

An Information Hiding Scheme Based on Pixel-Value-Ordering and Prediction-Error Expansion with Reversibility

Ching-Chiuan Lin

Department of Information
Management
Overseas Chinese University
Taichung, Taiwan

Shih-Chieh Chen

Department of Information
Management
Overseas Chinese University
Taichung, Taiwan

Kuo Feng Hwang

Department of Information
Technology
Overseas Chinese University
Taichung, Taiwan

Abstract—This paper proposes a data hiding scheme based on pixel-value-ordering and prediction-error expansion. In a natural image, most neighboring pixels have similar pixel values, i.e. the difference between neighboring pixels is small. Based on the observation, we may predict a pixel's value according to its neighboring pixels. The proposed scheme divides an image into non-overlapping blocks each of which consists of three pixels, and pixels in a block are sorted in a descending order. Messages are embedded into two difference values, where one is between the largest and medium pixels and the other is between the smallest and medium ones. In the embedding process, difference values equal to 0 or greater than 1 are unchanged or increased by 1, respectively, and those equal to 1 are also unchanged or increased by 1 if the message bit to be embedded is equal to 0 or 1, respectively. Calculating the difference value, one may extract a message bit of 0 or 1 if it is equal to 1 or 2, respectively. Recovering pixels is done by decreasing those difference values by 1 if they are equal to or larger than 2. Experimental results demonstrate that the proposed scheme may provide much larger embedding capacity, comparing to existing study, and a satisfied image quality.

Keywords—Reversible data hiding; Pixel-value-ordering; Prediction-error expansion

I. INTRODUCTION

Digital image is a digitized medium stored in an electronic file for presenting objects to people. If someone would like to know more about the image, a separated voice or text file is required, which is an inconvenient way. Alternatively, the owner of the image may type texts on the image for giving more information about the image to a viewer. This may damage or distort the image and the amount of added texts is limited. In addition, distorting an important image, e.g. a medical image, is unaccepted, since a distorted medical image may result in an incorrect diagnosis. Data hiding is a way of imperceptibly embedding important information into a medium, which may provide a way for annotating or watermarking an image, or secret communication. An image with embedded information is called a stego-image. Usually, the stego-image is visually the same as its original image so that people may not perceive the embedded objects.

When data are embedded into an image, the image may be distorted. In general, the more data we embed, the more the

image would be distorted. Embedding capacity and image distortion is a tradeoff. Therefore, a good data hiding scheme should be able to embed as many messages as possible and distort the cover image as slightly as it could. How to embed a large amount of data into an image and achieve a slightly distorted image is an important issue for data hiding applications. The issue is more important if we want the distorted image to be recoverable.

A famous scheme for reversibly embedding messages into an image, based on difference expansion, was proposed by Tian [1] in 2003. Based on differences between neighboring pixels in an image are small, his scheme expands a difference value between two neighboring pixels by increasing or decreasing their pixel values. Specifically, if a difference value d between pixels x and y is calculated as $d = x - y$, the difference is expanded to $2d = x' - y'$, where $x' = x + \lfloor d/2 \rfloor$ and $y' = y - \lfloor (d + 1)/2 \rfloor$ if $x \geq y$. Then, a message bit m is embedded into the expanded difference by setting $x'' = x' + m$, if d is expandable, i.e. x'' and y' are not over or under saturated. Later, for an expandable difference, the embedded message may be extracted by calculating $m = (x'' - y') \bmod 2 = (x' + m - y') \bmod 2 = (2d + m) \bmod 2$, and x and y are recovered by $x = (x'' - m) - \lfloor d'/2 \rfloor$ and $y = y' + \lfloor (d' + 1)/2 \rfloor$, where $d' = (x'' - m - y')/2$. Since we cannot guarantee that every difference is expandable, a location map recording whether a difference is expandable or not is required. Fortunately, most differences are expandable and the location map may be compressed in a satisfied compression ratio so that it consumes only a small part of embedding space. Later, a number of studies [2-5] inspired by Tian's scheme were proposed.

In 2006, a novel reversible data hiding scheme, based on shifting pixel histogram, was proposed by Ni et al. [6]. They calculated the number of pixels with the same pixel value and obtained a pixel histogram where the peak point was selected for embedding messages. Since most images contain few pixels with very small or large pixel values, the histogram on the right or left of peak point may be shifted one to the right or left side, respectively, so that there would be available space for embedding messages. A number of studies based on shifting histogram were proposed. References [7-10] improved Ni et al.'s scheme by shifting difference histogram, instead of

pixel histogram. Since difference between neighboring pixels usually is small, the peak point of difference histogram would be much higher than that of pixel histogram. Generally, these schemes have a higher embedding capacity comparing to those based on shifting pixel histogram.

