Lossless Data Hiding Scheme Using Adjacent Pixel Difference Based on Scan Path

Xianting Zeng[a,b]
[a]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[b]College of Information Engineering, China Jiliang University, Hangzhou, China
Email: mico@cjlu.edu.cn

Lingdi Ping, Zhuo Li
College of Computer Science and Technology, Zhejiang University, Hangzhou, China
Email: ldping@cs.zju.edu.cn, lizhuo84@gmail.com

***Abstract*—This paper presents a reversible data hiding scheme. The proposed scheme is based on the difference histogram shifting to spare space for data hiding. Nine basic scan paths are defined, and this means all-directional adjacent pixel differences can be obtained. Due to the fact that the grayscale values of adjacent pixels are close to each other, the all-directional adjacent pixel difference histogram contains a large number of points with equal values. Hence, more data can be embedded into the cover image than previous works based on histogram shifting. Furthermore, multi-layer embedding is used to increase the hiding capacity. In each embedding process, we can embed a large number of data into the cover image by choosing the best scan path and the optimized pixel difference. As experimental results have shown, the cover images are able to embed secret data at an average 12.5% of the size of the original images while all the PSNR values of the stego images remain larger than 30 dB.**

***Index Terms*—Lossless data hiding, Histogram shifting, Scan path, Multi-layer embedding, Pixel difference**

I. INTRODUCTION

Data hiding is a commonly used technique that hides information into digital media imperceptibly, such as images, videos, audios, etc. The information can serve as authentication codes, annotation, or secret data depending on the purpose of the application itself. In most cases of data hiding, the cover media may suffer permanent distortion due to data embedding and can not be inverted back to the original one. However, in some applications, such as medical image system, law enforcement and military imagery, it is desired to recover the original media with no distortion for some legal considerations or rare media themselves after data extraction. Therefore, a distortion-free, reversible, lossless data hiding technique is required.

Recently, some lossless data hiding algorithms have been reported in the literature. Honsinger et al. [1] in their patent used modulo-256 addition to embed the hash value of the original image for authentication. Although they can prevent overflow/underflow and achieve reversibility by using modulo-256 addition, their algorithm may cause salt-and-pepper artifacts and hinder watermark retrieval due to the many wrapped around pixel intensities. A very different approach was proposed by De Vleeschouwer et al. [2] based on the circular interpretation of the bijective transformations of the image histograms to reduce the salt-and-pepper visual artifacts found in [1].

The other category of approaches, such as Fridrich et al.'s [3,4] and Kalker et al.'s [5] schemes, involves methods to losslessly compress a set of selected features from an image and embed the payload in the space saved due to the compression. Celik et al. [6, 7] improved Fridrich et al.'s technique and proposed the generalized-LSB (G-LSB) scheme. Xuan [8] proposed a scheme based on integer wavelet transform (IWT) and created more space in high frequency sub-bands, which can provide payload two to five times as large as that in Fridrich et al.'s scheme.

Tian [9] proposed a high capacity reversible data embedding technique that is called difference-expansion (DE) embedding. The DE technique is able to embed significantly larger amounts of data than the other earlier approaches. The distortion introduced is also significantly less for comparable payload sizes. Alattar [10] improved Tian's scheme by introducing the difference expansion of a vector to obtain more extra space. Afterwards, many techniques, such as [11-13, 15-17], extended Tian's scheme and increased the embedding capacity and kept the distortion low. Kamstra et al. [14] also extended Tian's algorithm by using the information in the low-pass band to find suitable expandable differences in the high-pass band.

Another novel reversible watermarking technique, sometimes referred to as histogram-shifting technique, was proposed by Ni et al. [18]. After finding a peak point of image histogram and histogram shifting, the peak point of the histogram is selected to embed data. However, their payload is quite limited because few images contain a large number of pixels with equal gray values. Recently, Ni et al.'s technique has been extended by Chang et al. [19]. By using the pixel difference instead of simple pixel

value, they can obtain the higher peak point to embed a large amount of message.

In this paper, a new lossless data hiding scheme using adjacent pixel difference based on scan path is proposed. The basic concept of difference histogram shifting of the proposed scheme is based on the technique proposed in Ni et al.'s scheme [18]. We present an efficient extension of the histogram modification technique by calculating the differences between adjacent pixels in different scan paths instead of the same sequential order each time in Chang et al.'s scheme [19]. In this way, we can obtain all-directional adjacent pixel differences of a cover image. In addition, multi-layer data embedding is used to increase the hiding capacity. In each embedding process, we can embed a large number of data intothe cover image by choosing the best scan path and the optimized pixel difference. Due to the fact that the grayscale values of adjacent pixels are close to each other, the all-directional adjacent pixel difference histogram contains a large number of points with equal values. Hence, we can improve the embedding capacity and keep the distortion low. As experimental results have shown, our scheme outperforms Ni et al.'s [18] and Chang et al.'s [9] scheme in terms of the image quality and the payload of the stego image.

The rest of this paper is organized as follows. Related works are first briefly introduced in Section II, and the proposed scheme is presented in Section III. Experimental results are illustrated in Section IV, and conclusions are drawn in Section V.

