Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 1014–1031
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Image encryption by random grids

Shyong Jian Shyu*

Department of Computer Science and Information Engineering, Ming Chuan University, 5 Der Ming Rd., Gawi Shan, Taoyuan 333, Taiwan, ROC

Received 23 August 2005; received in revised form 20 February 2006; accepted 28 February 2006

Abstract

A random grid in this paper is a transparency comprising a two-dimensional array of pixels that are either transparent or opaque determined in a totally random way. We design algorithms by using random grids to accomplish the encryption of the secret gray-level and color images in such a way that neither of the two encrypted shares alone leaks the information of the secret image, whereas the secret can be seen when these two shares are superimposed. The decryption process is done by our visual system and no computation is required. As compared to the approaches in visual cryptography, our algorithms do not need the basis matrices to encode the shares so that the problem of pixel expansion exists no more; that is, the sizes of the secret image and the encrypted shares are the same.

Keywords: Image encryption; Random grid; Visual cryptography; Halftone; Color mixing; Color decomposition

1. Introduction

With the rapid growth of Internet applications and network users, the access, process and distribution of the digital images via Internet become very easy, yet vulnerable. To protect the digital images from unauthorized acquisition, *image encryption* becomes a significant and emergent topic for researchers and practitioners. The skills developed in traditional cryptography provide us with numerous elegant approaches to protect the digital information. Techniques in areas of information hiding such as watermarking, anonymity, steganography, cover channel, and so on, are other solutions to digital image protection. However, the decryption process consumes an additional computing cost when we access the digital images encrypted by using these encryption techniques.

The encryption of pictures and shapes by *random grids* was first introduced by Kafri and Keren in 1987 [1]. Their method is based on a generalized moiré effect obtained with random grids. They encrypted a secret *binary* picture or sharp into two random grids in which the areas containing information in the two grids are intercorrelated. When the two grids are superimposed together, the correlated areas will be resolved from the random background due to the difference in *light transmission* so that the secret picture or sharp can be seen visually. The main advantage of the scheme is that the decoding process is done by human visual system in one step, simply by superimposing two grids, where no computation is needed. A simple application may be to authenticate the owner of a credit card where the owner's portrait is encrypted into two random grids and one grid is printed on the card while the other is kept by the authenticator. The authenticator superimposes the two grids to validate the identity of the owner by recognizing the reconstructed portrait. A forger cannot replace the encrypted grid on the card without the reference grid.

Later, with the same benefit that the decryption needs no digital computations, but only human visual system instead, Naor and Shamir in 1995 [2] introduced the terminology, namely *visual cryptography*, to describe the study of the visual version of the *secret sharing* problem. They gave a formal definition to the *visual secret sharing scheme* and then

* Tel.: +886 3 3507001x3402; fax: +886 3 3593874.
E-mail address: sjshyu@mcu.edu.tw.

0031-3203/$30.00
doi:10.1016/j.patcog.2006.02.025

proposed and analyzed their visual secret sharing schemes for the binary images [2]. Consider a secret binary image B that is shared by two participants. Their scheme encodes B into two transparencies (or *shares*) which are distributed to the two participants, one share for each participant, in such a way that only when the two shares are stacked together can the participants see B, while any single participant obtains no information about B. Their scheme first produces two 2×2 0/1 *basis matrices* S^0 and S^1 as follows:

$$S^0 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad S^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Then each pixel, say b, of B is encoded into two shares with two subpixels in each share according to M^0 or M^1 where M^0 (M^1) is some random column permutation result of S^0 (S^1), and $M^0[k, 1]$ and $M^0[k, 2]$ ($M^1[k, 1]$ and $M^1[k, 2]$) are the two encrypted codes of the kth share when b is transparent (opaque) for $k \in \{1, 2\}$. In practical implementation, each pixel in B would be expanded into 2 pixels, referred to as *pixel expansion* and denoted as m ($m = 2$ in this case), in each share. Since a larger pixel expansion results in a larger size for the encrypted shares, we expect the pixel expansion to be as small as possible.

With regard to sharing a secret *color* image between two participants, the most effective result is by Hou [8] who developed several approaches to encrypt color images with a pixel expansion of four ($m = 4$). Even though the optimal pixel expansion is 2 which was proved in [2], Hou's choice of $m = 4$ (due to the fact that both the width and the height of the encoded shares are doubled with respect to those of the secret image) is to retain the aspect ratio of the secret image. Such a consideration mostly causes the pixel expansion to be even larger.

For a comprehensive review on visual cryptography and some interesting extensions, please refer to Refs. [2–9]. Essentially, most of the succeeding researches on visual cryptography were based on the definition introduced by Naor and Shamir [2]. Thus, the issue of pixel expansion, or the necessity of the basis matrices, is unavoidable.

The author was attracted by the random grid-based approaches in [1] since they need neither basis matrices nor extra pixel expansion, i.e., $m = 1$. That is, the size of the encrypted shares is not enlarged as compared to that of the secret image. This is a significant advantage, especially for practical applications, over those schemes developed in visual cryptography [2–9] where the pixel expansion is indispensable. Kafri and Keren's ideas [1] can only be applied onto the binary images. In this paper, the author shall propose algorithms for the encryption of the gray-level and color images by using random grids. The most attractive advantage in our algorithms is that there is no need to expand the pixels at all; therefore, the sizes of the encrypted shares and the secret image are exactly the same.

The rest of this paper is organized as follows. The basic features of random grids and the encryption algorithms proposed in [1] are introduced in Section 2. The design and verification of the proposed algorithm for encrypting gray-level images are presented in Section 3. The features of colors and the proposed encryption algorithms for color images are discussed in Section 4, where the correctness and experimental results of our algorithms are also examined. Section 5 gives some concluding remarks.

2. Encryption of binary images by random grids

In Section 2.1, we introduce the fundamental characteristics of random grids and define some new notations to ease the following discussions. Then, we describe the random grid-based algorithms for encrypting binary images and give formal proofs to verify these approaches in Section 2.2. Section 2.3 exhibits the implementation results and highlights the ideas in these algorithms for further extensions.

2.1. Random grids

A *random grid* was defined by Kafri and Keren [1] as a transparency comprising a two-dimensional array of pixels. Each pixel is either fully *transparent* or simply *opaque* and the choice between the alternatives is made by a coin-flip procedure. Thus there is no correlation between the values of different pixels in the array. The transparent pixels let through the light while the opaque pixels stop it. Since the number of the transparent pixels is probabilistically equal to that of the opaque pixels in a random grid, the *average light transmission* of a random grid is $\frac{1}{2}$. Assume that R is a random grid. We denote the average light transmission of R as

$$\mathscr{T}(R) = \tfrac{1}{2}.$$

Regarding a certain pixel r in R, the probability for r to be transparent is equal to that for r to be opaque. We have

$$\mathscr{P}rob(r = 0) = \mathscr{P}rob(r = 1) = \tfrac{1}{2},$$

where 0 denotes transparent and 1 denotes opaque. We call r a *random pixel* in R and refer to its light transmission as

$$\ell(r) = \frac{1}{2}.$$

Let $\otimes$ denote the generalized "or" operation which describes the relation of the superimposition of two random grids pixel by pixel. It is obvious that $R \otimes R$ is entirely the same as R so that

$$\mathscr{T}(R \otimes R) = \tfrac{1}{2} \quad \text{or} \quad \ell(r \otimes r) = \tfrac{1}{2} \tag{1}$$

for each pixel r in R.

Table 1
Results of the superimposition of two random pixels

r_1	r_2	$r_1 \otimes r_2$
0	0	0
0	1	1
1	0	1
1	1	1

Let R_1 and R_2 be two independent random grids with the same size. When R_1 and R_2 are superimposed pixel by pixel, each pixel (either transparent or opaque) in R_1 has an equal possibility to be stacked by a transparent pixel or an opaque pixel in R_2. We call $r_1 = R_1[i, j]$ the *corresponding pixel* of $r_2 = R_2[i', j']$ if and only if $i = i'$ and $j = j'$ (the positions of r_1 at R_1 and r_2 at R_2 are the same). Table 1 shows the superimposed results of the corresponding random pixels r_1 and r_2. There is only one outcome among the four possible combinations of $r_1 \otimes r_2$ shows transparent. Since the four possible combinations occur with an equal probability, the probability for $r_1 \otimes r_2$ to be transparent is $\frac{1}{4}$. That is, the average light transmission of the superimposition of R_1 and R_2 (r_1 and r_2) is $\frac{1}{4}$:

$$\mathscr{T}(R_1 \otimes R_2) = \tfrac{1}{4} \quad \text{or} \quad \ell(r_1 \otimes r_2) = \tfrac{1}{4}. \tag{2}$$

We define $\bar{R}$ to be the *inverse random grid* of R if and only if

$$r' = \bar{r}$$

for each r' in $\bar{R}$ where r is the corresponding pixel of r' in R and $\bar{r}$ denotes the inverse of r. It is easy to see that $r \otimes \bar{r} = 1$ and $R \otimes \bar{R} = \mathbf{1}$ ($\mathbf{1}$ denotes a grid in which all pixels are opaque) so that $\mathscr{P}rob(r \otimes \bar{r} = 0) = 0$. That is,

$$\mathscr{T}(R \otimes \bar{R}) = 0 \quad \text{or} \quad \ell(r \otimes \bar{r}) = 0. \tag{3}$$

For each pixel r' in $\bar{R}$, since $\mathscr{P}rob(r' = 0) = \mathscr{P}rob(\bar{r} = 0) = \mathscr{P}rob(r = 1) = \frac{1}{2}$, we obtain

$$\mathscr{T}(\bar{R}) = \tfrac{1}{2}. \tag{4}$$

Consider two independent random grids X and Y. There is another important property called the *principle of combination* [1]: if we cut a section, say A, from X and replace it with section B (with the same size as A) from Y, the result, denoted as $Z = (X \backslash A) \cup B$, is another random grid, i.e.,

$$\mathscr{T}(Z) = \mathscr{T}(X) = \mathscr{T}(Y) = \frac{1}{2}.$$

2.2. Encryption algorithms for binary images by random grids

Kafri and Keren [1] proposed three different algorithms to accomplish the encryption for the binary images. Given a secret binary image B, these algorithms as follows produce two random grids R_1 and R_2 such that they leak no information of B individually, yet they reveal B in our visual system when superimposed.

Algorithm 1–3. Encryption of a binary image by random grids

Input: A $w \times h$ binary image B where $B[i, j] \in \{0, 1\}$ (white or black), $1 \leqslant i \leqslant w$ and $1 \leqslant j \leqslant h$
Output: Two shares of random grids R_1 and R_2 which reveal B when superimposed where $R_k[i, j] \in \{0, 1\}$ (transparent or opaque), $1 \leqslant i \leqslant w$, $1 \leqslant j \leqslant h$ and $k \in \{1, 2\}$
Encryption(B)

Algorithm 1.

1. Generate R_1 as a random grid, $\mathscr{T}(R_1) = 1/2$
   // for (each pixel $R_1[i, j]$, $1 \leqslant i \leqslant w$ and $1 \leqslant j \leqslant h$) do
   // $R_1[i, j] =$ random_pixel(0, 1)
2. for (each pixel $B[i, j]$, $1 \leqslant i \leqslant w$ and $1 \leqslant j \leqslant h$) do
2.1 { if ($B[i, j] = 0$) $R_2[i, j] = R_1[i, j]$
   else $R_2[i, j] = \overline{R_1[i, j]}$
   }
3. output (R_1, R_2)

Algorithm 2.

1. Generate R_1 as a random grid, $\mathscr{T}(R_1) = 1/2$
2. for (each pixel $B[i, j]$, $1 \leqslant i \leqslant w$ and $1 \leqslant j \leqslant h$) do
2.1 { if ($B[i, j] = 0$) $R_2[i, j] = R_1[i, j]$
   else $R_2[i, j] =$ random_pixel(0, 1)
   }
3. output (R_1, R_2)

Algorithm 3.

1. Generate R_1 as a random grid, $\mathscr{T}(R_1) = 1/2$
2. for (each pixel $B[i, j]$, $1 \leqslant i \leqslant w$ and $1 \leqslant j \leqslant h$) do
2.1 { if ($B[i, j] = 0$) $R_2[i, j] =$ random_pixel(0, 1)
   else $R_2[i, j] = \overline{R_1[i, j]}$
   }
3. output (R_1, R_2)

Note that random_pixel(0, 1) is a function that returns a binary value 0 or 1 to represent a transparent or opaque pixel, respectively, by a coin-flip procedure and $\overline{R_1[i, j]}$ denotes the inverse of $R_1[i, j]$. The three algorithms are capsulated into a generic procedure named *Encryption* so that when *Encryption* (B) is called, each of Algorithms 1, 2 or 3 can be applied onto B. Also note that in this paper, 0 (1) denotes a white (black) pixel in the secret binary image or a transparent (opaque) pixel in the encrypted share interchangeably.

Let $B(0)$ ($B(1)$) denote the area of all of the transparent (opaque) pixels in B, that is, pixel b is in $B(0)$ ($B(1)$) if and only if $b = 0 (b = 1)$ where $B = B(0) \cup B(1)$ and $B(0) \cap B(1) = \varnothing$. We denote the area of pixels in random

grid R corresponding to $B(0)$ ($B(1)$) by $R[B(0)]$ ($R[B(1)]$), that is, pixel r is in $R[B(0)]$ ($R[B(1)]$) if and only if r's corresponding pixel b is in $B(0)$ ($B(1)$). Surely, $R = R[B(0)] \cup R[B(1)]$ and $R[B(0)] \cap R[B(1)] = \varnothing$.

Based upon the above notations, we have the following theorem.

Theorem 1. *Given a secret binary image B, R_1 and R_2 generated by Algorithms* 1, 2 *and* 3, *respectively*, *satisfy*:

(1) $\mathscr{T}(R_1) = \mathscr{T}(R_2) = \frac{1}{2}$; *and*
(2) $\mathscr{T}(S[B(0)]) > \mathscr{T}(S[B(1)])$ *where* $S = R_1 \otimes R_2$;

so that no information of B can be obtained from R_1 or R_2 individually; *while S reveals B in our visual system.*

Proof. Consider Algorithm 1 first. Since $R_2[B(0)] = R_1[B(0)]$ and R_1 is a pure random grid, $\mathscr{T}(R_2[B(0)]) = \mathscr{T}(R_1[B(0)]) = \frac{1}{2}$ (formula (1)). Besides, due to $R_2[B(1)] = \overline{R_1[B(1)]}$ and $\overline{R_1}$ is a random grid too (formula (4)), $\mathscr{T}(R_2[B(1)]) = \mathscr{T}(\overline{R_1[B(1)]}) = \frac{1}{2}$. By the principle of combination, $R_2 = R_2[B(0)] \cup R_2[B(1)]$ is a random grid with $\mathscr{T}(R_2) = \frac{1}{2}$ $(= \mathscr{T}(R_1))$.