A group of studies [11-16] explored neighboring pixels in an image and predicted a pixel value by its neighboring pixels. Then they adopted the histogram of predicted error for embedding messages. In general, the peak point of prediction error histogram is higher than difference histogram. Nevertheless, the embedding capacity of this kind of approach depends on its predictive method.

This paper proposes a data hiding scheme based on pixel-value-ordering and predication-error expansion. In a natural image, most neighboring pixels have similar pixel values, i.e. the difference between neighboring pixels is small. Based on the observation, we may predict a pixel's value according its neighboring pixels. The proposed scheme divides an image into non-overlapping blocks each of which consists of three pixels, and pixels in a block are sorted in a descending order. Messages are embedded into two difference values, where one is between the largest and medium pixels and the other is between the smallest and medium ones. In the embedding process, difference values equal to 0 or greater than 1 are unchanged or increased by 1, respectively, and difference values equal to 1 are also unchanged or increased by 1 if the message bit to be embedded is equal to 0 or 1, respectively. Calculating the difference value, one may extract a message bit of 0 or 1 if it is equal to 1 or 2, respectively. Recovering pixels is done by decreasing those difference values by 1 if they are equal to or larger than 2. Experimental results demonstrate that the proposed scheme may provide much larger embedding capacity, comparing to existing study, and a satisfied image quality.

The rest of this paper is organized as follows. Section II briefly reviews Li et al.'s scheme [15]. The proposed scheme is introduced in Section III. Section IV demonstrates our experimental results and compares the performance of the proposed scheme with that of Li et al.'s. Finally, conclusions are given in Section V.

II. RELATED WORK

Li et al. [15] proposed a data hiding scheme based on pixel-value-ordering and predication-error expansion. First, their scheme divides an image into non-overlapping blocks each of which consists of four pixels x_0, x_1, x_2 and x_3 as shown in Fig. 1(a). The four pixel values in a block are sorted in an ascending order as shown in Fig. 1(b) where p_0, p_1, p_2 and p_3 denote the sorted pixels. Then calculate

$$p'_3 = \begin{cases} p_3 & \text{if } p_3 - p_2 = 0, \\ p_3 + m & \text{if } p_3 - p_2 = 1, \\ p_3 + 1 & \text{otherwise,} \end{cases}$$

where p'_3 and m are the stego-pixel of p_3 and the message bit to be embedded, respectively. In Fig. 1(c), the stego-pixel is $p'_3 = 26$ (i.e. $x_1 = 26$) and this block may not embed a message bit since $p_3 - p_2 = 2$. Finally, the stego-pixels would

be $(x_0, x_1, x_2, x_3) = (20, 26, 22, 23)$. If $p_3 = 23$ (i.e. $x_1 = 23$), the embedding result would be unchanged, i.e. $(x_0, x_1, x_2, x_3) = (20, 23, 22, 23)$. In case of $p_3 = 24$, the embedding result would be $(x_0, x_1, x_2, x_3) = (20, 24, 22, 23)$ or $(x_0, x_1, x_2, x_3) = (20, 25, 22, 23)$ if the message bit to be embedded is $m = 0$ or $m = 1$, respectively.

x_0	x_1	x_2	x_3
20	25	22	23

(a)

p_0	p_1	p_2	p_3
x_0	x_2	x_3	x_1
20	22	23	25

(b)

p_0	p_1	p_2	p'_3
x_0	x_2	x_3	x_1
20	22	23	26

(c)

Fig. 1. An embedding process example of Li et al.'s scheme

The recovery process is the reverse of its embedding process. If one would like to extract an embedded message bit from a stego-block and recover a stego-pixel to its original pixel, he/she may calculate

$$m = \begin{cases} 0 & \text{if } p'_3 - p_2 = 1, \\ 1 & \text{if } p'_3 - p_2 = 2, \\ \text{null} & \text{otherwise,} \end{cases}$$

and

$$p_3 = \begin{cases} p'_3 & \text{if } p'_3 - p_2 \in \{0, 1\}, \\ p'_3 - 1 & \text{otherwise.} \end{cases}$$

The rationale of Li et al.'s scheme is based on the concept of pixel values in a block are similar for a natural image. Namely, the difference of pixel values in a block is small and they predicted a difference value (denoted by δ) of one. If the difference is larger than one (i.e. $\delta > 1$), they increased δ by one so that there would be an available space of $\delta = 1$ for embedding a message bit into a block. Then if the message bit m to be embedded is 0 or 1, δ is unchanged or set to 2, respectively.

