

Reversible Data Hiding using Block-wise Histogram Shifting and Run-length Encoding

Kandala Sree Rama Murthy¹, V. M. Manikandan²

Department of Computer Science and Engineering
SRM University-AP, Andhra Pradesh, India

Abstract—Histogram shifting-based Reversible Data Hiding (RDH) is a well-explored information security domain for secure message transmission. In this paper, we propose a novel RDH scheme that considers the block-wise histograms of the image. Most of the existing histogram shifting techniques will have additional overhead information to recover the overflow and/or the underflow pixels. In the new scheme, the meta-data that is required for a block is embedded within the same block in such a way that the receiver can perform image recovery and data extraction. As per the proposed data hiding process, all the blocks need not be used for data hiding, so we have used marker information to distinguish between the blocks which are used to hide data and the blocks which are not used for data hiding. Since the marker information needs to be embedded within the image, we have compressed the marker information using run-length encoding. The run-length encoded sequence is represented by an Elias gamma encoding procedure. The compression on the marker information ensures a better Embedding Rate (ER) for the proposed scheme. The proposed RDH scheme will be useful for secure message transmission also where we are also concerned about the restoration of the cover image. The proposed scheme's experimental analysis is conducted on the USC-SIPI image dataset maintained by the University of Southern California, and the results show that the proposed scheme performs better than the existing schemes.

Keywords—Histogram shifting; run-length encoding; secure message transmission; overflow; Elias gamma

I. INTRODUCTION

Data hiding is a well-explored information security area in which a hidden message can be safely transmitted by embedding it in a digital media mask [1, 2]. Digital images are often used as a cover medium for data hiding purposes. The conventional data hiding scheme makes permanent changes to the cover image pixel values during data hiding process and at the receiver side, only the hidden message will be extracted and not concerned about the cover image. The image obtained after data hiding is termed as stego image. The sender embeds the secret message into the cover image and communicates the stego image and a key for extraction to the sender. If the receiver can extract the hidden message and recover the original image completely then the scheme is called Reversible Data Hiding (RDH) [3]. For the past two decades, RDH schemes are widely studied and a number of algorithms are proposed in this domain. The overview of a RDH scheme is graphically shown in Fig. 1.

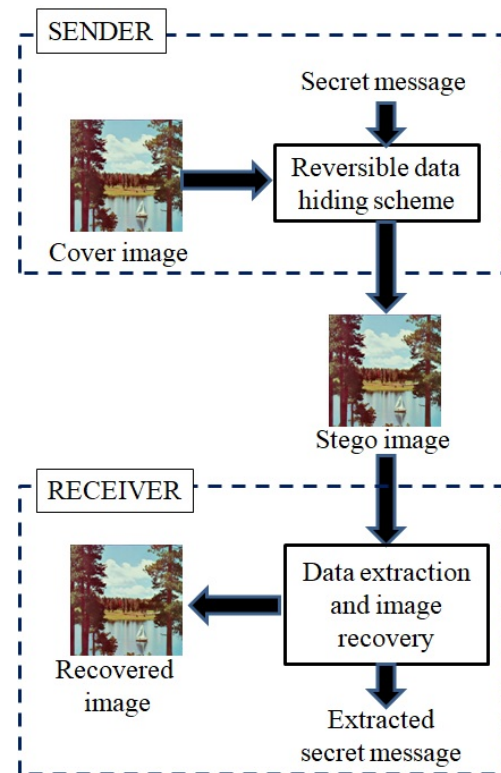


Fig. 1. Overview of RDH.

II. LITERATURE REVIEW

Most of the existing RDH schemes are centered around the idea of compressing a bit plane of image to create extra space for the secret message [3-5]. Difference expansion techniques [6-8] and histogram shifting based approach is discussed in [9-13]. There are a number of algorithms that use entirely different approaches for data hiding are also available in the literature [14-16].

In medical image transmission, patient records can be securely sent along with medical images using RDH methods which enable us to recover both the image and message without any loss of data. The reversible watermarking schemes are widely used for authenticating medical images in which a watermark will be embedded in the medical image instead of a secret message. The cloud service providers commonly use RDH schemes to hide some additional metadata on the images uploaded by the clients.

