Telecommun Syst (2010) 44: 241–251
DOI 10.1007/s11235-009-9262-x

Techniques and applications of intelligent multimedia data hiding

Hsiang-Cheh Huang · Wai-Chi Fang

Published online: 13 January 2010

Abstract In this paper, we present the intelligent multimedia data hiding techniques and their possible applications. An introduction on intelligent multimedia data hiding is described which covers backgrounds, recent advances, methodologies, and implementations. The recently developed research branch called reversible data hiding is also depicted. Two major classes for the implementation of reversible data hiding, namely, the difference expansion method, and the histogram-based scheme, are discussed. With its ease of implementation, the histogram-based reversible data hiding technique is then illustrated with simulation results and actual implementations. Possible extension for our work is also depicted in the concluding remarks.

Keywords Steganography · Watermarking · Reversible data hiding · Histogram · Difference expansion · Uniform Resource Locator (URL) · Peak signal to noise ratio (PSNR)

1 Introduction

With the widespread use of the Internet and the booming growth of the computer industry, people nowadays can easily retrieve multimedia contents with their own computers or mobile phones over the Internet or mobile channels, under the ubiquitous computing environments. Multimedia related researches and applications have greatly increased in the last twenty years. In addition to multimedia signal processing, data hiding techniques aiming at protecting copyright-related issues are of considerable interest in academia and industry.

In this paper, we will first describe the advances and methodologies in data hiding, which is suitable for applying over ubiquitous computing environments. The multimedia data is supposed to be transmitted over the Internet or the wireless networks, and the ease of delivery over the ubiquitous computing environments tends to get the multimedia contents infringed upon at any time. Data hiding is one of the useful schemes for delivering secret messages. Unlike cryptographic schemes to encrypt data into noise-like streams, the output after data hiding is perceived almost identically to its input counterpart. How to efficiently and effectively hide the message into multimedia data is one of the highlights in this part.

In addition to the brief description of the backgrounds of data hiding schemes, some possible applications are also pointed out in the second part of this paper. By employing the reversible data hiding scheme, at the encoding side, the Uniform Resource Locator (URL) of the webpage, or other sort of watermark pattern, can be hidden into the cover media. On the other hand, at the decoding side, both the URL can be retrieved and the webpage can be accessed directly, and the original image can be recovered intact. In the latter part, possible extensions the applications pointed out in this paper are also discussed.

H.-C. Huang (✉)
Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, R.O.C.
e-mail: huang.hc@gmail.com

W.-C. Fang
Department of Electronics Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, R.O.C.
e-mail: Dr.Wfang@gmail.com

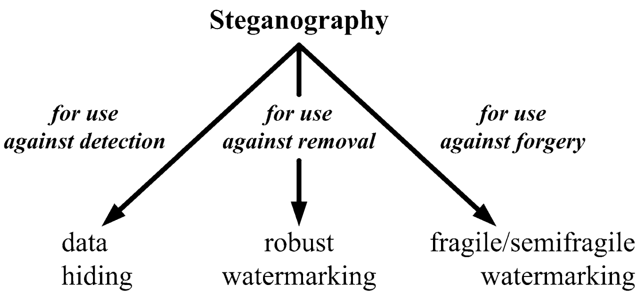


Fig. 1 Depiction of relationships in steganography, watermarking, and data hiding

2 Backgrounds of multimedia data hiding

2.1 Multimedia representations

Multimedia involve more than one medium and including image, sound, text, video, for example, [1–4]. Multiple media are simultaneously transmitted over the network, using the Internet or mobile channels. They can be divided into two categories. These are:

- **Continuous media** which is a series of consecutive units of equal duration. For instance, audio. They may include video and animation.
- **Discrete media** contains one presentation unit, which may be text, graphics on image.

Multimedia signal processing has the following goals:

- high quality,
- highly efficient compression,
- security-related issues,
- error resilient transmission, and
- interactivity.

2.2 Steganography for multimedia data

Steganography is the way of embedding hidden messages in such a way that no one apart from the sender and intended recipient even realizes there is a hidden message. Meanwhile, the very existence of the hidden message should hardly be detected by any third party [5]. Generally speaking, steganography can be classified into three relevant categories: data hiding, and robust and fragile watermarking. Their relationships can be depicted in Fig. 1.

2.3 Requirements of digital watermarking

There are many metrics used to evaluate the effectiveness of a watermarking algorithm [6, 7]. The requirements of watermarking algorithm design depend on the specific needs of the application. From algorithm design viewpoint, the most wanted and critical requirements are *watermark imperceptibility*, *watermark robustness*, *watermark capacity*, the amount of needed *side information*, and the *places* for embedding the watermark. There inter-relationships can be depicted as follows.

- Watermarking imperceptibility refers to the quality of the watermarked image. From subjective point of view, the watermarked image looks nearly untouched to compare to its original, or un-watermarked, counterpart. Objectively speaking, the error induced between the watermarked and original images should be as small as possible.
- Watermarking robustness means the sustainability to resist intentional or un-intentional image processing, called attacks. The watermark is supposed to be extracted after experiencing such attacks, and extracted watermark should be similar, or even identical, to the embedded one. From objective viewpoint, people often calculate the correlation between the extracted watermark and the embedded one. The higher the correlation, the better the outcome.
- Watermark capacity denotes the number of bits embedded into the original image, based on the well-designed watermarking algorithm. To speak heuristically, the capacity needs to be more than some threshold, so that after the extraction of watermark, it can be easily recognized.
- For most watermarking applications, it is commonly seen that a small amount of side information may be produced, and be transmitted reliably to the receiver. How to keep the size of the side information as small as possible is important for algorithm design and practical applications.
- The place for embedding the watermark is the major issue for the design of the algorithm. In order to retain the imperceptibility after watermark embedding, it is desired to embed the watermark into the less significant portion of the cover media, for instance, the least significant bits of the samples, the lower frequency bands of the coefficients, or the more active areas due to the attributes of human visual system (HVS) or human auditory system (HAS). On the contrary, for embedding into the areas above, the watermark tends to be vulnerable to attacks such as signal processing, resampling, resizing, etc. How to effectively search for the appropriate positions and tradeoff for watermark embedding is the main concern for algorithm developers.

