

Improving visual quality of reversible data hiding by twice sorting

Yang Yang^{1,2} · Weiming Zhang¹ · Xiaocheng Hu¹ · Nenghai Yu¹

Received: 11 October 2014 / Revised: 13 June 2015 / Accepted: 13 July 2015 /
Published online: 30 July 2015
© Springer Science+Business Media New York 2015

Abstract In most literatures on reversible data hiding (RDH), the visual quality of marked images is only assessed by PSNR, and the smoothness-priority-based sorting technique is efficient for improving PSNR. However, modifications in smooth areas are conflict with other criterion of visual quality such as Just Noticeable Difference (JND). To reconcile this contradiction, we propose a twice sorting scheme, in which the pixels are first sorted and divided into several levels with a smoothness criterion, and then sorted twice with JND in each level. According to the sorted order, message bits are embedded into the predicted errors of pixels based on rhombus prediction. Experimental results show that this novel method significantly outperforms previous JND-related RDH schemes on not only PSNR but also SSIM and JND distortion.

Keywords Reversible data hiding · Visual quality · Just noticeable difference

✉ Weiming Zhang
zhangwm@ustc.edu.cn

Yang Yang
skyyang@mail.ustc.edu.cn

Xiaocheng Hu
hxc@mail.ustc.edu.cn

Nenghai Yu
ynh@ustc.edu.cn

¹ School of Information Science and Technology, University of Science and Technology of China, Hefei, 230027, China

² School of Electronics and Information Engineering, Anhui University, Hefei, 230601, China

1 Introduction

Reversible data hiding (RDH) is one kind of information hiding techniques with the characteristics such that not only the secret message needs to be precisely extracted, but also the cover itself should be restored losslessly. This reversibility is important in some special scenarios such as medical imagery [1], military imagery and law forensics. In these applications, the cover is too precious or too important to be damaged.

So far, many RDH methods on images have been proposed. All these methods are realized through a process of semantic lossless compression [15, 18, 19], in which some space is saved for embedding extra data by lossless compressing the image. This compressed image should be “close” to the original image, so one can get a marked image with good visual quality. The residual part of images, e.g., the prediction errors (PE), has small entropy and thus can be easily compressed. Therefore, almost all recent RDH methods first generate PEs as the host sequence [3, 4, 6, 8, 12, 13], and then reversibly embed the message into the host sequence by modifying its histogram with methods like histogram shifting (HS) [7, 9] or difference expansion (DE) [10, 14]. Usually the PEs has a sharp distribution centered at zero. A more accurate prediction technique can generate PEs with a sharper histogram that is more suitable for RDH.

Many prediction methods have been applied to RDH, such as JPEG-LS prediction [13], rhombus prediction [12], two dimensional prediction [8], and local optimized prediction [3].

Another efficient technique to get a good host sequence for RDH is sorting [12] or pixel selection [6], which gives priority of modifications to PEs in smooth regions. Because the pixels in smooth areas can be accurately predicted, so a sharper histogram can be obtained in such areas. Obviously, from the view of human visual system (HVS), the changes in smooth areas are more noticeable. However, in most literatures on RDH, the quality of the marked image is only assessed by peak signal-to-noise ratio (PSNR), with which a modification in smooth region is equally risky as a modification in noisy region. That is why smoothness-based sorting technique is efficient for state-of-the-art RDH schemes. However there is an obviously conflict between such sorting criterion and the HVS model.

Therefore, some researchers proposed to assess RDH schemes according to not only PSNR but also other criteria of HVS such as Just Noticeable Difference (JND) [5, 11]. JND denotes the smallest change in a pixel value that human eyes can perceive. In [5], Jung et al. proposed to predict pixel value in a local causal window and calculated JND in spatial domain, and then, according to JND, selected PEs for embedding data. Qin et al. [11] proposed a similar method but calculated JND in DCT domain. However, both methods in [5] and [11] cannot keep high PSNR and small JND at the same time.

In fact, according to PSNR, one should first embed data into smooth areas, while according to HVS, one should first embed data into noisy areas. To reconcile this contradiction, we here propose a twice-sorting scheme for RDH, in which the pixels are first sorted and divided into several levels according to a smoothness criterion. Within each level, we sort the pixel again with JND, and then embed data into the PEs of pixels according to the finally sorted order. The experimental results show that this novel scheme significantly outperforms previous JND-based methods [5, 11] when assessing them with not only PSNR but also structural similarity (SSIM) [16] and JND.

This paper is organized as follows. In Section 2, after briefly introducing the rhombus predicting model, we elaborate the twice-sorting scheme. The performance of the proposed method is evaluated and compared with the other methods in Section 3, and conclusion is finally presented in Section 4.

2 Proposed method

2.1 Rhombus prediction

In this paper, we use rhombus prediction [12] to produce PEs, so we first briefly introduce rhombus prediction.

