ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(5), DOI: 10.17485/ijst/2016/v9i5/87191, February 2016* ISSN (Online) : 0974-5645

PVO based Reversible Data Hiding with Improved Embedding Capacity and Security

Shounak Shastri and V. Thanikaiselvan*

School of Electronics Engineering, VIT University, Vellore, Tamil Nadu - 632014, India;
shounak.mangeshkalika2015@vit.ac.in, thanikaiselvan@vit.ac.in

Abstract

Background/Objectives: Pixel Value Ordering (PVO) based embedding is a method for image based Reversible Data Hiding (RDH) to plant secret data in minimum and maximum pixel values of each block of the image. This paper serves as an extension of a recently proposed PVO scheme with higher security and embedding capacity. **Methods/Statistical Analysis:** In this proposed method the image is first divided into equal sized non-overlapping blocks. The pixels in the individual blocks are ordered according to their values and differences between the maximum and the second largest (or minimum and the second smallest) are computed. Then, histograms of these differences are modified to hide secret information. But original scheme rejects a lot of blocks belonging to textured regions of the image which can be used to embed even more data. **Findings:** The proposed scheme extends the original PVO method by using these rejected blocks to embed data thus increasing the Embedding Capacity. It is observed that although there is an increase in the overall embedding capacity, the degradation in the visual quality is negligible. It is also found that this increase in the capacity affects the robustness. So a simple randomization technique for the block selection process is also proposed. **Applications/Improvements:** This method can be applied for all the secret communication applications especially Defence, Telemedicine, etc. This proposed method developed further in terms of robust against various steganalysis tools. could be useful to the pharmaceutical industry for the bulk level production of medically important drugs.

Keywords: Embedding Capacity, Pixel Value Ordering (PVO), Reversible Data Hiding, Steganography, Security

1. Introduction

Reversible Data Hiding (RDH) is a technique for copyright insurance which conceals sensitive data in a host. It can not only extract the concealed data but can also restore the host image, so that it can be used to verify the integrity of sensitive covers as in military, medical and other secret communications. Usually, any RDH scheme is assessed by its Peak Signal to Noise Ratio (PSNR) conduct[1]. That is, one expects to reduce the degradation of the host image as much as possible for a given amount of data. As the main purpose of any RDH scheme is to provide security for the embedded data, the robustness of the scheme also plays an important role. However, an attempt to increase any of the three parameters, viz. the embedding capacity, PSNR and robustness, adversely affects the other two.

The early RDH algorithms take into account lossless compression[2-4]. They usually have a low Embedding Capacity (EC) and may prompt severe corruption of the host image quality. Difference Expansion (DE) technique[5]embeds the secret data by expanding difference between adjacent pixels. Prediction-Error Expansion (PEE)[6] is another method which gives a better result than DE. This method may exploit the pixel correlations in a larger neighborhood. In[7], the PEE method is extended by constructing a compressed location map. In similar fashion, an algorithm based on pixel selection is proposed in[8] which guarantee more data embedding in smoother parts by measuring local complexity.

An improved DE[9] method is sorting pixel pairs in a neighborhood to embed data. The pair is said to be in a flat region if the local variance is small and it can be expanded with small difference. Thus sorting only selects smooth pixel pairs for embedding data and the location

** Author for correspondence*

map can be compressed even further thereby reducing the size of the auxiliary information remarkably[8-10] show that a combination of sorting and other reversible techniques like integer transform or PEE can bring about an improvement in the embedding performance.

Histogram shifting[11] can be used for data hiding by first finding the peak and its nearest minimum value and then shifting the whole section from the peak to the minimum. This results in an empty bin in which the secret bits can be embedded[12] divides the image pixels based on their position into even and odd pixels and find the difference between them. The difference histogram is then modified to embed the secret data. Although these schemes have a good PSNR performance, their Embedding Capacity (EC) is very low. Building upon the scheme introduced[11,13] a multi-level reversible data hiding strategy which gave a better hiding capacity by hiding secret bits in histogram of difference images.

