International Journal of Security and Its Applications
Vol. 9, No. 5 (2015), pp. 243-254
http://dx.doi.org/10.14257/ijsia.2015.9.5.24

Image Steganography via Fully Exploiting Modification Direction

Xuejing Niu, Meng Ma*, Rui Tang and Zhaoxia Yin*

Key Laboratory of Intelligent Computing & Signal Processing, School of Computer Science & Technology, Anhui University, Hefei, China
niuxj2012@qq.com; mengma2@gmail.com; yinzhaoxia@ahu.edu.cn

Abstract

Embedding bits by replacing pixel's least-significant bit (LSB) is a simple but great idea. Nevertheless, this method is not safe enough. Afterwards, a series of effective and safe algorithms which are based on LSB substitution method were proposed such as EMD (exploiting modification direction) and EMD-2. EMD, modifying at most one of n *pixels' LSB to embed a* $(2n+1)$*-ary notational system number, achieved higher embedding efficiency than LSB method. EMD-2 modifys at most two of a group* n *pixels to embed a* $(10n-13)$*-ary* $(n>2)$ *notational system number, making embedding capacity improved compared with EMD. To improve embedding capacity further, we proposed a new method EMD-3 exploiting modification direction fully. EMD-3 modifys a group of* n *pixels by* ± 1 *to embed a secret digit in* 3^n*-ary notational system. The experimental results demonstrate that EMD-3 can improve embedding capacity and preserve good image quality at the same time. Besides, EMD-3 is as secure as EMD and EMD-2.*

Keywords: *Steganography, Modification Direction, Embedding Capacity, Image Quality, Security*

1. Introduction

The modern digit steganography can be divided into the following branches: digital media steganography, linguistic steganography, file system steganography and network steganography [1, 2]. Generally, people evaluate steganography algorithm from three aspects: embedding capacity, image quality and security. Embedding capacity is estimated with embedding rate ER. ER represents the number of bit embedded in each pixel and its unit is *bpp* (bit per pixel). Image quality is often estimated with PSNR (peak signal to noise ratio), MSSIM (mean structure similarity) [3]. However, improving ER is irreconcilable with reducing image distortion, so some algorithms focus on embedding capacity, such as [4-6], while some focus on image quality, such as [7-11].

LSB substitution method isn't safe since the presence of secret message could be detected by steganalysis. Then a series of methods were proposed after LSB substitution method. In 2006, Mielikainen proposed LSB-MR (LSB matching revisited) algorithm [12]. LSB-MR hides as many bits as LSB does. And utilizing equation (1), only one of two pixels (g_1, g_2) needs to be modified.

$$f(g_1, g_2) = LSB\left(\lfloor g_1/2 \rfloor + g_2\right) \tag{1}$$

Then Zhang and Wang proposed EMD [13]. EMD transforms secret data into $(2n+1)$-ary notational system digits. Every digit is carried by n pixels, and at most one pixel was allowed to be modified by ± 1. Later, in 2010, Kim et al. proposed EMD-2 [14] allowing two pixels to be modified by ± 1. EMD-2's capacity is much larger than EMD.

* Corresponding Author

ISSN: 1738-9984 IJSIA

In summary, [12-14] have improved a lot compared with LSB substitution method. However, modification direction still isn't exploited fully.

The proposed method EMD-3 can be implemented as easily as [13-14] can. Compared with [13-14], the embedding capacity of EMD-3 greatly improved. EMD-3's ER is higher than [14], and much higher than [13]. What's more, EMD-3's image quality is similar to [14], and slightly lower than [13].

Then, next is a brief introduction of related works. The third part is the proposed scheme. The forth part is experimental results, and the last part is conclusion.