II. RELATED WORKS

A. Histogram shifting

Ni et al. [18] firstly introduced a reversible data hiding technique based on histogram shifting (sometimes referred to as histogram modification) that we shall describe briefly in this section. Let P_{max} be a pixel value that corresponds to the peak point of image histogram, and all the pixels whose values are larger (or smaller) than P_{max} are increased (or decreased) by 1. The whole image has no pixel whose value is equal to P_{max}+1 (or P_{max}-1) after histogram shifting. Therefore, P_{max} can be selected to embed data. When embedding a bit 1, P_{max} is modified to P_{max}+1 (or P_{max}-1), otherwise P_{max} remains intact. Obviously, the amount of message that can be embedded into an image equals to the number of pixels which are associated with the peak point. Because few images contain a large number of pixels with equal gray values, the major drawback of Ni et al.'s scheme is low payload.

B. Extended histogram modification

In 2008, in order to achieve high embedding capacity, Chang et al. [19] improved Ni et al.'s scheme and presented a new scheme based on histogram modification. By calculating the differences between adjacent pixels instead of simple pixel values, they extended Ni et al.'s scheme. Since the pixel grayscale values in a local area are often highly correlated and spatial redundancy, the distribution of pixel difference has a prominent maximum. Hence, their scheme can offer higher embedding capacity and keep lower distortion than Ni et al.'s scheme. However, once the differences between adjacent pixels in a certain sequential order are utilized to embed data, the distribution of pixel difference does not maintain a prominent maximum. This means that the performance of their scheme will drastically decline after the initial period embedding is applied.

III. PROPOSED SCHEME

In this section, we shall present the proposed scheme, whose embedding and extracting processes will be illustrated by the block diagrams shown in Figs. 1 and 2, respectively.

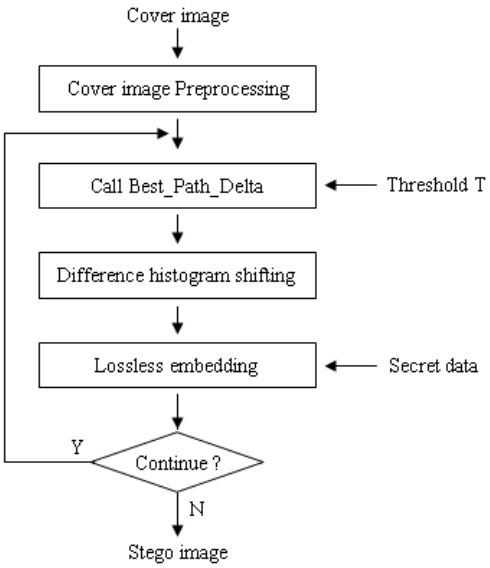


Figure 1. Data embedding algorithm.

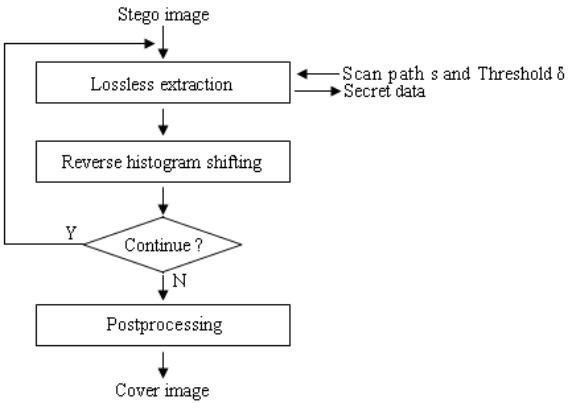


Figure 2. Data extraction algorithm.

To improve hiding capacity, we present an efficient extension of the histogram modification technique by calculating the differences between adjacent pixels in different scan paths instead of the same sequential order each time. In this way, we can obtain all-directional adjacent pixel differences of a cover image. Furthermore, multi-layer data embedding is used to increase the hiding capacity, and in each embedding process, we can offer

high embedding capacity and keep low distortion by choosing the best scan path and the optimized pixel difference. Details are given below.

A. Scan path

In order to obtain all-directional adjacent pixel differences of a cover image, various scan paths could be defined. Nine basic scan paths are defined in this work. For simplicity, the basic scan paths are shown for 4×4 regions, as shown in Fig.3. Let I be a grayscale image with H-row×W-column pixels, in which each pixel takes up 8 bits. After one scan path is assigned from Fig. 3, we can obtain a series of pixel values $p_1, p_2, p_3, \ldots, p_k$ along the pre-assigned scan path, we have k = H×W.

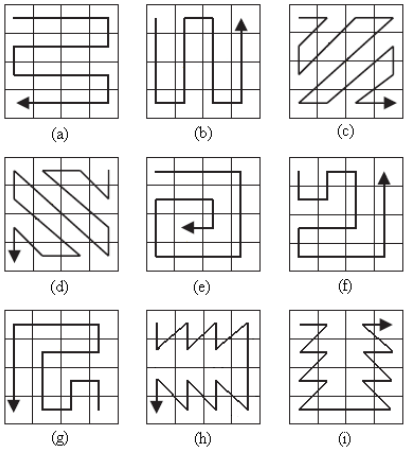


Figure 3. Nine basic scan paths. (a) path s1; (b) path s2; (c) path s3; (d) path s4; (e) path s5; (f) path s6; (g) path s7; (h) path s8; (i) path s9.