As shown in Fig. 1, the rhombus prediction pattern divides all pixels of the cover image into two sets denoted as "Cross" and "Dot". Note that two sets are independent of each other, so it is a twice-layered embedding scheme. In the first layer, we use the pixels of "Dot" set to predict the pixels of "Cross" set, and then embed data into the PEs; in the second layer, we predict the "Dot" set with the modified "Cross" set and embed data into the PEs. Since the two layers' embedding process are similar, we only take the Cross layer for illustration. The prediction value $\hat{u}_{i,j}$ is computed using its four nearest Dot pixels ($v_{i,j-1}, v_{i+1,j}, v_{i,j+1}, v_{i-1,j}$):

$$\hat{u}_{i,j} = \left\lfloor \frac{v_{i,j-1} + v_{i+1,j} + v_{i,j+1} + v_{i-1,j}}{4} \right\rfloor . \tag{1}$$

Based on the prediction value $\hat{u}_{i,j}$ and the original value $u_{i,j}$, the PE $e_{i,j}$ is computed as

$$e_{i,j} = u_{i,j} - \hat{u}_{i,j} . \tag{2}$$

As mentioned before, not all pixels' PEs will be used to embed data. The pixels usually are sorted and selected with some smoothness scores. We will also sort the pixels with JND scores. Both scores on the pixel $u_{i,j}$ will be estimated by its neighboring pixels belonging to the "Dot" set. A framework of the embedding process is shown in Fig. 2.

X	$v_{i-1,j}$	X	O
$v_{i,j-1}$	$u_{i,j}$	$v_{i,j+1}$	X
X	$v_{i+1,j}$	X	O
O	X	O	X

Fig. 1 Rhombus prediction pattern

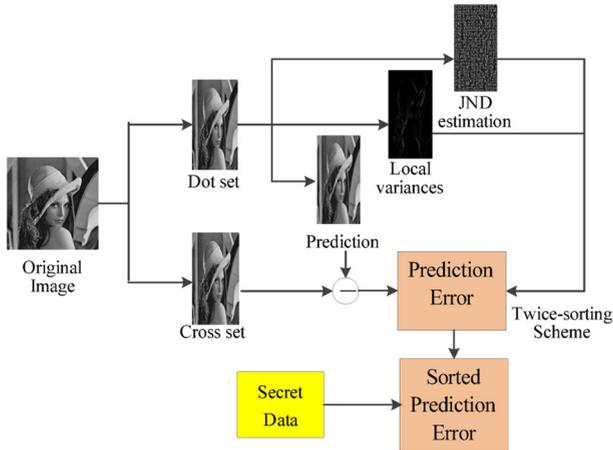


Fig. 2 Framework of cross set’s embedding process

2.2 Smoothness sorting

We measure the smoothness with the local variance (*LV*) [12]. The *LV* for pixel $u_{i,j}$ can be computed from the neighboring pixels $(v_{i,j-1}, v_{i+1,j}, v_{i,j+1}, v_{i-1,j})$ such that

$$LV(u_{i,j}) = \frac{1}{4} \sum_{k=1}^4 (\Delta v_k - \Delta \bar{v}_k)^2, \tag{3}$$

where $\Delta v_1 = |v_{i,j-1} - v_{i-1,j}|$, $\Delta v_2 = |v_{i-1,j} - v_{i,j+1}|$, $\Delta v_3 = |v_{i,j+1} - v_{i+1,j}|$, $\Delta v_4 = |v_{i+1,j} - v_{i,j-1}|$ and $\Delta \bar{v}_k = (\Delta v_1 + \Delta v_2 + \Delta v_3 + \Delta v_4) / 4$.

The pixels are sorted in ascending order of *LV* values. For example, assume that there are ten pixels of “cross set” $\{u_1, \dots, u_{10}\}$ with corresponding *LV*s $\{2.5, 1.25, 5, 2, 6.25, 2.25, 2.5, 0.25, 5.5, 4\}$ which is shown in Fig. 3a. After sorted, the set of pixels is $\{u_8, u_2, u_4, u_6, u_1, u_7, u_{10}, u_3, u_9, u_5\}$.

- LV*: $\{2.5, 1.25, 5, 2, 6.25, 2.25, 2.5, 0.25, 5.5, 4\}$
- Pixel: $\{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}\}$
- (a) Ten pixels and the corresponding *LV*s
- LV*: $\{0.25, 1.25, 2\}, \{2.25, 2.5, 2.5, 4\}, \{5, 5.5\}, \{6.25\}$
- Pixel: $\{u_8, u_2, u_4\}, \{u_6, u_1, u_7, u_{10}\}, \{u_3, u_9\}, \{u_5\}$
- (b) Smoothness sorting with interval length $P = 2$
- $\hat{j}(u_{i,j})$: $\{4.1, 7.8, 3.1\}, \{5.1, 5.7, 5.7, 5.3\}, \{7.2, 7.4\}, \{5.1\}$
- Pixel: $\{u_8, u_2, u_4\}, \{u_6, u_1, u_7, u_{10}\}, \{u_3, u_9\}, \{u_5\}$
- (c) Assign a JND estimation value to each pixel
- Pixel: $\{u_2, u_8, u_4, u_1, u_7, u_{10}, u_6, u_9, u_3, u_5\}$
- (d) Sorting in each level according to JND

Fig. 3 Example of the twice-sorting scheme

Furthermore, we divide the pixels into several levels according to *LVs*. To do that, we first divide the *LVs* into some intervals with equi-length. By denoting the interval length with P , the *LV* intervals have the following form: $[LV_{\min}, LV_{\min} + P)$, $[LV_{\min} + P, LV_{\min} + 2P)$ In the above example, we take $P = 2$, and thus the *LV* intervals are $[0.25, 2.25)$, $[2.25, 4.25)$, $[4.25, 6.25)$, $[6.25, 8.25)$. The sets of *LVs* in these intervals are $\{0.25, 1.25, 2\}$, $\{2.25, 2.5, 2.5, 4\}$, $\{5, 5.5\}$, $\{6.25\}$, according to which the 10 pixels are divided into four levels such that $\{u_8, u_2, u_4\}$, $\{u_6, u_1, u_7, u_{10}\}$, $\{u_3, u_9\}$, $\{u_5\}$ as shown in Fig. 3b.

