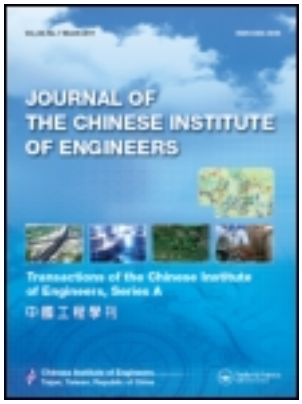


Journal of the Chinese Institute of Engineers
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcie20

Universal share for the sharing of multiple images

Wen-Pinn Fang [a b] & Ja-Chen Lin [c]

[a] Department of Computer Science , National Chiao Tung University , Hsinchu, Taiwan 300, R.O.C. Phone: 886-35721490 E-mail:

[b] Department of Computer Science and Information Engineering , Yuanpei University , Hsinchu, Taiwan 300, R.O.C.

[c] Department of Computer Science , National Chiao Tung University , Hsinchu, Taiwan 300, R.O.C.

Published online: 04 Mar 2011.

To cite this article: Wen-Pinn Fang & Ja-Chen Lin (2007) Universal share for the sharing of multiple images, Journal of the Chinese Institute of Engineers, 30:4, 753-757, DOI: 10.1080/02533839.2007.9671301

To link to this article: http://dx.doi.org/10.1080/02533839.2007.9671301

Short Paper

UNIVERSAL SHARE FOR THE SHARING OF MULTIPLE IMAGES

Wen-Pinn Fang* and Ja-Chen Lin

ABSTRACT

To share numerous grey-valued images (or numerous color-valued images), this study presents a system with a universal share. A company organizer can use this universal share to attend the recovery meeting of any shared image. No storage space is wasted; i.e. for each shared image, the total storage space occupied by all generated shares (including the universal share) is identical to the image size.

Key Words: image sharing, LSB hiding.

I. INTRODUCTION

In an (n, n) image sharing system (Thien and Lin, 2002; Thien and Lin, 2003a; and Wu *et al*., 2004), n shares $\{L_1, L_2, \cdots, L_n\}$ are created for a given image, e.g., Lena. The image can be revealed when all n shares are received, while less than n shares reveal nothing about the image. With sharing, nobody (even the company organizer) can view the image without attending a public meeting. Therefore, sharing is a safety process that is valuable in a company where no employee/investor alone should be trusted. Significantly, the original image can be discarded after the sharing; moreover, each of the n shares is $1/n$ of the size of the given image. Therefore, the sharing process does not waste storage space. To share another image, Monkey (which is grey-valued iff Lena is grey-valued), another n shares $\{M_1, M_2, \cdots, M_n\}$ are similarly created. Each employee/partner of a company can thus obtain a share from each image related to his job/investment. Consequently, if a company organizer obtains 1 share from each of the 100 important images shared, then he will have difficulty managing his 100 shares. Share management becomes increasingly difficult as the number of images rises. This study proposes the use of a "universal" share for a company's organizer by combining sharing and hiding (Lie and Chang, 1999; Maniccam and Bourbakis, 2004; Thien and Lin, 2003a; Wang *et al*., 2001; and Wu and Liu, 2003). The organizer only has to take the unique share (a single share with a compact size, see the top of Fig. 1 (c) and 2(c)) to attend the recovery meeting for any image. Although the method described here handles grey images, it can also be applied to color images by processing the three color components one by one. The color version was also found satisfactory, but its processing steps are not shown here to save space.