On the one hand, although these requirements are all very desirable, as pointed out in literature [8–11], they influence, or even conflict, with each other. After fixing one dimension, the remaining may then have conflicts between one another. Some tradeoff must then be developed. Based on the requirements pointed out, and also the purposes for applications, it is important to meet part of the requirements in designing the algorithm. Intelligent optimization techniques can be employed to tune parameters described above, including the embedding positions, the expected imperceptibility after watermark embedding, the simulated robustness after watermark extraction, the amount of capacity embedded, the side information produced, etc. And this research topic has been a popular direction in the last few years [12, 13].

On the other hand, researches for data hiding are popular in both academia and industry. Recently, research directions in data hiding not only include the development of new algorithms, but also focus on the integration between relating fields, including applying steganography into Unicode or Microsoft Word document for document tracking [14, 15], enhancing the speech quality for telephone communication by data hiding [16, 17], or controlling the accessibility for viewing the scalable multimedia by the combination of encryption and watermarking [18, 19]. With the observations above, researches and applications in data hiding are blooming topics that can further be explored.

3 Categories for reversible data hiding

As we point out, reversible data hiding is one of the popular trends in watermarking related researches. There are two major implementation schemes for reversible data hiding, one is the difference expansion scheme (DE), and the other is the histogram-based scheme. We briefly describe both schemes here, and also point out their advantages and disadvantages. Considering the ease of practical implementations, we choose to use the histogram-based scheme in this paper.

3.1 Concepts for the difference expansion scheme

The difference expansion (DE) scheme follows the concepts directly from wavelet transforms. In DE, every two neighboring pixels are grouped together as a pair (x, y). Next, simple calculations can be performed by

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor, \tag{1}$$

and

$$h = x - y, \tag{2}$$

where l and h can be regarded as the lower and higher frequency bands, respectively, and $\lfloor\cdot\rfloor$ denotes the floor function. We can also find that h is the difference between the pixels. For the embedding of one bit b, $b \in \{0, 1\}$, in the pair (x, y), the concept is to keep the lower frequency band the same, and only the higher frequency band is modified by

$$h' = 2 \times h + b. \tag{3}$$

Thus, h' is called the DE scheme.

For obtaining the image containing the hidden information, the new pair (x', y') can be calculated by

$$x' = l + \left\lfloor \frac{h'+1}{2} \right\rfloor, \tag{4}$$

and

$$y' = l - \left\lfloor \frac{h'}{2} \right\rfloor. \tag{5}$$

For achieving reversible data hiding with DE, the original needs to be perfectly restored. Hence, by induction, we observe that the lower frequency band l'' and higher frequency band h'', derived from (x', y') can be represented by

$$l'' = \left\lfloor \frac{x'+y'}{2} \right\rfloor = \left\lfloor \frac{2l}{2} + \frac{1}{2}\left(\left\lfloor \frac{h'+1}{2} \right\rfloor + \left\lfloor \frac{h'}{2} \right\rfloor \right) \right\rfloor$$
$$= \lfloor l \rfloor = l, \tag{6}$$

$$h'' = x' - y' = \left\lfloor \frac{h'+1}{2} \right\rfloor + \left\lfloor \frac{h'}{2} \right\rfloor = \left\lfloor \frac{2h'+1}{2} \right\rfloor = h'. \tag{7}$$

With h'', the hidden data can be determined by getting the least significant bit (LSB). Also, based on the DE scheme, we can see that $h = \lfloor \frac{h'}{2} \rfloor = \lfloor \frac{h''}{2} \rfloor$, and then the pair in the original image, (x, y), can be restored by

$$x = l'' + \left\lfloor \frac{\lfloor \frac{h''}{2} \rfloor + 1}{2} \right\rfloor, \tag{8}$$

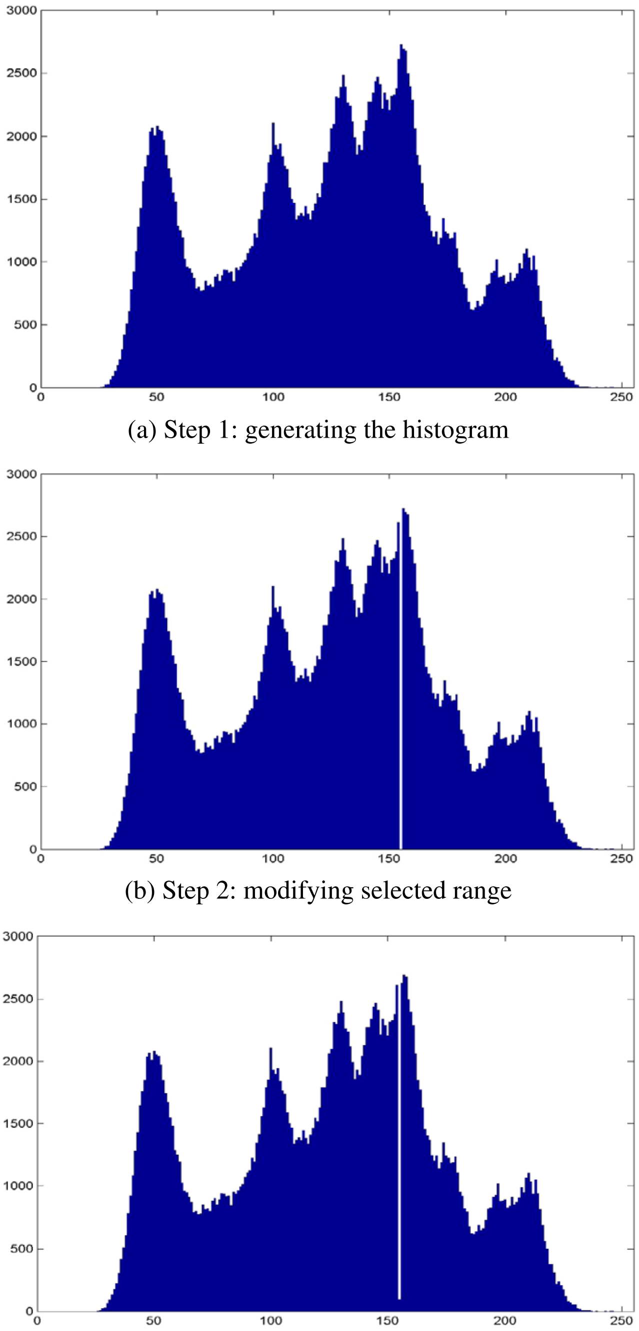
$$y = l'' - \left\lfloor \frac{\lfloor \frac{h''}{2} \rfloor}{2} \right\rfloor. \tag{9}$$

From the derivations above, we can see that for one image with the size of $M \times N$, at most $\frac{1}{2} \times M \times N$ bits of data can be embedded. However, to prevent the overflow and underflow problems such that x and y should lie in the range of $[0, 255]$, a portion of pairs may not be suitable for embedding data, and the locations of these pairs need to be recorded in advance.