2.3 JND sorting

In order to keep the quality of marked image according to the HVS model, we further sort the pixels with JND. The JND value is proportion to the limitation of human perception, so the pixels with bigger JND value are better for data hiding. Therefore, pixels are sorted in descending order of the JND value. Here, JND is derived from the frequency domain by discrete cosine transform (DCT) [17] which is also used in Qin et al’s method [11]. We use the JND of pixels in “dot set” to estimate the JND of pixels in “cross set”. To do that, we first sample all pixels of “dot set” and get a sub-image denoted as I_v . As show in Fig. 4. The procedure of estimating the JND of “cross set” is as follows.

- 1) Cover image is divided into cross set $\{u_{i,j}\}$ and dot set $\{v_{i,j}\}$. Collect all $v_{i,j}$ ’s to get the sub-image I_v as $V_{i,j}$.
- 2) Divide I_v into 8×8 non-overlapping blocks $V_{i,j}^{(t)}$. Conducting DCT on each block and calculate DCT coefficient matrix $C_{i,j}^{(t)} = DCT(V_{i,j}^{(t)})$.
- 3) Each block’s DCT coefficient $C_{i,j}^{(t)}$ adds with corresponding element of Watson matrix that represents the largest tolerable variation of each DCT coefficient to achieve largest imperceptible degree [11]. It can be expressed as

$$C'_{i,j}{}^{(t)} = \left[|C_{i,j}^{(t)}| + W_{i,j} \right] \cdot \text{sign} \left[C_{i,j}^{(t)} \right], \tag{4}$$

where $W_{i,j}$ is taken from the following Watson matrix [17]

$$W = \begin{pmatrix} 1.40 & 1.01 & 1.16 & 1.66 & 2.40 & 3.43 & 4.79 & 6.56 \\ 1.01 & 1.45 & 1.32 & 1.52 & 2.00 & 2.71 & 3.67 & 4.93 \\ 1.16 & 1.32 & 2.24 & 2.59 & 2.98 & 3.64 & 4.60 & 5.88 \\ 1.66 & 1.52 & 2.59 & 3.77 & 4.55 & 5.30 & 6.28 & 7.60 \\ 2.40 & 2.20 & 2.98 & 4.55 & 6.15 & 7.46 & 8.71 & 10.17 \\ 3.43 & 2.71 & 3.64 & 5.30 & 7.46 & 9.62 & 11.58 & 13.51 \\ 4.79 & 3.67 & 4.60 & 6.28 & 8.71 & 11.58 & 14.50 & 17.29 \\ 6.56 & 4.93 & 5.88 & 7.60 & 10.17 & 13.51 & 17.29 & 21.15 \end{pmatrix} \tag{5}$$

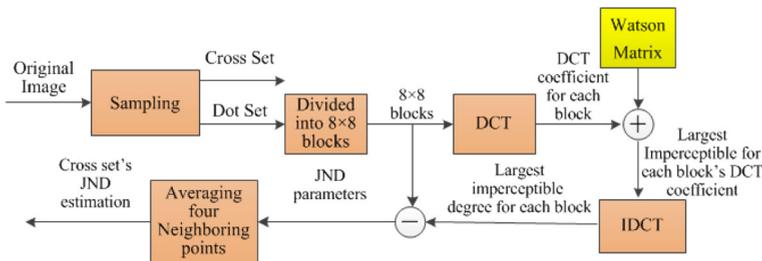


Fig. 4 Procedure of calculating cross set’s JND estimation

- 4) Largest imperceptible degree for each block is calculated by IDCT such that $V_{i,j}^{(t)} = IDCT(C_{i,j}^{(t)})$. Then, combine all block's $V_{i,j}^{(t)}$ to $V'_{i,j}$.
- 5) JND value of pixel $v_{i,j}$ of sub-image I_v is calculated by $\hat{J}(v_{i,j}) = |V'_{i,j} - V_{i,j}|$.
- 6) Finally the JND of pixel $u_{i,j}$ in cross set is estimated by the JND of pixels in the dot set around $u_{i,j}$. Taking Fig. 1 as an example, the JND of $u_{i,j}$ is estimated by

$$\hat{J}(u_{i,j}) = \frac{\hat{J}(v_{i,j-1}) + \hat{J}(v_{i-1,j}) + \hat{J}(v_{i,j+1}) + \hat{J}(v_{i+1,j})}{4} \tag{6}$$

Following the example in Fig. 3, we assign each pixel in Fig. 3b a JND value. Assume that the ten pixels have JND values {4.1, 7.8, 3.1}, {5.1, 5.7, 5.7, 5.3}, {7.2, 7.4}, {5.1} as shown in Fig. 3c. In each level, we sort the pixels according to JND values in descending order. The finally sorted order is { $u_2, u_8, u_4, u_1, u_7, u_{10}, u_6, u_9, u_3, u_5$ } as shown in Fig. 3d.

After achieving sorted index of pixels, we embed data into PEs of these pixels one by one according to the order.