A high capacity lossless recovery scheme was introduced in[14] employing VQ to embed data. The scheme uses a codebook which is first sorted and then divided into three groups. The group having the highest frequency is used for hiding the data while the other groups are used for the lossless recovery. But as the size of the codebook increases, the number of codes also increases and the capacity of the scheme decreases. Proposed a scheme which reduced the number of codes and increases the embedding capacity[15]. Proposed a locally adaptive coding scheme for hiding data reversibly using VQ indices which provides a better embedding capacity[16]. As the bugs are increasing in the complex design, verifying the designs through conventional techniques is time consuming in identifying the bugs. Many methodologies have been developed by Semiconductor companies for verification of the design. They are Open Verification Methodology (OVM), Universal Verification Methodology (UVM)[1], etc.

A Pixel-Value-Ordering (PVO) based algorithm was proposed in[17]. Data is embedded by modifying the maximum and minimum in a block of ordered pixels. In this method, a block having n number of pixels with values (p_1 ,..., p_n) is sorted in an ascending order to get ($p_{\omega(1)}$, ..., $p_{\omega(n)}$), where ω:{1,..., n} → {1,..., n} is a one-to-one mapping. Then the prediction-errors, E_{max} and E_{min}, are found for maximum and minimum pixel values. The data is finally embedded by modifying the E_{max} and E_{min} histograms. In this case, the histograms of E_{max} and E_{min} are defined on $[0, +\infty)$. Since they are always positive, the histogram peak is expanded to store secret data and the other bins are shifted. Here, a smooth block is selected for embedding while the rough ones are ignored. This gives a more concentrated histogram and thus, it gives a better embedding performance. However, the bin 0, which implies the maximum and the second maximum (or the minimum and the second minimum) have equal pixel values, is not used and hence, the smooth blocks are completely utilized.

A method with new differences and a different histogram modification strategy to better utilize the smooth blocks was proposed in[18]. In this variation, instead of E_{max} in[17], a new difference d_{max}, where $d_{max} = p_u - p_v$, $u = \min(\omega(n), \omega(n-1))$ and $v = \max(\omega(n), \omega(n-1))$ were used. The two differences E_{max} and d_{max} have a close relation to each other and their corresponding histograms have similar occurrences at 0 with d_{max} introducing a Laplacian-like distribution on $(-\infty, +\infty)$ and 0 as the center. The bits can be embedded in the bin 0 and thus, the blocks with $p_{\omega(n)} = p_{\omega(n-1)}$ can be better utilized. This method has been combined with histogram shifting in[19] to improve the visual quality of the marked images by utilizing the unused bins in the difference histograms.

This work introduces an improvement in the PVO strategy proposed in[8]. Now, this method rejects a lot of blocks as being a part of a textured or a rough portion of the image or simply because there is a possibility of overflow/underflow if we try to embed bits in those blocks. A lot of possible embeddable space is being wasted because of this consideration. This work suggests the utilization of these blocks rejected by the earlier method thus increasing the maximum Embedding Capacity without affecting the PSNR thus maintaining the visual quality of the image.

1.1 Releted Works

In this section, the works proposed by[17] and[18] are introduced. Our proposed method is an extension of these works.

1.1.1 PVO based RDH Scheme

A PVO-based scheme employing PEE is presented in work[17]. Secret bits can be concealed in an image by modifying the maximum and minimum values of pixels in a sorted block with sufficient visual quality. Here, the maximum modification embedding scheme is briefly presented. The minimum modification embedding is omitted for the sake of clarity.

