

# A Novel Bit-plane Compression based Reversible Data Hiding Scheme with Arnold Transform

V. M. Manikandan, V. Masilamani



**Abstract:** Reversible data hiding (RDH) is an active research area in the field of information security. The RDH scheme allows the transmission of a secret message by embedding it into a cover image, and the receiver can recover the original cover image along with the extraction of the secret message. In this paper, we propose a bit plane compression based RDH scheme to hide a sequence of secret message bits into a grayscale image. In the proposed method, a selected bit plane of the cover image will be compressed using run-length encoding (RLE) scheme. Further, the RLE sequence has been efficiently encoded as a binary sequence using Elias gamma encoding method. The Elias gamma encoded bit sequence concatenated with the secret message bits are used to replace the selected bit plane after performing a sequence of Arnold transform. The Arnold transform helps to find a new scrambled version of the bit plane which is very close to the original bit plane to ensure the visual quality of the stego image. The RLE is a lossless compression technique, therefore recovery of the original image is possible by the receiver. The experimental study of the proposed scheme on the images from standard image dataset (USC-SIPI image dataset) shows that the proposed scheme outperforms the existing scheme in terms of the visual quality of the stego image without compromising the data embedding rate.

**Keywords:** Reversible data hiding, Arnold transform, Bit-plane compression, Secret message communication, Run-length encoding

## I. INTRODUCTION

The image data hiding techniques provide a way to embed secret data into a cover image for purposes such as data authentication, copyright protection, user identification, secret communication, etc. [1, 2]. Most of the methods for image data hiding will modify the cover image permanently, and it cannot be restored exactly during the extraction of the hidden data. But a recently introduced technique called reversible data hiding (RDH) allows the extraction of hidden data along with the exact recovery of the cover image [3]. The reversible data hiding schemes have been more useful in sensitive applications such as military image communication, medical image transmission, etc. where the alterations on the original images are not tolerable. An overview of the RDH scheme is shown in Fig. 1. From Fig. 1, it can be seen that the sender can embed a secret message into a cover image. The cover image which is embedded with a secret message is termed as stego image. The receiver can recover the original cover image along with the extraction of the secret message. A data hiding key should be used by

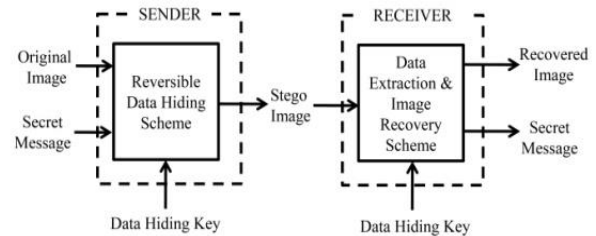


Fig. 1. Overview of reversible data hiding scheme

the sender and receiver to avoid the unauthorized decoding of the secret message and image recovery. Reversible data hiding schemes on images are roughly classified into three categories:

- **Lossless compression based RDH:** In this technique, the cover image has been compressed using a lossless compression based technique to make space for hiding secret messages [4].
- **Difference expansion based RDH:** In the difference expansion based technique, the difference between a pair of pixels has been expanded to hide a single secret message bit [5-9].
- **Histogram shifting based RDH:** In the histogram shifting based technique, the histogram of the cover image has been shifted to create a space to hide the secret message bits [10-13].

The existing compression based RDH schemes are not that much efficient to generate good quality stego images. The low quality of a stego image is a visual indication that the image contains some hidden information, and in such cases, malicious people may try to do image analysis operation to decode the hidden messages [14].

The difference expansion based RDH scheme selects a pair of pixels from the original image to embed a secret message bit. The quality of the stego image depends on the values of the selected pixel pairs, and if the selected pixels are very close then the quality of the stego image will be better. A high-capacity reversible watermarking scheme has been reported in [15], where the correlation between neighbourhood pixels has been used to determine the amount of difference expansion. Most of the schemes based on difference expansion are required to keep a location map to distinguish the pixels embedded with secret message bits from the other pixels because during difference expansion all the pixel values cannot be used for the embedding process due to overflow or underflow. The situations when the modified pixel values are going beyond the allowed pixel intensity range are known as overflow or underflow.

Revised Manuscript Received on May 25, 2020.

\* Correspondence Author

V. M. Manikandan\*, Assistant Professor, Department of Computer Science and Engineering, SRM University, Andhra Pradesh, India.

V. Masilamani, Associate Professor, Department of Computer Science and Engineering, Institute of Information Technology Design and Manufacturing Kancheepuram, Chennai, Tamil Nadu, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

During data extraction and image recovery phase, location map needs to be considered as a reference to distinguish the pixels with secret message bits and the unmodified pixels.

The histogram shifting based RDH schemes are also very popular. The key idea behind histogram shifting based RDH scheme is that increment all the pixels in the cover image with its pixel intensity value greater than  $P_k$  by one, where  $P_k$  is the pixel intensity value of the peak region in the histogram of the cover image. This process will create a gap in the histogram of the cover image just after the peak region. The embedding of the secret message bits can be carried out only on the pixels with intensity value  $P_k$ . To embed a watermark bit 1 in a pixel with intensity value  $P_k$ , modify it as  $P_k + 1$ ; to embed a watermark bit 0, keep the pixel value  $P_k$  as it is. For extraction of the secret message bits from the stego image, the receiver should know the peak intensity value  $P_k$ . A secret message bit 0 is to be extracted from a pixel with pixel intensity  $P_k$ , and a secret message bit 1 is to be extracted from the pixel with intensity value  $P_k + 1$ . All the pixel values greater  $P_k + 1$  should be decremented by one to recover the original cover image [16]. The histogram shifting process also may lead to overflow or underflow during the embedding process, and efficient methods to keep this additional information is required.

