A NEW APPROACH TO COLOR IMAGE SECRET SHARING

R. Lukac[1] *K.N. Plataniotis*[1] *B. Smolka*[2+] *A.N. Venetsanopoulos*[1]

[1] The Edward S. Rogers Sr. Department of ECE, University of Toronto, 10 King's College Road,
Toronto, Ontario M5S 3G4, Canada, *e-mails: {lukacr, kostas, anv}@dsp.utoronto.ca*
[2] Department of Automatic Control, Silesian University of Technology,
Akademicka 16 Str., 44-101 Gliwice, Poland, *e-mail: bsmolka@ia.polsl.gliwice.pl*

ABSTRACT

A new secret sharing scheme for color images is introduced. Using the $\{k,n\}$-secret sharing strategy the proposed method encrypts the color image into n color shares. The secret information is recovered only if the k (or more) allowed shares are available for decryption. Both encryption and decryption operations are performed by operating at the bit-levels of the decomposed color image. Modifying the spatial arrangements of the binary components the method produces color shares which vary in both the spectral characteristics among the RGB components and the spatial correlation between the neighboring color vectors. Since encryption is performed in the decomposed binary domain, there is no obvious relationship in the RGB color domain between any two color shares or between the original color image and any of the n shares. This increases protection of the secret information. Inverse cryptographic processing of the shares must be realized in the decomposed binary domain and the procedure reveals the original color image with perfect reconstruction.

1. INTRODUCTION

A $\{k,n\}$-visual secret sharing (VSS) scheme is a popular cryptographic tool used for protection of image information [6]. Encrypting the image into n, seemingly random, shares, the VSS technique allows for sharing of the secret image among a group of n participants. The shared secret can be recovered only when a coalition of k willing participants are polling their encrypted images, the so-called shares, together [1],[3]. The secret information can be visually revealed if any k (or more) recipients stack their shares printed as transparencies on an overhead projector. On the other hand, any $(k-1)$ or fewer shares cannot be used to decrypt the transmitted information.

Based on the nature of visual cryptography, the natural images must be first binarized and then encrypted. Image halftoning techniques [8],[9] are commonly used to convert continuous-tone images into images with a binary representation. Due to a frosted/transparent representation of the shares produced by the VSS schemes, the decrypted image is never identical with the original continuous-tone image. Moreover, the encryption procedure increases spatial resolution and decreases contrast of the decrypted binarized input. Thus, the color visual cryptography schemes [4] which are currently in use generate decrypted images with noticeable visual impairments.

The proposed secret sharing scheme operates directly on the bit planes of the color image. By stacking individually encrypted bit planes, the scheme produces n color shares useful for secure distribution over the untrusted public networks. Note that the shares vary in both the spectral characteristics among the RGB components and the spatial correlation between the neighboring color vectors. The decryption function recovers the original color image unchanged. Since the decrypted output is available in a digital format, this feature in conjunction with the overall simplicity of the approach make the method attractive for modern image processing and communication systems.

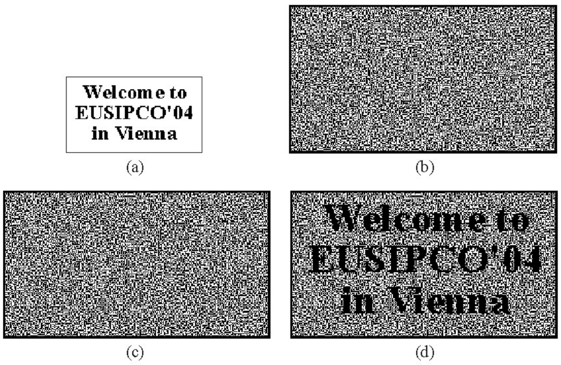

Figure 1: Visual cryptography demonstrated through a $\{2,2\}$-scheme: (a) original binary image, (b,c) share images, (d) decrypted output.

2. CONVENTIONAL SECRET SHARING SCHEME

Due to its algorithmic nature, conventional visual cryptography schemes operate on a binary input [5]. Assuming a $K_1 \times K_2$ binary image (black and white image with 0 values denoting the black and 1 values denoting the white), each binary pixel $r_{(i,j)}$ determined by spatial coordinates $i = 1,2,...,K_1$ and $j = 1,2,...,K_2$ is replaced via an encryption function $f_e(\cdot)$ with a $m_1 \times m_2$ block of black and white pixels in each of the n shares [7]. Repeating the process for each input pixel, a $K_1 \times K_2$ input binary image is encrypted into n binary shares each one with a spatial resolution of $m_1K_1 \times m_2K_2$ pixels. Since the spatial arrangement of the pixels varies from block to block, the original information cannot be revealed without accessing a predefined number of shares (*Figure 1*).

