SUMMARY (k,n)-visual secret sharing scheme ((k,n)-VSSS) is a method to divide a secret image into n images called shares that enable us to restore the original image by only stacking at least k of them without any complicated computations. In this paper, we consider (2,2)-VSSS to share two secret images at the same time only by two shares, and investigate the methods to improve the quality of decoded images. More precisely, we consider (2,2)-VSSS in which the first secret image is decoded by stacking those two shares in the usual way, while the second one is done by stacking those two shares in the way that one of them is used reversibly. Since the shares must have some subpixels that inconsistently correspond to pixels of the secret images, the decoded pixels do not agree with the corresponding pixels of the secret images, which causes serious degradation of the quality of decoded images. To reduce such degradation, we propose several methods to construct shares that utilize 8-neighbor Laplacian filter and halftoning. Then we show that the proposed methods can effectively improve the quality of decoded images. Moreover, we demonstrate that the proposed methods can be naturally extended to (2,2)-VSSS for RGB images.

key words: secret sharing, visual cryptography, threshold visual secret sharing schemes, halftone image, Laplacian filter

1. Introduction

(k,n)-threshold visual secret sharing scheme ((k,n)-VSSS), first proposed by Naor and Shamir[7], is a secret sharing scheme [9] to share a digital secret image among n parties under the condition that the secret image can be decoded only when k or more parties among n parties provide their information. The secret image is encoded into n images called shares, and each share is printed on a transparency. This enables us to decode the secret image not by complex computations but by just stacking shares.

In this paper, we consider VSSS in which a specific way of stacking shares is imposed. More concretely, we consider (2,2)-VSSS that encodes two secret images at the same time into two shares and both of secret images are decoded by these two shares. Such (2,2)-VSSS can be realized by constructing reversible shares, that is, the first secret image is decoded by stacking those two shares in the usual way, while the second one is done by stacking those two shares in the way that one of them is used reversibly.

Because of the reversible-use of shares, subpixels in the two shares at the identical positions and other subpixels in those shares at the symmetric positions are mutually dependent. Hence it is sometimes unavoidable to contradictorily determine the subpixels of shares. If such subpixels correspond to pixels that are located on or near the edges of the contents in the secret images, the decoded images will become obscure. On the other hand, if such subpixels correspond to pixels that are located in flat areas of the contents in the secret images, the decoded images will contain pixels with opposite color in those areas like noise. Anyway, the quality of the decoded image is seriously degraded.

It can be easily imagine that we are able to understand what is drawn or reflected in the decoded image even if some pixels in it incorrectly reproduce the corresponding pixels in the secret image. This is because the details of the decoded image of VSSS are not so important to understand the contents of the secret image. Hence in our VSSS, we allow some pixels in the secret image to be incorrectly decoded, while the whole of secret image is decoded so that the contents are well recognized. For this purpose, we devise in the proposed methods how to choose pixels that are allowed to be incorrectly decoded. In addition, the incorrectly decoded pixels are easily noticed if they are located in a large area that is constructed by solid-color pixels. To make the incorrectly decoded pixels relatively unnoticed, we propose to apply halftoning[6][12] to the given images before constructing shares of VSSS. Moreover, we also extend the proposed method to that of RGB images. The examples provided in this paper exhibit the effectiveness of the proposed methods.

It should be noted that a considerable number of VSSS that can treat more than one secret images in the same time have been studied[1][4][5][10]. Especially, some of those literatures[4][5] proposed VSSS with a particular way of stacking shares. The difference between the conventional and our studies is that in the conventional study, it is allowed for none of pixels in the secret images to be incorrectly decoded, while it is allowed for some of them in our method. In other words, the conventional method requires pixels in the secret images to strictly satisfy various conditions, while the proposed one does not at all. As the result of these differences, the class of secret images for which the conventional and the proposed methods are suitable is also different.

This paper is organized as follows. In Sect. 2, we formally express the conditions required in the construction of shares of (k,n)-VSSS, and give a procedure to construct...
shares of (2, 2)-VSSS. In Sect. 3, we investigate how to design shares of (2, 2)-VSSS with reversibly usable shares, and show some examples. In Sect. 4, we extend the methods proposed in Sect. 3 for RGB images and binary geometric images. Conclusions and future works are summarized in Sect. 5.

2. (k, n)-VSSS

2.1 Definition of VSSS

We denote by \( \mathbb{F}_2 := \{0, 1\} \) and \( \mathbb{N} \) the finite field of two elements and the set of all natural numbers, respectively.

We denote by \( X = (x_{i,j}) \in \mathbb{F}_2^{M \times N} \) \((M, N \in \mathbb{N})\) a binary secret image consisting of \( M \times N \) pixels \( x_{i,j} \). In \((k, n)\)-VSSS, \(X\) is encoded into \(n\) shares \( S^{(i)} = (s_{i,j}^{(i)}(x_{i,j})) \) \((1 \leq i \leq N, 1 \leq j \leq M, \ell = 1, 2, \ldots, n)\) where \( s_{i,j}^{(i)}(x_{i,j}) \in \mathbb{F}_2^{2 \times p} \) \((p \in \mathbb{N})\) is the subpixel corresponding to \(x_{i,j}\). For an arbitrary \(A \subset \{1, 2, \ldots, n\}\), a decoded image \(D = (d_{i,j}^{(A)})\) is determined as \(d_{i,j}^{(A)} := \vee_{\ell \in A} s_{i,j}^{(\ell)}(x_{i,j})\) where \(\vee\) denotes OR operation.