The host image is first divided into non-overlapping blocks of equal sizes. For a given block P having n pixels, the pixel values $(p_1,...,p_n)$ are sorted in ascending order to get $(p_{\omega(1)},...,p_{\omega(n)})$, where $\omega : \{1,...,n\} \rightarrow \{1,...,n\}$ is the one-to-one mapping in which $p_{\omega(1)} \le p_{\omega(n)}$, $\omega(i) \le \omega(j)$ if $p_{\omega(i)}=p_{\omega(j)}$ and i< j. The second largest value, $p_{\omega(n-1)}$, can be used to obtain a prediction for the maximum $p_{\omega(n)}$. Then the prediction-error is given by:

$$E_{max} = p_{\omega(n)} - p_{\omega(n-1)} \tag{1}$$

A histogram of E_{max} is generated. It can be seen that E_{max} will always be a positive quantity defined in the $[0,+\infty)$. The bin with the highest peak (which is usually 1) is expanded to insert secret bits and the bins larger than that are shifted for reversibility shown in Figure 1. For this case, the prediction error, E_{max}, is modified as

$$\tilde{E}_{\max} = \begin{cases} E_{\max} & if\ E_{\max} = 0 \\ E_{\max} + s & if\ E_{\max} = 1 \\ E_{\max} + 1 & if\ E_{\max} > 1 \end{cases} \tag{2}$$

Where $s \in \{0, 1\}$ is a data bit to be concealed. The maximum $p_{\omega(n)}$ is modified as

$$\tilde{P} = P_{\omega(n-1)} + \tilde{E}_{\max} = \begin{cases} P_{\omega(n)} E_{\max} & if\ E_{\max} = 0 \\ P_{\omega(n)+S} & if\ E_{\max} = 1 \\ P_{\omega(n)+1} & if\ E_{\max} > 1 \end{cases} \tag{3}$$

Rest of the values $p_{\omega(1)},...,p_{\omega(n-1)}$ are kept unchanged. Thus, the marked value of P becomes $(g_1,...,g_n)$, where $g_{\omega(n)}=\tilde{p}$ and $g_i = p_i$ for every $i \neq \omega(n)$.

In this process, as the maximum $p_{\omega(n)}$ either increases or remains the same, the pixel value order (mapping ω) remains the same after embedding. This ensures proper extraction of the embedded bits and a lossless image restoration. A high visual quality is also maintained as the change in the pixel values is 1 at the most.

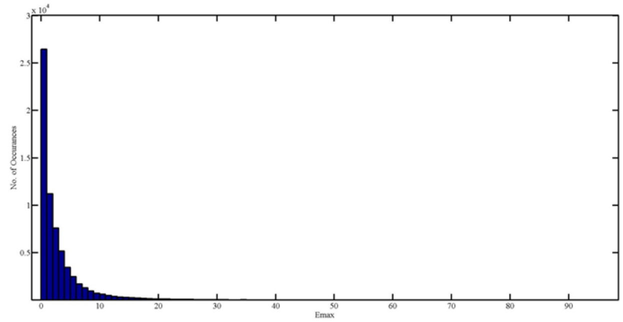

Figure 1. Histogram of E_{max} values for Lena with 2x2 blocks.

1.1.2 Improved PVO Based RDH Scheme

1.1.2.1 Maximum Pixel Value Modification

This scheme presented[18] improves the one in[17]. Secret bits are concealed in an image by modifying the bins 0 and 1 as opposed to only the maximum peak in the previous scheme, thus, improving the embedding capacity. This scheme can also employ larger blocks as compared to the previous scheme and thus use the image redundancies more effectively. This scheme is introduced in detail as the proposed work is based on this scheme.
Consider first,

Table 1. Comparison of pure payload Embedding Capacity

	Embedding Capacity			
	Ni et al. [11]	Lee et al. [12]	Peng et al. [18]	Proposed
Lena	5264	20591	120184	124962
Elaine	4910	14120	118548	123444
Tiffany	8676	24624	119486	124709
Baboon	5504	8409	81726	104522
Fishing Boat	11375	14181	107808	118160
Sailboat On Lake	6724	15004	107888	118221
Airplane (F - 16)	14628	31816	115898	122830
House	11889	26667	103626	116098
Splash	9426	31051	127916	125440
Peppers	5402	17205	120344	125035
Tree	2365	5860	25078	28641
Average Capacity	7833	19048	104409	112006