The conventional difference expansion and histogram shifting based RDH schemes require to make large modifications on the pixel values of the cover image, and it may lead to low-quality stego images. To improve the visual quality, and to achieve better embedding rate (by reducing the chances of overflow/underflow) prediction error based techniques have been introduced [17, 18]. The initial difference expansion and histogram shifting schemes considered the pixel intensity value as it is, but prediction error based techniques compute the prediction errors from actual pixel values based on some well-known prediction technique such as rhombus prediction [19]. Further, the pixel values of the cover image have been altered to modify the prediction errors to embed secret message bits.

Recently a few RDH schemes which are entirely different from the conventional methods have been reported. A sudoku based RDH scheme has been reported to achieve better embedding rate in [20]. But the quality of the stego image has been compromised here. A reversible watermarking scheme by scaling up the original image by a factor of 2 has been reported in [21], and it achieves an embedding rate of 0.75 bits per pixels (bpp). But here, for image transmission, the bandwidth requirement is 3 times more than the actual image size. Apart from these schemes, a few other schemes such as guided filtering based RDH [22] and RDH scheme with contrast enhancement have been reported in the literature [23].

In this manuscript, we propose a novel bit-plane compression based RDH scheme which provides better embedding rate without compromising the visual quality of the stego image. In the proposed scheme, a specific bit-plane is compressed using the run-length encoding (RLE) technique [24], and further, the run-length encoded sequence has been converted into a sequence of bits by using Elias gamma encoding scheme [25]. To achieve better visual quality for the stego image, the compressed data along with the secret message bits have been scrambled using Arnold transform, and an Arnold transform version which provides

minimum mean square error has been selected to replace the selected bit plane. All the other remaining bit planes will be kept as it is while merging of the bit planes to obtain the stego image.

The novelty of the proposed scheme is that to obtain good quality stego image even after data hiding process, a combination of run-length encoding, Elias gamma encoding, and Arnold transform are used for the first time in literature.

## II. PROPOSED SCHEME

The proposed bit-plane compression based RDH scheme is detailed in this section. The proposed scheme mainly consists of five major steps as given below:

1. Bit-plane slicing
2. Bit-plane compression
3. Data embedding process
4. Arnold transform on the modified bit-plane
5. Merging of bit-plane

### A. Proposed reversible data hiding process

Algorithm 1 describes the steps during reversible data hiding phase. An 8-bit grayscale cover image,  $I$ , having size of  $N \times N$  pixels and a secret message having size of  $L$  bits are taking as the input to the Algorithm 1. Initially, the cover image  $I$  will be divided into 8 different bit-planes, denoted as  $P_7, P_6, \dots, P_0$ . Note that each bit-plane will be a binary matrix of size  $N \times N$  bits. Basically the pixel intensity values at  $I[x, y] = (P_7 \cdot 2^7) + (P_6 \cdot 2^6) + (P_5 \cdot 2^5) + (P_4 \cdot 2^4) + (P_3 \cdot 2^3) + (P_2 \cdot 2^2) + (P_1 \cdot 2^1) + (P_0 \cdot 2^0)$ , where  $1 \leq x \leq N$  and  $1 \leq y \leq N$ . As per the proposed scheme, a selected bit-plane  $P_E$  will be compressed using the run-length coding scheme, further, the run-length sequence will be encoded using Elias gamma encoding scheme. The sequence of bits obtained as a result of Elias gamma encoding process is denoted by  $G$ . In the next step, we need to concatenate first  $(\lfloor \log_2(3 \cdot N) \rfloor + 1)$  number of bits extracted from bit plane  $P_0$ , Elias gamma encoded bit-sequence  $G$  and secret message bits  $S$ . This concatenated bit-sequence need to be converted into a 2-dimensional (2D) array of size  $N \times N$ , denoted by  $H$ . The sequence of Arnold transform on a 2D matrix of size  $N \times N$  integers will be periodic within  $3 \cdot N$  iterations, and to represent any value up to  $3 \cdot N$  requires maximum  $(\lfloor \log_2(3 \cdot N) \rfloor + 1)$  number of bits. Further, all possible versions of Arnold transform will be generated from  $H$ , and later, we will select an Arnold transform version which will give minimum mean square error (MSE) while comparing with the original bits in  $P_E$ .

For better understanding, the sequence of operations are defined mathematically. We need to find all the possible Arnold transform versions of  $H$ .

$$H_v = \{T^i(H) \mid 1 \leq i \leq q, \text{ where } T^q(H) = H\} \quad (1)$$

where  $T$  is Arnold transform function,  $T^q$  represents the matrix obtained after  $q$  Arnold transforms as mentioned in equation (2) and equation (3).