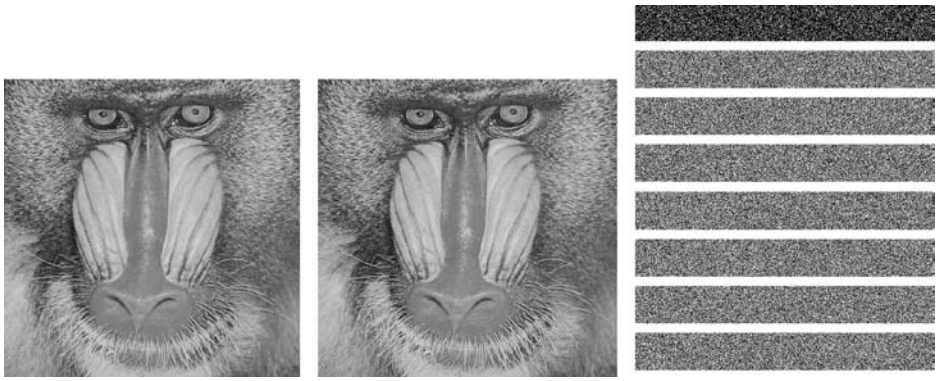
Below are reviewed some reported methods dealing with multiple secret images. The paper (Chan and Chang, 2005) is well-proved and mathematically sound; but it is not used here because (Chan and Chang, 2005) has no experiment data. Tsai *et al*. (2002) proposed an elegant method using visual cryptography and LSB hiding for multiple secret images. For $C(4,2) = 6$ secret images of size 200×200 each, Tsai *et al*. hid the corresponding $6 \times 200 \times 200$ bytes using only four stego images (each of size 600×600 and about 42.5dB in PSNR), of which any two can be combined to extract one of the six secret images. For our universal approach, assume that $n = 4$ shares are created for each image. The storage space before hiding the generated shares, is $200 \times 200 \times ((1/4) \times 1 + (3/4) \times 6) = 4.75 \times 200 \times 200$ bytes, where $(1/4) \times 1 = 1/n$ is for the universal share, and $(3/4) \times 6 = ((n - 1)/n) \times 6$ is for the non-universal shares of the six secret images. Our system thus saves more space than that of (Tsai *et al*., 2002) (the ratio is 4.75 : 6, before

*Corresponding author. (Tel: 886-35721490; Email: wpfang@cis.nctu.ed.tw)

The authors are with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C., and W. P. Fang is currently at Department of Computer Science and Information Engineering, Yuanpei University, Hsinchu, Taiwan 300, R.O.C.

(a) (b) (c)

Fig 1. A sharing result for $n = 8$. (a) initial image Lena; (b) modified image Lena*, which contains the image $\boldsymbol{U}$, from which all extracted a_i values are in the range 0-250 (see Eq. (1)); (c) the 8 shares that can recover (b) together; the top-most share in (c) is the universal share, which is identical to the image U.



(a) (b) (c)

Fig 2. Another sharing result for $n = 8$. (a) initial image Monkey; (b) the modified image Monkey*; (c) the 8 shares that can recover (b) together; the top-most share is identical to the top-most share in Fig. 1(c).

hiding). If the generated shares are also hidden, then the total space for the stego images is $4 \times 4.75 \times 200 \times 200 = 19 \times 200 \times 200$ bytes to obtain stego images of PSNRs much better than 42.5 dB; compared with $4 \times 600 \times 600 = 36 \times 200 \times 200$ bytes to obtain 42.5dB stego images in (Tsai *et al.*, 2002). However, the recovered secret images in (Tsai *et al.*, 2002) are without loss, while ours are lossy.

Feng *et al.* (2005) presented a gorgeous method for sharing multiple secret images based on summing up the variables used in the sharing polynomials. Their method uses, for instance, 5 shares $\{a, b, c, d, e\}$, and can reveal secret image 1 using $\{a, b, c\}$, and secret image 2 using $\{a, d, e\}$, and secret image 3 using some other combination of shares. The total size of the sharing result using Feng *et al.*'s method is typically 1-2 times the total size of the input secret images. That is, their method causes size expansion. By contrast, our method has a size reduction effect, because the total size of our sharing result is even smaller than the total size of the input secret images. More precisely, the size of our sharing result is only $[(1/n) \times 1 + (n-1)/n \times S]/S = (1-(1-S^{-1})/n) \times 100\%$ of the input, where S denotes the number of secret images, and n denotes the number of shares for each secret image. Therefore, our method has an advantage in economy of size. However, like Ref. (Tsai *et al.*, 2002), Ref. (Feng *et al.*, 2005)'s method can achieve lossless recovery of the secret images. As for the encoding or decoding speed, Ref. (Feng *et al.*, 2005) is slower than our method (because the degree of their sharing polynomial is one degree higher than ours), while Ref. (Tsai *et al.*, 2002) is faster than our method (because they use logic operations to achieve sharing, which are faster than the operations needed in the polynomial approach).