Let as assume a basic $\{2,2\}$-threshold structure which is the basic case designed within the $\{k,n\}$-VSS framework [4]. Assuming for simplicity a basic structure with 2×2 blocks $\mathbf{s}_1 = [s'_{(2i-1,2j-1)}, s'_{(2i-1,2j)}, s'_{(2i,2j-1)}, s'_{(2i,2j)}] \in S_1$ and $\mathbf{s}_2 =$

[+] Supported by the KBN grant 4T11F01824.

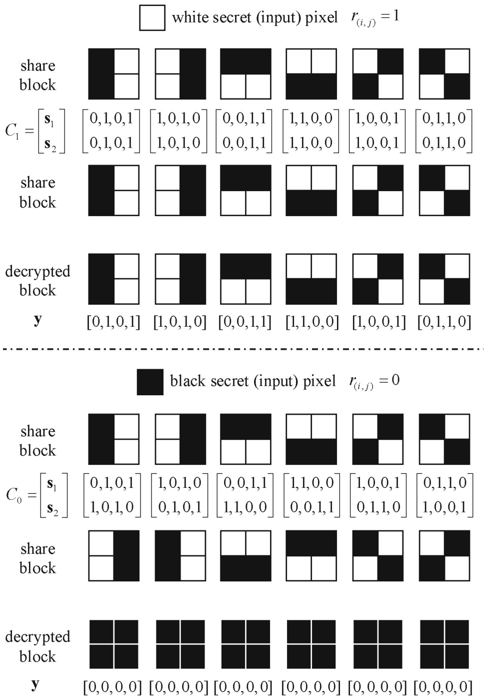
white secret (input) pixel $r_{(i,j)}=1$
share block
$C_1=\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix}$
$\begin{bmatrix}0,1,0,1\\0,1,0,1\end{bmatrix}$ $\begin{bmatrix}1,0,1,0\\1,0,1,0\end{bmatrix}$ $\begin{bmatrix}0,0,1,1\\0,0,1,1\end{bmatrix}$ $\begin{bmatrix}1,1,0,0\\1,1,0,0\end{bmatrix}$ $\begin{bmatrix}1,0,0,1\\1,0,0,1\end{bmatrix}$ $\begin{bmatrix}0,1,1,0\\0,1,1,0\end{bmatrix}$
share block
decrypted block
$\mathbf{y}$ [0,1,0,1] [1,0,1,0] [0,0,1,1] [1,1,0,0] [1,0,0,1] [0,1,1,0]
black secret (input) pixel $r_{(i,j)}=0$
share block
$C_0=\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix}$
$\begin{bmatrix}0,1,0,1\\1,0,1,0\end{bmatrix}$ $\begin{bmatrix}1,0,1,0\\0,1,0,1\end{bmatrix}$ $\begin{bmatrix}0,0,1,1\\1,1,0,0\end{bmatrix}$ $\begin{bmatrix}1,1,0,0\\0,0,1,1\end{bmatrix}$ $\begin{bmatrix}1,0,0,1\\0,1,1,0\end{bmatrix}$ $\begin{bmatrix}0,1,1,0\\1,0,0,1\end{bmatrix}$
share block
decrypted block
$\mathbf{y}$ [0,0,0,0] [0,0,0,0] [0,0,0,0] [0,0,0,0] [0,0,0,0] [0,0,0,0]

Figure 2: Visual cryptography strategy.