B. Adjacent pixel difference

As mentioned above, for a cover image, we can obtain a series of pixel values $p_1, p_2, p_3, \ldots, p_k$ along a pre-assigned scan path. Hence, we can compute k-1 adjacent pixel differences, and the sequence of adjacent pixel differences is defined as $<d_1, d_2, d_3, \ldots, d_{k-1}>$, and d_i is given by

$$d_i = p_{i+1} - p_i \quad . \tag{1}$$

where $1 \le i < k$.

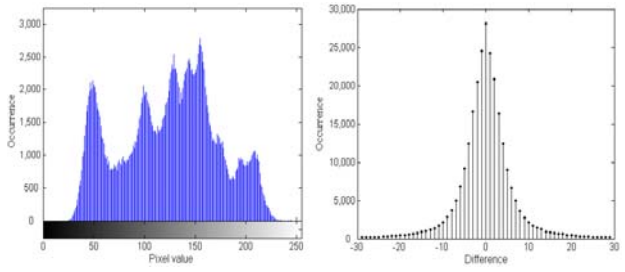


Figure 4. Ni et al.'s and our histograms for Lena. (a) Ni et al.'s histogram for Lena; (b) Our histogram for Lena.

For simplicity, in this paper, the terms "adjacent pixel difference" and "pixel difference" are interchangeable. The histogram of the pixel differences of image Lena with size 512×512×8 is illustrated in Fig. 4 (b), where the pre-assigned scan path are path s2. It is easy to see that difference histogram contains a larger number of points with equal values than image histogram as shown in Fig. 4(a). The difference histogram for most natural images would be similar to this, in the sense that difference values with small magnitudes occur more frequently [12].

C. Difference histogram shifting

Before data embedding, we can adjust pixel differences to obtain extra space. For this purpose, we introduce a threshold δ, which is a nonnegative integer. Assume that the pixel differences, whose absolute values are equal to δ, will be used to embed data. As Fig. 5 illustrates, two cases are taken into account for different δ.

Case 1. δ=0

The pixel differences are modified to

$$d_i^* = \begin{cases} d_i + 1 & \text{if } d_i > 0 \\ d_i & \text{otherwise} \end{cases}, \tag{2}$$

where $1 \le i < k$. After the pixel differences are modified, as Fig. 5 (a) shows, there is no pixel difference with value 1; in other words, we obtain extra space availably.

Case 2. $\delta \neq 0$

The pixel differences are modified to

$$d_i^* = \begin{cases} d_i + 1 & \text{if } d_i > \delta \\ d_i - 1 & \text{if } d_i < -\delta \\ d_i & \text{otherwise} \end{cases}, \tag{3}$$

where $1 \le i < k$. Also, as Fig. 5 (b) illustrates, after the pixel differences are modified, there is no pixel difference with value (δ+1) or –(δ+1) in the whole image.

The whole process described above is referred to as difference histogram shifting in this paper.

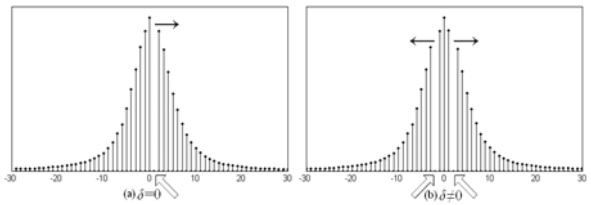


Figure 5. Difference histogram shifting. (a) case δ=0; (b) case $\delta \neq 0$.

Obviously, histogram shifting causes a smaller change in these differences than difference expansion (DE). Therefore, it is not necessary to check whether a histogram shift might cause overflow/underflow and the proposed scheme also eliminates the need to have a location map of the selected expandable locations. In addition, the computational intensity required for histogram shifting is much less than that required for the compression decompression engine.

A histogram shift can be easily reversed if δ is known. In fact, the reverse difference histogram shifting can be given by

$$d_i = \begin{cases} d_i^* - 1 & \text{if } d_i^* > 1 \\ d_i^* & \text{otherwise} \end{cases} \tag{4}$$

for the case δ=0, and

$$d_i = \begin{cases} d_i^* - 1 & \text{if } d_i^* > \delta + 1 \\ d_i^* + 1 & \text{if } d_i^* < -(\delta + 1) \\ d_i^* & \text{otherwise} \end{cases} \quad (5)$$

for the case $\delta \neq 0$.

D. Data embedding and Extraction

After the difference histogram shifting, as mentioned above, we obtain some extra space. Hence, we can embed data as follows by exploiting the extra space.

Case 1. δ=0

Assume that the pixel differences with value 0 will be employed to embed data, case δ=0 will deserve this occasion. All pixel differences are scanned one by one, once a d_i^* with value 0 is encountered, the to-be-embedded bit in the bit stream is checked. If the to-be-embedded bit is 1, the pixel difference with value 0 is added by 1. Otherwise, the pixel difference remains intact.

Case 2. $\delta \neq 0$

All pixel differences are scanned one by one, once a d_i^* with value δ or $-\delta$ is encountered, the to-be-embedded bit is checked. If the to-be-embedded bit is 1, the pixel difference with value δ is added by 1, and the pixel difference with value $-\delta$ is subtracted by 1. Otherwise, the pixel difference remains intact.