$d_{max} = p_u - p_x$ (4)

Where, u=min (ω (n), ω (n-1))
And v=max (ω (n), ω (n-1))

Here, the difference d_{max} is used instead of E_{max} which takes values in the interval (-∞, +∞). This range of values is due to the ability to take $p_{\omega(n)} - p_{\omega(n-1)}$ as well as $p_{\omega(n-1)} - p_{\omega(n)}$ which gives a laplacian like distribution with 0 as the centre. This enables us to use the blocks with $p_{\omega(n)} = p_{\omega(n-1)}$ which was not possible in the previous method.

Now Equations (2) and (3) can be modified as

$$\tilde{d}_{\max} = \begin{cases} d_{\max} & if\ d_{\max} = 0 \\ d_{\max} + s & if\ d_{\max} = 1 \\ d_{\max} + 1 & if\ d_{\max} > 1 \\ d_{\max} - s & if\ d_{\max} = -1 \\ d_{\max} - 1 & if\ d_{\max} < -1 \end{cases} \quad (5)$$

Where, s ∈ {0, 1} is the data to be concealed. The marked value of $p_{\omega(n)}$ can be computed by

$$\tilde{p} = p_{ù(u-1)} + |\tilde{d}_{\max}|$$

$$\tilde{p} = p_{\omega(n-1)} + |\tilde{d}_{\max}| = \begin{cases} p_{\omega(n)} & if\ d_{\max} = 0 \\ p_{\omega(n)} + s & if\ d_{\max} = 1 \\ p_{\omega(n)} + 1 & if\ d_{\max} > 1 \\ p_{\omega(n)} - s & if\ d_{\max} = -1 \\ p_{\omega(n)} - 1 & if\ d_{\max} < -1 \end{cases} \quad (6)$$

Now, on comparing the embedding rules in the Equations (2) and (3) it can be seen that they are exactly the same except that in Equation (3), the bin 0 is completely ignored. Seeing as bin 0 is the maximum in this case, the authors expand bins 0 and 1 to hide data instead of -1 and 1. Now d_{max} becomes,

$$\tilde{d}_{\max} = \begin{cases} d_{\max} + s & if\ d_{\max} = 1 \\ d_{\max} + 1 & if\ d_{\max} > 1 \\ d_{\max} - s & if\ d_{\max} = 0 \\ d_{\max} - 1 & if\ d_{\max} < 0 \end{cases} \quad (7)$$

Then the marked value of $p_{\omega(n)}$ is found as

$$\tilde{p} = p_{ù(n-1)} + |\tilde{d}_{\max}|$$

$$\tilde{p} = p_{\omega(n-1)} + |\tilde{d}_{\max}| = \begin{cases} p_{\omega(n)} + s & if\ d_{\max} = 1 \\ p_{\omega(n)} + 1 & if\ d_{\max} > 1 \\ p_{\omega(n)} - s & if\ d_{\max} = 0 \\ p_{\omega(n)} - 1 & if\ d_{\max} < 0 \end{cases} \quad (8)$$

Thus, the new value of P becomes $(g_1,...,g_n)$, where $g_{\omega(n)} = \tilde{P}$ and $g_i = p_i$ for every i ≠ω(n).

As the mapping remains the same, the extraction of the secret bits and the image restoration can be done in accordance to the marked values $(g_1...,g_n)$. Let $\tilde{d}_{\max} = g_u - g_v$ where u and v are defined in Equation (4).