III. PROPOSED SCHEME

The proposed scheme includes the embedding and extraction processes. The former embeds secret messages into a cover image and obtains a stego-image, and the latter extracts the embedded secret messages from the stego-image and completely recovers it to its original image. Most studies avoid the problem of saturated conditions (i.e. pixel value equal to 0 or 255 for a 256-gray-level image). It is worth mentioning that the proposed scheme also includes a solution for solving the

problem of saturated conditions. The two processes are presented in the following, respectively.

A. Embedding process

This section shows the embedding process for a cover image I with n pixels. Let the secret message, with r bits, to be embedded be a bit string $M = m_0 m_1 \dots m_j \dots m_{r-1}$, where $m_j \in \{0,1\}$ and $0 \leq j \leq r-1$. The embedding process is shown as follows.

1) For a gray-level image I , divide it into non-overlapping blocks each of which contains three neighboring pixels denoted by x_i, x_{i+1} , and x_{i+2} , respectively, where $0 \leq x_i, x_{i+1}, x_{i+2} \leq 255$.

2) For each block, sort their pixel values in a descending order denoted by $p_{i,max}$, $p_{i,med}$, and $p_{i,min}$, where $p_{i,max}$, $p_{i,med}$, and $p_{i,min}$, are the largest, medium, and smallest values, respectively, in block i .

3) Embed m_j by setting stego-pixel values $p'_{i,max}$ and $p'_{i,min}$ of $p_{i,max}$ and $p_{i,min}$, respectively, as

$$p'_{i,max} = \begin{cases} p_{i,max} & \text{if } E_{i,max} = 0, \\ p_{i,max} + m_j & \text{if } E_{i,max} = 1, \\ p_{i,max} + 1 & \text{otherwise,} \end{cases} \quad (1)$$

$$p'_{i,min} = \begin{cases} p_{i,min} & \text{if } E_{i,min} = 0, \\ p_{i,min} - m_{j+1} & \text{if } E_{i,min} = 1, \\ p_{i,min} - 1 & \text{otherwise,} \end{cases} \quad (2)$$

if $p_{i,min} = 1$ or $p_{i,max} = 254$, where $E_{i,max} = p_{i,max} - p_{i,med}$ and $E_{i,min} = p_{i,med} - p_{i,min}$ are the prediction errors of $p_{i,max}$ and $p_{i,min}$, respectively. Mark a block with $p'_{i,min} = 0$ or $p'_{i,max} = 255$ as overhead information. Let $H = h_0 h_1 \dots h_k \dots h_{t-1}$ be the required overhead information for extracting the embedded messages, where t is its length, $h_k \in \{0,1\}$, and $0 \leq k \leq t-1$.

4) Let M_0 be the sub-message embedded in step 3 and $M = M_0 || M_1$, where $||$ denotes concatenating. For each block i with $2 \leq p_{i,min}, p_{i,max} \leq 253$, embed H and M_1 by performing (1) and (2). Overhead information would not be generated in this step since $0 < p'_{i,min}, p'_{i,max} < 255$ in this step.

5) Finally, the stego-image I' is obtained.

In step 2, if the pixels in block i are sorted in an ascending, the embedding process in steps 3–4 would still be workable. However, for simplicity, the embedding process selects the descending order.

The proposed scheme applies the medium value $p_{i,med}$ in a block i to predict its neighboring pixel values $p_{i,max}$ and $p_{i,min}$. The rationale is that, in a natural image, most image blocks are smooth. Thereby we may expect that pixel values of $p_{i,max}$ and $p_{i,min}$ are similar to that of $p_{i,med}$ and the prediction error would be small. This implies that we would obtain more embedding space for embedding a message than without prediction.

The problem of recording saturated blocks with pixels modified is taken into account in step 3, and it is recorded by the overhead information embedded in the blocks mentioned in step 4. Note that we would not encounter the problem of saturated blocks in step 4.

B. Extraction process

Whenever a decoder gets the setgo-image I' , he/she may follow the following process to extract the embedded message and completely recover image I' to its original image I . In the process, the decoder could determine whether a saturated block needs to be recovered or not, from the extracted overhead information. The extraction process is presented as follows.

1) Divide setgo-image I' as it was divided by the encoder in the embedding process.

2) As in the embedding process, for each block i , sort their pixel values x'_i, x'_{i+1} , and x'_{i+2} in a descending order denoted by $p'_{i,max}, p'_{i,med}$, and $p'_{i,min}$, where $p'_{i,max}, p'_{i,med}$, and $p'_{i,min}$ are the largest, medium, and smallest stego-pixel values, respectively, in block i .