$[s''_{(2i-1,2j-1)}, s''_{(2i-1,2j)}, s''_{(2i,2j-1)}, s''_{(2i,2j)}] \in S_2$, the encryption process is defined as follows:

$$f_e(r_{(i,j)}) = \begin{cases} [\mathbf{s}_1,\mathbf{s}_2]^T \in C_0 & \text{for } r_{(i,j)} = 0 \\ [\mathbf{s}_1,\mathbf{s}_2]^T \in C_1 & \text{for } r_{(i,j)} = 1 \end{cases} \qquad (1)$$

where C_0 and C_1 are the sets obtained by permuting the columns of the $n \times m_1 m_2$ basis matrices A_0 and A_1, respectively [5]. Since $m_1 m_2$ represents the factor by which each share is larger than the original image, it is desirable to make $m_1 m_2$ as small as possible [2]. In the case of the $\{2,2\}$-VSS the optimal choice m_1 and m_2 leads to $m_1 = 2$ and $m_2 = 2$ resulting in 2×2 blocks $\mathbf{s}_1$ and $\mathbf{s}_2$.

Assuming the $\{2,2\}$-VSS the sets $C_0 = \left\{\begin{bmatrix}0,1,0,1\\1,0,1,0\end{bmatrix}, \begin{bmatrix}1,0,1,0\\0,1,0,1\end{bmatrix}, \begin{bmatrix}0,0,1,1\\1,1,0,0\end{bmatrix}, \begin{bmatrix}1,1,0,0\\0,0,1,1\end{bmatrix}, \begin{bmatrix}1,0,1,0\\0,1,0,1\end{bmatrix}, \begin{bmatrix}0,1,1,0\\1,0,0,1\end{bmatrix}\right.$ and $C_1 = \left\{\begin{bmatrix}0,1,0,1\\0,1,0,1\end{bmatrix}, \begin{bmatrix}1,0,1,0\\1,0,1,0\end{bmatrix}, \begin{bmatrix}0,0,1,1\\0,0,1,1\end{bmatrix}, \begin{bmatrix}1,1,0,0\\1,1,0,0\end{bmatrix}, \begin{bmatrix}1,0,0,1\\1,0,0,1\end{bmatrix}, \begin{bmatrix}0,1,1,0\\0,1,1,0\end{bmatrix}\right.$ include all matrices obtained by by permuting the columns of the 2×4 basis matrices A_0 and A_1, respectively [5]. The basic matrices considered here are defined as follows:

$$A_0 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, A_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \qquad (2)$$

Figure 2 shows the principle of both encryption and decryption used in visual cryptography. If a secret pixel is white, i.e. $r_{(i,j)} = 1$, then each pixel in $\mathbf{s}_1$ is equivalent to each pixel in $\mathbf{s}_2$, and thus, $[\mathbf{s}_1,\mathbf{s}_2]^T$ can be any member of set C_1. If a secret pixel is black, i.e. $r_{(i,j)} = 0$, then each pixel in $\mathbf{s}_1$ should complement each pixel in $\mathbf{s}_2$ and thus, $[\mathbf{s}_1,\mathbf{s}_2]^T$ should be selected from set C_0. The choice of $[\mathbf{s}_1,\mathbf{s}_2]^T$ is guided by a random number generator, which determines the random character of the shares.

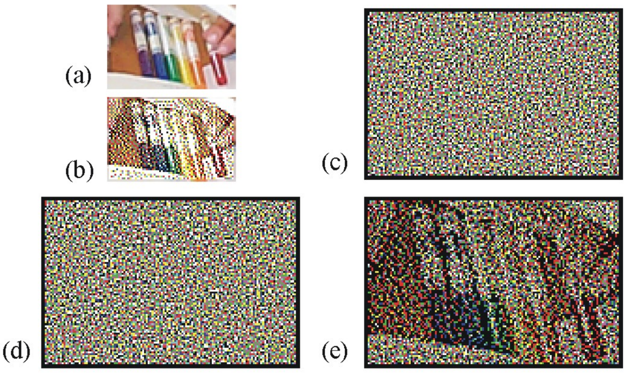

Figure 3: Images obtained using by halftoning based $\{2,2\}$-scheme [4]: (a) original color image, (b) halftone image, (c,d) share images, (e) decrypted output image.

Decrypting 2×2 share blocks $\mathbf{s}_1 = [s'_{(u,v)}, s'_{(u,v+1)}, s'_{(u+1,v)}, s'_{(u+1,v+1)}] \in S_1$ and $\mathbf{s}_2 = [s''_{(u,v)}, s''_{(u,v+1)}, s''_{(u+1,v)}, s''_{(u+1,v+1)}] \in S_2$, for $u = 1,3,...,2K_1 - 1$ and $v = 1,3,...,2K_2 - 1$, used in a $\{2,2\}$-scheme the decrypted block y of size 2×2 is produced as black $\mathbf{y} = [0,0,0,0]$ if $\mathbf{s}_1 \neq \mathbf{s}_2$. Otherwise the share blocks $\mathbf{s}_1$ and $\mathbf{s}_2$ are identical and the decrypted block is recovered with the same spatial arrangement of binary pixels as in the share blocks.