Let \(H(V)\) denote the Hamming weight of a binary vector \(V\). The definition of a \((k, n)\)-VSSS proposed by Naor and Shamir [4] with parameters \(k, n, m, d, \alpha\) is as follows where \(2 \leq k \leq n, 1 \leq d \leq m, 0 < \alpha < 1\).

**Definition:** A solution to the \((k, n)\)-VSSS is obtained by restricting each \(n \times m\) matrix \(C_0\) and \(C_1\). To share a black pixel, the dealer randomly chooses one of the matrices in \(C_0\), and to share a white pixel, the dealer randomly chooses one of the matrices in \(C_1\). The chosen matrix defines the color of the \(m\) sub-pixels in each one of the \(n\) transparencies. The solution is considered valid if the following three conditions are met:

1) For any \(S\) in \(C_0\), the “or” \(V\) of any \(k\) of the \(n\) rows satisfies \(H(V) \leq d - \alpha \cdot m\).
2) For any \(S\) in \(C_1\), the “or” \(V\) of any \(k\) of the \(n\) rows satisfies \(H(V) \geq d\).
3) For any subset \(\{i_1, i_2, \ldots, i_q\}\) of \(\{1, 2, \ldots, n\}\) with \(q < k\), the two collections of \(q \times m\) matrices \(D_t\) for \(t \in \{0, 1\}\) obtained by restricting each \(n \times m\) matrix \(C_t\) (where \(t = 0, 1\)) to rows \(i_1, i_2, \ldots, i_q\) are indistinguishable in the sense that they contain the same matrices with the same frequencies.

The first two are the contrast conditions that ensure the existence of a difference \(\alpha \cdot m\) between the Hamming weights in the results of OR operation of any \(k\) of the \(n\) rows in the matrices chosen from \(C_0\) and \(C_1\), so that our eyes are able to identify the color (white or black) of pixels. The third is the security condition which insists that \(D_0\) and \(D_1\) are indistinguishable in the sense that they contain the same matrices with the same frequencies to guarantee information-theoretic security on the color of pixels.

2.2 Realization of (2, 2)-VSSS

In this subsection, we give a concrete procedure to realize one of (2, 2)-VSSS. In the remaining part of this paper, we employ or refer this construction as a part of the constructions that are designed to construct the special kinds of shares of (2, 2)-VSSS.

**Construction 1:** For a secret image \(X = (x_{i,j})\), execute the following steps for all \((i, j)\).

Step 1 Choose the subpixel \(s_{i,j}^{(1)}(x_{i,j})\) of the share \(S^{(1)}\) randomly from the following six \(2 \times 2\) matrices:

\[
\begin{align*}
S_1 &= \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, & S_2 &= \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, & S_3 &= \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}, \\
S_4 &= \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, & S_5 &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, & S_6 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.
\end{align*}
\]

Step 2 Determine the subpixel \(s_{i,j}^{(2)}(x_{i,j})\) of the share \(S^{(2)}\) by

\[
s_{i,j}^{(2)}(x_{i,j}) := x_{i,j} \cdot 1 + s_{i,j}^{(1)}(x_{i,j}),
\]

where \(\cdot\) and + respectively denote the multiplication and the addition over \(\mathbb{F}_2\), and 1 denotes the all-1 \(2 \times 2\) matrix.

In the following, we explain how Construction 1 satisfies the conditions of Definition given in the previous subsection.

First of all, since we consider (2, 2)-VSSS, \(k = n = 2\). In addition, we consider \(2 \times 2\) subpixels in Construction 1, \(m\) is equal to 4. The collections \(C_t\) and \(D_t\) \((t = 0, 1)\) and remaining parameters \(d\) and \(\alpha\) for Construction 1 is determined in the following way.

According to the 6 matrices considered in Step 1 of Construction 1 and Eq. (1), \(C_0\) (resp. \(C_1\)) is a collection of all matrices obtained by taking a column permutation of the matrix \(B_0\) (resp. \(B_1\)) defined as

\[
B_0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.
\]

For every \(S \in C_0\) (resp. \(S \in C_1\)), we can easily verify from the definition of \(B_0\) (resp. \(B_1\)) that the Hamming weight of the “or” \(V\) of 2 rows in \(S\) is equal to 2 (resp. 4). This means that Construction 1 satisfies the conditions 1) and 2) of Definition with \(d = 4\) and \(\alpha = 0.5\).

