Far East Journal of Experimental and Theoretical Artificial Intelligence
Volume 4, Number 2, 2009, Pages 147-159
Published Online: March 5, 2010
This paper is available online at http://www.pphmj.com

ADAPTIVE HIGH CAPACITY RLE IMAGE DATA HIDING METHOD

WEN-CHUNG KUO[1], CHUN-CHENG WANG[2] and JIIN-CHIOU CHENG[2]

[1]Department of Computer Science and Information Engineering
National Formusa University
Taiwan, R. O. C.
e-mail: simonkuo@nfu.edu.tw

[2]Department of Computer Science and Information Engineering
Southern Taiwan University
Taiwan, R. O. C.

Abstract

Data hiding technology is that embedding the secret message into the cover image, forms the stego-image. The stego-image must keep well quality of image under high capacity of embedding for achieving security. In 2006, Chang et al. [2] proposed a data hiding method based on RLE which can carry beneficial result. Recently, Lin and Hu [8] proposed a similar high capacity RLE steganography which improves the compression ratio and image quality. However, their method is not excellent in aspect of adaptability. In order to improve this disadvantage, we will propose an adaptive high capacity RLE steganography in this paper. According to the experimental results, we can prove that our proposed scheme is not only to promote the adaptability of the high capacity RLE steganography but also to keep the better image quality when the encoding length is longer or the secret image is more complicated.

Keywords and phrases: run-length encoding, data-hiding, steganography.

This work is supported by National Science Council under (NSC 98-2219-E-150-001).

Communicated by Chuan-Yu Chang

Received November 15, 2009

1. Introduction

As the popularity of the Internet and the bandwidth is increasing rapidly, digital multimedia has become very popular in human's daily life and it is transmitted over the Internet increasing day by day. However, there are different dangers hidden behind the convenience of the internet; for instance, many personal private information can be intercepted, modified, or being used on any illegal matters by hackers during the data is being sent. Therefore, how to protect the digital multimedia security and intellectual property rights becomes a major problem during data transmission. Usually, steganography is one of widespread used network security technologies. Steganography can be divided into two relative categories: image watermarking and image data hiding. In fact, there are some different focuses between image watermarking and image data hiding. For image watermarking, its goals are aimed at increasing the powers of copyright protection [5, 7, 9] and authentication [4, 9] whereas the main aim of image data hiding is to increase the hiding capacity and imperceptibility. Here, our effort is dedicated in the topic of image data hiding. To avoid the malicious attack, the stego-image must keep a good image quality and enhance the capacity as far as possible at the same time. However, the quantity of secret message is usually very high when it is an image. Therefore, to achieve high embedding capacity, we usually compress the secret image before embedding it to keep the stego-image quality good.

Until now, there are many techniques about image data hiding proposed in literatures [2, 8]. Among them, the most well-known and simple steganographic technique is the least-significant-bit substitution (abbreviated as LSB) for achieving high embedding capacity usually. In 2006, Chang et al. [2] (abbreviated as CLW-scheme) proposed a new image data hiding method that scans a binary image in a "*Z*" order and then encodes it by Run-Length Encoding (abbreviated as RLE). This approach makes good compression ratio and the PSNR of the stego-image is quite high. However, different encoding length will affect the compression ratio. At the same time, the appropriate encoding length is decided for the secret binary image. In order to improve this disadvantage, Lin and Hu [8] (abbreviated as LW-scheme) proposed a high embedding capacity RLE image data hiding in 2009. They improved the encoding length double to enhance the compression ratio and make the stego-image quality better. However, LW-scheme is just only useful when the length of continuous the same bits is equal to or more than double encoding length. In

other words, the applicability of the scheme is the special case. To enhance their application category, we propose an adjustable high embedding capacity RLE image data hiding method and improve the encoding length double in this paper.

The remainder of this paper is organized as follows: In Section 2, we will introduce the RLE, RLE image data hiding method and high capacity RLE image data hiding method briefly. Then, we will propose the improvement scheme to overcome the applicability problem and give the experimental result in Section 3 and Section 4, respectively. Finally, conclusions will be drawn in Section 5.

