other traditional Gray codes.

In the new encryption algorithm for Case #1, the base n and parameter p of the (n, k, p)-Gray code act as security keys for image decomposition, bit-plane shuffling, and pixel scrambling processes. The combinations of these security keys are extremely important for authorized users who wish to recover the original images. An example of image reconstruction is shown in Fig. 9. The original image can be completely reconstructed, as shown in Fig. 9(b), only when the correct

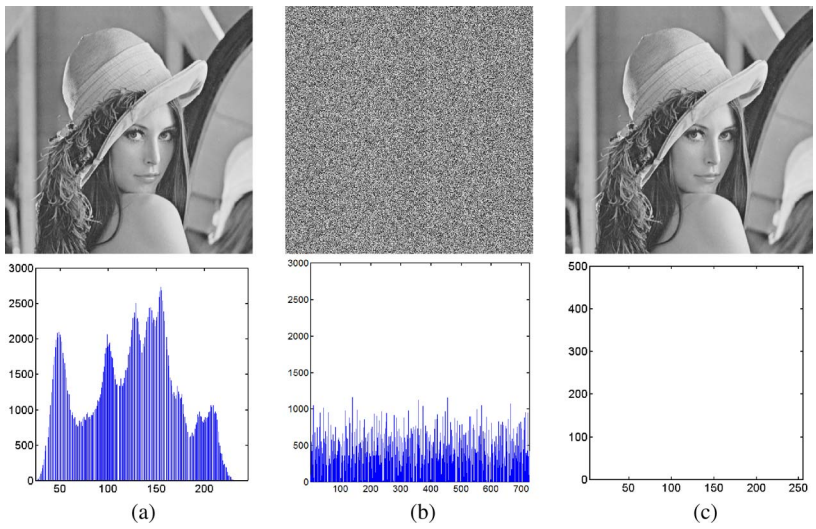

Fig. 10. Image encryption utilizing Case #2. (a) Original image and its histogram. (b) Encrypted image and its histogram; $n = 3$, $k = 6$, and $p = 1$. (c) Reconstructed image and the histogram of the difference between the image in (c) and the image in (a).

combination of the security keys is utilized. This prefect reconstruction can be verified by the histogram of the difference between the original image and the reconstructed image shown in Fig. 9(b). If the incorrect security keys are used, the reconstructed images are unrecognizable, as can be seen in Fig. 9(c) and (d).

The reconstructed results in Fig. 9(c) and (d) may also lead themselves to an alternative direction for image encryption, namely, using one set of security keys to encrypt the original image and a different set of security keys to reconstruct the image so that the final encrypted image can be obtained. In this manner, the histogram of the encrypted image will be much closer to a uniform distribution. However, this method may incur unwanted computational costs since it will undoubtedly require more processes for image encryption and decryption than the algorithm being presented here.

***Case* #2**:

1) For bit-plane shuffling process, we simply reverse the order of all the (n, k, p)-Gray-code bit planes.
2) The 2-D cat map [50] is used to scramble the pixel locations in each (n, k, p)-Gray-code bit plane in the pixel scrambling process.

Fig. 10 shows an encryption example of Case #2. The original image and the parameters for the image decomposition are the same as the example in Fig. 7. The original image is fully encrypted and completely reconstructed. The encrypted image is visually similar to the noise image, while its histogram distribution is close to uniform.

Fig. 11 shows several images encrypted by Case #1 and Case #2 and their corresponding histograms. All the encrypted images are visually similar to noise images. Their corresponding histograms have an almost uniform distribution. There is no significant difference between Case #1 and Case #2 when it comes to image encryption. This comparison demonstrates that the new encryption algorithm performs excellently for image encryption and that the new (n, k, p)-Gray code outperforms other traditional Gray codes, as can be seen in the histogram distribution.

We compare the presented new encryption algorithm with several existing bit-plane-decomposition-based encryption methods, i.e., the BPE-XOR, SBE-AES, and SBE-LBP algorithms. The encryption results are shown in Fig. 12. As can be seen, the presented new algorithm shows the best encryption performance. Its encryption image Fig. 12(c)] visually looks like a noise image which is completely different with the original image [Fig. 12(a)]. However, there are some information leakages in the encrypted images by other methods, as shown in Fig. 12(d)–(f).

D. Security Analysis and Comparison

Image encryption algorithms have been developed to ensure the security of images and videos. However, protected images are easily broken by unauthorized users if the security of an encryption algorithm is not carefully taken into account. Therefore, security is important for both the protected objects and for the encryption algorithm itself. In this section, we discuss several security issues associated with the new encryption algorithm.

The new encryption algorithm uses four techniques to improve the security level of the bit-plane-decomposition-based image encryption algorithms.