In this research, we explored a histogram shifting based RDH scheme and introduced a block-wise histogram shifting based RDH scheme with a better embedding rate by compressing the marker information using run-length encoding. The marker information is additional information that helps to distinguish the blocks which are used for data hiding and the blocks which are not used for data hiding.

For a better understanding of the proposed scheme, the basic histogram shifting based RDH scheme is detailed in Section III. The proposed scheme is detailed in Section IV. The experimental study and result analysis is detailed in Section V. A few other alternatives that we tried along with the proposed scheme are briefed in Section VI. The conclusion and a few insights to future work are given in Section VII.

III. PRELIMINARIES

In this section, we briefly discuss the basic histogram shifting based RDH scheme [9] since we extended the same in this manuscript. An overview of the block-wise histogram shifting based RDH scheme introduced in [13] is also discussed in this section for better understanding of the proposed scheme.

Algorithm I: Basic histogram shifting based RDH algorithm

Input	: Cover image I and the secret message D
Output	: Stego image S
Step 1	: Find the histogram H of the cover image I
Step 2	: Identify the peak pixel value P from the histogram H
Step 3	: Increment all the pixels which are greater than P by one to get the new image S . No need to do any changes on the pixels having intensity value P or less than P .
Step 4	: Read the pixels in S in a predefined order (either row-wise linear order or in a pseudo-random order based on a data hiding key) and if we find a pixel at location (X, Y) with the pixel value P in which we can hide one bit Q from the secret message D . Add the Q value with the pixel at location (X, Y) to perform data hiding. Note that the new pixel value will be $(P+1)$ if we hide the secret bit 1 and the pixel will be unchanged if we embed secret bit 0. Do this process to embed all the bits from secret message to get the final stego image S .
Step 5	: Return the stego image S

The data extraction and image recovery process in histogram shifting based RDH process is given in Algorithm II.

The major challenges with the basic histogram shifting based RDH scheme are listed below:

- Embedding Rate (ER) (the number of bits that can be embedded per pixel) is very low. For more information please refer to section IV. a.
- Overflow is quite common when the original cover image contains some pixel value 255.

Algorithm II: Data extraction and image recovery process

Input	: Stego image S , the pixel value P used for data hiding
Output	: Recovered image I and the extracted message D
Step 1	: Read one pixel K from S in a predefined order. Do the following to extract the hidden message D and recover the original image I : Case 1: $K < P$ Just keep K as it is in the recovered image I . Case 2: $K == P$ Extract a bit 0 and append it with the secret message D and no need to do any change for the pixel value keep as it is in the recovered image I . Case 3: $K == P+1$ Extract a bit 1 and append it with the secret message D and to get the recovered pixel value, decrement the pixel value by one and keep it in recovered image I . Case 4: $K > P+1$ Decrement the pixel value by one and keep it in I to recover the original pixel value and do not extract any secret message bit.
Step 2	: Return the recovered image I and extracted secret message D

A scheme is introduced in [17] to handle the overflow in the histogram shifting based on RDH. A block-wise histogram shifting based scheme is introduced by us [13] in which the overflow handling technique is used. In addition to that, block-wise histogram shifting based approach is used to improve the ER. The main idea is that instead of generating the histogram of the whole image, we divided the image into fixed size blocks and applied the histogram shifting algorithm on each block. The embedding rate in the histogram shifting algorithm is proportional to the peak value of a pixel, and by dividing the image into blocks, the overall embedding rate is increased significantly without affecting the visual quality of the stego image. We also eliminated the need of sending a key separately to the receiver by embedding the key, which is the peak pixel value, within the first a few pixels of each block using the least significant bit substitution.