When R_1 and R_2 are superimposed as S, $S[B(0)] = R_1[B(0)] \otimes R_2[B(0)] = R_1[B(0)]$ and $S[B(1)] = R_1[B(1)] \otimes R_2[B(1)] = R_1[B(1)] \otimes \overline{R_1[B(1)]} = \mathbf{1}$. From formulae (1) and (3),

$$(\mathscr{T}(S[B(0)]), \mathscr{T}(S[B(1)])) = (\tfrac{1}{2}, 0). \tag{5}$$

It is not hard to recognize that R_2 generated by Algorithms 2 or 3 is also a random grid. In Algorithm 2, $R_2[B(0)] = R_1[B(0)]$ and $R_2[B(1)] =$ random_pixel(0, 1). We have $\mathscr{T}(S[B(0)]) = \mathscr{T}(R_1[B(0)] \otimes R_2[B(0)]) = \mathscr{T}(R_1[B(0)]) = \frac{1}{2}$ and $\mathscr{T}(S[B(1)]) = \mathscr{T}(R_1[B(1)] \otimes R_2[B(1)]) = \frac{1}{4}$ (formula (2)). Thus,

$$(\mathscr{T}(S[B(0)]), \mathscr{T}(S[B(1)])) = (\tfrac{1}{2}, \tfrac{1}{4}). \tag{6}$$

In Algorithm 3, $R_2[B(0)] =$ random_pixel(0, 1) and $R_2[B(1)] = \overline{R_1[B(1)]}$. We obtain $\mathscr{T}(S[B(0)]) = \mathscr{T}(\overline{R_1[B(0)]} \otimes R_2[B(0)]) = \frac{1}{4}$ and $\mathscr{T}(S[B(1)]) = \mathscr{T}(R_1[B(1)] \otimes \overline{R_1[B(1)]}) = 0$. Thus,

$$(\mathscr{T}(S[B(0)]), \mathscr{T}(S[B(1)])) = (\tfrac{1}{4}, 0). \tag{7}$$

From formulae (5), (6) and (7), we know that $\mathscr{T}(S[B(0)]) > \mathscr{T}(S[B(1)])$ for all of the three algorithms. It means that the difference between the reconstructed pixels in $S[B(0)]$ and $S[B(0)]$ can be distinguished due to the difference between their light transmissions. We conclude that no information of B can be obtained from random grids R_1 or R_2 individually, while S reveals B in our visual system for all of the three algorithms. □

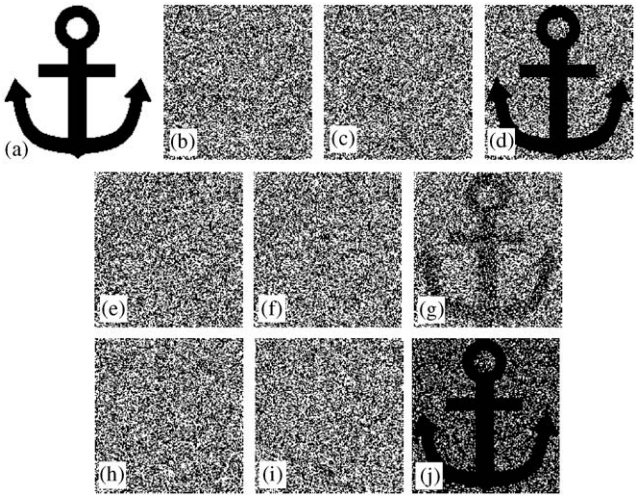

Fig. 1. Implementation results of Algorithms 1, 2 and 3 for encrypting binary image B: (a) B; (b), (c) and (d) two encrypted shares and reconstructed image by Algorithm 1; (e), (f) and (g) two encrypted shares and reconstructed image by Algorithm 2; (h), (i) and (j) two encrypted shares and reconstructed image by Algorithm 3.

2.3. Experiments and discussions for encrypting a binary image

Fig. 1 illustrates the results of the implementation of the above three algorithms. Fig. 1(a) is secret binary image B, Fig. 1(b) and (c) present the two random grids produced by Algorithm 1 and Fig. 1(d) is the superimposed result of these two shares ((b) and (c)). Fig. 1(e)–(g) illustrate the corresponding results by Algorithm 2, while Fig. 1(h)–(j) are the corresponding results by Algorithm 3. It can be easily seen from Fig. 1 that the encrypted shares (see Fig. 1(b), (c), (e), (f), (h), and (i)) are merely random pictures and no information about B can be obtained. Only when the two shares are superimposed (see Fig. (d), (g) and (j)), can we see B by our visual system. It is worthy of notifying that there is no extra pixel expansion in Fig. 1(b)–(j).

Fig. 2 shows the reconstructed results by using Naor and Shamir's approach for encrypting B in Fig. 1(a). The pixel expansion of Fig. 2(a) is 2, while that of Fig. 2(b) is 4 $\left(\text{by applying } S^0 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \text{ and } S^1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}\right)$. The former does not retain the aspect ratio with respect to B, while the latter does. Both sizes of the results in Fig. 2 are larger than that of B.

To evaluate the relative difference of the light transmissions between the transparent and opaque pixels in reconstructed image S by these random grid-based algorithms, we define the *contrast* of S with respect to B by algorithm A as

$$\frac{\mathscr{T}(S[B(0)]) - \mathscr{T}(S[B(1)])}{1 + \mathscr{T}(S[B(1)])},$$

where $S = R_1 \otimes R_2$ and R_1 and R_2 are the random grids of B encrypted by Algorithm A. Thus, the contrasts achieved

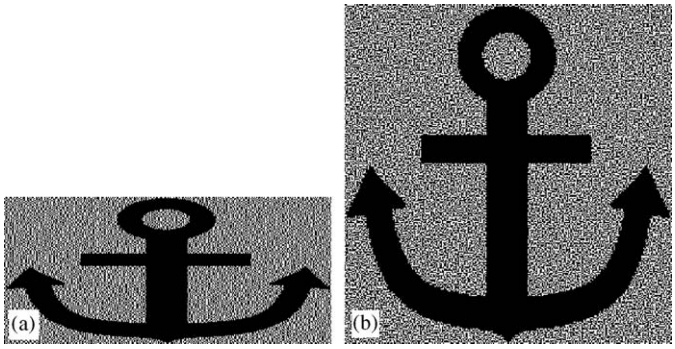

Fig. 2. Reconstructed results by Naor and Shamir's approach for binary image B in Fig. 1(a): (a) $m = 2$, (b) $m = 4$.

Table 2
Encoding b into r_1 and r_2 and results of $s = r_1 \otimes r_2$ by Algorithms 1, 2 and 3

	b	Probability	r_1	r_2	$s = r_1 \otimes r_2$	$\mathscr{P}rob(s{=}0)\ (\ell(s))$
Algorithm 1	□	$\frac{1}{2}$	□	□	□	$\frac{1}{2}$
		$\frac{1}{2}$	■	■	■	
	■	$\frac{1}{2}$	□	■	■	0
		$\frac{1}{2}$	■	□	■	
Algorithm 2	□	$\frac{1}{2}$	□	□	□	$\frac{1}{2}$
		$\frac{1}{2}$	■	■	■	
	■	$\frac{1}{4}$	□	□	□	$\frac{1}{4}$
		$\frac{1}{4}$	□	■	■	
		$\frac{1}{4}$	■	□	■	
		$\frac{1}{4}$	■	■	■	
Algorithm 3	□	$\frac{1}{4}$	□	□	□	$\frac{1}{4}$
		$\frac{1}{4}$	□	■	■	
		$\frac{1}{4}$	■	□	■	
		$\frac{1}{4}$	■	■	■	
	■	$\frac{1}{2}$	□	■	■	0
		$\frac{1}{2}$	■	□	■	

by Algorithms 1, 2 and 3 are $\frac{1}{2}$, $\frac{1}{5}$ and $\frac{1}{4}$, respectively. We may say that the reconstructed image obtained by Algorithm 1, which achieves the largest contrast among the three, can be recognized easier in our visual system than those by the other two algorithms (see Fig. 1(d), (g) and (j) accordingly).

To ease the extensions in the following, Table 2 summaries the encoding process of pixel b in secret image B into r_1 and r_2 by Algorithms 1, 2 and 3, respectively.

3. Encryption of gray-level images by random grids

The essential idea of our encryption algorithm for a gray-level image is to transform it into a halftone image and to exploit the encryption ability of random grids on the binary images. We thus briefly introduce the halftone technology in Section 3.1. The encryption algorithm for the gray-level images by random grids is presented in Section 3.2. Section 3.3 gives the results of our implementation.

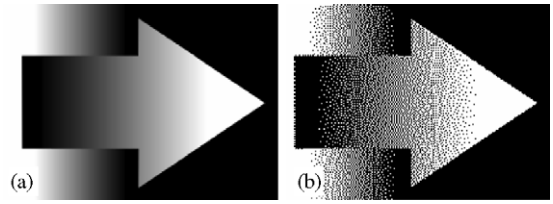

Fig. 3. Example of halftoning gray-level image G: (a) continuous tone, (b) halftone.

3.1. Halftone technology

The *halftone technology* [10–14] that utilizes the density of binary net dots to simulate the gray-level images is fairly mature and is widely applied in the printing or optics industrials. Basically, in a halftone image the binary net dots are sparse (dense) when the corresponding area of the original gray-level image is relatively brighter (darker). Even though the halftone image contains pixels with either white or black only, the human visual system still perceptually recognizes it as a gray-level image. Fig. 3 illustrates an example of halftoning a continuous tone gray-level image G by using the *error diffusion* algorithm [14,15]. As can be seen, both of Fig. 3(a) and (b) would be recognized as (almost) the same gray-level image by our visual system.

3.2. Encrypting gray-level images by random grids

Let $\mathbf{B}$ be the set of colors in a binary image, i.e., $\mathbf{B} = \{0, 1\}$. Let $\mathbf{G}$ denote the set of all shades of gray in a continuous tone gray-level image ranging from white to

black. Note that in most modern personal computer systems, $|\mathbf{G}| = 256$, that is, there are 256 shards of gray for a gray-level image. We represent the halftoning procedure that transforms a gray-level image G into its halftone version H by

$$H = \mathscr{H}(G),$$

where $g \in \mathbf{G}$ for each pixel g in G and $h \in \mathbf{B}$ for each pixel h in H. Since H is simply a binary image, we apply the encryption algorithms by using random grids mentioned in Section 2.2 onto H directly to accomplish the encryption of G.

Algorithm 4 presents such a straightforward idea.

Algorithm 4. Encryption of a gray-level image by random grids

Input: A $w \times h$ gray-level image G where $G[i, j] \in \mathbf{G}$, $1 \leqslant i \leqslant w$ and $1 \leqslant j \leqslant h$
Output: Two shares of random grids R_1 and R_2 which reveal G when superimposed where $R_k[i, j] \in \mathbf{B}$, $1 \leqslant i \leqslant w$, $1 \leqslant j \leqslant h$ and $k \in \{1, 2\}$

1. $H = \mathscr{H}(G)$
2. $(R_1, R_2) = Encryption(H)$
3. output (R_1, R_2)

Note that as mentioned in Section 2.2, procedure *Encryption* in Algorithm 4 can be implemented by any one of Algorithms 1, 2 or 3.

Theorem 2 is an immediate consequence from Theorem 1.

Theorem 2. *Given a secret gray-level image G, R_1 and R_2 generated by Algorithm 4 satisfy*:

(1) $\mathscr{T}(\boldsymbol{R}_1) = \mathscr{T}(\boldsymbol{R}_2) = \frac{1}{2}$; *and*
(2) $\mathscr{T}(S[H(0)]) > \mathscr{T}(S[H(1)])$ *where* $H = \mathscr{H}(G)$ *and* $S = R_1 \otimes R_2$;

so that no information of G can be obtained from R_1 or R_2 individually; *while S reveals G in our visual system.*

3.3. Implementation results for encrypting a gray-level image

We tested gray-level image G in Fig. 3(a) by Algorithm 4. Fig. 3(b) is its corresponding halftone result H and Fig. 4 illustrates the results of applying Algorithm 4 with different implementations of procedure *Encryption*. When *Encryption* is implemented by Algorithm 1, the two encrypted shares are shown in Fig. 4(a) and (b), Fig. 4(c) is the superimposed result of them. The corresponding results by Algorithm 2 are Fig. 4(d), (e) and (f), while those by Algorithm 3 are Fig. 4(g), (h) and (i). The encrypted shares (see Fig. 4(a), (b), (d), (e), (g) and (h)) are merely random grids and no information about G can be obtained. Only when the two corresponding shares are superimposed (see Fig. 4(c), (f) and (i)) can we see G by our visual system.

It is seen that Fig. 4(c) is more recognizable than Fig. 4(f) and (i). It means that when Algorithms 1, 2 and 3 are applied to encrypt a gray-level image, respectively, the superiority of Algorithm 1 among the three approaches in achieving the best contrast for the binary images holds still. Fig. 4(f) reveals that Algorithm 2 even makes the superimposed result hard to perceive. As we emphasized, the sizes of the secret image and the encrypted shares are exactly the same.

4. Encryption of color images by random grids

To encrypt the color images by using random grids, we integrated some features of colors with the encryption abilities of random grids on the binary/gray-level images in our algorithms. The color features including *color models*, *color mixture* and *color decomposition* [8,16–18] are briefly introduced in Section 4.1. We subsequently propose the random grid-based encryption algorithms for the color images and prove their correctness in Section 4.2. Section 4.3 gives the experimental results and discussions.

4.1. Color models

The constitutions of colors are commonly described by the *additive model* or the *subtractive model* [16–18]. The additive model mainly describes the mixture of colors when the colors are originated from *lights* (or simply the *mixture of colored lights*), while the subtractive model defines the mixture of colors when the colors are made of dyes, pigments, paints or other natural colorants as shown in Fig. 5. The discussions about color mixture in this paper are focused on the subtractive model for the reason that we use colored transparencies, on which the colorants are painted, as the encrypted shares in our algorithms.

4.1.1. Color mixture

In essence, the subtractive color system [16–18] involves colorants and their absorption (subtraction) of white light that is composed of red, green and blue (i.e. the *primary colors in the additive model*) lights. Subtractive color starts with a color object, which reflects light such as paper or canvas, or lets through light such as transparency or glass, and uses colorants (pigments or dyes) to subtract portions of white light illuminating the object to produce other colors.

The three *primary colors in the subtractive model* are cyan (**c**), magenta (**m**) and yellow (**y**). They cannot be created by mixing other colors and they are able to produce a wide range of colors by mixing various amounts of **c**, **m** and **y** colorants. Table 3 explains the relationship of color mixture according to the absorption of colors in the subtractive model where

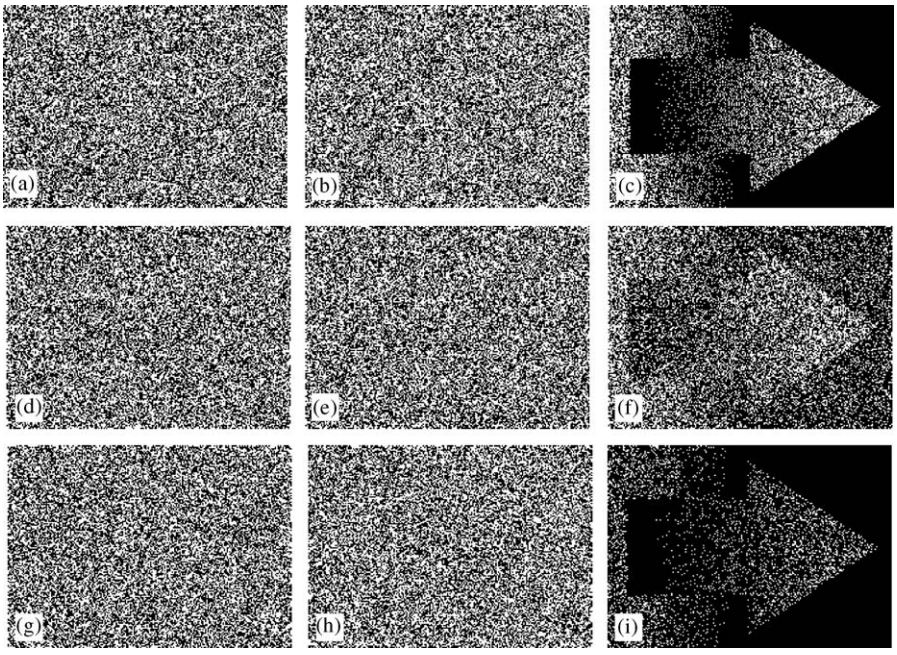

Fig. 4. Implementation results of Algorithm 4 for encrypting gray-level image G in Fig. 3: (a), (b) and (c) two encrypted shares and their superimposed result by using Algorithms 1 in *Encryption*; (d), (e) and (f) two encrypted shares and their superimposed result by using Algorithms 2 in *Encryption*; (g), (h) and (i) two encrypted shares and their superimposed result by using Algorithms 3 in *Encryption*.

