Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **2**, March **2018**

Local Blackness Preserving Visual Cryptography for Grayscale Secret Images

Bin Yan, Na Chen

College of Electronics, Communication and Physics
Shandong University of Science and Technology
579, Qian-Wan-Gang Road, Qingdao, P. R. China, 266590
yanbinhit@hotmail.com; bin.yan.cn@ieee.org

Hong-Mei Yang

College of Information Science and Engineering
Shandong University of Science and Technology
579, Qian-Wan-Gang Road, Qingdao, P. R. China, 266590
yhm1998@163.com

Jian-Jun Hao

College of Electronics, Communication and Physics
Shandong University of Science and Technology
579, Qian-Wan-Gang Road, Qingdao, P. R. China, 266590

Received October, 2017; revised November, 2017

Abstract. *Size-invariant visual cryptography (VC) for grayscale image has lower computational load during encoding, and lower memory and bandwidth requirements during storage and transmission. However, its visual quality is lower than the size-expanded VC, and sometimes unacceptable, especially for grayscale secret images. In order to improve the visual quality, we propose a local blackness preserving (LBP) VC algorithm. A block-based VC is designed with a mechanism to recover the local ratio between the black pixels and white pixels that is destroyed by the raw VC encoder. Comparison with recently proposed size-invariant VC algorithms confirms the effectiveness of the proposed method.*
Keywords: Visual cryptography; Grayscale image; Image Quality; Preserving local blackness; Size-invariant.

1. **Introduction.** Traditional visual cryptography (VC) algorithms are designed for binary secret images, such as binary text images or binary logo images [1]. But a grayscale image usually contains more information content than a binary image, hence the use of grayscale image as secret image in visual cryptography offers more visual details to the receiver [2]. In addition, these grayscale images can also be used as cover images in extended VC[3]. So in general, there are two typical scenarios involving grayscale images:

- Use the grayscale image as a secret image[2, 4, 5]. In this case, the generated shares are meaningless. Such a scenario is useful when the content of the secret needs to be described by an image rather than by a simple binary text/image. This application scenario is the focus of this work.

- Use the grayscale images as cover images in extended VC [3, 6, 7, 8]. In this case, the secret image is binary. Such a scenario is useful when the user is more concerned about camouflage, i.e., generating the VC shares so as not to arouse suspicion from the attacker. Furthermore these meaningful shares can also provide extra information to ease the management of these shares.

There is also a third scenario, where both the secret image and the cover images are grayscale images. But such a requirement imposes too much constrains on the design of the VC system. Currently, there is no existing algorithm that can provide acceptable visual quality.

The benefit of using grayscale secret image over binary image is that the grayscale contains more details than binary image. To make these details visible to the targeted receiver, the VC algorithm must be able to preserve the local brightness of the grayscale image. Since most VC algorithms are designed for binary image and a grayscale image can be represented by a halftone image, so a relevant *research problem* is: how to preserve the local brightness of the grayscale image when encoding its halftone counterpart.

In this paper, we propose to use block-based encoding and local blackness preserving algorithm to enhance the perceptual quality of the reconstructed secret image. The proposed algorithm is shown to provide higher tone similarity and higher structure similarity between the grayscale image and the reconstructed halftone image, when compared with typical VC for grayscale images.

This paper is organized as follows. In section 2, we review briefly the classical size-expanded VC, in order to introduce symbols and to lay the foundation for block based encoding. Then we present the proposed algorithm in section 3. In section 4, we discuss the experimental results and comparisons with a typical size-invariant VC algorithms for grayscale images. Finally, we conclude the paper in section 5.

2. **Review and Preliminaries.** In this section, we review briefly the size-expanded VC and multiple-pixel block encoding for size-invariant VC. The purpose is to introduce symbols and to lay the foundation for introducing the proposed algorithm.

We use the symbol '1' to represent a black pixel printed on the transparencies, and use the symbol '0' to represent a transparent (or white) pixel printed on the transparencies. So the set of colors on the shares and the stacked images is $\mathbb{Z}_2 \triangleq \{0, 1\}$. Next we define the stacking operation. Let $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_2^{1\times m}$, then the stacking operation between the two vectors are defined as element-wise stacking of the corresponding components from $\mathbf{x}$ and $\mathbf{y}$: $\mathbf{x} \boxplus \mathbf{y} \triangleq (x_1 \boxplus y_1, \cdots, x_m \boxplus y_m)$. The symbol $\boxplus$ here represents the logical 'OR' operation between two boolean quantities. This stacking operation can be extended to more than two vectors. The number of black pixels is an important feature of an image block, so we use the following operation to extract this feature: $\mathcal{B} : \mathbb{Z}_2^{1\times m} \to \mathbb{Z}_m$, which is defined as

$$\mathcal{B}(\mathbf{x}) = \sum_{i=1}^{m} x_i.$$

where $\mathbb{Z}_m \triangleq \{0, 1, \cdots, m-1\}$.

2.1. **Size-expanded VC.** Since our local blackness preserving VC uses the basis matrix from the size-expanded VC, so we review briefly the size-expanded VC as introduced by Naor and Shamir[1].

In a typical (k, n)-threshold VC algorithm, the secret is shared among n parties (which are usually called participants) with the requirement that only more than k shares can reveal the secret and less than k shares can't leak any information of the secret image. The construction of (k, n)-threshold VC algorithms usually utilizes the basis matrices.

