J. Vis. Commun. Image R. 40 (2016) 574–588

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Boosting up the data hiding rate through multi cycle embedment process ☆

A.H.M. Kamal [a,b,*], Mohammad Mahfuzul Islam [b]

[a] Department of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh
[b] Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh

ARTICLE INFO

Article history:
Received 29 February 2016
Revised 25 July 2016
Accepted 27 July 2016
Available online 29 July 2016

Keywords:
Data hiding
Multi-layer embedding
Multi-cycle embedding
Reversible data hiding
Prediction error histogram
Steganography

ABSTRACT

Prediction error based multi-layer data embedment schemes conceal secrets into several high frequency errors by modifying their prediction error histogram (PEH). It is investigated that *k*-times data embedment into *n*/*k* errors of PEH produces higher embedding payload, while maintaining better stego-image quality compared to those for embedding into *n* distinct errors for a single time only. This paper proposes a novel multi-cycle embedment scheme in which data is embedded into the errors of a defined range in each of its *k* cycles. Experiments were conducted to examine the performance of the proposed scheme comparing the multi-layer vs. multi-cycle embedding schemes individually and jointly. The scheme explores the points at which significantly better payloads can be obtained at the lower image distortions. Substantial improved performance were obtained during investigations, especially while large volume data embedment. The proposed scheme can embed massive and hybrid data of type text, numeric, image and audio.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Steganograpgy is an art of hiding data into a cover media like audio, video, text and image using a proper encoding method. During the process, the encoder alters the contents of the cover media partially or fully and the encoded cover media is then termed as the stego media. Some steganography schemes [1–13] can reconstruct the original cover media during the extraction process of secret message from the stego media at the receiver end. These schemes are known as the reversible data hiding schemes. Irreversible data hiding schemes [14–20], on the contrary, do not care about the reconstruction of original cover media while extracting the embedded message. The reversible data hiding policies are more challenging than irreversible ones as the information for both the secret and the cover media are preserved completely. Due to maintaining the constraint of reversibility for the cover media, the reversible schemes suffer from the less embedded bits per pixel (bpp) on average, called the embedding capacity. Nevertheless, there are no alternatives for the reversible schemes in the applications where retrieving both the hidden message and the original cover media are equally important [13,18], such as critical scientific analysis, medial and forensic applications and video coding [30]. These schemes also serve better data security, since decoding the stego media for extracting the hidden information from it is difficult without knowing the methodology used for data hiding [1,2,21].

Among the reversible embedding schemes, prediction error based policies [1,2,7,24–27,29] present higher embedding capacity and stronger data security. In these schemes, data embedment is performed into a certain prediction error values known as embeddable errors, rather than into image pixels. The embeddable errors are defined from the peak-presented errors in a *prediction error histogram* (PEH). Embedding is performed into one or two highest peak-presented errors in a single layer data embedment scheme. The single layer data embedment scheme, however, suffers from smaller embedding capacity but ensures the smaller degradation in the stego image quality [3,10,11].

In the recent literature, multiple high frequency layers of errors in the PEH are used to embed data in order to get better embedding capacity [1,2]. It is investigated that in the prediction error based schemes, the highest appeared error is always zero ('0') as shown in Fig. 1 (other than exceptions). Thus, in a single layer data embedment scheme, also known as the layer 0 data embedment scheme, hidden message is implanted either in zero or both in zero and −1. In an *L* layer data embedment scheme, message implantation is performed into the errors whose values range from −*L* to +*L*,

☆ This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author at: Department of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh.
E-mail addresses: kamal@jkkniu.edu.bd (A.H.M. Kamal), mahfuz@cse.buet.ac.bd (M.M. Islam).

http://dx.doi.org/10.1016/j.jvcir.2016.07.023
1047-3203/© 2016 Elsevier Inc. All rights reserved.

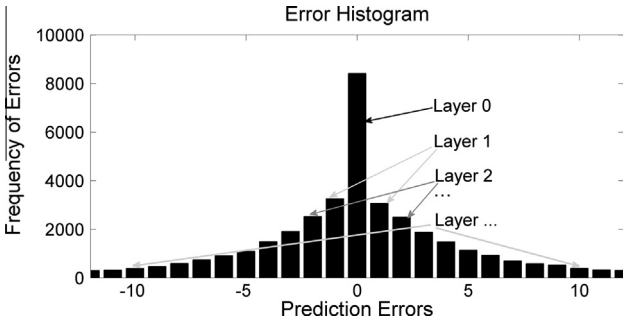


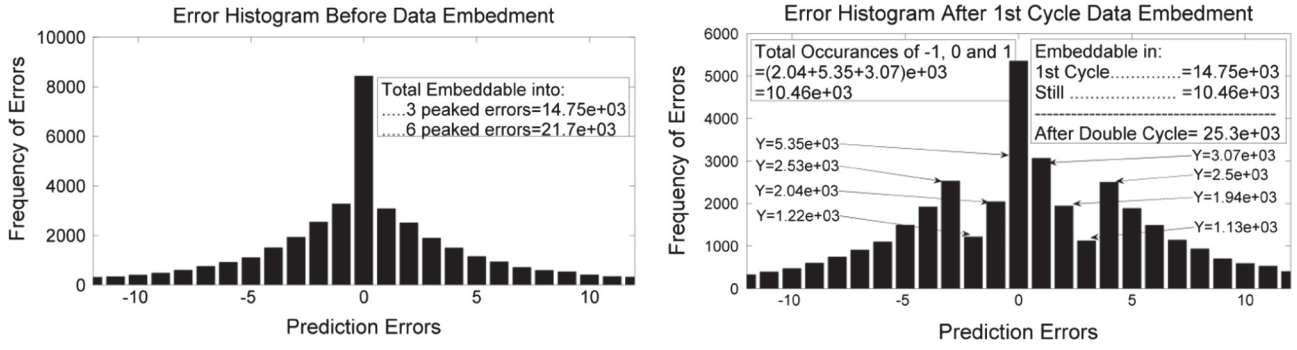
Fig. 1. Different layers in a prediction error histogram.

i.e., $2L$ +1 dissimilar embeddable errors are used. Fig. 1 demonstrates the definition of layers in the PEH. Arrow lines from a layer label indicate the range of the positive and the negative outer errors that are encompassed within that layer. Layer 0 encompasses only one error, called zero. Embedding secret message in layer 1 ($L = 1$) means that message bits are concealed into the prediction errors of -1, 0 and 1, i.e., in $(2L + 1)$ or 3 errors. Similarly, a scheme using 2-layer embedment ($L = 2$) implies that message bits are hidden into error values of -2, -1, 0, 1 and 2, i.e., in 5 errors.

Tai et al. in 2009 [23] and Zhao et al. in 2011 [24] applied adjacent pixel differences as an embedding space in their multi-layer embedding schemes. Luo et al. in 2011 [27] produced a difference histogram by exploiting spatial correlation among block pixels. All of these schemes applied multi-layer embedding approach to conceal message bits into these differences. Embedding capacity for these schemes is still low as they embed bits by modifying the histogram of adjacency pixel differences, known as spatial errors, and it is evident that the frequencies of several peak errors in the PEH are much higher than these in the spatial error histogram. Therefore, many schemes, e.g., Hong in 2012 [1], Leung et al. in 2013 [2], Govind et al. in 2015 [25] and Wang et al. in 2014 [28], utilize the prediction errors as an embedding space to achieve higher embedding capacity. Pan et al. in 2015 [29] introduced a different approach of multi-layer embedding process. They embedded into each block by modifying its pixel histogram. Two neighbor bins of the peak presented pixel were explored to hide data. The produced stego image in this round was reused in the next round to hide more data. In each next round the image block size was increased by a factor of two, e.g. $s \times s$ in first round, $2s \times 2s$ in second round, $4s \times 4s$ in third round and so on. The embedding process was same in each round. The mechanism of reusing the stego image for L times data implantation provides the scheme a flavor of multi-layer like embedding capability.

During the embedment of large volume of data, more error layers are employed in the multi-layer schemes. Nevertheless, using more higher layers ultimately results in smaller increase in embedding capacity because the frequency of each error decreases sharply regarding to its immediate smaller valued error (in their absolute magnitude), as shown in Fig. 1. Employment of higher embedding layers, indeed, increases total error shifting amount and hence, decreases PSNR (peak signal to noise ratio) value of the image. Therefore, rather than embedding data into the errors of L layers for single times, embedding these into L/k layer for k-times will provide larger embedding space as well as embedding payload and better stego image quality. Kamal and Islam in 2015 [26] had employed a double cycle data embedment process using the embedding rules of [22] to boost up the embedding capacity. The scheme successfully achieves the goal. Yet, it is a single layer data embedment scheme and the embedding cycle is fixed to two.

To justify the claims of obtaining better payload and PSNR values from the uses of L/k embedding layers for k-times, Fig. 2 is introduced here. A block median predictor is applied in the Lena image of size $255 * 255$ owing to generate a prediction error histogram, as shown in Fig. 2(a). That figure demonstrates that if 6 topmost peaked errors are exploited in the multi-layer single cycle scheme, the scheme can embed 21,700 bits only. Again, if 3 topmost peaked errors, i.e. errors points of -1, 0 and 1, are exploited, 14,750 bits are embedded by a single cycle process. After the completion of first cycle data embedment into the errors of -1, 0 and 1, the histogram of modified errors is depicted in Fig. 2(b). It states that if multi-layer scheme is applied once more by exploiting the error points of -1, 0 and 1 in Fig. 2(b), it will present more 10,460 bits embedding payload. Thus, embedding into 3 errors for double times will generate 25,300 bits payload which is much greater than 21,700 bits. In this scenario, multi-layer single cycle scheme has to recon 9 highest peaked errors, i.e. error points of $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$, to obtain the embedding payload of 25,300 bits or better because summation of the frequencies of 8 highest peaked errors in Fig. 2(a) is only 25,070. In the scenario of uses 9 peaked errors by single cycle scheme, the positive and negative valued non-embeddable errors have to be shifted respectively by 4 and 5 to prepare enough space for the movement of embeddable errors whereas in the multi-cycle scheme, these negative and positive valued not-embeddable errors are finally shifted by 2 and 4 respectively. The shifting of pixel values is deeply affected by the quantity of exploited peak points. Hence, a scheme will introduce higher order of image distortions by exploiting 9



(a) Prediction error histogram before data embedment

(b) Prediction error histogram after one time data embedment

Fig. 2. Prediction error histogram of Lena image before and after data embedment. From (a) it is noticeable that one can embed maximum 14,750 bits and 21,700 bits if the embedment is tried respectively into 3 and 6 peaked errors in multi-layer single cycle scheme. On the contrary, (b) demonstrate that total 25,300 bits can be embedded if one tries to embed into 3 peaked errors for double cycles.

error points in their single cycle process rather than 3 points in a double cycle process. The scenario will be more promising (both for payload and PSNR) if k is chosen to a larger value than 2. In addition, attacker may classify the highly distorted stego images as well as its embedded contents by employing statistical attacks like SPAM features [31], Benford's Generalized Law [33]. Consequently, increasing the embedding payload by employing more layers in the single cycle scheme may introduce a new dilemma of successful statistical attacks. On the contrary, in addition to improving the image quality, the use of multi-cycles will introduce new security features like number of exercised cycles and quantity of exploited embeddable error points which are parts of a shared key. From these analyses, authors of this article are motivated in favor of the uses of L/k embedding layers for k-times rather than L layers for single time.