The above mentioned data embedding process can be described by

$$d_i'' = \begin{cases} d_i^* + 1 & \text{if } d_i^* = \delta \text{ and } b = 1 \\ d_i^* - 1 & \text{if } \delta \neq 0 \text{ and } d_i^* = -\delta \text{ and } b = 1 \\ d_i^* & \text{otherwise} \end{cases}, \quad (6)$$

where $1 \le i < k$ and b is the to-be-embedded bit.

After embedding bits into the pixel differences, we can produce the pixel values of the stego image by

$$p_i^* = p_{i+1} - d_i'', \quad (7)$$

where $1 \le i < k$, and the pixel p_k remains unaltered in the data embedding process. In fact, in extracting phase, as a reference pixel, p_k is used to restore other pixel values.

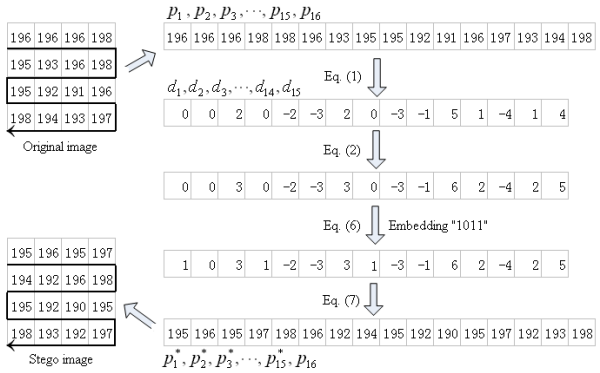


Figure 6. An illustration of data embedding of our scheme with a 4×4 image, scan path s1 and δ=0.

An example of data embedding is shown in Fig.6. Given a 4×4 image with pixel values being p_1=196, p_2=196, p_3=196, p_4=198, p_5=198, p_6=196, p_7=193, p_8=195, p_9=195, p_{10}=192, p_{11}=191, p_{12}=196, p_{13}=197, p_{14}=193, p_{15}=194 and p_{16}=198 along the scan path s1, the adjacent pixel differences are then generated by (1) as follows:

$d_1=p_2-p_1$=196-196=0, $d_2=p_3-p_2$=196-196=0, $d_3=p_4-p_3$=198-196=2, $d_4=p_5-p_4$=198-198=0, $d_5=p_6-p_5$=196-198=-2, $d_6=p_7-p_6$=193-196=-3, $d_7=p_8-p_7$=195-193=2, $d_8=p_9-p_8$=195-195=0, $d_9=p_{10}-p_9$=192-195=-3, $d_{10}=p_{11}-p_{10}$=191-192=-1, $d_{11}=p_{12}-p_{11}$=196-191=5,$d_{12}=p_{13}-p_{12}$=197-196=1, $d_{13}=p_{14}-p_{13}$=193-197=-4, $d_{14}=p_{15}-p_{14}$=194-193=1, $d_{15}=p_{16}-p_{15}$=198-194=4. Suppose the pixel differences with value 0 (i.e., δ=0) are used to embed data and the data bits are "1011", then the pixel differences d_1, d_2, d_4 and d_8 can be used to embed bits. After difference histogram is shifted by using (2), due to embedding 1, the differences d_1, d_4 and d_8 are added by 1, 1 and 1, respectively, and the differences d_2 remains intact due to embedding 0. The fifteen pixels of the stego image are generated by using (7), and p_{16} remains unchanged in the whole data embedding process.

Let h(x) denote the number of the pixel differences with value x, then the number of the absolute pixel differences with value $|x|$, which is denoted by h($|x|$), can be given by

$$\mathrm{h}(|\delta|) = \begin{cases} \mathrm{h}(\delta) & \text{for } \delta = 0 \\ \mathrm{h}(\delta) + \mathrm{h}(-\delta) & \text{for } \delta \neq 0 \end{cases}. \quad (8)$$

Obviously, the amount of message that can be embedded into an image equals to h($|\delta|$) in our scheme in a single pass.

Assume that there is no pixel with overflow or underflow in the embedding phase, in the worst case, all pixel values except for the reference pixel p_k will be altered by 1. Hence, the mean square error (MSE) is less than one, i.e., MSE<1. So, the PSNR of the stego image can be calculated by

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{255 \times 255}{\mathrm{MSE}}\right) > 48.13 \text{ (dB)}. \quad (9)$$

Data extraction process is a reverse process. With the stego image and the scan path and the threshold δ, we can extract the hidden data and recover the original cover image with no distortion.

E. Multi-layer embedding

As mentioned above, in the embedding phase, the original pixel values are only altered by 1 or kept unchanged. Therefore, as long as there is no pixel with value "<1" or ">254", the embedding process can be applied to the stego image once again, so that more data can be embedded into the cover image. In general, if there is no pixel with value "<α" or ">(255-α)", the embedding process can be applied to the stego image α times, and the whole process is called multi-layer embedding.

A tradeoff between the embedding capacity and the visual quality of the stego image is a critical issue for data hiding. From the angle of the embedding capacity, the best scan path s and the most optimized threshold δ, which can provide the maximum number of absolute pixel differences, should be used to embed data in each embedding pass. However, as mentioned in section 3.3, for a pre-assigned scan path, if we select a larger threshold δ to embed data, the number of the pixel differences need to be adjusted will be smaller and the visual quality of the stego image will be higher. Specially, as shown in Fig. 4(b), sometimes the numbers of the pixel differences are close to each other while the corresponding difference values are very different. For these pixel differences, we can choose a larger threshold δ to embed data, and this can ensure that the PSNR value is high while the embedding capacity remaining high relatively.

