Journal of
APPLIED COMPUTER SCIENCE & TECNOLOGY
(IEB)

A New Histogram-Shifting-Imitated Embedding Scheme

[1]A. H. M. Kamal, [2]Mohammad Mahfuzul Islam
[1]Deparment of Computer Sceince & Engineering, Jatiya Kabi Kazi Nazrul Islam University
[2]Deparment of Computer Sceince & Engineering, Bangladesh University of Engineering & Techonology
E-mail: ahmkctg@yahoo.com

Abstract
Histogram-shifting-imitated embedding scheme first computes a histogram of pixel values of an image. The scheme next divides the histogram into equal spaced n parts. It measures the highest appeared value in each part. Thus, it finds n number of bin values in the histogram, i.e., a one in each part, which contain the highest number of population in their respective partition. The scheme implants data bits in these pixel values which belong to these n most populated bins. To enhance the number of embeddable pixels, this article proposes to partition the image into blocks and to apply the stated histogram-shifting-imitated embedding process into each block separately. The experimental results reveal that the proposed scheme provides higher embedding payload, total implanted bits, than the competing scheme.

Keywords
Histogram shifting; imitate; data embedment; embedding payload; image steganography; image block.

I. Introduction

STEGANOGRAPHY is an art of hiding data in a media [1-2], e.g., image, audio, video, text. Among these media, the image is the most used one [3] because it contains enough redundant information. It is enough big to hide a secret in it. Its flexible size increases its scope to be communicated over the Internet in a frequent manner. The image to where the secret information is implanted is called a cover image. This is also called a carrier image because it carries the implanted information to the destination. After bit implantation, the image is termed as a stego image. An encoder, i.e., an embedding algorithm, implants the secret bits in a cover image by modifying the pixel values. At the receiver end, a decoder extracts the information from the stego image. These decoders are of two types. The first category of the decoders extracts the hidden information only. These do not care about the reconstruction of the cover image [4-7]. These schemes are called irreversible schemes. On the other hand, the second category of decoders both extracts the secret information from the image and reconstructs the cover image from the stego image [8-11]. These schemes are termed as reversible processes. The reversible schemes are more famous than irreversible ones because these improves the security of the implanted data. Additionally, many schemes use the cover image along with the extracted data at their receiver end for their further processing purpose.

Though there any many types of reversible schemes in the literature, one of the prominent one is to embed by histogram shifting and imitation. The histogram-shifting-imitated process [12] first computes a histogram of image pixels. The histogram computation method distributes the pixel values in their own value labeled bins. As the pixel's intensity varies from 0 to 255 in a gray color image, the histogram contains 256 bins, at most. It divides the histogram into p equal parts, e.g., 0-63 bins, 64-127 bins, 128-191 bins and 192-255 bins for p=4. In each of the p parts of the histogram, the method calculates the most populated bins. Thus, the method finds p populated bins in a histogram from its p parts. The embedding method implants secret message bits in the pixel values of these populated bins.

The benchmark scheme [12] computes the histogram for the entire image pixels and then applies the histogram shifting and imitation based embedding method for implanting secrets. It is investigated that if the scheme divides the image into a small sized block and then apply the imitation based embedding method, the scheme will enhance the volume of implanted bits. That investigation motivates us to propose a block partition based histogram shifting and imitation method to implant secrets. The proposed method demonstrates its superiority over the benchmark scheme empirically.

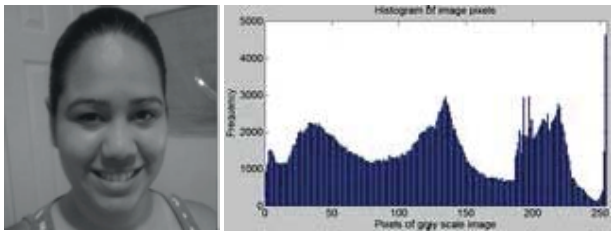
The article presents its remaining contents in the following four sections. Section II describes the histogram imitation based benchmark scheme [12]. Section III narrates the proposed method. The article

demonstrates its results and discusses them in section IV. Section V concludes the article.