2.4 Data embedding

To embedding data into the PEs with expansion technique [13], we first set two threshold values T_1 and T_2 . PEs belonging to $[T_1, T_2]$ can be expandable for embedding message bits, and those not belonging to $[T_1, T_2]$ are going to be shifted to make vacancies for expansion. The message bit $m \in \{0, 1\}$ is embedded with following manner

$$D_{i,j} = \begin{cases} 2e_{i,j} + m & \text{if } e_{i,j} \in [T_1, T_2] \\ e_{i,j} + T_2 + 1 & \text{if } e_{i,j} > T_2 \geq 0 \\ e_{i,j} + T_1 & \text{if } e_{i,j} < T_1 < 0 \end{cases}, \tag{7}$$

where $e_{i,j}$ is the PE of pixel $u_{i,j}$ and $D_{i,j}$ is the modified PE.

After embedding data, the cover pixel value $u_{i,j}$ is modified to $U_{i,j}$ such as

$$U_{i,j} = D_{i,j} + \hat{u}_{i,j}. \tag{8}$$

Note that, as other RDH schemes, we also use a location map to record the positions of overflow/underflow such that a pixel value, equal to $0/255$, is needed to be changed to $-1/256$. The location map is compressed and its size is N_{flow} . In addition, in order to extract message and recover cover image conveniently, proposed method records the least significant bits (LSB) of the first $48+N_{flow}$ pixels to obtain a binary sequence S_{LSB} , and then replaces LSB of the first $48+N_{flow}$ pixels by the auxiliary information as: two threshold values T_1 and T_2 , interval length P , payload size, size of compressed location map N_{flow} and compressed location map. Here, the LSB of the first $48+N_{flow}$ pixels S_{LSB} is also embedded as one part of the payload.

2.5 Data extraction and cover restoration

Due to using the rhombus prediction pattern, two sets (Cross set and Dot set) are independent. Hence, the double decoding scheme is the inverse of the double encoding scheme, namely, if Cross set is embedded firstly and the Dot set is recovered firstly. Computations for the Dot and Cross decoding schemes are similar. We just take Cross set for example.

1. Read LSB of the first 48 pixels of marked image to get the values of two threshold values T_1 and T_2 , interval length P , payload size, size of compressed location

map N_{flow} . Then, read LSB of the next N_{flow} pixels of marked image to obtain the compressed location map.

2. The same results of prediction value, local variance and JND value for the modified pixels can be acquired at the receiver side similarly as embedding process.
3. According to the interval length P , we can reproduce twice sorted order similarly as embedding process. Combined with payload size, the receiver can one by one calculate all modified PE by

$$D_{i,j} = U_{i,j} - \hat{u}_{i,j} \tag{9}$$

4. According to modified PE $D_{i,j}$, all embedded message bits are calculated as

$$m = D_{i,j} \bmod 2 \tag{10}$$

in which the last $48+N_{flow}$ bits are the S_{LSB} .

5. According to the two threshold values T_1 and T_2 , the PE is recovered as

$$e_{i,j} = \begin{cases} \lfloor D_{i,j}/2 \rfloor & \text{if } D_{i,j} \in [2T_1, 2T_2 + 1] \\ D_{i,j} - T_2 - 1 & \text{if } (D_{i,j} > 2T_2 + 1) \& (T_2 \geq 0) \\ D_{i,j} - T_1 & \text{if } (D_{i,j} < 2T_1) \& (T_1 < 0) \end{cases} \tag{11}$$

6. The pixel is recovered as

$$u_{i,j} = \hat{u}_{i,j} + e_{i,j} \tag{12}$$

7. Replace the LSB of the first $48+N_{flow}$ pixels by the sequence S_{LSB} which are extracted in step 4.

3 Experimental results

In order to evaluate visual quality of marked image, two image quality metrics, i.e., PSNR and SSIM [16] were firstly used in this paper. PSNR is a traditional image quality metric which depends on the quadratic sum of difference between original image and marked image and it doesn't consider the characteristics of the HVS. As mentioned in Section 1, PSNR only calculates the pixel intensity difference between two images, and smoothness-priority-based sorting technique is efficient for improving PSNR.

The SSIM considers the structural characteristic of the image and includes structure, luminance, and contrast comparison functions [16]. The essence of the structural comparison function in SSIM is the cosine value of the angle between two images, which reflects the image structure sensitivity characteristic of the human visual perception. It has been proved that SSIM was developed from the characteristics of the HVS. Thus SSIM and JND are both based on HVS model and are both prefer to embedding data into texture area.

In addition, we establish a new image quality metric based on JND. As described above, JND presents the smallest change in a pixel value that the human eyes can perceive. It means the higher the JND value, the larger the tolerable for imperceptible by visual perception. We present JND distortion ($JNDD$) as a new image quality metric for RDH, which is defined as

$$JNDD = \frac{1}{M \times N} \sum \frac{|U_{i,j} - u_{i,j}|}{J(u_{i,j})} \tag{13}$$

where $u_{i,j}$ is cover image pixel, $U_{i,j}$ is the modified pixel, $J(u_{i,j})$ is the JND value of the cover image, M and N are the size of cover image. The smaller $JNDD$ means less visual distortion caused by data embedding.

