Reversible Image Data Hiding with Contrast Enhancement

Hao-Tian Wu, *Member, IEEE*, Jean-Luc Dugelay, *Fellow, IEEE*, and Yun-Qing Shi, *Fellow, IEEE*

***Abstract*—In this letter, a novel reversible data hiding (RDH) algorithm is proposed for digital images. Instead of trying to keep the PSNR value high, the proposed algorithm enhances the contrast of a host image to improve its visual quality. The highest two bins in the histogram are selected for data embedding so that histogram equalization can be performed by repeating the process. The side information is embedded along with the message bits into the host image so that the original image is completely recoverable. The proposed algorithm was implemented on two sets of images to demonstrate its efficiency. To our best knowledge, it is the first algorithm that achieves image contrast enhancement by RDH. Furthermore, the evaluation results show that the visual quality can be preserved after a considerable amount of message bits have been embedded into the contrast-enhanced images, even better than three specific MATLAB functions used for image contrast enhancement.**

***Index Terms*—Contrast enhancement, histogram modification, location map, reversible data hiding, visual quality.**

I. Introduction

REVERSIBLE DATA HIDING (RDH) has been intensively studied in the community of signal processing. Also referred as invertible or lossless data hiding, RDH is to embed a piece of information into a host signal to generate the marked one, from which the original signal can be exactly recovered after extracting the embedded data. The technique of RDH is useful in some sensitive applications where no permanent change is allowed on the host signal. In the literature, most of the proposed algorithms are for digital images to embed invisible data (e.g. [1]–[8]) or a visible watermark (e.g. [9]).

To evaluate the performance of a RDH algorithm, the hiding rate and the marked image quality are important metrics. There exists a trade-off between them because increasing the hiding rate often causes more distortion in image content. To measure the distortion, the peak signal-to-noise ratio (PSNR) value of the marked image is often calculated. Generally speaking, direct modification of image histogram [2] provides less embedding capacity. In contrast, the more recent algorithms (e.g. [5]–[8]) manipulate the more centrally distributed prediction errors by exploiting the correlations between neighboring pixels so that less distortion is caused by data hiding.

Although the PSNR of a marked image generated with a prediction error based algorithm is kept high, the visual quality can hardly be improved because more or less distortion has been introduced by the embedding operations. For the images acquired with poor illumination, improving the visual quality is more important than keeping the PSNR value high. Moreover, contrast enhancement of medical or satellite images is desired to show the details for visual inspection. Although the PSNR value of the enhanced image is often low, the visibility of image details has been improved. To our best knowledge, there is no existing RDH algorithm that performs the task of contrast enhancement so as to improve the visual quality of host images. So in this study, we aim at inventing a new RDH algorithm to achieve the property of contrast enhancement instead of just keeping the PSNR value high.

In principle, image contrast enhancement can be achieved by histogram equalization [10]. To perform data embedding and contrast enhancement at the same time, the proposed algorithm is performed by modifying the histogram of pixel values. Firstly, the two peaks (i.e. the highest two bins) in the histogram are found out. The bins between the peaks are unchanged while the outer bins are shifted outward so that each of the two peaks can be split into two adjacent bins. To increase the embedding capacity, the highest two bins in the modified histogram can be further chosen to be split, and so on until satisfactory contrast enhancement effect is achieved. To avoid the overflows and underflows due to histogram modification, the bounding pixel values are pre-processed and a location map is generated to memorize their locations. For the recovery of the original image, the location map is embedded into the host image, together with the message bits and other side information. So blind data extraction and complete recovery of the original image are both enabled. The proposed algorithm was applied to two set of images to demonstrate its efficiency. To our best knowledge, it is the first algorithm that achieves image contrast enhancement by RDH. Furthermore, the evaluation results show that the visual quality can be preserved after a considerable amount of message bits have been embedded into the contrast-enhanced images, even better than three specific MATLAB functions used for image contrast enhancement.

The rest of this letter is organized as follows. Section II presents the details of the proposed RDH algorithm featured by contrast enhancement. The experimental results are given in Section III. Finally, a conclusion is drawn in Section IV.

Manuscript received June 14, 2014; revised July 19, 2014; accepted August 01, 2014. Date of publication August 13, 2014; date of current version August 26, 2014. This work was supported in part by the NSFC under Grant 61100169 and by the Fundamental Research Funds for the Central Universities of China. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jing-Ming Guo.

H.-T. Wu is with School of Digital Media, Jiangnan University, Wuxi, JS 214122, China, on leave at the Polytechnic School of Engineering, New York University, Brooklyn, NY 11201 USA (with the support of China Scholarship Council) (e-mail: htwu@jiangnan.edu.cn).