II. Histogram Shifting And Immitation Based Benchmark Scheme

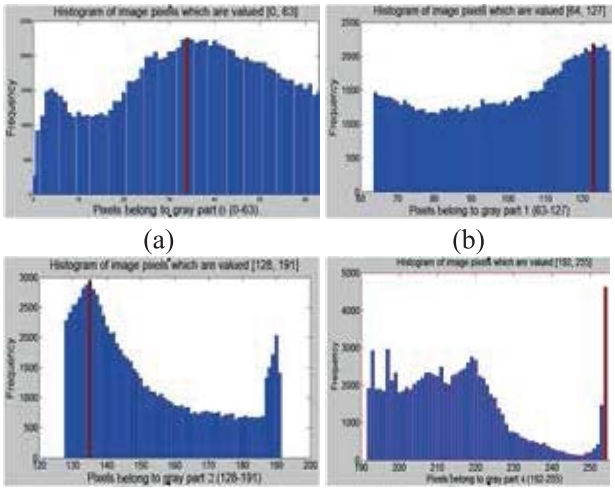
Histogram-Shifting-imitated embedding method (HSIEM) first draw a histogram of the image pixels, as shown in Figure 1 (b). The histogram of Figure 1(b) is drawn for the entire image pixels of Figure 1(a). The HSIEM divides the histogram of Figure 1(b) into p equal parts. If the length of a part is l, the value of p is 256/l. Figure 2 demonstrates the p parts of the histogram of Figure 1(b) for p=4.

The scheme marks the peak presenting bins in each part. The red marked line denotes the peaked bins. The scheme embeds message bits in the peak valued pixels and it remains the other pixels as unchanged. The scheme allows k bits to be implanted in each embeddable pixel, where.



(a) (b)

Figure 1: Histogram of an image: (a) a sample image, (b) its pixel histogram



(a) (b)

(c) (d)

Figure 2: Peaks in the histogram: (a) first part, (b) second part, (c) third part and (d) fourth part of the histogram of Figure 1(d).

A. Data Embedding Process

For a better explanation of the embedding method, consider a scenario where the pixel values range from 0 to 31 in an image C. Let the scheme divides the range value by four parts, i.e., 0-7, 8-15, 16-23 and 24-31. In this scenario, p is 4, l is 8 and k is 3. Consider the following image grid of Figure 3 (a). Let us explain the method of embedding secrets in the histogram part 16-23. Therefore, each concern pixel value is i, where . Following steps complete the embedding task:

Step 1: The pixels are shown in the grid with a light gray color.

Step 2: Among the pixels, the 22 appears for the highest number of times. The pixels, other than the 22s, are marked with a light gray color in Figure 3 (b).

Step 3: A "k-bit data to pixel value" (kBD2PV) map is generated, as shown in Figure 3(c). The kBD2PV map is a negotiated one between two communicating parties.

Step 4: Let the message stream M is 010110101111001000011101010…. The embedding scheme divides the message stream into each chunk of k bits. Here, k is 3. Hence, the chunked stream is M=010-110-101-111-001-000-011-101-010-….

Step 5: It does not implant any bit in the first 22. Rather, it pushes a 1 in the corresponding position of the location map and allow the 22 to remain as unchnaged.

Step 6: The scheme picks 3-bits of chunk to embed in the next 22. The chunk value is 010. It consults with the kBD2PV map and generates the stego value 18 for the cover value 22. It pushes a 1 in the corresponding position of the location map.

Step 7: The scheme picks next 3-bits of chunk to embed in the next 22. Let the chunk value is abc, where . It consults with the kBD2PV map and generates the stego value s for the message chunk abc. It replaces the 22 with the new value s. It pushes a 1 in the corresponding position of the location map.

Step 8: If all the chunks are not implanted, the scheme sends the control pointer to step 7.

Figure 3 (d) and (e) demonstrates the modified pixels and the corresponding location map. The sender side sends the stego image S, e.g., the stego grid of Figure

3(d), to a destination end. The sender side also shares the kBD2PV map and the location map L to the receiver end through another communication channel. The receiver end reconstructs the cover image, e.g., the grid of Figure 3(a), and extracts the implanted message chunks.

B. Data Extraction Process

The decoder in the receiver end first receive the kBD2PV map and the location map L. The following steps next extracts the secrets and reconstructs the cover image:

Step 1: It finds the first 1 in L. The scheme marks the corresponding value c in the stego image S, i.e., c=22 in the stated scenario.

Step 2: It finds the next 1 in L in a location (i,j).

Step 3: It picks the corresponding value s from the (i,j) location of S.

Step 4: The decoder replaces the value s with the value c in the (i,j) location of S.

Step 5: The decoder next finds the value s in the kBD2PV map. It then copies the corresponding k-bits from the map. Here, it is abc as it is of 3-bits in the stated example. That abc is the extracted message chunk from that stego pixels. Regarding the stated scenario, the first 3-bit chunk is 010, as the corresponding stego value is 18. That value is retrived from kBD2PV.

Step 6: If the decoder finds further a 1 in L in a new location (i,j), it sends the control pointer to step 3.