Let $\mathbf{B}_0, \mathbf{B}_1 \in \mathbb{Z}_2^{n\times m}$ be two basis matrices and $s \in \mathbb{Z}_2$ is a secret pixel to be shared. In the encryption side, according to the secret pixel s, the basis matrix $\mathbf{B}_s$ is chosen. After random permutation of the columns of $\mathbf{B}_s$, each row is distributed to one participant. Two conditions on the basis matrices must be satisfied: contrast condition and security condition. The contrast condition requires that the stacking of more than k rows of $\mathbf{B}_1$ must provide more black pixels than that of $\mathbf{B}_0$. The security condition requires that the $r < k$ rows taken from $\mathbf{B}_0$ and $\mathbf{B}_1$ are indistinguishable. So that from only $r < k$ shares, one can't figure out whether the secret pixel is 1 or 0.

2.2. **Review of Size-Invariant VC.** Using the size-expanded VC, each secret pixel is represented by m pixels on each share. When $m > 1$, this leads to pixel expansion. Pixel expansion may increase the processing time in encoding. In addition, it also increases the transmission bandwidth and the storage space.

To solve this problem, different size-invariant VC algorithms were proposed in the past decades. The typical algorithms include Ito's size-invariant VC[9], Yang's probabilistic approach[10], and the random grid based approaches[11, 12, 13]. However, the probabilistic nature of these algorithms only guarantees that globally the contrast between the black pixels and the white pixels are preserved. But locally, this contrast may be destroyed, leading to worse perceptual quality than the size-expanded VC. This degradation in perceptual quality is especially prominent for grayscale and halftone image.

To improve the visual quality for size-invariant VC, one must be able to preserve the local contrast. So block (or multiple pixel) based approach should be employed [5, 14, 15, 16].

Hou *et. al.* extended the basic Ito algorithm [5]. Instead of encoding each white/black pixel independently, r successive white/black pixels are taken from the image and encoded. This can ensure that for each r white/black pixels, the local contrast is guarantee. But these r white/black pixels may not be adjacent to each other.

In [14], image blocks are classified according to the number of black pixels in them. Then a counter is assigned to each type of block. For example, for the type-b block having b black pixels, use this counter to ensure that the matrix $\mathbf{B}_1$ is used exactly b times to encode this type of block. Here the contrast is guaranteed for each type of block, but not locally in a small region.

Chen *et. al.* use the average gray level of an image block to select the corresponding block on the stacked image: an image block with darker average gray value is mapped to a block on the stacked image with more black pixels [15]. However, each block is processed independently so that the loss of contrast in one block cannot be compensated by other blocks.

To remedy the existing problems outlined above, we propose to use local blackness preserving VC, where the loss of contrast in one block can be compensated by other adjacent blocks.

3. **Local Blackness Preserving (LBP) VC.** In this section, we describe the LBP algorithm. For a grayscale input secret image, the processing steps are illustrated in Fig.1. Here we consider the $(2, 2)$-threshold VC and the algorithm can be extended to the general (n, n)-threshold.

The secret image $g[\mathbf{n}]$ is indexed by $\mathbf{n} \triangleq [n_r, n_c]$, where n_r is the row index and n_c is the column index when the image is treated as a matrix.

3.1. **Equalization to limited range.** As the first step, the grayscale image is equalized to the range of $[0, 2^b - 1]$, where b is the number of bits used to represent each pixel. For example, $b = 8$ or $b = 16$ are quite commonly used in digital image processing.

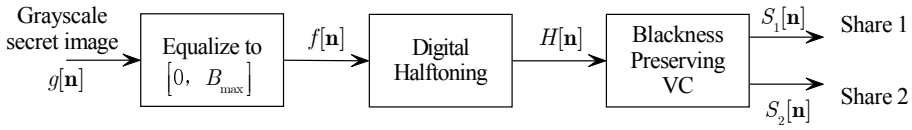

FIGURE 1. The overall block diagram of the proposed system.

Equalization may improve the visual quality but equalizing to a smaller range may seems counterintuitive. This is done in view of the contrast loss during the VC encoding process. As shown in [1], for a (n, n)-threshold VC, the maximum relative contrast on the stacked image is $\alpha < \frac{1}{2^{n-1}}$. So if $n = 2$, the range of the reconstructed secret image is $[0,\ 2^b - 1]$.