**Algorithm 1:** Proposed reversible data hiding scheme that provides image encryption

<b>Input:</b>	I, S /* I : Grayscale cover image of size N×N, S: Secret message bits of length L. */
<b>Output:</b>	W /* W : Stego image of size N×N pixels with the hidden message bit sequence S */.
Step 1:	Divide the image I into 8 different bit planes, say P <sub>7</sub> , P <sub>6</sub> , ..., P <sub>0</sub>
Step 2:	Select the bit plane P <sub>E</sub> and apply binary run-length encoding on it. Denote the run-length encoded sequence by R.
Step 3:	Apply Ellias gamma encoding method on the run-length sequence R, say the resultant binary sequence is G.
Step 4:	$n = (\lfloor \log_2(3.N) \rfloor + 1)$ /*n indicates the number of bits required to represent any possible Arnold transform iteration*/
Step 5:	Extract first n bits from the bit plane P <sub>0</sub> while scanning in row-wise manner, call the n-bit sequence as B.
Step 6:	Create a new bit sequence D by concatenating B, G, and S. Let us assume that the number of bits in D is N <sup>2</sup> .
Step 7:	Convert D into a 2D array of size N×N, say H.
Step 8:	Find an Arnold transformed version of H from all the possible Arnold transform matrices, say V, which will give the minimum mean square error between P <sub>E</sub> and V.
Step 9:	Find the number of further Arnold transform requires to recover H from V, say Y.
Step 10:	Replace the bit-plane P <sub>E</sub> with the scrambled version V.
Step 11:	Convert Y into its corresponding binary sequence in an n-bit representation, say Y <sub>b</sub> .
Step 12:	Replace the first n-bits of P <sub>0</sub> by using Y <sub>b</sub> while scanning the pixels in row-wise manner.
Step 13:	Initialize an integer (8-bit unsigned representation) 2D matrix of size N×N with 0's to keep the stego image.
Step 14:	Combine all the bit-planes P <sub>0</sub> , P <sub>1</sub> ,..., P <sub>7</sub> to obtain the final stego-image (in grayscale), denoted as W using $W[x,y] = (P_7.2^7) + (P_6.2^6) + (P_5.2^5) + (P_4.2^4) + (P_3.2^3) + (P_2.2^2) + (P_1.2^1) + (P_0.2^0)$ , where $1 \leq x \leq N$ and $1 \leq y \leq N$
Step 15:	Output the stego image W

Step 1	Divide the image W into 8 different bit planes, say P <sub>7</sub> , P <sub>6</sub> , ..., P <sub>0</sub>
Step 2	$n = (\lfloor \log_2(3.N) \rfloor + 1)$ /*n indicates the number of bits required to represent any possible Arnold transform iteration*/
Step 3	Extract the first n bits from P <sub>0</sub> while scanning in row-wise manner, say Y <sub>b</sub>
Step 4	Find the integer corresponds to Y <sub>b</sub> , say Y.
Step 5	Find the Y <sup>th</sup> Arnold transform version of P <sub>E</sub> , say H.
Step 6	By row-wise scanning, linearize H into a corresponding row vector denoted by H <sub>L</sub> .
Step 7	Copy first n bits from H <sub>L</sub> , and replace the first n bits of P <sub>E</sub> while considering bits in row-wise order.
Step 8	Perform Elias Gamma decoding procedure and run-length decoding process on (n+1) <sup>th</sup> bits onwards of H <sub>L</sub> , until we get a bit-sequence of size N <sup>2</sup> , say R <sub>L</sub> . Let us denote that the bits from (n+1) to Z have been used to obtain a bit-sequence of N <sup>2</sup> bits.
Step 9	Convert the row vector R <sub>L</sub> into a matrix of size N×N bits.
Step 10	Replace the bit-plane P <sub>E</sub> by using the bits of R.
Step 11	The bits from (Z+1) to N <sup>2</sup> in H <sub>L</sub> will be secret message bits and keep it in S.
Step 12	Initialize a 2D integer matrix (8-bit unsigned integer) I of size N×N with 0's to keep the recovered original image.
Step 13	Combine all the bit-planes P <sub>0</sub> , P <sub>1</sub> ,..., P <sub>7</sub> to obtain the final stego-image (in grayscale), denoted as I using $W[x,y] = (P_7.2^7) + (P_6.2^6) + (P_5.2^5) + (P_4.2^4) + (P_3.2^3) + (P_2.2^2) + (P_1.2^1) + (P_0.2^0)$ , where $1 \leq x \leq N$ and $1 \leq y \leq N$
Step 14	Return I, S

Algorithms 2 describes the sequence of steps in proposed data extraction and image recovery process.

**Algorithm 2:** Data extraction and image recovery

<b>Input:</b>	W, E /* W: 8-bit grayscale image of size N×N pixels which is embedded with a secret message as per the procedure described in Algorithm 1, E: The bit plane selected for embedding process */
<b>Output:</b>	I, S /* I: Recovered grayscale image of size N×N pixels, S: Extracted secret message*/

For better understanding, an example of Elias gamma encoding on a run-length sequence obtained on a binary matrix of size 8×8 bits shown in Fig. 2 is described here. Row-wise scanning on the above bit-plane will generate a run-length sequence as shown in Fig. 3. Further, the run-length sequence will be encoded using Elias gamma encoding scheme. Elias gamma code is a universal code for coding integers whose upper bound cannot be determined beforehand. In this scheme, to represent a N-bit integer, N-1 zeros will be placed before the binary representation of that integer. Finally, the Elias gamma encoded sequence of the run-length sequence will be obtained as shown in Fig. 4. It can be seen that to represent the sample 8×8 bit-plane considered here, 64 bits are required to represent it. But after obtaining the Elias gamma encoded version of the run-length sequence of the bit-plane, it can be represented using 56 bits.