J.-L. Dugelay is with the Department of Multimedia Communications, EURECOM, F-06410 Biot, Sophia Antipolis, France (e-mail: jld@eurecom.fr).

Y.-Q. Shi is with Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07103 USA (e-mail: shi@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2014.2346989

II. RDH Algorithm With Contrast Enhancement

A. Data Embedding by Histogram Modification

The algorithm to be presented is primarily for gray-level images but can be easily extended to color images. Given an 8-bit gray-level image I, the image histogram can be calculated by counting the pixels with a gray-level value j for $j \in \{0, 1, \cdots, 254, 255\}$. We use h_I to denote the image histogram so that $h_I(j)$ represents the number of pixels with a value j. Suppose I consists of N different pixel values. Then there are N nonempty bins in h_I, from which the two peaks (i.e. the highest two bins) are chosen and the corresponding smaller and bigger values are denoted by I_S and I_R, respectively. For a pixel counted in h_I with value i, data embedding is performed by

$$i' = \begin{cases} i-1, & \text{for } i < I_S \\ I_S - b_k, & \text{for } i = I_S \\ i, & \text{for } I_S < i < I_R \\ I_R + b_k, & \text{for } i = I_R \\ i+1, & \text{for } i > I_R, \end{cases} \tag{1}$$

where i' is the modified pixel value, and b_k is the k-th message bit (0 or 1) to be hidden. By applying Eq. (1) to every pixel counted in h_I, totally $h_I(I_S) + h_I(I_R)$ binary values are embedded. Given that there is no bounding value (0 or 255) in I (otherwise pre-process is needed), there will be $N + 2$ bins in the modified histogram. That is, the bins between the two peaks are unchanged while the outer ones are shifted outward so that each of the peaks can be split into two adjacent bins (i.e. $I_S - 1$ and I_S, I_R and $I_R + 1$, respectively).

The peak values I_S and I_R need to be provided to extract the embedded data. One way to keep them is to exclude 16 pixels in I from histogram computing. The least significant bits (LSB) of those pixels are collected and included in the binary values to be hidden. After applying Eq. (1) to each pixel counted in h_I for data embedding, the values of I_S and I_R (each with 8 bits) are used to replace the LSBs of the 16 excluded pixels by bitwise operation. To extract the embedded data, the peak values need to be retrieved and the histogram of the marked image I' is calculated excluding the 16 pixels aforementioned. Then the following operation is performed on any pixel counted in the histogram and with the value of $I_S - 1$, I_S, I_R or $I_R + 1$:

$$b'_k = \begin{cases} 1, & \text{if } i' = I_S - 1 \\ 0, & \text{if } i' = I_S \\ 0, & \text{if } i' = I_R \\ 1, & \text{if } i' = I_R + 1, \end{cases} \tag{2}$$

where b'_k is the k-th binary value extracted from the marked image I'. The extraction operations are performed in the same order as that of the embedding operations. According to Eq. (1), the following operation is performed on every pixel counted in the histogram to recover its original value:

$$i = \begin{cases} i'+1, & \text{for } i' < I_S - 1 \\ I_S, & \text{for } i' = I_S - 1 \text{ or } i' = I_S \\ I_R, & \text{for } i' = I_R \text{ or } i' = I_R + 1 \\ i'-1, & \text{for } i' > I_R + 1 \end{cases}. \tag{3}$$

The original LSBs of 16 excluded pixels are obtained from the extracted binary values. The excluded pixels can be restored by writing them back so as to recover the original image.

B. Pre-Process for Complete Recovery

In the aforementioned algorithm, it is required that all pixels counted in h_I are within $\{1, \cdots, 254\}$. If there is any bounding pixel value (0 or 255), overflow or underflow will be caused by histogram shifting. To avoid it, the histogram needs to be pre-processed prior to the histogram modification operations. Specifically, the pixel values of 0 and 255 are modified to 1 and 254, respectively. Therefore, no overflow or underflow will be caused because the possible change of each pixel value is ± 1. To memorize the pre-processed pixels, a location map with the same size as the original image is generated by assigning 1 to the location of a modified pixel, and 0 to that of an unchanged one (including the 16 excluded pixels). The location map can be pre-computed and included into the binary values to be hidden. In the extraction and recovery process, it can be obtained from the data extracted from the marked image so that the pixels modified in the pre-process can be identified. By restoring the original values of those pixels accordingly, the original image can be completely recovered.