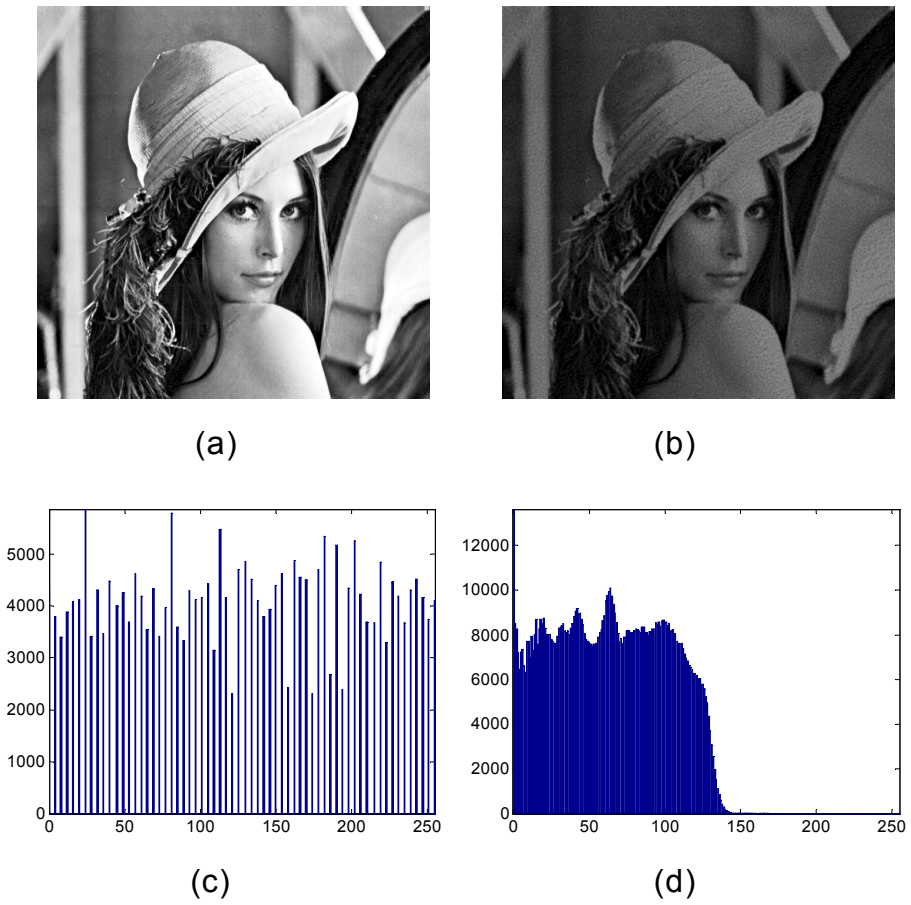

FIGURE 2. Illustrate the loss of contrast from VC. (a) Equalized Lena image, (b) filtered stacking result, (c) the histogram of (a), and (d) the histogram of (b).

The loss of contrast from VC is illustrated in Fig.2. In order to illustrate the shrinkage of histogram due to VC, first we equalized the secret image Lena to the range $[0, 255]$. The equalized image is shown in Fig.2(a) and the corresponding histogram is shown in Fig.2(c). We then perform VC on the equalized image and stack the two shares. The VC algorithm used here is the basic Naor VC algorithm with pixel expansion 4 [1]. A Gaussian filter with variance 4 is then used to smooth the stacking result and the histogram is calculated from this smoothed image. The smoothed stacking result and the corresponding histogram are shown in Fig.2(b) and Fig.2(d) respectively. Comparing Fig.2(c) and Fig.2(d), it is apparent that the global contrast is reduced after VC encoding and decoding/stacking.

The change of vertical scale from Fig.2(c) to Fig.2(d) is due to pixel expansion, i.e., the number of pixels on the stacked image is 4 times of that of the original image. The theoretical result from Naor and the observation from Fig. 2 validate our *equalization to limited range* operation. Furthermore, this loss of contrast also suggests that when evaluating the quality of the stacking result for size-invariant VC, we should use the stacking result from size-expanded VC as a reference image.

3.2. **Halftoning via error diffusion.** In the second step, we transform the equalized grayscale image $f[\mathbf{n}]$ into a halftone image $H[\mathbf{n}]$. we use the well-known error diffusion halftoning with a minor modification, due to its good tradeoff between quality and computational complexity. The block diagram is shown in Fig. 3.

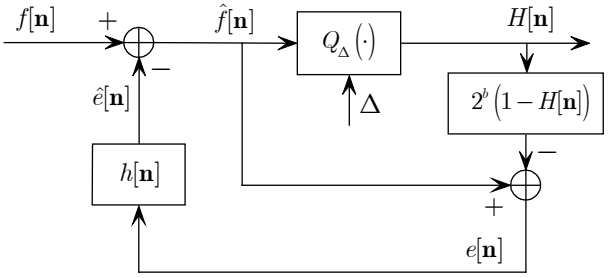

FIGURE 3. The modified error diffusion based halftoning.

In error diffusion, the pixel intensity is quantized by a quantizer $Q_\Delta(\cdot)$, where Δ is the quantization level. If b bits are used to represent each pixel, then $\Delta = 2^{b-1}$. Considering the convention of using $H[\mathbf{n}] = 1$ to represent a black pixel in printed shares, so the quantizer output $Q_\Delta\left(\hat{f}[\mathbf{n}]\right)$ should be:

$$H[\mathbf{n}] = \begin{cases} 0, & \text{if } \hat{f}[\mathbf{n}] \geq \Delta; \\ 1, & \text{if } \hat{f}[\mathbf{n}] < \Delta. \end{cases}$$

Then the quantization error can be calculated as $e[\mathbf{n}] = \hat{f}[\mathbf{n}] - (1 - H[\mathbf{n}]) \times 2^b$. To compensate for this error, the current pixel is modified by:

$$\hat{f}[\mathbf{n}] = f[\mathbf{n}] + \sum_{\mathbf{m} \in \mathcal{N}(\mathbf{n})} h[\mathbf{m}]\, e[\mathbf{m}],$$

where $\mathcal{N}(\mathbf{n})$ is the neighbors of the current pixel $\mathbf{n}$ and $h[\mathbf{m}]$ is the coefficient of the diffusion filter. Different diffusion method uses different neighborhood. In this paper, we use the simple Floyd-Steinberg diffusion, where the neighborhood is:

$$\mathcal{N}(\mathbf{n}) = \{(n_r - 1, n_c - 1), (n_r - 1, n_c), (n_r - 1, n_c + 1), (n_r, n_c - 1)\}.$$

The corresponding diffusion filter coefficients are:

$$(h(n_r - 1, n_c - 1), h(n_r - 1, n_c), h(n_r - 1, n_c + 1), h(n_r, n_c - 1)) = \left(\frac{1}{16}, \frac{5}{16}, \frac{3}{16}, \frac{7}{16}\right).$$

After the halftoning process, we get a binary image $H[\mathbf{n}] \in \mathbb{Z}_2^{M \times N}$, whose local average brightness/blackness is equal to that of the corresponding grayscale pixel. But unfortunately, when generating the shares, this local blackness is destroyed by ordinary size-invariant VC encoder. In the next step, we design a VC algorithm which can preserve this local blackness.