A major hurdle in histogram shifting is the problem of overflow bits. When we are shifting the pixels by one position, the end pixel value can't be shifted ahead. The solution we provided is to make a marker list of all the end pixel values before and after shifting and store this information within each block by embedding it along with the message. On the retrieval of the message, this marker information can be utilized to distinguish the overflow bits and deal with them separately. To store this marker information within a block along with a portion of the secret message, each block's peak value must be greater than the number of overflow bits plus the binary size of the key. This condition disqualifies a few blocks to be used for data embedding. The marker information is embedded in the first block of the image and thus the entire first block is omitted in the rest of the algorithm.

The scheme discussed in [13] divides the original image into blocks of size $B \times B$ and we need to find the histogram of each block. If the peak in the histogram of the selected block is capable to provide a sufficient embedding rate to hide the peak pixel value, overflow information, and at least one secret message bit then it will be treated as a block suited for data hiding. All the blocks will be verified at the beginning itself to generate marker information in which 0 corresponded to a block indicates that the block is not usable for data hiding and 1 indicates that the block can be used for data hiding.

Let us assume that the original image is of the size $R \times C$ pixels then the original image will have N non-overlapping blocks of size $B \times B$ where N is defined as follows:

$$N = \left\lfloor \frac{R}{B} \right\rfloor \times \left\lfloor \frac{C}{B} \right\rfloor \quad (1)$$

All remaining ($N-1$) blocks have been used for data hiding in the current scheme except for the first block, and the size of marker information M will be $N-1$. The marker bits are embedded in the first block of the cover image by replacing the $N-1$ least significant bits (LSB) from the first $N-1$ pixels in the first block of the cover image. It may be noted that at the receiver side, we must recover the original image as it is. For this purpose, the LSBs of the first $N-1$ pixels are embedded in the image itself while hiding the secret message. If there is a possibility to reduce the number of bits in the marker information then it will help to improve the actual embedding rate. It should be noted that the actual embedding rate is defined based on the number of secret message bits that we can embed in the image without considering all the other overhead such as embedding marker information, the embedding of overflow information, etc.

The proposed scheme explored the possibility of reducing the number of bits required for marker information. During the existing scheme's experimental study, it is observed that in most of the images most of the blocks are used for data hiding. In such cases, the marker information will be a sequence of bits in which very rarely a bit 0 will be present. Note that if the J^{th} bit is 0 in the marker information, it indicates that the J^{th} block is not used to hide data.

In this manuscript, we propose an RDH scheme that compresses the marker information using run-length encoding. The run-length encoded sequence is encoded using a variable length encoding scheme called Elias gamma. The run-length encoding sequence is popularly used to compress the data when it contains redundant information. We observed that the marker information is always having a high amount of redundancy (1's are coming consequently as most of the blocks are used for data hiding) which motivated us to use run-length encoding to compress the marker information. The run-length sequence should be converted into a sequence of bits and we have used Elias gamma encoding process in the proposed scheme. The Elias gamma is a well-known variable length encoding scheme. For better understanding, the run-length encoding process is briefly described here with an example. Let us assume C is a bit sequence as follows:

C : 10000000000000011111110000000000000000

The corresponding run-length sequence L will be

L : 1, 1, 15, 7, 17

In run-length sequence we will keep the starting bit of the C as it is. In the above example the first bit is 1, so we kept it as it is in the run-length sequence. The second value 1 indicates that 1 is repeating 1 time in C . The third value 15 indicates that 0 is repeating 15 times in C . The next value 7 indicates that 1 repeats 7 times in C and so on.

The run-length encoded sequence L can be converted into sequence of bits E by using Elias gamma encoding scheme. The Elias gamma encoding scheme is given Algorithm III. The input will be run-length sequence and the output will be E .

Algorithm III: Elias gamma encoding on run-length sequence

Input	: Run-length sequence L which consists of N values
Output	: Elias gamma encoded sequence E
Step 1	: Take the starting bit value (first value from L) and keep it in E
Step 2	: $X=1$
Step 3	: While $X < N$
Step 4	: $D=L_X // X^{\text{th}}$ value from L
Step 5	: Convert D into binary, say B
Step 6	Find the number of bits K in B
Step 7	: Append $(K-1)$ 0's to E then append B to E .
Step 8	: $X=X+1$
Step 9	: EndWhile
Step 10	: Return the Elias gamma encoded sequence E

If we apply the run-length encoding procedure on the run-length sequence that we have obtained in the previous step, the corresponding binary sequence E will be the following:

E : 11000111100111000010001

The total number of compressed data bits is 23 and we achieved a compression ratio of $30/23=1.30$.