The present twice-sorting scheme can reconcile the contradiction between smoothness-priority-based sorting technique for pursuing high PSNR and texture-priority-based sorting technique based on HVS by adjusting the interval length P . In order to evaluate the impact of interval length P on three image quality metrics, we apply the proposed method to test

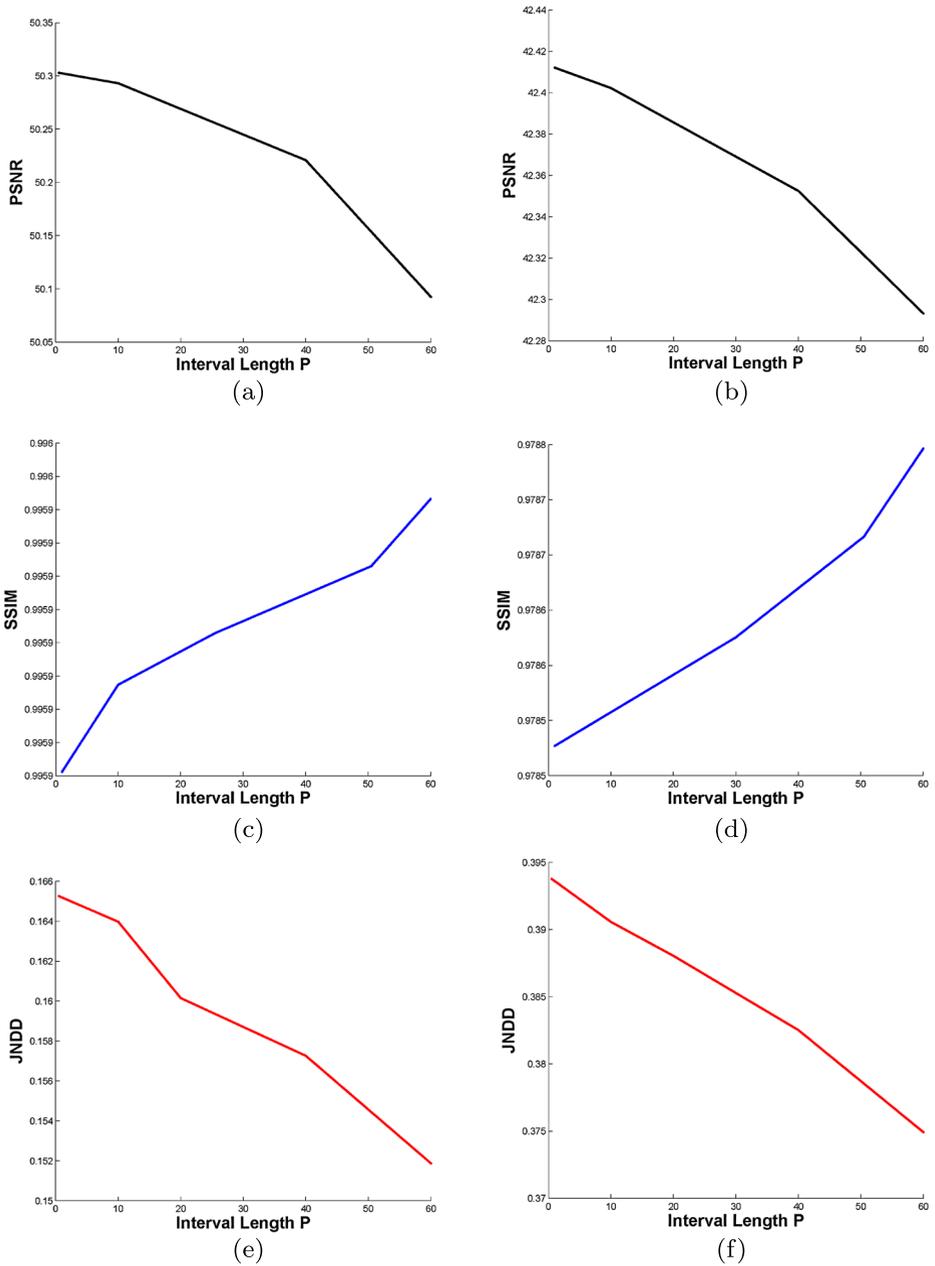


Fig. 5 The impact of interval length P on three image quality metrics, in which Fig (a,c,e) embedding rate is 0.2bpp and Fig (b,d,f) embedding rate is 0.5bpp