1) It introduces the (n, k, p)-Gray-code bit-plane decomposition in place of traditional binary bit-plane decomposition. Its decomposition results and the number of decomposed bit planes change with the values of the base n and parameter p. The attacker will thus have difficulty predicting the decomposed results. Furthermore, the correctly decomposed results are extremely important for authorized users to be able to reconstruct images because they directly affect the success of data

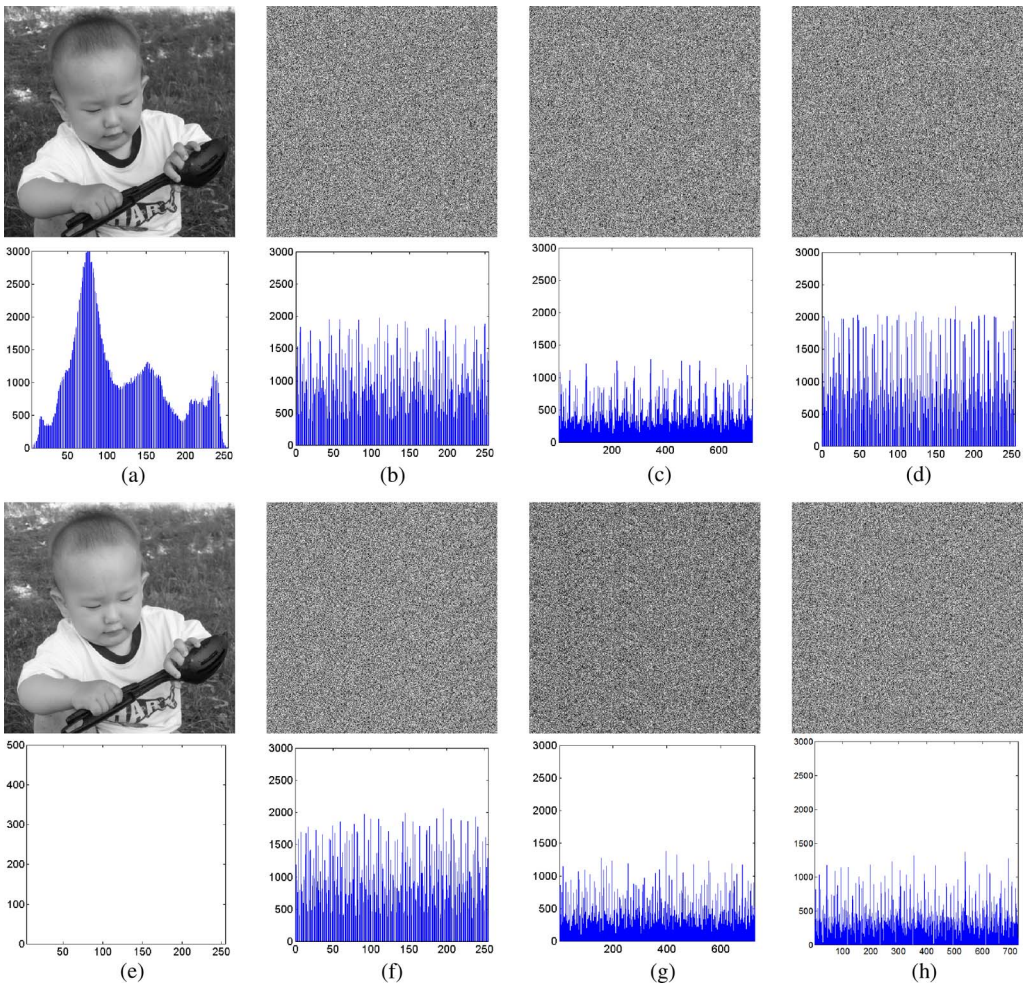

Fig. 11. Comparison of image encryption using Case #1 and Case #2. (a) is the original image and its histogram. (e) is the reconstructed image and the histogram of the difference between the images in (a) and (e). (b)–(d) are the encrypted images using Case #1 and their corresponding histograms. (f)–(h) are the encrypted images using Case #2 and their corresponding histograms. (b) Binary Gray code; $n = 2$. (c) Ternary Gray code; $n = 3$. (d) (n, k, p)-Gray code; $n = 2$, $k = 8$, and $p = 1$. (f) Binary Gray code; $n = 2$. (f) Ternary Gray code; $n = 3$. (h) (n, k, p)-Gray code; $n = 3$, $k = 6$, and $p = 2$.