In the present study, the authors mitigate both the challenges of improving the embedding capacity and managing better image quality, i.e., increasing PSNR, by proposing a novel multi-cycle embedding scheme where same embedding process is exercised for multiple rounds into the errors of a defined layer. The estimation process for an optimal layer value in multi-cycle scheme is presented in this article. The image distortions for the same embedding payload will be smaller when the proposed multi-layer multi-cycle (MLMC) scheme is used in comparison with the existing multi-layer single cycle (MLSC) scheme; because, to achieve the same capacity of 'k cycle embedding into L/k layers', the multi-layer single cycle scheme has to recon more than L layers. The experimental results and analysis prove that the multi-cycle embedding scheme is better for obtaining both the higher embedding capacity and the image quality simultaneously. Therefore, that proposed scheme will be very useful in supporting the applications of medical, forensic and law-enforcement agencies by hiding their massive and hybrid data generated from interview, interrogation and investigation. Text and audio data related to investigations, interviews and reports are usually large in volume. During hiding such type of massive data and in hybridized manner, data hiding scheme requires large embeddable space. Our proposed scheme will be a useful one to fulfill such higher capacity demand.

The rest of the paper is organized as follows:

In Section 2, reviews of reversible data hiding methods of Leung et al. [2], Wang et al. [28] and Pan et al. [29] are outlined briefly. Section 3 is devoted to detail the proposed multi-cycle embedding model. In the next section, the experimental setup, results and their discussions are demonstrated. Resistance against statistical attacks is tested in Section 5 using SPAM features in support-vector machine [31] and generalized Benford's Law [33]. The paper is finally concluded in Section 6.

2. Related works

This section briefly introduces three recently reported reversible data hiding policies (proposed by Leung et al. [2], 2013; Wang et al. [28], 2014; and Pan et al. [29], 2015) and focuses only to the generation of embedding space, e.g. prediction errors, and the basic process of data embedment by modifying the histogram of either the prediction errors or the image pixels of the related schemes. Some common and obvious parts of literature including assorted stego distortion minimization process, different underflow and overflow (exceeding gray range) handling mechanisms, methodologies of improving location map (binary map to trace the extreme pixels like 255, 0) and assistant information (additional data that assist in reversibility) management policies are not discussed in this literature review section, as our research does not make any amendment to them.

2.1. Review of Leung et al.'s scheme

The scheme divides a cover image into blocks. The median value of each kth block stands for prediction value $P^k_{i,j}$ against the block pixels $B^k_{i,j}$. For smaller sized block, the accuracy of that predictor is very high. The prediction errors are measured by $e^k_{i,j} = B^k_{i,j} - P^k_{i,j}$. Regarding the cover image, these block information $e^k_{i,j}$ and $P^k_{i,j}$ are generalized by $e_{i,j}$ and $P_{i,j}$ respectively. These $e_{i,j}$ are used as an embedding space. The embedding rules used in this scheme are precisely defined by the following Eq. (1).

$$\tilde{e}_{i,j} = \begin{cases} e_{i,j} - L & \text{if } e_{i,j} < -L \\ e_{i,j} + L + 1 & \text{if } e_{i,j} > L \\ 2e_{i,j} + m & 0 \leqslant e_{i,j} \leqslant L \\ 2e_{i,j} - m + 1 & \text{if } -L \leqslant e_{i,j} < 0 \end{cases} \tag{1}$$

The stego image $\tilde{I}$ is formed by adding these modified errors with their prediction values, i.e. $\tilde{I} = P_{i,j} + \tilde{e}_{i,j}$. In each block, the block median value pixel is remained unchanged for the convenience of managing the reversibility, i.e. for generating $\tilde{e}_{i,j}$ and then retrieving message bits and $e_{i,j}$.

2.2. Review of Wang et al.'s scheme

The scheme divides the image pixels into two groups 'O' and 'X' in a chessboard fashion. First, it predicts the pixels of 'O' by associating the pixels of 'X' in the prediction rule and embeds message bits into the prediction errors. Next, it predicts the pixels of 'X' from the stego pixels of 'O' and embeds remaining data bits into these newly computed prediction errors.

It is evident that, other than boarder pixels, each pixel $I_{i,j}$ in the group 'O' has four neighbor pixels 'a', 'b', 'c' and 'd', in the cover image. The chessboard like distribution of pixels into 'O' and 'X' ensures that these 'a', 'b', 'c' and 'd' are the pixels of 'X'. The vice versa is true for the pixels in 'X'. Each pixel $I_{i,j}$ in 'O' is then predicted by averaging the values of these four pixels, i.e. 'a', 'b', 'c' and 'd'. A histogram is drawn from the computed prediction errors $e_{i,j}$. Let the target payload is C. The scheme now works as follows:

It presents two different embedding methodologies based on the target volume of payload. It uses 'rate-distortion model' to embed smaller amount of data in the error histogram. The rate-distortion model aims to minimize the amount of error shifting. Another model named 'sub-optimal scheme' is defined to implant larger amount of data. These two processes are outlined in the following.

2.2.1. Rate-distortion model

(i) Find an optimal bin with frequency f_r in the right side of the histogram so that $f_r > C$. If such bin exists, embed data bits into these bin errors. Otherwise go to next step.

(ii) Find an optimal bin with frequency f_l in the left side of the histogram so that $f_l > C$. If such bin exists, embed data bits into these bin errors. Otherwise, go to next step.

(iii) Checking simultaneously, find two bins with frequency f_l and f_r so that $(f_l + f_r) > C$. If such bins exist, embed data bits into these bin errors. Otherwise go to next step (i.e. first step in sub-optimal scheme).

2.2.2. Sub-optimal scheme: Multi-layer model

(iv) Find two peak presented bins with the frequencies f_l and f_r respectively in the left and right side of the error histogram. If $(f_l + f_r) < C$, measure the maximum value 'f' of these two frequencies and the value 'e' of the corresponding peaked error.
(v) Embed data bits into the errors of value 'e'. Now, remaining payload is $C = C - f$.
(vi) Repeat step iv and v until $(f_l + f_r) < C$. Otherwise go to step vii.
(vii) As a final layer, apply rate-distortion model.

2.3. Review of Pan et al.'s scheme

The scheme divides the cover image into blocks of size $s \times s$. Histogram of spatial values is measured for each block individually. Highest peak point Bp is marked in each block histogram. If Bp is not at the extreme of the histogram (the leftmost or the rightmost), its immediate right and left neighboring peaks will be found at $Bp + 1$ and $Bp - 1$ respectively. If the peak point Bp does not have these two neighbors (i.e.Bp is in extreme in the histogram or bins in either $Bp + 1$ or $Bp - 1$ is empty) the block is skipped. Otherwise, data is embedded by the following rules starting from layer $m = 0$.

i. In each block, the scheme embeds message bits into pixels of $Bp + 1$ and $Bp - 1$ shifting its pixel histogram in right and left directions respectively. It concatenates these modified blocks to form stego image at embedding layer m.
ii. If data bits to be embedded are still in hand, for the next embedding layer $m + 1$ it reuses the stego image and does several operations: sets $m = m + 1$; partitions the stego image into blocks by doubling its dimension, i.e. $2s \times 2s$ for $m = 1$, $4s \times 4s$ for $m = 2$ and so on; compute the histogram of block pixels; measures $Bp + 1$ and $Bp - 1$ in each histogram; and goes to step i.

The scheme does not change the peak presented pixels for the purpose of data extraction and cover image reconstruction.

3. Proposed multi-cycle embedding scheme

The objective of the proposed scheme is to enhance the embedding capacity manipulating the prediction errors in the multi-layer scheme in an efficient way rather than intending to increase the prediction accuracy. Therefore, what predictor is applied in the embedding scheme is not a concerned issue of that section. Now, let $e_{i,j}$ is the measured prediction error of each location (i, j) in the image I and the MC_L is the embedding layers of the proposed multi-cycle scheme. Then, it comprehends that embedding capacity achieved by concealing bits into the errors of MC_L layers of PEH for k times or k-cycles $(k > 1)$, (i.e., $MC\_L = L/k$), will be much higher than implanting bits into the errors of L layers for single cycle for the same or less amount of error-shifting for non-embeddable errors. At the first round of the proposed multi-cycle scheme, bits are embedded into the errors of MC_L layers of the PEH, i.e. into the error points of $-MC\_L, \ldots, -1, 0, 1, \ldots, MC\_L$. The errors are modified due to the bit implantation in this round. The stego errors of first embedding cycle, i.e., errors modified by first cycle, are reused for data embedment in second cycle. Each cycle employs the same embedding procedure to hide data bits into its error points of $-MC\_L, \ldots, -1, 0, 1, \ldots, MC\_L$. The process repeats for k times for a k-cycle embedding scheme. The flowchart depicted in Fig. 3 portrays the process.

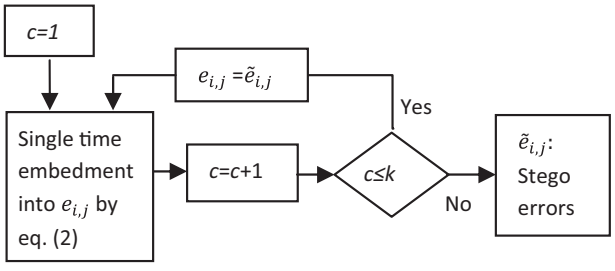


Fig. 3. Flowchart of multi-cycle embedding scheme.