2. Related Works

2.1. EMD Method

Zhang and Wang's novel method EMD transforms secret data into several $(2n+1)$-ary notational system digits, and each digit is carried by n pixels. During each embedding procedure, at most one pixel's value would be increased or decreased by 1, then there're $2n$ options of modification and 1 option without modification, totally $(2n+1)$ options, mapping to different digits. A group of n pixels are expressed as an *n*-dimensional vector $G_n = [g_1, g_2, ..., g_n]$, and calculate function f_1 as weighted sum modulo $(2n+1)$.

$$f_1(G_n) = \left\lceil \sum_{i=1}^{n} (g_i \times i) \right\rceil \bmod (2n+1) \tag{2}$$

If f_1 equals secret digit *d*, no pixels need to be modified. If not, we should calculate $s = (d - f_1) \bmod (2n+1)$. If s isn't greater than n, g_s plus 1, otherwise, g_{2n+1-s} minus 1. Receiver can get the secret digit easily by calculating function f_1 with modified pixels.

EMD's embedding rate achieves 1.585*bpp* where $n=1$, and 1.16*bpp* where $n=2$. Compared with LSB, EMD's embedding capacity greatly improved.

2.2. EMD-2 Method

Another novel embedding method EMD-2 is based on $(2w+1)$-ary notational system, and each secret data is carried by a group of n pixels, too. When $n=2$, w is 4; when $n>2$, w is $8+5(n-3)$. Before embedding procedure, we need to construct an n-dimension basis vector B_n first. When $n=2$, the best option of B_2 is $[1,3]$; when $n>2$, the best option of B_n is $[1,2,...,6+5(n-3)]$ [14]. In order to exploit the modification directions further, EMD-2 allows two pixels to be modified, which implies at most two pixels' value would increase or decrease by 1. Different modifications map to different secret digits, resulting in higher embedding efficiency. *n*-dimension vector $G_n = [g_1, g_2, ..., g_n]$ consists of n cover pixels' value. Extraction function f_2 is inner product of G_n and B_n modulo $(2w+1)$.

$$f_2(G_n, B_n) = \left\lceil \sum_{i=1}^{n} (g_i \times b_i) \right\rceil \bmod (2w+1) \tag{3}$$

$$s = \left\lceil \sum_{i=1}^{n} (s_i \times b_i) \right\rceil \bmod (2w+1) \tag{4}$$

If f_2 equals secret data d, we needn't modify any pixel. Otherwise, we should calculate $s = (d - f_2) \bmod (2w+1)$. Then find s's *n*-dimension coefficient vector $C_s = [s_1, s_2, ..., s_n]$ which makes equation (4) true (the possible value of s_i is -1,1,0). At

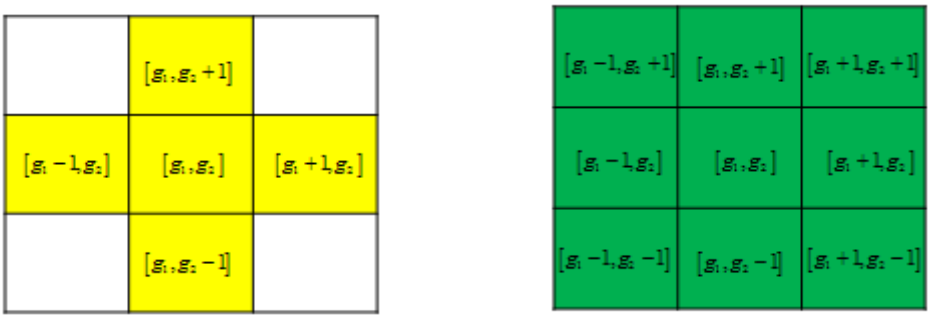
last, the modified pixels $G_n' = G_n + C_s$. Receiver obtains the secret digit easily by calculating function f_2 as long as he knows modified pixels.

3. Main Focus of the Paper

EMD can embed a $(2n+1)$-ary notational system number with 1 of n pixels modified; EMD-2 can embed a $(10n-13)$-ary $(n>2)$ notational system with 2 of n pixels modified. Since the modification direction of cover pixels is exploited further, EMD-2 achieves larger embedding capacity. However, EMD-2's exploitation is still not full.