C. Contrast Enhancement

In Section II-A, each of the two peaks in the histogram is split into two adjacent bins with the similar or same heights because the numbers of 0s and 1s in the message bits are required to be almost equal. To increase the hiding rate, the highest two bins in the *modified* histogram are further chosen to be split by applying Eq. (1) to all pixels counted in the histogram. The same process can be repeated by splitting each of the two peaks into two adjacent bins with the similar heights to achieve the *histogram equalization* effect. In this way, data embedding and contrast enhancement are simultaneously performed. Given that the pair number of the histogram peaks to be split is L, the range of pixel values from 0 to $L-1$ are added by L while the pixels from $256 - L$ to 255 are subtracted by L in the pre-process (noting L is a positive integer). A location map is generated by assigning 1s to the modified pixels, and 0s to the others.

The location map can be pre-computed and compressed to be firstly embedded into the host image. The value of L, the size of the compressed location map, and the previous peak values, in contrary, are embedded with the last two peaks to be split, whose values are stored in the LSBs of the 16 excluded pixels. In the extraction process, the last split peak values are retrieved and the data embedded with them are extracted with Eq. (2). After restoring the histogram with Eq. (3), the data embedded with the previously split peaks can also be extracted by processing them pair by pair. At last, the location map is obtained from the extracted data to identify the pixel values modified in the pre-process.

D. Procedure of the Proposed Algorithm

The procedure of the proposed algorithm is illustrated in Fig. 1. Given that totally L pairs of histogram bins are to be split for data embedding, the **embedding** procedure includes the following steps:

1) Pre-process: The pixels in the range of $[0, L-1]$ and $[256 - L, 255]$ are processed as mentioned in Section II-C excluding the first 16 pixels in the bottom row. A location map is generated to record the locations of those pixels

and compressed by the JBIG2 standard [11] to reduce its length.

2) The image histogram is calculated without counting the first 16 pixels in the bottom row.
3) Embedding: The two peaks (i.e. the highest two bins) in the histogram are split for data embedding by applying Eq. (1) to every pixel counted in the histogram. Then the two peaks in the *modified* histogram are chosen to be split, and so on until L pairs are split. The bitstream of the compressed location map is embedded before the message bits (binary values). The value of L, the length of the compressed location map, the LSBs collected from the 16 excluded pixels, and the previous peak values are embedded with the last two peaks to be split.
4) The lastly split peak values are used to replace the LSBs of the 16 excluded pixels to form the marked image.

The **extraction** and **recovery** process include the following steps:

1) The LSBs of the 16 excluded pixels are retrieved so that the values of the last two split peaks are known.
2) The data embedded with the last two split peaks are extracted by using Eq. (2) so that the value of L, the length of the compressed location map, the original LSBs of 16 excluded pixels, and the previously split peak values are known. Then the recovery operations are carried out by processing all pixels except the 16 excluded ones with Eq. (3). The process of extraction and recovery is repeated until all of the split peaks are restored and the data embedded with them are extracted.
3) The compressed location map is obtained from the extracted binary values and decompressed to the original size.
4) With the decompressed map, those pixels modified in preprocess are identified. Among them, a pixel value is subtracted by L if it is less than 128, or increased by L otherwise. To comply with this rule, the maximum value of L is 64 to avoid ambiguity. At last, the original image is recovered by writing back the original LSBs of 16 excluded pixels.

III. Experimental Results

In the experiments, 8 USC-SIPI test images with the size of 512×512 [12] and 24 Kodak test images with the size of 768×512 [13] were employed and converted into grey-level images. The only parameter in the proposed algorithm is L, i.e. the pair number of histogram peaks to be split. The message bits to be hidden can be any string of binary values in which the numbers of 0s and 1s are almost equal, or some extra bits can be appended to make so. As shown in Fig. 2, the pure hiding rates were generally increased by using more histogram peaks for data embedding. For all test images, blind extraction and complete recovery were achieved for any $L \in [1, 64]$. When 64 pairs of histogram peaks were split, the pure hiding rate was 1.536 bit per pixel (bpp) for F-16, 0.732 bpp for Baboon, and averagely 1.085 bpp for the 8 USC-SIPI images, while the average for the 24 Kodak images was 1.194 bpp. It should be noted that the hiding rates were calculated by subtracting the bit number

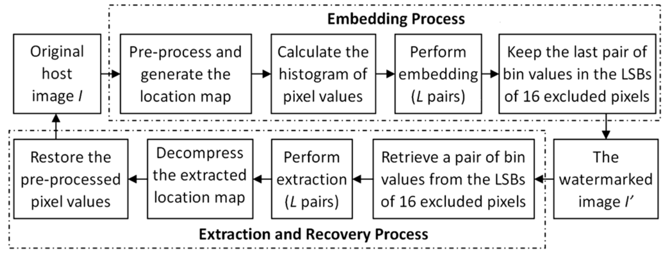

Fig. 1. Procedure of the proposed RDH algorithm.