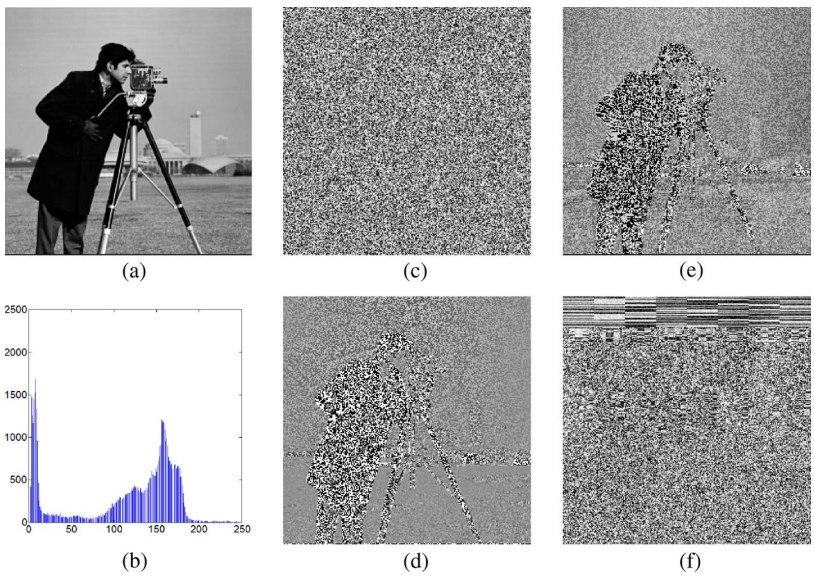

Fig. 12. Performance comparison of image encryption algorithms. (a) and (b) are the original image and its histogram, respectively. (c)–(f) are the images encrypted by different algorithms. (c) Our new algorithm. (d) BPE-XOR. (e) SBE-LBP. (f) SBE-AES.

TABLE III
COMPARISON OF BIT-PLANE-DECOMPOSITION-BASED IMAGE ENCRYPTION ALGORITHMS

	BPE-XOR	SBE-AES	SBE-LBP	Our Algorithm
Decomposition results	fixed	fixed	fixed	Parameter-dependent
Data encryption	XOR operation	AES	XOR operation	MOD operation
Shuffling process	NO	NO	NO	Parameter-dependent
Pixel scrambling	NO	NO	NO	Parameter-dependent
Change image data values	YES	YES	YES	YES
Change image pixel locations	NO	NO	NO	YES

decryption, bit-plane unshuffling, and pixel unscrambling processes.

2) In a similar manner to the binary XOR operation, the mod operation in the data encryption process works on the arbitrary base. It can keep the data range while changing pixel values.
3) The goal of the bit-plane shuffling is to change image pixel values by changing the order of the (n, k, p)-Gray-code bit planes. Bit-plane shuffling is a parameter-dependent process conducive to image encryption. It further increases the attackers' difficulty of decoding the images encrypted by the new algorithm.
4) The pixel scrambling process is designed to scramble the pixel locations in each bit plane. This process changes both the image pixel values and the image pixel locations. It enhances the new algorithms' immunity for plaintext attacks.

Table III compares the new encryption algorithm to the BPE-XOR, SBE-AES, and SBE-LBP algorithms. In terms of security and from the cryptographic point of view, the new algorithm possesses more advantages than existing methods. As a result, the new algorithm presents greater opportunities for improving the level of security protection compared to existing bit-plane-decomposition-based image encryption methods.

E. Security Key Space

For an encryption algorithm, the larger the key space is, the more possible combinations the security keys have. As a result, unauthorized users have more difficulty for obtaining the correct combination of security keys by means of an exhaustive search of all possible cases in the security key space and, thus, for decoding the encrypted images.

The new algorithm consists of four processes: image decomposition, data encryption, bit-plane shuffling, and pixel scrambling. To demonstrate how the key space of the new encryption algorithm is calculated, we use an $M \times N$ grayscale image with gray levels between 0 and 255 as an example.

1) The input image is decomposed into $B(B = \lceil \log_{n_D} 255 \rceil)$ bit planes in the image decomposition process. The possible choices of the security keys n_D and p_D in this process are $K_1 = K_{n_D} K_{p_D} = 254B$.
2) In the data encryption process, the security key plane is generated from the logistic map specified by two parameters: initial value x_0 and weight coefficient r. Those two parameters act as the security keys for the data encryption process. Theoretically, the number of their possible choices is unlimited because x_0 and r can be any real number within their limitation ranges: $0 < x_0 < 1$, and $3.5699456 \leqslant r \leqslant 4$. On the other hand, they may have a limited number of combinations since the output of the logistic map may have the same or similar results as some combinations of x_0 and r. We assume that their possible choices are K_x and K_r, respectively. Thus, the possible choices of the security keys in the data encryption process are $K_2 = K_x K_r$.
3) Any existing or new data shuffling algorithm can be used for the bit-plane shuffling process. Thus, the possible changes of the order of the bit planes are $K_3 = B!$.
4) Any existing or new image scrambling algorithm can be used for scrambling pixel positions in each bit plane in the pixel scrambling process. Therefore, the possible changes in this process are $K_4 = (M!N!)^B$.