- If $\tilde{d}_{\max} > 0$, then $g_u > g_v$. This means that ω(n) < ω(n-1), u = ω(n) and v = ω(n-1):
- If $\tilde{d}_{\max} \in \{1,2\}$, then the secret bit is $S = \tilde{d}_{\max} - 1$ and the original maximum for that block will be $p_{\omega(n)} = g_u - s$.
- If $\tilde{d}_{\max} > 2$, then data is not hidden this block and the original maximum for it will be $p_{\omega(n)} = g_u - 1$.
- If $\tilde{d}_{\max} \leq 0$, then $g_u \leq g_v$. This means that ω(n) > ω(n-1), u = ω(n-1) and v = ω(n):
- If $\tilde{d}_{\max} \in \{0,-1\}$, then the secret bit is $S = -\tilde{d}_{\max}$ and the original maximum for that block will be $p_{\omega(n)} = g_v - s$.
- If $\tilde{d}_{\max} < -1$, then data is not hidden this block and the original maximum for it will be $p_{\omega(n)} = g_v - 1$.

The D_{max} for Lena has been computed and shown in Figure 2.

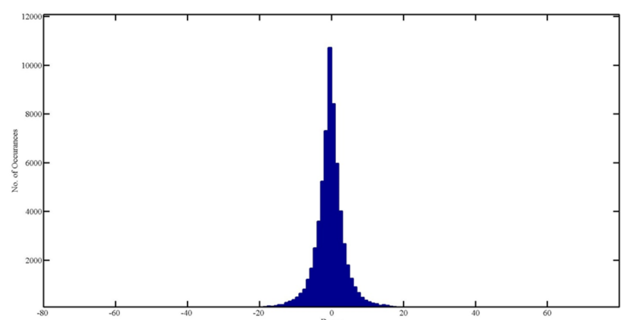

Figure 2. Histogram of d_{max} for the image Lena for blocks of 2×2.

1.1.2.2 Maximum Pixel Value Modification

The method given in section 1.1.2.1 can be directly applied here with minor changes as shown below. Consider first,

$d_{min} = p_q - p_r$ (9)

Where, q=min (ω (1), ω (2))
And r=max (ω (1), ω (2))
Here, the difference d_{min} is modified to

$$\tilde{d}_{\min} = \begin{cases} d_{\min} + s & if\ d_{\min} = 1 \\ d_{\min} + 1 & if\ d_{\min} > 1 \\ d_{\min} - s & if\ d_{\min} = 0 \\ d_{\min} - 1 & if\ d_{\min} < 0 \end{cases} \quad (10)$$

Where, $s \in \{0, 1\}$ is the data to be concealed. The marked value of $p_{\omega(n)}$ can be computed by

$$\tilde{p} = p_{\omega(2)} - \tilde{d}_{\min} = \{(p_{(\omega(1))}) + s \quad if\ d_{\min} = 1 @ p_{(\omega(1))} + 1 \tag{11}$$

Thus, to summarize, the value of P becomes $(g_1,\ldots,g_n)$, where $g_{\omega(1)} = \tilde{P}$ and $g_i = p_i$ for every $i \neq \omega(1)$.

Again, as the mapping remains the same, the extraction of the secret bits and the image restoration can be done in accordance to the marked values $(g_1\ldots,g_n)$. Let $\tilde{d}_{\min} = g_q - g_r$ where q and r are defined in Equation (9)

- If $\tilde{d}_{\max} > 0$, then $g_q > g_r$. This means that $\omega(1) > \omega(2)$, $r = \omega(1)$ and $q = \omega(2)$:
- If $\tilde{d}_{\min} \in \{1,2\}$, then the secret bit is $S = \tilde{d}_{\min} - 1$ and the original minimum for that block will be $p_{\omega(1)} = g_r + s$
- If $\tilde{d}_{\min} > 2$, then data is not hidden this block and the original minimum for it will be $p_{\omega(1)} = g_r + 1$
- If $\tilde{d}_{\min} \leq 0$, then $g_q \leq g_r$. This means that $\omega(2) > \omega(1)$, $q = \omega(1)$ and $r = \omega(2)$:
- If $\tilde{d}_{\min} \in \{0, -1\}$, then the secret bit is $S = -\tilde{d}_{\min}$ and the original minimum for that block will be $p_{\omega(1)} = g_q + s$
- If $\tilde{d}_{\max} < -1$, then data is not hidden this block and the original minimum for it will be $p_{\omega(1)} = g_q + 1$

2. Proposed Method

In this section, the modification to[18] has been proposed which increases the Embedding Capacity. This modification however reduces the robustness of the algorithm. So in the later part of this section a randomization of the block selection procedure is also introduced.