The application of a conventional $\{k,n\}$-secret sharing scheme to a $K_1 \times K_2$ natural image requires halftoning [4],[5]. Using the approach of [4], the original color image (*Figure 3a*) is first transformed into a $K_1 \times K_2$ halftone image (*Figure 3b*) by using the density of the net dots to simulate the intensity levels [8]. Since each color channel of the halftone image is a binary image, it is appropriate for VSS-based encryption. *Figure 3c* and *Figure 3d* show two $2K_1 \times 2K_2$ color shares obtained using the $\{2,2\}$ sharing scheme. A $2K_1 \times 2K_2$ color image depicted in *Figure 3e* correspond to the decrypted output.

Visual inspection of both the original image shown in *Figure 3a* and the recovered image depicted in *Fig.3e* indicates that the decrypted image is darker, the input image is of quarter size compared to the decrypted output, and the output color image contains a number of color artifacts and shifted colors resulting from the nature of the algorithm.

3. PROPOSED METHOD

Let $\mathbf{x} : Z^2 \rightarrow Z^3$ be a $K_1 \times K_2$ Red-Green-Blue (RGB) color image representing a two-dimensional matrix of the three-component color vectors (pixels) $\mathbf{x}_{(i,j)} = [x_{(i,j)1}, x_{(i,j)2}, x_{(i,j)3}]$ located at the spatial position (i,j), for $i = 1,2,...,K_1$ and $j = 1,2,...,K_2$. Assuming that c describes the color channel (i.e. $c = 1$ for Red, $c = 2$ for Green, and $c = 3$ for Blue) and the color component $x_{(i,j)c}$ is coded with B bits allowing $x_{(i,j)c}$ to take an integer value between 0 and $2^B - 1$, the color vector $\mathbf{x}_{(p,q)}$ can be equivalently expressed in a binary form as follows:

$$\mathbf{x}_{(i,j)} = \sum_{b=1}^{B} \mathbf{x}^b_{(i,j)} 2^{B-b} \qquad (3)$$

where $\mathbf{x}^b_{(i,j)} = [x^b_{(i,j)1}, x^b_{(i,j)2}, x^b_{(i,j)3}] \in \{0,1\}^3$ denotes the binary vector at the b-bit level, with $b = 1$ denoting the most significant bits (MSB).

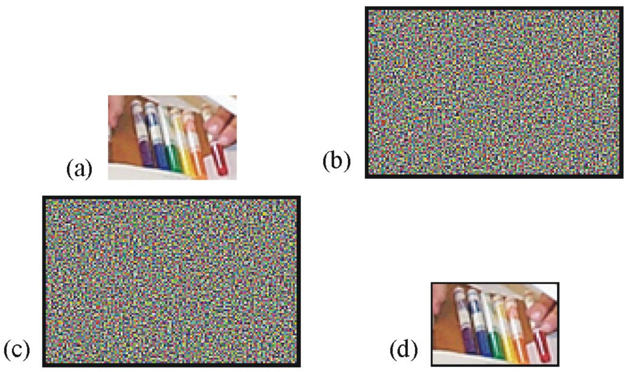

Figure 4: Images obtained using the proposed scheme: (a) original color image, (b,c) share images, (d) decrypted output image.

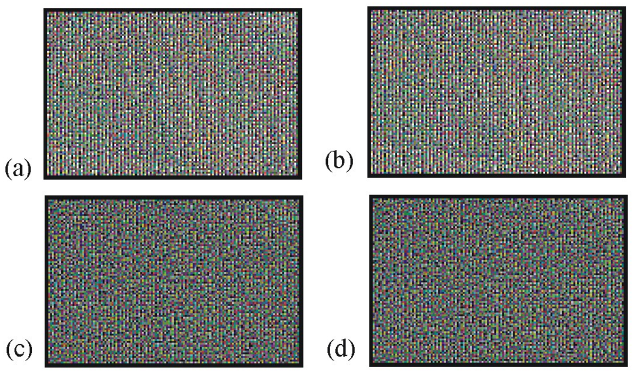

Figure 5: Color shares S_1 and S_2 obtained encrypting only the selected bit-levels: (a,b) MSB $b=1$, (c,d) two most significant bits $b=1$ and $b=2$.