3.2 Concepts for the histogram-based scheme

We first describe the embedding scheme for histogram-based reversible data hiding. Let the original image and the image containing hidden data be $\mathbf{X}$ and $\mathbf{X}'$, respectively, both have the image size of $M \times N$. Steps for reversible data hiding by using the histogram of the original image are described as follows. Figure 2 demonstrates the corresponding steps, where the horizontal axis denotes the luminance value, and the vertical axis means the number of occurrence, respectively.

Step 1. *Generation of histogram of the image*: The histogram of the original image is produced. The luminance



(a) Step 1: generating the histogram

(b) Step 2: modifying selected range

(c) Step 3: embedding hidden data

Fig. 2 Depiction of histogram-based reversible data hiding steps with the Lena test image; horizontal and vertical axes mean the luminance value and the number of occurrences, respectively

value with the maximal occurrences is treated as the "max" point, and the all the luminance values that have no occurrence are regarded as "zero" points. In particular, among the group of luminance values of zero points that are larger than that of the max point, the smallest luminance value in this group is treated as the "min" point. The luminance values of max (represented by a) and min points (represented by b) are treated as secret keys. Apparently, $b > a$. For those smaller than the luminance of max point, similar procedure can be performed.

Step 2. *Modifying selected range with the secret keys.* All the luminance values between a and b are added by 1.

Step 3. *Embedding the hidden data.* The hidden data, represented by bits, is ready to be embedded. In the generated data in Step 2, we choose the pixel positions equal to the luminance value of $(a + 1)$. If bit 0 is embedded, then the luminance value of that position is modified back to a. If bit 1 is embedded, keep the value to $(a + 1)$. The capacity of the hidden data is the number of occurrences at max point.

In contrast with the data embedding scheme, the hidden data must be extracted without any loss to make the algorithm applicable, and it is the main reason for the reversibility. Reversible data extraction can be done with the following steps.

Step 1. *Generation of histogram of the image containing hidden data*: Again, the histogram of the image containing hidden data is calculated and produced. The luminance values of "max" and "min" points, regarded as secret keys, are also received.

Step 2. *Obtaining selected range with secret keys.* The luminance values between the max and min points are compared.

Step 3. *Extracting the watermark relating to the original.* Every pixel in the watermarked image is scanned and examined sequentially to extract the watermark bits.

- For the case that the luminance value of the max point is smaller than that of the min point, when the luminance of one pixel is exactly larger than that of the "max" points by 1, output bit '1' as extracted watermark bit, and decrease its luminance value by 1 simultaneously. On the contrary, when the luminance of one pixel is exactly smaller than that of the "max" points by 1, output bit '0' as extracted watermark bit, and increase its luminance value by 1.
- In contrast, if the luminance value of the max point is smaller than that of the min point, the reverse process of the previous item is performed.

Step 4. *Re-generating the original data from watermark.* The extracted watermark bits are gathered, and the hidden data can be deciphered.

Let the original image and the image containing hidden data be $\mathbf{X}$ and $\mathbf{X}'$, respectively, both have the image size of c. For measuring the image quality after embedding the hidden data, people usually use the peak signal-to-noise ratio (PSNR) for objective evaluation, the higher the PSNR value, the better the result. And the PSNR is calculated from the mean square error (MSE) between the original image $\mathbf{X}$, and the image containing hidden data $\mathbf{X}'$. The MSE can be calculated by

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (X'(i,j) - X(i,j))^2.$$

Consequently, the PSNR, with its unit in dB, can be calculated by

$$\mathrm{PSNR} = 10 \log \left(\frac{225^2}{\mathrm{MSE}} \right).$$

Similarly, the larger PSNR value means the better quality objectively. Considering the worst case such that after hiding the data, the luminance value for every pixel is changed, which means the increase of luminance value by 1 for every pixel. With this scenario, the MSE becomes

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} 1^2 = \frac{M \times N}{M \times N} = 1.$$

And the corresponding PSNR becomes

$$\mathrm{PSNR} = 10 \log \left(\frac{225^2}{1} \right) = 48.13 \text{ (dB)}.$$

This means that with histogram-based scheme, the imperceptibility in image quality can be guaranteed.

3.3 Comparisons between the DE and the histogram-based schemes

Both the DE and histogram-based schemes can achieve reversible data hiding. For the DE scheme, it can hide as many as $\frac{1}{2} \times M \times N$ bits, or $\frac{1}{16} \times M \times N$ bytes, while the capacity for the histogram-based scheme is restricted by the maximal amount of occurrences in the histogram of the original image, which is much smaller than its counterpart for the DE scheme. However, since a portion of pairs in the DE schemes are not suitable for data embedding due to overflow and underflow, these pairs needs to be recorded, which is considered as the side information. The side information for the histogram-based scheme is two bytes only, that is, the luminance value of max point, represented by a, and that of the min point, represented by b. In addition, the image quality for the histogram-based scheme is guaranteed to be more than 48.13 dB. Summing up, considering the applications for reversible data hiding, for embedding a reasonable amount of capacity, we choose to use the histogram-based scheme in this paper, and present three implementations in the next section entitled "Applications of data hiding and its implementation."