(a) Lena



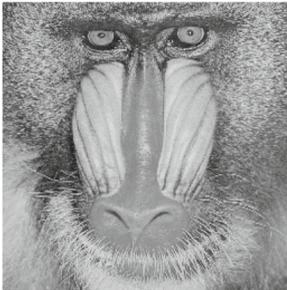
(b) Candy



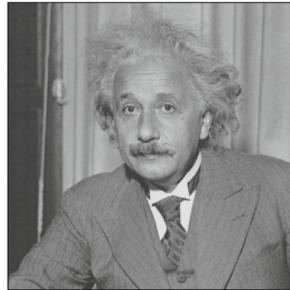
(c) Peppers



(d) Lake



(e) Baboon



(f) Einstein



(g) Boat



(h) Airplane

Fig. 6 Eight test images

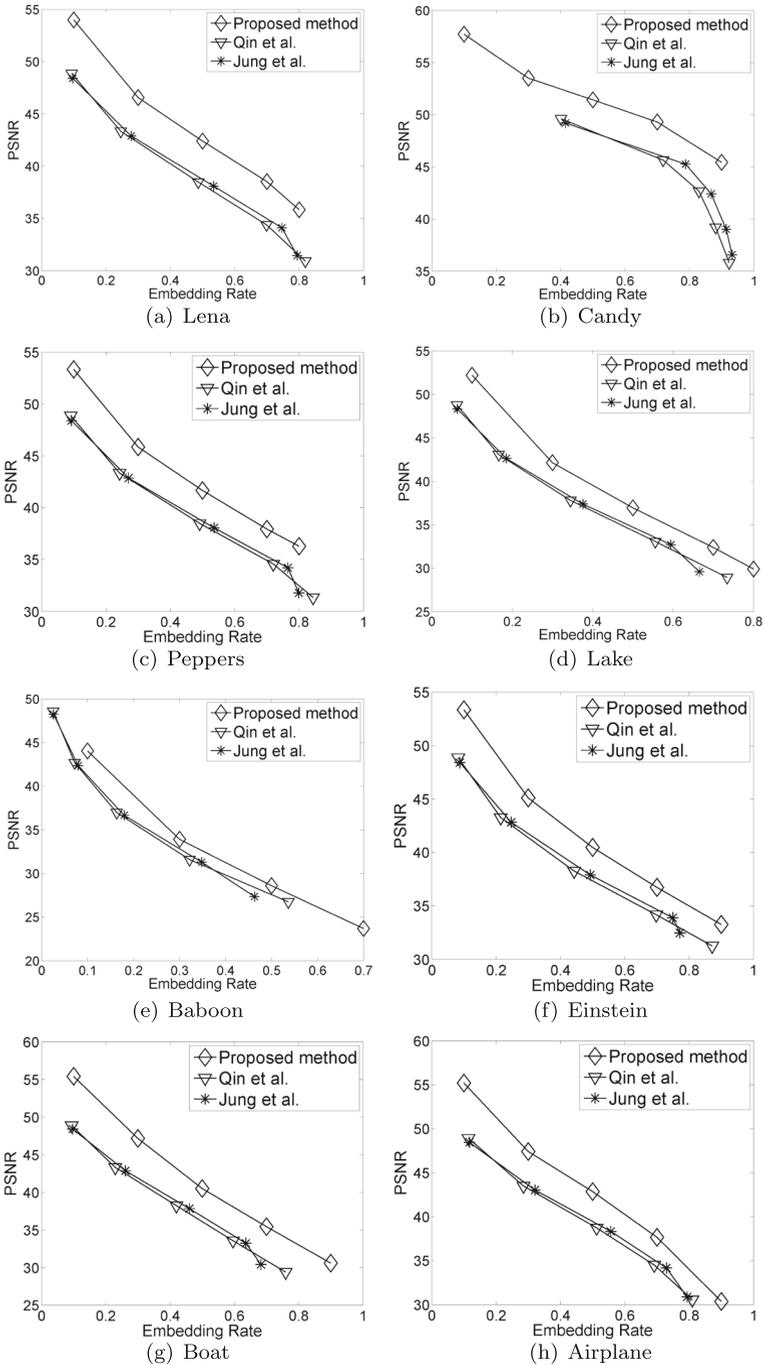


Fig. 7 Performance comparison using PSNR metric

image "Lena" with different interval length P when embedding rate is 0.2bpp and 0.5bpp respectively in Fig. 5. One can see from that, as expected, with the increase of the interval

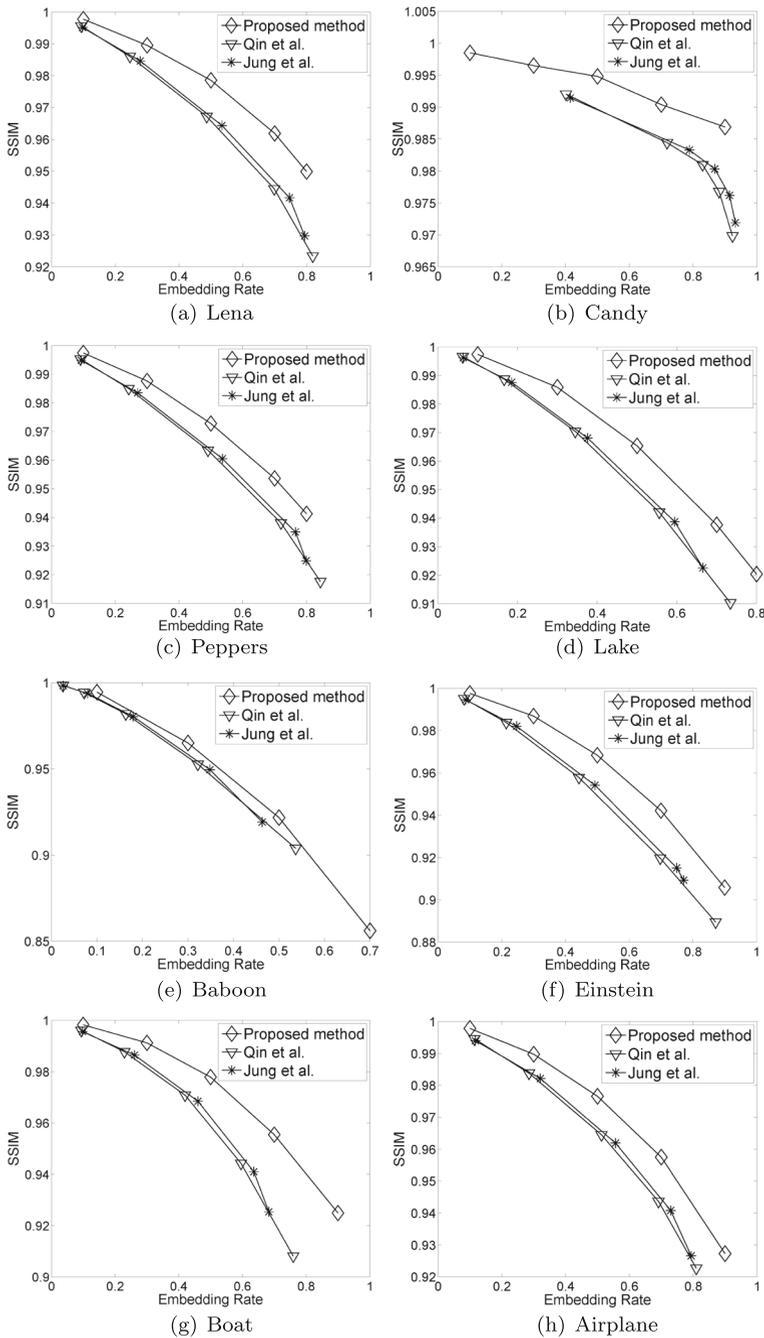


Fig. 8 Performance comparison using SSIM metric

length P , PSNR and JNDD value are decreased while SSIM value is increased. Note that the decreasing of JNDD value means increasing of performance in JNDD image quality