3) For each block i with $1 \leq p'_{i,min}, p'_{i,max} \leq 254$, calculate $E'_{i,max} = p'_{i,max} - p'_{i,med}$ and $E'_{i,min} = p'_{i,med} - p'_{i,min}$ and extract

$$HM_v = \begin{cases} 0 & \text{if } E'_{i,max} = 1, \\ 1 & \text{if } E'_{i,max} = 2, \\ \text{Null} & \text{otherwise,} \end{cases} \quad (3)$$

$$HM_{v+1} = \begin{cases} 0 & \text{if } E'_{i,min} = 1, \\ 1 & \text{if } E'_{i,min} = 2, \\ \text{Null} & \text{otherwise,} \end{cases} \quad (4)$$

where HM_v is a bit of mixed messages H and M . Then perform

$$p_{i,max} = \begin{cases} p'_{i,max} & \text{if } E'_{i,max} \leq 1, \\ p'_{i,max} - 1 & \text{if } E'_{i,max} \geq 2, \end{cases} \quad (5)$$

$$p_{i,min} = \begin{cases} p'_{i,min} & \text{if } E'_{i,min} \leq 1, \\ p'_{i,min} - 1 & \text{if } E'_{i,min} \geq 2. \end{cases} \quad (6)$$

Extract H from those blocks with $2 \leq p_{i,min}, p_{i,max} \leq 253$. Note that the bit order of h_k is determined by the block index. Specifically, if h_k is extracted from block a , h_{k+1} would be extracted from block b where $b > a$.

4) Given H in step 3, recover stego-pixels and extract message bits from those blocks with $p'_{i,min} = 0$ or $p'_{i,max} = 255$.

5) According to the block index, rearrange the message bits extracted from blocks, in steps 3 and 4, with $p_{i,min} = 1$ or $p_{i,max} = 254$ to get the sub-message M_0 . Another sub-message M_1 may be extracted from blocks, in step 3, with $2 \leq p_{i,min}, p_{i,max} \leq 253$. Finally, $M = M_0 || M_1$ is obtained and the original cover image I is completely recovered.

Block	x_i	x_{i+1}	x_{i+2}	p_{i_max}	p_{i_med}	p_{i_min}	H/M	p'_{i_max}	p'_{i_med}	p'_{i_min}	x'_i	x'_{i+1}	x'_{i+2}
0	145	148	147	148	147	145	1	149	147	144	144	149	147
1	147	148	149	149	148	147	01	149	148	146	146	148	149
2	146	146	145	146	146	145	1	146	146	144	146	146	144
3	254	250	254	254	254	250		254	254	249	254	249	254
4	1	2	4	4	2	1	0	5	2	1	1	2	5
5	1	2	4	4	2	1	1	5	2	0	0	2	5
6	0	2	3	3	2	0		3	2	0	0	2	3