We have used run-length encoding with Elias gamma to compress the marker information. Run-length encoding is a lossless compression scheme that will help us to decompress the information without any loss.

In the next section, the proposed algorithms are given where we have used all the basic concepts we have discussed here.

IV. PROPOSED SCHEME

This segment addresses the proposed scheme. The sequence of operations using the run-length encoding scheme in the proposed block-wise histogram-shifting based RDH is defined:

Algorithm IV: Proposed histogram shifting based RDH using run-length encoding

- Input : The cover image I and the secret message D
- Output : The stego image S
- Step 1 : Divide the cover image into blocks (non-overlapping) of size $B \times B$ pixels.
- Step 2 : Access one block at a time and check whether or not the block is suitable for hiding data. To check this for the K^{th} block, compute the histogram of the block and find the peak intensity P_K value and its corresponding count C_K .
- If $C_K > F + 8$
- $M_K = 1$ // K^{th} marker bit
- Else
- $M_K = 0$
- Where $K = 1, 2, \dots, N-1$ and F is the number of 255's or 254's in the K^{th} block which indicates the overflow information required. 8 bits are needed to keep the LSB's of the first 8 pixels which will be used to store the peak intensity value.
- Step 3 : Apply the run-length encoding process on the marker information M to get the run-length run-length sequence L
- Step 4 : Apply Elias gamma encoding procedure on L to get compressed binary sequence E
- Step 5 : Find the length of E , say Z
- Step 6 : Access the first block and extract Z LSB's from the first Z pixels, say T is the list of Z LSB's.
- Step 7 : Replace the LSB's of first Z pixels by the bits in E and keep the modified block as the first block of the stego image S .
- Step 8 : The secret message D should be appended at the end of T
- Step 9 : Access the K^{th} block G_K for data hiding purpose in the predefined order. Find the peak pixel value from the block G_K by considering the histogram of the image block and store the 8-bit peak pixel binary value at the LSB position of the first 8 pixels of the block, if and only if the block is suitable for hiding the data. The LSBs of the first 8 pixels W , along with the overflow information, should be embedded in the same block itself. The overflow handling bits O are computed by traversing the whole block one by one and the 0 bit will be used to mark the pixels originally with 254 and 1's will be used to mark the pixels with overflow (originally 255).

The same process that is described in Algorithm I is used for data hiding but the only difference is that we will be applying the same thing on a block (except the first 8 pixels) and the bits that are going to hide consists of LSB information of the first 8 pixels, the overflow information and some bits from actual secret message. The altered GK blocks will be positioned at the corresponding position in I .

- Step 10 : Step 9 should be repeated for all the blocks in the image I to get final stego image S
- Step 11 : Return the stego image S .
-

The sender just needs to send the stego image S to the receiver for data retrieval and image recovery during the data hiding. The marker information about the order followed during data hiding is embedded in stego image. In our experimental study, we have followed row-wise linear order during data hiding phase. The image recovery and data extraction is possible from the stego image S . The process is described in Algorithm V.