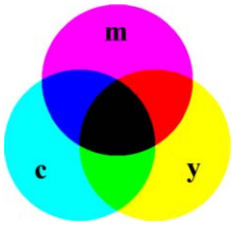

Fig. 5. Color mixture in the subtractive model.

Table 3
Colors and their absorption of white light in the subtractive model

Color	Absorbs	Reflects (or lets through)	Appears
$\varnothing$	None	Red, green and blue light	0
c	Red light	Green and blue light	**c**
m	Green light	Red and blue light	**m**
y	Blue light	Red and green light	**y**
c + **m**	Red and green light	Blue light	**b**
c + **y**	Red and blue light	Green light	**g**
m + **y**	Green and blue light	Red light	**r**
c + **m** + **y**	Red, green and blue light	None	1

"+" denotes the operation of color mixing.

b, **g** and **r** denote blue, green and red colors, respectively. Thus, when the colored transparency (or the superimposed transparencies) is held up to white light, our visual system detects the colors on the transparency according to the rules in Table 3.

In the domain of linear algebra, **c**, **m** and **y** constitute a *linearly independent color basis* that can generate all other colors in the color space by their linear combinations. To formalize the following discussions, we shall introduce some terminologies for colors. Let $\mathbf{E} = \{\mathbf{c}, \mathbf{m}, \mathbf{y}\}$ be the set of the primary colors in the subtractive model. Let $\mathbf{C} = \{0, \mathbf{c}, \mathbf{m}, \mathbf{y}, \mathbf{r}, \mathbf{g}, \mathbf{b}, 1\}$ denote the set of colors mixed by all subsets of $\mathbf{E}$ with an equal amount of colorant(s). Let $\breve{\mathbf{C}}$ denote the set of all colors mixed by all subsets of $\mathbf{E}$ with various amounts of colorant(s). Since $\breve{\mathbf{C}}$ contains all colors in the color space mixed by colors in $\mathbf{E}$, we call $\breve{\mathbf{C}}$ the *universal color set mixed out of* $\mathbf{E}$. It will be explained later that a secret color image may contain all colors in $\breve{\mathbf{C}}$, while an encrypted color share contains only the eight colors in $\mathbf{C}$. Furthermore, we define $\mathbf{E}^{\mathbf{c}} = \{0, \mathbf{c}\}$, $\mathbf{E}^{\mathbf{m}} = \{0, \mathbf{m}\}$ and $\mathbf{E}^{\mathbf{y}} = \{0, \mathbf{y}\}$ to be the *binary monochromatic color sets* in the subtractive model. Just as colors between 0 and 1 (in $\mathbf{B}$) can be divided into some amount of shades of gray in $\mathbf{G}$ ranging from 0 to 1, we can divide the colors between 0 and $\boldsymbol{x}$ (in $\mathbf{E}^{\boldsymbol{x}}$) into a certain amount of shades of color $\boldsymbol{x}$ for $\boldsymbol{x} \in \mathbf{E}$. Accordingly, we call $\breve{\mathbf{E}}^{\boldsymbol{x}}$ the *universal monochromatic color set* of $\boldsymbol{x}$ which contains all shades of color $\boldsymbol{x}$ for $\boldsymbol{x} \in \mathbf{E}$.

4.1.2. Color decomposition

Consider a secret color image P in which $p \in \breve{\mathbf{C}}$ for each *color pixel* p in P. We can decompose each color pixel p into three *monochromatic components/pixels*, namely $p^{\mathbf{c}}$,

$p^{\mathbf{m}}$ and $p^{\mathbf{y}}$, in terms of the three primary colors **c**, **m** and **y**, respectively, where $p^{\boldsymbol{x}} \in \breve{\mathbf{E}}^{\boldsymbol{x}}$ for $\boldsymbol{x} \in \mathbf{E}$. We call this process the *color decomposition* of p. When all pixels in P are decomposed in the same way, the **c**, **m** and **y**-colored monochromatic images composing of all $p^{\mathbf{c}}$'s, $p^{\mathbf{m}}$'s and $p^{\mathbf{y}}$'s are referred to as $P^{\mathbf{c}}$, $P^{\mathbf{m}}$ and $P^{\mathbf{y}}$, respectively. We denote the process of the color decomposition of color pixel $p \in \breve{\mathbf{C}}$ in P by function $\mathscr{d}$: $\breve{\mathbf{C}} \to \breve{\mathbf{E}}^{\mathbf{c}} \times \breve{\mathbf{E}}^{\mathbf{m}} \times \breve{\mathbf{E}}^{\mathbf{y}}$ as follows:

$$\mathscr{d}(p) = (p^{\mathbf{c}}, p^{\mathbf{m}}, p^{\mathbf{y}}),$$

where $p^{\boldsymbol{x}} \in \breve{\mathbf{E}}^{\boldsymbol{x}}$ in $P^{\boldsymbol{x}}$ is the corresponding monochromatic pixel of p for $\boldsymbol{x} \in \mathbf{E}$. We represent the procedure of the color decomposition of P into $P^{\mathbf{c}}$, $P^{\mathbf{m}}$, $P^{\mathbf{y}}$ as

$$\mathscr{D}(P) = (P^{\mathbf{c}}, P^{\mathbf{m}}, P^{\mathbf{y}}).$$

Note that in most modern personal computer systems, $P^{\boldsymbol{x}}$ is represented as a continuous tone ***x*-colored gray-level image** in which each pixel $p^{\boldsymbol{x}}$ is colored by one of the 256 shades of color $\boldsymbol{x}$ ranging from 0 (white) to 255 (color $\boldsymbol{x}$), i.e., $|\breve{\mathbf{E}}^{\boldsymbol{x}}| = 256$ for $\boldsymbol{x} \in \breve{\mathbf{E}}$. Besides, $|\breve{\mathbf{C}}| = 256 \times 256 \times 256$ and $\mathscr{d}$ is a one to one and onto function.

It is easy to transform $P^{\boldsymbol{x}}$ into the corresponding ***x*-colored halftone image** $\boldsymbol{P}^{\boldsymbol{x}}$ by applying the halftone technology such that each pixel $\boldsymbol{p}^{\boldsymbol{x}}$ in $\boldsymbol{P}^{\boldsymbol{x}}$ is either 0 or color $\boldsymbol{x}$ ($\boldsymbol{p}^{\boldsymbol{x}} \in \mathbf{E}^{\boldsymbol{x}}$) for $\boldsymbol{x} \in \mathbf{E}$. The $\boldsymbol{x}$-colored halftoning procedure that transforms $P^{\boldsymbol{x}}$ into $\boldsymbol{P}^{\boldsymbol{x}}$ is represented as

$$\boldsymbol{P}^{\boldsymbol{x}} = \mathscr{H}^{\boldsymbol{x}}(P^{\boldsymbol{x}}),$$

where $p^{\boldsymbol{x}} \in \breve{\mathbf{E}}^{\boldsymbol{x}}$ and $\boldsymbol{p}^{\boldsymbol{x}} \in \mathbf{E}^{\boldsymbol{x}}$ for each pixel $p^{\boldsymbol{x}}$ in $P^{\boldsymbol{x}}$ and $\boldsymbol{p}^{\boldsymbol{x}}$ in $\boldsymbol{P}^{\boldsymbol{x}}$ and $\boldsymbol{x} \in \mathbf{E}$.

Since $\boldsymbol{p}^{\boldsymbol{x}} \in \mathbf{E}^{\boldsymbol{x}}$ ($=\{0, \boldsymbol{x}\}$) for $\boldsymbol{x} \in \mathbf{E}$, $(\boldsymbol{p}^{\mathbf{c}}, \boldsymbol{p}^{\mathbf{m}}, \boldsymbol{p}^{\mathbf{y}})$ has eight possible assignments which result in eight distinct colors (i.e., colors in **C**) for their mixing pixel $\boldsymbol{p}$ as shown in Table 4 (according to the rules in Table 3). Let $\mathscr{m}$: $\mathbf{E}^{\mathbf{c}} \times \mathbf{E}^{\mathbf{m}} \times \mathbf{E}^{\mathbf{y}} \to \mathbf{C}$ be the function defining the color mixture relationship between $(\boldsymbol{p}^{\mathbf{c}}, \boldsymbol{p}^{\mathbf{m}}, \boldsymbol{p}^{\mathbf{y}})$ and their mixing pixel $\boldsymbol{p}$ in the subtractive model. The *composition or mixing* of $\boldsymbol{p}^{\mathbf{c}}, \boldsymbol{p}^{\mathbf{m}}, \boldsymbol{p}^{\mathbf{y}}$ into $\boldsymbol{p}$ can be denoted by

$$\boldsymbol{p} = \mathscr{m}(\boldsymbol{p}^{\mathbf{c}}, \boldsymbol{p}^{\mathbf{m}}, \boldsymbol{p}^{\mathbf{y}}).$$

We call $\boldsymbol{p}^{\boldsymbol{x}}$ ($\in \mathbf{E}^{\boldsymbol{x}}$) the ***x*-colored halftone pixel** of $\boldsymbol{p}$ for $\boldsymbol{x} \in \mathbf{E}$. Let $\mathscr{M}$ denote the procedure that mixes the three corresponding monochromatic images $\boldsymbol{P}^{\mathbf{c}}$, $\boldsymbol{P}^{\mathbf{m}}$ and $\boldsymbol{P}^{\mathbf{y}}$ into $\boldsymbol{P}$ by applying function $\mathscr{m}$ onto all $\boldsymbol{p}^{\mathbf{c}}$'s, $\boldsymbol{p}^{\mathbf{m}}$'s and $\boldsymbol{p}^{\mathbf{y}}$'s correspondingly. We have

$$\boldsymbol{P} = \mathscr{M}(\boldsymbol{P}^{\mathbf{c}}, \boldsymbol{P}^{\mathbf{m}}, \boldsymbol{P}^{\mathbf{y}}).$$

Table 4 also reveals that when $\boldsymbol{p}^{\boldsymbol{x}} = 0$ or $\boldsymbol{x}$ is randomly determined for $\boldsymbol{x} \in \mathbf{E}$, the probability for $\boldsymbol{p}$ to be each of the eight colors in **C** is $\frac{1}{8}$. The following lemma holds from the observation given in Table 4.

Lemma 1. *For each pixel $\boldsymbol{p} = \mathscr{m}(\boldsymbol{p}^{\mathbf{c}}, \boldsymbol{p}^{\mathbf{m}}, \boldsymbol{p}^{\mathbf{y}})$ in halftone image $\boldsymbol{P}$ where $\boldsymbol{p}^{\boldsymbol{x}} \in \mathbf{E}^{\boldsymbol{x}}$ for $\boldsymbol{x} \in \mathbf{E}$,*

(1) $\boldsymbol{p} \in \mathbf{C}$; *and*
(2) *if $\mathscr{P}rob(\boldsymbol{p}^{\boldsymbol{x}} = 0) = \mathscr{P}rob(\boldsymbol{p}^{\boldsymbol{x}} = \boldsymbol{x}) = \frac{1}{2}$, $\mathscr{P}rob(\boldsymbol{p} = z) = \frac{1}{8}$ for each $z \in \mathbf{C}$.*

Lemma 1 implies that applying color decomposition on color image P, i.e., $\mathscr{D}(P) = (P^{\mathbf{c}}, P^{\mathbf{m}}, P^{\mathbf{y}})$, and halftoning $P^{\boldsymbol{x}}$ into $\boldsymbol{P}^{\boldsymbol{x}}$ ($\boldsymbol{P}^{\boldsymbol{x}} = \mathscr{H}^{\boldsymbol{x}}(P^{\boldsymbol{x}})$) for $\boldsymbol{x} \in \mathbf{E}$ cause $\boldsymbol{P}$ to contain at most the eight colors in **C** where $\boldsymbol{P} = \mathscr{M}(\boldsymbol{P}^{\mathbf{c}}, \boldsymbol{P}^{\mathbf{m}}, \boldsymbol{P}^{\mathbf{y}})$. Note that in this paper $p\,(p^{\boldsymbol{x}}, \boldsymbol{p}, \boldsymbol{p}^{\boldsymbol{x}})$ denotes some pixel in P ($P^{\boldsymbol{x}}$, $\boldsymbol{P}$, $\boldsymbol{P}^{\boldsymbol{x}}$, respectively) or a certain color painted on it interchangeably whenever there is no ambiguity.

Table 4
Possible colors mixed by **c**, **m** and **y**-colored halftone pixels

$\boldsymbol{p}^{\mathbf{c}}$	$\boldsymbol{p}^{\mathbf{m}}$	$\boldsymbol{p}^{\mathbf{y}}$	$\boldsymbol{p} = \mathscr{m}(\boldsymbol{p}^{\mathbf{c}}, \boldsymbol{p}^{\mathbf{m}}, \boldsymbol{p}^{\mathbf{y}})$	Mixing color
□	□	□	□	0
□	□	■	■	**y**
□	■	□	■	**m**
□	■	■	■	**r**
■	□	□	■	**c**
■	□	■	■	**g**
■	■	□	■	**b**
■	■	■	■	1

4.2. Encrypting color images by random grids

Consider a secret color image P, $\mathscr{D}(P) = (P^{\mathbf{c}}, P^{\mathbf{m}}, P^{\mathbf{y}})$ and $\boldsymbol{P}^{\boldsymbol{x}} = \mathscr{H}^{\boldsymbol{x}}(P^{\boldsymbol{x}})$ for $\boldsymbol{x} \in \mathbf{E}$. Since $\boldsymbol{p}^{\boldsymbol{x}} \in \mathbf{E}^{\boldsymbol{x}}$ in $\boldsymbol{P}^{\boldsymbol{x}}$ can be regarded as a binary (0 or $\boldsymbol{x}$) pixel, we can encrypt $\boldsymbol{P}^{\boldsymbol{x}}$ into two shares, namely $\boldsymbol{R}_1^{\boldsymbol{x}}$ and $\boldsymbol{R}_2^{\boldsymbol{x}}$, by exploiting the ideas of binary image encryption out of Algorithms 1, 2 or 3 so that neither $\boldsymbol{R}_1^{\boldsymbol{x}}$ nor $\boldsymbol{R}_2^{\boldsymbol{x}}$ leaks information about $\boldsymbol{P}^{\boldsymbol{x}}$, yet $\boldsymbol{S}^{\boldsymbol{x}} = \boldsymbol{R}_1^{\boldsymbol{x}} \otimes \boldsymbol{R}_2^{\boldsymbol{x}}$ reveals the secret in $\boldsymbol{P}^{\boldsymbol{x}}$ (and $P^{\boldsymbol{x}}$ consequently) for $\boldsymbol{x} \in \mathbf{E}$.