Fig. 2. An example of the proposed scheme

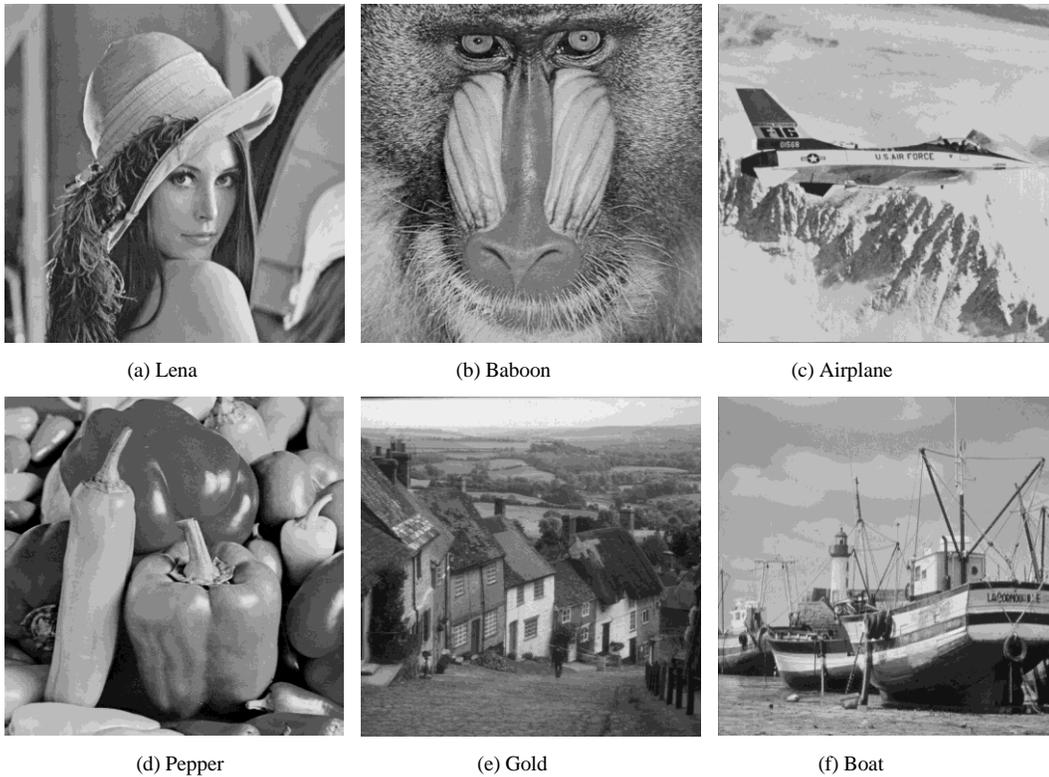


Fig. 3. Test images

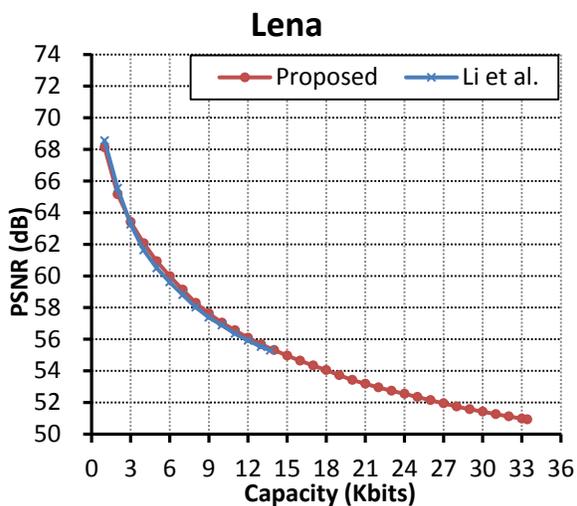
C. An example of the proposed scheme

An example with 7 blocks is given in this section to illustrate the proposed scheme. The example image is a gray-level one with pixel values between 0 and 255.

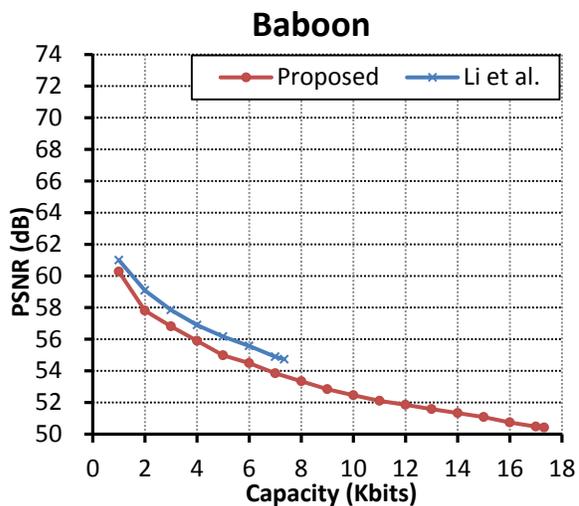
In the embedding process, the divided blocks are shown in Fig. 2, where column H/M denotes both overhead information and message bits to be embedded. Pixel values in a block are sorted in a descending order and the largest and smallest pixel values in a block are marked by red and blue color, respectively. Let the message to be embedded be $M = 0111$. The first processed block is, in step 3, block 3 which embeds nothing, and $m_0 = 0$ is embedded into block 4 by setting $p'_{i_max} = 5$ and $p'_{i_min} = 1$. Then $m_1 = 1$ is embedded into block 5. Block 6 is a saturated block and its pixels is remained unchanged, and $M_0 = 01$ in this step. For simplicity, assume the overhead information is $H = 10$.

In step 4, H is embedded into blocks 0 and 1, and $M_1 = 11$ is embedded into blocks 1 and 2. Note that block 1 embeds two bits, one is from the second bit of H and the other is from the first bit of M_1 . After M_1 is embedded, the embedding process is completed.