Thus, if all the (n, k, p)-Gray-code bit planes are encrypted individually, the key space of the new encryption algorithm is sufficiently large and defined by

$$S = K_1 K_2 K_3 K_4 = 254 B K_x K_r B! (M!N!)^B. \tag{17}$$

F. Adjacent Pixel Correlation Analysis

Adjacent pixel correlation analysis is to show an algorithm's capability for withstanding statistic attacks [50]–[52]. Here, we analyze the intensity distribution of two horizontally, vertically, and diagonally adjacent pixels in the original and its corresponding encrypted images by the presented new encryption algorithm.

Some 2048 pixels are randomly selected from the original image [Fig. 10(a)] and the encrypted image [Fig. 10(b)], respectively. Fig. 13 plots the distribution of these 2048 sample pixels and their adjacent pixels at the horizontal, vertical, and diagonal directions. The top row shows the distribution of adjacent pixels in the original image. As can be seen, pixels are located in or around the diagonal line. This means the adjacent pixels in the original images are equal or close to each other. They have high correlations. On the other hand, the bottom row in Fig. 13 shows

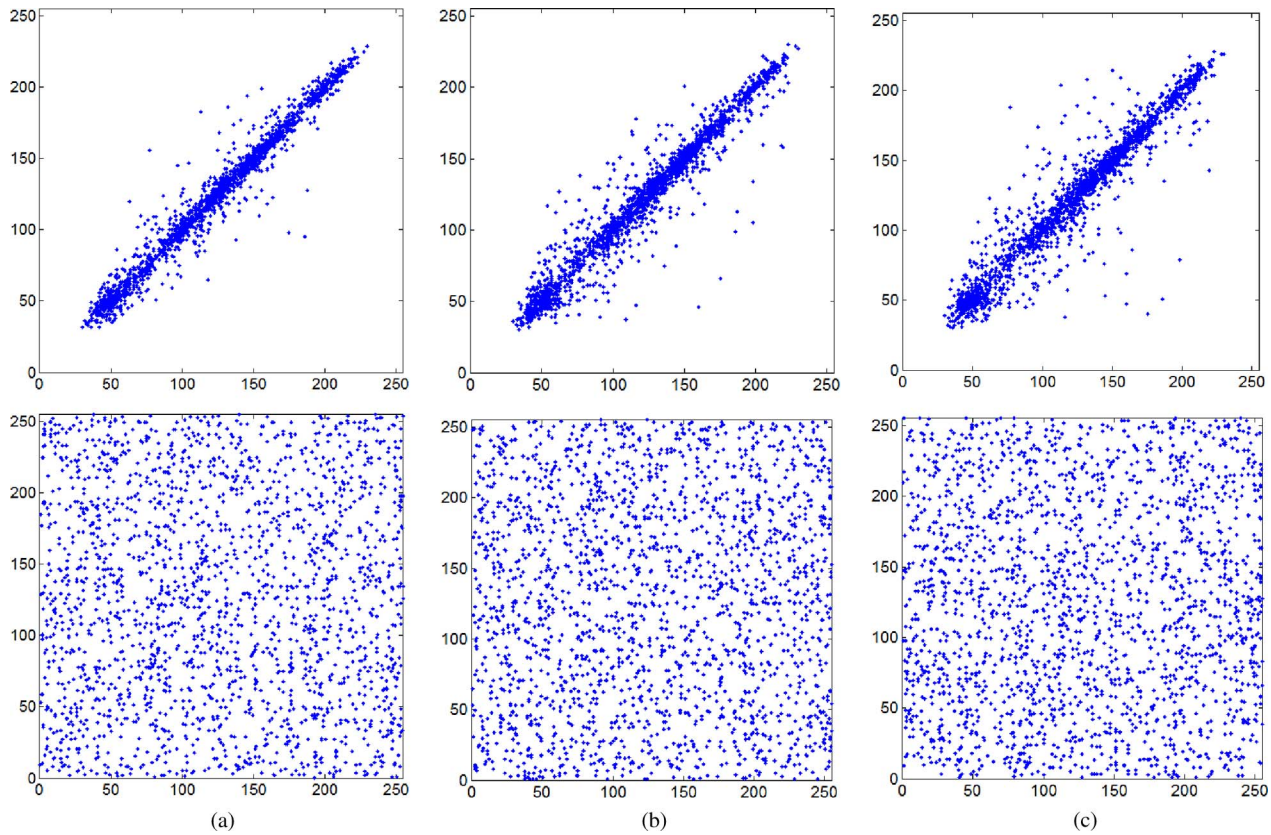

Fig. 13. Correlation of adjacent pixels at different directions before and after image encryption. The top row shows adjacent pixel correlation in the original Lena image [Fig. 10(a)]; the bottom row shows adjacent pixel correlation in the encrypted Lena image [Fig. 10(b)]. (a) Horizontal direction. (b) Vertical direction. (c) Diagonal direction.

the distributions of adjacent pixels in the encrypted image. These adjacent pixels show less correlation. Their pixel values significantly vary and spread out in the entire range of the image pixel values. This demonstrates that the presented encryption algorithm is able to withstand the statistic attack.