2.1 Modification to the Existing Scheme[18]

Noise Level is a measure to establish whether a block belongs to the smooth region or the edge region. The Noise level is given by

$$N = p_{\omega(n-1)} | p_{\omega(2)} \tag{12}$$

A threshold level T is iteratively calculated for a given Embedding Capacity such that all the bits can be embedded. The noise level is then compared with this threshold. If $N < T$, The block belongs to a smooth region, then 2 bits can be embedded in that block by using the maximum and minimum modification schemes. But if $N \geq T$, the block is said to be in the edge region and no data is embedded. This rejection of blocks in rough areas leaves a lot of unutilized space. The proposed algorithm embeds not more than one bit in blocks with $N \geq T$. This increases the EC considerably.

2.2 Randomization of the Block Selection Procedure

The use of blocks with $N \geq T$, compromises the robustness of the procedure as all the blocks (except the ones with a possibility of overflow or underflow) are sequentially used for data implanting. In order to improve the robustness of the scheme, a pseudorandom binary map of the blocks is created which can be used for block selection. In this process, instead of embedding bits sequentially, first all the 1s are found in the map and the bits are embedded in those blocks. Then, embed the bits are embedded in the blocks with 0s thus randomizing the embedding process. Figure 3 further explains the working of this process.

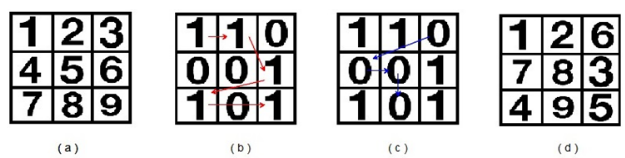

Figure 3. The order of selection of blocks: **(a)** Initially, the bits are embedded sequentially in the blocks **(b)** Randomization by embedding bits in blocks marked as 1 **(c)** Randomization by embedding bits in blocks marked as 0 **(d)** Random embedding is brought about by first embedding bits in blocks marked as 1 then in blocks marked as 0.

2.3 Data Embedding Procedure

Step 1 (Image partitioning): Partition the cover image into non-overlapping blocks $\{P_1\ldots P_N\}$ where each block has 'n' number of pixels. For each block P_i, sort the pixel values $(p_1\ldots p_n)$ in ascending order to get $(p_{\omega(1)},\ldots, p_{\omega(n)})$.

Step 2 (Location map and randomization matrix): A simple Location map and a binary randomization matrix are created in this step to check for overflow and underflow problems in a block and randomize the block selection procedure. For a block Pi, if there exists a $p_{\omega(1)} = 0$ or $p_{\omega(n)}=255$, then modification in this block may result in overflow/underflow problems. So this block is labeled as $C(i) = 1$. Else, it is labeled as $C(i) = 0$. A pseudo-random binary matrix *R*, having the same dimensions as that of the location map is also created. The location map and the randomization matrix are then compressed using a suitable lossless compression technique to obtain C_L and C_R.

Step 3 (Data embedding): In this step, the data is embedded in the host image. First calculate the noise level for each block Pi by Equations (12). Then, for each 1 in the randomization matrix,

- If L (i) = 1, there is a possibility of overflow/underflow in that particular block then no data is embedded in it.
- If L (i) = 0 and $N \geq T$, the block belongs to the edge region and 1bitdata is embedded using either maximum or minimum pixel value modification. Care should be taken that the technique used to embed bits in such blocks is uniform throughout the entire process.
- If L (i) = 0 and $N < T$, the block belong to the smooth region. Two bits are embedded in this block using the maximum and minimum pixel modification. So, dmax and dmin are calculated using equations (4) and (9). Then the maximum and minimum are shifted or expanded to embed data according to Equations (8) and (11).