The basic ideas in Algorithms 1, 2 and 3 generate first R_1 as a binary random grid with $\mathscr{T}(\boldsymbol{R}_1) = \frac{1}{2}$ for binary image B. With regard to $\boldsymbol{P}^{\boldsymbol{x}}$, an $\boldsymbol{x}$-colored halftone (0 or $\boldsymbol{x}$) image, we hence generate an ***x*-color random grid** $\boldsymbol{R}_1^{\boldsymbol{x}}$ with $\mathscr{T}(\boldsymbol{R}_1^{\boldsymbol{x}}) = \frac{1}{2}$ in which each pixel $\boldsymbol{r}_1^{\boldsymbol{x}}$ in $\boldsymbol{R}_1^{\boldsymbol{x}}$ is either 0 (transparent) or $\boldsymbol{x}$, i.e., $\boldsymbol{r}_1^{\boldsymbol{x}} \in \mathbf{E}^{\boldsymbol{x}}$, and

$$\mathscr{P}rob(\boldsymbol{r}_1^{\boldsymbol{x}} = 0) = \mathscr{P}rob(\boldsymbol{r}_1^{\boldsymbol{x}} = \boldsymbol{x}) = \tfrac{1}{2}$$

for $\boldsymbol{x} \in \mathbf{E}$. We refer to $\boldsymbol{r}_1^{\boldsymbol{x}}$ as an ***x*-colored random pixel** in $\boldsymbol{R}_1^{\boldsymbol{x}}$.

As summarized in Table 2, once r_1 in R_1 is determined, there are three elementary operations for assigning binary colors (transparent or opaque) to the corresponding pixel r_2 in R_2 in Algorithms 1, 2 and 3: (1) $r_2 = r_1$, (2) $r_2 = \bar{r}_1$,

and (3) $r_2 = \text{random\_pixel}(0, 1)$. With regard to the encryption of x-color halftone pixel p^x in P^x on condition that r_1^x in R_1^x is determined, we simply modify these three operations accordingly as: (1) $r_2^x = r_1^x$, (2) $r_2^x = \overline{r_1^x}$, and (3) $r_2^x = \text{random\_pixel}(0, x)$ where r_1^x and r_2^x are the corresponding pixels in R_1^x and R_2^x, respectively, the inverse of $r_1^x \in \mathbf{E}^x$ is defined as

$$\overline{r_1^x} = \begin{cases} x & \text{if } r_1^x = 0, \\ 0 & \text{otherwise} (r_1^x = x), \end{cases}$$

and function random_pixel$(0, x)$ returns 0 or color x randomly.

Based upon Table 1, the superimposition of two independent x-colored random pixels r_1^x and r_2^x ($\in \mathbf{E}^x$) can be defined as follows

$$r_1^x \otimes r_2^x = \begin{cases} 0 & \text{if } r_1^x = r_2^x = 0, \\ x & \text{otherwise.} \end{cases}$$

Further, it is easy to obtain $\mathscr{P}rob(r_1^x \otimes r_2^x = 0) = \frac{1}{4}$. By using these notations, we have:

Lemma 2. *If r_1^x is an x-color random pixel and r_2^x is defined by*: (1) $r_2^x = r_1^x$, (2) $r_2^x = \overline{r_1^x}$, or (3) $r_2^x = \text{random\_pixel}(0, x)$, *respectively, then r_2^x is also an x-color random pixel and $s^x = r_1^x \otimes r_2^x$ satisfies*: (1) $\mathscr{P}rob(s^x = 0) = \frac{1}{2}$, (2) $\mathscr{P}rob(s^x = 0) = 0$, *or* (3) $\mathscr{P}rob(s^x = 0) = \frac{1}{4}$ *accordingly.*

Proof. Since r_1^x is an x-color random pixel, $\mathscr{P}rob(s^x = 0) = \mathscr{P}rob(s^x = x) = \frac{1}{2}$. When $r_2^x = r_1^x$, $s^x = r_1^x \otimes r_2^x = r_1^x$. We obtain $\mathscr{P}rob(r_2^x = 0) = \mathscr{P}rob(r_1^x = 0) = \frac{1}{2}$ and $\mathscr{P}rob(s^x = 0) = \mathscr{P}rob(r_1^x = 0) = \frac{1}{2}$. When $r_2^x = \overline{r_1^x}$, $s^x = r_1^x \otimes r_2^x = r_1^x \otimes \overline{r_1^x} = x$. We have $\mathscr{P}rob(r_2^x = 0) = \mathscr{P}rob(\overline{r_1^x} = 0) = \mathscr{P}rob(r_1^x = x) = \frac{1}{2}$ and $\mathscr{P}rob(s^x = 0) = \mathscr{P}rob(r_1^x \otimes \overline{r_1^x} = 0) = 0$. When $r_2^x = \text{random\_pixel}(0, x)$, both r_1^x and r_2^x are independent x-colored random pixels so that $\mathscr{P}rob(s^x = 0) = \frac{1}{4}$. □

Based upon Theorem 1 and Lemma 2, we can easily modify Algorithms 1, 2 or 3 to accomplish the encryption of P^x for $x \in \mathbf{E}$. Without losing generality, we tailor Algorithm 1 into Algorithm 5 in the following. Note that Algorithm 5 is designed as a procedure named *Encryption_color* that can be called by passing parameters x and P^x for $x \in \mathbf{E}$.

Algorithm 5. Encryption of an x-colored halftone image by x-colored random grids

Input: A color x and a $w \times h$ x-colored halftone image P^x where $P^x[i, j] \in \mathbf{E}^x$, $x \in \mathbf{E}$, $1 \leqslant i \leqslant w$ and $1 \leqslant j \leqslant h$

Output: Two shares of x-colored random grids R_1^x and R_2^x which reveal P^x when superimposed where $R_k^x[i, j] \in \mathbf{E}^x$, $x \in \mathbf{E}$, $1 \leqslant i \leqslant w$, $1 \leqslant j \leqslant h$ and $k \in \{1, 2\}$

Table 5
Encoding p^x into r_1^x and r_2^x and results of $s^x = r_1^x \otimes r_2^x$ by Algorithm 5

x	p^x	Probability	r_1^x	r_2^x	$s^x = r_1^x \otimes r_2^x$	$\mathscr{P}rob(s^x = 0)$ $(\ell(s^x))$
c	□	$\frac{1}{2}$	□	□	□	$\frac{1}{2}$
		$\frac{1}{2}$	■	■	■	
	■	$\frac{1}{2}$	□	■	■	0
		$\frac{1}{2}$	■	□	■	
m	□	$\frac{1}{2}$	□	□	□	$\frac{1}{2}$
		$\frac{1}{2}$	■	■	■	
	■	$\frac{1}{2}$	□	■	■	0
		$\frac{1}{2}$	■	□	■	
y	□	$\frac{1}{2}$	□	□	□	$\frac{1}{2}$
		$\frac{1}{2}$	■	■	■	
	■	$\frac{1}{2}$	□	■	■	0
		$\frac{1}{2}$	■	□	■	

Encryption_color(x, P^x)

1. Generate R_1^x as an x-colored random grid, $\mathscr{T}(R_1^x) = \frac{1}{2}$.
   // for (each pixel $R_1^x[i, j]$, $1 \leqslant i \leqslant w$ and $1 \leqslant j \leqslant h$) do
   // $R_1^x[i, j] = \text{random\_pixel}(0, x)$
2. for (each pixel $P^x[i, j]$, $1 \leqslant i \leqslant w$ and $1 \leqslant j \leqslant h$) do
2.1 { if $(P^x[i, j] = 0)$ $R_2^x[i, j] = R_1^x[i, j]$
   else $R_2^x[i, j] = \overline{R_1^x[i, j]}$
   }
3. output (R_1^x, R_2^x)

Table 5 illustrates the process of encoding a certain x-colored halfone pixel p^x in P^x into r_1^x in R_1^x and r_2^x in R_2^x and the results of $s^x = r_1^x \otimes r_2^x$ for $x \in \mathbf{E}$ by Algorithm 5.

We realize from Table 5 that the probability for reconstructed pixel $s^x = r_1^x \otimes r_2^x$ in $S^x = R_1^x \otimes R_2^x$ to let through the light, defined as the *average light transmission* of s^x, is

$$\ell(s^x) = \mathscr{P}rob(s^x = 0) = \begin{cases} \frac{1}{2} & \text{if } p^x = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $s^x, p^x \in \mathbf{E}^x$ and $x \in \mathbf{E}$.

When $\mathscr{P}rob(s^x = 0)$ $(= \ell(s^x))$ describes the average light transmission of s^x, $\mathscr{P}rob(s^x = x)$ $(= 1 - \mathscr{P}rob(s^x = 0))$ exposes the *average color intensity* of x on s^x with respect to p^x. In the following, we shall examine the possible mixing colors of $m(s^c, s^m, s^y)$ in terms of the average color intensity due to the reason that it would be more appropriate to describe the mixing colors in the subtractive model by the color intensity instead of the light transmission. Hence, we define the *average color intensity of x-colored pixel s^x* as

$$i(s^x) = \mathscr{P}rob(s^x = x) = 1 - \mathscr{P}rob(s^x = 0) = 1 - \ell(s^x) \tag{9}$$

and the *average color intensity of* $\boldsymbol{x}$*-colored grid* $\boldsymbol{S}^{\boldsymbol{x}}$ as

$$\mathscr{I}(\boldsymbol{S}^{\boldsymbol{x}}) = 1 - \mathscr{T}(\boldsymbol{S}^{\boldsymbol{x}})$$

for $\boldsymbol{x} \in \mathbf{E}$. From formulae (8) and (9), for $\boldsymbol{s}^{\boldsymbol{x}}, \boldsymbol{p}^{\boldsymbol{x}} \in \mathbf{E}^{\boldsymbol{x}}$ and $\boldsymbol{x} \in \mathbf{E}$ we have

$$\imath(\boldsymbol{s}^{\boldsymbol{x}}) = \begin{cases} \frac{1}{2} & \text{if } \boldsymbol{p}^{\boldsymbol{x}} = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

Theorem 3 demonstrates the correctness of Algorithm 5.

Theorem 3. *Given an* $\boldsymbol{x}$*-colored halftone image* $\boldsymbol{P}^{\boldsymbol{x}}$, $\boldsymbol{R}_1^{\boldsymbol{x}}$ *and* $\boldsymbol{R}_2^{\boldsymbol{x}}$ *generated by Algorithm* 5 *satisfy*:

(1) $\mathscr{T}(\boldsymbol{R}_1^{\boldsymbol{x}}) = \mathscr{T}(\boldsymbol{R}_2^{\boldsymbol{x}}) = \frac{1}{2}$ $(or \mathscr{I}(\boldsymbol{R}_1^{\boldsymbol{x}}) = \mathscr{I}(\boldsymbol{R}_2^{\boldsymbol{x}}) = \frac{1}{2})$; *and*
(2) $\mathscr{T}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]) > \mathscr{T}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})])$ $(or \mathscr{I}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})]) > \mathscr{I}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]))$ *where* $\boldsymbol{S}^{\boldsymbol{x}} = \boldsymbol{R}_1^{\boldsymbol{x}} \otimes \boldsymbol{R}_2^{\boldsymbol{x}}$;

so that no information of $\boldsymbol{P}^{\boldsymbol{x}}$ *can be obtained from* $\boldsymbol{R}_1^{\boldsymbol{x}}$ *and* $\boldsymbol{R}_2^{\boldsymbol{x}}$ *individually*; *while* $\boldsymbol{S}^{\boldsymbol{x}}$ *reveals* $\boldsymbol{P}^{\boldsymbol{x}}$ *in our visual system.*

Proof. In Algorithm 5, $\mathscr{T}(\boldsymbol{R}_1^{\boldsymbol{x}}) = \frac{1}{2}$ so that $\mathscr{T}(\boldsymbol{R}_1^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]) = \mathscr{T}(\boldsymbol{R}_1^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(1)]) = \frac{1}{2}$. Thus, $\mathscr{T}(\boldsymbol{R}_2^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]) = \mathscr{T}(\boldsymbol{R}_1^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]) = \frac{1}{2}$ and $\mathscr{T}(\boldsymbol{R}_2^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(1)]) = \mathscr{T}(\overline{\boldsymbol{R}_1^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(1)]}) = \frac{1}{2}$ (Lemma 2). Consequently, we have $\mathscr{T}(\boldsymbol{R}_2^{\boldsymbol{x}}) = \mathscr{T}(\boldsymbol{R}_1^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)] \cup \boldsymbol{R}_2^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]) = \frac{1}{2}$. Or $\mathscr{I}(\boldsymbol{R}_2^{\boldsymbol{x}}) = \frac{1}{2}$ $(= \mathscr{I}(\boldsymbol{R}_1^{\boldsymbol{x}}))$. It implies that both $\boldsymbol{R}_2^{\boldsymbol{x}}$ and $\boldsymbol{R}_2^{\boldsymbol{x}}$ are $\boldsymbol{x}$-colored random grids.

Since $\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)] = \boldsymbol{R}_1^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)] \otimes \boldsymbol{R}_2^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)] = \boldsymbol{R}_1^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]$ and $\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(1)] = \boldsymbol{R}_1^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(1)] \otimes \boldsymbol{R}_2^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(1)] = \boldsymbol{R}_1^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(1)] \otimes \overline{\boldsymbol{R}_1^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(1)]}$, we obtain $\mathscr{T}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]) = \frac{1}{2}$ and $\mathscr{T}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})]) = 0$, or equivalently, $\mathscr{I}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]) = \frac{1}{2}$ and $\mathscr{I}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})]) = 1$. Thus, our visual system sees pure color $\boldsymbol{x}$ in $\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})]$, while we see transparent or $\boldsymbol{x}$ with an equal probability in $\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]$. That is, $\boldsymbol{S}^{\boldsymbol{x}}$ reveals $\boldsymbol{P}^{\boldsymbol{x}}$ due to the difference between $\mathscr{T}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)])$ and $\mathscr{T}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})])$ (or $\mathscr{I}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)])$ and $\mathscr{I}(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})])$). We prove the theorem. □

Note that for each $\boldsymbol{p}^{\boldsymbol{x}}$ in $\boldsymbol{P}^{\boldsymbol{x}}$, its corresponding $\boldsymbol{s}^{\boldsymbol{x}}$ in $\boldsymbol{S}^{\boldsymbol{x}}$ is with $\imath(\boldsymbol{s}^{\boldsymbol{x}}) = 1$ when $\boldsymbol{p}^{\boldsymbol{x}} = \boldsymbol{x}$ and $\imath(\boldsymbol{s}^{\boldsymbol{x}}) = \frac{1}{2}$ otherwise for $\boldsymbol{x} \in \mathbf{E}$ (formula (10)). It means that $\boldsymbol{s}^{\boldsymbol{x}}$ in $\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})]$ fully recovers $\boldsymbol{p}^{\boldsymbol{x}}$ in $\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})$, while that in $\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]$ recovers $\boldsymbol{p}^{\boldsymbol{x}}$ in $\boldsymbol{P}^{\boldsymbol{x}}(0)$ only 50% on average. We may say that the difference of the average color intensities between $\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)]$ and $\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})]$ degrades 50% as compared to that between $\boldsymbol{P}^{\boldsymbol{x}}(0)$ and $\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})$ with respect to color $\boldsymbol{x}$.