3.3. **VC encryption and local blackness preservation.** If we use $\hat{H}[\mathbf{n}]$ to denote the recovered secret image by stacking the two share images $S_1[\mathbf{n}]$ and $S_2[\mathbf{n}]$, then we would like that in a small region of $\hat{H}[\mathbf{n}]$, the ratio between black pixels and white pixels is the same as that of in the same region of $H[\mathbf{n}]$. To attain this goal, we take a small block of size $K \times K$ in $H[\mathbf{n}]$, say $\mathbf{A}$, and encode it into shares. But after stacking the shares, the corresponding block $\hat{\mathbf{A}}$ may have different blackness due to the loss of contrast in VC. Here we introduce a *blackness compensation* approach. If $\mathcal{B}(\hat{\mathbf{A}}) - \mathcal{B}(\mathbf{A}) = c$, where $c > 0$, then we borrow c black pixels from the neighboring blocks.

In the proposed LBP VC algorithm, the halftone image is processed block by block in a raster scanning order, from the top left to the bottom right. The current block is encrypted using a size expanded VC according to its blackness.

We use the set $\mathcal{F} = \{\beta_1, \cdots, \beta_P\}$ to denote the set of gray levels that can be reproduced in the stacking result. Here β_i is the number of black pixels, or blackness, in a $K \times K$ block. For example, for a $(2, 2)$-threshold scheme with $m = 4$ and basis matrices

$$\mathbf{B}_0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \ \mathbf{B}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \tag{1}$$

the set $\mathcal{F} = \{\beta_1, \beta_2, \beta_3\} = \{2, 3, 4\}$. Then, a new set of basis matrices can be designed in order to reproduce the set of colors in $\mathcal{F}$. These new basis matrices are based on the original basis matrices $\mathbf{B}_0, \mathbf{B}_1$. For example, for the basis matrices example in Eq.1, we can obtain

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \ \mathbf{M}_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \ \mathbf{M}_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{2}$$

To get a type β_i block on $\hat{H}[\mathbf{n}]$, we use the basis matrix $\mathbf{M}_i$. The columns of $\mathbf{M}_i$ are randomly permuted and then the rows are distributed to shares, just like in the basic Naor scheme.

Unfortunately, the blackness of the block taken from $H[\mathbf{n}]$ may take values outside of $\mathcal{F}$. For example, for $m = 4$, the set of possible blackness on $H[\mathbf{n}]$ is $\mathcal{H} = \{0, 1, 2, 3, 4\} = \mathbb{Z}_5$. So from $\mathcal{H}$ to $\mathcal{F}$, there should be a non-invertible mapping (many-to-one).

The key problem here is: how to design the mapping from $\mathcal{H}$ to $\mathcal{F}$ to minimize the possible loss of contrast? In [15], a histogram width/depth equalization is utilized, while in [4], the skewness of the histogram is explored to design this type of mapping. But these methods are global ones, where the local contrast is not preserved. Here we use a different approach. First, we use the *histogram equalization to limited range* to increase the possibility that the type of blocks are in $\mathcal{F}$. Then, each block type in $\mathcal{H}$ is mapping to its closest one in $\mathcal{F}$. For the example above, block types are mapped as follows: $0, 1, 2 \to 2, 3 \to 3$, and $4 \to 4$. This 'lossy' mapping may cause the type 0,1,2 in $\mathcal{H}$ to be indistinguishable on the stacked image, leading to loss of local contrast. Fortunately, this loss of local contrast can be remedied by the following *blackness compensation* procedure.

3.4. **Blackness compensation.** After encrypting the current block, the blocks on the share images are concurrently superimposed to find out the stacked block on the reconstructed image $\hat{H}[\mathbf{n}]$. Then the black pixels in the neighbors are adjusted to preserve the local blackness that is changed by the VC process. An example is shown in Fig.4 to illustrate this basic idea. The current block has blackness 1. But after VC encoding and stacking, the blackness of the stacking result is 2. If we leave the neighbors unchanged, then the local blackness on the stacked image will be higher than that of the original halftone image. To preserve the local blackness, one black pixel is borrowed from one of the neighboring blocks. In this paper, we consider neighborhood system that is

similar to those in the error diffusion. The difference is that here each neighbor is a $K \times K$ block, while in error diffusion, each neighbor is a pixel. After flipping a black pixel in a neighboring block, we make the total number of black pixels unchanged in this small neighborhood. In general, we may need to flip more than one black pixels in this neighborhood, depending on the difference $\mathcal{B}(\hat{\mathbf{A}}) - \mathcal{B}(\mathbf{A})$.

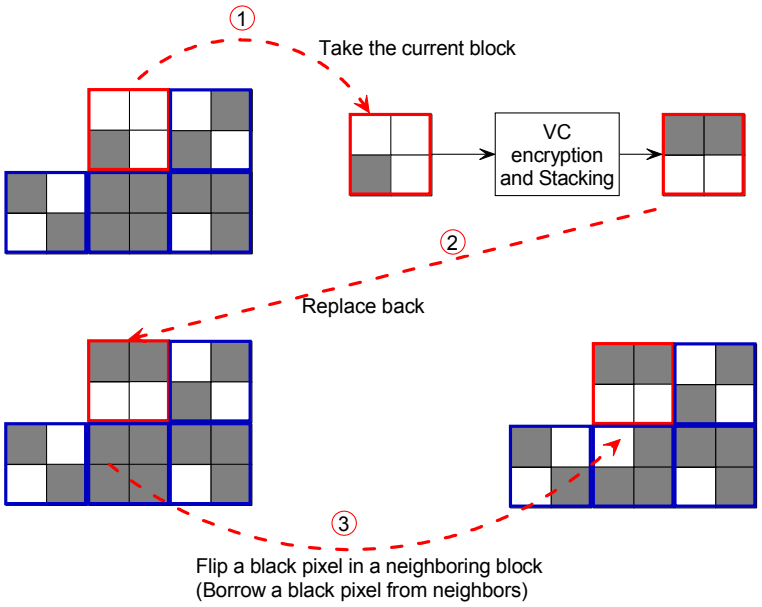

FIGURE 4. Illustration of the basic idea of local blackness preservation.