Algorithm V: Image recovery and data extraction from the proposed scheme

- Input : The stego image S
- Output : The extracted message D and the recovered image I .
- Step 1 : Divide the stego image S into blocks (non-overlapping) of size $B \times B$ pixels.
- Step 2 : Find the number of non-overlapping blocks N in the image S .
- Step 3 : Keep on extracting the LSB's of the pixels from the first block and apply the Elias gamma decoding and run-length decoding procedure on the fly. Continue this process until we are getting a bit sequence M having size $(N-1)$. This will be the actual marker information which is used to identify the blocks used for data hiding. Let us assume that we have accessed V number of pixels to get a bit sequence M of $N-1$ bits.
- Step 4 : Access the K^{th} block G_K from stego image S and process it if M_K is 1. Otherwise just ignore it.
- Step 5 : If the G_K is used during data hiding then extract the first 8 bits from the first 8 pixels and convert into corresponding binary P_K . Apply data extraction and image recovery process (almost same as Algorithm II) from the block G_K (excluding first 8 pixels). The first 8 bits from the extracted message should be used to recover the first 8 pixels by replacing the LSBs. Next W number of bits O will be used to retrieve the pixels having value 255. The size of the W will be nothing but number of 255's in the current block G_K . The bit 0 in O says that the 255's should be decremented by one and the bit 1 in O says that the 255 should keep as it in the recovered image. The remaining bits extracted from this will be kept on appending with a bit sequence D' . Repeat this process for all the blocks in the image and store the values in I .
- Step 6 : Extract the first V number of pixels from D' and replace it in the LSB's of first V pixels to recover the first block and keep the block as the first block of I .
- Step 7 : Remove the V number of bits from D' to get D which will be the actual extracted message.
- Step 8 : Return extracted message D and recovered image I .
-

V. EXPERIMENTAL STUDY AND RESULT ANALYSIS

The experimental research and analysis of outcomes is performed on every image from the USC-SIPI image dataset

[18] managed by the Southern California University. Since the dataset consists of both color images and grayscale images of various sizes, to attain uniformity we have converted all images into 8-bit grayscale images of size 512×512 pixels. Our algorithm can be implemented on color images as well without any preprocessing. In all the images we have embedded the maximum possible number of bits as secret message. A pseudo-random bit sequence is generated as the secret message. The USC-SIPI image dataset has four different categories of images: aerials, textures, sequences and miscellaneous (misc.). The optimum block size is empirically decided as 16×16 pixels and all the experimental study of the proposed scheme is performed by taking this block size. We have conducted an experimental with various block sizes such as 128×128, 64×64, 32×32, 16×16 and 8×8 and picked the block size which provides the maximum ER. We have analyzed the following efficiency parameters during the study:

- 1) Embedding rate
- 2) Peak signal to noise ratio (PSNR)
- 3) Structural similarity index (SSIM)
- 4) Natural image quality evaluator (NIQE)
- 5) Blind image spatial quality evaluator (BRIQUE)

A. Analysis of Embedding Rate

Embedding rate (ER) is defined as follows:

$$ER = \frac{K}{R \times C} \quad (2)$$

where K is the maximum number of bits in the message that can be embedded in a R×C size image. The embedding rate is usually denoted by bits per pixel (bpp). The embedding rate of the scheme proposed is purely dependent on the distribution of pixels in the image. In the proposed scheme we are utilizing some of the actual embedding possibilities to hide the pixel recovery information. So we basically considered the effective embedding rate by deducting all the overhead information. Table I provides the average embedding rate obtained from all four types of images from the proposed scheme.

If the image consists of too much smooth regions with less possibility of overflow then it will be capable to provide a high embedding rate. From Table I, it can be observed that the average ER from all four categories of images is better than the existing schemes. Some additional overhead bits are reduced in the proposed scheme through the use of run-length encoding process. The embedding rate from the four well-known images such as airplane, boat, baboon and peppers is given in Table II. It also shows the embedding rate from the existing schemes [9], [13]. Table II shows that for all the four well-known images, the embedding rate of the proposed scheme is better than the embedding rate of the existing schemes mentioned in [9] and [13].

B. Analysis of PSNR and SSIM

Two image quality measurement tests are the PSNR and SSIM values, which help to analyze the image's quality deterioration based on a reference image. The low quality

images maybe an indication of some hidden images and the attackers may try to do steganalysis on such images to extract the hidden messages. So the researchers are very much concerned about the stego image quality while proposing new RDH scheme. If the original image is exactly same as the stego image, then PSNR will be ∞ and SSIM will be 1. But this will not happen since we are supposed to do some alteration in the pixels to hide the secret message and such changes will lead to reduction in PSNR and SSIM measure. If the PSNR and SSIM measure are high it indicates that there is low quality degradation on the stego image due to data hiding process. Table III presents a comparison between the PSNR for well-known images from the existing schemes and the proposed scheme.