3.1. Defining embeddable errors in MLMC schemes

The MLMC scheme employs more than one layers, i.e., $MC\_L \geqslant 1$, for k-times to embed data bits in its k-cycle scheme. To relate the schemes of MLSC and MLMC for the measurement of embedding efficiency, total encountered error points $(2L + 1)$ in MLSC and $k(2MC\_L + 1)$ in MLMC, i.e. $(2MC\_L + 1)$ for k-times, should be equal valued i.e. $(2L + 1) = k(2MC\_L + 1)$. As L, MC_L and k are integer values, $(2L + 1)$ and $k(2MC\_L + 1)$ will not be equal for all the values of k, L and MC_L, e.g. if $L = 4$ then both $(2L + 1)$ and $k(2MC\_L + 1)$ will be equal only for $k = 3$, $MC\_L = 1$ and if $L = 7$ then the equality will be held only for $k = 5$, $MC\_L = 1$ or $k = 3$, $MC\_L = 2$. To encounter the equal number of embeddable error points or to make this number closer to MLSC's scheme, total encountered embeddable errors in each cycle of MLMC data embedment is measured by $(2MC\_L + 1) = \left\lfloor \frac{l + mod(l,k)}{k} \right\rfloor$, where $l = (2L + 1)$, $mod(l, k)$ is the modulus of l and k and $\lfloor . \rfloor$ stands for the mathematical floor operation. For example, if total encountered error points are 9 in MLSC scheme, it will be 3 in each of the three cycles in MLMC scheme. The $\left\lfloor \frac{l + mod(l,k)}{k} \right\rfloor$ can be either an odd or an even number. Let, $l' = \left\lfloor \frac{l + mod(l,k)}{k} \right\rfloor$. Then, the negative valued embeddable error points will be -1 to $-\left\lfloor \frac{l'}{2} \right\rfloor$ (i.e., total $\left\lfloor \frac{l'}{2} \right\rfloor$ different errors) whereas the positive valued embeddable error points will be either 0 to $\left\lfloor \frac{l'}{2} \right\rfloor - 1$ or 0 to $\left\lfloor \frac{l'}{2} \right\rfloor$ (i.e., $\left\lfloor \frac{l'}{2} \right\rfloor$ or $\left\lfloor \frac{l'}{2} \right\rfloor + 1$ different errors) depending on whether l' is even or odd respectively. Hence, the proposed scheme further replaces the concept of the MC_L by two parameters: number of different negative valued embeddable error points (nNEP) and number of different positive valued embeddable error points (nPEP), while, the negative valued embeddable errors are 0 to $-nNEP$ and positive valued embeddable errors are 0 to $nPEP - 1$. The following pseudo code initializes these two parameters $nNEP$ and $nPEP$.

if $mod(l', 2) = 0$ then $nPEP = \frac{l'}{2}$ and $nNEP = \frac{l'}{2}$
else $nPEP = \left\lfloor \frac{l'}{2} \right\rfloor + 1$ and $nNEP = \left\lfloor \frac{l'}{2} \right\rfloor$
end

The replacement of MC_L by $nNEP$ and $nPEP$ will allow the scheme to extend its scope of selecting arbitrary number of embeddable error points in a side depending on different types of histogram properties and demand of the applications.

3.2. Defining the Principalities of Data Embedment in MLMC Scheme

Each message bit m is implanted into an error $e_{i,j}$, where $-nNEP \leqslant e_{i,j} \leqslant (nPEP - 1)$. Data implantation rules stated in Eq. (2) modify the errors. These modified errors are $\tilde{e}$.

$$\tilde{e}_{i,j} = \begin{cases} e_{i,j} - nNEP & \text{if } e_{i,j} < -nNEP \\ e_{i,j} + nPEP & \text{if } e_{i,j} > nPEP \\ 2e_{i,j} + m & \text{Otherwise} \end{cases} \quad (2)$$

The MLMC scheme utilizes a transposition chipper key to transpose the message bits. It then implants the bit stream M into $nPEP + nNEP$ error points by its k-cycle process. Following steps 1 to 7 define its k-cycle process. This k-cycle process embeds $|M|$ bits, say tBits, where $|.|$ stands for length of a string.

Step 1: *Set cycle* = 1, $tBits = |M|$ *and assign values to k, nPEP and nNEP*
Step 2: *Embed each bit m,* $m \in M$, *into* $e_{i,j}$ *by Eq.* (2)
Step 3: $tBits = tBits - 1$,
Step 4: *If* $tBits \neq 0$ *and errors in all (i, j) locations,* $e_{i,j}$, *are not tried go to step* 2
Step 5: *Set cycle* = *cycle* + 1.
Step 6: *If* $cycle \leqslant k$ *and* $tBits \neq 0$
Set $e_{i,j} = \tilde{e}_{i,j}$ *and go to step* 2.
Else Go to step 7.
End if
Step 7: *Announce the completion and stop the task.*

As a sample, Fig. 4 depicts a 2-cycle embedding scenario where both the $nNEP$ and $nPEP$ are 4. The figure narrates the functionality of Eq. (2) in each embedding cycle. Finally, the stego image $S_{i,j}$ of the cover image $I_{i,j}$ is formed by adding these modified errors $\tilde{e}_{i,j}$ with its corresponding predicted values.

3.3. Defining the principalities of data extraction and cover image reconstruction in the MLMC scheme

The pixels participated in the prediction rules in the encoder side were remained unchanged by the encoder [2] or the same are generated by this time by the decoder [1]. This implies that the same predicted values are measurable by the decoder. The decoder easily computes stego errors $\tilde{e}_{i,j}$ by subtracting the predicted values from the respective stego contents.

In each embedding cycle, the encoder modified and dispersed positive valued embeddable errors from 0 to $2nPEP - 1$ and negative valued embeddable errors from -1 to $-2nNEP$ by the embedding rules, i.e. Eq. (2). Following Eq. (3) extracts the message bits from these $[-2nNEP, 2nPEP - 1]$ ranged stego errors, i.e., from $\tilde{e}_{i,j}$ where $-2nNEP \leqslant \tilde{e}_{i,j} < 2nPEP$.

$$m = \begin{cases} 1 & \text{if } mod(|\tilde{e}_{i,j}|, 2) = 1 \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

All the modified errors in a cycle are reconstructed by Eq. (4)

$$e_{i,j} = \begin{cases} \lceil \tilde{e}_{i,j}/2 \rceil & \text{if } -2nNEP \leqslant \tilde{e}_{i,j} \leqslant -1 \\ \lceil \tilde{e}_{i,j}/2 \rceil & \text{if } 0 \leqslant \tilde{e}_{i,j} < 2nPEP \\ \tilde{e}_{i,j} + nNEP & \text{if } \tilde{e}_{i,j} < -2nNEP \\ \tilde{e}_{i,j} - nPEP & \text{if } \tilde{e}_{i,j} \geqslant 2nPEP \end{cases} \tag{4}$$

where $\lceil . \rceil$ and $|.|$ stand for mathematical ceiling and absolute value of an expression respectively.

Following steps 1–6 are executed to extract the message bits from the stego prediction errors $\tilde{e}_{i,j}$ and to reconstruct the original prediction errors $e_{i,j}$.

Step 1: *Set embedding cycle counter c = k*
Step 2: *Apply* Eq. (3) *to extract each of the message bit m of M which were embedded at* cth-*cycle.*
Step 3: *Apply* Eq. (4) *to reconstruct the errors which were found after* $(c - 1)$th *embedding cycle.*
Step 4: *Set* $c = c - 1$.
Step 5: *If* $c > 0$
Set $\tilde{e}_{i,j} = e_{i,j}$ *and go to step* 2.
Else Go to step 6.
End if
Step 6: *Announce the completion and stop the task.*

Finally, the decoder reconstructs the cover contents by adding the predicted values with the corresponding $e_{i,j}$. The applied predictor, its parameters, the value of k, $nNEP$, $nPEP$ and the message bit transposition chipper key act in a group as a secret key for providing stronger data security at its bit extraction, error reconstruction and data bit reorganization phase.

3.4. Selecting embeddable errors when payload is fixed

In most applications, the length of to-be-embedded data, i.e. $|M|$, is a fixed value. Let, define it by C. Our scheme will embed that target payload C into $nNEP$ negative valued and $nPEP$ positive valued errors by its k-cycle data embedment process. In each cycle, 50% of the samples of the original embeddable errors will exceed the embedding range by the embedding rules. This means that, after each cycle c, $-nNEP$ to $-nNEP/2$ valued negative errors and $nPEP/2$ to $nPEP - 1$ valued positive errors will become non-embeddable for $(c + 1)$th cycle. This implies that -1 to $-nNEP/2$ negative valued error points and 0 to $nPEP/2$ positive valued error points will be reused to embed data in the $(c + 1)$th cycle. As it is shown in Fig. 4, samples of embeddable errors are {−4, −3, −2,

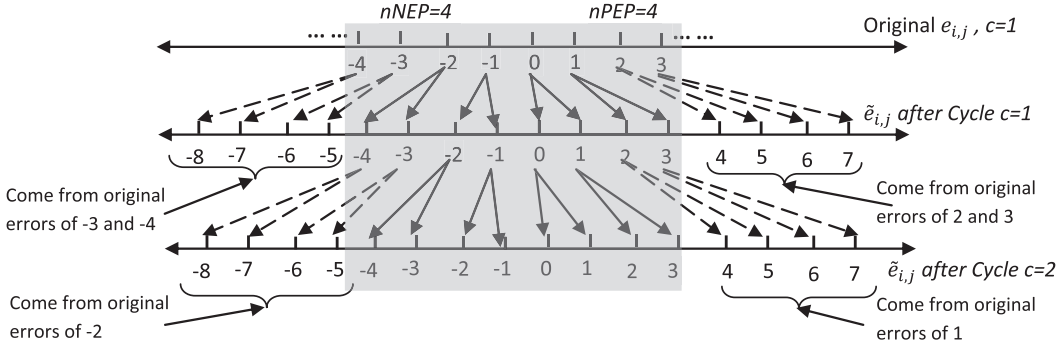


Fig. 4. Error shifting by first and second cycle data embedment process. Original errors $e_{i,j}$ are labeled on the first horizontal line, i.e., −4 to 3. Second and third horizontal lines label the stego errors $\tilde{e}_{i,j}$ after the completion of first and second cycle data embedment respectively. Shaded area states the range of embeddable errors. The possible modification of each error in a cycle is depicted by a pair of down arrowed lines. The dashed lines are used to highlight the errors which are crossing the embedding range after the embedding cycle.

−1, 0, 1, 2, 3}. All these errors accept message bit during their first cycle data embedment phase. The errors dispersed from the original errors of −1, −2, 0 and 1 (i.e. 1/2 of original samples {−4, −3, −2, −1, 0, 1, 2, 3}) are reusable as an embeddable errors in the second cycle. Similarly, original errors −1 and 0 (i.e. 1/4 of original samples {−4, −3, −2, −1, 0, 1, 2, 3}) are reusable for third cycle data implantation. Thus, in the embedding cycle $c = 1$, $c = 2$, $c = 3$, $c = 4, \ldots, c = k$ the scheme will reuse the original positive valued errors which are within the range $[0, nPEP/2^0]$, $[0, nPEP/2^1]$, $[0, nPEP/2^2]$, $[0, nPEP/2^3], \ldots, [0, nPEP/2^{k-1}]$ respectively as an embeddable errors. The picture will be same for the negative valued errors. Consequently, after the completion of k-th cycle, the embedding capacity will be $\sum_{j=0}^{k-1}\sum_{i=-1}^{nNEP/2^j} h(i) + \sum_{j=0}^{k-1}\sum_{i=0}^{nPEP/2^j} h(i)$, where $h(i)$ represent the number of frequencies of an error i. To meet the demand for the embedding payload, this theoretical capacity should be equal to or greater than C. Hence,

$$\sum_{j=0}^{k-1}\sum_{i=-1}^{-nNEP/2^j} h(i) + \sum_{j=0}^{k-1}\sum_{i=0}^{nPEP/2^j} h(i) \geqslant C \tag{5}$$

It is notable that different set of k, $nNEP$ and $nPEP$ values will satisfy the relation, $\geqslant$, in Eq. (5). Among these set of verified values (which satisfy the relation $\geqslant$), a one is defined as an optimal value for k, $nNEP$ and $nPEP$, for which the left side of Eq. (5) generates minimum value. These optimal values can also be defined by relating these with the embedding layer L of single cycle data embedment process while L layers are just enough to accept C-bits of data. Among Eq. (5) satisfying values of k, $nNEP$ and $nPEP$, one set of values is defined as optimal one by following Eq. (6).

$$\underset{k\in\{1,2,\ldots\}, nNep\in\{1,2,\ldots\}, nPEP\in\{1,2,\ldots\}}{\arg\min} k(nNEP + nPEP) \leqslant L \tag{6}$$

Thus, the values of $nNEP$ and $nPEP$ are definable by Eqs. (5) and (6).