2. Review the Data Hiding Scheme based on RLE Techniques

CLW's RLE method [2] and Lin-Hu's high capacity RLE data hiding method [8] are reviewed in Sections 2.2 and 2.3, respectively.

2.1. Run-length encoding (RLE)

The RLE concept was proposed in the 1950s and has become the common compression standard in fax transmissions and bitmap image coding. It encodes continuous 0s or 1s for less data. The encoding method is encoding the continuous 0s or 1s into (run_count, run_value), where run_count is the number of continuous 0s or 1s, and run_value is the value of run_count. For example, a binary message 000111100_2 can be encoded to $(3, 0)$, $(4, 1)$ and $(2, 0)$ by using RLE.

2.2. RLE image data hiding method [2]

In 2006, Chang et al. [2] proposed a new image data hiding method based on Run-Length Encoding. First, they scan the binary secret image with repeated bit strings in a "*Z*" order and then reduce these sequential bit strings by RLE method. Finally, these secret data will be embedded into a two-pixel block of coherent pixels in a cover image. Therefore, the embedded procedure in CLW-scheme is described as following steps:

Step 1. Decide the maximum encoding length k and encode the secret binary message to (run_count, run_value) by using RLE, where $run\_count \leqq k$.

Step 2. Let (c_i, v_i) be the gray-values of any two-pixel block in cover images and $c_i = k \times q_i + r_i$, where $q_i = \lfloor c_i / k \rfloor$ and $k > r_i \geqq 0$.

Step 3. Compute the value of *sign*:

$$sign = \begin{cases} 0 & \text{if} \quad r_i = (run\_count - 1), \\ 1 & \text{if} \quad (run\_count - 1) - r_i > 0, \\ -1 & \text{if} \quad (run\_count - 1) - r_i < 0. \end{cases}$$

Step 4. Compute *a* and *b*:

$$a = c_i + (sign)\,|\,(run\_count_i - 1) - r_i\,|,$$

$$b = c_i - (sign)(k - |\,(run\_count_i - 1) - r\,|).$$

Step 5. The pixel c_i' after embedding is:

$$c_i' = \begin{cases} a & \text{if} \quad |\,c_i - a\,| \le \lfloor k/2 \rfloor, \\ b & \text{otherwise.} \end{cases}$$

Step 6. Replace the least significant bit of v_i with the run_value_i to form v_i'.

Here, we give an example to explain the embedded procedure in CLW-scheme.

Example 1. We assume that the secret binary message is 0000011111111000_2, and the five pixel pairs in cover image are $(135, 84)$, $(99, 37)$, $(48, 204)$, $(62, 145)$ and $(53, 176)$.

Step 1. Decide maximum encoding length $k = 4$ and encode the binary message to $(4, 0)$, $(1, 0)$, $(4, 1)$, $(4, 1)$ and $(3, 0)$.

Step 2. Let $(c_1, v_1) = (135, 84)$, $(c_2, v_2) = (99, 37)$, $(c_3, v_3) = (48, 204)$, $(c_4, v_4) = (62, 145)$ and $(c_5, v_5) = (53, 176)$. Therefore, $r_1 = 3$, $r_2 = 3$, $r_3 = 0$, $r_4 = 2$ and $r_5 = 1$.

Step 3. Compute $sign_1 = 0$, $sign_2 = -1$, $sign_3 = 1$, $sign_4 = 1$, $sign_5 = 1$.

Step 4. Compute $a_1 = 135$, $b_1 = 135$, $a_2 = 96$, $b_2 = 100$, $a_3 = 51$, $b_3 = 47$, $a_4 = 63$, $b_4 = 59$, $a_5 = 54$ and $b_5 = 50$.

Step 5. Calculate $c_1' = 135$, $c_2' = 100$, $c_3' = 47$, $c_4' = 63$ and $c_5' = 54$.

Step 6. Compute $v_1' = 84$, $v_2' = 36$, $v_3' = 205$, $v_4' = 145$ and $v_5' = 176$.