After completing the embedding process for 1s in R the same process is repeated for the 0s in R. This step is repeated till all the secret steps have been embedded. The element in R at which this process stops is marked as Pstop.

Step 4 (Key embedding): In this step, the auxiliary information which will be used for decoding, C_L and C_R, is embedded in the rest of the host image. First the Least Significant Bits (LSBs) of the first 12 + 3[log2N] $+I_{CL} + I_{CR}$ are copied into a binary sequence SLSB and are replaced by the auxiliary information, the compressed location map and the compressed randomization matrix. The auxiliary information consists of

- Block dimensions k (2 bits) and l (2 bits) where n = k × l.
- Noise threshold T (8 bits)
- Indicator for the cursor position in the randomization matrix Pstop ($\lceil \log_2 N \rceil$)
- Length of C_L and C_R given by I_{CL} andI_{CR}. ($2 \times \lceil \log_2 N \rceil$)

Finally, embed the extracted sequence S_{lsb} using the same method in step 3 but starting from P_{stop} + 1 in R.

2.4 Data Extraction Procedure

Step 1 (Auxiliary information, location map and randomization matrix extraction): Auxiliary information is nothing but the LSBs of first 12 + 3log2N pixels of the marked image. This contains the lengths I_{CL} and I_{CR}. Read the LSBs of the next I_{CL} and I_{CR} pixels to get $C_L$and C_R. Decompress them to get L and R.

Step 2 (Sequence SLSB extraction): The procedure is same as data embedding. Partition the marked image in non-overlapping blocks $\{G_1,...,G_N\}$ where each G_i has 'n' pixels. Extract SLSB from the blocks $\{G_{Pstop+1},...,G_N\}$. For a block G_i ($i > P_{stop}$) having ($g_{\omega(1)},..., g_{\omega(n)}$) as pixel values in ascending order:

- If L (i) = 0 and ($g_{\omega(n-1)} - g_{\omega(2)}$) < T, extract the secret data and retrieve the original values from $g_{\omega(n)}$ and $g_{\omega(1)}$ as described in the decoding methods in Sections 1.2.1 and 1.2.2
- Otherwise, there is 1 bit hidden in the minimum or the maximum, whichever has been decided. So again, extract the secret data and retrieve the original values as described earlier.

This step will continue until all of SLSB has been extracted.

Step 3 (Message extraction): Restore the LSBs of the first 12 + 3log2N + I_{CL}+ I_{CR}. Using the LSBs extracted in Step 2. Then again apply the same procedure to blocks $\{G_1,...,G_{Pstop}\}$ which will extract the whole data and also automatically restore the cover image.

3. Results and Discussions

The proposed method is evaluated by comparing it with the PVO-based method[18], histogram shifting based scheme in[11] and the difference histogram expansion scheme in[12]. Eleven standard gray scale images from the USC-SIPI image database namely, Lena, Baboon, Airplane (F-16), Peppers, Fishing boat, Sailboat on lake, Elaine, Tiffany, Splash house and Tree were used in our experiments.

Table 1 shows the comparison between the methods in[11,12,18] and the proposed method on the basis of the pure payload embedding capacity. Pure payload is the data which is to be embedded without the auxiliary information. From this table we can see that the maximum pure payload capacity of the proposed algorithm is significantly higher than the other methods.

The proposed randomization embeds the data according to the randomization matrix. The randomization matrix uses a pseudo random number generator to create a series of binary numbers and the data is embedded according to the positions of the 1s and 0s in the randomization matrix. This scrambles the data at the embedding process and avoids inserting the data sequentially thus increasing the robustness of the scheme.

The size of the blocks and the noise threshold T is set according to the amount of data that needs to be embedded. The larger the size of the blocks and the noise threshold the lesser will be the embedding capacity. But, larger blocks means less pixels would be altered during the embedding process and thus it would yield an output image with a higher visual quality. The threshold T as mentioned in section 2.1 calculated iteratively based on the amount of data that needs to be embedded.