4. Result analysis

The prime objective of the proposed MLMC scheme is to improve the embedding capacity. To evaluate the performance of the proposed scheme, digital data are embedded into 200 collected standard images, 500 images of BOSS database and 5000 images of CALTECH101 image dataset (in total 5700 images) and results are investigated on scenarios of MLSC and MLMC circumstances. All the embedding parameters, e.g. payload/capacity, PSNR, structural similarity index (SSIM) and time complexity are analyzed in the process of proposed MLMC scheme and MLSC schemes. As a MLSC scheme, the scheme of Hong in 2012 [1], Leung et al. in 2013 [2], Tai et al. in 2009 [23], Wang et al. in 2014 [28] and Pan et al. in 2015 [29] are experimented and the comparison of these schemes regarding the embedding parameters like payload, PSNR, SSIM and time complexity with the proposed MLMC scheme are demonstrated in the following discussions. First, in the MLMC scheme, results of multi-layer double cycles (MLDC), multi-layer triple cycles (MLTC), multi-layer quadruple cycles (MLQC), multi-layer pentadruple cycles (MLPC), multi-layer sextuple cycles (MLSEXC), multi-layer septuple cycles (MLSEPC) and multi-layer octuple cycles (MLOC) are evaluated among themselves. Thereafter, these are compared with MLSC schemes regarding their performance parameters in this study.

4.1. Experimental setup

Digital data of type text, numerical records, smaller images of about 80 × 80 pixels, tiny audio of amr format are concealed into an image of size 510 × 510 both individually and in hybridized manner. A transposition cipher is used to increase the data security. Firstly, 500 images are randomly copied from all the 5700 stated images to test the ability of concealing hybrid data of large volume. These are resized to 510 × 510 and experimented on a laptop. At the same time, in another computer, 5700 images are resized to 210 × 210 to embed an arbitrary bit stream for the purpose of analyzing and comparing its performance parameters with its competing schemes. The hybrid data is generated from an audio file of 'amr' format of 16 s duration (reading as characters from the file), text of 1000 characters and an image of size 80 × 80. The binary stream M of hybrid data that is to be implanted is consisted of 260,720 bits in total. Again, after the message extraction by the decoder, the binary stream of the audio is converted to characters stream. Though some additional noises are introduced (for reading as characters by the data hider), the reconstructed audio is clear to realize. Hence, in hiding text, image and smaller audio data of types instructions, statements, investigations or interviews in a hybridized manner, the proposed scheme will lead all the other existing schemes.

In the second computer a long binary stream is generated by a random bit ('0' or '1') generator. The MLSC and MLMC schemes are experimented on 5700 images and analyzed thereby sequentially. The data embedment is performed for 1–8 cycles into each of the 99 error points separately in the multi-cycle schemes. All the results are outlined in the following sub-sections.

4.2. Analysis on embedding payload

4.2.1. Mathematical analysis on the embedding payload

Eq. (2) states that the $-nNEP, \ldots, -2, -1, 0, 1, 2, \ldots, nPEP - 1$ valued embeddable prediction errors are modified by $\{-2nNEP, -2nNEP + 1\}, \ldots, \{-4, -3\}, \{-2, -1\}, \{0, 1\}, \{2, 3\}, \{4, 5\}, \ldots, \{2nPEP - 2, 2nPEP - 1\}$ respectively in each embedding cycle depending on message bit m. Let the frequencies of the original prediction errors $z_-, \ldots, -1, 0, 1, 2, \ldots, z_+$ are $h(z_-), ldots, h(-1)$, $h((0), h(1), h(2), ldots, h(z_+)$ respectively in the prediction error histogram where z_- and z_+ denote the highest valued negative and positive error correspondingly. Every embeddable prediction error is diffused into 2^c modified errors in the $MLMC$ scheme after the completion of c-th embedding cycle. The scheme diffuses 0 and 1 valued errors into [0, 3] and [4,7] ranged modified errors respectively after the completion of 2nd cycles, as shown in Fig. 4. In the first cycle of $MLMC$ scheme, all the embeddable errors $nNEP, \ldots, -1, 0, 1, \ldots, nPEP$ contribute to the payload by an amount of $h(-nNEP) + \cdots + h(-1) + h(0) + h(1) + \cdots + h(nPEP - 1)$. By this time, 50% of the original samples of embeddable prediction errors (OSEPE), i.e. 50% of the members of the errors set $\{nNEP, nNEP + 1, \ldots, 0, 1, \ldots, + nPEP - 1\}$, will move to non-embedding range. In the second cycle, only the errors within the range $[-nNEP/2, nPEP/2 - 1]$ will accept the message bits. Consequently, the embedding payload in the second cycle is $h(-nNEP/2) + \cdots + h(-1) + h(0) + h(1) + \cdots + h((nPEP/2) - 1))$. The third cycle will augment the payload by $h(-nNEP/2^2) + \cdots + h(-1) + h(0) + h(1) + \cdots + h((nPEP/2^2) - 1)$. Thus, the total embedding payload for MLMC scheme is approximated by Eq. (7).

$$Payload_{MLMC} = \sum_{j=0}^{k-1}\sum_{i=-1}^{-\lfloor nNEP/2^j \rfloor} h(i) + \sum_{j=0}^{k-1}\sum_{i=0}^{\lfloor nPEP/2^j \rfloor - 1} h(i) \tag{7}$$

4.2.2. Justifying the necessity of using higher embedding cycle for embedding hybrid and massive data

The average and maximum of 500 payloads in each of the cycles and layers in MLMC scheme are measured. These are summarized in Table 1. Only the shaded cell values in the table can meet the demand of the payload of 260,720 bits. The tabulated results state that the single cycle, even the double cycle cannot meet the

Table 1
Achieved average (Avg.) and maximum (Max) payloads (1000×) among the 500 images of size 510 × 510 at various embedding layers and embedding cycles.

Cycle→	1		2		3		4	
Layer↓	Avg.	Max	Avg.	Max	Avg.	Max	Avg.	Max
1	79	100	154	199	215	263	254	303
2	82	101	159	201	231	287	276	341
3	85	101	164	202	237	302	293	364
4	87	101	167	202	247	303	304	380
5	88	102	170	203	248	303	315	390
6	89	102	173	203	250	304	322	399
7	90	102	175	203	254	304	327	404
8	91	102	177	204	257	304	331	404
9	92	103	179	204	261	305	334	405
10	93	103	180	205	262	305	337	405

requirement of that higher embedding capacity within the embedding layer 10. Yet, the average payload that is achieved within the embedding layer 8 in the triple cycle and the layer 1 in the quadruple cycle are still lower than the demanded payload. However, all the maximum payloads in the triple cycle are greater than the stated demanded figure.

Again, let $PL_{l,c}$ represents the payload that are achieved from embedding into l layers for c cycles. If $l_1 \times c_1 = l_2 \times c_2$ for two different values of l and c, where $c_2 > c_1$, the achieved payload in c_2 cycles will be higher than the attained payloads in c_1 cycles, i.e., $PL_{l_2,c_2} > PL_{l_1,c_1}$. For example, regarding to tabulated maximum payloads:

$PL_{1,3}(= 263,000 \text{ bits}) \gg PL_{3,1}(= 101,000 \text{ bits})$,
$PL_{2,3}(= 287,000 \text{ bits}) \gg PL_{3,2}(= 202,000 \text{ bits})$,
$PL_{2,3}(= 287,000 \text{ bits}) \gg PL_{6,1}(= 102,000 \text{ bits})$,
$PL_{2,4}(= 341,000 \text{ bits}) \gg PL_{4,2}(= 202,000 \text{ bits})$
and $PL_{2,4}(= 341,000 \text{ bits}) \gg PL_{8,1}(= 102,000 \text{ bits})$.

From these discussions, it can be understood that to meet the larger embedding payload, higher cycles should be employed and the stated demand for embedding payload can only be met by embedding for triple and its higher cycles.

4.2.3. Analyzing average embedding payload and embedding capacity in MLMC scheme

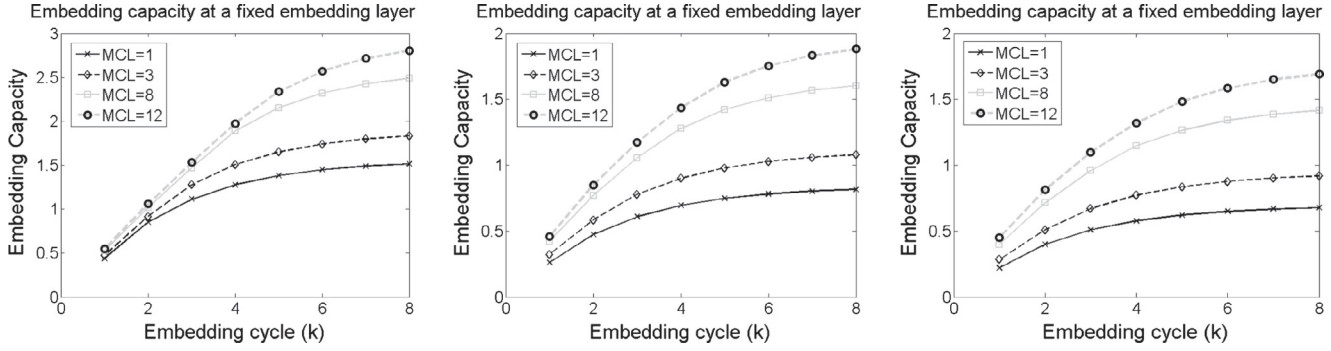
In the MLSC and MLMC schemes, data are implanted into a wider range of error points depending on the requirement of the embedding payloads. In our experiment, the embedding is done into 2–99 error points separately, i.e. up to 49 layers, to analyze the embedding parameters at higher layers. Fig. 5 demonstrates the average embedding capacities achieved in different embedding cycles. The capacities are measured from the values of payload and image size. First the average payloads of n images for each nEP embeddable error points are computed by Eq. (8) for $2 \leqslant nEP \leqslant 99$.

$$AvgPayload(nEP) = \frac{\sum_{r=1}^{n} payload(r, nEP)}{n} \tag{8}$$

where $payload(r, nEP)$ returns the value of embedding payload that is obtained by implanting bits into nEP error points in the r-th image. All average embedding capacities depicted in this article are measured by $AvgPayload(nEP)$/(image_size).