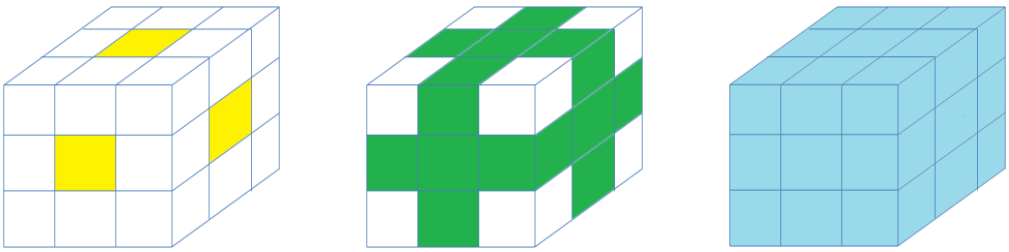
3.1. Modification Direction Exploiting

Since n cover pixels can construct an *n*-dimension vector $G_n = [g_1, g_2, \ldots, g_n]$, so modification direction exploiting could be implemented in *n*-dimension space. When $n=2$, $G_2 = [g_1, g_2]$, only one pixel could be modified by ± 1 in EMD. Thus, five modification directions, $\{[g_1, g_2+1], [g_1-1, g_2], [g_1, g_2], [g_1+1, g_2], [g_1, g_2-1]\}$, are exploited as shown in Figure 1 (a). In EMD-2, two pixels can be modified by ± 1. Thus nine modification directions, $\{[g_1-1, g_2-1], [g_1, g_2-1], \ldots, [g_1+1, g_2+1]\}$, are exploited as shown in Figure 1 (b). Obviously, (b) exploits modification direction fully, while (a) not.



(a) (b)

Figure 1. Modification Directions Exploited by Different Methods when *n*=2 (a) EMD, (b) EMD-2



(a) (b) (c)

Figure 2. Modification Directions Exploited by Different Methods when *n*=3 (a) EMD, (b) EMD-2, (c) EMD-3

When $n=3$, it's not convenient to write pixels' value in cubes, so we only represent modification directions with colored cubes in 3-dimension space shown in Figure 2. The center of the big cube is $[g_1, g_2, g_3]$. In EMD, shown as Figure 2 (a), 7 blocks can be colored with at most one pixel modified, so 7 directions have been exploited. In EMD-2,

nineteen blocks can be colored with at most two pixels modified, so 19 directions have been exploited. But in fact, EMD-2 just exploited 17 directions. To exploit modification directions of n pixels fully, as shown in Figure 2 (c), EMD-3 is proposed here.

4. Proposed Method

In EMD-3, every pixel is allowed to be modified. We find a group optimal weight $[1,3,\ldots,3^{n-1}]$ for n pixels $[g_1, g_2, \ldots, g_n]$, which makes a 3^n-ary notational system to be constructed with n pixels. Thus, when $n > 2$, EMD-3's embedding capacity is much larger than [13-14].

4.1. Embedding & Extracting

For a group of n pixels, each pixel has three possible presentations after modification: plus 1, minus 1, and invariant, mapping to 3^n digits in total. We can calculate extraction function f_3 as weighted sum modulo 3^n.

$$f_3(G_n) = \left\lceil \sum_{i=1}^{n} \left(g_i \times 3^{i-1} \right) \right\rceil \bmod 3^n \tag{5}$$

If $f_3(G_n)$ exactly equals secret digit *d*, *d* can be embedded in n pixels with no modification; if not, we calculate $s = \left(d - f(G_n)\right) \bmod 3$, then we should modify n pixels one by one. For the i th pixel, we defined function.

$$f_4(i) = \left\lfloor \left(s - \frac{3^{i-1} - 1}{2} - 1 \right) / 3^{i-1} \right\rfloor \bmod 3 \tag{6}$$

When $s > \left(3^{i-1} - 1\right)/2$ and $f_4(i) = 0$, increase g_i by 1; when $s > \left(3^{i-1} - 1\right)/2$ and $f_4(i) = 1$, decrease g_i by 1.