Finally, \(D_0\) (resp. \(D_1\)) is a collection of matrices obtained by taking every column permutation of the matrix \([1100]\) (resp. \([0101]\), \([0011]\)). Therefore the elements of \(D_t\) \((t \in \{0, 1\}\) are enumerated as

\[
\begin{align*}
D_0 &= \{[1100], [1010], [1001], [0110], [0101], [0011]\}, \\
D_1 &= \{[1100], [1010], [1001], [0110], [0101], [0011]\}
\end{align*}
\]

and we conclude that \(D_0\) and \(D_1\) contain the same matrices with the identical frequencies. Hence it can be verified that Construction 1 satisfies the conditions of Definition.
3. VSSS with the Symmetric Constraint on Subpixels of Shares

3.1 Condition on Which the Symmetric Shares of (2, 2)-VSSS Can Be Constructed

In this subsection, we will clarify the necessary and sufficient condition on which the shares of (2, 2)-VSSS satisfying the symmetric condition can be realized.

We denote by $A' = (a_{i,j}')$ the $M \times N$ image $A$ flipped horizontally, that is, $a_{i,j}' = a_{i,N+1-j}$ for all $1 \leq i \leq M$ and $1 \leq j \leq \lceil \frac{N}{2} \rceil$. Let $X = (x_{i,j})$ and $Y = (y_{i,j}) \in \mathbb{F}_2^{M \times N}$ ($N = 2q$, $q \in \mathbb{N}$) be two secret images of the same size and we would construct shares of (2, 2)-VSSS $S^{(t)} = (s^{(t)}_{i,j}(x_{i,j}))$ (resp. $T^{(t)} = (t^{(t)}_{i,j}(y_{i,j}))$) ($t = 1, 2$) for $X$ (resp. $Y$) with the additional conditions expressed as

$$T^{(1)} = S^{(1)}$$
$$T^{(2)} = (S^{(2)})' \tag{2}$$

In other words, $X$ and $Y$ are encoded into not four but only two shares that are used in two different manners, that is, stacking $S^{(1)}$ and $S^{(2)}$ to decode $X$, and stacking $T^{(1)} = S^{(1)}$ and $T^{(2)} = (S^{(2)})'$ to decode $Y$.

Fix a position $(i, j)$. Then as we have seen in Step 2 of Construction 1, the subpixels $s_{i,j}^{(1)}(x_{i,j})$ and $s_{i,j}^{(2)}(x_{i,j})$ have to satisfy

$$s_{i,j}^{(1)}(x_{i,j}) + s_{i,j}^{(2)}(x_{i,j}) = x_{i,j} \cdot 1 \tag{3}$$

in order for $x_{i,j}$ to be correctly decoded. In addition, when the conditions given in Eq. (2) are embedded in the shares, we see from the same reason for $x_{i,j}$ that the subpixels $t_{i,j}^{(1)}(y_{i,j}) = s_{i,j}^{(1)}(x_{i,j})$ and $t_{i,j}^{(2)}(y_{i,j}) = (s_{i,j}^{(2)}(x_{i,N+1-j}))'$ have to satisfy

$$s_{i,j}^{(1)}(x_{i,j}) + s_{i,j}^{(2)}(x_{i,N+1-j})' = y_{i,j} \cdot 1 \tag{4}$$

in order for $y_{i,j}$ to be correctly decoded, and the subpixels $t_{i,j}^{(1)}(y_{i,N+1-j}) = s_{i,j}^{(1)}(x_{i,N+1-j})$ and $t_{i,j}^{(2)}(y_{i,N+1-j}) = (s_{i,j}^{(2)}(x_{i,j}))'$ have to satisfy

$$s_{i,j}^{(1)}(x_{i,N+1-j}) + s_{i,j}^{(2)}(x_{i,N+1-j})' = y_{i,N+1-j} \cdot 1 \tag{5}$$

in order for $y_{i,N+1-j}$ to be correctly decoded, respectively. Moreover, in order for $x_{i,N+1-j}$ in Eqs. (4) and (5) to be also correctly decoded, it must hold that

$$s_{i,j}^{(1)}(x_{i,N+1-j}) + s_{i,j}^{(2)}(x_{i,N+1-j}) = x_{i,N+1-j} \cdot 1 \tag{6}$$

Hence the secret images $X$ and $Y$ are decoded correctly if and only if the subpixels of $S^{(1)}$ and $S^{(2)}$ are all assigned so that Eqs. (3), (4), (5) and (6) hold at the same time for all $(i, j)$.

According to the argument given above, we propose the following procedure that intends to construct shares satisfying Eq. (2) for $X$ and $Y$.

Construction 2: For secret images $X = (x_{i,j})$ and $Y = (y_{i,j})$, execute the following steps for $(i, j)$ with $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, \lfloor \frac{N}{2} \rfloor$.

Step 1 Execute Step 1 of Construction 1.
Step 2 Execute Step 2 of Construction 1.
Step 3 Determine the subpixel $s_{i,N+1-j}^{(2)}(x_{i,N+1-j})$ of the share $S^{(2)}$ by

$$s_{i,N+1-j}^{(2)}(x_{i,N+1-j}) := y_{i,j} \cdot 1 + (s_{i,j}^{(2)}(x_{i,j}))' \tag{7}$$

Step 4 Determine the subpixel $s_{i,N+1-j}^{(1)}(x_{i,N+1-j})$ of the share $S^{(1)}$ by

$$s_{i,N+1-j}^{(1)}(x_{i,N+1-j}) := y_{i,N+1-j} \cdot 1 + (s_{i,j}^{(2)}(x_{i,j}))' \tag{8}$$

Note that the expressions employed in Step 3 and Step 4 of Construction 2 are respectively obtained from Eq. (4) and Eq. (5). Moreover, we use the relations $(A')' = A$ and $(x \cdot 1)' = x$ in the derivation of the equation in Step 3.