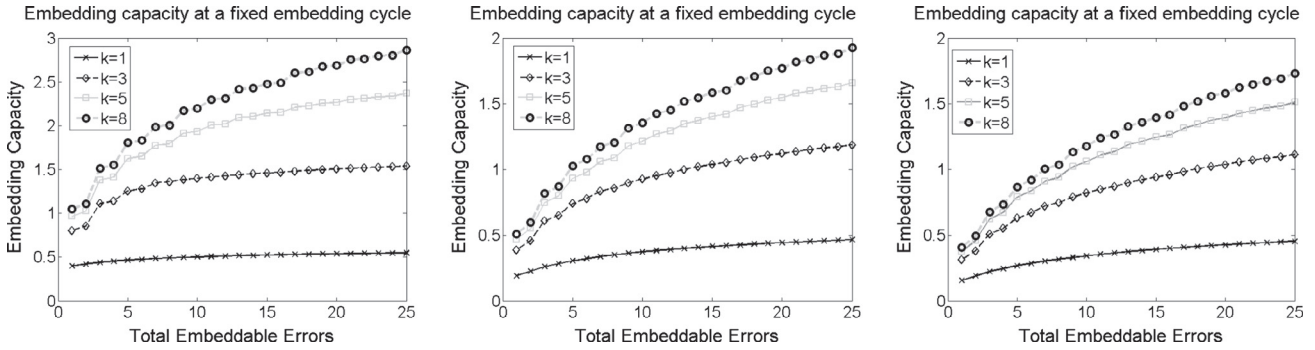
4.2.3.1. Investigating average embedding capacity fixing embedding layer. In this experiment, the behaviors of embedding capacity are investigated in each individual embedding layer for embedding cycle of 1–8. The average capacities achieved in layer 1, 3, 8 and 12 are drawn against 8 embedding cycles in Fig. 5 (other layers are avoided for clear depictions) separately for BOSS, CalTech101 and standard image datasets in the Fig. 5(a), (b) and (c) respectively. The demonstrated results for each of the embedding layers 1, 3, 8 and 12 state that the embedding capacity increases with the increment in the embedding cycles. It is also noticeable that the capacity enhances in the upper embedding layers. Those similarities in the behaviors of embedding capacities are observed in all the image datasets. If the higher layers are employed, e.g. $EL = 8$ or $EL = 12$, then within embedding cycle 4, the capacity is improved by a factor of about 2 with respect to its immediate lower cycle. In overall, geometrical progressions in the embedding capacities are investigated with the advancement in the embedding cycles.

4.2.3.2. Investigating embedding capacity fixing embedding cycle. In this experiment, each time the embedding cycle c is defined to a fix value, say 1 for the first time. The embeddings are performed into each of the 99 embeddable errors for each embedding cycle separately. The results of some sample embedding cycles 1, 3, 5 and 8 for first 25 embeddable errors (for better depiction) are demonstrated in the Fig. 6(a)–(c) independently for three different image datasets. These figures state that the embedding capacity increases both for the employment of higher embedding cycles and more embeddable error points. Though, the embedding capacity increases sharply for first 5 embeddable errors, thereafter, the rate of increment in the embedding capacity decreases gradually because the accounted higher valued errors (resides in the upper layer) contribute fewer to the payload as the frequencies of these errors are smaller. Nevertheless, the achieved capacities in the higher cycles are much greater than the capacities achieved in



(a) Capacity in the BOSS images (b) capacity in the CalTech101 images (c) Capacity in the collected standard images

Fig. 5. Average embedding capacity achieved in each of the embedding layers 1, 3, 8 and 12 in (a) BOSS images, (b) CalTech101 images and (c) standard images. The capacities are investigated at different embedding cycles.



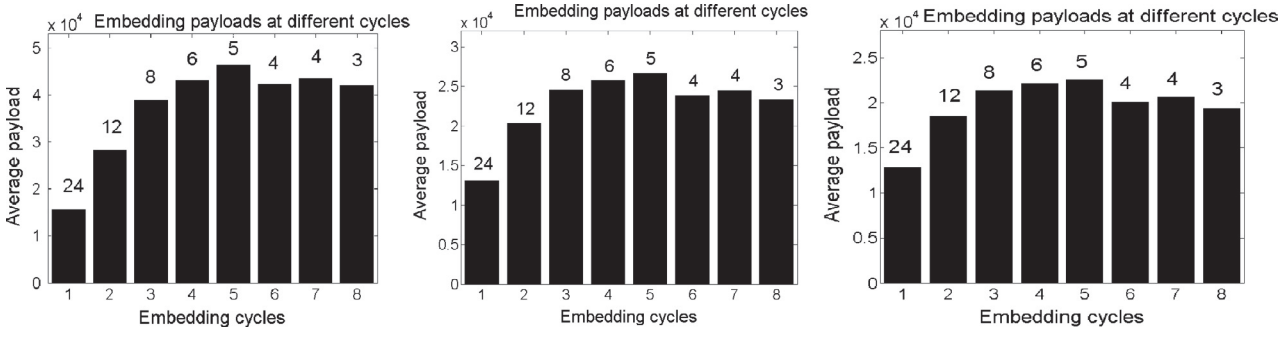
(a) Capacity in the BOSS Images (b) Capacity in the CalTech101 Images (c) Capacity in the collected standard images.

Fig. 6. Average embedding capacity in each of the sample cycles 1, 3, 5 and 8. The results are investigated for embeddable errors 1–25.

MLSC schemes (MLSC when $k = 1$). Another perceptible issue is that the embedding capacities in 3 cycle process are more than the double of the single cycle process. This rate of boosting up the embedding capacities by the process of 5 and 8 cycles reduces gradually. The reason lies in the shifting properties of the embeddable errors. It is already discussed that after each completed cycle, 50% of the members of OSEPE, i.e. $0.5(nNEP + nPEP)$, move to the non-embedding range. Consequently, a good number of embeddable errors become non-embeddable errors after the first few cycles due to their movement by the embedding rules. It implies that the frequencies of the embeddable errors decrease dramatically after few of the cycles. Consequently, the rate of contribution to the embedding payload declines at each of its next cycles. Though, the rate of capacity improvement decreases in the upper cycles, total capacity depicted in each higher cycle is much greater than the single cycle.

4.2.3.3. Investigating embedding payload at equal number of encountered error points. Fig. 7 demonstrates the achieved average embedding payloads in the different embedding cycles. The number annotated on the top of each bar indicates the quantity of embeddable error points nEP_c ($=nNEP + nPEP$) that are applied in the cycle c. The values of nEP_c in each cycle are selected in such a way that it becomes equal to or close to nEP_1/c, i.e. $nEP_c \approx nEP_1/c$. In Fig. 7, nEP_1 is 24. Therefore, total encountered error points are 24, 24, 24, 24, 25, 24, 28 and 24 in the cycle 1, 2, 3, 4, 5, 6, 7, and 8 respectively. This way the encountered error points in all the cycles are equalized at the time of comparing the average payloads in the different cycles individually in each image datasets (Fig. 7(a), (b) and (c) respectively). The payload increases with the increment in the embedding cycles from 1 to 5. Thereafter, it decreases gradually. Though the 7-cycle embedding process demonstrates a bit improvement in the embedding payload compared with 6-cycle embedding process, it considers $nEP_7 = 4$, i.e. total $4 \times 7 = 28$ encountered error points, instead of $nEP_7 = 24/7$ due to the fractional value in 24/7. Hence, a smaller increment in the embedding payload is observed; indeed, the payload decreases there too. The decrements in the payloads at the higher embedding cycles, e.g. $c > 5$ in the Fig. 7, are very rational because, thereafter, the quantity of nEP_c decreases multiplicatively (i.e. by c as $nEP_c = nEP_1/c$). Besides, after each embedding cycle, half of the primitives in OSEPE are shifted to the non-embeddable area. It is already observed in Fig. 2 that in the PEH $h_i \gg h_{i+1}$ and $h_{-i} \gg h_{-i-1}$ for smaller i. Hence, for smaller value of nEP_c, the shifting of 50% of nEP_c to non-embedding range implies that a larger quantity of embeddable error points will cross the embedding range by the embedding rules after the completion of each cycle. Consequently, after executing the embedding process for a few of cycles into nEP_c embeddable errors, the added payloads by these embeddable errors with remaining frequencies become nominal. However, the achieved payloads in cycle 6, 7 and 8, depicted in last three bars in Fig. 7, are much greater than the achieved payload by equal encountered error points in MLSC scheme, depicted in first bar in the same figure. Hence multi-cycle process are better capacity generative than single cycle processes.

Another noticeable point regarding the embedding payload is that the payloads in the higher cycles in the BOSS images, as in Fig. 7(a), are larger than the payloads in the CalTech101 and stan-



(a) Payloads in BOSS images (b) Payloads in CalTech101 images (c) Payloads in each cycle in the collected standard images

Fig. 7. The average payloads in various embedding cycles c; in equal error points; and in individual image databases. The number on the top of each bar indicates the quantity of embeddable errors, nE, in the cycle. Total encountered errors in each cycle c is $nE \times c$, e.g., these are 24, 24, 24, 24, 25, 24, 28 and 24 in cycle 1, 2, 3, 4, 5, 6, 7, and 8 respectively.

dard images, as in Fig. 7(b) and (c) respectively. The reason lies in the less variations among the pixel values in each of the BOSS images. Each of the BOSS images is captured for a single object. Hence, variances of pixel values in these images are smaller than those in the images with natural scene. The less variant pixels aid the predictor to predict more accurately. The frequencies of embeddable errors increase due to better predictions. Consequently, before crossing the embeddable range these errors contribute to the embedding payloads by a large quantity. As a result, better payloads are investigated in the BOSS images.

4.3. Analysis on PSNR

PSNR measures the level of distortions in the stego image with respect to its cover image. Higher valued PSNR signifies less image distortions. Hence, it is expected to have the PSNR value larger. The PSNR is computed from the following Eq. (9).

$$PSNR = 10\log\left(\frac{255^2}{MSE}\right) \tag{9}$$

where the value of mean square errors (*MSE*) between cover image *I* and stego image *S* is defined by

$$MSE = \frac{\sum_{i=1}^{row\_size}\sum_{j=1}^{column\_size}(I_{i,j} - S_{i,j})^2}{row\_size * column\_size}$$

4.3.1. Mathematical representations of MSE in MLMC scheme

From Fig. 4, it is observed that after the completion of *c* embedding cycles, the '0' valued errors are modified by $x_j, 0 \leqslant j < 2^c$ while $x_j \in [0, 2^c - 1]$. The embedding rules apportion all the positive valued embeddable errors *i*, $0 \leqslant i \leqslant nPEP$, to the range $[2^c \times i, 2^c \times (i+1) - 1]$, e.g. '1' diffuses from 8 to 15 for 3 cycles data embedment when $nPEP \geqslant 8$. The total shifting amount of the embeddable errors depends on the quantity of bit '1' in the message stream, embedding layers and embedding cycles. Each positive valued non-embeddable error is shifted by $nPEP * 2^k$ after the completion of *k* cycles. Now, for the simplicity, let the frequency of each error *i*, i.e. h_i, is equally distributed into $[2^c \times i, 2^c \times (i+1) - 1]$ ranged 2^k modified errors after the completion of *k* embedding cycles. The issue of crossing the embedding range by the embeddable errors is also ignored here for the minimalism. These two considerations are taken into account just to measure approximate distortions. Then, a summation of the square of the stego displacement (SSSD) by the positive valued errors is measured in Eq. (10.1).

$$SSSD_{MLMC+} = \sum_{i=0}^{nPEP-1}\sum_{j=0}^{2^k-1}(j + i \times 2^k)^2 h_i/2^k + \sum_{l=nPEP}^{Z_+}(nPEP \times 2^k)^2 h_l \tag{10.1}$$