Let $D[s, \delta]$ denote the number of the absolute pixel differences with value δ along the scan path s, and MAX_N is the maximum value of $D[s, \delta]$ among all s and δ, and assume $D[s_{max}, \delta_{max}]$ = MAX_N.

Then, we introduce another threshold T. If $D[s, \delta] \geq$(MAX_N-T), we say that MAX_N and $D[s, \delta]$ are close to each other. Assume the sequence of the numbers of absolute pixel differences, which satisfy $D[s, \delta] \geq$(MAX_N-T), is defined as $<D[s_1, \delta_1], D[s_2, \delta_2], D[s_3, \delta_3], \ldots, D[s_q, \delta_q]>$. Let δ_m be the largest δ among the sequence of $<\delta_1, \delta_2, \delta_3, \ldots, \delta_q>$. If $|\delta_m-\delta_{max}| \leq 1$, i.e., δ_m is close to δ_{max}, s_{max} and δ_{max} are referred to as the best scan path and the optimized threshold δ; otherwise s_m and δ_m are referred to as the best scan path and the optimized threshold δ. The detailed process is given by Best_Path_Delta function which is described next.

Clearly, if the threshold T is set to 0, the maximum number of absolute pixel differences will be used to embed data. In general, along with the increase of T, the PSNR value of the stego image will increase, and the embedding capacity will decrease. In this way, by selecting a suitable T, we can reach a good balance between the embedding ratio and the stego image quality. By employing the Best_Path_Delta function, we can obtain the best scan path s and the optimized threshold δ, hence, we can embed more data by using the best scan path s and the optimized threshold δ each time. After that, the new best scan path s and new optimized threshold δ could be obtained again using the Best_Path_Delta function, and we can repeat the embedding process once more.

```
ALGORITHM Best_Path_Delta(I, T, s, δ)
/* Inputs: image, I; the threshold, T */
/* Output: the best scan path s and threshold δ */
{
For each sp∈{s1, s2 , … , s9}
    D[sp, 0:255] ← 0
    Get p1, p2, p3, …, pk from I along the scan path sp
    For i = 1 to k-1
        d ← |pi+1-pi|
        D[sp, d] ←D[sp, d] + 1
    End for
End for
MAX_N ← max(D[s1:s9, 0:255]);
Find smax and δmax that satisfy D[smax, δmax] = MAX_N;
w ←1; s ← smax; δ ←δmax ;
For each sp∈{s1, s2, … , s9}
    For d = 0 to 255
        If (D[sp, d]≥( MAX_N - T) and d>δmax+w)
            s ← sp; δ ← d; w ← w+1;
        End if
    End for
End for
}
```

F. Prevention of Overflow/Underflow

As mentioned above, in multi-layer embedding (assume that α-layer embedding would be applied), the maximum change of pixel values in the cover image will be $\pm\alpha$. For an 8-bit grayscale image, the permitted range is [0, 255]. Hence, if some pixel values in the cover image are larger than (255-α) or less than α, overflow/underflow may occur. For the pixels with value less than α or larger than (255-α), we can preprocess them as follows.

Save their locations and grayscale values orderly in a defined format to a file, and reset these pixel values like this: If the original pixel value is less than α, reset it to α; if the original pixel value is larger than (255-α), reset it to (255-α).

The overhead file will be compressed losslessly. The compressed overhead information and secret data are concatenated and embedded into the cover image together. As a result, actual data embedding capacity (pure payload) is less than or equal to the maximum data embedding capacity (payload).

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

Eight grayscale images sized 512×512× 8 each, as shown in Fig. 7(a)-(h), are used to evaluate the performance of the proposed scheme. The secret data used in our experiments are generated by a pseudo-random number generator. The measurement of the image quality used in the test is the peak signal to noise ratio (PSNR). In general, the larger the PSNR is, the higher the image quality is. Because the stego image can be inverted to its original one after data extraction, the embedding capacity is a significant factor when the visual quality of the stego image does not decline to an unacceptable degree, e.g., PSNR>30 dB[17].

A. Test image preprocessing

All test images are preprocessed before embedding data. In our experiments, assume that 12-layer embedding would be applied at most. Hence, the pixels with value less than 12 or larger than (255-12) in every test image have been preprocessed, i.e., we have saved their locations and grayscale values to a file and reset them to 12 and 243, respectively. In this way, we generate some overhead information, and the overhead information is then compressed losslessly. As a result, images Lena, Baboon, Boat, Peppers and Zelda have compressed

overhead information of size 87, 587, 94, 7921 and 3845 bytes, respectively, and the PSNR values of these images after preprocessing are 95.33, 58.40, 79.91, 47.28 and 54.67 dB, respectively.