The average SSIM value for all four categories of images from the existing schemes and the proposed scheme is given in Table IV.

Table V contains the SSIM values from the existing schemes and the proposed scheme for well-known images.

TABLE I. COMPARISON OF AVERAGE ER FROM ALL CATEGORY OF IMAGES

Image category	Scheme in [9]	Scheme in [13]	Proposed scheme
Misc.	0.0388	0.1081	0.1089
Textures	0.0230	0.0521	0.0529
Sequences	0.0266	0.0765	0.0774
Aerials	0.0252	0.0588	0.0597

TABLE II. COMPARISON OF ER (BPP) FOR WELL-KNOWN IMAGES

Image category	Scheme in [9]	Scheme in [13]	Proposed scheme
Baboon	0.0105	0.0201	0.0210
Peppers	0.0105	0.0438	0.0447
Boat	0.0221	0.0442	0.0451
Airplane	0.0317	0.0903	0.0912

TABLE III. COMPARISON OF PSNR (IN DB)

Image name	Scheme in [9]	Scheme in [13]	Proposed scheme
Baboon	50.4139	50.9370	50.9405
Peppers	50.0807	51.0778	51.0815
Boat	51.9712	50.9220	50.9259
Airplane	53.9372	52.2065	52.2121

TABLE IV. COMPARISON OF AVERAGE SSIM FROM PROPOSED SCHEME AND EXISTING SCHEMES

Image category	Scheme in [9]	Scheme in [13]	Proposed scheme
Misc.	0.9993	0.9987	0.9987
Textures	0.9996	0.9994	0.9994
Sequences	0.9997	0.9989	0.9989
Aerials	0.9996	0.999	0.999

TABLE V. COMPARISON OF SSIM FROM PROPOSED SCHEME AND EXISTING SCHEMES FOR WELL-KNOWN IMAGES

Image category	Scheme in [9]	Scheme in [13]	Proposed scheme
Baboon	0.9998	0.9996	0.9996
Peppers	0.9998	0.9989	0.9989
Boat	0.9996	0.9991	0.9991
Airplane	0.9996	0.9987	0.9987

The results shown in Table III, Table IV and Table V indicate that from all the well-known images we are getting a very high PSNR and SSIM measure. In general, the data hiding schemes that can generate stego image with a PSNR of 50 dB or more is treated as a good scheme. The SSIM values are very close to 1 also. Two sample cover image and the corresponding stego image obtained after data hiding process is given in Fig. 2.

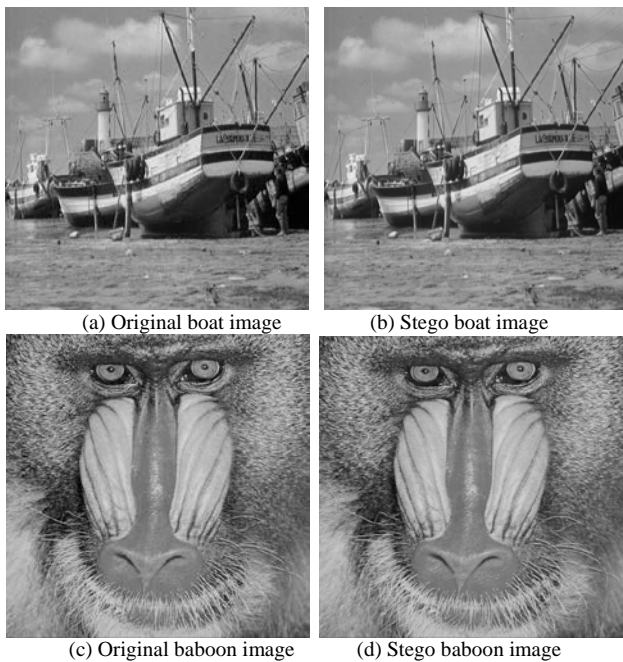


Fig. 2. Sample Results.