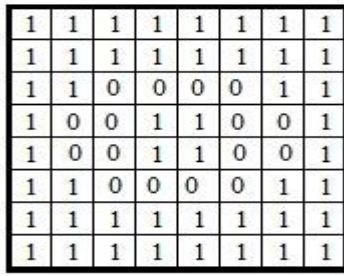


Fig. 2. Sample image

The generalized 2D Arnold transform is a mapping which transforms each coordinate (x,y) in a N×N grid to a new location (x',y') given by equation (2) and equation (3).

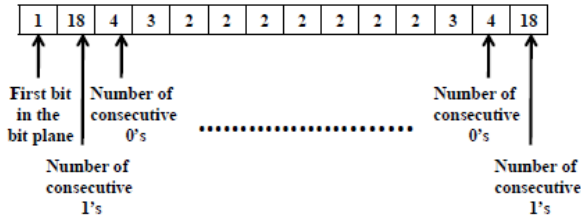


Fig. 3. Run-length code obtained from sample image

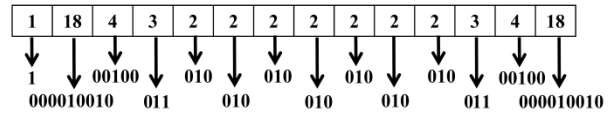


Fig. 4. Elias-gamma encoded sequence obtained from run-length code

$$x' = a_1x + a_2y + KN \pmod N \quad (2)$$

$$y' = a_3x + a_4y + KN \pmod N \quad (3)$$

Where (a<sub>1</sub>,a<sub>4</sub>)-(a<sub>2</sub>,a<sub>3</sub>)=±1, and K=maximum(|a<sub>1</sub>|, |a<sub>2</sub>|, |a<sub>3</sub>|, |a<sub>4</sub>|)

The number of Arnold transform iterations required to recover the original content will be kept in the first (⌊log<sub>2</sub>(3.N)⌋+1) bits of P<sub>0</sub>. The selected Arnold transform version, modified P<sub>0</sub>, and the unmodified other bit planes will be used to construct the stego image W. For better understanding, a schematic diagram of the proposed RDH scheme is given in Fig. 5.

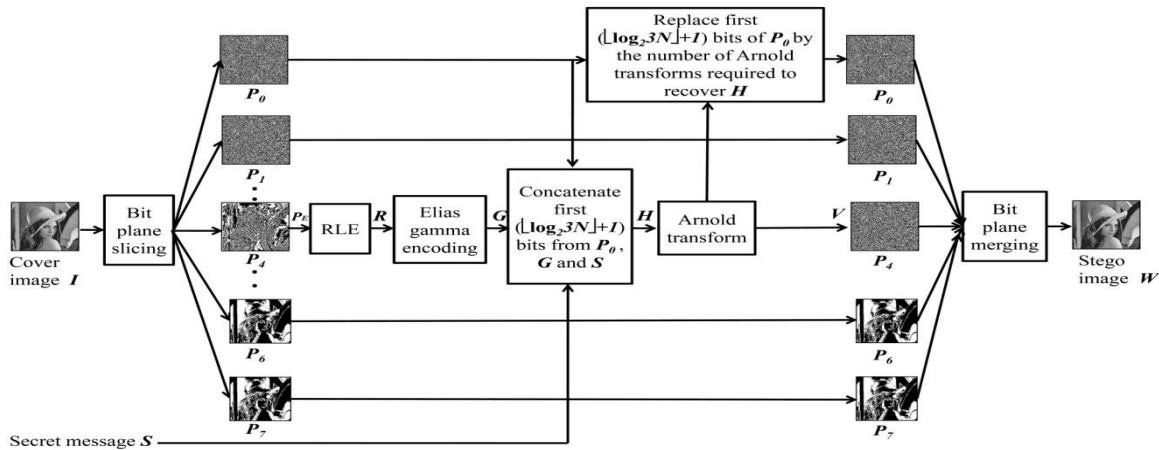


Fig. 5 Illustrative example of proposed reversible data hiding process

### B. Data Extraction and Image Recovery Process

The data extraction and image recovery process is given in Algorithm 2. In a reversible data hiding scheme, the users should be able to extract the hidden message along with the recovery of the original cover image from the stego image. Firstly, divide the given stego image \$W\$ into 8 bit planes P<sub>7</sub>, P<sub>6</sub>, ..., P<sub>0</sub>. Note that the pixel intensity values at W[x,y]=(P<sub>7</sub>.2<sup>7</sup>)+(P<sub>6</sub>.2<sup>6</sup>)+(P<sub>5</sub>.2<sup>5</sup>)+(P<sub>4</sub>.2<sup>4</sup>)+(P<sub>3</sub>.2<sup>3</sup>)+(P<sub>2</sub>.2<sup>2</sup>)+(P<sub>1</sub>.2<sup>1</sup>)+(P<sub>0</sub>.2<sup>0</sup>), where 1 ≤ x ≤ N and 1 ≤ y ≤ N. Further, extract the first (⌊log<sub>2</sub>(3.N)⌋+1) bits from the P<sub>0</sub>, which tells us the number of Arnold transform iterations required to restore the actual bits in P<sub>E</sub>. Let us assume that the first n=(⌊log<sub>2</sub>(3.N)⌋+1) bits correspond to the integer value Y. Apply Y number of Arnold transform on the P<sub>E</sub>, and denote the transformed matrix by H. Extract the first (⌊log<sub>2</sub>(3.N)⌋+1) bits from H and use these bits to replace the first (⌊log<sub>2</sub>(3.N)⌋+1) bits of P<sub>0</sub>. This process will lead to recovering the original bit plane P<sub>0</sub> of the cover image. Next, extract the remaining bits from H, and do the Elias gamma decoding procedure. This process should continue until we are able to generate a run-length sequence which can produce a binary sequence of N<sup>2</sup> bits. These N<sup>2</sup> bits will be converted