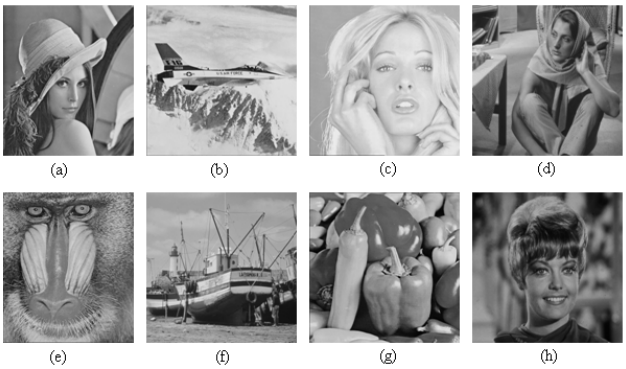


Figure 7. Test images. (a) Lena; (b) Airplane; (c) Tiffany; (d) Barbara; (e) Baboon; (f) Boat; (g) Peppers; (h) Zelda.

B. The result of 3-layer embedding

We first embed secret data into each test image by applying 3-layer embedding, and the results are depicted in Table I. The parameters δ and s are determined by employing the Best_Path_Delta function, where the threshold T is set to 1000.We re-implement Chang et al.'s scheme on all the preprocessed test images, and the results are also shown in Table I. As shown in Table I, all test images except for "Barbara" image, our scheme outperforms theirs in terms of payload and image quality. For image Barbara, the payload of the proposed scheme is higher than that of Chang et al.'s scheme, and their PSNR value is a little higher than ours. It should be noted that images with high textured areas and low correlation, such as "Baboon", produce less payload than smooth images, like "Airplane", and hence, embed less pure payload size at lower PSNR.

TABLE I.
PERFORMANCE COMPARISONS BETWEEN CHANG ET AL.'S ALGORITHM AND THE PROPOSED SCHEME ON ALL TEST IMAGES.

Images	Chang et al.			Proposed		
	PSNR (dB)	Payload	Pure payload	PSNR (dB)	Payload	Pure payload
Lena	40.350	0.389	0.386	40.72	0.431	0.428
Airplane	40.220	0.523	0.523	41.26	0.579	0.579
Tiffany	39.730	0.468	0.468	41.79	0.541	0.541
Barbara	39.870	0.303	0.303	39.63	0.370	0.370
Baboon	39.220	0.176	0.158	39.34	0.177	0.159
Boat	40.340	0.392	0.389	40.85	0.438	0.435
Peppers	39.210	0.506	0.264	40.21	0.524	0.282
Zelda	40.180	0.384	0.267	40.82	0.481	0.364

C. The result of 12-layer embedding

Next, we apply 12-layer embedding to all the test images, and the results are illustrated in Table II. The threshold δ and scan path s are determined by employing the Best_Path_Delta function, where the threshold T is set to 1000. As shown in Table II, our scheme can achieve average pure payload 1.02 bpp, and this means that our scheme can offer very high payload. In our experiments, we let threshold T be in the range of [200, 2500], and the experimental results show that T=1000 is a good candidate on the whole.

TABLE II.
THE PERFORMANCE OF THE PROPOSED SCHEME WITH 12-LAYER EMBEDDING ON ALL TEST IMAGES

Image	PSNR(dB)	Payload [a]		Pure Payload [b]	
		bits	bpp	bits	bpp
Lena	30.12	282147	1.08	281451	1.07
Airplane	30.08	338492	1.29	338492	1.29
Tiffany	29.90	328307	1.25	328307	1.25
Barbara	29.91	234631	0.90	234631	0.90
Baboon	30.06	133158	0.51	128462	0.49
Boat	30.36	276547	1.06	275795	1.05
Peppers	29.87	327825	1.25	264457	1.01
Zelda	30.11	312339	1.19	281579	1.07

a Average payload = 1.06 bpp; b Average pure payload = 1.02 bpp.

TABLE III.
THE PERFORMANCE OF THE PROPOSED SCHEME ON ALL TEST IMAGES FOR PSNR>30 DB

Image	PSNR(dB)	Payload [a]		Pure Payload [b]	
		bits	bpp	bits	bpp
Lena	30.12	282147	1.08	281451	1.07
Airplane	30.08	338492	1.29	338492	1.29
Tiffany	30.69	314142	1.20	314142	1.20
Barbara	30.56	223896	0.85	223896	0.85
Baboon	30.06	133158	0.51	128462	0.49
Boat	30.36	276547	1.06	275795	1.05
Peppers	30.71	311404	1.19	248036	0.95
Zelda	30.11	312339	1.19	281579	1.07

a Average payload = 1.05 bpp; b Average pure payload = 1.00 bpp.

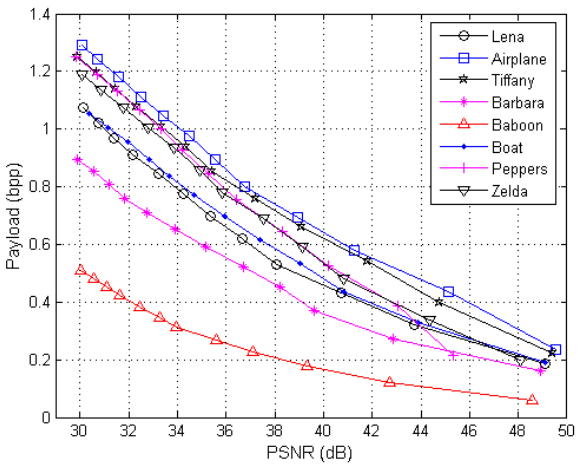
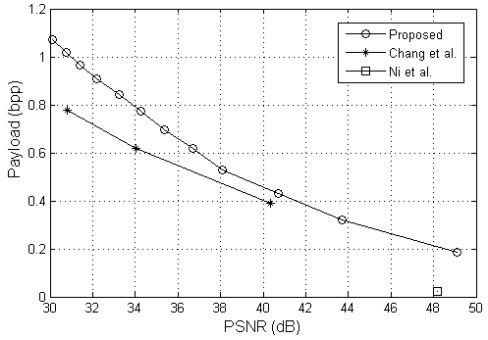