Step 7: Finally, it copies the S to C.

Thus, the scheme extracts the secrets M and reconstructs the cover image C.

III. Proposed Modified Histogram Immitation Based Embedment Scheme

The histogram-shifting-imitated data embedment process is explained in section II. The scheme computes a histogram for the entire image pixels. The scheme next partitions the histogram into p parts and implants bits in each part individually. This article performs the same embedding algorithm in a different way. It first divides the image C into blocks. The scheme works in each block separately. In each block, it applies the same embedding process of [12]. The working procedure is explained in the following:

22	22	20	22	23	21	23	21
14	18	22	24	24	22	21	22
17	14	19	20	22	21	21	22
16	16	20	22	21	22	22	22
20	19	20	19	22	23	22	24
22	24	22	24	25	23	22	23
20	22	24	25	25	24	24	24
21	22	23	22	25	24	22	24

(a)

22	22	20	22	23	21	23	21
14	18	22	24	24	22	21	22
17	14	19	20	22	21	21	22
16	16	20	22	21	22	22	22
20	19	20	19	22	23	22	24
22	24	22	24	25	23	22	23
20	22	24	25	25	24	24	24
21	22	23	22	25	24	22	24

(b)

Pixel value	16	17	18	19	20	21	22	23
3-bit Data	101	001	010	100	011	111	110	000

(c)

22	18	20	22	23	21	23	21
14	18	16	24	24	21	21	17
17	14	19	20	23	21	21	20
16	16	20	16	21	18	20	19
20	19	20	19	16	23	17	24
22	24	23	24	25	23	21	23
20	23	24	25	25	24	24	24
21	18	23	16	25	24	17	24

(d)

1	1	0	1	0	0	0	0
0	0	1	0	0	1	0	1
0	0	0	0	1	0	0	1
0	0	0	1	0	1	1	1
0	0	0	0	1	0	1	0
1	0	1	0	0	0	1	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	1	0

(e)

Figure 3: Data implantation process:

(a) a sample image grid,
(b) highlighted embeddable pixels,
(c) k-bit data to pixel value map,
(d) stego grid and (e) location map.

Step 1: Divides the image C into blocks, where, $i \in [1,(h/d)*(w/d)]$ h and w are the height and width of C and $d \times d$ is the size of a block.

Step 2: Set i=1.

Step 3: Set $D=B$, where the size of D is $d \times d$.

Step 4: Compute a histogram of pixel values of D in a scale of 0 to 255.

Step 5: Divide the histogram into p parts and set j=1.

Step 6: Separate k-bits of message chunk sequentially from the message stream M where $k = \log_2^l$.
Step 7: Compute the sum of bin values of j-th part. Let the sum is f.

Step 8: If f>0 then
Apply HSIEM using kBD2PV map
Implant k-bits of message chunk.

End if $\tilde{B}_i = \tilde{D}$
Let the modified block is $\tilde{D}$.
Now set

Step 9: Set $j=j$+1.

Step 10: If $j<p$ go to step 6.

Step 11: Set $i=i$+1.

Step 12: If $i< (h/d)*(w/d)$ go to step 3.

After executing the whole steps, the proposed method generates a stego image $\tilde{B}$. The sender side sends that, the kBD2PV map and the location map to the destination. The receiver at the destination applies the data extraction and the cover reconstruction process as it is stated in section II. The only difference is that it works in each block separately, rather than working for the entire image.

IV. Result Analysis

The proposed and the competing schemes are experimented in MATLAB. The schemes are verified in the BOSS image dataset [13]. Table 1 tabulates the payload values in first 10 images for the block of size. The term payload stands for number of implanted bits. The table shows a positive gain in all the images and the gain values range from 4% to 19%. The results ensure that the proposed scheme embeds more bits in an image than the competing scheme. Table 2 shows the values of payload for the image block of size . The table demonstrates a similar result as of Table 1. In Table 2, the gains vary from 1.2% to 10.7%. These gain values are also positive figures. It is also noticeable from these tables that the embedding payloads decrease for larger image block. However, it is concluded that the proposed block-based scheme provides higher embedding payload in all the cases than the competing scheme.

Table 3 shows the computed PSNR [14] gains in the experimented images. All the PSNR gains are negative valued. It means that the proposed scheme degrades the image quality than the competting scheme. This is reasoning because the proposed scheme embeds secrets in more number of pixels than the competing scheme. As the proposed scheme embeds in more number of pixels, it modifies the values of more pixels. Nevertheless, the degradation amounts are negligible, however, the payload gains are noticeable. Therefore, the proposed scheme will contribute to the literature remarkably.