Unlike our method, Refs. (Tsai *et al.*, 2002) and (Feng *et al.*, 2005) have no super share (the share for company organizer). Therefore, our method can be utilized when the boss of a company wishes to control

every secret, while (Tsai *et al*., 2002) as well as (Feng *et al*., 2005) are suitable for team work in which every teammate is of equal importance.

II. THE METHOD

1. Embedding U in the LSB of the Secret Image

Let $p \times q$ denote the standard size of each image to be shared. The organizer randomly grabs or creates an extra image $\boldsymbol{U}$ of size $p \times q/n$ (all pixel values of $\boldsymbol{U}$ are less than 251, as they are used later in Eq. (1)). This image $\boldsymbol{U}$, whether noisy or not, has $8pq/n$ bits, which are embedded into the $p \times q$ image Lena by the least-significant-bits (LSB) replacement method (Lie and Chang,1999; Maniccam and Bourbakis, 2004; Thien and Lin, 2002; Thien and Lin, 2003a; Thien and Lin, 2003b; Wang *et al*., 2001). For instance, if $n \geq 8$, then Image $\boldsymbol{U}$ has at most $p \times q$ bits, so $\boldsymbol{U}$ can be hidden using the least-significant-bits (1 bit per pixel) of Lena, which has $p \times q$ pixels. If $4 \leq n \leq 7$, then $\boldsymbol{U}$ is hidden using the last two bits of Lena's pixels. After embedding, Lena becomes a distorted image called Lena*. Since only some less-important bits of Lena are replaced (assuming $n \geq 4$), the distortion is invisible when comparing Lena* and Lena.

2. Partitioning Each Sector

Decompose Lena* into non-overlapping sectors of n pixels each ($8n$ bits per sector). Then share the $8n$ bits of each sector among the n shares. To save paper length, we assume $n = 8$ below; other values of n are handled analogously.

(i). The LSBs of the $n = 8$ pixels of the sector form an 8-bit number, called a_0. Notably, $a_0 < 251$ by Sec. II.1.

(ii). The remaining $8 \times 7 = 56$ bits of the sector are then partitioned into another 7 numbers $\{a_1, \cdots, a_7\}$ of 8 bits each (see Fig. 3); i.e., $\{a_i = (a_{i1}, a_{i2}, \cdots, a_{i8}) | 1 \leq i \leq 7\}$. For $1 \leq i \leq 7$, Fig. 3 ensures that the most significant bit (MSB) a_{i1} of every $a_i = (a_{i1}, a_{i2}, \cdots, a_{i8})$ is the bit next to the LSB of a Lena* pixel. This property avoids visible damage to the image Lena* if the bit value of some a_{i1} is changed from 1 to 0 so that all a_i stay in the 0-250 range before utilizing Eq. (1) (as noted below Eq. (1)).

3. Sharing

We already have $\{a_0, a_1, \cdots, a_7\}$. Affix to each Share k, where $0 \leq k \leq n - 1 = 7$, a value

$$f(k)=(a_0 + a_1k + a_2k^2 + \cdots + a_{n-1}k^{n-1}) \bmod 251, \quad (1)$$

where 251 is the prime number suggested in (Thien and Lin, 2002). In Eq. (1), for recovery purposes, each a_i must be in the range 0-250. Therefore, some bits in the sector might need adjustment before (1) can be applied. The image after this minor adjustment is still called Lena*. Since each n-pixel sector only contributes a value $f(k)$ to Share k, each share is n times smaller than the image Lena* (and hence Lena). The total size of the n shares is therefore identical to that of Lena; hence, no storage space is wasted.

4. Using the Universal Share $\boldsymbol{U}$

The company organizer keeps Share 0, whose value is $f(0) = a_0$ for each sector. Since the set $\{a_0\}$ is formed of the LSB of image Lena*, it is also formed of image $\boldsymbol{U}$, because $\boldsymbol{U}$ is embedded in Lena's LSB to obtain Lena* in Sec. II.1. Hence, Share 0 is identical to $\boldsymbol{U}$, the image created earlier by the organizer. This statement is true even if Lena is replaced by any other image (e.g., Monkey). Share 0, i.e. image $\boldsymbol{U}$, is thus the desired universal share.