into a matrix of size N×N bits, and used as the bit plane P<sub>E</sub> of the recovered image. The remaining bits in the binary matrix H will be extracted as the secret message.

### III. RESULTS AND DISCUSSION

The experimental study of the proposed scheme has been carried out on the standard images downloaded from USC-SIPI image dataset [26]. This image dataset contains the well-known images such as baboon, peppers, boat, airplane, etc. During the experimental study, all the images are converted into 8-bit grayscale images of size 256×256 pixels, and a pseudo-random bit sequence has been used as the secret message. All the experiments have been carried out using a workstation with 32 GB RAM and Intel(R) Xeon(R) CPU of 3.46 GHz. All simulations have been done using Matlab2017a. In general, the efficiency of a reversible data hiding scheme is analyzed by the measures: peak signal to noise ratio (PSNR) [27], structural similarity index (SSIM) [28], embedding rate, and time complexity.

Experimental results obtained from the proposed scheme is discussed in this section in detail.

**A. Analysis of Visual Quality of the Stego Image**

The PSNR is an effective measure to find the visual quality degradation on a cover image due to the data hiding process. A higher PSNR value between the cover image G and the stego image F indicates the goodness of a data hiding scheme. The PSNR between the original cover image G and the stego image F can be computed by the equation (4) Let us assume that both images are 8-bit grayscale images of size M×N pixels. Similarly, the SSIM between two images basically compare the structural similarity between two images.

$$PSNR=10\log_{10}\left(\frac{255^2}{MSE}\right) \quad (4)$$

where MSE is defined as

$$MSE=\frac{1}{MN}\sum_{i=1}^M\sum_{j=1}^N(G[i,j]-F[i,j])^2 \quad (5)$$

While considering the proposed RDH scheme, the selection of bit plane highly influences the PSNR value. The data embedding process in higher bit planes will lead to more degradation in the stego image which in turn gives lower PSNR value. The embedding process on lower bit-planes may provide better PSNR value, but the embedding rate will be less as compared to the same from the higher bit-planes. Empirically, we decided that bit plane P<sub>4</sub> or P<sub>5</sub> can be considered for data hiding process which gives nominal embedding rate without compromising the visual quality.

The PSNR is not very well matched to the perceived visual quality, therefore SSIM has been introduced. SSIM value between two images I and W will be 1, if both are exactly identical. PSNR and SSIM values obtained for well-known images from the proposed RDH scheme while P<sub>4</sub> has been considered for data hiding process is shown in Table I.

**Table 1. PSNR and SSIM obtained from proposed scheme**

Image name	PSNR (in dB)	SSIM
Boat	33.33	0.83
Peppers	31.92	0.83
Airplane	33.11	0.85
Lake	32.33	0.86

The sample results obtained during experimental study is shown in Figure. 6. Figure. 6 (a) shows the original images, Figure. 6 (b) and Figure. 6 (c) are the corresponding stego images when data hiding has been carried out on plane 4 and plane 5 respectively.



**Fig. 6. Sample results**

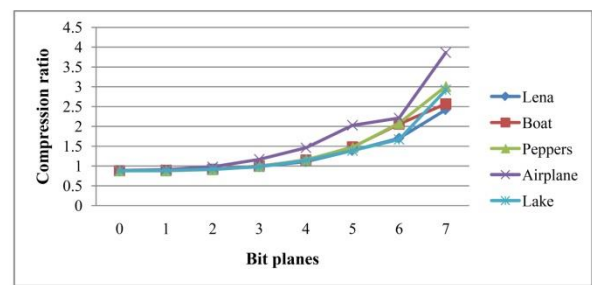
**B. Analysis of Embedding Rate**

The embedding rate of an RDH scheme is defined as the number of bits that can be embedded into a pixel. In general,

embedding rate is denoted as bits per pixel (bpp). In our proposed scheme, the embedding rate is highly dependent on the amount of compression that can be achieved for a selected bit plane. The amount of compression that can be achieved by a compression scheme can be measured by using compression ratio. The compression ratio (CR) can be computed by equation (6).

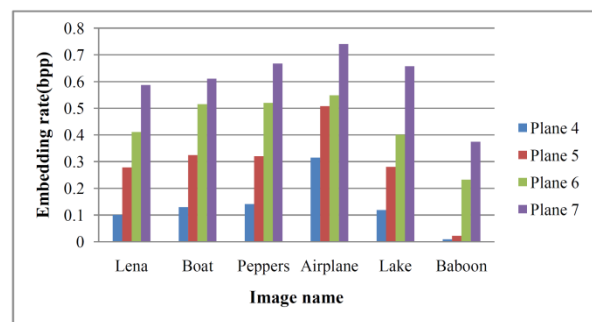
$$CR = \frac{\text{Uncompressed size}}{\text{Compressed size}} \quad (6)$$

The compression ratios obtained from different images while considering all the 8 bit planes P<sub>0</sub>, P<sub>1</sub>, ..., P<sub>7</sub> are shown in Fig. 7. From Fig. 7, it can be seen that compression ratio obtained for higher bit planes is more as compared to the same from the bit planes in lower levels. The compression ratio obtained less than 1 indicates that the number of bits required to represent a bit plane in the compressed domain is more as compared to its uncompressed representation.



