

Lossless Data Hiding for Binary Document Images Using n -Pairs Pattern

Cheonshik Kim^{1(✉)}, Jinsuk Baek², and Paul S. Fisher²

¹ Department of Digital Media Engineering, Anyang University,
Anyang-si, Gyeonggi-do, Korea
mipsan@paran.com

² Department of Computer Science, Winston-Salem State University,
Winston-Salem, NC, USA
{baekj, fisherp}@wssu.edu

Abstract. Lossless data embedding theory has entered a new era for data hiding and information security. In a lossless scheme, the original data and the embedded data should be completely recoverable. Our n -pairs pattern method is a significant advance in lossless data hiding schemes. This paper shows that the proposed n -pairs pattern method can achieve greater embedding capacity while keeping distortion at the same level as the PS-K method (Pattern Substitution by pseudo random number generator to produce a key K). The performance of the n -pairs pattern method is thus shown to be better than the performance of PS-K.

Keywords: Binary document · Data hiding · Lossless · n -Pair pattern

1 Introduction

Data hiding techniques [1–5] can be used in various applications such as annotations, communications, and copyrights [6]. For most images, the modification of the pixel values is not detectable by the human visual system as long as the modification is extremely small. However, a binary image is very sensitive and is easily affected by weak pixel flipping [6–8]. Therefore, many researchers have focused on developing reversible data hiding methods related to binary images [9–15]. While existing methods are efficient, they each have limitations.

Ho et al. [6] propose a reversible data hiding method using pattern substitution that has a reasonable performance. The method uses 4-bit vector patterns as the binary sequences for the difference images. *Tseng et al.* [8] use matrix encoding to hide data in binary images. The input binary image is divided into 3×3 blocks (or larger). The flipping priorities of pixels in a 3×3 block are then computed and those with the lowest scores are changed to embed the data. In *Chen et al.* [10], an input binary image is divided into blocks of 8×8 pixels. The numbers of black and white pixels in each block are then altered by embedding data bits 1 and 0. A data bit 1 is embedded if the percentage of white pixels is

less than a given threshold. This method is robust against noise if the difference between the thresholds for the data bits 1 and 0 is sufficiently large; however, this technique also decreases the quality of the marked document.

In this paper, we propose a lossless data hiding scheme for binary document images using the n -pairs pattern method. Although the proposed scheme is very simple, it has a high embedding capacity and produces an image of reasonable quality. Our new scheme makes it possible to control the embedding capacity and quality of an image, so that the stego-image has a higher quality than those produced by traditional approaches under the same embedding scale.

The remainder of this paper is organized as follows. Section 2 describes the histogram modification method for reversible data hiding. Section 3 presents our new n -pairs pattern method in detail and our proposed data hiding scheme for binary document images. Section 4 discusses our experimental results, and Sect. 5 presents our conclusions.

2 Histogram Modification Method

Many researchers have proposed reversible data hiding schemes that use histogram pairs of gray-scale images. In the histogram we first find a *zero point*, and then a *peak point*. A *zero point* corresponds to the gray-scale value that no pixel in the given image has. A *peak point* corresponds to a gray-scale value that the largest number of pixels in the image has. The purpose of finding a *peak point* is to increase the embedding capacity as much as possible because, as we show below, the number of bits that can be embedded into an image by this algorithm is equal to the number of pixels that are associated with the *peak point* ([9]).

In the following discussion we will refer to the maximum and minimum points. For an $M \times N$ image, each pixel has a gray-scale value $x \in [0, 255]$:

Step 1: Generate its histogram $H(x)$.

Step 2: In the histogram $H(x)$, find the maximum point $h(\alpha)$, $\alpha \in [0, 255]$ and the minimum point $h(\beta)$, $\beta \in [0, 255]$.

Step 3: If the minimum point $h(\beta) \geq 0$, then record the coordinates (i, j) of those pixels and the pixel gray-scale value β as overhead bookkeeping information (referred to as *overhead information* for short), and set $h(\beta)$ to 0.