In Algorithm 1, we list the pseudo-code of basic VC encryption for one block when given a basis matrix $\mathbf{B}$. A bit flipping algorithm is used to 'borrow' bits from the neighboring blocks. This bit flipping algorithm is shown in Algorithm 2. Using these algorithms as building blocks, the blackness compensation algorithms is built, and pseudo-code is listed in Algorithm 3.

Algorithm 1 $(\mathbf{S}_1, \mathbf{S}_2) \leftarrow \text{VcEnc}(\mathbf{B})$: VC encoding using the basis matrix $\mathbf{B}$.

Input:

Basis matrix $\mathbf{B} \in \mathbb{Z}_2^{n \times m}, n = 2$

Output:

Share blocks: $\mathbf{S}_1, \mathbf{S}_2 \in \mathbb{Z}_2^{K \times K}, m = K^2$

$\hat{\mathbf{B}} = \begin{bmatrix} \hat{\mathbf{b}}_1 \\ \hat{\mathbf{b}}_2 \end{bmatrix} \leftarrow$ Randomly permutate the columns of $\mathbf{B}$

for $i \leftarrow 1$ to K **do**
  for $j \leftarrow 1$ to K **do**
    $S_1(i,j) \leftarrow \hat{\mathbf{b}}_1(i + (j-1)K)$;
    $S_2(i,j) \leftarrow \hat{\mathbf{b}}_2(i + (j-1)K)$;
  end for
end for

3.5. **Security analysis.** The security of our approach is guaranteed by the security of the basic size-expanded VC. If the security condition of the basic matrices are guaranteed, i.e., the $r < k$ rows taken from $\mathbf{B}_0$ and $\mathbf{B}_1$ are indistinguishable, then from any r share blocks one cannot infer the secret image block. So our algorithm is secure.

Algorithm 2 $\left\{\hat{\mathbf{A}}_i\right\}_{i=1}^{4} \leftarrow \text{FlipOneBit}\left\{\mathbf{A}_i\right\}_{i=1}^{4}$: Flip one pixel/bit in one of the blocks from $\left\{\mathbf{A}_i\right\}_{i=1}^{4}$ which has the highest blackness.

Input:

A set of input blocks: $\left\{\mathbf{A}_i\right\}_{i=1}^{4}$, with $\mathbf{A}_i \in \mathbb{Z}_2^{K\times K}$

Output:

A set of output blocks: $\left\{\hat{\mathbf{A}}_i\right\}_{i=1}^{4}$

$k \leftarrow \arg\max_{i\in\{1,2,3,4\}} \mathcal{B}\left(\mathbf{A}_i\right)$
if $k \neq 0$ **then**
  $\hat{\mathbf{A}}_k \leftarrow$ Flip one black pixel in $\mathbf{A}_k$.
  $\hat{\mathbf{A}}_i \leftarrow \mathbf{A}_i, \forall i \neq k$.
else
  $\hat{\mathbf{A}}_i \leftarrow \mathbf{A}_i, \forall i$.
end if

Algorithm 3 LBPVC: Local Blackness Preserving VC Encoding

Input:

Grayscale secret image $g\left[\mathbf{n}\right] \in \mathbb{Z}_{256}^{M\times N}$.
Size of the block: K (Assume $K = 2$ here)

Output:

Share images: $S_1\left[\mathbf{n}\right] \in \mathbb{Z}_2^{M\times N}, S_2\left[\mathbf{n}\right] \in \mathbb{Z}_2^{M\times N}$;