3.4 Latest progress in reversible data hiding

Since both the DE and the histogram-based schemes have been published in 2003 and 2006, respectively, the latest progress in reversible data hiding can be pointed out in this subsection, with some representative papers in [20–23]. In [20], the authors extend conventional DE algorithm by adjusting difference value between pixel pairs. The overhead for embedding is reduced; hence the capacity can be increased. In [21], the authors present an improvement of the DE scheme by studying the appropriate regions for data hiding. Laplacian distribution is employed to help increase the embedding capacity. In [22], the authors apply reversible data hiding for image authentication. It has the capability to mitigate the effect of image compression and small incidental alteration. In [23], the authors proposed to consider the multi-level hiding method based on the histogram-based scheme. By exploring the relationships between the original image and the marked one, the histogram-based scheme is applied, which results in a much higher embedding capacity to compare with the conventional histogram-based scheme in [24].

With the brief descriptions about the latest progress in reversible data hiding, possible extensions and improvements can be integrated with our applications to be described in the next section.

4 Applications of data hiding and its implementation

Here we employ the histogram-based reversible data hiding scheme [24] for our implementation, and we point out three types of application in this section with reversible data hiding, including:

- Hiding the Uniform Resource Locator (URL) into original images,
- Recovering the original from marked image, and
- Removing visible watermark from marked image.

The flowchart for implementing first application, Application #1, can be briefly illustrated in Fig. 3(a), and that for the latter two applications, Applications #2 and #3, can be depicted in Fig. 3(b), respectively. In addition, the block of "Change of histogram for information extraction" in Fig. 3(b) performs the same steps as those depicted in Fig. 2(a) to Fig. 2(c) by modifying the histogram for reversible data hiding.

4.1 Application #1: hiding URL into cover image

We propose an application by embedding the Uniform Resource Locators (URL) [25] of two webpages into the cover

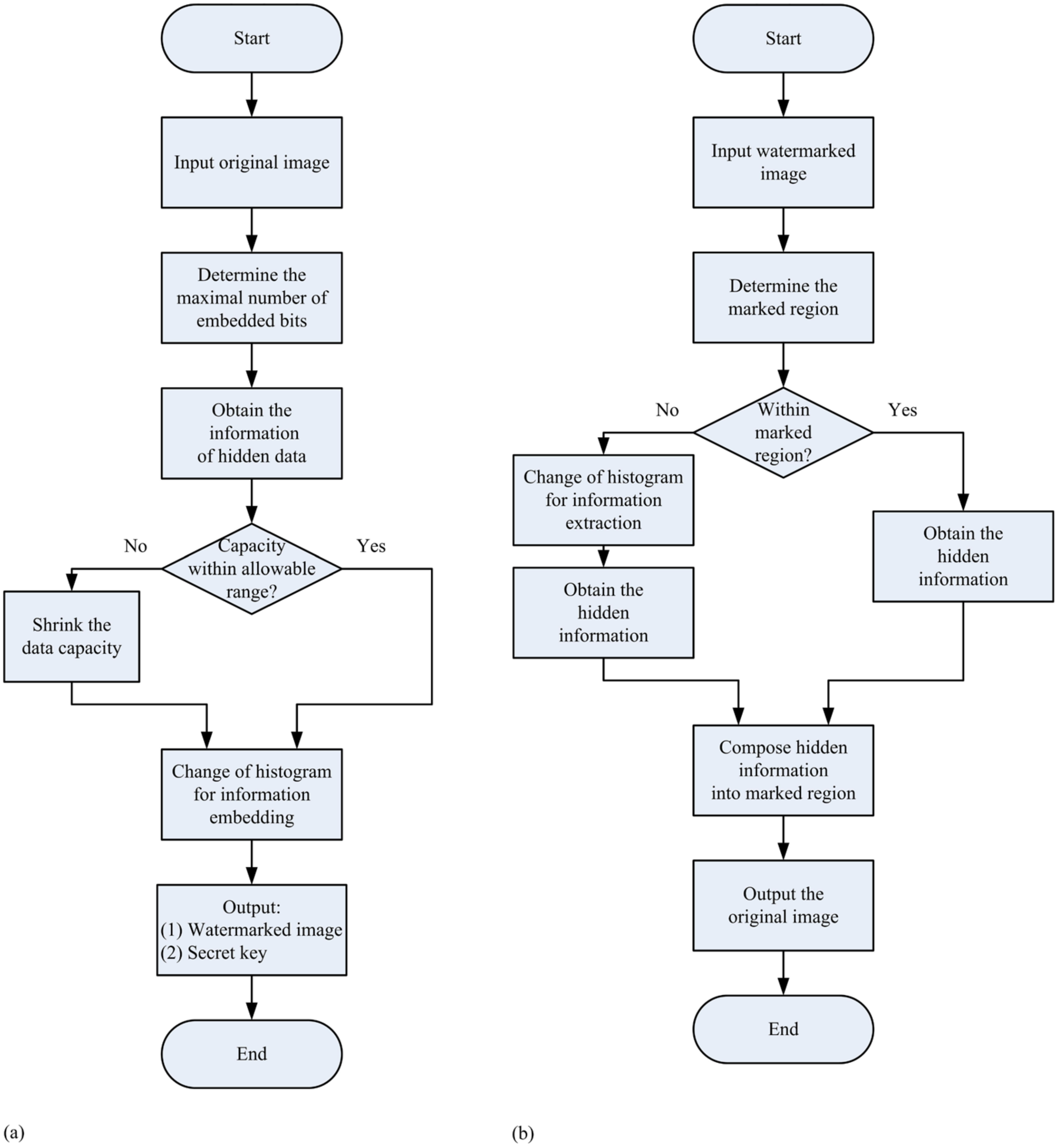
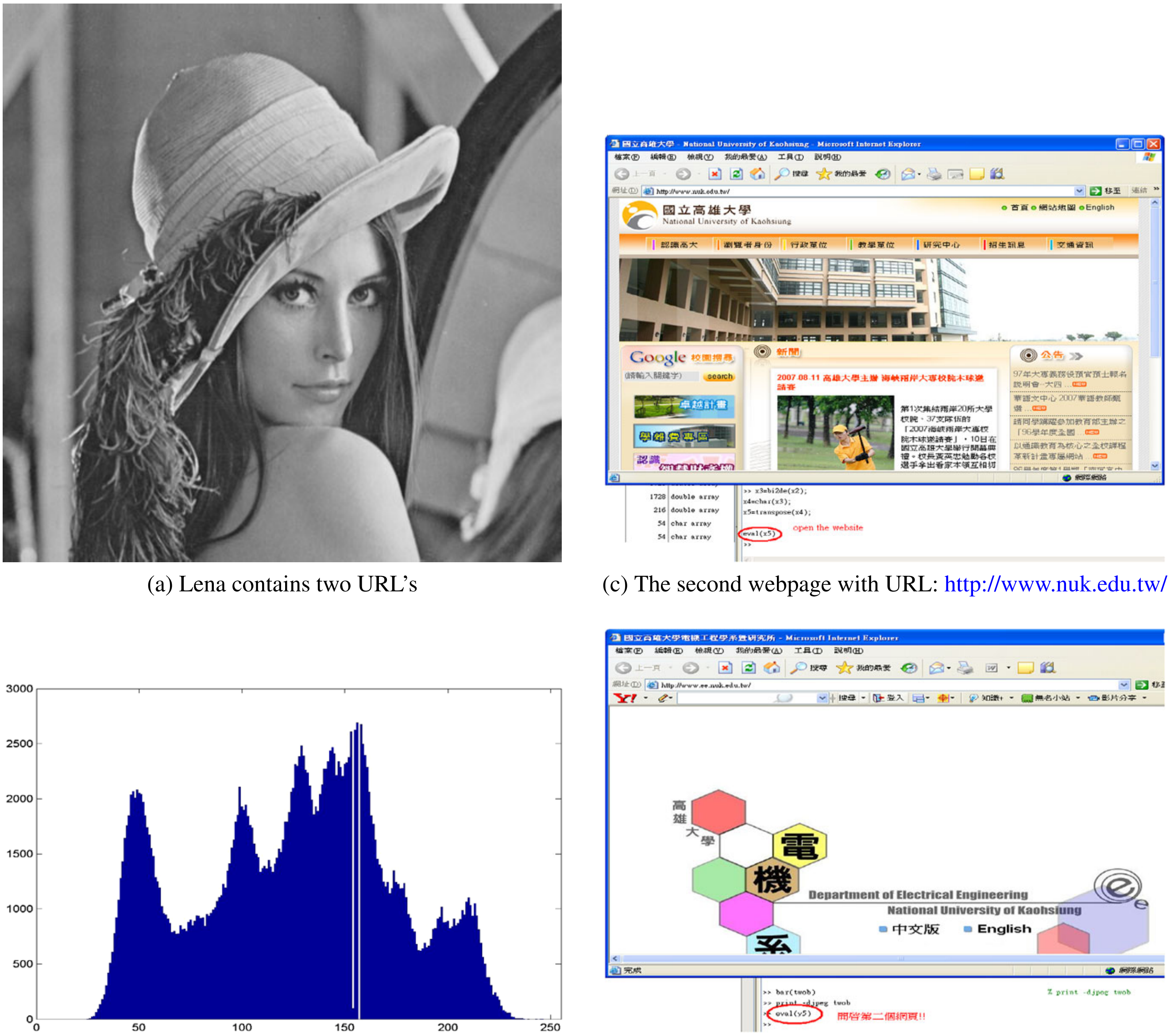