Step 4: Without loss of generality, assume that $\alpha < \beta$. Move the whole part of the histogram $H(x)$ with $x \in (\alpha, \beta)$ to the right by 1 unit. The result is that all of the pixel gray-scale values in the interval (α, β) are incremented by 1.

Step 5: Scan the image until meeting the pixel whose gray-scale value is α ; check the to-be-embedded bit. If the to-be-embedded bit is 1, the pixel gray-scale value is changed to $\alpha + 1$. If the bit is 0, the pixel value remains α .

In this way, the actual data embedding capacity, C , is calculated as follows:

$$C = h(\alpha) - O, \quad (1)$$

where O denotes the amount of data used to represent the overhead information. We refer to O as *pure payload* in this paper. Clearly, if the required payload is greater than the actual capacity, more pairs of maximum and minimum points need to be used. The embedding algorithm with multiple pairs of maximum and minimum points is presented in next Section.

3 Our Proposed n -Pair Pattern Scheme

3.1 n -Pair Pattern Basic

Generally, data hiding with reversibility using histogram pairs ([9–12, 14, 15]) has the advantage that the original image is recoverable, but the hiding capacity of this method is restricted. To obtain reversible data hiding with a greater hiding capacity, we propose an n -pair pattern scheme. In the proposed scheme, the relationship between neighboring pixels is explored.

We have two reasons for proposing an n -pair pattern. First, the reversible data hiding method in [9, 13] is based on gray-scale images. Thus, these techniques cannot be directly applied to binary images for data hiding. Our proposed n -pair pattern scheme extends histogram pairs for binary data hiding. Second, our proposal for reversible binary data hiding does not require a location map. Because it is possible to remove location map by adjustment number of n -pair pattern. Next, we sketched out our proposed scheme. First of all, each group of n -pixel in an image transform into a decimal number. These decimal numbers represent gray-scale colors such that 0 is black, $2n - 1$ is white, and the values in between are various shades of gray. Based on these decimal numbers, an n -pairs pattern is generated.

Figure 1 shows a histogram generated from decimal numbers of three consecutive bits. Position 0 and 7 are all white and black, so they are not fit to

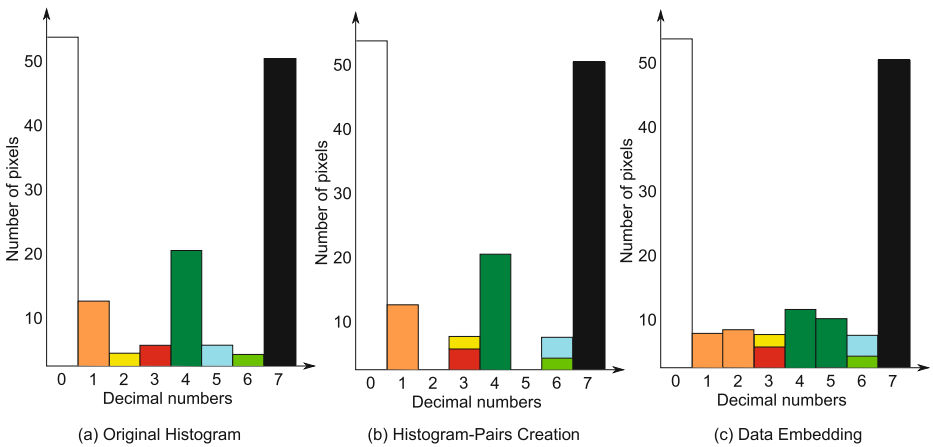


Fig. 1. Concept of data embedding using histogram modification

conceal to data hiding. Note that the number pixels at position 1 is dominating in the range of position 1 and 3. Therefore, the position 2 is emptied. Similarly, the position 5 is cleared. Of course, the position 2 merge with position 3. The position 2 should be marked using the location map.