$f\left[\mathbf{n}\right] \leftarrow$ Equalize $g\left[\mathbf{n}\right]$ to the range [0, 127].
$H\left[\mathbf{n}\right] \leftarrow$ Halftone $f\left[\mathbf{n}\right]$ by error diffusion.
for $i \leftarrow 1$ to $\lfloor \frac{M}{K} \rfloor$ **do**
  for $j \leftarrow 1$ to $\lfloor \frac{N}{K} \rfloor$ **do**
    $\mathbf{H}_{ij} \leftarrow$ The (i, j)-th block in $H\left[\mathbf{n}\right]$.
    if $\mathcal{B}\left(\mathbf{H}_{ij}\right) = 4$ **then**
      $(\mathbf{S}_{ij}^1, \mathbf{S}_{ij}^2) \leftarrow \text{VcEnc}(\mathbf{M}_3)$ {The matrix $\mathbf{M}_i$ is defined in Eq.(2).}
    else if $\mathcal{B}\left(\mathbf{H}_{ij}\right) = 3$ **then**
      $(\mathbf{S}_{ij}^1, \mathbf{S}_{ij}^2) \leftarrow \text{VcEnc}(\mathbf{M}_2)$
    else if $\mathcal{B}\left(\mathbf{H}_{ij}\right) = 2$ **then**
      $(\mathbf{S}_{ij}^1, \mathbf{S}_{ij}^2) \leftarrow \text{VcEnc}(\mathbf{M}_1)$
    else if $\mathcal{B}\left(\mathbf{H}_{ij}\right) = 1$ **then**
      $(\mathbf{S}_{ij}^1, \mathbf{S}_{ij}^2) \leftarrow \text{VcEnc}(\mathbf{M}_1)$
      $\mathcal{N}_{ij} \triangleq \{(i, j+1), (i+1, j-1), (i+1, j), (i+1, j+1)\}$.
      $\{\mathbf{H}_{k\ell} : (k, \ell) \in \mathcal{N}_{ij}\} \leftarrow \text{FlipOneBit}\left(\{\mathbf{H}_{k\ell} : (k, \ell) \in \mathcal{N}_{ij}\}\right)$
    else if $\mathcal{B}\left(\mathbf{H}_{ij}\right) = 0$ **then**
      $(\mathbf{S}_{ij}^1, \mathbf{S}_{ij}^2) \leftarrow \text{VcEnc}(\mathbf{M}_1)$
      $\mathcal{N}_{ij} \triangleq \{(i, j+1), (i+1, j-1), (i+1, j), (i+1, j+1)\}$.
      $\{\mathbf{H}_{k\ell} : (k, \ell) \in \mathcal{N}_{ij}\} \leftarrow \text{FlipOneBit}\left(\{\mathbf{H}_{k\ell} : (k, \ell) \in \mathcal{N}_{ij}\}\right)$
      $\{\mathbf{H}_{k\ell} : (k, \ell) \in \mathcal{N}_{ij}\} \leftarrow \text{FlipOneBit}\left(\{\mathbf{H}_{k\ell} : (k, \ell) \in \mathcal{N}_{ij}\}\right)$
    end if
    Insert the share blocks $\mathbf{S}_{ij}^1, \mathbf{S}_{ij}^2$ into the (i, j)-th block in $S_1\left[\mathbf{n}\right]$ and $S_2\left[\mathbf{n}\right]$ respectively.
  end for
end for

4. **Experiments.** In this section, we present the experimental result on a set of testing images.

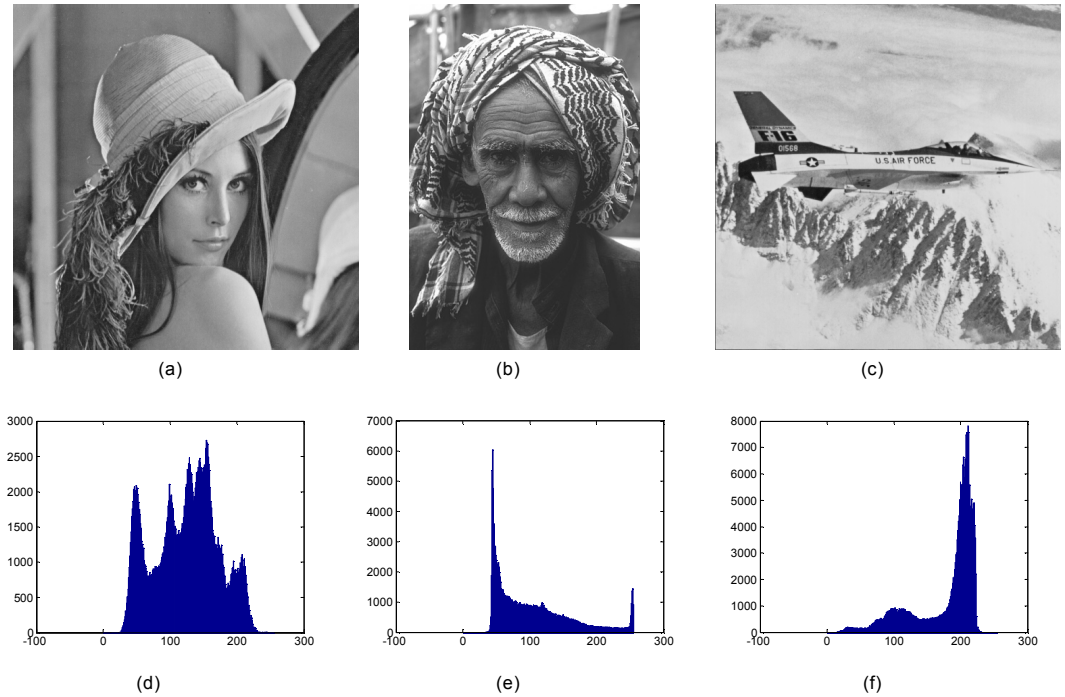

FIGURE 5. Three types of images used in the experiments. (a) Lena image with centered histogram, (b) Man image with left-skewed histogram, (c) Airplane image with right-skewed histogram, (d) histogram of (a), (e) histogram of (b), and (f) histogram of (c).

The performance of the algorithm is characterized by its perceptual quality. This perceptual quality between the grayscale image and the reconstructed halftone image is measured by PSNR (Peak Signal to Noise Ratio) and MSSIM (Mean Structural Similarity Measure). The PSNR measures the tone similarity. Before calculating PSNR, the halftone image is smoothed using a Gaussian filter with variance 4. This Gaussian filtering is used to simulate the low-pass characteristic of human visual system (HVS). The MSSIM measures the local structural similarity between two images [17]. So they reflect different aspects of the quality of the reconstructed secret image.

Since the PSNR measures tone similarity, so the images involved in calculating PSNR must have the same dynamic range. If the two images have quite different dynamic range, then a large PSNR value may result even though the two images are structurally quite similar. Considering the loss of contrast due to VC encoding, as discussed in section 3.1, we use the stacking result from size-expanded VC as the reference image. The smoothed stacking result from Naro's size-expanded VC is downsampled by a factor of 2 before calculating the PSNR value.