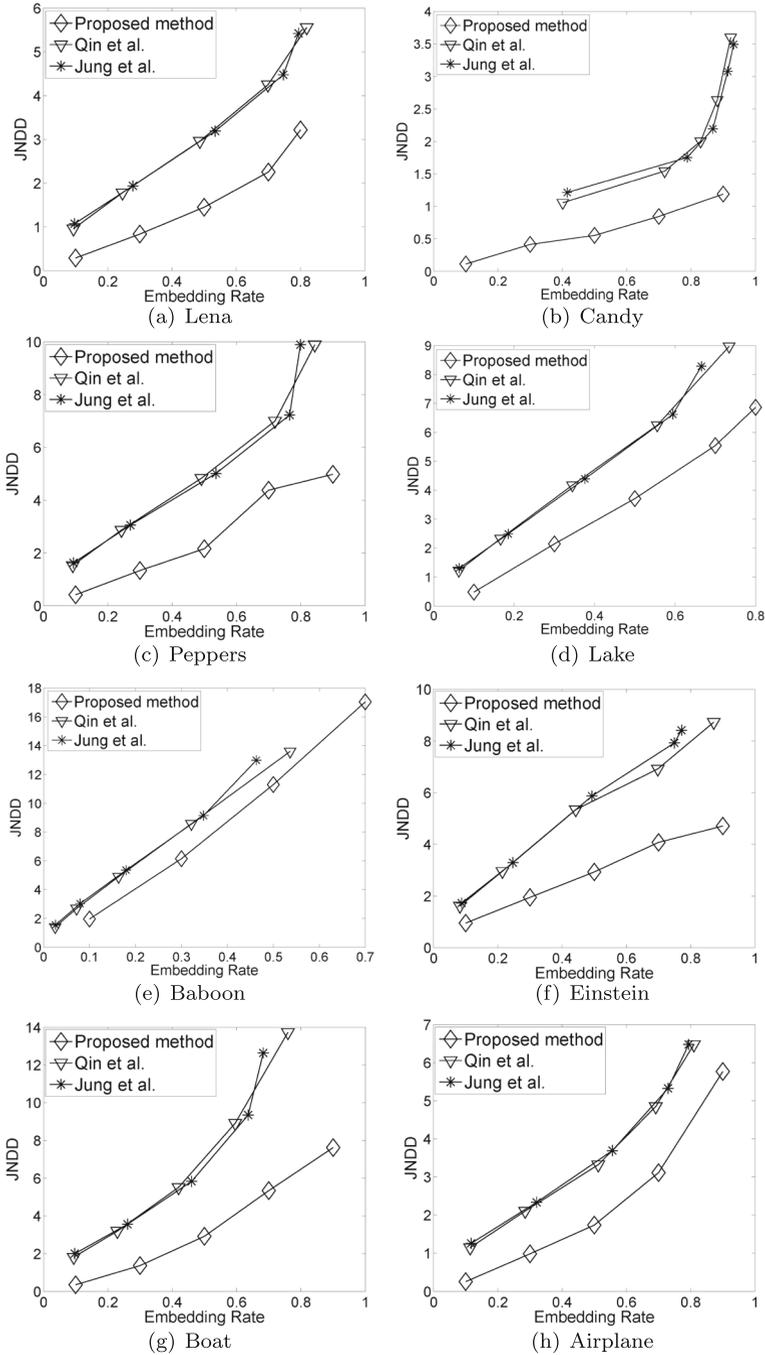


Fig. 9 Performance comparison using JNDD metric

metric. Hence, the result means the performance of JNDD and SSIM are increased while performance of PSNR is decreased. It is consistent with the change trend in Fig. 5. That is

because with the increase of the interval length P , the greater effect of the JND sorting. In this case, data are embedded into texture area preferentially. It is consistent with SSIM and JNDD metrics while contradict with PSNR metric. Hence, interval length P can be chosen according to the image quality metric.

In addition, eight 256×256 sized test images including Lena, Candy, Peppers, Lake, Baboon, Einstein, Boat and Airplane are also used in our experiments (see Fig. 6) [2]. In order to demonstrate the performance, the proposed method is compared with two JND-related RDH methods: Jung et al.'s method [5], and Qin et al.'s method [11].

Both Jung et al.'s and Qin et al.'s methods used the causal window to calculate prediction value. They proved that causal window size $B=3$ and edge threshold $T=200$ are optimal choice. Therefore, we set $B=3$ and $T=200$ in the comparison experiments. In our proposed method, expansion region $[T_1, T_2]$ is empirically determined as $[-1, 0]$ as initial values and it can be changed according to the embedding capacity [12], twice-sorting interval P is empirically determined as 0.5 by considering both the running time and experiment effect. Here, three image quality metrics of PSNR, SSIM and JNDD were also used in comparative experiment.