Similarly, another location map is necessary for the merged position 6. After securing vacant positions, reversible data embedding starts by distributing the numbers in the positions 2 and 5 according to the message to be hidden. The secret message string includes the location map. The decoding procedure is very simple. In the beginning, the location map and the secret message string are recovered from position 2 and 3, 5 and 6. Figure 1(c) is changed to Fig. 1(b) after recovering positions and obtaining the hidden message. Based on the recovered location map, the contents of the merged positions can be exactly split and reconstructed as Fig. 1(a).

$$\delta = h_1, h_2, \dots, h_n || m_1, m_2, \dots, m_n. \quad (2)$$

These procedures are called n -pair pattern methods. We assume that δ is a payload with the included messages $h_1 \dots h_n$ and the location maps $m_1 \dots m_n$ in Eq. (2). δ is composed of the bits 0 and 1, as specified in Eq. (3). The variable hh in Eq. (4) is a decimal value that is calculated from the binary bitmap image, where n is the size of payload, p is the size of bitmap pairs and dec is a function that transforms a binary value into a decimal value. We assume that a vector of decimal values is considered to be a discrete gray-scale image hh .

$$\delta_i \in \{0, 1\}, 1 \leq i \leq n, \text{ where } n = \text{size}(\text{payload}), \quad (3)$$

$$hh_k^{\frac{(M \times N)}{p}} = \sum_{i=1}^{M \times N} \sum_{j=1}^{M \times N} dec(BI(j : j + p)), \quad (4)$$

To hide data, we find the maximum pattern α and minimum pattern β from the hh (n -pair pattern), using Eqs. (5) and (6). Equation (7) shows the embedding method, that is, how messages are embedded into hh_i . If hh_i is an α pattern and the message bit is 0, then there is no change in hh_i ; otherwise, add $(\beta - \alpha)$ to hh_i (see Eq. (7)).

As can be seen, hh is a decimal value vector which is hiding a secret message. Therefore, we need to transform hh into BI (Binary Image), which is reconstructed as the stego image and which includes the secret message. As can be seen in Fig. 1, our proposed scheme is very simple, and in these examples we have introduced a reversible hiding scheme.

$$\alpha = \sum_{i=1}^n MAX(hh_i), \quad (5)$$

where n is the number of pairs.

$$\beta = \sum_{i=1}^n MIN(hh_i), \quad (6)$$

where n is the number of pair.

$$hh'_i = \begin{cases} hh_i & \text{if } hh_i = \alpha \text{ and } \delta_i = 0 \\ hh_i + (\beta - \alpha) & \text{if } hh_i = \alpha \text{ and } \delta_i = 1 \end{cases} \quad (7)$$

3.2 Embedding Procedure

The flowchart in Fig. 2 below depicts the procedure of embedding messages into binary document images.

Input. Original binary document image BI of size $M \times N$ (in pixels) and the secret data, $\delta = [h_1, h_2, , h_{size(message)} || m_1, m_2, , m_n], cnt = |\delta|$.

Output. Stego image SI , the maximum point α , and the minimum point β .

Step 1: Divide BI into a group of n -bits with non-overlap. All of the sub-groups are then transformed into one decimal value, x , and these are stored in vector hh . The histogram H is created from the vector hh , where $i \in \{0, 1, \dots, 2n - 1\}$ and n is the number of pixels used to transform a decimal number, i.e., $n = [5, \dots, 8]$. In H , obtain maximum point α and minimum point β . If the number of point $\beta > 0$, record the coordinate of those pixels and the pixel value as overhead keeping information.