The inner summation $\sum_{j=0}^{2^k-1}$ sums up for the distortions that are introduced by the frequencies of a single error *i*, i.e. h_i, due to its diffusions into $[2^k \times i, 2^k \times (i+1) - 1]$ modified errors. The outer summation $\sum_{i=0}^{nPEP-1}$ repeats the inner summation for each of the embeddable errors of 0 to $nPER - 1$. The third summation $\sum_{l=NPEP}^{Z_+}$ computes the distortions occurred by the non-embeddable errors. Similarly, the SSSD for the negative valued errors in MLMC scheme, $SSSD_{MLMC-}$, can be defined by Eq. (10.2).

$$SSSD_{MLMC-} = \sum_{I=-1}^{-nNEP2^k-1}\sum_{J=0}(-j + (i+1) \times 2^k)^2 h_i/2^k + \sum_{l=-nNEP-1}^{z_-}(nNEP \times 2^k)^2 h_i \tag{10.2}$$

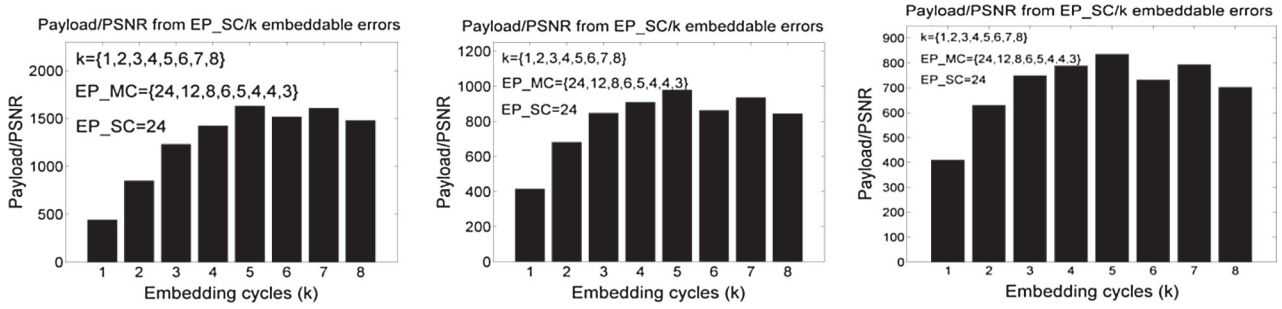
Finally, the total *MSE* is calculated by Eq. (10.3).

$$MSE_{MLMC} = \frac{SSSD_{MLMC+} + SSSD_{MLMC-}}{h \times w} \tag{10.3}$$

where *h* and *w* stands for the height and width of the image.

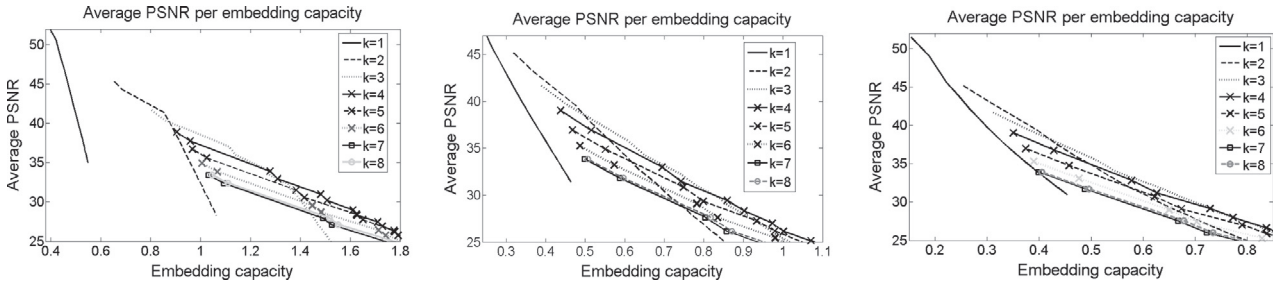
4.3.2. Analysis on the ratio of payload per PSNR in the schemes

Analyzing on payload per PSNR, it is investigated that the MLMC scheme can manage a better tradeoff between the PSNR, embedding payloads and embedding cycles at or above the PSNR of 30 dBm (lower margin of visually distortion realization [4]). The average payloads per PSNR for each embedding cycle are delineated in Fig. 8. The results are depicted separately for BOSS images, CalTech101 images and standard images respectively in Fig. 8(a)–(c). In all the figures, the ratio between payload and PSNR increases logistically for the first five embedding cycles. Thereafter, it starts to decrease because the image distortions in the higher embedding cycles increases by the multiple of about $2^{(k-1)^2}$, e.g. by 2^{0^2} for first cycle, 2^{1^2} for second cycle, 2^{2^2} for third cycle and so on, which are inferred from Eqs. (10.1) and (10.2). Besides, the rate of increment in the payloads after 5th cycles decreases, as in Fig. 7. The reasons are described during the analysis of payloads in this figure. Consequently, the payloads per PSNR decrease after embedding cycle 5. The same results are observed in the images of all the image data-



(a) Payload/PSNR in BOSS images (b) Payload/PSNR in CalTech101 images (c) Payload/PSNR in the collected standard images

Fig. 8. The amount of payload per PSNR against each embedding cycle when the number of encountered error points are about equal. The number of embeddable error points in multi-layer single cycle is *EP_SC*. Here *EP_SC* = 24. The number of embeddable error points in multi-layer multi-cycle is *EP_MC* = ceil (*EP_SC*/*k*). Here, *EP_MC* are {24,12,8,6,5,4,4,3} for cycles {1,2,3,4,5,6,7,8} respectively. The experimental values for *EP_SC* and *EP_MC* are annotated in each figure (a)–(c) for BOSS, CalTech101 and standard image datasets.



(a) PSNR per capacity in the BOSS images (b) PSNR per capacity in CalTech101 images (c) PSNR per capacity in the collected standard images

Fig. 9. Average PSNR per embedding capacity in different embedding cycles.

sets. However, regarding the single cycle process, depicted in the first bar in the Fig. 8, the payloads per PSNR in all the higher cycles are still in dominating level.

4.3.3. Analysis on the ratio of PSNR per capacity in the schemes

Fig. 9 demonstrates the behaviors of PSNR per embedding capacity in all the embedding cycle processes. As the embedding cycle goes higher, the embedding capacity increases. Meanwhile, the PSNR decreases. However, above the embedding capacity of about 0.459 bpp in BOSS images, 0.245 bpp and 0.21 bpp respectively in CalTech101 images and standard images as in figure (a)–(c) of Fig. 9, the multi-cycle embedding processes demonstrate remarkably better PSNR. This implies that when the requirement of embedding capacity is higher, MLMC schemes provide better stego image quality.

4.3.4. Analyzing the PSNR among the MLMC schemes

It is investigated in Fig. 9(b) that after the embedding capacity of 0.245 bpp, double cycle embedding process exhibits improved PSNR value and that superiority of proving higher PSNR by double cycle process continues up to embedding capacity of 0.5 bpp. Within that embedding capacity range, PSNR in the double cycle embedding process varies from 47.5 dBm to 38.2 dBm as in Fig. 9 (b). Again, after the embedding capacity of 0.5 bpp, triple cycle embedding process takes the leads on providing enhanced PSNR value. That leads continues until the PSNR value of the triple cycle falls below 30.8 dBm. At the 30.8 dBm, triple cycle embedment process reaches its embedding capacity to 0.855 bpp. Though thereafter, quadruple and its higher cycles provides improved PSNR regarding to first three cycles, the PSNR falls below 30 dBm. The value of PSNR bellow 30dBm indicates noticeable image distortions [4]. Therefore, in the CalTech101 images, if the required embedding capacity is 0.246–0.5 bpp, the double cycle embedding process meets the requirement of the embedding capacity as well as manages the image quality. The triple cycle is recommended only to meet the requirement for larger embedding capacity. The quadruple and its higher cycles demonstrate worsening results and thus these are not recommended in the applications of image quality controlled based reversible steganography schemes.

4.3.5. Analysis on embedding capacity and embedding cycles at a fixed PSNR

The target of the scheme is to meet the demand for higher embedding capacity and in the meantime to manage the PSNR value of 30 dBm or more. The reason of analyzing embedding capacity at PSNR level of 30 dBm is that the effect of the modification in an image is not visually sensible if PSNR is not less than 30 dBm [4]. Fig. 9 demonstrates the behaviors of PSNR per embedding capacity in all the embedding cycle processes. The figure states that the embedding cycles of 1–8 provide embedding capacity of 0.59 bpp, 1.05 bpp, 1.4 bpp, 1.53 bpp, 1.49 bpp, 1.43 bpp, 1.35 bpp and 1.36 bpp in BOSS image dataset, 0.485 bpp, 0.71 bpp, 0.815 bpp, 0.82 bpp, 0.785 bpp, 0.72 bpp, 0.685 bpp and 0.695 bpp in CalTech101 image dataset and 0.48 bpp, 0.625 bpp, 0.7 bpp, 0.7 bpp, 0.65 bpp, 0.61 bpp, 0.56 bpp and 0.57 bpp in standard image dataset at the PSNR value of 30dBm. These notated capacities conclude that at a same deterioration level, here PSNR of 30dBm, multi-cycles processes provide higher embedding capacity and the trend of progress in the embedding capacity continues till 4th cycles. Though, thereafter, the embedding capacity decreases, these are still better than MLSC and even than first few of the cycles.

From the analysis of Fig. 8 and 10, it can be concluded that multi-cycle embedding schemes present higher PSNR values during the embedment of large data.

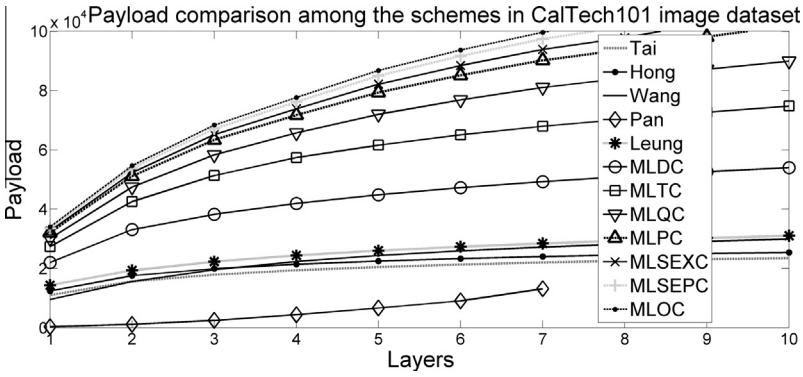


Fig. 10. Average payload in the scheme of Tai et al., Hong, Leung et al., multi-layer double cycle, multi-layer triple cycle, multi-layer quadruple cycle, multi-layer pentadruple cycle, multi-layer sextuple cycle, multi-layer septuple cycle and multi-layer octuple cycle.

4.4. Performance comparison with other schemes

In this subsection the results of the proposed MLMC scheme are compared with the reviewed MLSC schemes, e.g. the schemes of Hong [1], Leung et al. [2], Tai et al. [23], Wang et al. [28] and Pan et al. [29]. The image dataset of CalTech101 is a rich one (consisted of 5000 images) among our experimented datasets. Hence, only the results obtained in that dataset are demonstrated in Figs. 10–12.