G. Plaintext Attack

The plaintext is the original information to be encrypted. The ciphertext is the plaintext encrypted by an encryption algorithm [53], [54]. There are two types of plaintext attacks: the known-plaintext attack and the chosen-plaintext attack.

In the known-plaintext attack, the attacker tries to obtain the security keys of the encryption algorithm by studying a number of plaintexts and their corresponding ciphertexts. In the chosen-plaintext attack, on the other hand, the attacker can choose a number of plaintexts and obtain their corresponding ciphertexts. According to cryptanalysis, the chosen-plaintext attack is a more advanced form of attacks because the attacker can select any useful information as plaintext to deduce the encryption algorithm security keys; either that or the attacker can reconstruct the original plaintexts from the unknown ciphertexts. Generally speaking, if an encryption algorithm can overcome the chosen-plaintext attack, it can also withstand other types of attacks such as ciphertext-only attacks and known-plaintext attacks.

For the new encryption algorithm, data encryption, bit-plane shuffling, and pixel scrambling are all important processes. Using mod operation, the data encryption directly changes pixel values individually within each (n, k, p)-Gray-code bit plane. The bit-plane shuffling changes the bit positions of image pixels in the vertical direction. The pixel scrambling changes pixel positions in the horizontal direction. These processes are parameter dependent and change the image pixel values based on the different security keys. As a result, unauthorized users have difficulty breaking the encrypted images via plaintext attacks.

To test the performance of the new algorithm for the plaintext attacks, an $M \times N$ matrix defined in (18) is designed as a plaintext

$$T(i, j) = j + (i - 1)M \tag{18}$$

where i and j are integers, $1 \leqslant i \leqslant M$, and $1 \leqslant j \leqslant N$.

The characteristic of this plaintext is that each element has a value different from each other. This plaintext can be used to break the permutation-only-based image encryption algorithms. These algorithms change only pixel locations of the plaintext. Therefore, searching each pixel value is able to locate all position changes within the ciphertext.

Fig. 14 shows a visual example using this chosen-plaintext attack defined in (18). The plaintext is shown in Fig. 14(d), and its corresponding ciphertext in Fig. 14(e) is obtained by the presented new encryption algorithm. The reconstructed result using the chosen-plaintext attack is shown in Fig. 14(c), which is completely different from the original image [Fig. 14(a)]. The histogram [Fig. 14(f)] of the difference between the original and the reconstructed images also verifies this. This is because the

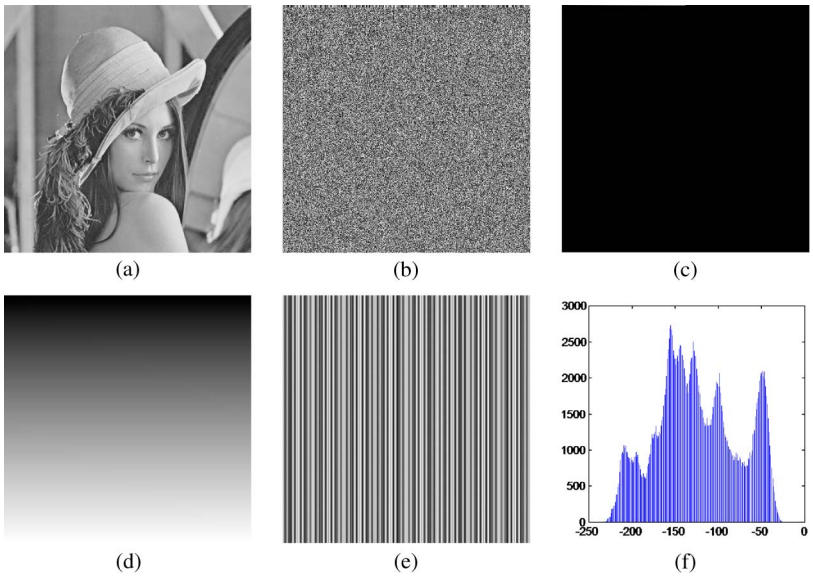

Fig. 14. Chosen-plaintext attack for the new encryption algorithm. (a) Original image. (b) Encrypted image with $n_D = n_S = n_E = 2$ and $p_D = p_S = p_E = 1$. (c) Reconstructed image using the chosen-plaintext attack. (d) Plaintext. (e) Ciphertext. (f) Histogram of the difference between (c) and (a).

presented algorithm changes the image pixel values and locations using three processes: data encryption, bit-plane shuffling, and pixel scrambling. Therefore, searching pixel values is no longer a feasible method to locate the original image pixels in the ciphertext. This chosen-plaintext attack fails to break the images encrypted by the proposed new algorithm.