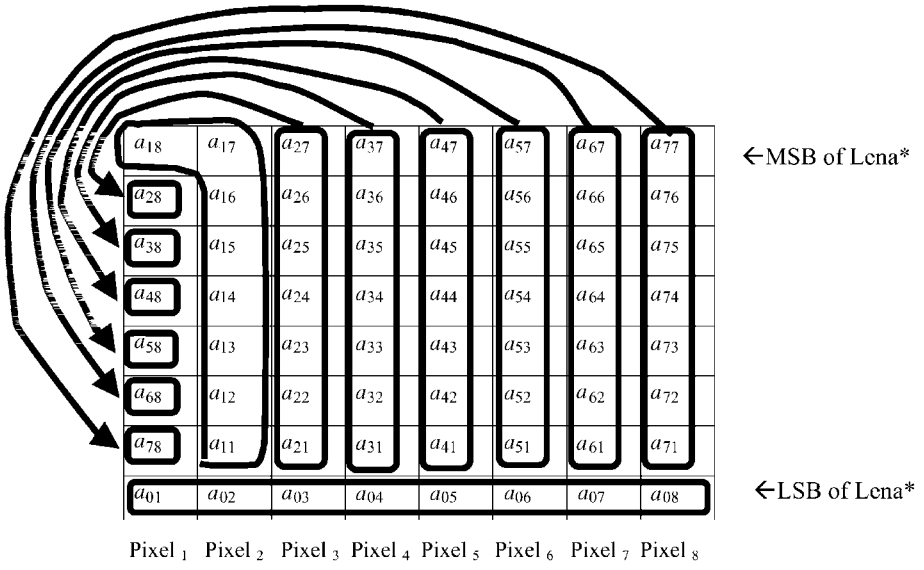
During the recovery phase, when all n shares are collected, recover the values $\{a_0, \cdots, a_{n-1}\}$ from $\{f(0), \cdots, f(n-1)\}$ using Lagrangian interpolation polynomials. This is an ordinary, routine procedure used in the sharing field, as described in Refs. (Wu *et al*., 2004; Feng *et al*., 2005) The modified image, (regardless OR irrespective) of whether it is Lena* or Monkey*, can thus be recovered sector-by-sector.

III. EXPERIMENTS

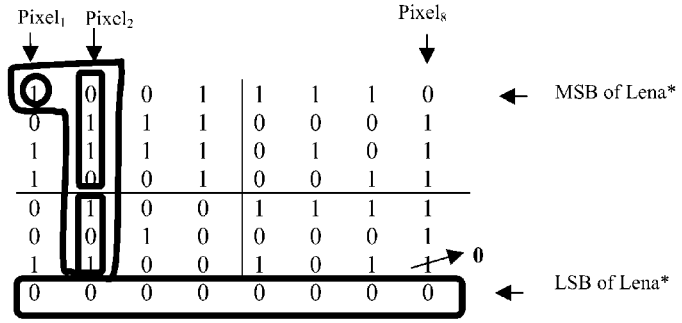
Assume that $n = 8$. Fig. 1(a) depicts the input image Lena, and the company organizer arbitrarily creates his own share (the top-most noisy image $\boldsymbol{U}$ in Fig 1(c), whose pixel values are all below 251). Fig. 1(b) is the modified image Lena*, which not only hides the entire image $\boldsymbol{U}$ in its LSB, but also has the property that all a_i extracted from it are in the range 0-250 (see the explanations in Sec. II.2 and II.3). Then, Lena* is shared. Share 0 is the given image $\boldsymbol{U}$, and the remaining $n - 1 = 7$ shares are generated using $k = 1, 2, \cdots, n - 1$ in Eq. (1). All 8 shares are shown in Fig. 1(c). These 8 shares can together recover the Lena* displayed in Fig. 1(b). Fig. 2 shows another experiment in which Lena is replaced by Monkey. Its Share 0 (the top-most noisy image in 2(c)) is identical to Share 0 in Fig. 1(c), because both are identical to $\boldsymbol{U}$.