4.4.1. Comparing the payloads among the schemes as a measure of performance

Fig. 10 demonstrates the achieved payloads in different schemes at their various embedding layers. Among the MLSC schemes, the most worsening results are investigated in the scheme of Pan et al. for several reasons. Firstly, it embeds data by modifying pixel histogram rather than PEH. Secondly, though it embeds data into two different pixel values $B_p - 1$ and $B_p + 1$, it leaves the highest frequency pixel B_p as unchanged. Thirdly, it skips the block when $B_p - 1$ or $B_p + 1$ does not exist in the histogram (when B_p becomes the leftmost or the rightmost bin in the histogram) or the frequency of, at least one of, them is zero. The payloads in the scheme of Tai et al., Hong, Wang et al. and Leung et al. are very closer and mainly differ by a small quantity for their prediction policies and histogram properties. Among these schemes, the scheme of Leung et al. provides a bit higher payloads because this estimates pixel values more accurately in a smaller sized image block. Again, among the multi-cycle schemes, each upper cycle takes the lead on its immediate lower cycle. The rates of payload improvements in the double cycle and triple cycle processes are happened in an extensive scale. The improvement rate is moderate in quadruple and pentadruple cycles regarding to their lower cycles. In the upper cycles of these two, the payloads increase very gradually. As a whole, the payloads increase in the MLMC processes and a payload improvement is observed toward their upper cycles. To explain that claim, consider the payload of MLTC in layer 3. Fig. 10 depicts that this payload is 4.95×10^4 bits. To consider the equal number of embeddable error points, layer values in MLQC, MLDC and MLSC will be 2.25, 4.5 and 9 respectively (fractional value of layer is taken only for the comparisons). In these equalized embedding layers, the payloads in MLQC, MLDC, Tai et al., Hong, Leung et al., Wang et al. and Pan et al. are 5.1×10^4 - bits, 4.2×10^4 bits, 2.1×10^4 bits, 2.74×10^4 bits, 2.9×10^4 bits, 2.8×10^4 bits and 1.2×10^4 bits respectively. These quoted values states that payloads obtained in MLMC processes are some multiples of the obtained payloads in MLSC schemes. These values also elucidate that higher cycles provide improved payloads.

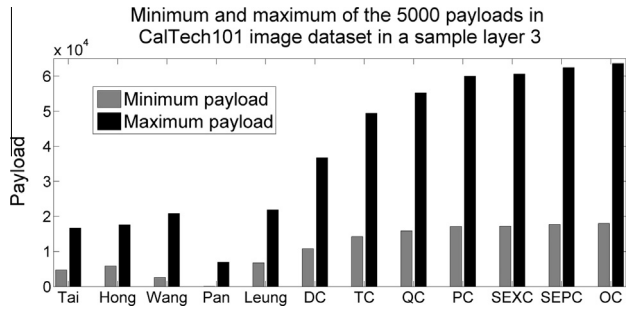


Fig. 11. Minimum and maximum payload that are obtained from the results of 5000 images of CalTech 101 image dataset in a sample layer 3 in the different embedding processes.

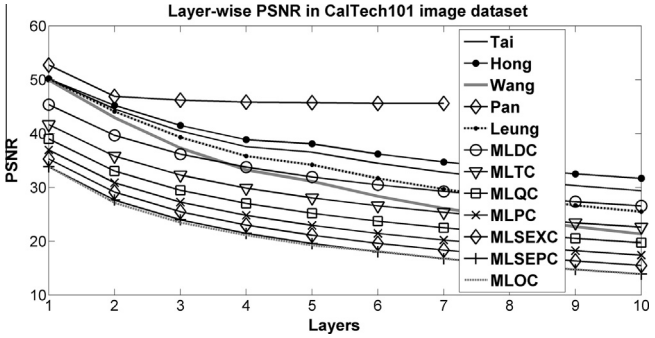


Fig. 12. The average PSNR in MLSC and MLMC schemes in different embedding layers.

Another experiment is carried out to test the behaviors of the minimum and maximum payloads among the experimental results in a specific layer. The results are demonstrated in Fig. 11. Regarding the maximum payload, the multi-cycle scheme depicts higher values. Even the minimum payload of MLQC and its upper cycles are very close to the maximum payloads of the MLSC schemes and higher than some of the maximum payload in MLSC, e.g. than Pan et al. and Tai et al.

4.4.2. Comparing the stego image quality among the schemes

Image quality is measured either by 'pixel difference measurement' or from 'human visual system based matrices'. Mean square error (MSE) and peak signal to noise ratio (PSNR) are two common quality measuring parameters which are computed from the pixel differences of cover and stego images. On the other hand, either universal image quality index (UIQI) or structural similarity index (SSIM) are generally used as human visual system based matrices to compare the structural changes between two images. In the experiment, values of PSNR and SSIM are analyzed to measure the stego quality. The details and the experimental results are discussed in the following two subsections.

4.4.2.1. Comparing the PSNR values among the schemes. Fig. 12 depicts an average of PSNR values (computed from 5000 images) separately in each embedding layer of MLSC and MLMC schemes. It delineates that Pan et al. exhibits outstanding performance because during implanting into second and higher embedding layers, this scheme reuses the stego image. Before each reuses this partitions the stego image varying the size of image block and measures the highest frequency pixel in each block to embed data bits into its immediate smaller and greater valued pixels. During each reuses, the new peak takes place at different position in the histogram of block pixels. Consequently, many pixels which were shifted in the left direction in their earlier cycles, these might be shifted in right direction by the properties of histogram in their next cycle. Therefore, overall displacement of pixel values decreases. The figure also demonstrates that all the MLSC schemes provide larger PSNR values than all the multi-cycle schemes. Among the multi-cycle schemes, the PSNR values decrease as the embedding cycle goes higher. Though, the figure discourages one to choose multi-cycle processes, this does not depict the PSNR at equal number of encountered error points or the PSNRs per embedded bit. For example, at embedding layer 2, PSNR in MLDC process is about 40 dBm. The PSNR values in Tai et al., Hong, Leung et al., Wang et al. and Pan et al. are about 37.5 dBm, 38.5 dBm, 35 dBm, 33.2 dBm and 45.8 dBm respectively at equal number of encountered error points, i.e., $L = 4$. Similarly the PSNR at layer = 2 in MLTC and at layer = 6 in Tai et al., Hong, Leung et al., Wang et al. and Pan et al. are about 35.3 dBm, 34 dBm, 34.9 dBm, 31 dBm, 29 dBm and 45.5 dBm respectively. The PSNR in MLDC and MLTC schemes are higher than the respective PSNR in the MLSC schemes other than the Pan et al.'s method. However, Pan et al.'s method provides very small embedding capacity. Indeed, when the target

is to embed large amount of data and the size of data is fixed, the MLMC schemes provide better PSNR value. Single layer scheme like Tai et al., Hong, Leung et al., Wang et al. and Pan et al. can only be used to meet smaller embedding capacity.

4.4.2.2. Comparing the SSIM values among the schemes. SSIM is widely used to compare the similarities between two images. To measure the SSIM values, both the cover and the stego image are divided into equal parts named block. Mean and variance of each block both in cover and stego image are measured. Say, these are μ_C, μ_S and σ_C^2, σ_S^2 respectively for block in cover and stego image. The covariance between the cover and the stego block is σ_{CS}^2. The SSIM of j-th block is measured by the following Eq. (11).

$$SSIM_j = \frac{(2\mu_C\mu_S + c_1)(2\sigma_{CS} + c_2)}{(\mu_C^2 + \mu_S^2 + c_1)(\sigma_C^2 + \sigma_S^2 + c_2)} \tag{11}$$

with constant $c_1 = (0.01R)^2$ and $c_1 = (0.03R)^2$ where R stands for color range, e.g., R is 255 in gray image.

Finally, an average of all the blocks SSIM are computed as a structural similarity index value.

The results of average SSIM obtained in CalTech101 image dataset are tabulated in Table 2. The scheme of Pan et al. and Hong yields higher similarity index values between cover and stego image because, these two methods skip lot of pixels without modifying by their block skipping criteria and embedding rules. The proposed scheme provides very competitive SSIM value compared to the schemes of Tai et al., Leung et al. and Wang et al. MLMC processes are better than all the other MLSC schemes regarding the average SSIM values per embedded bit. For example, average SSIM is 0.854 for $L = 3$ and $k = 2$. Considering equal number of encountered error points (i.e. at $L = 6$), average SSIM in Wang et al. is 0.843. On the other hand, the rate of boosting up the embedding capacity is significantly higher in MLMC scheme, e.g. in Fig. 10, at $L = 3$, $k = 2$ (MLDC), embedding payload is 4.00×10^4 bits and at $L = 6$, payloads in the single cycle schemes of Tai et al., Hong., Leung et al., Wang et al. and Pan et al. are 2.0×10^4 bits, 2.1×10^4 bits, 2.65×10^4 bits, 2.5×10^4 bits and 1.01×10^4 bits respectively. These results prove that distortion per embedding capacity is smaller in MLMC. The same is observed in the upper subsections regarding PSNR values.

4.4.3. Analyzing the complexity

We compare the time complexity of different schemes by analyzing their required data embedment times. Time complexity of the schemes is measured experimentally. First, the embedding layer is set to 7. Then, each scheme separately embeds same data bits into CalTech101 images. The total time to embed data bits into 5000 images is recorded. The average spending time for each image is tabulated in Table 3. The scheme of Pan et al. presents smallest time complexity because it does not perform any predictions. Rather it embeds data bits by modifying the pixels histogram. The associated affairs in computing the pixel differences in Tai et al. and prediction errors in Leung et al. are less complex than measuring the prediction errors in Hong. Hence, required times in these two schemes are smaller than the required times in Hong. The scheme of Wang et al. takes the highest amount of time to complete the embedding task. Each time it embeds message bits into a single layer. The process repeats for m-times and thus it is addressed as m-layer data embedment method. At each embedding layer, it embeds data bits into the highest peaked errors in the error histogram. The embedding rules perform the data embedment by shifting the errors histogram. It employs the same error shifting policy at its consecutive embedding layers. Thus, for m-layer data embedment, it applies the error shifting policy for m-times. For that reason, it takes much time. On the contrary, in each cycle, our proposed scheme accesses and shifts the errors for single time only during the operation of m-layer data embedment. The error shifting policy is repeated for each embedding cycle and not for embedding layers. Therefore, the time complexity increases along with the increment of embedding cycles. Nevertheless, time complexities in MLDC, MLTC and MLQC are very close to that of other MLSC schemes and much smaller than the required times of Wang et al.'s scheme.

Table 2
Comparison of average values of SSIM in the different schemes.