Start
Input original image
Determine the maximal number of embedded bits
Obtain the information of hidden data
Capacity within allowable range?
No
Yes
Shrink the data capacity
Change of histogram for information embedding
Output:
(1) Watermarked image
(2) Secret key
End
(a)
Start
Input watermarked image
Determine the marked region
Within marked region?
No
Yes
Change of histogram for information extraction
Obtain the hidden information
Obtain the hidden information
Compose hidden information into marked region
Output the original image
End
(b)

Fig. 3 Flowcharts for implementing our applications in this paper. (**a**) Flowchart for reversible data hiding with URL information. (**b**) Flowchart for reversible data extraction and original image recovery

image. On the one hand, the image containing the URL's and its original counterpart looks almost identical, and the PSNR value is high, which is illustrated in Fig. 4(a). The hidden data, or the two URL's, are http://www.nuk.edu.tw/ and http://www.ee.nuk.edu.tw/, respectively. The histogram with the embedding of URL is depicted in Fig. 4(b). On the other hand, once the hidden URL's are extracted, the browser can automatically link to the designated webpages in Fig. 4(c) and Fig. 4(d), respectively.

For measuring the image quality after data embedding, we use the peak signal-to-noise ratio (PSNR) to measure the image quality objectively. The PSNR is high enough with


3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
國立高雄大學 National University of Kaohsiung
open the website
Department of Electrical Engineering
National University of Kaohsiung
中文版
English
開啓第二個網頁!!

```
1728 double array
 216 double array
  54 char array
  54 char array
```

```
>> x3=bi2de(x2);
x4=char(x3);
x5=transpose(x4);

eval(x5)
>>
```

```
>> bar(twob)                                    % print -djpeg twob
>> print -djpeg twob
>> eval(y5)
>>
```

(a) Lena contains two URL’s

(c) The second webpage with URL: http://www.nuk.edu.tw/

(b) Histogram of the image in (a)