To demonstrate the quality improvement, we compare our algorithm with recent result reported in [4]. This reference algorithm provided state of art performance and was shown to outperform Ito's size invariant VC [9] and Chen's algorithm [15]. Three types of testing images with different histograms are used in the experiments, as shown in Fig. 5. The Lena image has a symmetric and centered histogram. The histogram of the Man image is skewed to the left and the histogram of the Airplane image is skewed to the right.

The stacking results are compared in Fig. 6. Better tone approximation can be observed from the LBP method, for example, the mirror frame in Lena image, and the dark areas in

the Airplane image. The result from Lee's method exhibits saturation in dark region. In addition, more details are visible from the Man's image when using LBP. The numerical results for PSNR and MSSIM are shown in Table 1. LBP can provide consistently better performance than Lee's method. Obviously, these improvements are results of preserving the local blackness.

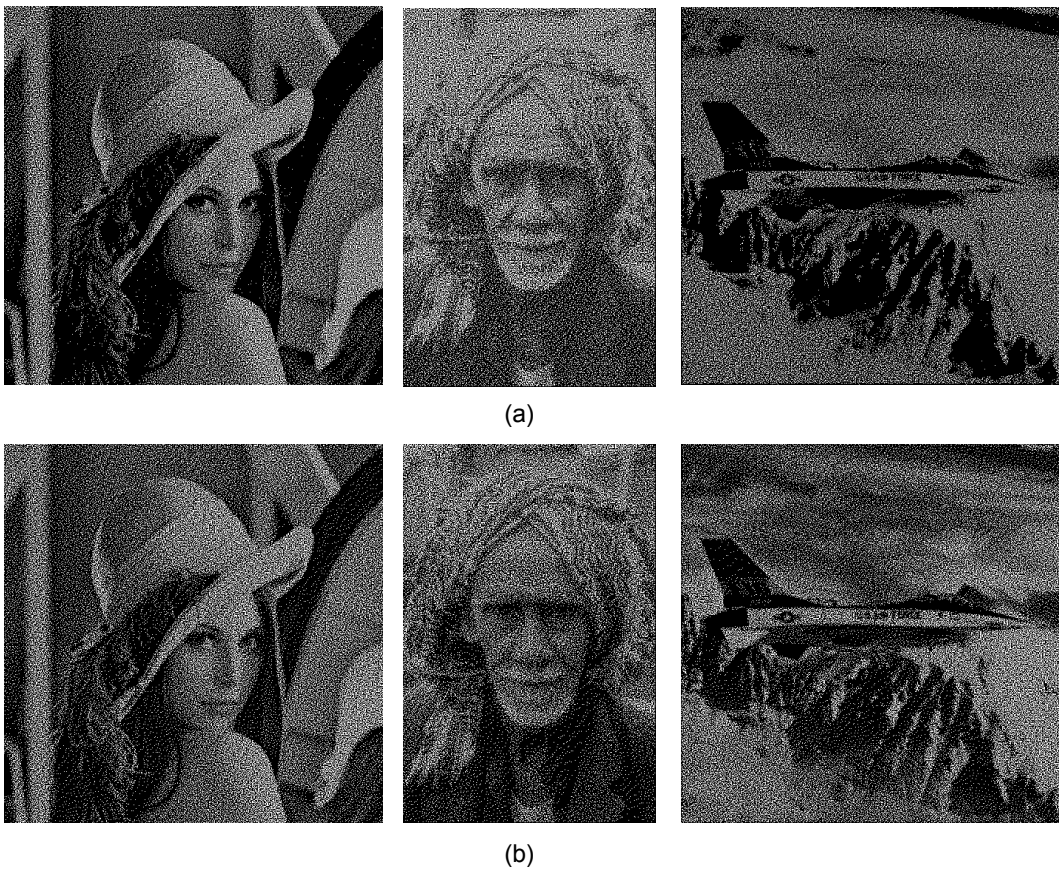

FIGURE 6. Comparison of visual quality. (a) Result from Lee [4], (b) result from our LBP method. Please zoom in to see the details.

TABLE 1. Performance Metric

Image	*Metric*	*Lee 2014* [4]	*LBP*
Lena	PSNR(dB)	18.77	20.65
	MSSIM	0.6377	0.6960
Man	PSNR(dB)	12.29	18.47
	MSSIM	0.5252	0.6283
Airplane	PSNR(dB)	13.17	13.78
	MSSIM	0.5490	0.5511

4.1. **Batch Test Result.** To evaluate the performance of the LBP algorithm on a larger database, we choose the popular 24 images from the Kodak database. The size of the images are 768×512 and 512×768, as shown in Fig. 7. Each of the test images is converted from RGB to gray level. Then the VC algorithms are applied on this gray level image. The average PSNR and average MSSIM is measured for Lee's algorithm and our

LBP algorithm. The results are reported in Table 2. A sample reconstructed images are shown in Fig. 8. The missing numbers on Fig. 8(a) are visible in Fig. 8(b).