There's a simple example: assume that cover pixels construct 3-dimension vector $G_3 = [31, 22, 32]$, and we want to embed 27-ary secret digit 26, so $f_3(G_3) = 7$, $s = 19$. According to (6), for $g_1 = 31$, $s > \left(3^0 - 1\right)/2$ and $f_4(1) = 0$, so g_1 need plus 1; for $g_2 = 22$, $s > \left(3^1 - 1\right)/2$ and $f_4(2) = 2$, so g_2 is not modified; for $g_3 = 32$, $s > \left(3^2 - 1\right)/2$ and $f_4(3) = 1$, so g_3 need minus 1. The vector $G_3^{'}$ is $[32, 22, 31]$.

The extracting procedure of EMD-3 is as easy as EMD and EMD-2. As long as receiver calculates $f_3(G_n)$ using modified image pixels, $f_3(G_n)$ is precisely the embedded digit.

There's an example: assume receiver receives modified image and constructs 3-dimension vector $G_3 = [32, 22, 31]$, then calculates $f_3(G_3) = 26$, which is precisely what embedded above.

5. Experimental Results

In this part, all experiments were performed with software Matlab 2013a. Cover images used here are commonly used 512×512 grayscale images such as Lena, Peppers, Splash, Baboon, Jet, Boat, Sailboat.

5.1. Embedding Capacity

Embedding capacity is often evaluated by embedding rate (ER, bit per pixel). Compared with previous methods, the advantage of our method is large embedding

capacity. For a group of n pixels, the proposed method can hide greater numbers. For example, with 3 pixels, EMD can hide a 7-ary number; EMD-2 can hide 17-ary number; while EMD-3 can hide 27-ary number. Then the maximum ER of EMD is 0.9358; EMD-2, 1.3625; EMD-3, 1.585. We make a simple comparison in Table 1.

Table 1. The Biggest Number can be Embedded in *n* Pixels with EMD, EMD-2 and EMD-3

	EMD	EMD-2	EMD-3
$n = 1$	3-ary		3-ary
$n = 2$	5-ary	9-ary	9-ary
$n = 3$	7-ary	17-ary	27-ary
$n = 4$	9-ary	27-ary	81-ary
$n = 5$	11-ary	37-ary	243-ary

When n is very large, EMD-3 is much more efficient than EMD and EMD-2. EMD can hide a $(2n+1)$-ary digit in n pixels, so

$$ER_{EMD} = \frac{\log_2(2n+1)}{n} \tag{7}$$

According to the property of function, ER_{EMD} is a monotonically decreasing function. EMD's optimal ER is achieved only when $n = 1$. EMD-2 can hide 9-ary digit when $n = 2$, $(10n-13)$-ary when $n > 2$, so

$$ER_{EMD-2} = \begin{cases} \log_2 3 & n = 2 \\ \dfrac{\log_2(10n-13)}{n} & n > 2 \end{cases} \tag{8}$$

ER_{EMD-2} is greater than ER_{EMD}, but its property is similar to ER_{EMD}, and the best ER is achieved when $n = 2$. EMD-3 can hide 3^n–ary digit, so

$$ER_{EMD-3} = \frac{\log_2 3^n}{n} = \log_2 3 \tag{9}$$

Obviously, ER_{EMD-3} is the largest one and remains unchanged when n increase.