Once $\boldsymbol{P}^{\boldsymbol{x}}$ is obtained, $P^{\boldsymbol{x}}$ from which $\boldsymbol{P}^{\boldsymbol{x}}$ is transformed can be seen visually. The results of the experiments will be presented in the next subsection. Based upon Lemma 2 and Theorem 3, the encryption algorithms for $\boldsymbol{x}$-colored image $\boldsymbol{P}^{\boldsymbol{x}}$ tailored from Algorithms 2 and 3 can be easily designed and their correctness can also be accordingly proved.

After $\boldsymbol{R}_1^{\mathbf{c}}$ and $\boldsymbol{R}_2^{\mathbf{c}}$, $\boldsymbol{R}_1^{\mathbf{m}}$ and $\boldsymbol{R}_2^{\mathbf{m}}$, as well as $\boldsymbol{R}_1^{\mathbf{y}}$ and $\boldsymbol{R}_2^{\mathbf{y}}$ are independently generated for $\boldsymbol{P}^{\mathbf{c}}$, $\boldsymbol{P}^{\mathbf{m}}$ and $\boldsymbol{P}^{\mathbf{y}}$, respectively, we then compose (mix) $\boldsymbol{R}_1^{\mathbf{c}}$, $\boldsymbol{R}_1^{\mathbf{m}}$ and $\boldsymbol{R}_1^{\mathbf{y}}$ as well as $\boldsymbol{R}_2^{\mathbf{c}}$, $\boldsymbol{R}_2^{\mathbf{m}}$ and $\boldsymbol{R}_2^{\mathbf{y}}$ into two colored grids, namely $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$, accordingly. That is,

$$\boldsymbol{R}_1 = \mathscr{M}(\boldsymbol{R}_1^{\mathbf{c}}, \boldsymbol{R}_1^{\mathbf{m}}, \boldsymbol{R}_1^{\mathbf{y}}) \quad \text{and} \quad \boldsymbol{R}_2 = \mathscr{M}(\boldsymbol{R}_2^{\mathbf{c}}, \boldsymbol{R}_2^{\mathbf{m}}, \boldsymbol{R}_2^{\mathbf{y}}).$$

Then we claim that $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are random grids with colors in $\mathbf{C}$ so that no information about P is exposed by $\boldsymbol{R}_1$ or $\boldsymbol{R}_2$ individually, while $\boldsymbol{S} = \boldsymbol{R}_1 \otimes \boldsymbol{R}_2$ recovers P visually.

The whole idea is summarized in Algorithm 6.

Algorithm 6. Encryption of a color image by color random grids

Input: A $w \times h$ color image P where $P[i, j] \in \breve{\mathbf{C}}$, $1 \leqslant i \leqslant w$ and $1 \leqslant j \leqslant h$
Output: Two shares of color random grids $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ which reveal P when superimposed where $\boldsymbol{R}_k[i, j] \in \mathbf{C}$, $1 \leqslant i \leqslant w$, $1 \leqslant j \leqslant h$ and $k \in \{1, 2\}$
1. Decompose P into $P^{\mathbf{y}}$, $P^{\mathbf{m}}$ and $P^{\mathbf{c}}$, that is, $\mathscr{D}(P) = (P^{\mathbf{y}}, P^{\mathbf{m}}, P^{\mathbf{c}})$
2. for (each $\boldsymbol{x} \in \mathbf{E}$) do $\boldsymbol{P}^{\boldsymbol{x}} = \mathscr{H}^{\boldsymbol{x}}(P^{\boldsymbol{x}})$ // Transform $P^{\boldsymbol{x}}$ into $\boldsymbol{x}$-colored halftone image $\boldsymbol{P}^{\boldsymbol{x}}$
3. for (each $\boldsymbol{x} \in \mathbf{E}$) do $(\boldsymbol{R}_1^{\boldsymbol{x}}, \boldsymbol{R}_2^{\boldsymbol{x}}) = \mathit{Encryption\_color}(\boldsymbol{x}, \boldsymbol{P}^{\boldsymbol{x}})$
4. $\boldsymbol{R}_1 = \mathscr{M}(\boldsymbol{R}_1^{\mathbf{c}}, \boldsymbol{R}_1^{\mathbf{m}}, \boldsymbol{R}_1^{\mathbf{y}})$
5. $\boldsymbol{R}_2 = \mathscr{M}(\boldsymbol{R}_2^{\mathbf{c}}, \boldsymbol{R}_2^{\mathbf{m}}, \boldsymbol{R}_2^{\mathbf{y}})$
6. output$(\boldsymbol{R}_1, \boldsymbol{R}_2)$

Note that $\mathit{Encryption\_color}(\boldsymbol{x}, \boldsymbol{P}^{\boldsymbol{x}})$ can be implemented by Algorithm 5, the tailored version of Algorithm 1, or other approaches, such as the tailored versions of Algorithms 2 or 3. In the following, if not specified explicitly, we implement it by Algorithm 5.

Due to the independence of the three primary colors, it is reasonable to represent the *average light transmission and average color intensity* of $\boldsymbol{R} = \mathscr{M}(\boldsymbol{R}^{\mathbf{c}}, \boldsymbol{R}^{\mathbf{m}}, \boldsymbol{R}^{\mathbf{y}})$ as 3-tuple vectors in terms of the transmissions and color intensities of $\boldsymbol{R}^{\mathbf{c}}$, $\boldsymbol{R}^{\mathbf{m}}$ and $\boldsymbol{R}^{\mathbf{y}}$, respectively:

$$\mathscr{T}(\boldsymbol{R}) = (\mathscr{T}(\boldsymbol{R}^{\mathbf{c}}), \mathscr{T}(\boldsymbol{R}^{\mathbf{m}}), \mathscr{T}(\boldsymbol{R}^{\mathbf{y}})) \quad \text{and}$$
$$\mathscr{I}(\boldsymbol{R}) = (\mathscr{I}(\boldsymbol{R}^{\mathbf{c}}), \mathscr{I}(\boldsymbol{R}^{\mathbf{m}}), \mathscr{I}(\boldsymbol{R}^{\mathbf{y}})).$$

Moreover, we define $\boldsymbol{R} = \mathscr{M}(\boldsymbol{R}^{\mathbf{c}}, \boldsymbol{R}^{\mathbf{m}}, \boldsymbol{R}^{\mathbf{y}})$ to be a *color random grid* if and only if

$$\mathscr{T}(\boldsymbol{R}^{\mathbf{c}}) = \mathscr{T}(\boldsymbol{R}^{\mathbf{m}}) = \mathscr{T}(\boldsymbol{R}^{\mathbf{y}}) = \tfrac{1}{2} \left(\mathscr{T}(\boldsymbol{R}) = \left(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}\right)\right),$$

or equivalently,

$$\mathscr{I}(\boldsymbol{R}^{\mathbf{c}}) = \mathscr{I}(\boldsymbol{R}^{\mathbf{m}}) = \mathscr{I}(\boldsymbol{R}^{\mathbf{y}}) = \tfrac{1}{2} \quad \left(\mathscr{I}(\boldsymbol{R}) = \left(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}\right)\right).$$

Each pixel $\boldsymbol{r} = m(\boldsymbol{r}^{\mathbf{c}}, \boldsymbol{r}^{\mathbf{m}}, \boldsymbol{r}^{\mathbf{y}})$, referred to as a *color random pixel*, in $\boldsymbol{R}$ satisfies

$$\mathscr{P}rob(\boldsymbol{r}^{\boldsymbol{x}} = 0) = \mathscr{P}rob(\boldsymbol{r}^{\boldsymbol{x}} = \boldsymbol{x}) = \tfrac{1}{2},$$
i.e.,
$$t(\boldsymbol{r}^{\boldsymbol{x}}) = \tfrac{1}{2}; \text{ or equivalently, } \imath(\boldsymbol{r}^{\boldsymbol{x}}) = \tfrac{1}{2},$$

Table 6
All possible assignments of colors to $\boldsymbol{r}_1 = m(\boldsymbol{r}_1^{c}, \boldsymbol{r}_1^{m}, \boldsymbol{r}_1^{y})$ and $\boldsymbol{r}_2 = m(\boldsymbol{r}_2^{c}, \boldsymbol{r}_2^{m}, \boldsymbol{r}_2^{y})$ with respect to $\boldsymbol{p} = m(\boldsymbol{p}^{c}, \boldsymbol{p}^{m}, \boldsymbol{p}^{y})$, results of $\boldsymbol{s} = \boldsymbol{r}_1 \otimes \boldsymbol{r}_2$, $\ell(\boldsymbol{s})$ and $\imath(\boldsymbol{s})$ by Algorithm 6

$\boldsymbol{p}$	probability	(r_1^c, r_1^m, r_1^y)	(r_2^c, r_2^m, r_2^y)	(s^c, s^m, s^y)	$\boldsymbol{s}$	$\ell(\boldsymbol{s})$	$\imath(\boldsymbol{s})$
white (□, □, □)	$\frac{1}{8}$	(□, □, □)	(□, □, □)	(□, □, □)	white	$(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$
	$\frac{1}{8}$	(□, □, Y)	(□, □, Y)	(□, □, Y)	yellow		
	$\frac{1}{8}$	(□, M, □)	(□, M, □)	(□, M, □)	magenta		
	$\frac{1}{8}$	(□, M, Y)	(□, M, Y)	(□, M, Y)	red		
	$\frac{1}{8}$	(C, □, □)	(C, □, □)	(C, □, □)	cyan		
	$\frac{1}{8}$	(C, □, Y)	(C, □, Y)	(C, □, Y)	green		
	$\frac{1}{8}$	(C, M, □)	(C, M, □)	(C, M, □)	blue		
	$\frac{1}{8}$	(C, M, Y)	(C, M, Y)	(C, M, Y)	black		
yellow (□, □, Y)	$\frac{1}{8}$	(□, □, □)	(□, □, Y)	(□, □, Y)	yellow	$(\frac{1}{2}, \frac{1}{2}, 0)$	$(\frac{1}{2}, \frac{1}{2}, 1)$
	$\frac{1}{8}$	(□, □, Y)	(□, □, □)	(□, □, Y)	yellow		
	$\frac{1}{8}$	(□, M, □)	(□, M, Y)	(□, M, Y)	red		
	$\frac{1}{8}$	(□, M, Y)	(□, M, □)	(□, M, Y)	red		
	$\frac{1}{8}$	(C, □, □)	(C, □, Y)	(C, □, Y)	green		
	$\frac{1}{8}$	(C, □, Y)	(C, □, □)	(C, □, Y)	green		
	$\frac{1}{8}$	(C, M, □)	(C, M, Y)	(C, M, Y)	black		
	$\frac{1}{8}$	(C, M, Y)	(C, M, □)	(C, M, Y)	black		
magenta (□, M, □)	$\frac{1}{8}$	(□, □, □)	(□, M, □)	(□, M, □)	magenta	$(\frac{1}{2}, 0, \frac{1}{2})$	$(\frac{1}{2}, 1, \frac{1}{2})$
	$\frac{1}{8}$	(□, □, Y)	(□, M, Y)	(□, M, Y)	red		
	$\frac{1}{8}$	(□, M, □)	(□, □, □)	(□, M, □)	magenta		
	$\frac{1}{8}$	(□, M, Y)	(□, □, Y)	(□, M, Y)	red		
	$\frac{1}{8}$	(C, □, □)	(C, M, □)	(C, M, □)	blue		
	$\frac{1}{8}$	(C, □, Y)	(C, M, Y)	(C, M, Y)	black		
	$\frac{1}{8}$	(C, M, □)	(C, □, □)	(C, M, □)	blue		
	$\frac{1}{8}$	(C, M, Y)	(C, □, Y)	(C, M, Y)	black		
red (□, M, Y)	$\frac{1}{8}$	(□, □, □)	(□, M, Y)	(□, M, Y)	red	$(\frac{1}{2}, 0, 0)$	$(\frac{1}{2}, 1, 1)$
	$\frac{1}{8}$	(□, □, Y)	(□, M, □)	(□, M, Y)	red		
	$\frac{1}{8}$	(□, M, □)	(□, □, Y)	(□, M, Y)	red		
	$\frac{1}{8}$	(□, M, Y)	(□, □, □)	(□, M, Y)	red		
	$\frac{1}{8}$	(C, □, □)	(C, M, Y)	(C, M, Y)	black		
	$\frac{1}{8}$	(C, □, Y)	(C, M, □)	(C, M, Y)	black		
	$\frac{1}{8}$	(C, M, □)	(C, □, Y)	(C, M, Y)	black		
	$\frac{1}{8}$	(C, M, Y)	(C, □, □)	(C, M, Y)	black		
cyan (C, □, □)	$\frac{1}{8}$	(□, □, □)	(C, □, □)	(C, □, □)	cyan	$(0, \frac{1}{2}, \frac{1}{2})$	$(1, \frac{1}{2}, \frac{1}{2})$
	$\frac{1}{8}$	(□, □, Y)	(C, □, Y)	(C, □, Y)	green		
	$\frac{1}{8}$	(□, M, □)	(C, M, □)	(C, M, □)	blue		
	$\frac{1}{8}$	(□, M, Y)	(C, M, Y)	(C, M, Y)	black		
	$\frac{1}{8}$	(C, □, □)	(□, □, □)	(C, □, □)	cyan		
	$\frac{1}{8}$	(C, □, Y)	(□, □, Y)	(C, □, Y)	green		
	$\frac{1}{8}$	(C, M, □)	(□, M, □)	(C, M, □)	blue		
	$\frac{1}{8}$	(C, M, Y)	(□, M, Y)	(C, M, Y)	black		
green (C, □, Y)	$\frac{1}{8}$	(□, □, □)	(C, □, Y)	(C, □, Y)	green	$(0, \frac{1}{2}, 0)$	$(1, \frac{1}{2}, 1)$
	$\frac{1}{8}$	(□, □, Y)	(C, □, □)	(C, □, Y)	green		
	$\frac{1}{8}$	(□, M, □)	(C, M, Y)	(C, M, Y)	black		
	$\frac{1}{8}$	(□, M, Y)	(C, M, □)	(C, M, Y)	black		
	$\frac{1}{8}$	(C, □, □)	(□, □, Y)	(C, □, Y)	green		
	$\frac{1}{8}$	(C, □, Y)	(□, □, □)	(C, □, Y)	green		
	$\frac{1}{8}$	(C, M, □)	(□, M, Y)	(C, M, Y)	black		
	$\frac{1}{8}$	(C, M, Y)	(□, M, □)	(C, M, Y)	black		

Table 6 (*continued*)

p	probability	(r_1^c, r_1^m, r_1^y)	(r_2^c, r_2^m, r_2^y)	(s^c, s^m, s^y)	s	$t(s)$	$i(s)$
	$\frac{1}{8}$					$(0,0,\frac{1}{2})$	$(1,1,\frac{1}{2})$
	$\frac{1}{8}$						
	$\frac{1}{8}$						
	$\frac{1}{8}$						
	$\frac{1}{8}$						
	$\frac{1}{8}$						
	$\frac{1}{8}$						
	$\frac{1}{8}$						
	$\frac{1}{8}$					$(0,0,0)$	$(1,1,1)$
	$\frac{1}{8}$						
	$\frac{1}{8}$						
	$\frac{1}{8}$						
	$\frac{1}{8}$						
	$\frac{1}{8}$						

for $\boldsymbol{x} \in \mathbf{E}$. In the same way, we denote the average light transmission and average color intensity of $\boldsymbol{r}$ in $\boldsymbol{R}$ as follows, respectively:

$$t(\boldsymbol{r}) = (t(\boldsymbol{r}^{\mathbf{c}}), t(\boldsymbol{r}^{\mathbf{m}}), t(\boldsymbol{r}^{\mathbf{y}})) = (\mathscr{P}rob(\boldsymbol{r}^{\mathbf{c}} = 0), \mathscr{P}rob(\boldsymbol{r}^{\mathbf{m}} = 0), \mathscr{P}rob(\boldsymbol{r}^{\mathbf{y}} = 0))$$

and

$$i(\boldsymbol{r}) = (i(\boldsymbol{r}^{\mathbf{c}}), i(\boldsymbol{r}^{\mathbf{m}}), i(\boldsymbol{r}^{\mathbf{y}})) = (\mathscr{P}rob(\boldsymbol{r}^{\mathbf{c}} = \mathbf{c}), \mathscr{P}rob(\boldsymbol{r}^{\mathbf{m}} = \mathbf{m}), \mathscr{P}rob(\boldsymbol{r}^{\mathbf{y}} = \mathbf{y})).$$

Based upon the concept of the *color random grid* (*pixel*), we have Lemma 3.