(d) The second webpage with URL: http://www.ee.nuk.edu.tw/

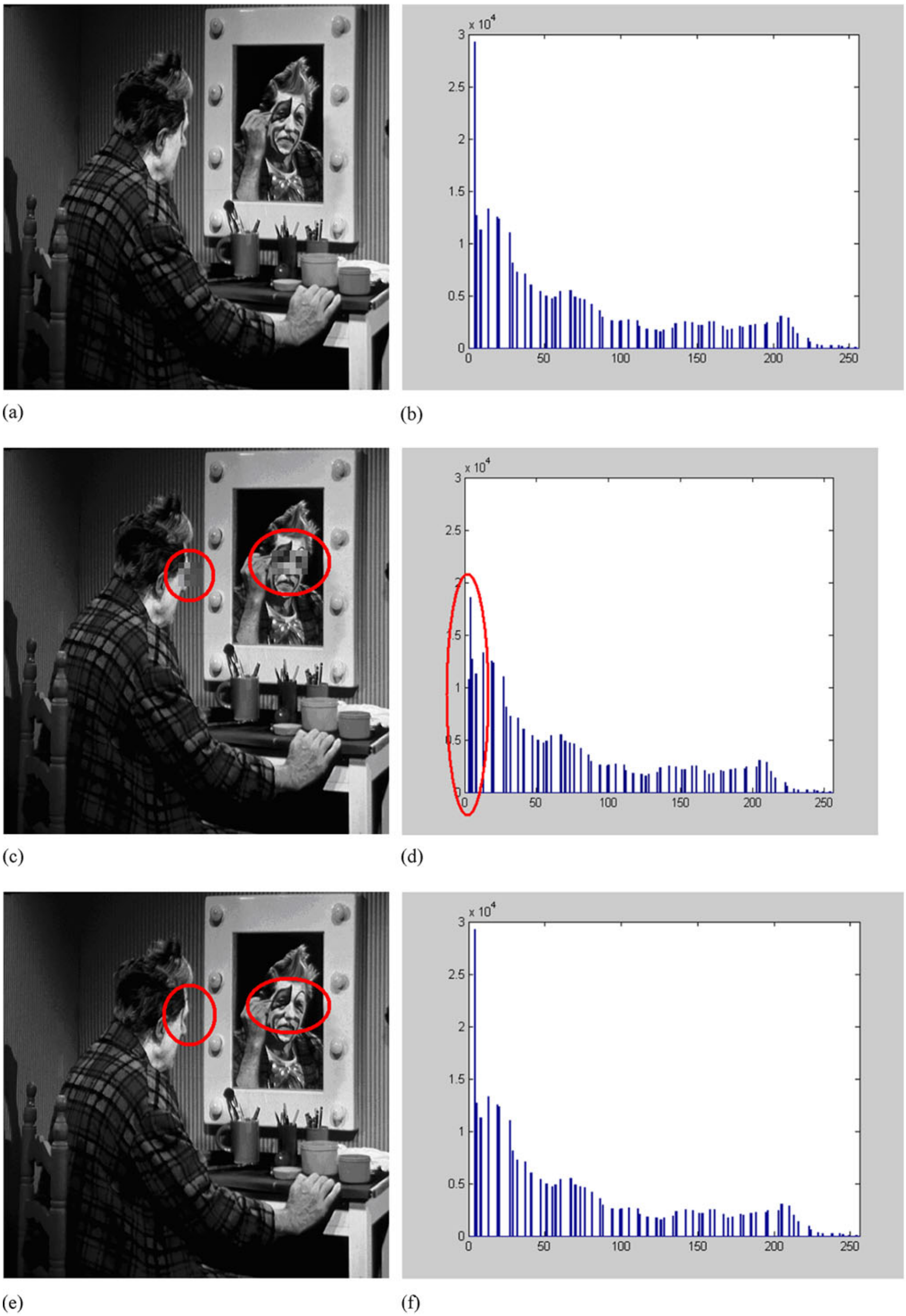
Fig. 4 (**a**) Lena image contains two URL’s. PSNR = 48.22 dB. (**b**) Histogram of the image in (**a**); horizontal and vertical axes mean the luminance value and the number of occurrences, respectively. (**c**) Browser links to the first webpage from the extracted data. (**d**) Browser links to the second webpag

the numerical value of 48.22 dB. Also, the value is larger than the worst case of 48.13 dB described above. From subjective viewpoint, the image with hidden data can be hardly differentiated from its original counterpart.

4.2 Application #2: recovering the original from marked image

When we read the newspapers or view images with the browsers, sometimes we can see that secret information such as some VIP’s car plate numbers is intentionally marked before the delivery of such news clips. The marked information might need to be recognized afterwards, and without any prior information, the marked region cannot be recovered.

Therefore, we employ the reversible data hiding techniques to hide the information to be marked in advance. When the secret information needs to be extracted, we can gather it back from the rest of the image with the aid of secret keys.



(a) (b) (c) (d) (e) (f)

Fig. 5 (**a**) Clown original image. (**b**) Histogram of the image in (**a**). (**c**) The two selected areas to mark the man's face and reflection. (**d**) Corresponding histogram after data embedding in (**c**). (**e**) Recovered image, identical to part (**a**). (**f**) Histogram of the image in (**e**), which is identical to part (**b**)

The areas that contain the confidential information in the original image, for instance, the man's head portion and its reflection in the mirror in Fig. 5(a), should be located first manually, or by using pattern recognition schemes. Its corresponding histogram is illustrated in Fig. 5(b). The areas are assumed to be rectangular regions.

The width w_{m1} and w_{m2}, the height h_{m1} and h_{m2}, and the coordinate of the upper left corner of these regions, (x_1, y_1) and (x_2, y_2), are served as the secret information to be included into the secret key. The subscript m in the widths and heights denotes the region to be marked. If the size of the original image is 512×512, the eight parameters above should be within the region between 0 and 511, meaning that each parameter can be represented by using 9 bits at most. Thus, the secret key should be at most 72 bits in length. More generally, if the original image has the size of $W \times H$, it requires that $0 \le x_1,\ x_2 \le W - 1$, $0 \le y_1,\ y_2 \le H - 1$, $0 \le w_{m1} \le W - 1 - x_1$, $0 \le w_{m2} \le W - 1 - x_2$, $0 \le h_{m1} \le$

$H - 1 - y_1$ and $0 \leq h_{m2} \leq H - 1 - y_2$. Then, the secret key would be $(\lceil \log_2(x_1) \rceil + \lceil \log_2(y_1) \rceil + \lceil \log_2(w_{m1}) \rceil + \lceil \log_2(h_{m1}) \rceil) + (\lceil \log_2(x_2) \rceil + \lceil \log_2(y_2) \rceil + \lceil \log_2(w_{m2}) \rceil + \lceil \log_2(h_{m2}) \rceil)$—bit in length, where $\lceil \bullet \rceil$ denotes the ceiling function.

Next, the binary representation of the two selected regions, pixel by pixel, is served as the data to be reversibly hidden into the original image. We employ the grey level image to perform simulations. And it is assumed that each pixel is represented by one byte, or eight bits. Every pixel is represented by an 8-bit string. Concatenating the binary form of every pixel, the hidden data can be produced. And the two regions are intentionally marked to protect the identity of the man's face and reflection.

4.3 Application #3: removing visible watermark from marked image

Visible watermarking is one of the earliest applications for digital watermarking and data hiding. For visible watermarking, the owner of the media, an image for example, often puts a mark into one of the corners in the original. The mark is easily seen by the viewers, and it's the main purpose of ownership annotation for visible watermarking. This is commonly seen in TV programs with the logo of the TV station, or the 'favorite icon' [26] in the upper-left or upper-right corner of the screen today. Due to its simplicity, only a few portion of watermarking researches focus on visible watermarking [27, 28].