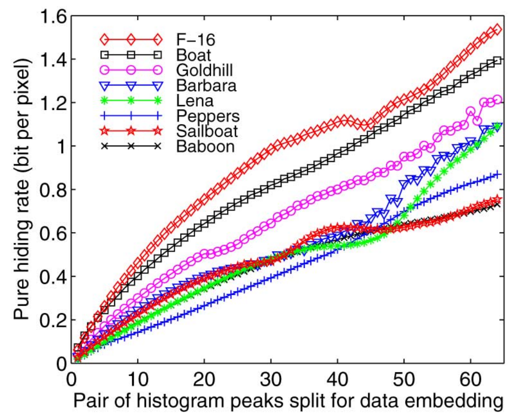

Fig. 2. The hiding rates generally increase with the number of histogram peaks used for data embedding.

of the side information for recovery (including the compressed location map) from the total amount of the embedded bits.

The original and marked images of "Lena" and "Goldhill" are shown in Fig. 3 and Fig. 4, respectively. The marked images were obtained by splitting 10, 15 and 20 pairs of histogram peaks for data embedding, respectively. It can be seen that the embedded data were invisible in the contrast-enhanced images. The more histogram peaks were split for data embedding, the more contrast enhancement effect was obtained. Although the PSNR value of the contrast-enhanced images decrease with the data hiding rate, the visual quality has been preserved, as shown in Fig. 3 and Fig. 4.

Besides the PSNR value, the relative contrast error (RCE), relative entropy error (REE), relative mean brightness error (RMBE) and relative structural similarity (RSS) used in [14] were calculated between the original and contrast-enhanced images to evaluate the enhancement effect and image quality. The RCE and REE values greater than 0.5 indicate the enhanced contrast and increased image data, respectively. The less difference in mean brightness from the original image, the closer RMBE is to 1. The greater the structural similarity between them, the closer RSS is to 1. We further compare the proposed algorithm with three MATLAB functions used for image contrast enhancement, i.e. *imadjust*, *histeq*, and *adapthisteq*. The MATLAB routines were applied on each test image with the default settings. For each of the contrast-enhanced images, the five evaluation values were calculated, including RCE, REE, RMBE, RSS and PSNR.

Table I and Table II show the statistical results of two sets of test images, respectively. Each item listed in the two tables is the mean of 8 or 24 test images. The proposed algorithm using 10, 15 and 20 pairs of histogram peaks are denoted by *Prop.10p*,

Fig. 3. The original and contrast-enhanced images of "Lena" by splitting 10, 15 and 20 pairs of histogram peaks in the proposed algorithm. (a) Original image of "Lena". (b) 10 pairs: 0.185 bpp, 29.10 dB. (c) 15 pairs: 0.268 bpp, 25.97 dB. (d) 20 pairs: 0.345 bpp, 24.91 dB.

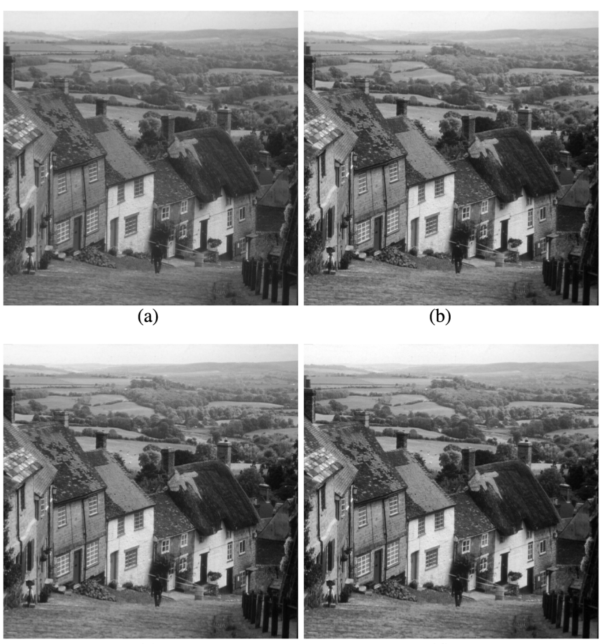

Fig. 4. The original and contrast-enhanced images of "Goldhill" by splitting 10, 15 and 20 pairs of histogram peaks in the proposed algorithm. (a) Original image of "Goldhill". (b) 10 pairs: 0.299 bpp, 30.64 dB. (c) 15 pairs: 0.411 bpp, 26.92 dB. (d) 20 pairs: 0.506 bpp, 24.64 dB.