In Construction 2, $s_{i,j}^{(1)}(x_{i,j})$ among four subpixels is first assigned for $(i, j)$. However, it can be easily verified that there is no problem if we start the assignment from any of other three subpixels. Moreover, the subpixel among four subpixels that is first assigned can be changed for each $(i, j)$.

In the following argument, we will show that there may occur a problem about the assignment of subpixels if we simply apply Construction 2 to construct shares $S^{(1)}$ and $S^{(2)}$ satisfying Eq. (2) for given $X$ and $Y$.

Assume that $s_{i,j}^{(1)}(x_{i,j})$ and $s_{i,j}^{(2)}(x_{i,j})$ are determined in Step 1 and Step 2, respectively. Then $s_{i,N+1-j}^{(1)}(x_{i,N+1-j})$ and $s_{i,N+1-j}^{(1)}(x_{i,N+1-j})$ are successively determined by Step 3 and Step 4, respectively. Then, by adding both sides of Eqs. (3), (4) and (5) respectively, it can be shown that $s_{i,N+1-j}^{(2)}(x_{i,N+1-j})$ ($\ell = 1, 2$) must satisfy

$$s_{i,N+1-j}^{(1)}(x_{i,N+1-j}) + s_{i,N+1-j}^{(2)}(x_{i,N+1-j}) = (x_{i,j} + y_{i,j} + y_{i,N+1-j}) \cdot 1 \tag{9}$$

Then it is obvious that Eqs. (6) and (7) hold at the same time if and only if it holds that

$$x_{i,j} + y_{i,j} + y_{i,N+1-j} + y_{i,N+1-j} \equiv 0 \mod 2 \tag{10}$$

Then from the observation given above, we find that Eq. (8) is the necessary and sufficient condition on which the subpixels $s_{i,j}^{(1)}(x_{i,j})$ and $s_{i,N+1-j}^{(1)}(x_{i,N+1-j})$ ($\ell = 1, 2$) can be assigned so that all elements of $P_{i,j} := \{x_{i,j}, y_{i,j}, x_{i,N+1-j}, y_{i,N+1-j}\}$ are decoded correctly, which is summarized in the following proposition.

**Proposition 1:** The shares $S^{(t)}$ ($\ell = 1, 2$) of (2, 2)-VSSS such that the secret image $X$ (resp. $Y$) is correctly decoded from $S^{(1)}$ and $S^{(2)}$ (resp. $S^{(1)}$ and $S^{(2)}$) can be constructed by Construction 2 if and only if all elements of $P_{i,j}$ satisfy Eq. (8) for all $(i, j)$.

**Example 1:** Assume that the secret images $X = (x_{i,j})$ and $Y = (y_{i,j})$ are symmetric, that is, $x_{i,j} = x_{i,N+1-j}$ and $y_{i,j} = y_{i,N+1-j}$.
Figure 1(c) shows the decoded images that are obtained from the shares \( S^{(\ell)} (\ell = 1, 2) \) for \( X_1 \) and \( Y_1 \), where \( S^{(\ell)} \) and \( T^{(\ell)} \) are individually constructed by Construction 2. By comparing Figs. 1(c) and (d), we can see that the secret images are correctly decoded in Fig. 1(d), though the symmetric condition on the shares is embedded.

### 3.2 Degradation of the Quality of Decoded Images by Allowing Incorrectly Decoded Pixels

Assume that Eq. (8) does not hold for some \((i, j)\) in the secret images \( X \) and \( Y \). Then Proposition 1 states that it is necessary to change at least one of four pixels in \( P_{i,j} \), denoted by \( z \), into \( \overline{z} := z + 1 \) to obtain appropriate subpixels of shares at the positions \((i, j)\) and \((i, N + 1 - j)\). But this means that the estimated pixel \( \hat{z} \) in the decoded image corresponding to \( z \) becomes \( \hat{z} = \overline{z} \), which is the incorrect result. Hence if \( X \) and \( Y \) contain many positions \((i, j)\) for which Eq. (8) does not hold, many pixels of \( X \) and \( Y \) must be estimated incorrectly, and therefore, serious degradation of the quality of the decoded image will occur. To reduce such degradation, we propose the following procedure that constructs shares.

**Construction 3:** For secret images \( X = (x_{i,j}) \) and \( Y = (y_{i,j}) \), execute the following steps for all \((i, j)\) with \( i = 1, 2, \ldots, M \) and \( j = 1, 2, \ldots, N/2 \).

- **Step 1** If Eq. (8) does not hold for \((i, j)\), choose one of elements of \( P_{i,j} \), denoted by \( z \), in appropriate way and change it into \( \overline{z} \).

- **Step 2** Execute from Step 1 to Step 4 of Construction 2.