C. Analysis of NIQE and BRISQUE

The NIQE and BRISQUE are two well-known no-reference image quality measures [19, 20]. In addition to the reference based image quality assessment techniques such as PSNR and SSIM, we have analyzed the quality of the stego images using no-reference image quality assessment approaches such as NIQE and BRISQUE. In Fig. 3 and Fig. 4, the NIQE and BRISQUE measures of well-known images obtained from the proposed scheme and the existing schemes are given.

In Fig. 3, NIQE-1 indicates the NIQE measure from the original image and NIQE-2 is the NIQE measure from the stego image. A high NIQE measure means that the image quality is good.

The data shown in Fig. 3 and Fig. 4 indicates that the stego image quality is not much deviated from the quality of the original image.

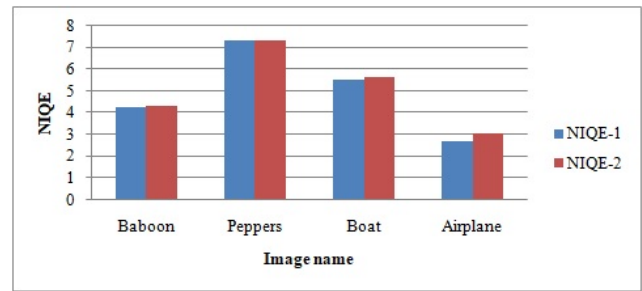


Fig. 3. NIQE Measure from Original Image and Stego Image.

The BRISQUE image quality measures of original image and stego image are given in Fig. 4. It may be noted that the low BRISQUE measure indicates high image quality. In Fig. 4, BRISQUE-1 is the BRISQUE measure from the original image and BRISQUE-2 is the stego image's BRISQUE measure.

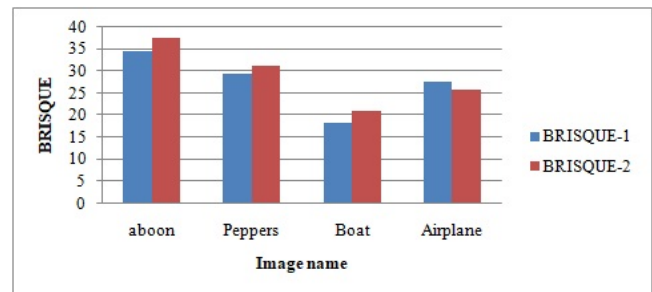


Fig. 4. BRISQUE Measure from Original Image and Stego Image.

VI. ALTERNATE APPROACHES FOR IMPROVEMENT

In addition to the Algorithms discussed in Section II, we have attempted a few other alternatives to improve the proposed scheme. The histogram of each block can also be left-shifted instead of the default right shift operation based on the following considerations:

1) The number of pixel values less than the peak pixel value is less than the number of pixels greater than the peak value. This demands us to change fewer pixel values and thus the stego image quality will be improved.

2) The number of underflow marker bits (0's and 1's) is less than the number of overflow marker bits (255's and 254's). This reduces the marker bits that need to be embedded along with the message and thus the embedding rate will increase.

In all the methods we have proposed so far, we have omitted the entire first block of the image for storing marker information. Since we have compressed the number of marker information that needs to be stored in the first block, we no more need nearly more than half of the first block for storing the marker bits. Hence we can also utilize the first block for data hiding which will marginally improve the embedding rate. We have implemented the above-mentioned techniques also but didn't incorporate them with the proposed algorithm as there were only marginal and mixed improvements in results.

VII. CONCLUSION

A block-wise histogram shifting based RDH scheme with a high embedding rate is proposed in this manuscript. In the proposed scheme, marker information is generated by checking the suitability of a block for data hiding and this information is compressed by using run-length encoding with Elias gamma encoding procedure. In order to facilitate the data extraction and image recovery process, the Elias gamma encoded marker information is embedded in the first block of the cover image. A better embedding rate without losing the visual quality of the stego image is attained in the experimental analysis of the proposed scheme. The visual quality of the stego images is measured using a metric of reference image quality, such as the peak signal to noise ratio and the structural similarity index. Most of the stego images gave a PSNR value greater than 50 dB and SSIM values close to 1. In addition, the visual accuracy of the stego image is measured using non-reference image quality tests such as natural image quality evaluator and blind image spatial quality evaluator, and the results show that the quality of the stego image does not differ significantly from the quality of the original image. The future works can be focused to apply histogram shifting on the histogram of regions generated through segmentation rather than on the non-overlapping blocks.