However, even though visible watermarks can claim and annotate the copyright, they generally degrade the values of the original images, and how to effectively remove the visible watermark becomes an interesting topic. Based on the experiences in the two applications just described, we propose one algorithm based on reversible data hiding for effectively removing visible watermark here.

Figure 6 is an illustration for removing the visible watermark with the histogram-based reversible data hiding. Figure 6(a) is the 512 × 512 Lena image containing a small watermark with size 16 × 16, and Fig. 6(b) is the visible watermark enlarged four times for the ease of recognition. Considering the first application for hiding the URL, the 16 × 16 favicon can also be used to serve as the visible watermark. With reversible data hiding, when receiving the luminance values between a and b in the histogram generation procedure, the watermark can be totally removed. Finally, the image can be perfectly recovered, shown in Fig. 6(c), which is identical to its original counterpart, and its histogram is the same as that in Fig. 2(a).

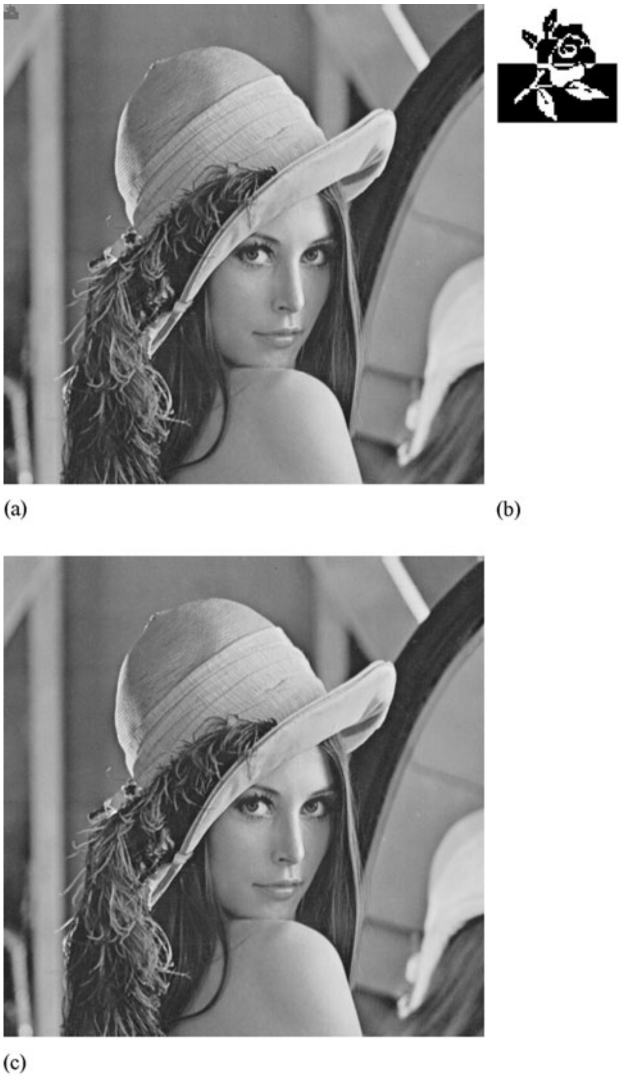


Fig. 6 (**a**) Lena image with a visible watermark on the upper left corner. (**b**) The visible watermark, four times enlarged. (**c**) The recovered Lena image, which is identical to its original

5 Conclusions

In this paper, we described about the fundamentals of watermarking and data hiding, and pointed out possible applications and their implementations with reversible data hiding.

Before delivery of original image, the target URL's can be hidden first, or the area containing sensitive information can be spread into other parts of the image by using the histogram-based reversible data hiding techniques with the aid of short secret keys. And after transmission, the both original and secret information can be perfectly recovered and retrieved, and the target webpage can be linked automatically. Simulation results depict the applicability and ease of implementation of the proposed algorithm, and these results depict the utility of the proposed algorithm.

As we can see from the concepts of the difference expansion (DE) scheme and the histogram-based one, both of them have their inherent advantages and disadvantages, and improvements can be expected by combining the advantages altogether. Regarding to the embedding capacity, the DE scheme can embed a much more amount of data than the histogram-based one. Regarding to the side information produced, with the histogram-based scheme, two bytes denoting the max and zero points are served as the side information, while with the DE scheme, a much more bits of side information that can be as many as $\frac{1}{16}$ of the size of image, denoting the locations suitable for data embedding, can be produced. Thus, by integrating the two schemes, we can expect to propose an algorithm with a large capacity for data embedding, while a very few amount of side information, in the future.

Unlike conventional schemes to serve the random sequences as the data for embedding, we hide meaningful data, namely, the URL and some part of the contents covered by visible watermarks, into the original image. Considering the ease of practical implementations, the histogram-based scheme is employed, and three applications have been presented in this paper to demonstrate the effectiveness of our claims.

Summing up, reversible data hiding has its potential for developing new algorithms and realizing new implementations. We presented several directions for practical implementations. These schemes can be directly extendable to digital forensics.

Acknowledgement This work was supported by National Science Council (NSC), Taiwan, R.O.C. under grant number NSC 98-2221-E-390-017.