So, we can find out the five pixel pairs in stego-image such as $(135, 84)$, $(100, 36)$, $(47, 205)$, $(63, 145)$ and $(54, 176)$. Subsequently, the extract procedure is described as following steps:

Step 1. Extract the codes:

$$run\_count_i = (c_i' \bmod k) + 1,$$

$$run\_value_i = v_i' \bmod 2.$$

Step 2. Decode the codes to the secret message.

Now, we assume that the receiver gets the stego-pixel pair $(135, 84)$. Then, he can calculate the $run\_count = (135 \bmod 4) + 1 = 4$ and the $run\_value = 84 \bmod 2 = 0$. So, the secret message is $(4, 0) = 0000_2$. Hence, we can recover the original secret message 0000011111111000_2 from the five pixel pairs $(135, 84)$, $(100, 36)$, $(47, 205)$, $(63, 145)$ and $(54, 176)$ in Example 1.

2.3. High capacity RLE data hiding method [8]

In CLW-scheme, they greatly improved the embedding capacity and kept the better image quality by using RLE. However, how to choose the value of k is very important. If the value of k is small, then the number of continuous the same bits is greater than k easily. Thus, this data must be split into two codes and the data compression efficacy will be reduced. Conversely, if the value of k is big, then the pixel value does not only change very large after embedding data but also reduce the stego-image's quality. To overcome this drawback, Lin and Hu [8] proposed a high embedding capacity RLE image data hiding scheme in 2009. They used an additional code value (add_bit) to make the maximum encoding length double and also keep the stego-image's quality as good as CLW-scheme. Their main approach is that divides the RLE code into three parts $(run\_count, add\_bit, run\_value)$, where the add_bit is used to show that the encoding length is larger than k or not. If the encoding length is larger than k, then $add\_bit = 1$. Otherwise, the $add\_bit = 0$. Now, the embedded procedure in Lin-Hu's scheme is described as following steps:

Step 1. Decide the maximum encoding length k and encode the secret binary image to $(run\_count, add\_bit, run\_value)$ by using RLE, where $run\_count \leqq k$.

Step 2. Let (c_i, v_i) be the gray-values of any two-pixel block in cover images and $c_i = k \times q_i + r_i$, where $q_i = \lfloor c_i / k \rfloor$ and $k > r_i \geqq 0$.

Step 3. Compute the value of *sign*:

$$sign = \begin{cases} 0 & \text{if} \quad r_i = (run\_count - 1), \\ 1 & \text{if} \quad (run\_count - 1) - r_i > 0, \\ -1 & \text{if} \quad (run\_count - 1) - r_i < 0. \end{cases}$$

Step 4. Compute *a* and *b*:

$$a = c_i + (sign)\,|\,(run\_count_i - 1) - r_i\,|,$$

$$b = c_i - (sign)(k - |\,(run\_count_i - 1) - r\,|).$$

Step 5. The pixel c_i' after embedding is:

$$c_i' = \begin{cases} a & \text{if} \quad |\,c_i - a\,| \leq \lfloor k/2 \rfloor, \\ b & \text{otherwise.} \end{cases}$$

Step 6. Replace the least significant bit of v_i with the run_value_i to form v_i'.

Step 7. Replace the second least significant bit of v_i' with the add_bit_i.

According to Example 1, we give the same example to explain the embedded procedure in the Lin-Hu's scheme.

Example 2. It assumes that the secret binary message is 0000011111111000_2 and only four pixel pairs in the cover image are needed $(135, 84)$, $(99, 37)$, $(48, 204)$ and $(62, 145)$.

Step 1. Decide maximum encoding length $k = 4$ and encode the binary message to $(4, 0, 0)$, $(1, 0, 0)$, $(4, 1, 1)$, $(3, 0, 0)$.

Step 2. Let $(c_1, v_1) = (135, 84)$, $(c_2, v_2) = (99, 37)$, $(c_3, v_3) = (48, 204)$ and $(c_4, v_4) = (62, 145)$. Therefore, $r_1 = 3$, $r_2 = 3$, $r_3 = 0$ and $r_4 = 2$.