**Fig. 7. Bit planes Vs Compression ratio**

As per the proposed RDH scheme, the embedding rate is directly proportional to the compression ratio. For most of the images the compression ratio obtained from P<sub>0</sub>, P<sub>1</sub>, P<sub>2</sub>, and P<sub>3</sub> are less than 1, and so these bit planes cannot be used for data hiding process. So we have considered bit planes P<sub>4</sub>, P<sub>5</sub>, P<sub>6</sub>, and P<sub>7</sub> for data hiding process and the embedding rate obtained for the well-known images are shown in Fig. 8. The property of the cover image and the position of the selected bit plane determines the embedding rate. From Fig. 8, it can be seen that the images like airplane and peppers give more embedding rate as they have fewer texture.



**Fig. 8. Embedding rate obtained for various images**

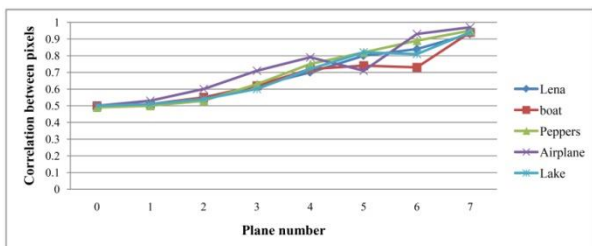
Run-length encoding scheme on a bit plane can give better compression when the bits within the bit plane are highly correlated. The correlation (C) of adjacent bits within a bit plane of size N×N bits can be computed by equation (7).

$$C = \frac{\sum_{x=1}^N \sum_{y=1}^{N-1} P(x, y) \cdot P(x, y + 1)}{\sqrt{\sum_{x=1}^N \sum_{y=1}^{N-1} P(x, y)^2} \sqrt{\sum_{x=1}^N \sum_{y=1}^{N-1} P(x, y + 1)^2}} \quad (7)$$

The correlation values obtained from all the 8 bit-planes for the well-known images such as lena, boat, peppers, airplane and lake are shown in Table 2. From Table 2, it can be seen that bit planes in higher levels are highly correlated as compared to lower bit planes. Due to this adjacency property, run-length encoding based compression used in the proposed RDH scheme is able to give better compression on higher bit planes as compared to lower bit planes. In general, if the bits in a bit plane is highly correlated, then run-length may give better compression ratio. Since the higher bit planes in airplane and peppers are highly correlated, the embedding rate will be more on this images. The graphical representation of Table 2 is given in Figure. 9.

**Table 2. Correlation between adjacent pixels**

Plane No.	Image name				
	Lena	Boat	Peppers	Airplane	Lake
0	0.50	0.50	0.49	0.50	0.50
1	0.50	0.51	0.50	0.53	0.51
2	0.54	0.55	0.53	0.71	0.54
3	0.62	0.62	0.63	0.79	0.60
4	0.70	0.72	0.75	0.71	0.72
5	0.80	0.74	0.82	0.93	0.82
6	0.84	0.73	0.89	0.97	0.81
7	0.93	0.94	0.95	0.97	0.94



**Fig. 9. Correlation between adjacent bits in bit planes of different images**

### C. Time Complexity Analysis

The major operations in the proposed RDH scheme are run-length encoding, Elias gamma encoding operation and Arnold transformations carried out on the modified bit plane. To obtain the run-length code sequence from any bit plane of an image with  $N \times N$  bits, only one pass over all the  $N^2$  bits in the plane is required. Further, for Elias gamma encoding process is required to consider the whole run-length sequence. The run-length sequence size is bounded by  $O(N^2)$ . While considering the next primitive operation, Arnold transforms, maximum  $3N$  Arnold transforms sequences can be generated from a bit plane of size  $N \times N$  bits. So the time complexity to find all possible Arnold transform version of a bit plane with  $N^2$  bits is  $O(N^3)$ . Hence, the theoretical time complexity of the proposed data hiding scheme is computed as  $O(N^3)$ .

Similarly during data extraction and image recovery phase, to recover the bits in the embedded bit plane, a sequence of Arnold transform is required. In the worst case, this will take  $O(N^3)$  time. Elias gamma decoding procedure and run-length decoding operations can be performed on

$O(N^2)$ . Finally, we can say that both data hiding algorithm and data extraction process require  $O(N^3)$  time.

Similar to theoretical time complexity, average execution time plays a major role to demonstrate the usefulness of RDH scheme in time restricted applications. The proposed RDH scheme took 6.35 seconds to finish the data hiding process, and it spends an average time of 1.97 seconds for data extraction and image recovery process (specification details of the workstation and the images used for the experimental study have been described in the starting of this section).