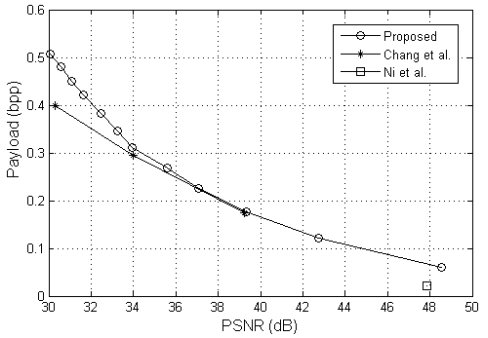
Figure 8. The embedding capacity versus PSNR by the proposed scheme with 12-layer embedding.

Due to the difference between the original image and the stego image being unnoticeable by the human eye when the PSNR value is higher than 30 dB [17], we collect the results which the PSNR values of the stego images are larger than 30 dB in Table III. It can be observed that the average pure payload of the test images for PSNR>30 dB can be up to 1.0 bpp. This implies that the proposed scheme can offer high embedding capacity and low image degradation. It can also be observed that the payload of each image is different according to the complexity of the cover image. In general, the higher the

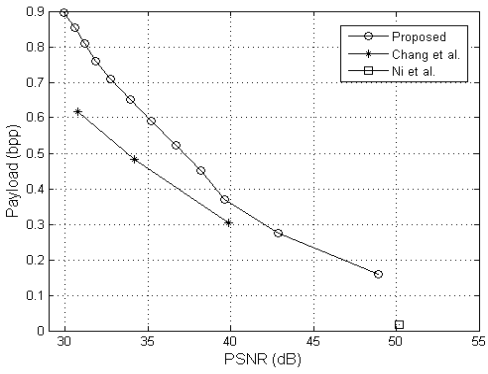
complexity of the cover image is, the lower the embedding capacity is. The image Baboon has the least payload among all the test images, and the embedding capacity of image Airplane is up to 1.29 bpp.



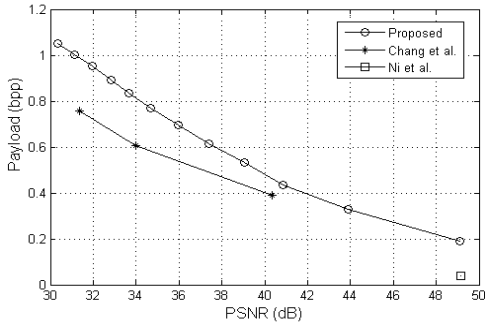
(a) Lena



(b) Baboon



(c) Barbara



(d) Boat

Figure 9. Performance comparison with existing reversible schemes [18-19] based on histogram shifting. (a) Lena; (b) Baboon; (c) Barbara; (d) Boat.

The whole results that 12-layer embedding is applied to each test image are illustrated in Fig. 8. Along with the increase of embedding times, the payload increases while the PSNR value decreases slowly. Hence, by controlling the embedding times, we can reach a good tradeoff between the embedding ratio and the stego image quality depending on the purpose of the application itself.

Fig. 9 shows the performance comparison of payload in bpp versus image quality in PSNR of the proposed scheme with that of some existing histogram-based lossless hiding schemes [18-19] for four commonly used images. We can see that Ni et al.'s scheme [18] has low embedding capacity compared to the others. Chang et al. [19] have improved Ni et al.'s work and have derived higher payload. Among the histogram-based schemes, it is clear that our proposed scheme performs better than Ni et al.'s and Chang et al.'s algorithms at all embedding rates, for four test images.

D. Comparison between our scheme and some other existing schemes with high capacity

Finally, to know exactly how competitive our work is, we compare it with many well-accepted techniques. The test images with size 512×512, namely Lena and Baboon are used across the evaluated schemes to see how they performed, and the comparison results are depicted in Table IV. As shown in Table IV, our scheme can offer the highest payload among all the schemes, and the PSNR values of larger than 30dB show that the visual quality does not decline to an unacceptable degree. Our scheme outperforms Tian's in terms of payload and image quality. Though some schemes have higher PSNR values, their payloads are limited relatively. The embedding capacities of the proposed scheme in Table IV are pure payload.

TABLE IV.
PERFORMANCE COMPARISONS AMONG SOME EXISTING REVERSIBLE DATA HIDING SCHEMES ON TWO STANDARD IMAGES: LENA AND BABOON.