Step 3. Compute $sign_1 = 0$, $sign_2 = -1$, $sign_3 = 1$, and $sign_4 = 0$.

Step 4. Compute $a_1 = 135$, $b_1 = 135$, $a_2 = 96$, $b_2 = 100$, $a_3 = 51$, $b_3 = 47$, $a_4 = 62$ and $b_4 = 62$.

Step 5. Calculate $c'_1 = 135,\ c'_2 = 100,\ c'_3 = 47,$ and $c'_4 = 62.$

Step 6. Compute $v'_1 = 84,\ v'_2 = 36,\ v'_3 = 205$ and $v'_4 = 144.$

Step 7. Compute $v'_1 = 84,\ v'_2 = 36,\ v'_3 = 207$ and $v'_4 = 144.$

So, we can find out the four pixel pairs $(135, 84),\ (100, 36),\ (47, 204)$ and $(62, 144)$ in the stego-image. Subsequently, the extract procedure is described as following steps:

Step 1. Extract the codes:

$$run\_count_i = (c'_i \bmod k) + 1,$$

$$add\_bit_i = \lfloor (v'_i \bmod 4)/2 \rfloor,$$

$$run\_value_i = v'_i \bmod 2.$$

Step 2. Decode the codes to the secret message.

After the receiver gets the first stego-pixel pair $(135, 84),$ he can calculate $run\_count = (135 \bmod 4) + 1 = 4,\ add\_bit = \lfloor (84 \bmod 4)/2 \rfloor = 0$ and $run\_value = 84 \bmod 2 = 0.$ So, the secret message is $(4, 0, 0) = 0000_2.$ Hence, the original secret binary messages 0000011111111000_2 are recovered from these pixel pairs $(135, 84),\ (100, 36),\ (47, 207)$ and $(62, 144)$ in Example 2.

3. Adaptive High Capacity RLE Image Data Hiding Method

Recently, Lin and Hu [8] used the maximum encoding length double to improve the secret data embedding capacity. However, this method is effective just only the number of continuous the same bits is equal to k or $2k.$ However, the number of continuous the same bits k' is greater than k and less than $2k$, in other words, $k < k' < 2k,$ it is still split into two codes in Lin-Hu scheme. Therefore, when the value of k is large, the efficiency of Lin-Hu scheme will be reduced. To improve this weakness, we also classify the secret codes into three parts, i.e., $(run\_count,\ odd\_even, run\_value),$ where $odd\_even \in \{0, 1\}.$ The encoding length k' can be expressed as $k' - 1 = run\_count \times 2 + odd\_even.$ So, the $k' \in [k, 2k]$ can be also encoded into one code. Now, we will describe the embedded procedure as following steps:

Step 1. Decide the maximum encoding length $k' = 2k$ and encode the secret binary message to $(run\_count, odd\_even, run\_value)$ by using RLE, where $run\_count \leqq k$.

Step 2. Let (c_i, v_i) be the gray-values of any two-pixel block in cover images and $c_i = k \times q_i + r_i$, where $q_i = \lfloor c_i / k \rfloor$ and $k > r_i \geqq 0$.

Step 3. Compute the value of *sign*:

$$sign = \begin{cases} 0 & \text{if} \quad r_i = run\_count, \\ 1 & \text{if} \quad run\_count - r_i > 0, \\ -1 & \text{if} \quad run\_count - r_i < 0. \end{cases}$$

Step 4. Compute a and b:

$$a = c_i + (sign)| run\_count - r_i |,$$

$$b = c_i - (sign)(k - | run\_count_i - r |).$$

Step 5. The pixel c_i' after embedding is:

$$c_i' = \begin{cases} a & \text{if} \quad | c_i - a | \leq \lfloor k/2 \rfloor, \\ b & \text{otherwise.} \end{cases}$$

Step 6. Replace the least significant bit of v_i with the run_value_i to form v_i'.

Step 7. Replace the second least significant bit of v_i' with the odd_even_i.

As usual, we give the same example to explain the embedded procedure in our scheme.