IV. SUMMARY

In summary, our sharing method is space-saving and with a convenient universal share. The advantages are achieved by tolerating an invisible distortion in the



(a)



Input (gray values):

Pixel$_1$=1011-0010

Pixel$_2$=0110-1010

Pixel$_8$=0111-1110

Output:

a_0 = 0000 - 0000

a_1 = 101 - 0110 - 1

a_2 = 010 - 0110 - 0

a_3 = 000 - 1111 - 1

a_4 = 101 - 0001 - 1

a_5 = 001 - 0101 - 0

a_6 = 101 - 1001 - 0

a_7 = 111 - 1110 - 1

(Reduce a_7 to 011-1110-1 so that $a_i < 251 \; \forall \, i$. This means pixel $_8$ is now 0111-1100, a little different from original 0111-1110, but the gray value change is quite small.)

(b)

Fig 3 Partition of a sector of Lena* (see Section II.2). (a) the grouping of bits, (b) an example illustrating (a).

recovered images. For instance, the recovered images in Figs. 1(b) and 2(b) are 52.5 dB in PSNR, when being compared with the original images in Figs. 1(a) and 2(a). As a remark, our program can be run repeatedly to handle any number of secret images, for instance, 1000 secret images, without the need for reprogramming. Additionally, the universal share ***U*** can be non-noisy, because ***U*** can be any kind of image, including ordinary photos. Hence, only non-universal shares, which always look noisy, need post-processing hiding.

ACKNOWLEDGMENTS

The anonymous reviewers are thanked for their

valuable suggestions. The work was supported by NSC project NSC 95-2221-E-009-256.

REFERENCES

Chan, C. W., and Chang, C. C., 2005, "A Scheme for Threshold Multi-Secret Sharing," *Applied Mathematics and Computation*, Vol. 166, No. 1, pp. 1-14.

Feng, J. B., Wu, H. C., Tsai, C. S., and Chu, Y. P., 2005, "A New Multi-Secret Images Sharing Scheme Using Largrange's Interpolation," *Journal Systems and Software*, Vol. 76, No. 4, pp. 327-339.

Lie, W. N., and Chang, L. C., 1999, "Data Hiding in Images with Adaptive Numbers of Least Significant Bits Based on the Human Visual System," *International Conference on Image Processing*, Kobe, Japan, Oct., Vol. 4, pp. 286-290.

Maniccam, S. S., and Bourbakis, N., 2004, "Lossless Compression and Information Hiding in Images", *Pattern Recognition*, Vol. 37, No. 3, pp. 475-486.

Thien, C. C., and Lin, J. C., 2002, "Secret Image Sharing," *Computers & Graphics*, Vol. 26, No. 5, pp. 765-770.

Thien, C. C., and Lin, J. C., 2003a, "An Image-Sharing Method with User-Friendly Shadow Images," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 12, pp. 1161-1169.

Thien, C. C., and Lin, J. C., 2003b, "A Simple and High-Hiding Capacity Method for Hiding Digit-by-Digit Data in Images Based on Modulus Function," *Pattern Recognition*, Vol. 36, No. 12, pp. 2875-2881.

Tsai, C. S., Chang, C. C., and Chen, T. S., 2002, "Sharing Multiple Secrets in Digital Images," *Journal of Systems and Software*, Vol. 64, No. 2, pp. 163-170.

Wang, R. Z., Lin, C. F., and Lin, J. C., 2001, "Image Hiding by Optimal LSB Substitution and Genetic Algorithm," *Pattern Recognition*, Vol. 34, No. 3, pp. 671-683.

Wu, M., and Liu, B., 2003, "Data Hiding in Image and Video: I. Fundamental Issues and Solutions," *IEEE Transactions on Image Processing*, Vol. 12, No. 6, pp. 685- 695.

Wu, Y. S., Thien, C. C., and Lin, J. C., 2004, "Sharing and Hiding Secret Images with Size Constraint," *Pattern Recognition*, Vol. 37, No. 7, pp. 1377-1385.

Manuscript Received: Oct. 13, 2005
Revision Received: Nov. 03, 2006
and Accepted: Dec. 15, 2006