Prop.15p and *Prop.20p*, respectively. It can be seen that the contrast of test images was gradually enhanced by splitting more histogram peaks in the proposed algorithm but more differences were introduced in brightness and structural similarity. With *Prop.20p*, the obtained RCEs are lower than those of *imadjust* and *histeq*, but higher than *adapthisteq*, indicating more contrast enhancement effect was obtained than *adapthisteq*. Meanwhile, the obtained REE values indicate that image data have

TABLE I
STATISTICAL EVALUATION (MEAN) OF 8 USC-SIPI IMAGES

Algorithm	RCE	REE	RMBE	RSS	PSNR (dB)	Payload (bpp)
Prop.10p	0.5253	0.5155	0.9926	0.9694	30.34	0.270
Prop.15p	0.5367	0.5210	0.9891	0.9561	27.20	0.370
Prop.20p	0.5477	0.5255	0.9887	0.9438	25.05	0.458
imadjust	0.5631	0.4958	0.9643	0.9223	22.37	null
histeq	0.5941	0.4182	0.9539	0.8728	18.62	null
adapthisteq	0.5475	0.5383	0.9689	0.8820	18.62	null

TABLE II
STATISTICAL EVALUATION (MEAN) OF 524 KODAK IMAGES

Algorithm	RCE	REE	RMBE	RSS	PSNR (dB)	Payload (bpp)
Prop.10p	0.5159	0.5144	0.9947	0.9781	30.38	0.313
Prop.15p	0.5331	0.5239	0.9889	0.9556	27.19	0.420
Prop.20p	0.5435	0.5281	0.9852	0.9418	24.89	0.511
imadjust	0.5451	0.4956	0.9529	0.9310	24.09	null
histeq	0.6074	0.4217	0.9027	0.8248	15.49	null
adapthisteq	0.5408	0.5356	0.9377	0.8664	17.67	null

been increased by our proposed algorithm and *adapthisteq*, but decreased by *imadjust* and *histeq*. As for the RMBE and RSS values, *Prop.20p* outperformed all of the three MATLAB functions, indicating that less changes were made to image brightness and structural similarity. Besides the higher PSNR values, the original image can be directly recovered from the contrast-enhanced image generated with the proposed algorithm. So it is more advantageous than the three MATLAB functions for image contrast enhancement.

IV. CONCLUSION

In this letter, a new reversible data hiding algorithm has been proposed with the property of contrast enhancement. Basically, the two peaks (i.e. the highest two bins) in the histogram are selected for data embedding so that histogram equalization can be simultaneously performed by repeating the process. The experimental results have shown that the image contrast can be enhanced by splitting a number of histogram peaks pair by pair. Compared with the special MATLAB functions, the visual quality of the contrast-enhanced images generated by our algorithm is better preserved. Moreover, the original image can be exactly recovered without any additional information. Hence the proposed algorithm has made the image contrast enhancement reversible. Improving the algorithm robustness, and applying it to the medical and satellite images for the better visibility, will be our future work.

ACKNOWLEDGMENT

The authors would like to sincerely thank the editors and anonymous reviewers for their valuable comments.

REFERENCES

[1] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[2] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[3] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.

[4] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Process. Lett.*, vol. 14, no. 4, pp. 255–258, Apr. 2007.

[5] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, Jul. 2009.

[6] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3524–3533, Jan. 2011.

[7] Z. Zhao, H. Luo, Z.-M. Lu, and J.-S. Pan, "Reversible data hiding based on multilevel histogram modification and sequential recovery," *Int. J. Electron. Commun. (AEÜ)*, vol. 65, pp. 814–826, 2011.

[8] H. T. Wu and J. Huang, "Reversible image watermarking on prediction error by efficient histogram modification," *Signal Process.*, vol. 92, no. 12, pp. 3000–3009, Dec. 2012.

[9] Y. Yang, X. Sun, H. Yang, C.-T. Li, and R. Xiao, "A contrast-sensitive reversible visible image watermarking technique," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 656–667, May 2009.

[10] J. A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 889–896, May 2000.

[11] P. G. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W. J. Rucklidge, "The emerging JBIG2 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 7, pp. 838–848, Jul. 1998.

[12] The USC-SIPI Image Database [Online]. Available: http://sipi.usc.edu/database/

[13] Kodak Lossless True Color Image Suite [Online]. Available: http://www.r0k.us/graphics/kodak/

[14] M.-Z. Gao, Z.-G. Wu, and L. Wang, "Comprehensive evaluation for HE based contrast enhancement techniques," *Adv. Intell. Syst. Applicat.*, vol. 2, pp. 331–338, 2013.