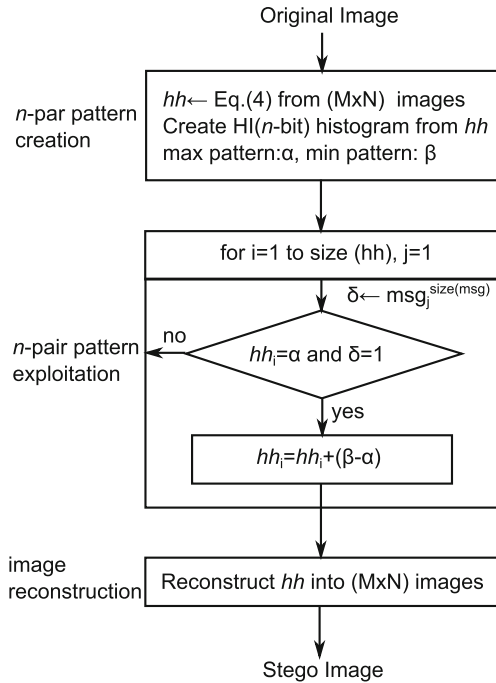


Fig. 2. Flowchart for the embedding algorithm based on an n -pair pair pattern

- Step 2:** Scan the hh until meeting the pixel whose virtual gray-scale value is α ; check the to-be-embedded bit. If the to-be-embedded bit is '1', the virtual pixel gray-scale value is changed to $\alpha + (\beta - \alpha)$. If the bit is '0', the pixel value remains α .
- Step 3:** Let $cnt = cnt - 1$. if $cnt = 0$ go to Step 4; otherwise, go to Step 2 to continue the embedding processes.
- Step 4:** When the process of embedding secret messages has been completed, convert hh into an $M \times N$ image SI , which is the stego image.

3.3 Extracting Procedure

Figure 3 below shows the flowchart for extracting messages from binary document images. We first create virtual gray-scale images and then extract messages from these images. The extraction procedure is as follows:

- Input.** Stego image SI , the maximum point α , the minimum point β , and the length of the payload $cnt = |p|$.
- Output.** Original binary document image BI and the secret data p .
- Step 1:** Divide SI into a group of n -bits with non-overlap. All of the sub-groups are then transformed into decimal values, and these are stored in vector hh .

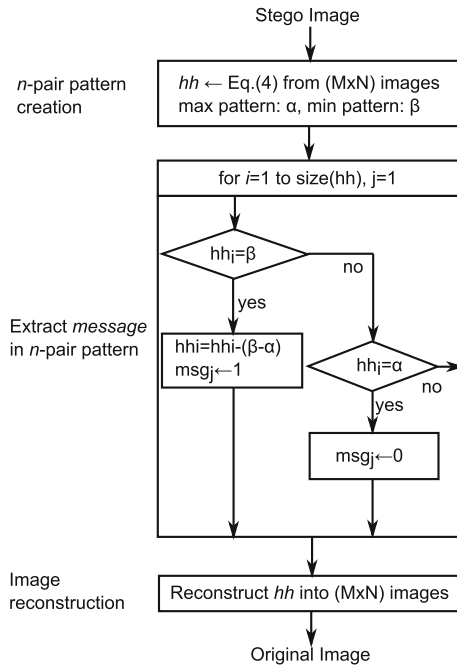


Fig. 3. Flowchart for the extraction algorithm based on the n -pair pattern scheme

Step 2: Scan the decimal values from hh_i in the same order as in the embedding phase. If the scanned value is a β pattern, then a hidden bit can be recognized by 1 in the hh . $\delta = \delta||'1'$. After that, a β pattern in hh_i can be the original pattern as $hh_i = hh_i - (\beta - \alpha)$. If the scanned value is an α pattern, then extract 0 from the virtual image. $\delta = \delta||'0'$.

Step 3: $cnt = cnt - 1$; Repeat Step 2 until $cnt = 0$.

Step 4: Reconstruct the binary image to transform hh into BI .

3.4 Computational Complexity

Our proposed scheme is in the spatial domain. The methods computational complexity is low since it does not need to perform DCT (Discrete Cosine Transform) or DWT (Discrete Wavelet Transform,). The method requires generating the n -pair pattern for a cover image, determining maximum and minimum points α and β based on the n -pair pattern, hiding information, and performing the inverse transformation in the spatial domain. Thus, the execution time needed for the proposed scheme is quite small. Assume that the block size is $P \times Q$ and that there are k blocks in a cover image. Our proposed scheme only needs to scan each block five times during the hiding phase. Hence the computational complexity for the scanning is $O(5PQ)$ and as a result, the overall computational complexity is $O(5PQk)$.