It can be easily understood that the shares constructed by Construction 3 vary dependently on how to choose \( z \) from \( P_{i,j} \) in Step 1. Probably, the most primary way to choose \( z \) is the random selection. So we refer to Construction 3 with the random selection of \( z \) in Step 1 as Construction 3 (rand.).

**Example 2:** If one or both of the secret images are asymmetric, Eq. (8) does not hold for a considerable number of \((i, j)\) and the quality of the decoded images will be much degraded. To demonstrate this, we consider two binary asymmetric images \( X_2 \) and \( Y_2 \) shown in Fig. 2(b) as secret images. These images are respectively generated from RGB images shown in Fig. 2(a) by the similar procedure employed in Example 1.

We first demonstrate the case that no condition on the usage of shares is considered. For this case, the shares \( S^{(\ell)} (\ell = 1, 2) \) for \( X_2 \) and \( T^{(\ell)} (\ell = 1, 2) \) for \( Y_2 \) can be individually constructed by Construction 1. Then as we see from the

---

1. Newsprint filter of GIMP [2] is used to obtain halftone images from gray scale images.
decoded images for $X_2$ and $Y_2$ shown in Fig. 2 (c), the secret images are correctly decoded and the quality of the decoded images is satisfactory.

Figure 2 (d) shows the decoded images for $X_2$ and $Y_2$ obtained from the shares constructed by Construction 3 (rand.). Because of the symmetric condition on the shares, Eq. (8) does not hold for some $(i, j)$ in $X_2$ and $Y_2$. This results in 10.4% of pixels of the two secret images being incorrectly decoded, and the quality of the decoded images gets worse than that of the decoded images shown in Fig. 2 (c). However, since incorrectly decoded pixels are scattered all over both of the secret images, we can also say that the degradation of the quality of the decoded images is rather moderated.

3.3 Selection of Incorrectly Decoded Pixels by 8-Neighbor Laplacian Filter

In order to further reduce the degradation of the quality of the decoded images perceived in Example 2, we devise a method to choose one of $P_{i,j}$ that is allowed to be incorrectly decoded.

We can imagine that if incorrectly decoded pixels are located on or near the edges of the contents appear in the decoded images, the quality of the decoded images becomes worse. To avoid such kind of degradation, one of four pixels in $P_{i,j}$ that is incorrectly decoded is chosen based on how far from the edges of the contents. For this purpose, we employ 8-neighbor Laplacian filter (8-LF). The filter is often used to detect edges in an image by the result of convolution between a digital image $Z = (z_{i,j})$ and the following matrix $K_8$ known as kernel of 8-LF:

\[
K_8 = \begin{bmatrix}
1 & 1 & 1 \\
1 & -8 & 1 \\
1 & 1 & 1
\end{bmatrix}
\]

Let $a_{z_{i,j}} (0 \leq |a_{z_{i,j}}| \leq 8)$ denote the result for $z_{i,j}$ obtained by applying 8-LF. It is known [8] that when $z_{i,j}$ is on an edge, $|a_{z_{i,j}}|$ tends to be a large value. Therefore, it can be efficiently decided by applying 8-LF to the secret images whether the pixel is likely located on an edge or not.

Assume that Eq. (8) does not hold for given $(i, j)$. Once we find the pixel of $P_{i,j}$ that is the least likely located on an edge among pixels of $P_{i,j}$, we allow it to be the incorrectly decoded pixel. Then it can be expected that the degradation of the quality of the decoded image is reduced. Based on the principle described above, we propose another procedure to construct shares, in which the pixel $z$ with the smallest $|a_z|$ is chosen from $P_{i,j}$ in Step 1 of Construction 3. This procedure is named as Construction 3 (8-LF). Note that, if there are two or more pixels having the smallest $|a_z|$, choose randomly one of them.

Example 3: Figure 2 (e) shows the decoded images for $X_2$ and $Y_2$ obtained from the shares constructed by Construction 3 (8-LF). By comparing these decoded images with those of Fig. 2 (d), we can say that the decoded images shown in Fig. 2 (e) exhibit a higher contrast. This results in the sharper edge. For example, the eyes of the tiger are more easily recognized in the right image of Fig. 2 (e) than those in the right image of Fig. 2 (d).

Example 4: We show in Figs. 3 and 4 images that magnify...
the images in Fig. 2. The original image (Fig. 2 (a)) is shown again as Fig. 3 (a) for ease of comparison. Then Fig. 3 (b) magnifies the part of images indicated in Fig. 3 (a) by white square whose size is 50 × 50 (pixels). The images shown in Fig. 3 (c) and all figures in Fig. 4 correspond to this part. We also show in Fig. 5 the shares for the decoded images given in Fig. 4 (c) that are constructed by Construction 3 (8-LF). The left (resp. right) image of Fig. 4 (c) is decoded by S1 and S2 (resp. S1 and (S2)′) shown in Figs. 5 (a) and (b) (resp. Figs. 5 (a) and (c)). The original and secret images (Figs. 3 (b) and (c)) are printed in 25dpi, while the decoded images (Figs. 4 (a), (b) and (c)) and shares (Figs. 5 (a), (b) and (c)) are done in 50dpi to present those images in equal size.