### D. Comparative Study

In this section, the proposed RDH scheme is compared with the recently proposed, well-known RDH scheme reported in [29]. The PSNR and embedding rate of the existing scheme [29] is dependent on the number of pairs of histogram peaks selected for data embedding. For comparison, we considered the number of pairs as 10, 15 and 20. In the proposed scheme the PSNR is highly dependent on the selection of bit plane for compression and data hiding. We have considered bit plane 4 and bit plane 5 for compression and data hiding process. The average PSNR and average embedding rate obtained from existing scheme [] and proposed scheme are shown in Table 3. The proposed scheme able to provide a PSNR value of 29.68 while bit plane at position 5 has been considered for data embedding process.

**Table 3. Comparison of PSNR (in dB) and Embedding Rate (bpp)**

RDH Schemes		PSNR (in dB)	Embedding rate (bpp)
Existing scheme [29]	10 pairs	30.34	0.270
	15 pairs	27.20	0.370
	20 pairs	25.03	0.458
Proposed Scheme	Plane 4	32.07	0.161
	Plane 5	29.68	0.342

The results obtained from the proposed scheme is better than or incomparable to the results obtained from existing scheme. We say results are incomparable when one result is better than the other on one parameter (PSNR) and weaker than the other parameter (embedding rate).

### IV. CONCLUSION

A new lossless compression based reversible watermarking scheme which provides better embedding rate without compromising the visual quality of the stego image is proposed in this paper. A selected bit-plane of the cover image has been compressed using run-length encoding technique along with Elias-gamma encoding scheme. The Arnold transformation technique has been used to scramble the compressed data and the secret message to improve the visual quality of the stego image.

The experimental study of the proposed scheme has been carried out on the standard images downloaded from USC-SIPI dataset shows that the proposed RDH scheme provides an embedding rate without compromising the visual quality of the stego image. The future works can be focused to design and develop robust reversible data hiding schemes.

## REFERENCES

1. BENDER, W., GRUHL, D., MORIMOTO, N., & LU, A. (1996). TECHNIQUES FOR DATA HIDING. IBM SYSTEMS JOURNAL, 35(3.4), 313-336.
2. SALOMON, D. (2003). DATA HIDING IN IMAGES. IN DATA PRIVACY AND SECURITY (PP. 269-337). SPRINGER, NEW YORK, NY.
3. AWRANGJEB, M. (2003, DECEMBER). AN OVERVIEW OF REVERSIBLE DATA HIDING. IN PROCEEDINGS OF THE SIXTH INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY (PP. 75-79).
4. CELIK, M. U., SHARMA, G., TEKALP, A. M., & SABER, E. (2005). LOSSLESS GENERALIZED-LSB DATA EMBEDDING. IEEE TRANSACTIONS ON IMAGE PROCESSING, 14(2), 253-266.
5. TIAN, J. (2003). REVERSIBLE DATA EMBEDDING USING A DIFFERENCE EXPANSION. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 13(8), 890-896.
6. ALATTAR, A. M. (2004). REVERSIBLE WATERMARK USING THE DIFFERENCE EXPANSION OF A GENERALIZED INTEGER TRANSFORM. IEEE TRANSACTIONS ON IMAGE PROCESSING, 13(8), 1147-1156.
7. LI, X., YANG, B., & ZENG, T. (2011). EFFICIENT REVERSIBLE WATERMARKING BASED ON ADAPTIVE PREDICTION-ERROR EXPANSION AND PIXEL SELECTION. IEEE TRANSACTIONS ON IMAGE PROCESSING, 20(12), 3524-3533.
8. DRAGOI, I. C., & COLTUC, D. (2014). LOCAL-PREDICTION-BASED DIFFERENCE EXPANSION REVERSIBLE WATERMARKING. IEEE TRANSACTIONS ON IMAGE PROCESSING, 23(4), 1779-1790.
9. NATARAJAN, V. (2016). HYBRID LOCAL PREDICTION ERROR-BASED DIFFERENCE EXPANSION REVERSIBLE WATERMARKING FOR MEDICAL IMAGES. COMPUTERS & ELECTRICAL ENGINEERING, 53, 333-345.
10. TAI, W. L., YEH, C. M., & CHANG, C. C. (2009). REVERSIBLE DATA HIDING BASED ON HISTOGRAM MODIFICATION OF PIXEL DIFFERENCES. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 19(6), 906-910.
11. TSAI, P., HU, Y. C., & YEH, H. L. (2009). REVERSIBLE IMAGE HIDING SCHEME USING PREDICTIVE CODING AND HISTOGRAM SHIFTING. SIGNAL PROCESSING, 89(6), 1129-1143.
12. ZHAO, Z., LUO, H., LU, Z. M., & PAN, J. S. (2011). REVERSIBLE DATA HIDING BASED ON MULTILEVEL HISTOGRAM MODIFICATION AND SEQUENTIAL RECOVERY. AEU-INTERNATIONAL JOURNAL OF ELECTRONICS AND COMMUNICATIONS, 65(10), 814-826.
13. CHEN, X., SUN, X., SUN, H., XIANG, L., & YANG, B. (2015). HISTOGRAM SHIFTING BASED REVERSIBLE DATA HIDING METHOD USING DIRECTED-PREDICTION SCHEME. MULTIMEDIA TOOLS AND APPLICATIONS, 74(15), 5747-5765.
14. BAO, L., & ZHOU, Y. (2015). IMAGE ENCRYPTION: GENERATING VISUALLY MEANINGFUL ENCRYPTED IMAGES. INFORMATION SCIENCES, 324, 197-207.
15. CHANG, C. C., & LU, T. C. (2006). A DIFFERENCE EXPANSION ORIENTED DATA HIDING SCHEME FOR RESTORING THE ORIGINAL HOST IMAGES. JOURNAL OF SYSTEMS AND SOFTWARE, 79(12), 1754-1766.
16. NI, Z., SHI, Y. Q., ANSARI, N., & SU, W. (2006). REVERSIBLE DATA HIDING. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 16(3), 354-362.
17. FU, D. S., JING, Z. J., ZHAO, S. G., & FAN, J. (2014). REVERSIBLE DATA HIDING BASED ON PREDICTION-ERROR HISTOGRAM SHIFTING AND EMD MECHANISM. AEU-INTERNATIONAL JOURNAL OF ELECTRONICS AND COMMUNICATIONS, 68(10), 933-943.
18. JAARA, E. N., & JAFAR, I. F. (2015, NOVEMBER). REVERSIBLE DATA HIDING BASED ON HISTOGRAM SHIFTING OF PREDICTION ERRORS USING TWO PREDICTORS. IN 2015 IEEE JORDAN CONFERENCE ON APPLIED ELECTRICAL ENGINEERING AND COMPUTING TECHNOLOGIES (AEECT) (PP. 1-5). IEEE.
19. COLTUC, D., & DRAGOI, I. C. (2013, JUNE). CONTEXT EMBEDDING FOR RASTER-SCAN RHOMBUS BASED REVERSIBLE WATERMARKING. IN PROCEEDINGS OF THE FIRST ACM WORKSHOP ON INFORMATION HIDING AND MULTIMEDIA SECURITY (PP. 215-220).
20. NGUYEN, T. S., & CHANG, C. C. (2015). A REVERSIBLE DATA HIDING SCHEME BASED ON THE SUDOKU TECHNIQUE. DISPLAYS, 39, 109-116.
21. PARAH, S. A., AHAD, F., SHEIKH, J. A., & BHAT, G. M. (2017). HIDING CLINICAL INFORMATION IN MEDICAL IMAGES: A NEW HIGH CAPACITY