References

1. Steinmetz, R., & Nahrstedt, K. (2004). *Multimedia systems*. Berlin/Heidelberg: Springer.
2. Poynton, C. A. (2003). *Digital video and HDTV: algorithms and interfaces*. San Francisco: Morgan Kaufmann.
3. Symes, P. (2003). *Digital video compression*. Columbus: Glencoe/McGraw-Hill.
4. Mandal, M. K. (2002). *Multimedia signals and systems*. Dordrecht: Kluwer Academic.
5. Fridrich, J., & Lisonek, P. (2007). Grid colorings in steganography. *IEEE Transactions on Information Theory*, *53*(4), 1547–1549.
6. Altun, O., Sharma, G., Celik, M. U., & Bocko, M. (2006). A set theoretic framework for watermarking and its application to semifragile tamper detection. *IEEE Transactions on Information Forensics and Security*, *1*(4), 479–492.
7. Altun, O., Sharma, G., & Bocko, M. (2006). Optimum watermark design by vector space projections. In *Proc. IEEE intl. conf. image proc.* (pp. 1413–1416).
8. Pan, J. S., Huang, H.-C., Jain, L. C., & Fang, W. C. (Eds.) (2007). *Intelligent multimedia data hiding: new directions*. Berlin-Heidelberg: Springer.
9. Barni, M., Bartolini, F., De Rosa, A., & Piva, A. (2000). Capacity of full frame DCT image watermarks. *IEEE Transactions on Image Processing*, *9*, 1450–1455.
10. Kirovski, D., & Malvar, H. S. (2003). Spread spectrum watermarking of audio signals. *IEEE Transactions on Signal Processing*, *51*, 1020–1033.
11. Lin, C. Y., & Chang, S. F. (2001). Watermarking capacity of digital images based on domain-specific masking effects. In *Int'l conf. information technology: coding and computing* (pp. 90–94).
12. Chu, S. C., Huang, H. C., Shi, Y., Wu, S. Y., & Shieh, C. S. (2008). Genetic watermarking for zerotree-based applications. *Circuits, Systems, and Signal Processing*, *27*(2), 171–182.
13. Wu, W. T., & Shih, F. Y. (2006). Genetic algorithm based methodology for breaking the steganalytic systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, *36*(1), 24–31.
14. Mabry, F. J., James, J. R., & Ferguson, A. J. (2007). Unicode steganographic exploits: maintaining enterprise border security. *IEEE Security & Privacy Magazine*, *5*(5), 32–39.
15. Liu, T. Y., & Tsai, W. H. (2007). A new steganographic method for data hiding in Microsoft Word documents by a change tracking technique. *IEEE Transactions on Information Forensics and Security*, *2*(1), 24–30.
16. Chen, S., Leung, H., & Ding, H. (2007). Telephony speech enhancement by data hiding. *IEEE Transactions on Instrumentation and Measurement*, *56*(1), 63–74.
17. Chen, O. T. C., & Wu, W. C. (2008). Highly robust, secure, and perceptual-quality echo hiding scheme. *IEEE Transactions on Audio, Speech, and Language Processing*, *16*(3), 629–638.
18. Chang, F. C., Huang, H. C., & Hang, H. M. (2007). Layered access control schemes on watermarked scalable media. *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, *49*(3), 443–455.
19. Huang, H. C., & Chen, Y. H. (2009). Genetic fingerprinting for copyright protection of multicast media. *Soft Computing*, *13*(4), 383–391.
20. Weng, S., Zhao, Y., Pan, J. S., & Ni, R. (2008). Reversible watermarking based on invariability and adjustment on pixel pairs. *IEEE Signal Processing Letters*, *15*, 721–724.
21. Kim, H. J., Sachnev, V., Shi, Y. Q., Nam, J., & Choo, H. G. (2008). A novel difference expansion transform for reversible data embedding. *IEEE Transactions on Information Forensics and Security*, *3*(3), 456–465.
22. Ni, Z., Shi, Y. Q., Ansari, N., Su, W., Sun, Q., & Lin, X. (2008). Robust lossless image data hiding designed for semi-fragile image authentication. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(4), 497–509.
23. Lin, C. C., Tai, W. L., & Chang, C. C. (2008). Multilevel reversible data hiding based on histogram modification of difference images. *Pattern Recognition*, *41*(12), 3582–3591.
24. Ni, Z., Shi, Y. Q., Ansari, N., & Su, W. (2006). Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, *16*(3), 354–362.
25. Berners-Lee, T., Fielding, R. T., & Masinter, L. (2005). Uniform Resource Identifier (URI): Generic Syntax. Internet Society. RFC 3986; STD 66, Jan.
26. Favicon, (2008). http://en.wikipedia.org/wiki/Favicon.
27. Hu, Y., & Kwong, S. (2001). Wavelet domain adaptive visible watermarking. *Electronics Letters*, *37*(20), 1219–1220.
28. Hu, Y., Kwong, S., & Huang, J. (2006). An algorithm for removable visible watermarking. *IEEE Transactions on Circuits and Systems for Video Technology*, *16*(1), 129–133.

Hsiang-Cheh Huang received his Ph.D. from the Department of Electronics Engineering at National Chiao-Tung University in 2001. He is currently with Department of Electrical Engineering, National University of Kaohsiung in Taiwan.

Dr. Huang's research interests include digital watermarking, video compression, and error resilient coding. He has published over 70 papers, 12 book chapters, and 2 Taiwan patents.

Dr. Huang is an IEEE Senior Member. He has served as a TPC member for a variety of international conferences including the International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP, 2005–2009), and the International Conference on Security Technology (SecTech, 2008). He is also an Associate Editor for the International Journal of Innovative Computing, Information and Control (IJICIC) since 2005. For more details, please refer to the website at http://ee.nuk.edu.tw/~hchuang/.

Wai-Chi Fang received his Ph.D. from the Electrical Engineering Department at University of Southern California in 1992. He is currently the TSMC Distinguished Chair Professor of National Chiao-Tung University.

Dr. Fang's subjects of interest include VLSI multimedia signal processing and networking systems, VLSI neural networks and intelligent information systems, integrated bio-medical microsystems, and integrated space avionic systems. He has published over a hundred papers. He is the recipient of 1995 IEEE VLSI Transactions Best Paper Award. He holds seven US patents and thirteen NASA new technologies. He won the NASA Space Act award in 2002 and 2003.

Dr. Fang is an IEEE Fellow. He serves as the Vice President of IEEE Systems Council. He is also a member of the Board of Governor of IEEE Circuits and Systems Society. He serves on the Advisory Board of IEEE Systems Journal. He serves on organization committee and technical program committee of many international conferences and workshops. He served as Associated Editor for the IEEE Transactions on Very Large Scale Systems (1997–1998), IEEE CASS Circuit and Device Magazine (1999–2000), IEEE Transactions on Multimedia (2000–2001), and IEEE Transactions on Circuits and Systems I (2002–2003). He was Chairman of three IEEE CAS Technical Committees that include Nanoelectronics and Gigascale Systems TC, Multimedia Systems and Applications TC, and Neural Systems and Applications TC.