If the c-th component of the binary vector $\mathbf{x}^b_{(i,j)}$ is white ($x^b_{(i,j)c}=1$), encryption is performed through $[\mathrm{s}_1,\mathrm{s}_2]^T \in C_1$ replacing $x^b_{(i,j)c}$ by binary blocks s_1 and s_2 in each of the two shares. Otherwise, the reference binary component is black ($x^b_{(p,q)c}=0$), and encryption is defined via $[\mathrm{s}_1,\mathrm{s}_2]^T \in C_0$. This forms an encryption function defined as follows:

$$f_e(x^b_{(i,j)c}) = \begin{cases} [\mathrm{s}_1,\mathrm{s}_2]^T \in C_0 & \text{for } x^b_{(i,j)c}=0 \\ [\mathrm{s}_1,\mathrm{s}_2]^T \in C_1 & \text{for } x^b_{(i,j)c}=1 \end{cases} \tag{4}$$

By replacing the binary components $x^b_{(i,j)c}$ with binary blocks s_1 and s_2 for one particular b, the process generates two $2K_1 \times 2K_2$ vector-valued binary shares S^b_1 and S^b_2, respectively. A random number generator guides the choice of $[\mathrm{s}^b_1,\mathrm{s}^b_2]^T$ and determines the random character of S^b_1 and S^b_2. Thus, the process modifies both the spatial correlation between spatially neighboring binary vectors $\mathrm{s}'^b_{(u,v)} = [s'^b_{(u,v)1}, s'^b_{(u,v)2}, s'^b_{(u,v)3}] \in S^b_1$ or $\mathrm{s}''^b_{(u,v)} = [s''^b_{(u,v)1}, s''^b_{(u,v)2}, s''^b_{(u,v)3}] \in S^b_1$, for $u=1,2,...,2K_1$ and $v=1,2,...,2K_2$, and the spectral correlation among components $s'^b_{(u,v)c}$ or $s''^b_{(u,v)c}$, for $c=1,2,3$, of the individual binary vectors $\mathrm{s}'^b_{(u,v)}$ or $\mathrm{s}''^b_{(u,v)}$, respectively. Bit-level stacking of the encrypted bit-levels produces the color vectors $\mathrm{s}'_{(u,v)} \in \mathbf{S}_1$ and $\mathrm{s}''_{(u,v)} \in \mathbf{S}_2$ as

$$\mathrm{s}'_{(u,v)} = \textstyle\sum_{b=1}^{B} \mathrm{s}'^b_{(u,v)} 2^{B-b} \tag{5}$$

$$\mathrm{s}''_{(u,v)} = \textstyle\sum_{b=1}^{B} \mathrm{s}''^b_{(u,v)} 2^{B-b} \tag{6}$$

Due to random processing taking place at the bit-levels, the color shares $\mathbf{S}_1$ and $\mathbf{S}_2$ contain only random, color noise like information (*Figure 4b,c*). Since encryption is realized in the decomposed binary vector space, no detectable relationship between the original color vectors $\mathbf{x}_{(p,q)}$ and the color noise of $\mathbf{S}_1$ or $\mathbf{S}_2$ can be found in the RGB color domain. This considerably increases security and prevents unauthorized decryption through brute-force enumeration.