4 Experimental Results

In binary document images, a number of flipped pixels can conceal messages. We need to find an empty pair pattern in the image. In the case of the 4-pair pattern, it is difficult to find the empty pair pattern. As the number n increases, some local peaks become more dominant compared to neighboring pixels. When n is large, there appear to be a large number of empty pixels next to the local peak pixels. By choosing other peaks or increasing the size of the n -pair pattern, we can control the image quality. Additionally, it is possible to increase the embedding capacity by increasing the size of patterns for flipping.

Table 1 shows the number of bits that can be embedded into the English-1 image (Fig. 4(a)). In the 5-pair pattern method, position 1 (00000_b in binary representation) has a population of 2,539 bits, while position 10 (01010_b) has 0. Therefore, the location map has to mark 0 bits. As n increases, the capacity for data hiding in the binary document image will decrease. If there are *high peaks* with relatively high neighboring pixels, a large location map is required. If there are relatively *high peaks* with very low or empty neighboring pixels, the location map size is negligible or zero.

Table 2 compares the data-hiding capacities of PS-K [6] and our proposed scheme, using the 5-bit pair method. PS-K has a case where the location map is relatively large, while the location map for our proposed scheme is no or few (with $n = 5$).

I believe in God, the Father Almighty, Creator of Heaven and earth. I believe in Jesus Christ, His only Son, our Lord. He was conceived by the power of the Holy Spirit and was born of the Virgin Mary. He suffered under Pontius Pilate, was crucified, died, and was buried. He descended to the dead. On the third day He rose again. He ascended into Heaven, and is seated at the right hand of the Father. He will come again to judge the living and the dead. I believe in the Holy Spirit, the Holy Catholic Church, the Communion of Saints, the forgiveness of sins, the resurrection of the body, and life everlasting.

a. English 1 Original Image (355x510)

당신의 손끝만 스쳐도 소리 없이 열릴 돌문이 있습니다. 못사람이 조바심치나 굳이 닫힌 이 돌문 안에는, 석벽 난간 열두 층계 위에 이제 검푸른 이끼가 앉았습니다. 당신이 오는 날까지는, 길이 꺼지지 않을 촛불 한 자루도 간직하였습니다. 이는 당신의 그리운 얼굴이 이 희미한 불 앞에 어리울 때까지는, 천 년이 지나도 눈 감지 않을 저희 슬픈 영혼의 모습입니다. 길숙한 속눈썹에 향시 어리운 이 두어 방울 이슬은 무엇입니까?

c. Korea 1 Original Image (334x446)

Remember, O most gracious Virgin Mary, that never was it known that anyone who fled to thy protection, implored thy help, and sought thy intercession, was left unaided. Inspired with this confidence, I fly unto thee, O Virgin of virgins, my Mother; to thee I come; before thee I stand sinful and sorrowful. O Mother of the Word Incarnate, despise not my petitions, but, in thy mercy, hears and answers me. Amen.

b. English 2 Original Image (282x459)

당신의 남긴 푸른 도포 자락으로 이 눈썹을 씻으렵니까? 두 불은 옛날 그대로 복사꽃빛이지만, 안숨에 절로 입술이 푸르러 감을 어찌입니까? 몇만 리 굽이치는 강물을 건너와 당신의 따스 손끝이 겨의 목덜미를 어루만질 때, 그 때야 겨는 자취도 없이 안 증 티끌로 사라지게됩니다. 어두운 밤 아늘 여공 중천에 바람처럼 사라지는 겨의 옷자락은, 눈을 어린 눈이 아니고는 보이지 못아오리다. 여기 돌문이 있습니다. 원안도 사무질 양이면 자극만 생생에 열리지 않는 돌문이 있습니다. 당신이 오셔서 다시 천 년토록 앉아 기다리라고, 슬픈 비바람에 넘어 가는 돌문이 있습니다.

d. Korea 2 Original Image (446x449)

Fig. 4. Original document images used for experiments

When there is no empty pair pattern in an image, we use the minimum pattern for data hiding. In this case, we don't need to record the original minimum pattern in the location map. For the 1-bit payload, 20 bits for location map are required. The set of hidden bits for the proposed method is large compared to the size needed for PS-K.