VI. Conclusion

We have introduced the new (n, k, p)-Gray code in this paper which yields a new type of non-Boolean Gray code when its base is greater than two. We have investigated its applications in image systems and provided illustrative examples in bit-plane decomposition, image denoising, and encryption.

To show its application in image decomposition, we have presented a new image decomposition method using the (n, k, p)-Gray code, which can decompose images into base-2 (binary) or arbitrary base (nonbinary) Gray-code bit planes according to different n values. Its decomposition results and the number of decomposed bit planes differ with different base-n and parameter p values. Unlike the traditional (n, k)-Gray-code methods, the content of each decomposed bit plane will be different based on the value of parameter p.

We have shown the application of the (n, k, p)-Gray code in image denoising. A new image denoising algorithm has been introduced. The simulation results and comparison demonstrated that the (n, k, p)-Gray code shows better performance in image denoising than several types of traditional Gray codes.

To demonstrate the applicability of the (n, k, p)-Gray code in image encryption, we have introduced a new image encryption algorithm to improve the security level of existing bit-plane-decomposition-based encryption methods. The new algorithm offers the users flexibility to select any method for bit-plane shuffling and pixel scrambling. Several new (n, k, p)-Gray-code transforms have been presented for applications of bit-plane shuffling and pixel scrambling in image encryption.

The experimental results and comparison have shown that the presented new encryption algorithm shows excellent performance in image encryption. It could be used for protecting privacy in biometrics, medical imaging systems, and video surveillance systems. Our future research will further improve and analyze the performance of the (n, k, p)-Gray code in image denoising and encryption.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valued comments which helped to improve the manuscript.

References

[1] *Gray Code*, Wikipedia, San Francisco, CA.
[2] E. Gilbert, "Gray codes and paths on the n-cube," *Bell Syst. Tech. J.*, vol. 37, no. 1, pp. 815–826, 1958.
[3] M. Schwartz and T. Etzion, "The structure of single-track Gray codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2383–2396, Nov. 1999.
[4] G. Bhat and C. Savage, "Balanced Gray codes," *Electron. J. Combinatorics*, vol. 3, no. 1, pp. 1–11, 1996.
[5] M. C. Er, "On generating the n-ary reflected Gray codes," *IEEE Trans. Comput.*, vol. C-33, no. 8, pp. 739–741, Aug. 1984.
[6] B. D. Sharma and R. K. Khann, "On m-ary Gray codes," *Inf. Sci.*, vol. 15, no. 1, pp. 31–43, Jul. 1978.
[7] D.-J. Guan, "Generalized Gray code with applications," *Proc. Nat. Sci. Counc. ROC(A)*, vol. 22, no. 6, pp. 841–848, 1998.
[8] K. J. Sankar, V. M. Pandharipande, and P. S. Moharir, "Generalized Gray codes," in *Proc. Int. ISPACS*, 2004, pp. 654–659.
[9] C. Savage, "A survey of combinatorial Gray codes," *SIAM Rev.*, vol. 39, no. 4, pp. 605–629, Dec. 1997.
[10] D. Raymond and T. Garrett, "In-circuit digital tester," U.S. Patent 4 216 539, Aug. 5, 1980.
[11] C. C. Chang, H. Y. Chen, and C. Y. Chen, "Symbolic Gray code as a data allocation scheme for two-disc systems," *Comput. J.*, vol. 35, no. 3, pp. 299–305, Jun. 1992.
[12] D. Richards, "Data compression and Gray-code sorting," *Inf. Process. Lett.*, vol. 22, no. 4, pp. 201–205, Apr. 1986.
[13] C. Faloutsos, "Gray codes for partial match and range queries," *IEEE Trans. Softw. Eng.*, vol. 14, no. 10, pp. 1381–1393, Oct. 1988.