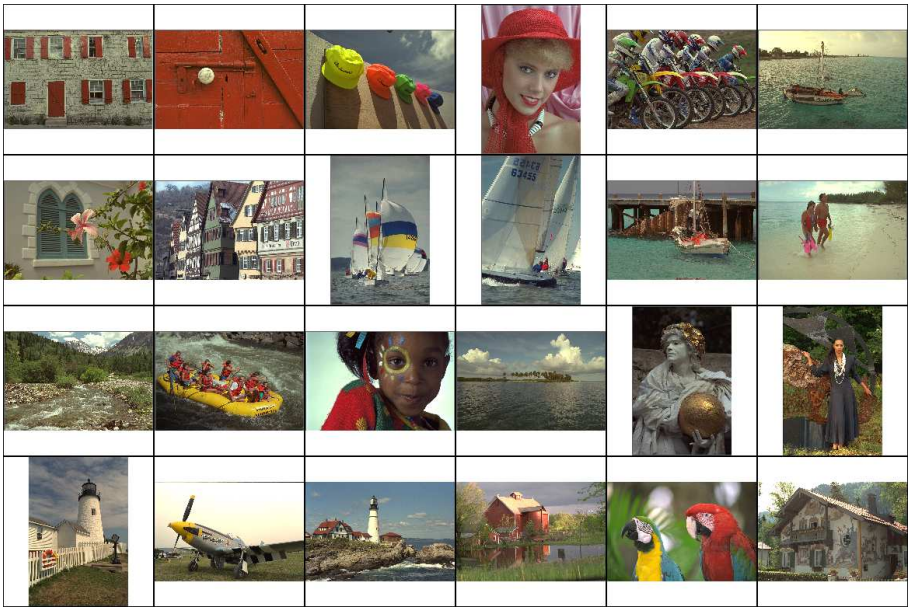

FIGURE 7. The set of images from the Kodak database used in the batch test. The indices of the images are from 1 to 24, from left to right and from top to bottom.

TABLE 2. Performance Metric on Kodak Images

Algorithm	*Avg. PSNR*	*Avg. MSSIM*
Lee [4]	11.0306	0.5455
LBP	16.5695	0.6881

5. **Conclusions.** In this paper, we identified the existing problem in size-invariant VC and proposed a local contrast preserving VC algorithm. It is confirmed by experiments that using the proposed LBP method, more details of the images can be recovered on the stacking results. Both the PSNR and MSSIM metrics show improved performance when compared with Lee's recently proposed size-invariant VC. This algorithm can be easily extended to any (n, n)-threshold VC scheme.

Acknowledgement. This work is supported by Shandong Provincial Natural Science Foundation (No. ZR2014JL044), the Key Statistics Research Project of Shandong Province (No. KT15104), the National Natural Science Foundation of China (NSFC)(No. 61272432). The work of Hong-Mei Yang is also supported by Qingdao Scientific Development Plan (No. KJZD-13-28-JCH).

REFERENCES

[1] M. Naor and A. Shamir, Visual cryptography, in *EUROCRYPT'94*, pp. 1–12, 1994.
[2] Y.-C. Hou, Visual cryptography for color images, *Pattern Recognition*, vol. 36, pp. 1619–1629, 2003.
[3] Z.-M. Wang, G. R. Arce, and G. D. Crescenzo, Halftone visual cryptography via error diffusion, *IEEE Transactions on Information Forensics and Secruity*, vol. 4, no. 3, pp. 383–396, 2009.

(a) (b)

FIGURE 8. Comparison of visual quality of the 10-th image in Kodak dataset. (a) Result from Lee [4], (b) result from our LBP method. Please zoom in to see the details.

[4] C.-C. Lee, H.-H. Chen, H.-T. Liu, G.-W. Chen, and C.-S. Tsai, A new visual cryptography with multi-level encoding, *Journal of Visual Languages and Computing*, vol. 25, no. 3, pp. 243 – 250, 2014.

[5] Y.-C. Hou and S.-F. Tu, A visual cryptographic technique for chromatic images using multi-pixel encoding method, *Journal of Research and Practice in Information Technology*, vol. 37, pp. 179–191, 2005.

[6] F. Liu and C. K. Wu, Embedded extended visual cryptography schemes, *IEEE Transactions on Information Forensics and Security*, vol. 6, pp. 307–322, June 2011.

[7] X. Yan, S. Wang, X. Niu, and C.-N. Yang, Halftone visual cryptography with minimum auxiliary black pixels and uniform image quality, *Digital Signal Processing*, vol. 38, pp. 53 – 65, 2015.

[8] B. Yan, Y.-F. Wang, L.-Y. Song, and H.-M. Yang, Size-invariant extended visual cryptography with embedded watermark based on error diffusion, *Multimedia Tools and Applications*, vol. Online First, pp. 1–24, 2015.

[9] R. Ito, H. Kuwakado, and H. Tanaka, Image size invariant visual cryptography, *IEICE Transactions on Fundamentals*, vol. E82-A, pp. 2172–2177, Oct. 1999.

[10] C.-N. Yang, New visual secret sharing schemes using probabilistic method, *Pattern Recognition Letters*, vol. 25, pp. 481–494, 2004.

[11] O. Kafri and E. Keren, Encryption of pictures and shapes by random grids, *Optics Letters*, vol. 12, pp. 377–379, Jun 1987.

[12] S. J. Shyu, Image encryption by random grids, *Pattern Recognition*, vol. 40, pp. 1014–1031, Mar. 2007.

[13] R. D. Prisco and A. D. Santis, On the relation of random grid and deterministic visual cryptography, *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 653–665, 2014.

[14] S.-F. Tu and Y.-C. Hou, Design of visual cryptographic methods with smooth looking decoded images of invariant size for grey-level images, *The Imaging Science Journal*, vol. 55, no. 2, pp. 90–101, 2007.

[15] Y.-F. Chen, Y.-K. Chan, C.-C. Huang, M.-H. Tsai, and Y.-P. Chu, A multiple-level visual secret-sharing scheme without image size expansion, *Information Sciences*, vol. 177, no. 21, pp. 4696 – 4710, 2007.

[16] Y.-W. Chow, W. Susilo, and D. Wong, Enhancing the perceived visual quality of a size invariant visual cryptography scheme, in *Information and Communications Security* (T. Chim and T. Yuen, eds.), vol. 7618 of *Lecture Notes in Computer Science*, pp. 10–21, Springer Berlin Heidelberg, 2012.
[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.