Lemma 3. *Given a color image P, $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ generated by Algorithm* 6 *satisfies*:

(1) $\mathscr{I}(\boldsymbol{R}_1) = \mathscr{I}(\boldsymbol{R}_2) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$; *and*
(2) $\mathscr{P}rob(\boldsymbol{r}_k = z) = \frac{1}{8}$ *for each color* $z \in \mathbf{C}$ *where pixel* $\boldsymbol{r}_k = m(\boldsymbol{r}_k^{\mathbf{c}}, \boldsymbol{r}_k^{\mathbf{m}}, \boldsymbol{r}_k^{\mathbf{y}})$ *is in* $\boldsymbol{R}_k = \mathscr{M}(\boldsymbol{R}_k^{\mathbf{c}}, \boldsymbol{R}_k^{\mathbf{m}}, \boldsymbol{R}_k^{\mathbf{y}})$ *and* $\boldsymbol{r}_k^{\boldsymbol{x}}$ *is in* $\boldsymbol{R}_k^{\boldsymbol{x}}$ *for* $\boldsymbol{x} \in \mathbf{E}$ *and* $k \in \{1, 2\}$.

Proof. By Theorem 3, we realize that $\boldsymbol{R}_1^{\boldsymbol{x}}$ and $\boldsymbol{R}_2^{\boldsymbol{x}}$ generated by *Encryption_color* (Algorithm 5) are with $\mathscr{T}(\boldsymbol{R}_1^{\boldsymbol{x}}) = \mathscr{T}(\boldsymbol{R}_2^{\boldsymbol{x}}) = \frac{1}{2}$ for $\boldsymbol{x} \in \mathbf{E}$. In Algorithm 6, $\boldsymbol{R}_1 = \mathscr{M}(\boldsymbol{R}_1^{\mathbf{c}}, \boldsymbol{R}_1^{\mathbf{m}}, \boldsymbol{R}_1^{\mathbf{y}})$ and $\boldsymbol{R}_2 = \mathscr{M}(\boldsymbol{R}_2^{\mathbf{c}}, \boldsymbol{R}_2^{\mathbf{m}}, \boldsymbol{R}_2^{\mathbf{y}})$, thus, $\mathscr{I}(\boldsymbol{R}_1) = (\mathscr{I}(\boldsymbol{R}_1^{\mathbf{c}}), \mathscr{I}(\boldsymbol{R}_1^{\mathbf{m}}), \mathscr{I}(\boldsymbol{R}_1^{\mathbf{y}})) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ and $\mathscr{I}(\boldsymbol{R}_2) = (\mathscr{I}(\boldsymbol{R}_2^{\mathbf{c}}), \mathscr{I}(\boldsymbol{R}_2^{\mathbf{m}}), \mathscr{I}(\boldsymbol{R}_2^{\mathbf{y}})) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. That is, they are both color random grids.

Since $\boldsymbol{r}_1^{\boldsymbol{x}}(\boldsymbol{r}_2^{\boldsymbol{x}}) \in \mathbf{E}^{\boldsymbol{x}}$ and $\mathscr{P}rob(\boldsymbol{r}_1^{\boldsymbol{x}} = 0) = \mathscr{P}rob(\boldsymbol{r}_1^{\boldsymbol{x}} = \boldsymbol{x}) = \frac{1}{2}$ ($\mathscr{P}rob(\boldsymbol{r}_2^{\boldsymbol{x}} = 0) = \mathscr{P}rob(\boldsymbol{r}_2^{\boldsymbol{x}} = \boldsymbol{x}) = \frac{1}{2}$) for each pixel $\boldsymbol{r}_1 = m(\boldsymbol{r}_1^{\mathbf{c}}, \boldsymbol{r}_1^{\mathbf{m}}, \boldsymbol{r}_1^{\mathbf{y}})(\boldsymbol{r}_2 = m(\boldsymbol{r}_2^{\mathbf{c}}, \boldsymbol{r}_2^{\mathbf{m}}, \boldsymbol{r}_2^{\mathbf{y}}))$ in $\boldsymbol{R}_1(\boldsymbol{R}_2)$, by Lemma 1 we know that $\mathscr{P}rob(\boldsymbol{r}_1 = z) = \frac{1}{8}$ ($\mathscr{P}rob(\boldsymbol{r}_2 = z) = \frac{1}{8}$) for each color $z \in \mathbf{C}$. □

Now let us focus on the colors that may appear on $\boldsymbol{s} = \boldsymbol{r}_1 \otimes \boldsymbol{r}_2$ (or more precisely, the color intensities of $\boldsymbol{s}$) with respect to $\boldsymbol{p}$ which may be any of the eight colors in $\mathbf{C}$ in Algorithm 6. We enumerate in Table 6 all possible assignments of colors to $\boldsymbol{r}_1^{\boldsymbol{x}}$ and $\boldsymbol{r}_2^{\boldsymbol{x}}$ for $\boldsymbol{x} \in \mathbf{E}$ that mix up $\boldsymbol{r}_1 = m(\boldsymbol{r}_1^{\mathbf{c}}, \boldsymbol{r}_1^{\mathbf{m}}, \boldsymbol{r}_1^{\mathbf{y}})$ and $\boldsymbol{r}_2 = m(\boldsymbol{r}_2^{\mathbf{c}}, \boldsymbol{r}_2^{\mathbf{m}}, \boldsymbol{r}_2^{\mathbf{y}})$, respectively, with respect to $\boldsymbol{p} = m(\boldsymbol{p}^{\mathbf{c}}, \boldsymbol{p}^{\mathbf{m}}, \boldsymbol{p}^{\mathbf{y}})$, the superimposed result of $\boldsymbol{s} = \boldsymbol{r}_1 \otimes \boldsymbol{r}_2$, $t(\boldsymbol{s})$ and $i(\boldsymbol{s})$ by Algorithm 6.

Table 6 demonstrates that all of the possible color combinations for the eight reconstructed pixels $\boldsymbol{s}$'s with respect to $\boldsymbol{p} = 0, \boldsymbol{p} = \mathbf{y}, \boldsymbol{p} = \mathbf{m}, \boldsymbol{p} = \mathbf{r}, \boldsymbol{p} = \mathbf{c}, \boldsymbol{p} = \mathbf{g}, \boldsymbol{p} = \mathbf{b}$ and $\boldsymbol{p} = 1$, respectively, are different from each other. We highlight such a significant finding about the colors of the reconstructed pixels by Algorithm 6 as a lemma.

Lemma 4. *Given two distinct pixels $\boldsymbol{p}$ and $\boldsymbol{p}'$ in $\boldsymbol{P}(u)$ and $\boldsymbol{P}(v)$, respectively, (or simply $\boldsymbol{p} = \boldsymbol{u}$ and $\boldsymbol{p}' = \boldsymbol{v}$) for $\boldsymbol{u}, \boldsymbol{v} \in \mathbf{C}$, if $\boldsymbol{u} \neq \boldsymbol{v}$, then $i(\boldsymbol{s}) \neq i(\boldsymbol{s}')$ where $\boldsymbol{s}$ ($\boldsymbol{s}'$) in $\boldsymbol{S} = \boldsymbol{R}_1 \otimes \boldsymbol{R}_2$ is the reconstructed pixels of $\boldsymbol{p}$ ($\boldsymbol{p}'$), and $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are produced by Algorithm* 6.

Consider any color image P with $\mathscr{D}(P) = (P^{\mathbf{c}}, P^{\mathbf{m}}, P^{\mathbf{y}})$. Lemma 1 indicates that $\boldsymbol{P} = \mathscr{M}(\boldsymbol{P}^{\mathbf{c}}, \boldsymbol{P}^{\mathbf{m}}, \boldsymbol{P}^{\mathbf{y}})$ may contain at most the eight colors in $\mathbf{C}$ where $\boldsymbol{P}^{\boldsymbol{x}} = \mathscr{H}^{\boldsymbol{x}}(P^{\boldsymbol{x}})$ for $\boldsymbol{x} \in \mathbf{E}$. Lemma 3 shows that $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ produced by Algorithm 6 with respect to P are color random grids (containing colors in $\mathbf{C}$ randomly) and each of them leaks no information about $\boldsymbol{P}$. Lemma 4 implies that given any pair of two areas $\boldsymbol{P}(\boldsymbol{u})$ and $\boldsymbol{P}(\boldsymbol{v})$ in $\boldsymbol{P}$ for $\boldsymbol{u}, \boldsymbol{v} \in \mathbf{C}$ and $\boldsymbol{u} \neq \boldsymbol{v}$, we can tell the

difference between $\mathscr{I}(\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{u})])$ and $\mathscr{I}(\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{v})])$ by our visual system where $\boldsymbol{S}=\boldsymbol{R}_1\otimes\boldsymbol{R}_2$. Therefore, different colors in P can be identified from $\boldsymbol{S}$ because their corresponding reconstructed colors are different from each other in $\boldsymbol{S}$ visually. The following theorem is an immediate consequence of Lemmas 1, 3 and 4.

Theorem 4. *Given a secret color image P, $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ generated by Algorithms* 6 *satisfy*:

(1) $\mathscr{T}(\boldsymbol{R}_1)=\mathscr{T}(\boldsymbol{R}_2)=\frac{1}{2}$; *and*
(2) $\mathscr{I}(\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{u})])\neq\mathscr{I}(\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{v})])$ *for* $\boldsymbol{u}\neq\boldsymbol{v}$, $\boldsymbol{u},\boldsymbol{v}\in\mathbf{C}$ *where* $\boldsymbol{P}=\mathscr{M}(\boldsymbol{P}^{\mathbf{c}},\boldsymbol{P}^{\mathbf{m}},\boldsymbol{P}^{\mathbf{y}})$, $\boldsymbol{P}^{\boldsymbol{x}}=\mathscr{H}^{\boldsymbol{x}}(P^{\boldsymbol{x}})$ *for* $\boldsymbol{x}\in\mathbf{E}$, $\mathscr{D}(P)=(P^{\mathbf{c}},P^{\mathbf{m}},P^{\mathbf{y}})$ *and* $\boldsymbol{S}=\boldsymbol{R}_1\otimes\boldsymbol{R}_2$;

so that no information of P can be obtained from $\boldsymbol{R}_1$ or $\boldsymbol{R}_2$ individually; *while $\boldsymbol{S}$ reveals P in our visual system.*

4.3. Experimental results and discussions for encrypting color images

Fig. 6 shows the results of the color decomposition and halftoning of color image P. With regard to P in Fig. 6(a), (b), (c) and (d) are the monochromatic images $P^{\mathbf{c}}$, $P^{\mathbf{m}}$ and $P^{\mathbf{y}}$, respectively, where $\mathscr{D}(P)=(P^{\mathbf{c}},P^{\mathbf{m}},P^{\mathbf{y}})$ and Fig. 6(e), (f) and (g) show the $\boldsymbol{x}$-colored halftone $\boldsymbol{P}^{\mathbf{c}}$, $\boldsymbol{P}^{\mathbf{m}}$ and $\boldsymbol{P}^{\mathbf{y}}$, respectively, in which $\boldsymbol{P}^{\boldsymbol{x}}=\mathscr{H}^{\boldsymbol{x}}(P^{\boldsymbol{x}})$ for $\boldsymbol{x}\in\mathbf{E}$.

Fig. 7 illustrates the results of the implementation of Algorithm 5 on $\boldsymbol{P}^{\mathbf{c}}$, $\boldsymbol{P}^{\mathbf{m}}$ and $\boldsymbol{P}^{\mathbf{y}}$ (i.e., Fig. 6(e), (f) and (g), respectively). Fig. 7(a) and (b) are encrypted shares $\boldsymbol{R}_1^{\mathbf{c}}$ and $\boldsymbol{R}_2^{\mathbf{c}}$ of $\boldsymbol{P}^{\mathbf{c}}$, respectively, and (c) shows $\boldsymbol{S}^{\mathbf{c}}=\boldsymbol{R}_1^{\mathbf{c}}\otimes\boldsymbol{R}_2^{\mathbf{c}}$ which reveals $\boldsymbol{P}^{\mathbf{c}}$. Fig. 7(d) and (e) illustrate encrypted shares $\boldsymbol{R}_1^{\mathbf{m}}$ and $\boldsymbol{R}_2^{\mathbf{m}}$ of $\boldsymbol{P}^{\mathbf{m}}$, respectively, and (f) gives $\boldsymbol{S}^{\mathbf{m}}=\boldsymbol{R}_1^{\mathbf{m}}\otimes\boldsymbol{R}_2^{\mathbf{m}}$ which reveals $\boldsymbol{P}^{\mathbf{m}}$. Fig. 7(g) and (h) present encrypted shares $\boldsymbol{R}_1^{\mathbf{y}}$ and $\boldsymbol{R}_2^{\mathbf{y}}$ of $\boldsymbol{P}^{\mathbf{y}}$, respectively, and (i) is $\boldsymbol{S}^{\mathbf{y}}=\boldsymbol{R}_1^{\mathbf{y}}\otimes\boldsymbol{R}_2^{\mathbf{y}}$ which reveals $\boldsymbol{P}^{\mathbf{y}}$. Actually Fig. 7 gives the visualized demonstrations for the correctness of Algorithm 5 (proved in Theorem 3).