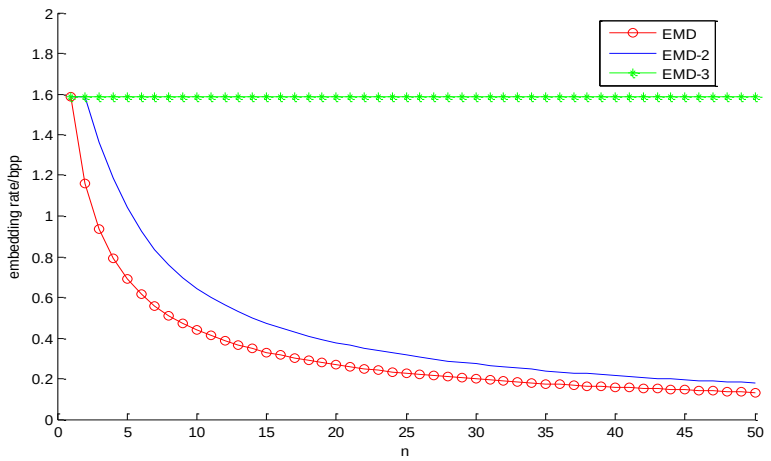


Figure 3. The Variation Tendency of Embedding Rate of EMD, EMD-2 and EMD-3

As shown in Figure 3, EMD-3 has great advantage in ER compared with [13-14]. In EMD, at most one of n pixel is allowed to be modified, which ensured good image quality but resulted in ER's decrement when n increased. In EMD-2, at most 2 of n pixels are allowed to be modified. Compared with EMD, its embedding rate improved, but it didn't solve the problem that embedding rate would decrease when n increased. Nevertheless, the proposed algorithm fully exploited modification direction, and greatly improved embedding rate on the basis of previous works.

5.2. Image Quality

Except embedding capacity, stego image quality is used to evaluate information hiding method, too. In this paper, PSNR and MSSIM are applied to image quality evaluation. PSNR is a usual criterion for image quality evaluation, and its formula is as follow:

$$MSE = \frac{1}{H \times W} \sum_{i=1}^{H \times W} \left(I_i - I_i' \right)^2 \tag{10}$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{ (dB)} \tag{11}$$

$H \times W$ is the size of image, and I_i, I_i' is the i th pixels of cover image and stego image respectively. People can't distinguish the difference of two images when PSNR is greater than 30dB.

To compare stego image quality of different methods with the same n, several cover images are used, and the results are similar. Here take 512×512 Lena as an example. Figure 4 (a) is cover image. Figure 4 (b) is stego image generated by EMD with $n = 3$ and maximum ER 0.9358*bpp*. The PSNR of (b) is 53.58dB which is the best among (b)(c)(d). Figure 4 (c) is generated by EMD-2 with $n = 3$, $ER = 1.3625$ *bpp*. And its PSNR is 50.73dB. Figure 4 (d) is generated by EMD-3 with $n = 3, ER = 1.585$ *bpp*. And its PSNR is 49.89 dB which is similar to EMD-2, slightly lower than EMD. But note that EMD-3's ER is much larger than EMD and EMD-2. Although EMD-3 modified more pixels than previous methods, the image quality is acceptable.

(a) Original Image

(b) EMD(*n*=3,ER:0.935*bpp*,PSNR:53.58dB)

(c)EMD-2(*n*=3,ER:1.362*bpp*,PSNR:50.73dB) (d)EMD3(*n*=3,ER:1.585*bpp*,PSNR:49.89dB)

Figure 4. Images with Full Payload

In addition, stego image quality of EMD-2 and EMD-3 for the same ER is tested with Lena shown in Figure 5, which demonstrates that the PSNR of EMD-3 is almost the same as EMD-2's with the same ER. However, EMD-3's maximum ER is much larger than EMD-2.