[14] I. N. Suparta and A. J. van Zanten, "Balanced maximum counting sequences," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3827–3830, Aug. 2006.
[15] J. Ludman, "Gray code generation for MPSK signals," *IEEE Trans. Commun.*, vol. COM-29, no. 10, pp. 1519–1522, Oct. 1981.
[16] G. Ben-Artzi, H. Hel-Or, and Y. Hel-Or, "The Gray-code filter kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 382–393, Mar. 2007.
[17] H.-W. Tseng and C.-C. Chang, "Anti-pseudo-gray coding for VQ encoded images over noisy channels," *IEEE Commun. Lett.*, vol. 11, no. 5, pp. 443–445, May 2007.
[18] W.-S. Chen, K.-H. Chih, S.-W. Shih, and C.-M. Hsieh, "Personal identification technique based on human IRIS recognition with wavelet transform," in *Proc. IEEE ICASSP*, 2005, vol. 2, pp. 949–952.
[19] W. Ding, W. Yan, and D. Qi, "Digital image scrambling," *Progr. Nat. Sci.*, vol. 11, no. 6, pp. 454–460, 2001.
[20] J. Zou and R. K. Ward, "Introducing two new image scrambling methods," in *Proc. IEEE PACRIM*, 2003, vol. 2, pp. 708–711.
[21] I. Nasir, W. Ying, and J. Jianmin, "A new robust watermarking scheme for color image in spatial domain," in *Proc. 3rd Int. IEEE Conf. SITIS*, 2007, pp. 942–947.
[22] S. Erturk, "Locally refined Gray-coded bit-plane matching for block motion estimation," in *Proc. 3rd ISPA*, 2003, vol. 1, pp. 128–133.
[23] S.-J. Ko, S.-H. Lee, S.-W. Jeon, and E.-S. Kang, "Fast digital image stabilizer based on Gray-coded bit-plane matching," *IEEE Trans. Consum. Electron.*, vol. 45, no. 3, pp. 598–603, Aug. 1999.
[24] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 2008.
[25] B. Govindarajani, K. Panett, and S. Agaian, "Progressive edge detection on multi-bit images using polynomial-based binarization," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2008, vol. 7, pp. 3714–3719.
[26] S. Kamata, R. O. Eason, and E. Kawaguchi, "Depth-first coding for multivalued pictures using bit-plane decomposition," *IEEE Trans. Commun.*, vol. 43, no. 5, pp. 1961–1969, May 1995.
[27] T. Loncar-Turukalo, V. Crnojevic, and Z. Trpovski, "Image compression by decomposition into bit planes," in *Proc. 5th Int. Conf. TELSIKS*, 2001, vol. 2, pp. 419–422.
[28] S. S. Yu and N. P. Galatsanos, "Binary decompositions for high-order entropy coding of grayscale images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 1, pp. 21–31, Feb. 1996.
[29] J.-W. Han, C.-S. Park, D.-H. Ryu, and E.-S. Kim, "Optical image encryption based on XOR operations," *Opt. Eng.*, vol. 38, no. 1, pp. 47–54, Jan. 1999.
[30] D. Moon, Y. Chung, S. B. Pan, K. Moon, and K. Chung, "An efficient selective encryption of fingerprint images for embedded processors," *ETRI J.*, vol. 28, no. 4, pp. 444–452, Aug. 2006.
[31] M. Podesser, H. Schmidt, and A. Uhl, "Selective bitplane encryption for secure transmission of image data in mobile environments," in *Proc. 5th NORSIG*, 2002, p. 1037.
[32] W. Zheng, Z.-G. Cheng, and Y.-l Cui, "Image data encryption and hiding based on wavelet packet transform and bit planes decomposition," in *Proc. 4th Int. Conf. WiCOM*, 2008, pp. 1–4.
[33] S. Dey, A. Abraham, and S. Sanyal, "An LSB data hiding technique using prime numbers," in *Proc. Int. Symp. 3rd IAS*, 2007, pp. 101–108.
[34] Z. Sun and H. Jiang, "An adaptive video watermarking based on decomposing of bit planes," in *Proc. 3rd Int. Conf. WGEC*, 2009, pp. 324–326.
[35] H. Noda, J. Spaulding, M. N. Shirazi, and E. Kawaguchi, "Application of bit-plane decomposition steganography to JPEG2000 encoded images," *IEEE Signal Process. Lett.*, vol. 9, no. 12, pp. 410–413, Dec. 2002.
[36] H. Noda, J. Spaulding, M. N. Shirazi, M. Niimi, and E. Kawaguchi, "Application of bit-plane decomposition steganography to wavelet encoded images," in *Proc. Int. Conf. Image Process.*, 2002, vol. 2, pp. II-909–II-912.
[37] J.-H. Wang, W.-J. Liu, and L.-D. Lin, "Histogram-based fuzzy filter for image restoration," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 2, pp. 230–238, Apr. 2002.
[38] O. Renaud, J. L. Starck, and F. Murtagh, "Wavelet-based combined signal filtering and prediction," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1241–1251, Dec. 2005.
[39] C.-S. Lee, S.-M. Guo, and C.-Y. Hsu, "Genetic-based fuzzy image filter and its application to image processing," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 4, pp. 694–711, Aug. 2005.
[40] L. Chen and K.-H. Yap, "An effective technique for subpixel image registration under noisy conditions," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 4, pp. 881–887, Jul. 2008.
[41] S. S. Agaian, E. E. Danahy, and K. A. Panetta, "Logical system representation of images and removal of impulse noise," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 6, pp. 1349–1362, Nov. 2008.
[42] R. Oten and R. J. P. de Figueiredo, "Adaptive alpha-trimmed mean filters under deviations from assumed noise model," *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 627–639, May 2004.
[43] H. Hwang and R. A. Haddad, "Adaptive median filters: New algorithms and results," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 499–502, Apr. 1995.
[44] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*, 2nd ed. Knoxville, TN: Gatesmark Publ., 2009.
[45] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[46] Y. Mao and M. Wu, "A joint signal processing and cryptographic approach to multimedia encryption," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 2061–2075, Jul. 2006.
[47] B. B. Zhu, Y. Chun, W. Yidong, and L. Shipeng, "Scalable protection for MPEG-4 fine granularity scalability," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 222–233, Apr. 2005.
[48] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Trans. Signal Process.*, vol. 48, no. 8, pp. 2439–2451, Aug. 2000.
[49] D. Engel, E. Pschernig, and A. Uhl, "An analysis of lightweight encryption schemes for fingerprint images," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 173–182, Jun. 2008.
[50] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos Solitons Fractals*, vol. 21, no. 3, pp. 749–761, Jul. 2004.
[51] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image Vis. Comput.*, vol. 24, no. 9, pp. 926–934, Sep. 2006.
[52] Y. Zhou, K. Panetta, S. Agaian, and C. L. P. Chen, "Image encryption using p-Fibonacci transform and decomposition," *Opt. Commun.*, vol. 285, no. 5, pp. 594–608, Mar. 2012.
[53] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1997.
[54] B. Schneier, *Applied Cryptography*, 2nd ed. Hoboken, NJ: Wiley, 1996.