- AND REVERSIBLE DATA HIDING TECHNIQUE. JOURNAL OF BIOMEDICAL INFORMATICS, 66, 214-230.
22. YAO, H., QIN, C., TANG, Z., & TIAN, Y. (2017). GUIDED FILTERING BASED COLOR IMAGE REVERSIBLE DATA HIDING. JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION, 43, 152-163.
23. GAO, G., WAN, X., YAO, S., CUI, Z., ZHOU, C., & SUN, X. (2017). REVERSIBLE DATA HIDING WITH CONTRAST ENHANCEMENT AND TAMPER LOCALIZATION FOR MEDICAL IMAGES. INFORMATION SCIENCES, 385, 250-265.
24. GONZALES, R. C., & WOODS, R. E. (2002). DIGITAL IMAGE PROCESSING.
25. ELIAS, P. (1975). UNIVERSAL CODEWORD SETS AND REPRESENTATIONS OF THE INTEGERS. IEEE TRANSACTIONS ON INFORMATION THEORY, 21(2), 194-203.
26. USC, SIPI IMAGE DATASET, [HTTP://SIPI.USC.EDU/DATABASE/](http://sipi.usc.edu/database/), [ONLINE; ACCESSED 19-JANUARY-2018].
27. HUYNH-THU, Q., & GHANBARI, M. (2008). SCOPE OF VALIDITY OF PSNR IN IMAGE/VIDEO QUALITY ASSESSMENT. ELECTRONICS LETTERS, 44(13), 800-801.
28. WANG, Z., BOVIK, A. C., SHEIKH, H. R., & SIMONCELLI, E. P. (2004). IMAGE QUALITY ASSESSMENT: FROM ERROR VISIBILITY TO STRUCTURAL SIMILARITY. IEEE TRANSACTIONS ON IMAGE PROCESSING, 13(4), 600-612.
29. WU, H. T., DUGELAY, J. L., & SHI, Y. Q. (2014). REVERSIBLE IMAGE DATA HIDING WITH CONTRAST ENHANCEMENT. IEEE SIGNAL PROCESSING LETTERS, 22(1), 81-85.

## AUTHORS PROFILE



V. M. Manikandan is currently working as an Asst. Professor in Computer Science and Engineering at SRM University AP, Andhra Pradesh, India. He did his in Computer Science and Engineering from Indian Institute of Information Technology Design and Manufacturing Kancheepuram, Tamilnadu, India after his M.Tech in Software Engineering from Cochin University of Science and Technology, Kerala, India. Currently, his research is focused on the domain of reversible data hiding in images, copy-move forgery detection and digital image watermarking. He is a lifetime member of The Institution of Engineers (India).



V. Masilamani is currently working as an Associate Professor in Computer Science and Engineering at Institute of Information Technology Design and Manufacturing Kancheepuram, Chennai, Tamilnadu, India. He did his Ph.D. in Computer Science and Engineering from Indian Institute of Technology (IIT) Madras, India after his M.Tech in Computer Science and Engineering from Indian Institute of Technology (IIT) Kharagpur, India. He is currently working in design and analysis of digital image processing algorithms. He is also working in the areas such as image quality assessment, digital image watermarking, medical image analysis, etc.