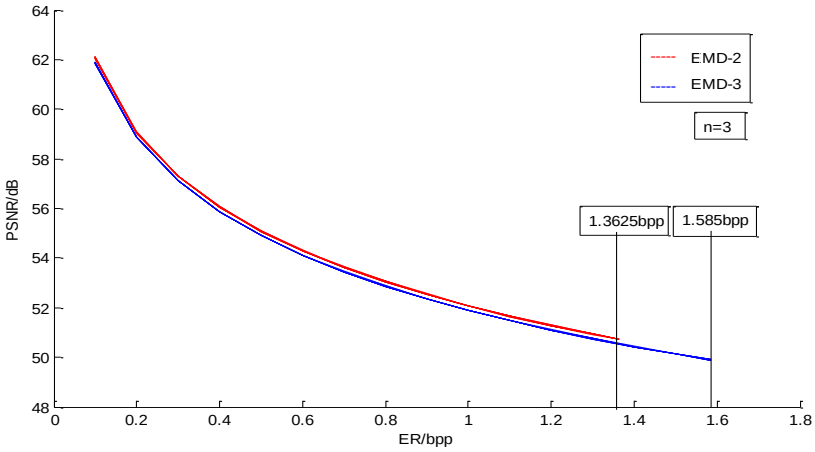


Figure 5. PSNR-ER of EMD-2 and EMD-3

To evaluate the image quality of EMD-3 further, another commonly used indicator, MSSIM, is adopted. MSSIM compares two image from three aspects defined as follow:

$$\text{Luminance comparison: } \begin{cases} l(x,y) = \dfrac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\ \mu_x = \dfrac{1}{N}\sum_{i=1}^{N} x_i \\ \mu_y = \dfrac{1}{N}\sum_{i=1}^{N} y_i \\ C_1 = (K_1 L)^2 \end{cases} \tag{12}$$

$$\text{Contrast comparison: } \begin{cases} c(x,y) = \dfrac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ \sigma_x = \left(\dfrac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_x)^2 \right)^{\frac{1}{2}} \\ \sigma_y = \left(\dfrac{1}{N-1} \sum_{i=1}^{N} (y_i - \mu_y)^2 \right)^{\frac{1}{2}} \\ C_2 = (K_2 L)^2 \end{cases} \tag{13}$$

$$\text{Structure comparison: } \begin{cases} s(x,y) = \dfrac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \\ \sigma_{xy} = \dfrac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_x)(y_i - \mu_y) \\ C_3 = \dfrac{1}{2} C_2 \end{cases} \tag{14}$$

$$SSIM(x,y) = \left[l(x,y) \right]^{\alpha} \times \left[c(x,y) \right]^{\beta} \times \left[s(x,y) \right]^{\gamma} \tag{15}$$

$$MSSIM = \frac{1}{M} \sum_{j=1}^{M} SSIM(x_j, y_j) \tag{16}$$

According to Ref. [3], we divide images into *M* groups, each containing *N* pixels, and set $K_1 = 0.01, K_2 = 0.03, L = 255$, $\alpha = \beta = \gamma = 1$. The biggest MSSIM value is 1. Two images are similar to each other when MSSIM close to 1. Seven 512×512 images: Lena, Peppers, Splash, Baboon, Jet, Boat, Sailboat are used to test EMD-3's MSSIM value here. Table 2 shows that MSSIM of stego images with $ER = 1.585bpp$ are all above 0.99.

Table 2. MSSIM of EMD-3 with ER=1.585*bpp*

	Lena	Peppers	Splash	Baboon	Jet	Boat	Sailboat
n=2	0.9952	0.9957	0.9940	0.9986	0.9946	0.9967	0.9966
n=3	0.9952	0.9956	0.9939	0.9986	0.9946	0.9967	0.9967

5.3. Security Analysis

Security is a significant problem for steganography. It has been proved that EMD and EMD-2 are robust against steganalysis. Difference histogram is used here to prove that EMD-3 is also secure. Take Lena as example. Difference histogram of EMD and EMD-3 when $ER = 1bpp,\ n = 2$ as (a), (b) in Figure 6 show; difference histogram of EMD-2 and EMD-3 when $ER = 1bpp, n = 3$, as (c), (d) in Figure 6 show. All of them closely match original image's difference histogram, so EMD-3 is as robust as EMD and EMD-2.