Both images in Fig. 3 (c) correspond to the part indicated by the red frame in Fig. 5 (d). Hence S1 and S2 correspond to that part in S(1) and S(2), respectively. On the other hand, since the right image of Fig. 4 (c) is obtained by stacking one of two shares reversibly, (S2)′ corresponds to the part indicated by the blue frame in Fig. 5 (d) in S(2). As shown in Fig. 5 (d), the red and blue frames are located at the horizontally symmetrical position.

We can verify from images shown in Fig. 4 that the decoded images generated by the proposed method can well reproduce the secret images even if they contain incorrectly decoded pixels. Moreover, by noting the density of black and white pixels, we can say that Construction 3 (8-LF) reproduces the secret images more precisely than Construction 3 (rand) does. □
3.4 Making Incorrectly Decoded Pixels Less Recognizable by Halftoning

In this subsection, we consider the effectiveness of halftoning that is applied to the proposed methods.

Recall that we have applied in Examples 1 to 3 halftoning to generate the binary secret images from the gray scale images. The following example shows the significance of halftoning.

**Example 5:** The images $X_3$ and $Y_3$ shown in Fig. 6 (a) are generated from $O_{X_2}$ and $O_{Y_2}$ shown in Fig. 2 (a) by binarizing them. The decoded images for $X_3$ and $Y_3$ obtained from the shares constructed by Construction 3 (rand.) are shown in Fig. 6(b). By comparing the decoded images shown in Fig. 6(b) with those shown in Fig. 2 (d), we find that the position of incorrectly decoded pixels in Fig. 6(b) are more

---

*Image > Mode > Indexed Command of GIMP [2] converts a image to indexed mode. Black and white (1-bit) palette is used to binarize images.*
3.5 Comparison with a Conventional Method

In this subsection, we compare the proposed method with one of conventional methods presented by Shyu et al. [10]. Since it is allowed in the proposed method that some pixels in the secret images are incorrectly decoded, some conditions concerned in the construction of shares can be relaxed. Then subpixels that represent a white pixel of decoded image obtained by the proposed method can contain more white pixels than that by Shyu’s method. The subpixels that represent a white pixel in the decoded image are compared in Fig. 7.

We compare in Figs. 8 (a) and (b) the decoded images obtained by Construction 3 (8-LF) and and Shyu’s methods. We also show in Figs. 8 (c) and (d) the magnifications of parts indicated by white frame in Figs. 8 (a) and (b). The size of the part is 100 × 100 (pixels) and the magnified images are printed in 50dpi. Because of the difference of the number of white pixels in the subpixels, which is shown in Fig. 7, we verify that the decoded image of the proposed method is brighter than that of Shyu’s method. In the actual implementation by printing shares on transparencies, this characteristic of the proposed method is desirable to recognize what is presented in the decoded image.

3.6 Implementation of the Proposed Method

In the implementation of VSSS, shares are usually printed...
on transparencies. If the shares are printed in small size compared with the number of pixels, the quality of the decoded image is improved. But it is also true that the pixel-positioning operation becomes difficult when we put one share of such small size on top of the other. Hence it is important to decide appropriate print size for a given secret image. For example, if we print shares for Fig. 2 (b) on A4-size transparencies, its number of pixels $800 \times 600$ is too large compared with the dimension of the print.

In order to investigate appropriate printing size of shares, we have attempted the additional experiments, that is, constructing shares by Constructions 3 (rand) and 3 (8-LF) for binary images of $200 \times 150$ pixels shown in Fig. 9 (b), which are obtained from images shown in Fig. 9 (a) by converting them into gray scale images and halftoning them, and printing the shares for those secret images on transparencies in various dimensions. Then we have found that when shares are printed on 1/4 or 1/2 size of A4-transparencies, which is approximately equivalent to 70dpi or 50dpi, the quality of the decoded images and the difficulty of the pixel-positioning operation are well-balanced.

Figures 9 (c), (d), and (e) give the whole of the decoded images obtained by Construction 1, 3 (rand) and 3 (8-LF), respectively. The magnifications of the parts of those decoded image, which correspond to the part indicated by the white square of $50 \times 35$ pixels in Fig. 9 (f), are shown in Fig. 10, where all images are printed in 50dpi. The number of pixels and depth of images shown in Figs. 9 and 10 are indicated in their captions.

We see from figures in Fig. 10 that the quality of the decoded image is well maintained even if the shares constructed by the proposed method are printed on the transparency of size that is easy to handle.

4. Application of the Proposed Methods

4.1 Application to RGB Images

Since an RGB image consists of R, G and B channels and each channel can be regarded as a gray scale image, our proposed methods can be naturally extend. For this purpose, we begin the explanation with the procedure to generate a secret RGB image from an original RGB image. We first decompose \(^1\) the original RGB images into gray scale images of R, G and B channels. Then, we convert these gray scale

---

\(^1\)The function of Decompose in GIMP [2] is used to obtain images of R, G and B channels from RGB images.
images into binary images $X_c$ ($c \in \{R, G, B\}$) by halftoning. By recomposing $X_R$, $X_G$ and $X_B$, we have the secret RGB image $X$.