Table 1. Comparison of hiding capacity and quality in a 355 × 510 image (Fig. 4(a)) with different size patterns

<i>n</i> -bits	Payload	Location map	Hidden bits
5	2539	0	2539
6	1766	0	1766
7	1228	0	1228
8	882	0	882

Table 2. Comparison of PS-K and our proposed method

Images	PS-K			Proposed Method		
	Payload	Location Map	Hidden Bits	Payload	Location Map	Hidden Bits
English 1 (355 x 510)	2076	13 x 20 bit	1816	2539	0	2539
English 2 (282 x 459)	1160	17 x 20 bit	820	832	0	832
Korea 1 (344 x 446)	1139	22 x 20 bit	699	839	0	839
Korea 2 (446 x 449)	2073	8 x 20 bit	1913	2393	0	2393

I believe in God, the Father Almighty, Creator of Heaven and earth. I believe in Jesus Christ, His only Son, our Lord. He was conceived by the power of the Holy Spirit and was born of the Virgin Mary. He suffered under Pontius Pilate, was crucified, died, and was buried. He descended to the dead. On the third day He rose again. He ascended into Heaven, and is seated at the right hand of the Father. He will come again to judge the living and the dead. I believe in the Holy Spirit, the Holy Catholic Church, the Communion of Saints, the forgiveness of sins, the resurrection of the body, and life everlasting.

a. English 1 Stego Image - 1,000 bits

당신의 손끝만 스쳐도 소리 없이 열릴 돌문이 있습니다. 못사람이 조바심치나 글이 달린 이 돌문 안에는, 석력 난간 열두 층계 위에 이제 검투를 이끼가 앉았습니다. 당신이 오는 날까지는, 길이 꺼지지 않을 꽃불 한 자루도 잔직하였습니다. 이는 당신의 그리는 얼굴이 이 희미한 불 앞에 어리를 때까지는, 천 년이 지나도 눈 감지 않을 저희 슬픈 영혼의 모습입니다. 길숙한 속눈썹에 향시 어리운 이 두어 방울 이슬은 두었었습니까?

c. Korea 1 Stego Image - 800 bits

Remember, O most gracious Virgin Mary, that never was it known that anyone who fled to thy protection, implored thy help, and sought thy intercession, was left unaided. Inspired with this confidence, I fly unto thee, O Virgin of virgins, my Mother; to thee I come; before thee I stand sinful and sorrowful. O Mother of the Word Incarnate, despise not my petitions, but, in thy mercy, hears and answers me. Amen.

b. English 2 Stego Image - 800 bits

당신의 남긴 푸른 도포 자락으로 이 눈썹을 씻으렵니까? 두 불은 옛날 그대로 복사꽃빛이지만, 안숨에 짙로 입술이 푸르려 감을 어찌하십니까? 몇만 리 굽이치는 강물을 건너와 당신의 따스 손길이 거의 목덜미를 어루만질 때, 그 때야 겨는 자취도 없이 안 숨 티끌로 사라져갑니다. 어두운 밤 아늘 어금 증편에 바람처럼 사라지는 거의 옷자락은, 눈을 어린 눈이 아니고는 보이지 못아오리다. 여기 돌문이 있습니다. 원안도 사무질 양이면 저극한 쟁쟁에 열리지 않는 돌문이 있습니다. 당신이 오셔서 다시 천 년토록 앉아 기다리라고, 슬픈 비바람에 날아 가는 돌문이 있습니다.

d. Korea2 Stego Image - 1,000 bits

Fig. 5. Stego document images including secret messages

According to [6], the maximum data hiding capacity (MDHC) can be calculated as:

$$MDHC \leq \frac{(M \times N)}{4}, \quad (8)$$

where M is the width, and N is the height of a binary image. On the other hand, our proposed scheme can be computed as Eq. (9).

$$MDHC \leq \frac{(M \times N)}{n}, \quad (9)$$

where n is $\{3, 4, \dots, 8\}$. Therefore, n -pair pattern show higher embedding capacity compared to PS-K.