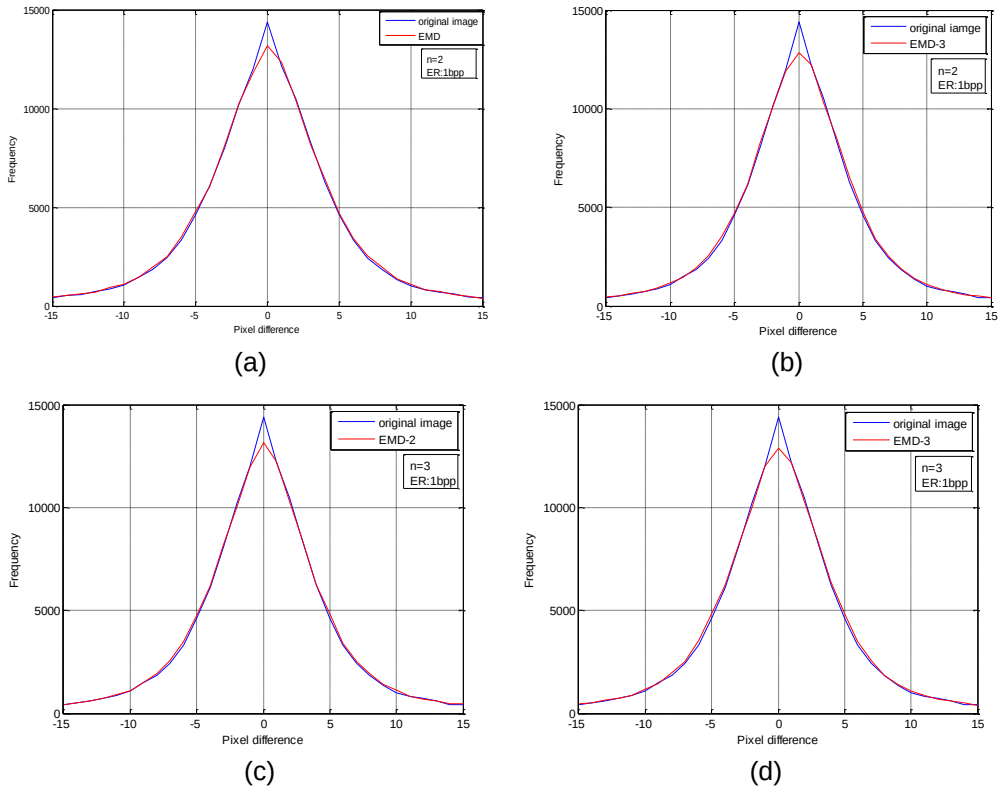


Figure 6. Difference Histogram of (a) EMD and (b) EMD-3 when n=2,ER=1*bpp*; (c) EMD-2 and (d) EMD-3 when n=3,ER=*bpp*

6. Conclusion

In this paper, we proposed an algorithm EMD-3. For a group of n cover pixels, EMD-3 exploits modification direction further and makes full use of redundancy space to embed data. With n pixels, EMD can embed a digit in $(2n+1)$-ary notational system; EMD-2 can embed a digit in $(10n-13)$-ary $(n>2)$ notational system; EMD-3 can embed a digit in 3^n-ary notational system since exploiting modification direction fully. Thus, EMD-3 outperforms [13-14] in embedding capacity. Experimental result shows that EMD-3 can preserve good image quality and is as robust as EMD and EMD-2 to difference histogram analysis. At last, the embedding and extracting procedure are simple and easy to implement which makes EMD-3 can be universally used.

Acknowledgments

This research work is supported by National Natural Science Foundation of China (No. 61502009, No. 6130005), Anhui Provincial Natural Science Foundation (No. 1508085SQF216), Project gxyqZD2016011 supported by the Key Program for Excellent Young Talents in Colleges, Universities of Anhui Province and Quality Engineering Program for Colleges and Universities in Anhui Province (2015jyxm042) and Undergraduates Training Foundation of Anhui University (J18520229, J18511158, J18515316).