V. Conclusion

The proposed scheme enhances the embedding payload notably. It will attrack the data hiding applications to use for their data implantation. In our future works, we intend to apply the proposed scheme in the prediction error space. In the prediction error space, zero-valued errors become remarkably higher. Therefore, the proposed scheme will find more pixels to implant chunks of secret message. In that case, the scheme will use prediction error histogram to find the highest appeared errors. Thereafter, the highest appeared error generating pixels will be brought under the data implantation process. That scheme will then enhance the embedding capacity further.

Table 1: Payload gains for the block of size 8×8

Images	Wang *et al.*'s Scheme	Proposed Scheme	Gain
Image 1	196112	207930	6.026148
Image 2	181962	206966	13.74133
Image 3	142689	169285	18.63914
Image 4	140507	167520	19.22538
Image 5	191370	225876	18.03104
Image 6	176941	192047	8.537309
Image 7	213422	222133	4.081585
Image 8	210653	221932	5.354303
Image 9	166610	192126	15.31481
Image 10	197760	208768	5.566343

Table 2: Payload gains for the block of size 16×16

Images	Wang *et al.*'s Scheme	Proposed Scheme	Gain
Image 1	196112	201300	2.6454
Image 2	181962	201468	10.71982
Image 3	142689	154531	8.299168
Image 4	140507	152458	8.505626
Image 5	191370	194253	1.506505
Image 6	176941	189824	7.280958
Image 7	213422	217796	4.730259
Image 8	210653	213169	1.19438
Image 9	166610	202567	4.52859
Image 10	197760	205880	4.105987

Table 3: PSNR gains for the block of size 16×16

	Images									
	1	2	3	4	5	6	7	8	9	10
Gains	-1.6	-1.7	-1.8	-1.5	-1.6	-1.8	-1.4	-2.9	-1.3	-1.6

References

[1] Kamal, A. H. M., and Mohammad M. Islam. "Enhancing embedding capacity and stego image quality by employing multi predictors." Journal of Information Security and Applications 32 (2017): 59-74

[2] Tai, Wei-Liang, Chia-Ming Yeh, and Chin-Chen Chang, "Reversible data hiding based on histogram modification of pixel differences.", Circuits and Systems for Video Technology, IEEE Transactions on, 19.6 (2009): 906-910

[3] Liao, Xin, and Changwen Shu., "Reversible Data Hiding in Encrypted Images Based on Absolute Mean Difference of Multiple Neighboring Pixels.", Journal of Visual Communication and Image Representation, 28 (2015): 21-27

[4] Kamal A. H. M. and Islam M. M., "Enhancing the performance of the data embedment process through encoding errors", Journal of Electronics, 5.4 (2016): 79-95

[5] Hong, Wien, and Tung-Shou Chen, "A novel data embedding method using adaptive pixel pair matching.", Information Forensics and Security, IEEE Transactions on, 7.1 (2012): 176-184

[6] Lin, Ching-Chiuan, "An information hiding scheme with minimal image distortion.", Computer Standards & Interfaces, 33.5 (2011): 477-484

[7] Kamal, A. H. M., and M. Mahfuzul Islam, "Facilitating and securing offline e-medicine service through image steganography.", Healthcare Technology Letters, 1.2 (2014): 74-79

[8] Kamal, A H M and Islam, M. M.,. Boosting up the data hiding rate multi cycle embedment process, J. Vis. Commun. Image R., 40(2016): 574-588

[9] Wang J., Ni J. and Hu Y., "An efficient reversible data hiding scheme using prediction and optimal side information selection", Journal of Visual Communication and Image Representation, 25.6 (2014): 1425-1431

[10] Ma, Xiaoxiao, et al., "High-fidelity reversible data hiding scheme based on multi-predictor sorting and selecting mechanism.", Journal of Visual Communication and Image Representation, 28 (2015): 71-82

[11] Habiba S., Kamal A. H. M. and Islam M. M., "Enhancing the robustness of visual degradation based HAM reversible data hiding", Journal of Computer Science, 12.2 (2016): 88-97

[12] Wang, Zhi-Hui, Chin-Feng Lee, and Ching-Yun Chang, "Histogram-shifting-imitated reversible data hiding.", Journal of Systems and Software, vol. 86, no. 2, pp. 315-323, 2013

[13] BOSS: Bank of standardized stimuli, https://drive.google.com/drive/folders/0B3m1Sf0USgt8b1VET3NLcTVIc0U, last visited: 16 January 2019

[14] d Z. Pan et al., Reversible Data Hiding Based on Local Histogram Shifting with Multilayer Embedding, J. Vis. Commun. Image R., 31(2015): 64-74