Fig. 8 illustrates the results of the implementation of Algorithm 6 for encrypting the color image P shown in Fig. 6(a). Fig. 8(a) and (b) show the shares $\boldsymbol{R}_1=\mathscr{M}(\boldsymbol{R}_1^{\mathbf{c}},\boldsymbol{R}_1^{\mathbf{m}},\boldsymbol{R}_1^{\mathbf{y}})$ and $\boldsymbol{R}_2=\mathscr{M}(\boldsymbol{R}_2^{\mathbf{c}},\boldsymbol{R}_2^{\mathbf{m}},\boldsymbol{R}_2^{\mathbf{y}})$, respectively, where $\boldsymbol{R}_i^{\boldsymbol{k}}$'s can be found in Fig. 7 for $\boldsymbol{x}\in\mathbf{E}$ and $k\in\{1,2\}$, which are both color random grids with $\mathscr{I}(\boldsymbol{R}_1)=\mathscr{I}(\boldsymbol{R}_2)=(\frac{1}{2},\frac{1}{2},\frac{1}{2})$. Fig. 8(c) illustrates the reconstructed image $\boldsymbol{S}=\boldsymbol{R}_1\otimes\boldsymbol{R}_2$ which reveals P.

To examine the colors in the secret image against those in the reconstructed image, we tested Fig. 9(a) on purpose, which contains eight areas of pixels corresponding to the eight colors in $\mathbf{C}$ exactly. Fig. 9 shows the results of the implementation of Algorithm 6. The two secret shares $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are shown in Fig. 9(b) and (c) which are merely color random grids, while the superimposed image $\boldsymbol{S}=\boldsymbol{R}_1\otimes\boldsymbol{R}_2$ is given in Fig. 9(d). The original color image (Fig. 9(a)) can be visually recognized from the reconstructed image (Fig. 9(d)). The eight areas of the reconstructed color pixels in Fig. 9(d) correspond to the eight areas of the original color pixels in Fig. 9(a) one by one. No color would be recognized as another one by our visual system.

Let us focus our attention on the area of pixels in $\boldsymbol{P}$ with some color $\boldsymbol{z}$, i.e., $\boldsymbol{P}(\boldsymbol{z})$, and its corresponding area of pixels in $\boldsymbol{S}$, i.e., $\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{z})]$, for $\boldsymbol{z}\in\mathbf{C}$. We call $\boldsymbol{x}$ ($\in\mathbf{E}$) the *composing component* of $\boldsymbol{z}=\mathscr{m}(\boldsymbol{z}^{\mathbf{c}},\boldsymbol{z}^{\mathbf{m}},\boldsymbol{z}^{\mathbf{y}})$, if $\boldsymbol{z}^{\boldsymbol{x}}=\boldsymbol{x}$; while the *non-composing component* of $\boldsymbol{z}$, otherwise ($\boldsymbol{z}^{\boldsymbol{x}}=0$). Let $\boldsymbol{p}=\mathscr{m}(\boldsymbol{p}^{\mathbf{c}},\boldsymbol{p}^{\mathbf{m}},\boldsymbol{p}^{\mathbf{y}})$ and $\boldsymbol{s}=\mathscr{m}(\boldsymbol{s}^{\mathbf{c}},\boldsymbol{s}^{\mathbf{m}},\boldsymbol{s}^{\mathbf{y}})$ denote the corresponding pixels in $\boldsymbol{P}(\boldsymbol{z})$ and $\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{z})]$, respectively. By applying formula (10), we know that $\mathscr{i}(\boldsymbol{s}^{\boldsymbol{x}})=1$ if $\boldsymbol{p}^{\boldsymbol{x}}=\boldsymbol{x}$; while $\mathscr{i}(\boldsymbol{s}^{\boldsymbol{x}})=\frac{1}{2}$ otherwise for $\boldsymbol{p}^{\boldsymbol{x}}\in\mathbf{E}^{\boldsymbol{x}}$ and $\boldsymbol{x}\in\mathbf{E}$. That is, all of the pixels in $\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{z})]$ preserve the color intensities for all composing components of $\boldsymbol{z}$ with a degradation of 50% in the difference of color intensities for each of the non-composing components of $\boldsymbol{z}$. Our visual system still recognize $\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{z})]$ as color $\boldsymbol{z}$ rather than any other in $\mathbf{C}\backslash\{\boldsymbol{z}\}$. For instance, consider $\boldsymbol{p}=\mathbf{b}$ with $\mathscr{i}(\boldsymbol{s})=(1,1,\frac{1}{2})$ in Table 6. All reconstructed pixels in $\boldsymbol{S}[\boldsymbol{P}(\mathbf{b})]$ preserve both $\mathbf{c}$ and $\mathbf{m}$ (composing components of $\mathbf{b}$), while 50% of them have $\mathbf{y}$ component (non-composing component of $\mathbf{b}$). There is a 50% degradation with respect to $\mathbf{y}$ (whereas no degradation with respect to $\mathbf{c}$ and $\mathbf{m}$) between pixels in $\boldsymbol{P}(\mathbf{b})$ and those in $\boldsymbol{S}[\boldsymbol{P}(\mathbf{b})]$. Thus, $\boldsymbol{S}[\boldsymbol{P}(\mathbf{b})]$ looks like $\mathbf{b}$ as opposed to others in $\mathbf{C}$ (see Fig. 9(d)).

Following the experiment in Fig. 9, Fig. 10(a) and (b) show the reconstructed results of Algorithm 6 by implementing *Encryption_color* with the ideas out of Algorithms 2 and 3, respectively. It is seen that all of the three reconstructed results (Figs. 9(d), 10(a) and (b)) reveal the information of Fig. 9(a). Yet further observation discloses that the differences among the color intensities in Fig. 9(d) are easier to recognize by our visual system than those in both of Fig. 10(a) and (b). In general, it is not hard to see that the order of the performances (on the recognition of the reconstructed results) among Algorithms 1, 2 and 3, when they are designed for encrypting the color images, is the same as that when they are applied for encrypting the binary images.

In fact, when *Encryption_color* is implemented by using ideas in Algorithms 1, 2 and 3, $(\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(0)],\boldsymbol{S}^{\boldsymbol{x}}[\boldsymbol{P}^{\boldsymbol{x}}(\boldsymbol{x})])=(\frac{1}{2},1),(\frac{1}{2},\frac{3}{4})$ and $(\frac{3}{4},1)$, respectively, for $\boldsymbol{x}\in\mathbf{E}$ in Algorithm 6. Consequently, the color intensity $\mathscr{i}(\boldsymbol{s})$ with respect to $\boldsymbol{p}\in\mathbf{C}$ in $\boldsymbol{P}$ for pixel $\boldsymbol{s}$ in $\boldsymbol{S}$ regarding Algorithms 1, 2 or 3 can be easily computed as shown in Table 7.

From Table 7 we know that by using any of Algorithms 1, 2 or 3, the color intensities of the eight reconstructed colors are indeed different from each other within the corresponding reconstructed image. To ease the comparison among the reconstructed colors in a visual sense, Table 8 illustrates the colors of areas $\boldsymbol{R}_1[\boldsymbol{P}(\boldsymbol{p})]$, $\boldsymbol{R}_2[\boldsymbol{P}(\boldsymbol{p})]$ and $\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{p})]=\boldsymbol{R}_1[\boldsymbol{P}(\boldsymbol{p})]\otimes\boldsymbol{R}_2[\boldsymbol{P}(\boldsymbol{p})]$ with respect to area $\boldsymbol{P}(\boldsymbol{p})$ in $\boldsymbol{P}$ where $\boldsymbol{p}\in\mathbf{C}$ by applying Algorithms 1, 2 and 3, respectively, in Algorithm 6.

Fig. 6. Color decomposition and halftoning of color image P: (a) P; (b) $P^{\mathbf{c}}$, (c) $P^{\mathbf{m}}$, (d) $P^{\mathbf{y}}$; (e) $\boldsymbol{P}^{\mathbf{c}}$, (f) $\boldsymbol{P}^{\mathbf{m}}$, (g) $\boldsymbol{P}^{\mathbf{y}}$.

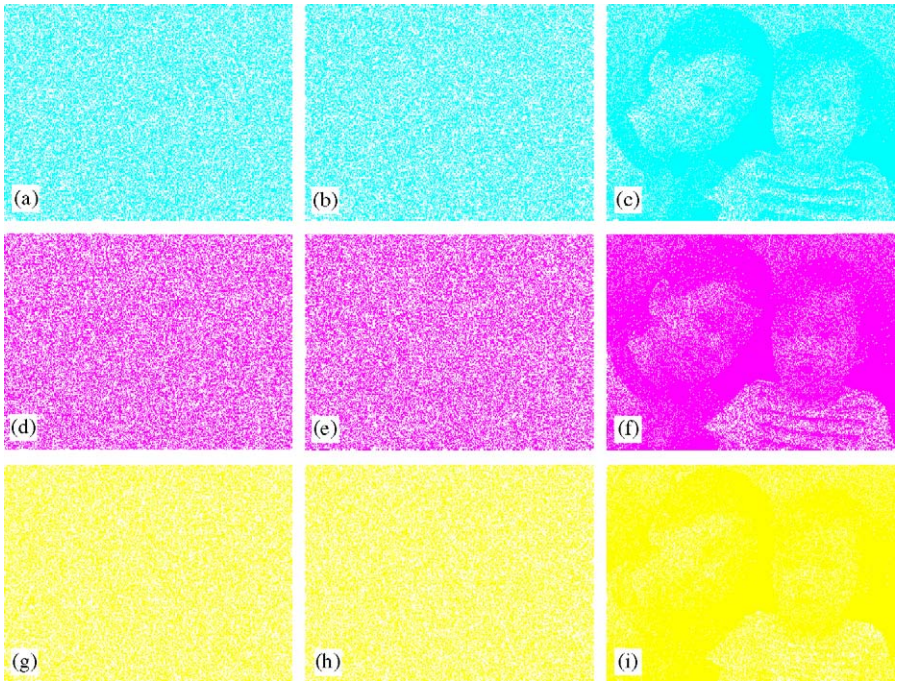

Fig. 7. Implementation results of Algorithm 5 on $\boldsymbol{P}^{\mathbf{c}}$, $\boldsymbol{P}^{\mathbf{m}}$ and $\boldsymbol{P}^{\mathbf{y}}$: (a) $\boldsymbol{R}_1^{\mathbf{c}}$, (b) $\boldsymbol{R}_2^{\mathbf{c}}$, (c) $\boldsymbol{S}^{\mathbf{c}} = \boldsymbol{R}_1^{\mathbf{c}} \otimes \boldsymbol{R}_2^{\mathbf{c}}$; (d) $\boldsymbol{R}_1^{\mathbf{m}}$, (e) $\boldsymbol{R}_2^{\mathbf{m}}$, (f) $\boldsymbol{S}^{\mathbf{m}} = \boldsymbol{R}_1^{\mathbf{m}} \otimes \boldsymbol{R}_2^{\mathbf{m}}$; (g) $\boldsymbol{R}_1^{\mathbf{y}}$, (h) $\boldsymbol{R}_2^{\mathbf{y}}$, (i) $\boldsymbol{S}^{\mathbf{y}} = \boldsymbol{R}_1^{\mathbf{y}} \otimes \boldsymbol{R}_2^{\mathbf{y}}$.

It can be easily observed from Table 8 that (1) for each of the three algorithms, the eight reconstructed colors are truly distinct and visually recognizable; (2) for a certain color, the three reconstructed colors according to the three algorithms may be different; (3) all of the encoded shares are color random grids. Actually, Figs. 8–10 and Table 8 provide the

Fig. 8. Results of the implementation of Algorithm 6: (a) $\boldsymbol{R}_1 = \mathscr{M}(\boldsymbol{R}_1^{\mathbf{c}}, \boldsymbol{R}_1^{\mathbf{m}}, \boldsymbol{R}_1^{\mathbf{y}})$, (b) $\boldsymbol{R}_2 = \mathscr{M}(\boldsymbol{R}_2^{\mathbf{c}}, \boldsymbol{R}_2^{\mathbf{m}}, \boldsymbol{R}_2^{\mathbf{y}})$, (c) $\boldsymbol{S} = \boldsymbol{R}_1 \otimes \boldsymbol{R}_2$.

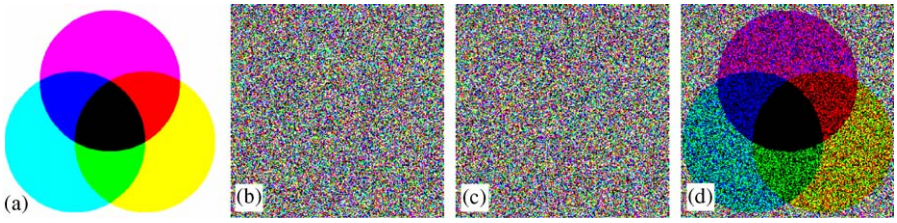

Fig. 9. Results of testing a color image with the exact eight colors in $\mathbf{C}$ by Algorithm 6: (a) eight-colored image P; (b) $\boldsymbol{R}_1$, (c) $\boldsymbol{R}_2$, (d) $\boldsymbol{S} = \boldsymbol{R}_1 \otimes \boldsymbol{R}_2$.

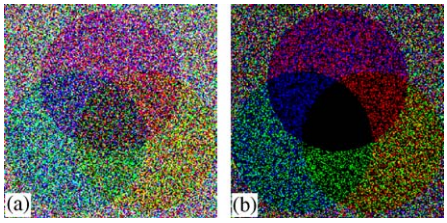

Fig. 10. Results of the reconstructed images of Algorithm 6 when *Encryption_color* is implemented by using ideas in (a) Algorithm 2; (b) Algorithm 3.

Table 7
Results of $\dot{\iota}(\boldsymbol{s})$ with respect to $\boldsymbol{p}$ by Algorithms 6 with various implementations of *Encryption_color*

$\boldsymbol{p}$	$\dot{\iota}(\boldsymbol{s})$		
	Algorithm 1	Algorithm 2	Algorithm 3
o	$(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$	$(\frac{3}{4}, \frac{3}{4}, \frac{3}{4})$
c	$(1, \frac{1}{2}, \frac{1}{2})$	$(\frac{3}{4}, \frac{1}{2}, \frac{1}{2})$	$(1, \frac{3}{4}, \frac{3}{4})$
m	$(\frac{1}{2}, 1, \frac{1}{2})$	$(\frac{1}{2}, \frac{3}{4}, \frac{1}{2})$	$(\frac{3}{4}, 1, \frac{3}{4})$
y	$(\frac{1}{2}, \frac{1}{2}, 1)$	$(\frac{1}{2}, \frac{1}{2}, \frac{3}{4})$	$(\frac{3}{4}, \frac{3}{4}, 1)$
r	$(\frac{1}{2}, 1, 1)$	$(\frac{1}{2}, \frac{3}{4}, \frac{3}{4})$	$(\frac{3}{4}, 1, 1)$
g	$(1, \frac{1}{2}, 1)$	$(\frac{3}{4}, \frac{1}{2}, \frac{3}{4})$	$(1, \frac{3}{4}, 1)$
b	$(1, 1, \frac{1}{2})$	$(\frac{3}{4}, \frac{3}{4}, \frac{1}{2})$	$(1, 1, \frac{3}{4})$
l	$(1, 1, 1)$	$(\frac{3}{4}, \frac{3}{4}, \frac{3}{4})$	$(1, 1, 1)$

Table 8
Colors of $\boldsymbol{R}_1[\boldsymbol{P}(\boldsymbol{p})], \boldsymbol{R}_2[\boldsymbol{P}(\boldsymbol{p})]$ and $\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{p})]$ with respect to $\boldsymbol{P}(\boldsymbol{p})$ by Algorithms 6 with various implementations of *Encryption_color*

$\boldsymbol{p}$	$\boldsymbol{P}(\boldsymbol{p})$	$\boldsymbol{R}_1[\boldsymbol{P}(\boldsymbol{p})]$ Algorithm			$\boldsymbol{R}_2[\boldsymbol{P}(\boldsymbol{p})]$ Algorithm			$\boldsymbol{S}[\boldsymbol{P}(\boldsymbol{p})]$ Algorithm		
		1	2	3	1	2	3	1	2	3
o										
c										
m										
y										
r										
g										
b										
l										

visualized evidences to the correctness of Algorithm 6 (proved in Theorem 4).