Since the proposed method is designed for computer-centric processing in modern image communication systems which should utilize the complete image characteristics of the original color input the decryption procedure must satisfy the perfect reconstruction property. The original color data must be recovered from the color shares $\mathbf{S}_1$ and $\mathbf{S}_2$ using inverse algorithmic steps. Therefore, the decryption procedure is applied to the decomposed binary vector arrays of the color shares. Assuming that (i,j), for $i=1,2,...,K_1$ and $j=1,2,...,K_2$, denotes the spatial position in the original image and c denotes the color channel, the corresponding 2×2 binary share blocks are $\mathrm{s}'^b_c = \{s'^b_{(2i-1,2j-1)c}, s'^b_{(2i-1,2j)c}, s'^b_{(2i,2j-1)c}, s'^b_{(2i,2j)c}\}$ and $\mathrm{s}''^b_c = \{s''^b_{(2i-1,2j-1)c}, s''^b_{(2i-1,2j)c}, s''^b_{(2i,2j-1)c}, s''^b_{(2i,2j)c}\}$. Based on the arrangements of the basis matrices A_0 and A_1 of the $\{2,2\}$-VSS used in this paper for image encryption, if both blocks are consistent, i.e. $\mathrm{s}'^b_c = \mathrm{s}''^b_c$, the decrypted original bit $x^b_{(i,j)c}$ is assign white, i.e. $x^b_{(i,j)c}=1$. Otherwise, the blocks are inconsistent, i.e. $\mathrm{s}'^b_c \neq \mathrm{s}''^b_c$ and the original bit is recovered as black, i.e. $x^b_{(i,j)c}=0$. This logical comparison forms the following decryption function

$$x^b_{(i,j)c} = f_d(\mathrm{s}'^b_c, \mathrm{s}'^b_c) = \begin{cases} 1 & \text{for } \mathrm{s}'^b_c = \mathrm{s}''^b_c \\ 0 & \text{for } \mathrm{s}'^b_c \neq \mathrm{s}''^b_c \end{cases} \tag{7}$$

which is used to restore the binary vectors $\mathbf{x}^b_{(i,j)}$. The procedure completes with the bit-level stacking (3) resulting in the original color vector $\mathbf{x}_{(i,j)}$.

Note that more generally, the decryption function is described as

$$x^b_{(i,j)c} = f_d(\mathrm{s}'^b_c, \mathrm{s}'^b_c) = \begin{cases} 0 & \text{for } [\mathrm{s}'^b_c, \mathrm{s}''^b_c]^T \in C_0 \\ 1 & \text{for } [\mathrm{s}'^b_c, \mathrm{s}''^b_c]^T \in C_1 \end{cases} \tag{8}$$

where reciprocal operations to (1) are searched. The decrypted color output is depicted in *Figure 4d*. Since the proposed method satisfies the perfect reconstruction property, the output image is identical to the original depicted in *Figure 4a*.

Figure 5 allows for the visual comparison of the color shares when cryptographic processing is applied to a reduced set of binary levels. It can be seen that due to spatial variations of the binary components included in the sets C_0 and C_1 of the encryption function (1) as well as the additional encryption level obtained by modifying the spectral characteristics of the image, a sufficient level of protection is achieved

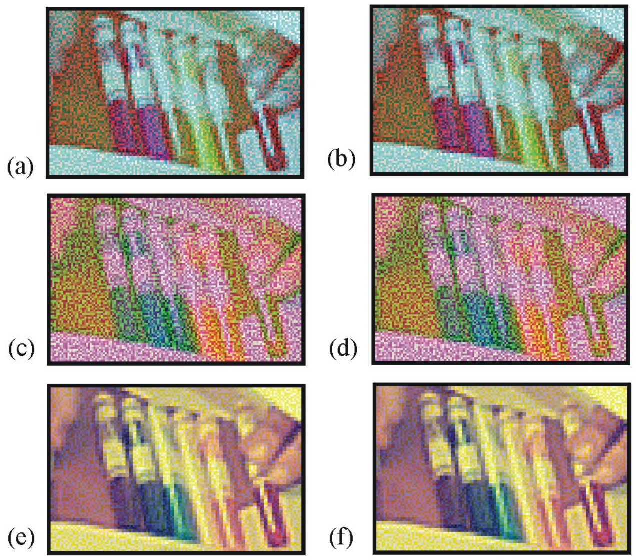

Figure 6: Color shares S_1 and S_2 obtained encrypting all the bits $b = 1, 2, ..., B$ only in a single-color channel: (a,b) R channel with $c = 1$, (c,d) G channel with $c = 2$, (e,f) B channel with $c = 3$.

by cryptographically processing the first two most significant bits ($b = 1, 2$) in all three RGB channels $c = 1, 2, 3$. The remaining bits of the original image vectors can be simply copied into the shares unchanged. If this option is selected, image decryption has to be also performed only for $b = 1, 2$. Applying the cryptographic operations for the MSB (*Figure 5a,b*) of the color image only, fine details are sufficiently encrypted, however, large flat regions can be visually revealed. However, encrypting the two most significant bits $b = 1, 2$ of the color image (*Figure 5a,b*), the color shares should be sufficiently protected against unauthorized attacks.