Schemes	Lena (512×512)		Baboon (512×512)	
	Payload (bits)	PSNR (dB)	Payload (bits)	PSNR (dB)
Celik et al. [6]	74,600	38	15,176	38
Xuan et al. [8]	85,507	36.6	14,916	32.8
Tian [9]	233,067	29.97	95,852	29.41
Alattar [10]	173,655	36.6	86,264	36.6
Tseng et al. [11]	274,992	30.44	94,576	30.54
Kamstra [14]	135547	35.20	103653	30.12
Our scheme	281,451	30.12	128,462	30.06

V. CONCLUSION

This paper proposes a new lossless data hiding scheme using adjacent pixel difference based on scan path. Nine basic scan paths are defined, and this means all-directional adjacent pixel differences can be obtained. In addition, multi-layer data embedding is used to increase the hiding capacity, and in each embedding process, we can offer high embedding capacity and keep low distortion by choosing the best scan path and the optimized threshold δ. Experiments have been conducted

to show the effectiveness of the proposed method. Furthermore, the proposed scheme can conveniently apply the multi-layer embedding to the cover image for many times depending on the purpose of the application itself.

ACKNOWLEDGMENT

This work was supported in part by the Major Science and Technology Special Project of Zhejiang Province, China (No. 2007C11088) and the Science and Technology Project of Zhejiang Province, China (No. 2008C21077).

REFERENCES

[1] C.W. Honsinger, P. Jones, M. Rabbani, and J.C. Stoffel, "Lossless recovery of an original image containing embedded data, " U.S. Patent 6278791, Aug. 2001.
[2] C.D. Vleeschouwer, J.F. Delaigle, and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management, " IEEE Trans. Multimedia, 5(3),2003, pp.97-105.
[3] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," Proc. SPIE Security and Watermarking of Multimedia Contents, San Jose, CA, Jan. 2001, pp.197-208.
[4] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding - New paradigm in digital watermarking. " EURASIP J. Applied Signal Processing. 2002 (2), 2002, pp.185-196.
[5] A. A. C. M. Kalker and F. M. J. Willems, "Capacity bounds and constructions for reversible data-hiding," in Proc. 14th Int. Conf. Digital Signal Processing, Jul. 2002, vol. 1, pp.71–76.
[6] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding, "IEEE Trans. Image Processing. 14(2),2005, pp.253-266.
[7] M.U. Celik, G. Sharma, E. Saber, and A.M. Tekalp, "Lossless watermarking for image authentication: A new framework and an implementation, " IEEE Trans. Image Processing. 15(4), 2006, pp.1042-1049.
[8] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform, " IEE Electron. Lett., vol. 38, no. 25, Dec. 2002, pp.1646–1648.
[9] J. Tian, "Reversible data embedding using a difference expansion, " IEEE Trans. Circuits and Systems for Video Technology, 13(8), 2003, pp.890-896.
[10] A.M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform, " IEEE Trans. Image Processing. 13(8), 2004, pp.1147-1156.
[11] H.W. Tseng, and C.C. Chang, "An extended difference expansion algorithm for reversible watermarking, " Image and Vision Computing. 26, 2008, pp.1148-1153.
[12] Diljith M. Thodi and Jeffrey J. Rodríguez: "Expansion embedding techniques for reversible watermarking, " IEEE Transactions on Image Processing, Volume 16, Issue 3, March 2007, pp.721-730.
[13] Kim, H. J.; Sachnev, V.; Shi, Y. Q.; Nam, J.; Choo, H. G.: "A novel difference expansion transform for reversible data embedding, " IEEE Transactions on Information Forensics and Security, Volume 3, Issue 3, Sept. 2008, pp.456 - 465.
[14] L. Kamstra and H. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," IEEE Trans. Image Process.,vol. 14, no. 12, pp.2082–2090, Dec. 2005.
[15] C.C. Chang, and T. C. Lu, "A difference expansion oriented data hiding scheme for restoring the original host image, " The Journal of Systems and Software, vol. 79, 2006, pp. 1754-1766.
[16] C.C. Lin, S. P. Yang, and N. L. Hsueh, "Lossless Data Hiding Based on Difference Expansion without a Location Map," Doi 10.1109/CISP.2008.64, pp.8-12.
[17] C.C. Lee, , H.C. Wu, , C.S. Tsai, and Y.P. Chu, "Adaptive lossless steganographic scheme with centralized difference expansion," Pattern Recognition, vol. 41, 2008, pp.2097-2106.
[18] Z. Ni, Y.Q. Shi, N. Ansari, and W. Su, "Reversible data hiding, IEEE Trans. Circuits and Systems for Video Technology, " 16(3), 2006, pp.354-362.
[19] C.C. Chang, W.L. Tai, and K.N. Chen, "Lossless Data Hiding Based on Histogram Modification for Image Authentication, " Doi 10.1109/EUC.2008.20, pp.506-511.

Xianting Zeng is an associate professor in College of Information Engineering, China Jiliang University, Hangzhou, China. Currently, he is a Ph.D. candidate in college of computer science and technology, Zhejiang University. He received his B.S degree in computer science from Sichuan University, Chengdu, China, in 1986, and the M.S degree in computer engineering from South China University of Technology, Guangzhou, China, in 1989. His research interests include image processing, data hiding.

Lingdi Ping received her B.S degree from Zhejiang University, Hangzhou, China, in 1969. She is currently a professor of computer science, Zhejiang University. Her research interests include digital watermarking, signal and image processing.

Zhou Li received his B.S degree from Zhejiang University, Hangzhou, China, in 2005. He is currently a Ph.D. candidate in college of computer science and technology, Zhejiang University. His research interests include information security, signal processing and data hiding.