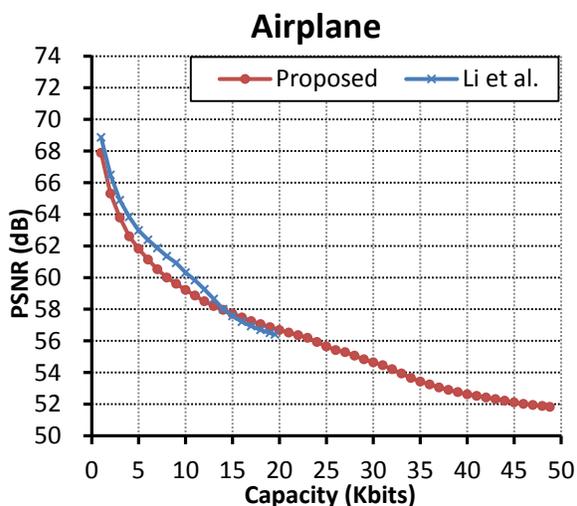
Moving to the extraction process, we divide setgo-image I' into non-overlapping blocks as it was divided by the encoder in the embedding process and sort pixel values in a descending order for each block. First, in step 3, mixed messages of H and M , 10110, are extracted from blocks 0–4 by performing (3) and (4) and these blocks are recovered by performing (5) and (6). Next, the overhead information $H = 10$ is extracted from blocks 0–3, since $2 \leq p_{i_min}, p_{i_max} \leq 253$ in these blocks. For simplicity, assume that $H = 10$ means that block 5 needs to be recovered. Then a bit of 1 is extracted from block 5, and the block is recovered. After all messages are extracted and