4. Conclusion

In this paper an RDH scheme is presented as an extension of[18] with improved embedding capacity. The proposed scheme utilizes the image redundancies better than the previous work. But in turn the robustness is compromised. A randomization matrix is introduced to deal with this situation which randomizes the block selection procedure. It was checked experimentally that the proposed technique outperforms the original. Future work consists of increasing the robustness of the technique further by devising a better randomization for the embedding.

5. References

1. Caldelli R, Filippini F, Becarelli R. Reversible watermarking techniques: An overview and a classification. EURASIP Journal on Information Security. 2010.
2. Fridrich J, Goljan M, Du R. Lossless data embedding–new paradigm in digital watermarking. EURASIP Journal on Applied Signal Processing. 2002; 2002(1):185–96.
3. Celik MU, Sharma G, Tekalp AM, Saber E. Lossless generalized-LSB data embedding. IEEE Transactions on Image Processing. 2005; 14(2):253–66.
4. Celik MU, Sharma G, Tekalp AM. Lossless watermarking for image authentication: A new framework and an implementation. IEEE Transactions on Image Processing. 2006; 15(4):1042–9.
5. Tian J. Reversible data embedding using a difference expansion. IEEE Transactions on Circuits Systems for Video Technology. 2003; 13(8):890–6.
6. Thodi DM, Rodriguez JJ. Expansion embedding techniques for reversible watermarking. IEEE Transactions on Image Processing. 2007; 16(3):721–30.
7. Hu Y, Lee HK, Li J. DE-based reversible data hiding with improved overflow location map. IEEE Transactions on Circuits Systems for Video Technology. 2009; 19(2):250–60.
8. Li X, Yang B, Zeng T. Efficient reversible watermarking based on adaptive prediction error expansion and pixel selection. IEEE Transactions on Image Processing. 2011; 20 (12):3524–33.
9. [9] Kamstra L, Heijmans H J A M. Reversible data embedding into images using wavelet techniques and sorting, IEEE Transactions on Image Processing, 2005, 14 (12), pp. 2082–90.
10. Sachnev V, Kim HJ, Nam J, Suresh S, Shi YQ. Reversible watermarking algorithm using sorting and prediction. IEEE Transactions on Circuits Systems for Video Technology. 2009; 19(7):989–99.
11. Ni Z, Shi YQ, Ansari N, Su W. Reversible data hiding. IEEE Transactions on Circuits Systems for Video Technology. 2006; 16(3):354–62.
12. Lee SK, Suh YH, Ho YS. Reversible image authentication based on watermarking. Proceedings of the IEEE International Conference on Multimedia and Expo; Toronto, Ont. 2006. p. 1321–4.
13. Lin C-C, Tai W-L, Chang C-C. Multilevel reversible data hiding based on histogram modification of difference images. Pattern Recognition. 2008; 41(12):3582–91.
14. Chang CC, Wu WC, Hu Y-C. Lossless recovery of a VQ index table with embedded secret data. Journal of Visual Communication and Image Representation. 2007; 18(3):207–16.
15. Tu T-Y, Wang C-H. Reversible data hiding with high payload based on referred frequency for VQ compressed codes index. Signal Processing. 2015; 108:278–87.
16. Ma X, Pan Z, Hu S, Wang L. New high-performance reversible data hiding method for VQ indices based on improved locally adaptive coding scheme. Journal of Visual Communication and Image Representation. 2015; 30:191–200.
17. Li X, Li J, Li B, Yang B. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. Signal Processing. 2013; 93(1):198–205.
18. Peng F, Li X, Yang B. Improved PVO based reversible data hiding. Digital Signal Processing. 2014; 25:255-65.
19. Subitha P, Vaithiyanathan V. Novel reversible pixel-value-ordering technique for secret concealment. Indian Journal of Science and Technology. 2015; 8(8):731-40.