We give two more sets of experimental results in Figs. 11 and 12 to further validate the feasibilities and capabilities of our algorithms in color image encryption.

In order to realize the relative performance of applying Algorithms 1, 2 and 3, respectively, in Algorithm 6 for dealing with color image encryption, we adopt the criterion, namely the *color recovery ratio*, to measure the degree of similarity between the original image and the reconstructed result.

Fig. 11. Further results of Algorithm 6 where *Encryption_color* is implemented by Algorithms 1, 2 or 3: (a) secret image P_a, (b) halftone image $\boldsymbol{P}_a$; (c), (d) and (e) $\boldsymbol{R}^1_{a1}, \boldsymbol{R}^1_{a2}$ and $\boldsymbol{S}^1_a = \boldsymbol{R}^1_{a1} \otimes \boldsymbol{R}^1_{a2}$ by Algorithm 1; (f), (g) and (h) $\boldsymbol{R}^2_{a1}, \boldsymbol{R}^2_{a2}$ and $\boldsymbol{S}^2_a = \boldsymbol{R}^2_{a1} \otimes \boldsymbol{R}^2_{a2}$ by Algorithm 2; (i), (j) and (k) $\boldsymbol{R}^3_{a1}, \boldsymbol{R}^3_{a2}$ and $\boldsymbol{S}^3_a = \boldsymbol{R}^3_{a1} \otimes \boldsymbol{R}^3_{a2}$ by Algorithm 3.

Let N be the number of the total pixels in $\boldsymbol{P}$ (the halftone image of secret image P). Let n denote the number of the pixels in $\boldsymbol{S}$ which have the same colors as their corresponding pixels in $\boldsymbol{P}$ (or the number of the colors pixels that are exactly recovered) where $\boldsymbol{S}$ is reconstructed by Algorithm A. The color recovery ratio $r_A(\boldsymbol{P}, \boldsymbol{S})$ for P and $\boldsymbol{S}$ by Algorithm A is computed as

$$r_A(\boldsymbol{P}, \boldsymbol{S}) = \frac{n}{N}.$$

It describes the ability of Algorithm A for recovering color pixels of $\boldsymbol{P}$ in $\boldsymbol{S}$. Intuitively we prefer an algorithm with a higher color recovery ratio.

Table 9 summarizes the color recovery ratios of the reconstructed results in Figs. 11 and 12 where the test images are labeled as P_a and P_b, respectively. It can be seen from Table 9 that in terms of the color recovery ratios Algorithm 1 achieves the highest values and Algorithm 2 gets the lowest. We may say that Algorithm 1 gives the best performance, while Algorithm 2 the worst among the three approaches for color image encryption. By observing Figs. 11 and 12, our visual system discloses the same outcome.

5. Concluding remarks

We propose effective random grid-based algorithms for the encryption of the gray-level and color images in this paper. The most attractive advantage of our idea is that the decryption process is done by the human visual system and that no computational device is needed. When compared with the approaches in visual cryptography, the most significant contribution in our encryption algorithms is that there is no need for the basis matrices as well as no extra pixel expansion. Our algorithms are simple and can be easily implemented. The correctness of our algorithms was also proved in a formal way. The experimental results demonstrate the capabilities of our approaches. In binary image encryption, we define the contrast in terms of light transmission to evaluate the relative difference between transparent and opaque pixels in the reconstructed image. With regard to color image encryption, we adopt the color recovery ratio to measure the degree of similarity between the original and reconstructed images.

The techniques of color decomposition and halftone are well developed in the areas of computer graphics and image processing. We have many choices to exploit these techniques in our algorithm. The contrasts achieved by Algorithms 1, 2 and 3 are different so that it might be an interesting idea to adopt them in a hybrid way according to the content of the secret images or the scenario of the applications. Fig. 13 shows some simple hybridization results of using Algorithms 1, 2 or 3. We may say that our algorithms can be adapted in a flexible way to accommodate to a wide range of practical applications.

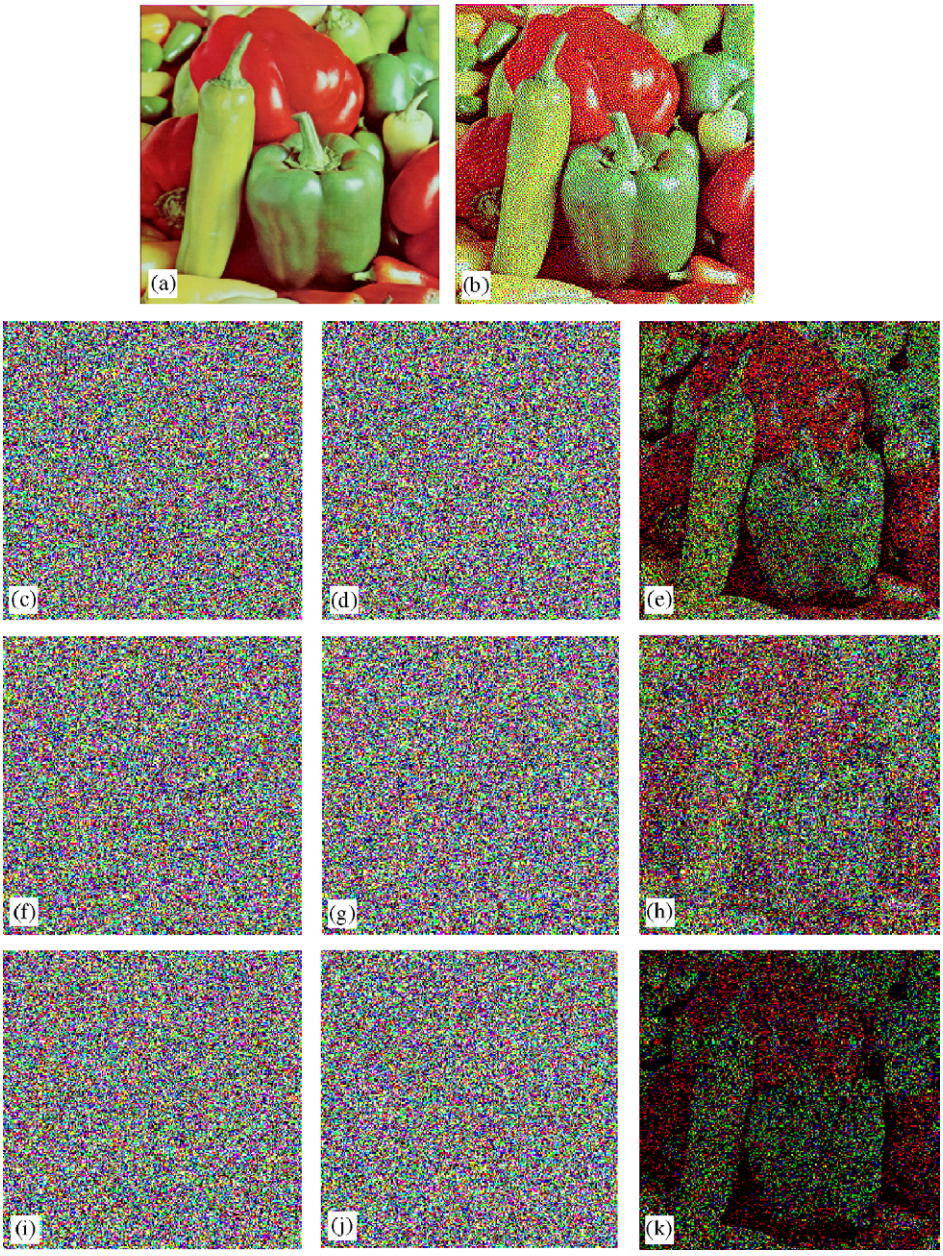

Fig. 12. More results of Algorithm 6 where *Encryption_color* is implemented by Algorithms 1, 2 or 3: (a) secret image P_b, (b) halftone image $\boldsymbol{P}_b$; (c), (d) and (e) $\boldsymbol{R}^1_{b1}$, R^1_{b2} and $\boldsymbol{S}^1_b = \boldsymbol{R}^1_{b1} \otimes \boldsymbol{R}^1_{b2}$ by Algorithm 1; (f), (g) and (h) $\boldsymbol{R}^2_{b1}, \boldsymbol{R}^2_{b2}$ and $\boldsymbol{S}^2_b = \boldsymbol{R}^2_{b1} \otimes \boldsymbol{R}^2_{b2}$ by Algorithm 2; (i), (j) and (k) $\boldsymbol{R}^3_{b1}, \boldsymbol{R}^3_{b2}$ and $\boldsymbol{S}^3_b = \boldsymbol{R}^3_{b1} \otimes \boldsymbol{R}^3_{b2}$ by Algorithm 3.

Table 9
Results of $r_A(\boldsymbol{P}, \boldsymbol{S})$ for test images P_a and P_b by Algorithms 1, 2 and 3

Test image	$r_A(\boldsymbol{P}, \boldsymbol{S})$		
	Algorithm 1	Algorithm 2	Algorithm 3
P_a	0.715	0.337	0.617
P_b	0.489	0.263	0.322

The introduction of using random grids for image encryption also uncovers some related research topics, such as the design of other random grid-based algorithms, the precise definition and analysis to the relative differences among color intensities in a visual sense (it is noticed that the proposed evaluation metric, i.e., color recovery ratio, does not involve the color intensities of reconstructed pixels so that it does not interpret the visual effect of colors in a reconstructed image), the applications in the topics of visual identification or authentication, to solve the general k-out-of-n threshold secret sharing, just to name a few. They are interesting and worthy of further study.

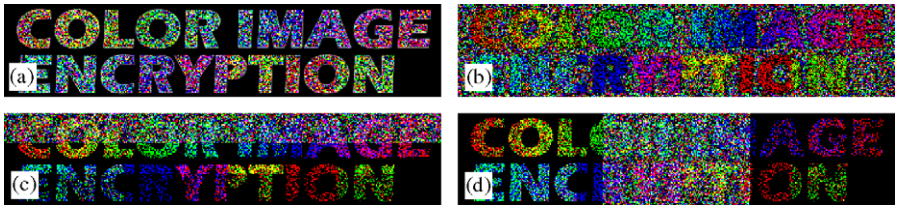

Fig. 13. Some hybridization results of using Algorithms 1, 2 or 3 for $\boldsymbol{P}_a$: (a) encoding black color by Algorithm 1 while the others by Algorithm 2; (b) encoding black color by Algorithm 2 while the others by Algorithm 1; (c) encoding the three evenly and vertically decomposed areas of $\boldsymbol{P}_a$ by Algorithms 2, 1 and 3, respectively; (d) encoding the three evenly and horizontally decomposed areas of $\boldsymbol{P}_a$ by Algorithms 1, 2 and 3, respectively.

Acknowledgements

The author would like to express his earnest appreciation to the anonymous referees for their constructive suggestions. This research was partly supported by National Science Council of the Republic of China under contract NSC93-2213-E-130-008.

References

[1] O. Kafri, E. Keren, Encryption of pictures and shapes by random grids, Opt. Lett. 12 (1987) 377–379.
[2] M. Naor, A. Shamir, Visual cryptography, in: A. De Santis (Ed.), Advances in Cryptology: Eurpocrypt'94, Lecture Notes in Computer Science, vol. 950, 1995, pp. 1–12.
[3] G. Ateniese, C. Blundo, A. De Santis, D.R. Stinson, Visual cryptography for general access structures, Inf. Comput. 129 (1996) 86–106.
[4] S. Droste, New results on visual cryptography, in: N. Koblitz (Ed.), Advances in Cryptology: CRYPTO'96, Lecture Notes in Computer Science, vol. 1109, 1996, pp. 401–415.
[5] M. Naor, B. Pinkas, Visual authentication and identification, in: B.S. Kaliski Jr. (Ed.), Advances in Cryptology: CRYPTO'97, Lecture Notes in Computer Science, vol. 1294, 1997, pp. 322–336.
[6] E.R. Verheul, H.C.A. Van Tilborg, Constructions and properties of k out of n visual secret sharing schemes, Designs Codes Cryptography 11 (1997) 179–196.
[7] G. Ateniese, C. Blundo, A. De Santis, D.R. Stinson, Extended capabilities for visual cryptography, Theoret. Comput. Sci. 250 (2001) 143–161.
[8] Y.-C. Hou, Visual cryptography for color images, Pattern Recognition 36 (2003) 1619–1629.
[9] S.J. Shyu, Efficient visual secret sharing scheme for color images, Pattern Recognition 39 (2006) 866–880.
[10] J.F. Jarvis, C.N. Judice, W.H. Ninke, A survey of techniques for the display of continuous tone pictures on bilevel displays, Comput. Graphics Image Process. 5 (1976) 13–40.
[11] R.W. Floyd, L. Steinberg, An adaptive algorithm for spatial grey scale, Proc. Soc. Inf. Display 17 (1976) 75–77.
[12] R. Ulichney, Digital Halftoning, The MIT Press, Cambridge, 1987.
[13] J. Sullivan, R. Miller, G. Pios, Image halftoning using a visual model in error diffusion, J. Opt. Soc. Am. A 10 (1993) 1714–1724.
[14] R. Ulichney, A review of halftoning techniques, in: R. Eschbach, G.G. Marcu (Eds.), Proceedings of SPIE: Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts V, vol. 3963, 2000, pp. 378–391.
[15] K.T. Knox, Error diffusion: a theoretical view, in: J.P. Allebach, B.E. Rogowitz (Eds.), Proceedings of SPIE: Human Vision, Visual Processing, and Digital Display IV, vol. 1913, 1993, pp. 326–331.
[16] J.J. Itten, The Elements of Color, Van Nostrand Reinhold, New York, 1970.
[17] R. Jackson, L. MacDonald, K. Freeman, Computer Generated Colour: A Practical Guide to Presentation and Display, Wiley, New York, 1993.
[18] C.A. Poynton, Frequently asked questions about color, ⟨http://www.poynton.com/ColorFAQ.html⟩, 2000.

About the Author—SHYONG JIAN SHYU received his B.S. degree in Computer Engineering from the National Chiao Tung University in 1985, the M.S. degree in Computer and Decision Sciences in 1987 and the Ph.D. degree in Computer Science in 1991 from the National Tsing Hua University, Taiwan, ROC. From 1993 to 1994, he worked as a researcher at Academia Sinica Computer Centre, Taiwan. Currently, he is a professor of the Department of Computer Science and Information Engineering at Ming Chuan University, Taiwan. His research interests include the design and analysis of algorithms, parallel computing, visual cryptography and computational biology.