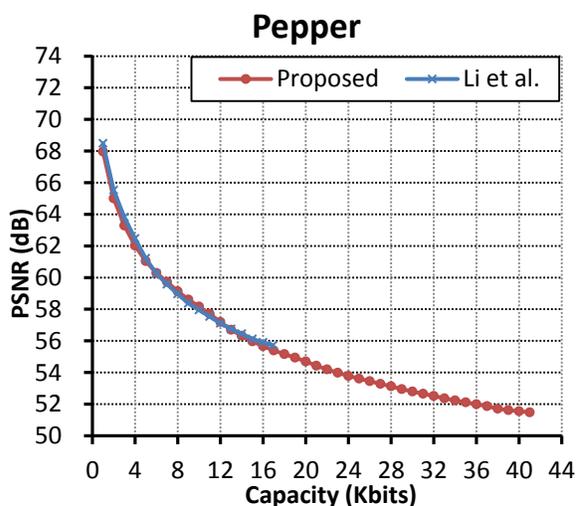
(a) Lena



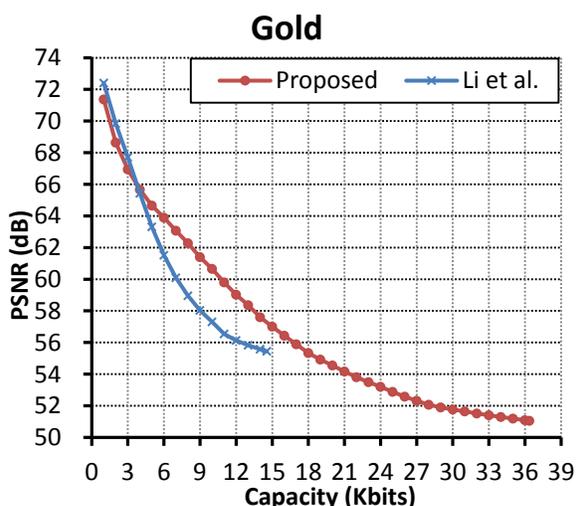
(b) Baboon



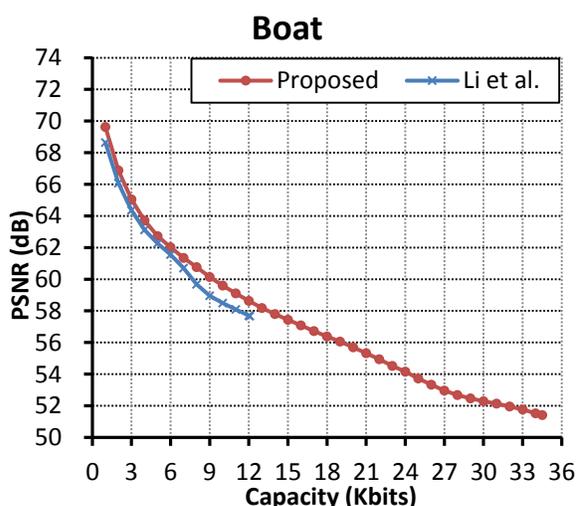
(c) Airplane



(d) Pepper



(e) Gold



(f) Boat

Fig. 4. Comparing performance of the proposed scheme with Li et al.'s scheme

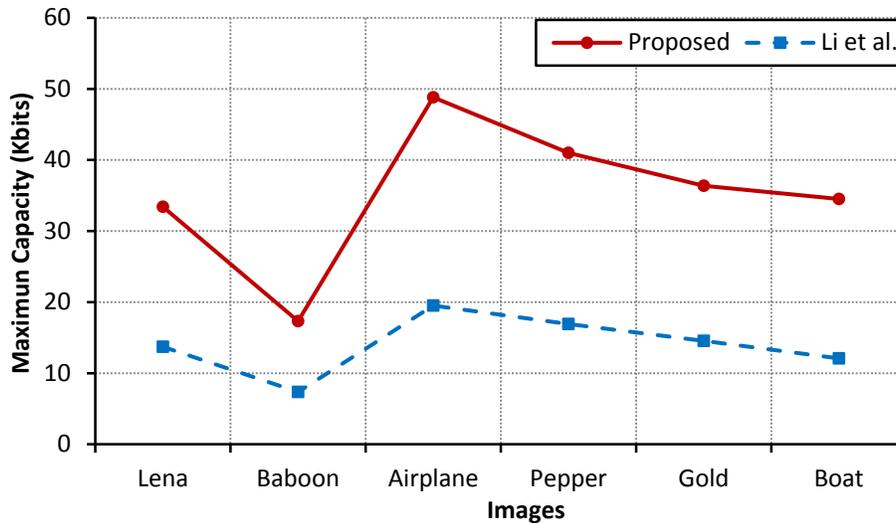


Fig. 5. Comparison of maximum embedding capacity

stego-pixels are recovered, $M_0 = 01$ is obtained from blocks with $p_{i_{min}} = 1$ or $p_{i_{max}} = 254$ by rearranging the extracted bit string according to their block indexes. Similarly, $M_1 = 11$ is obtained from the above mixed messages 10110 but removing $H = 10$ and the last bit (i.e. the first bit of M_0). Finally, $M = M_0 || M_1 = 0111$ is extracted and the cover image is completely recovered.

From the above example, we observe that both stego-blocks 5 and 6 are saturated blocks. However, stego-block 5 needs to be recovered and, in the extraction process, the block must be distinguished from stego-block 6 which was not changed in the embedding process. The problem about whether a saturated block needs to be recovered or not may be solved by the embedded overhead information. In step 3 in the extraction process, we first extract messages from stego-blocks with $p'_{i_{min}} = 1$ or $p'_{i_{max}} = 254$, since these blocks would not be saturated ones. Then we decode overhead information from blocks with $2 \leq p_{i_{min}}, p_{i_{max}} \leq 253$. As long as the overhead information was decoded, we can recognize which saturated blocks need to be recovered. Note that blocks with $p_{i_{min}} = 1$ or $p_{i_{max}} = 254$ embed user's messages, if any, instead of overhead information in step 3 in the embedding process. The example in the section has illustrated how the overhead information is embedded into and extracted from a stego-image.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of proposed scheme, we implemented the proposed scheme in Java on a personal computer and embedded randomly generated secret messages into cover images, as shown in Fig. 3, which were downloaded from [17]. All cover images are grayscale with 256 levels and the dimension is 512×512 . A test image may be divided into $\lceil 512 \times 512 / 3 \rceil = 87381$ non-overlapping blocks. In the extreme condition, if a block embeds two bits, the embedding capacity of an image may be up to 174762 bits. Each $p_{i_{max}}$ and $p_{i_{min}}$ in a block may be modified no more than one. For a 256-gray-level image with n pixels, the image quality, or the

similarity between a stego-image and its cover image, is evaluated by peak signal to noise ratio (PSNR) calculated as

$$\text{PSNR} = 10 \times \log_{10}(255 \times 255 / \text{MSE}),$$

In the above equation, MSE is mean square error calculated as

$$\text{MSE} = \sum_{j=0}^{n-1} (x_i - x'_i)^2 / n$$

where x_i and x'_i denote cover and stego-pixel values, respectively.

We also implemented Li et al.'s scheme, on the same platform, to compare the performance of our scheme with their scheme's. An image applying the proposed scheme may obtain a larger number of blocks comparing to applying Li et al.'s. Therefore, the image may provide a larger embedding capacity if it applies the proposed scheme.

A smooth image (e.g. Airplane) may contain a larger number of smooth blocks comparing to a complex image (e.g. Baboon). Here a smooth block is an image block with similar pixel values, and it may result in a smaller prediction error. The proposed scheme embeds a message bit into a block with prediction error equal to one which is a smaller prediction error. Since Baboon and Airplane are smooth and complex images, respectively, their prediction errors are usually larger and smaller, respectively, than the other test images.