Now we propose Construction 4 that construct shares satisfying the conditions given in Eq. (2) for two secret RGB images.

**Construction 4:** Execute the following steps.

1. Decompose the secret RGB images $X$ and $Y$ respectively into the binary images $X_c$ and $Y_c$ ($c \in \{R, G, B\}$).
2. For $c \in \{R, G, B\}$, construct $S^{(\ell)}_c$ ($\ell = 1, 2$) for $X_c$ and $Y_c$.
3. Recompose $S^{(\ell)}_c$ ($\ell = 1, 2, c \in \{R, G, B\}$) to construct RGB shares $S^{(\ell)}$ ($\ell = 1, 2$).

Note that the method to construct shares $S^{(\ell)}_c$ in Step 2 of Construction 4 is selected from Construction 1, Construction 3 (rand.) and Construction 3 (8-LF) in advance of the execution of Construction 4. We refer to Construction 4 with Construction 1, Construction 3 (rand.) and Construction 3 (8-LF) in Step 2 as Construction 4-1, Construction 4-3 (rand.) and Construction 4-3 (8-LF), respectively.

**Example 6:** We again consider the images $O_{X_2}$ and $O_{Y_2}$ shown in Fig. 2 (a).

Before constructing shares, we convert these original RGB images into the secret RGB images $X_4$ and $Y_4$ by the procedure described at the beginning of this subsection. Then Fig. 11 shows the secret RGB images $X_4$ and $Y_4$ and decoded images for $X_4$ and $Y_4$ that are obtained from shares constructed by the three different methods. Tables 1 and 2 respectively show the number and percentage of positions $(i, j)$ of $X_4$ and $Y_4$ for which Eq. (8) does not hold. Since Step 2 of Construction 4 is executed for each $c \in \{R, G, B\}$, the number of $c$ for which Eq. (8) does not hold varies from 0 to 3 for each position $(i, j)$. Therefore, the number of positions shown in Tables 1 and 2 is counted for the respective number of $c \in \{R, G, B\}$.

The decoded images shown in Fig. 11 (b) are obtained from shares without any constraint. For this case, the shares $S^{(\ell)}_c$ ($\ell = 1, 2$) for $X_4$ and $T^{(\ell)}_c$ ($\ell = 1, 2$) for $Y_4$ can be individually constructed by Construction 4-1. Though the decoded images are darker than the secret images, the shape of contents can be well recognized and their color is reproduced satisfactorily.

The decoded images shown in Fig. 11 (c) are obtained from shares constructed by Construction 4-3 (rand.). According to Table 1 (resp. Table 2), 26.6% (resp. 30.1%) of pixels of the decoded image for $X_4$ (resp. $Y_4$) have at least one incorrectly decoded color among RGB. In spite of this fact, both the shape of contents and the color are fairly reproduced.

The decoded images shown in Fig. 11 (d) are obtained from shares constructed by Construction 4-3 (8-LF).
Though the outline of contents is emphasized, the color of the decoded images becomes worse. This may be because the incorrectly decoded pixels are gathered by 8-neighbor Laplacian filter.

Comparing the decoded images shown in Figs. 11(c) and (d), the former exhibits the better result with respect to color than the latter, while the latter exhibits the better result with respect to sharpness than the former. Hence, we can conclude that the application of 8-LF for RGB images has both merits and demerits, and we should determine whether we employ 8-LF in the proposed methods depending on the characteristics of the original images.

### 4.2 Application to Binary Geometric Images

In this subsection, we consider the application of the proposed methods to binary geometric images.

In order to reduce the number of incorrectly decoded pixels that are recognizable in the decoded images, we have demonstrated in Sect. 3.4 the application of halftoning in the generation of secret images from given images is quite effective. But it should be noted that direct halftoning causes no effect on binary images. In order to solve this problem, we convert a binary image into gray scale images with two grayish colors before applying halftoning. More precisely, we convert black and white pixels in an original image respectively into pixels with gray of appropriate percentages.

**Example 7:** We begin this example by considering the images shown in Fig. 12(a) as the secret images $X_5$ and $Y_5$. The

![Example Images](image-url)

**Fig. 10** Results of experiments for images with small pixel size (2).

#### Table 1

<table>
<thead>
<tr>
<th>Number of $c$</th>
<th>Number of positions</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>352179</td>
<td>73.4%</td>
</tr>
<tr>
<td>1</td>
<td>108763</td>
<td>22.7%</td>
</tr>
<tr>
<td>2</td>
<td>17873</td>
<td>3.7%</td>
</tr>
<tr>
<td>3</td>
<td>1191</td>
<td>0.2%</td>
</tr>
<tr>
<td>Total</td>
<td>480000</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

#### Table 2

<table>
<thead>
<tr>
<th>Number of $c$</th>
<th>Number of positions</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>335307</td>
<td>69.9%</td>
</tr>
<tr>
<td>1</td>
<td>120218</td>
<td>25.0%</td>
</tr>
<tr>
<td>2</td>
<td>22390</td>
<td>4.7%</td>
</tr>
<tr>
<td>3</td>
<td>2085</td>
<td>0.4%</td>
</tr>
<tr>
<td>Total</td>
<td>480000</td>
<td>100.0%</td>
</tr>
</tbody>
</table>
decoded images for $X_5$ and $Y_5$ obtained from the shares constructed by Construction 3 (8-LF) are shown in Fig. 12 (b).