Example 3. We assume that the secret binary message is 0000011111111000_2 and we just only need three pixel pairs $(135, 84)$, $(99, 37)$ and $(48, 204)$ in cover image.

Step 1. Decide maximum encoding length $k' = 8$ and encode the binary message to $(2, 0, 0)$, $(3, 1, 1)$, $(1, 0, 0)$.

Step 2. Let $(c_1, v_1) = (135, 84)$, $(c_2, v_2) = (99, 37)$ and $(c_3, v_3) = (48, 204)$. Therefore, $r_1 = 3$, $r_2 = 3$ and $r_3 = 0$.

Step 3. Compute $sign_1 = -1,\ sign_2 = 0$ and $sign_3 = 1$.

Step 4. Compute $a_1 = 134,\ b_1 = 138,\ a_2 = 99,\ b_2 = 99,\ a_3 = 49$ and $b_3 = 45$.

Step 5. Calculate $c_1' = 134,\ c_2' = 99$ and $c_3' = 49$.

Step 6. Compute $v_1' = 84,\ v_2' = 37$ and $v_3' = 204$.

Step 7. Compute $v_1' = 84,\ v_2' = 39$ and $v_3' = 204$.

So, we can find out the three stego-pixel pairs such as $(134, 84)$, $(99, 39)$ and $(49, 204)$. Subsequently, the extract procedure is described as following steps:

Step 1. Extract the codes:

$$run\_count_i = c_i' \bmod k,$$

$$odd\_even_i = \lfloor (v_i' \bmod 4)/2 \rfloor,$$

$$run\_value_i = v_i' \bmod 2.$$

Step 2. Decode the codes to the secret message.

After the receiver gets the first stego-pixel pair $(134, 84)$, he can calculate the $run\_count = (134 \bmod 4) + 1 = 2$, the $add\_bit = \lfloor (84 \bmod 4)/2 \rfloor = 0$ and the $run\_value = 84 \bmod 2 = 0$. So, the secret message is $(2, 0, 0)$. Then he calculates the $k' = 2 \times 2 + 0 + 1 = 5$ and decodes to 00000_2. Hence, the original secret message 0000011111111000_2 is recovered from these stego-pixel pairs $(134, 84)$, $(99, 39)$ and $(49, 204)$ in Example 3.

4. Experimental Result

For testing the results, the standard 512×512 grayscale image was used as the test cover image as shown in Figure 1 and the four 512×512 secret binary images (as shown in Figures 2(a)-2(d)) were also used in the experiments. Here, the stego-image quality is determined by PSNR (peak signal-to-noise ratio), i.e., the stego-image with a larger PSNR value indicating higher image quality. Figures 3(a)-3(d) show the stego-images that were made by using our proposed method from the secret images in Figures 2(a)-2(b) with the same cover image. As shown in these images, the distortions are not obvious to the human eye. At the same, Tables 1 and 2 show the results of detailed comparisons of various methods in terms of PSNR

value with different secret images when $k = 4$ and $k = 9$, respectively. According to the simulation results in Tables 1 and 2, we can find out that when the value of k is larger, the PSNR of our proposed method is better than CLW-scheme and Lin-Hu's scheme. In other words, the number of continuous the same bits k' between k and $2k$, will encode k' into two codes by using Lin-Hu's method when k becomes larger. However, by using our proposed method encode it into one code. Hence, our proposed method has good applicability to reduce the number of codes and further improve the stego-image's quality. Specially, this application has bettered a lot when the secret image is more complicated such as Baboon.

Figure 1. The cover image: Lena.