ACKNOWLEDGMENT

We kindly acknowledge the endless support received from the parent institute for successfully completing the project.

REFERENCES

- [1] Bender, Walter, et al. "Techniques for data hiding." IBM systems journal 35.3.4 (1996): 313-336.
- [2] Swanson, Mitchell D., Bin Zhu, and Ahmed H. Tewfik. "Robust data hiding for images." 1996 IEEE Digital Signal Processing Workshop Proceedings. IEEE, 1996.
- [3] Celik, Mehmet Utku, et al. "RDH." Proceedings. International Conference on Image Processing. Vol. 2. IEEE, 2002.
- [4] Abbasi, Rashid, et al. "Efficient lossless compression based RDH using multilayered n-bit localization." Security and Communication Networks 2019 (2019).
- [5] V. M. Manikandan, V. Masilamani, "A Novel Bit-plane Compression based RDH Scheme with Arnold Transform", International Journal of Engineering and Advanced Technology, vol. 9, issue 5, pp. 417-423, 2020.
- [6] Tian, Jun. "Reversible data embedding using a difference expansion." IEEE transactions on circuits and systems for video technology 13.8 (2003): 890-896.
- [7] Hu, Yongjian, et al. "Difference expansion based RDH using two embedding directions." IEEE Transactions on Multimedia 10.8 (2008): 1500-1512.
- [8] El-sayed, Hala S., S. F. El-Zoghdy, and Osama S. Faragallah. "Adaptive difference expansion-based RDH scheme for digital images." Arabian Journal for Science and Engineering 41.3 (2016): 1091-1107.
- [9] Ni, Zhicheng, et al. "RDH." IEEE Transactions on circuits and systems for video technology 16.3 (2006): 354-362.
- [10] Jia, Yujie, et al. "RDH based on reducing invalid shifting of pixels in histogram shifting." Signal Processing 163 (2019): 238-246.
- [11] Li, Yuanzhang, et al. "A high-imperceptibility and histogram-shifting data hiding scheme for JPEG images." IEEE Access 7 (2019): 73573-73582.
- [12] Manikandan, V. M., and V. Masilamani. "Histogram shifting-based blind watermarking scheme for copyright protection in 5G." Computers & Electrical Engineering 72 (2018): 614-630.
- [13] Murthy, Kandala Sree Rama, and V. M. Manikandan. "A Block-wise Histogram Shifting based RDH Scheme with Overflow Handling." 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2020.
- [14] Nguyen, Thai-Son, and Chin-Chen Chang. "A RDH scheme based on the Sudoku technique." Displays 39 (2015): 109-116.
- [15] Parah, Shabir A., et al. "Hiding clinical information in medical images: a new high capacity and RDH technique." Journal of biomedical informatics 66 (2017): 214-230.
- [16] Weng, Shaowei, et al. "Dynamic improved pixel value ordering RDH." Information Sciences 489 (2019): 136-154.
- [17] Manikandan, V. M., and P. Renjith. "An Efficient Overflow Handling Technique for Histogram Shifting based RDH." 2020 International Conference on Innovative Trends in Information Technology (ICITIIT). IEEE, 2020.
- [18] USC-SIPI image dataset: <http://sipi.usc.edu/database/>, accessed on 05-11-2020
- [19] Mittal, A., R. Soundararajan, and A. C. Bovik. "Making a Completely Blind Image Quality Analyzer." IEEE Signal Processing Letters. Vol. 22, Number 3, March 2013, pp. 209-212.
- [20] Mittal, Anish, Anush K. Moorthy, and Alan C. Bovik. "Blind/referenceless image spatial quality evaluator." 2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR). IEEE, 2011.