Figures 7, 8 and 9 are the comparison results of embedding rate versus PSNR, SSIM and JNDD respectively. For all images, the proposed method outperforms Jung et al.'s and Qin et al.'s methods in PSNR, SSIM and JNDD.

4 Conclusion

So far, most RDH schemes are only assessed by PSNR. To improve PSNR, smooth regions are more suitable for data embedding and thus smoothness-based sorting technique is efficient, which however is obviously conflict with other criterion of HVS such as JND. To make a reasonable tradeoff between PSNR and JND, we propose a twice-sorting scheme, in which the pixels are first sorted based on a smoothness criterion and then sorted based on JND. The experiment results show that the proposed method can significantly outperforms previous JND-related methods.

Acknowledgments The authors sincerely thank the editors and anonymous reviewers for their valuable comments and insightful suggestions to improve the paper quality.

This work was supported in part by the Natural Science Foundation of China under Grant 61170234, by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA06030600, by the Funding of Science and Technology on Information Assurance Laboratory under Grant KJ-13-02, by the Doctoral Scientific Research Foundation of Anhui University under Grant J01001319, and by the Backbone Teacher Training Program of Anhui University.

References

1. Bao F, Deng RH, Ooi BC et al (2005) Tailored reversible watermarking schemes for authentication of electronic clinical atlas. *IEEE Trans Inf Technol Biomed* 9(4):554–563
2. Computer Vision Group, University of Granada. CVG-UGR Image Database. <http://decsai.ugr.es/cvg/dbimagenes/g256.php>
3. Dragoi I-C, Coltuc D (2014) Local-prediction-based difference expansion reversible watermarking. *IEEE Trans Image Process* 23(4):1779–1790
4. Hu X, Zhang W, Hu X et al (2013) Fast estimation of optimal marked-signal distribution for reversible data hiding. *IEEE Trans Inf Forensic Secur* 8(5):779–788
5. Jung SW, Ha LT, Ko SJ (2011) A new histogram modification based reversible data hiding algorithm considering the human visual system. *IEEE Signal Process Lett* 18(2):95–98

6. Li X, Yang B, Zeng T (2011) Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Trans Image Process* 20(12):3524–3533
7. Ni Z, Shi Y, Ansari N, Wei S (2006) Reversible Data Hiding. *IEEE Trans Circ Syst Video Technol* 16(3):354–362
8. Ou B, Li X, Zhao Y, Ni R, Shi Y (2013) Pairwise prediction-error expansion for efficient reversible data hiding. *IEEE Trans Image Process* 22(12):5010–5012
9. Qin C, Chang CC, Huang YH et al (2013) An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism. *IEEE Trans Circ Syst Video Technol* 23(7):1109–1118
10. Qin C, Chang CC, Liao LT (2012) An adaptive prediction-error expansion oriented reversible information hiding scheme. *Pattern Recogn Lett* 33(16):2166–2172
11. Qin C, Chang CC, Lin CC (2013) An adaptive reversible steganographic scheme based on the just noticeable distortion, multimedia tools and applications. doi:[10.1007/s11042-013-1733-0](https://doi.org/10.1007/s11042-013-1733-0)
12. Sachnev V, Kim HJ, Nam J, Suresh S, Shi YQ (2009) Reversible watermarking algorithm using sorting and prediction. *IEEE Trans Circ Syst Video Technol* 19(7):989–999
13. Thodi D, Rodriguez J (2007) Expansion embedding techniques for reversible watermarking. *IEEE Trans Image Processing* 16(3):721–730
14. Tian J (2003) Reversible data embedding using a difference expansion. *IEEE Trans Circ Syst Video Technol* 13(8):890–896
15. Willems F, Maas D, Kalker T (2004) Semantic Lossless Source Coding, 42nd Annual Allerton Conference on Communication, Control and Computing, Illinois, USA, pp 1411–1418
16. Wang Z, Bovik AC, Sheikh HR et al (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612
17. Watson AB (1993) DCT Quantization Matrices Visually Optimized for Individual Images. Proceedings of SPIE Human Vision, Visual Processing, and Digital Display IV, Bellingham, WA, pp 202–216
18. Zhang WM, Hu XC, Yu NH et al (2013) Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression. *IEEE Trans Image Process* 22(7):2775–2785
19. Zhang WM, Chen B, Yu NH et al (2012) Improving various reversible data hiding schemes via optimal codes for binary covers. *IEEE Trans Image Process* 21(6):2991–3003



Yang Yang received her M.S. degree and PH.D. degree in 2007 and 2013 respectively from Anhui University and University of Science and Technology of China. She currently as a postdoctoral researcher with the University of Science and Technology of China, and she is also a lecturer with the Anhui University. Her research interests include reversible information hiding and image quality assessment.



Weiming Zhang received his M.S. degree and PH.D. degree in 2002 and 2005 respectively from the Zhengzhou Information Science and Technology Institute, Zhengzhou, China. Currently, he is an associate professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include information hiding and cryptography.



Xiaocheng Hu received his B.S. degree in 2010 from University of Science and Technology of China, where he is now pursuing the Ph.D. degree in University of Science and Technology of China. His research interests include multimedia security, image and video processing, video compression and information hiding.



Nenghai Yu received his B.S. degree in 1987 from Nanjing University of Posts and Telecommunications, M.E. degree in 1992 from Tsinghua University and Ph.D. degree in 2004 from University of Science and Technology of China, where he is currently a professor. His research interests include multimedia security, multimedia information retrieval, video processing and information hiding.