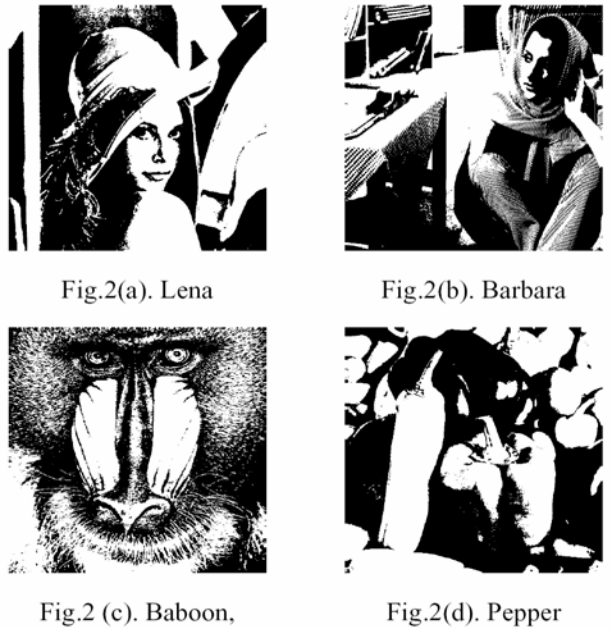

Figure 2. The secret binary image.

Fig.3(a). Lena+Lena

Fig.3(b). Lena+Barbara

Fig.3(c). Lena+Baboon,

Fig.3(d). Lena+Pepper

Figure 3. The stego-image.

Table 1. The PSNR comparison table $(k = 4)$

Secret images	CLW-scheme [2]	Lin-Hu's scheme [8]	Our proposed scheme
Barbara	49.1 dB	49.0 dB	49.1 dB
Baboon	45.8 dB	45.3 dB	48.0 dB
Lena	49.5 dB	49.5 dB	50.4 dB
Pepper	50.1 dB	50.3 dB	50.6 dB

Table 2. The PSNR comparison table $(k = 9)$

Secret images	CLW-scheme [2]	Lin-Hu's scheme [8]	Our proposed scheme
Barbara	41.7 dB	41.8 dB	46.8 dB
Baboon	36.7 dB	36.6 dB	45.7 dB
Lena	43.1 dB	43.5 dB	50.1 dB
Pepper	44.9 dB	45.6 dB	50.7 dB

5. Conclusion

In this paper, we propose an adaptive RLE data hiding method by changing the RLE's format. This proposed method can be represented by any binary secret value between k and $2k$ by using run_count and odd_even. According to our experiment, our proposed method does not only to enhances the compression ratio and the stego-image's quality but also to improve the drawback in Chang's scheme and Lin-Hu's scheme, in special, when the value of k is from 4 to 9, the PSNR decreasing rate of our proposed method is better than the others.

References

[1] C.-K. Chan and L. M. Cheng, Hiding data in images by simple LSB substitution, Pattern Recognition 37(3) (2004), 469-474.

[2] C.-C. Chang, C.-Y. Lin and Y.-Z. Wang, New image steganographic methods using run-length approach, Inform. Sci. 176(22) (2006), 3393-3408.

[3] C.-C. Chang and H. W. Tseng, A steganographic method for digital images using side match, Pattern Recognition Lett. 25(12) (2004), 1431-1437.

[4] J. Fridrich, Image watermarking for tamper detection, Proc. of the IEEE International Conference on Image Processing, Chicago, IL, USA, 1998, pp. 404-408.

[5] S. Katzenbeisser and F. A. P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Norwood, MA, 2000.

[6] W.-C. Kuo, D.-J. Jiang and Y.-C. Huang, Reversible data hiding based on histogram, Lecture Notes in Computer Science 4682, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 1152-1161.

[7] W.-N. Lie and L. C. Chang, Data hiding in images with adaptive numbers of least significant bits based on the human visual system, Proc. of IEEE International Conference on Image Processing, Kobe, Japan, 1999, pp. 286-290.

[8] R.-S. Lin and S.-W. Hu, A modified run-length image data hiding for high embedding capacity, Fifth International Conference on Information Assurance and Security-16, 2009.

[9] C.-S. Lu and H.-Y. M. Liao, Multipurpose watermarking for image authentication and protection, IEEE Transactions on Image Processing 10(10) (2001), 1579-1592.

[10] J. Mielikainen, LSB matching revisited, IEEE Signal Processing Letters 13(5) (2006), 285-287.

[11] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Techn. 13(8) (2003), 890-896.

[12] R.-Z. Wang, C.-F. Lin and J.-C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recognition 34(3) (2001), 671-683.