We see from these decoded images that there are a considerable number of unnecessary lines and dots are appeared.

In order to reduce such degradation of the quality of decoded images, we next consider the images $\tilde{X}_5$ and $\tilde{Y}_5$ shown in Fig. 12 (c) as interim secret images. These interim secret images are respectively generated from $X_5$ and $Y_5$ by first converting black (resp. white) pixels into 80% (resp. 20%) gray pixels, then halftoning them. We can say that the contents of those interim secret images are well recognized, though the contrast becomes lower.

Firstly, we construct shares for $\tilde{X}_5$ and $\tilde{Y}_5$ by Construction 3 (rand.). As we see in the decoded images shown in Fig. 12 (d), some unnecessary lines are attenuated compared with the decoded images shown in Fig. 12 (b). But the contrast of the decoded images becomes lower.

Secondly, we construct shares for $\tilde{X}_5$ and $\tilde{Y}_5$ by Construction 3 (8-LF). As we see in the decoded images shown in Fig. 12 (e), some unnecessary lines are attenuated compared with the decoded images shown in Fig. 12 (b). But some necessary lines are also disappeared. On the other hand, it can be pointed out that the contrast of the decoded image is higher than that of Fig. 12 (d). So, at the viewpoint of contrast, 8-neighbor Laplacian filter can improve the quality of decoded images of the proposed method.

Example 7 shows that the application of halftoning for binary geometric images as a pre-processing has both merits and demerits. By combining the results shown in Examples 5 and 7, we should determine whether we employ halftoning as a pre-processing depending on the characteristics of original images.
5. Conclusion

In this paper, we have considered the implementation of (2, 2)-VSSS to share two secret images at the same time only by two shares that can be used reversibly, and investigated the methods to improve the quality of decoded images.

In the proposed methods to construct shares, we have applied 8-neighbor Laplacian filter to choose incorrectly decoded pixels in the secret images. The examples have shown that this approach can be effective to make the decoded image sharp. We have also applied halftoning to the secret images before constructing shares by the proposed methods. Then the examples have shown that this approach has succeeded to make the incorrect pixels in the decoded images relatively unnoticed. Finally, we have extended the proposed methods for RGB images.

The most principal difference between the proposed and conventional VSSS is that, in the proposed method, we allow some pixels in the secret image to be incorrectly decoded. This characteristic implies that original images such that the incorrectly decoded pixels are not conspicuous in the decoded image are suited to the proposed method. More concretely, images in which pixels of various colors are spread, i.e., RGB nature images, are suited to the proposed method. On the other hand, images that are divided into parts with only a few colors, i.e., binary geometric images, are not suited to the proposed method.

In addition, when incorrectly decoded pixels exist in the decoded image, its contrast becomes lower. Therefore, original images whose contrast is relatively high and that contain a few objects with clear outline, i.e., face of animals, front of flowers, and so on, are suited to the proposed method.

In this paper, we have concentrated on (2, 2)-VSSS with reversible shares. But it should be emphasized that the technique employed in the proposed methods, that is, 8-neighbor Laplacian filter and halftoning, are applicable to not only (2, 2)-VSSS with reversible shares but also general (k, n)-VSSS with various constraints on the usage of shares. The most important issues are how to choose the pixels that are allowed to be incorrectly decoded and how to make them relatively unnoticed.

Hence it needs a further study to develop the methods how to choose the incorrectly decoded pixels in the secret images. The prospective methods will improve the contrast for binary images and the color for RGB images. Moreover, we should clarify how to adjust the percentages of gray in the conversion of binary images into gray scale images before applying halftoning. It is also important as a future work to generalize the proposed methods for (2, 2)-VSSS to the methods for arbitrary (k, n)-VSSS.

Acknowledgments

The authors are grateful to the associated editor and the anonymous reviewers for their helpful comments and suggestions that greatly improve the manuscript.

References


Mariko Fujii received B.E. degree in information and communication sciences and M.E. degree in information sciences from Sophia University, Tokyo, Japan, in 2017 and 2019, respectively. Her research interests are the theory of information security and its applications, especially, various kinds of secret sharing schemes.
Tomoharu Shibuya received B.E., M.E., and Ph.D. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1992, 1994 and 1999, respectively. He joined Tokyo Institute of Technology in 1994 as a research associate in the Department of Electrical and Electronic Engineering, and served as a research associate in the Department of Communications and Integrated Systems from 1999 to 2003. He served as an associate professor in R&D Department, National Institute of Multimedia Education from 2003 to 2008. He joined Sophia University in 2008, and since 2015, he has served as a professor in the Department of Information and Communication Sciences. He received the Young Engineer Award from the IEICE of Japan in 1997 and the Ericsson Young Scientist Award in 2005. His research interests are error control coding, data security, information theory, and communication theory. He is a member of IEEE, SIAM, MSJ, and IPSJ.