Yicong Zhou (M'07) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA.

He is currently an Assistant Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His current research interests focus on multimedia security, image/signal processing, medical imaging, and object recognition.

Dr. Zhou is a member of the International Society for Photo-Optical Instrumentations Engineers.

Karen Panetta (S'84–M'85–SM'95–F'08) received the B.S. degree in computer engineering from Boston University, Boston, MA, and the M.S. and Ph.D. degrees in electrical engineering from Northeastern University, Boston.

She is currently a Professor with the Department of Electrical and Computer Engineering, Tufts University, Medford, MA, where she is the Director of the Simulation Research Laboratory. Her research focuses on developing efficient algorithms for simulation, modeling, signal, and image processing for security and biomedical applications.

Dr. Panetta is the Editor-in-Chief of the *IEEE Women in Engineering Magazine*. She served as the 2011 Chair of the IEEE Boston Section, which has over 8500 members. During 2007 to 2009, she served as the World Wide Director for IEEE Women in Engineering, overseeing the world's largest professional organization supporting women in engineering and science.

Sos Agaian (M'98–SM'00) received the M.S. degree (*summa cum laude*) in mathematics and mechanics from Yerevan State University, Yerevan, Armenia, the Ph.D. degree in mathematics and physics and the Doctor of Engineering Sciences (equivalent to the U.S. Doctor of Electrical and Computer Engineering) degree from the Academy of Sciences of the USSR, Moscow, Russia, and the Diploma in computer science (equivalent to the U.S. Ph.D. degree in computer science) from the Supreme Attestation Board of the USSR, Moscow.

He is currently the Peter T. Flawn Distinguished Professor with the College of Engineering, The University of Texas, San Antonio, and an Adjunct Professor with the Department of Electrical and Computer Engineering, Tufts University, Medford, MA. He has authored more than 450 scientific papers and 7 books and is the holder of 14 patents. He is an Associate Editor of the *Journal of Real-Time Imaging* and the *Journal of Electronic Imaging* and an Editorial Board Member of the *Journal of Pattern Recognition and Image Analysis*. His current research interests lie in the broad area of signal/image processing, bioinformatics, cancer imaging, information security, computer vision, and mobile imaging and secure communication.

He is a Fellow of the International Society of Photo-Optical Instrumentation Engineers and the American Association for the Advancement of Science.

C. L. Philip Chen (S'88–M'88–SM'94–F'07) received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN.

He was the Associate Dean for Research and Graduate Studies with the College of Engineering, The University of Texas, San Antonio, where he was a Professor and the Chair of the Department of Electrical and Computer Engineering, after he served as an Assistant Professor, an Associate Professor, and a Full Professor with the Department of Computer Science and Engineering, Wright State University, Dayton, OH. He is currently a Chair Professor and the Dean of the Faculty of Science and Technology, University of Macau, Macau, China. His research interests include computer networking, neural networks, fuzzy–neural systems, intelligent systems, robotics, and computer-aided design/computer-aided manufacturing.

Dr. Chen has served as a member of organizing committee for many IEEE conferences under different capacities. He is currently the President of the IEEE Systems, Man, and Cybernetics Society. He is an Accreditation Board of Engineering and Technology Education Program Evaluator for Computer Engineering, Electrical Engineering, and Software Engineering programs. He is a fellow of the American Association for the Advancement of Science.