References

[1] E. Zielińska, W. Mazurczyk and K. Szczypiorski, “Trends in Steganography”, Communications of the ACM, vol. 57, no. 3, **(2014)**, pp. 86-95.

[2] A. Ker, P. Bas, R. Böhme, R. Cogranne, C. Scott, T. Filler, J. Fridrich and T. Pevny, "Moving steganography and steganalysis from the laboratory into the real world", ACM Information Hiding and Multimedia Security Workshop, **(2013)** June 45-58, Montpellier, France.
[3] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity", IEEE Transaction on Image Processing, vol. 13, no. 4, **(2004)**, pp. 600-612.
[4] Z. X. Yin, J. Tang, Y. J. Liu and B. Luo, "Data hiding algorithm with high payload based on pixel pair matching", Systems Engineering-Theory & Practice, vol. 33, no. 11, **(2013)**, pp. 2972-2979.
[5] J. Chen, C. W. Shiu and M. C. Wu, "An improvement of diamond encoding using characteristic value positioning and modulus function", Journal of Systems and Software, vol. 86, no. 5, **(2013)**, pp. 1377-1383.
[6] C. C. Wang, Y. F. Chang, C. C. Chang, J. K. Jan and C. C. Lin, "A high capacity data hiding scheme for binary images based on block pattern", Journal of Systems and Software, vol. 93, **(2014)**, pp. 152-162.
[7] W. Hong and T. S. Chen, "A novel data embedding method using adaptive pixel pair matching", IEEE Transaction on Information Forensics and Security, vol. 7, no. 1, **(2012)**, pp. 176-184.
[8] W. Hong, "Adaptive image data hiding in edges using patched reference table and pair-wise embedding technique", Information Science, vol. 221, no. 1, **(2013)**, pp. 473-489.
[9] C. Y. Wang and X. H. Li, "A digital watermarking embedding algorithm based on biorthogonal wavelet transform for color images", Journal of Anhui University, vol. 37, no. 6, **(2013)**, pp. 65-71.
[10] F. Wei, D. Liang, C. Zhang and W. X. Bao, "Watermarking algorithm for digital image based on compressive sensing measurements", Journal of Anhui University, vol. 37, no. 3, **(2013)**, pp. 61-68.
[11] C. Y. Weng, Y. H. Zhang, L. C. Lin and S. J. Wang, "Visible watermarking images in high quality of data hiding", Journal of Supercomputing, vol. 66, no. 2, **(2013)**, pp. 1033-1048.
[12] J. Mielikainen, "LSB Matching Revisited", IEEE Signal Processing Letters, vol. 13, no. 5, **(2006)**, pp. 285-287.
[13] X. P. Zhang and S. Z. Wang, "Efficient Steganographic Embedding by Exploiting Modification Direction", IEEE Communications Letters, vol. 10, no. 11, **(2006)**, pp. 781-783.
[14] H. J. Kim, C. Kim, Y. Choi, S. Wang and X. Zhang, "Improved Modification Direction Methods", Computers and Mathematics with Applications, vol. 60, no. 2, **(2010)**, pp. 319-352.

Author

Xuejing Niu, she is a college student of Anhui University. She participated in Research Training Foundation for Student of Anhui University in 2014. Her research interest is information hiding.

Meng Ma, he is associate professor of School of Computer Science, Anhui University. He obtained a Ph.D in computer science at University of Science and Technology of China in 2008. His research focuses on development of genome scale data mining algorithm for biology discovery, and pathogenetic mechanism study of disease variants.

Rui Tang, he is a university student majoring in Computer Science and technology in Anhui University. He participated in Research Training Foundation for Student of Anhui University 2014. His research interests include steganography, AI, and cryptology.

Zhaoxia Yin, she received her PhD degree from Anhui University in 2014. Now she is ACM member, academic committee member of CCF YOCSEF Hefei, and she works in Anhui University as a lecturer. Her research interests include digital image processing, information hiding, and information security.