Fig. 4 illustrates the comparison of performance between the proposed and Li et al.'s schemes in terms of image quality (PSNR) and embedding capacity. The figure shows, in a low embedding capacity, the two schemes have similar performance. We can observe that the PSNR is higher than 50 dB for each test images in Fig. 4, and the proposed scheme may provide image quality similar to Li et al.'s scheme but a much higher embedding capacity than their scheme.

In the worst condition, if $p_{i_{max}}$ is increased by one and $p_{i_{min}}$ is decreased by one, we have $\text{MSE} = (1^2 + 1^2) / 3 =$

0.667 and $PSNR = 10 \times \log_{10}(255 \times 255/0.667) = 49.89$ dB. This means the proposed scheme may guarantee the image quality higher than 49.89 dB for a 256-gray-level image. Even in this condition, the difference between a cover image and its stego-image would not be detected by human eye.

Fig. 5 shows the comparison of maximum embedding capacity between the proposed and Li et al.'s schemes. For the test images in Fig. 3, the embedding capacity of the proposed scheme is more than twice as high as Li et al.'s. A reason is that we may embed, at most, two message bits into a block, whereas Li et al.'s scheme may embed, also at most, only one message bit into a block. In addition, the proposed scheme could provide more blocks than Li et al.'s. The proposed scheme may satisfy more applications' requirement if they need a higher embedding capacity and satisfied image quality.

V. CONCLUSIONS

We have introduced an information hiding scheme, with reversibility, based on pixel-value-ordering and prediction-error expansion. The proposed scheme divides an image into non-overlapping blocks each of which contains three pixels and sorts pixels in a block in a descending order. After embedding, the property of pixel-value-ordering in a block is invariant so that the image can be recovered. Comparing to Li et al.'s scheme, the proposed scheme can achieve more blocks for embedding. In addition, a block may embed up to two message bits. Consequently, the proposed scheme can obtain a higher embedding capacity and satisfied image quality. Experimental results show that the proposed scheme achieves an embedding capacity more than twice as high as Li et al.'s scheme on the same level of image quality. The proposed scheme is a good candidate for reversible data-hiding applications which need a high embedding capacity and low distortion.

REFERENCES

- [1] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 890–896, August 2003.
- [2] A. M. Alattar, "Reversible watermarking using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, pp. 1047–1156, August 2004.
- [3] C. C. Chang and T. C. Lu, "A difference expansion oriented data hiding scheme for restoring the original host images," *Journal of Systems and Software*, vol. 79, pp. 1754–1766, December 2006.
- [4] D. M. Thodi and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, pp. 721–730, March 2007.
- [5] O. M. Al-Qershi and B. E. Khoo, "High capacity data hiding schemes for medical images based on difference expansion," *Journal of Systems and Software*, vol. 84, pp. 105–112, January 2011.
- [6] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 354–362, March 2006.
- [7] C.-C. Lin and N.-L. Hsueh, "A lossless data hiding scheme based on three-pixel block differences," *Pattern Recognition* vol. 41, pp. 1415–1425, April 2008.
- [8] C.-F. Lee and H.-L. Chen, "Adjustable prediction-based reversible data hiding," *Digital Signal Processing*, vol. 22, pp. 941–953, December 2012.
- [9] W.-L. Tai, C.-M. Yeh, and C.-C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, pp. 906–910, June 2009.
- [10] C.-C. Lin, W.-L. Tai, and C.-C. Chang, "Multilevel reversible data hiding based on histogram modification of difference images," *Pattern Recognition*, vol. 41, pp. 3582–3591, December 2008.
- [11] P. Tsai, Y.-C. Hu, and H.-L. Yeh, "Reversible image hidings cheme using predictive coding and histogram shifting," *Signal Processing*, vol. 89, pp. 1129–1143, June 2009.
- [12] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, pp. 989–999, July 2009.
- [13] W. Hong and T.-S. Chen, "A local variance-controlled reversible data hiding method using prediction and histogram-shifting," *Journal of Systems and Software*, vol. 83, pp. 2653–2663, December 2010.
- [14] W. Hong, T.-S. Chen, Y.-P. Chang, and C.-W. Shiu, "A high capacity reversible data hiding scheme using orthogonal projection and prediction error modification," *Signal Processing*, vol. 90, pp. 2911–2922, November, 2010.
- [15] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," *Signal Processing*, vol. 93, pp. 198–205, January 2013.
- [16] B. Ou, X. Li, Y. Zhao, and R. Ni, "Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion," *Signal Processing: Image Communication*, in press.
- [17] <http://sipi.usc.edu/database/>