Figure 5 shows the stego document images including secret data as a result of experiments. In fact, it is difficult to find many vestiges from the stego images since our scheme shows a similar level of PSNR with PS-K method. More importantly, it can be possible to completely recover original images from the stego images.

5 Conclusions

In this paper we proposed the n -pair pattern method as a scheme of data hiding for binary document images, and we used this method to annotation or copyright for binary document images. Our proposed data hiding scheme has the advantage of having a high data hiding capacity and reversibility, which is achieved with low deterioration of visual quality. Moreover, the proposed scheme is easy to implement with a simple efficient algorithm. The n -pair pattern method does not require the use of the original cover image to carry out the decoding process. In near future, we will focus on more improving the quality of the binary document images.

Acknowledgements. This research was supported by the Basic Science Research Program Through the National Research Foundation of Korea (NRF) by the Ministry of Education, Science and Technology (20120192).

References

1. Kim, H.J., Kim, C., Choi, Y., Wang, S., Zhang, X.: Improved modification direction methods. *Comput. Math. Appl.* **60**(2), 319–325 (2010)
2. Kim, C., Shin, D., Shin, D., Zhang, X.: Improved steganographic embedding exploiting modification direction in multimedia communications. *Commun. Comput. Inf. Sci.* **186**, 130–138 (2011)
3. Kim, C.: Data hiding by an improved exploiting modification direction. *Multimedia Tools Appl.* **69**(3), 569–584 (2014)
4. Kim, C., Yang, C.N.: Improving data hiding capacity based on hamming code. In: Park, J.J., Zomaya, A., Jeong, H.-Y., Obaidat, M. (eds.) *Frontier and Innovation in Future Computing and Communications*. LNEE, pp. 697–706. Springer, The Netherlands (2014)

5. Baek, J., Kim, C., Fisher, P., Chao, H.: (N, 1) Steganography approach for secret sharing with digital images. In: Proceedings of 2010 IEEE International Conference on Wireless Communications, Networking and Information Security, pp. 325–329 (2010)
6. Ho, Y.A., Chan, Y.K., Wu, H.C., Chu, Y.P.: High-capacity reversible data hiding in binary images using pattern substitution. *Comput. Stand. Interfaces* **31**, 787–794 (2009)
7. Tsai, C.L., Chiang, H.F., Fan, K.C., Chung, C.D.: Reversible data hiding and lossless reconstruction of binary images using pair-wise logical computation mechanism. *Pattern Recogn.* **38**(11), 1993–2006 (2005)
8. Tseng, Y.C., Chen, Y.Y., Pan, H.K.: A secure data hiding scheme for binary images. *IEEE Trans. Commun.* **50**(8), 1227–1231 (2002)
9. Ni, Z., Shi, Y.Q., Asari, N., Su, W.: Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **16**(3), 354–362 (2006)
10. Chen, M., Wong, E.K., Memon, N., Adams, S.: Recent developments in document image watermarking and data hiding. In: Proceedings of the SPIE Conference 4518: Multimedia Systems and Applications IV, pp. 166–176 (2001)
11. Lee, S.-K., Suh, Y.-H., Ho, Y.-S.: Lossless data hiding based on histogram modification of difference images. In: Aizawa, K., Nakamura, Y., Satoh, S. (eds.) PCM 2004. LNCS, vol. 3333, pp. 340–347. Springer, Heidelberg (2004)
12. Lin, C.C., Tai, W.L., Chang, C.C.: Multilevel reversible data hiding based on histogram modification of difference images. *Pattern Recogn.* **41**, 3582–3591 (2008)
13. Tian, J.: Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **13**(8), 890–896 (2003)
14. Tsai, P., Hu, Y.C., Yeh, H.L.: Reversible image hiding scheme using predictive coding and histogram shifting. *Signal Process.* **89**, 1129–1143 (2009)
15. Xuan, G., Shi, Y.Q., Ni, Z., Chai, P., Cui, X., Tong, X.: Reversible data hiding for JPEG images based on histogram pairs. In: Kamel, M.S., Campilho, A. (eds.) ICIAR 2007. LNCS, vol. 4633, pp. 715–727. Springer, Heidelberg (2007)