Encrypting only either one color channel (*Figure 6*) or two color channels (*Figure 7*) of the RGB color image, the procedure significantly modifies color information in the shares and introduces random, noise-like information. However such an encryption operation allows to reveal the image content. These results clearly show that for a sufficient level of security all the channels of the RGB image must be encrypted.

4. CONCLUSION

A new secret sharing scheme with perfect reconstruction of the color inputs was introduced. The method encrypts the color image replacing the bit components with a block of bits for each of the color shares. Using the bit-level encryption of the color image the method produces color shares, each with unique spatial and spectral characteristics. The proposed bit-level encryption increases protection against attacks performed in the RGB color domain. The decryption operations performed at the bit-levels are designed to satisfy the perfect reconstruction property and thus, the procedure recovers the original color image unchanged. This makes the proposed method attractive for a modern image processing and communication system, where the decrypted output can be used for subsequent processing tasks.

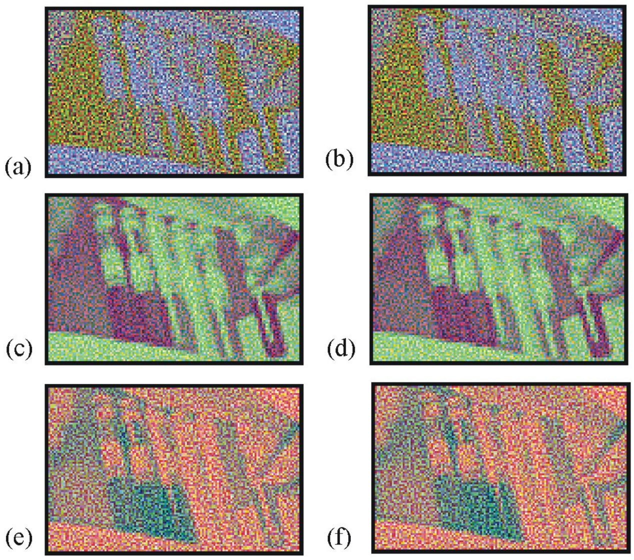

Figure 7: Color shares S_1 and S_2 obtained encrypting all the bits $b = 1, 2, ..., B$ in two color channels: (a,b) RG channels with $c = 1$ and $c = 2$, (c,d) RB channels with $c = 1$ and $c = 3$, (e,f) GB channels with $c = 2$ and $c = 3$.

REFERENCES

[1] G. Ateniese, C. Blundo, A. de Santis, and D.G. Stinson, "Visual cryptography for general access structures," *Information and Computation*, vol. 129, pp. 86-106, September 1996.

[2] P.A. Eisen and D.R. Stinson, "Threshold visual cryptography schemes with specified levels of reconstructed pixels," *Design, Codes and Cryptography*, vol. 25, no.1, pp. 15-61, January 2002.

[3] T. Hofmeister, M. Krause, and H.U. Simon, "Contrast optimal *k* out of *n* secret sharing schemes in visual cryptography," *Theoretical Computer Science*, vol. 240, no. 2, pp. 471-485, June 2000.

[4] J.C. Hou, "Visual cryptography for color images," *Pattern Recognition*, vol. 36, no. 7, pp. 1619-1629, July 2003.

[5] C.C. Lin and W.H. Tsai, "Visual cryptography for gray-level images by dithering techniques," *Pattern Recognition Letters*, vol. 24, no. 1-3, pp. 349-358, January 2003.

[6] M. Naor and A. Shamir, "Visual Cryptography," *Proc. EUROCRYPT'94, LNCS*, vol. 950, pp. 1-12, 1994.

[7] C.C Chang and J.C. Chuang, "An image intellectual property protection scheme for gray-level images using visual secret sharing strategy," *Pattern Recognition Letters*, vol. 23, no. 8, pp. 931-941, June 2002.

[8] R.A. Ulichney, "Dithering with blue noise," *Proceedings of the IEEE*, vol. 76, no. 1, pp. 56-79, January 1988.

[9] P.W. Wong and N.S. Memon "Image processing for halftones," *IEEE Sig. Proc. Mag.*, vol. 20, no. 4, pp. 59-70, July 2003.