Layers (L)	MLDC ($k = 2$)	MLTC ($k = 3$)	MLQC ($k = 4$)	Tai	Hong	Leung	Wang	Pan
1	0.886	0.862	0.839	0.874	0.971	0.895	0.875	0.972
2	0.867	0.831	0.791	0.873	0.970	0.891	0.873	0.971
3	0.854	0.806	0.755	0.872	0.969	0.887	0.868	0.969
4	0.842	0.781	0.717	0.871	0.968	0.884	0.861	0.967
5	0.831	0.762	0.691	0.869	0.967	0.881	0.853	0.964
6	0.819	0.740	0.659	0.868	0.966	0.877	0.843	0.961
7	0.810	0.724	0.638	0.866	0.964	0.874	0.832	0.959

5. Resistance against statistical attacks

There are many steganalyzers which can measure the probability of the presence of hidden data inside into an image. In our experiments, two latest steganalyzers are employed to test the resistance of our scheme against such statistical attacks. As in the upper section, the higher embedding cycles are not recommended for data embedment due to the objective of maintaining the image quality, only first four cycles are analyzed in this section. The descriptions of these steganalyzers, their experimenting methodologies, obtained results and the discussions are provided in the following sub-sections.

5.1. Security analysis in SPAM features

Subtractive pixel adjacency matrix (SPAM) based steganalysis [31,32] is a very effective and well known method. The SPAM features are measured by subtracting the adjacency pixels. The differences of adjacency pixels in two opposite horizontal directions (i.e. $\rightarrow$ and $\leftarrow$) constitute two SPAMs. For example, SPAMs in $\rightarrow$ and $\leftarrow$ directions are measured by $I_{i,j} - I_{i,j-1}$ and $I_{i,j-1} - I_{i,j}$ respectively. Similarly, two SPAMs for each of the vertical ($\downarrow$ and $\uparrow$), major ($\searrow$ and $\nwarrow$) and major diagonal ($\nearrow$ and $\swarrow$) axis are calculated. A first order or a second order Markov model utilizes these eight matrices separately to estimate their transition probabilities, i.e. probabilities of changing states in the Markov model. Among these eight transition probability tables, four tables which are found from two horizontal ($\rightarrow$, $\leftarrow$) SPAMs and two vertical ($\downarrow$, $\uparrow$) SPAMs are averaged to form a single table. Likewise, the other four transition tables formed from four diagonal ($\searrow$, $\nwarrow$, $\nearrow$ and $\swarrow$) SPAMs are averaged to form another transition table. A threshold T is used

Table 3
Execution time per image (in second).

Schemes	Tai	Hong	Wang	Pan	Leung	DC	TC	QS	PS	SEXC	SEPC	OC
Time (in s)	3.8	4.5	16.3	3.1	3.9	4.9	5.7	6.4	7.3	9.1	10.9	12.3

in the Markov model to minimize the processing complexity as well as the features in the transition table. A support vector machine (SVM) uses these two tables both to train and to test these features. The SVM is first trained by the features of half of the images. The features of the remaining images are used to test the SVM. In these test phases, the SVM classifies an image as a stego or a cover.

MATLAB tools svmtrain is used to train the *SPAM* features of the images. The values of the parameter 'label' of svmtrain are initialized by '0' and '1' to classify the 'cover' and 'stego' images respectively. The threshold T is set to 4 for the first order Markov model. The features of n cover images and n stego images are applied in first order Markov model with a linear kernel. Another Matlab tool svmclassify is used to test the images and to classify these as stego and cover. Total false positive (classifying cover as stego) and false negative (classifying stego as cover) are collected from the classifier confusion matrix of the tool. The average of false positive probability, P_{FP} and false negative probability P_{FN} i.e. $P_{Err} = \frac{1}{2}(P_{FP} + P_{FN})$, is used as the measure of the performance of the classifier. The values are tabulated in Table 4. Most of the classification errors in the first column are higher than the respective one in the second and third column. The reason is that the embedding process applied into a single error (as $nNEP + nPEP = 1$ in the first column) will shifts the pixel values in the stego image by a smaller amount. Therefore, the miss-classification rate (i.e. P_{FP} and P_{FN}) increases. Again, in the same column, the values decrease from *MLSC* to *MLDC*, *MLDC* to *MLTC* and so on because at higher cycles the shifting amount of each pixel value increases. Consequently, the SPAM features in the stego images differ more from the SPAM features in the cover images and thus the classification error rates decrease. However, in the quadruple cycle process, the classification error increases a bit comparing to the triple cycle process. This is due to the value of threshold T. Though, many of the differences of adjacency pixels exceed $[-T, +T]$ due to their larger shifting by the quadruple cycles, these do not match with any error within $[-4, +4]$ during the search for the matching by the Markov model process. Thus, these do not contribute to transition table. Consequently, the dissimilarities between the transition tables computed for cover and stego image are increased. A better classification accuracy is observed for $nNEP = 2$ and $nPEP = 3$. The phenomenon is due to the shifting of pixel values into a marginal range. The amount of dissimilarities between two transition tables increases and it aids the scheme to discriminate the stego images more successfully. Again in the third column, the classification errors have been increased because the quantity of pixels which are shifted by 4 or more is increased as the number of embeddable error is 9. Thus, lot of pixel differences exceed $[-T, +T]$ and do not contributed to the transition table. That is why, classifier performance decreases there. Finally, it can be concluded that though fluctuations among the classification error rates are observed, the errors are closer and stay within the range [0.2, 0.27]. Thus, multi-cycle processes are not precisely differentiable from the single cycle processes by the *SPAM* feature sets.

Table 4
SVM's classification error rates in first four multi-cycle processes. These error rates are measured in the three image datasets by embedding into the errors of three layers 0, 2 and 4.

Image databases	Schemes	*nNEP*	*nPEP*	*nNEP*	*nPEP*	*nNEP*	*nPEP*
		0	1	2	3	4	5
BOSS 500	MLSC*	0.2771		0.249		0.26908	
	MLDC*	0.2711		0.24699		0.25703	
	MLTC*	0.247		0.249		0.25502	
	MLQC*	0.259		0.251		0.25904	
CALTECH101 5000	MLSC	0.264		0.236		0.264	
	MLDC	0.246		0.236		0.244	
	MLTC	0.234		0.238		0.256	
	MLQC	0.236		0.264		0.252	
Standard 200	MLSC	0.2581		0.2339		0.2016	
	MLDC	0.25		0.2581		0.2258	
	MLTC	0.25		0.2258		0.2097	
	MLQC	0.2258		0.2419		0.2177	

5.2. Security analysis in generalized Benford's law

Steganalysis based on Benford's Law is a very latest, faster and effective mechanism to detect larger modifications in a large volume of natural data [33,34]. The Benford's law is stated from the investigation that, among the significant digits, the probability of appearing '1' as a first significant digit in a large set of natural numbers is more than the probability of appearing '2' and so on, i.e. $P(1) > P(2) > ,\ldots, P(9)$, where $P(n)$ is the appearing probability of n as a first significant digit and $n = 1, 2, \ldots, 9$. The expected values of these $p(n)$ in a large volume of discrete cosine transformed (DCT) coefficients can be measured by the generalized Benford's law (gBL) which is stated in Eq. (12).

$$p(n) = N \cdot \log_{10}\left(1 + \frac{1}{s + n^q}\right), \quad n = 1, 2, \ldots, 9 \tag{12}$$

Here N, q and s define the accuracy of the relation at different compression quality factors. A goodness-of-fit of these three parameters N, q and s are measured using Matlab curve fitting toolbox in [33] for each of the different quality factors 50, 60, 70, 75, 80, 90 and 100. The goodness-of-fit values for N, q and s are used in (12) to estimate the best value by it. For example, for the compression quality factor of 75 the values are fitted to $N = 1.396$, $q = 1.731$ and $s = -0.3549$.

In the experiments, the 'mean distributions' of each digit i, say μ_i, in the DCT coefficients are measured. At the same time the expected distributions p_i of each digit i are computed by Eq. (12). Let the mean distributions and expected distributions in the cover and stego images are μ_i^C, μ_i^S, p_i^C and p_i^S respectively. The percentage of differences between expected distributions and mean distributions for all the digits i are measured both for cover and stego images by $d_i^C = \frac{(p_i^C - \mu_i^C)}{p_i^C} 100\%$ and $d_i^S = \frac{(p_i^S - \mu_i^S)}{p_i^S} 100\%$. If the changes in the stego image due to the data embedment are not happen in a large scale, the values of d_i^C and d_i^S will be closer. A difference d_i between d_i^C and d_i^S, i.e. $d_i = d_i^C - d_i^S$, must be smaller than a threshold T_i, where T_i is a very small value. Eq. (13) will classify an image as a cover or a stego one.

$$ImClass = \begin{cases} Cover \text{ if } d_i < T_i, & i = 1, 2, \ldots, 9. \\ \text{Stego} & \text{Otherwise} \end{cases} \tag{13}$$

For the different quality factors, the minimum value of T_2, i.e. for the digit 2, is listed in [33]. It is 3 for the compression factor of 75%.

Following the goodness-of-fit table in [33], the static parameters N, q and s of Eq. (12) and the threshold T_2 are initialized to 1.396, 1.731, −0.3549 and 3 respectively.

The successfully stego detections by gBL in the embedding layer 2 and 4 for each of the embedding cycles of MLSC, MLDC, MLTC, MLQC are tabulated in Table 5 separately for three different image datasets. It is observed in Table 5 that the detection rate increases for each higher embedding cycles without an exception in the triple cycle process. The reasons of triple cycle becoming an exception are explained in the following. In the curve for triple cycle embedding process in Fig. 6, it can be verified that at embedding layer 2, i.e. embeddable errors 5, the embedding capacity of triple cycles in BOSS, CalTech101 and standard images are 1.2 bpp, 0.75 bpp and 0.61 bpp respectively. From another Fig. 9, it can be

Table 5
Successfully stego detections by gBL when each of the MLSC, MLDC, MLTC and MLQC schemes are applied in the embedding layer 2 and 4 separately.

Image database	MLMC process	Successfully Stego detection			
		Total in $L = 2$	% in $L = 2$	Total in $L = 4$	% in $L = 4$
BOSS 500 images	MLSC	55	11	48	9.6
	MLDC	55	11	50	10
	MLTC	47	9.4	53	10.6
	MLQC	63	12.6	50	10
CalTech101 5000 images	MLSC	490	9.8	607	12.14
	MLDC	607	12.14	617	12.34
	MLTC	529	10.58	539	10.78
	MLQC	666	13.32	588	11.76
Author's 200 images	MLSC	19	9.5	23	11.5
	MLDC	23	11.5	24	12
	MLTC	20	10	21	10.5
	MLQC	26	13	24	12

checked that single cycle, double cycle, triple cycle and quadruple cycle processes provide better image quality respectively into embedding capacity range {[0, 0.459], [0.46, 0.85], [0.851, 1.2], [1.21, 1.7]} in BOSS images, {[0, 0.245], [0.246, 0.5], [0.51, 0.855], [0.856, 1.4]} in CalTech101 images and {[0, 0.21], [0.211, 0.4], [0.41, 0.8], [0.81, 1.235]} in standard images. Therefore, at these embedding capacities of 1.2 bpp in BOSS, 0.75bpp in CalTech101 and 0.61 bpp in standard images, triple cycle provides better image quality. That is why, performance of gBL decreases in triple cycle. Again, at embedding layer 4, i.e. when embeddable errors are 9, it is noticeable in Fig. 6 that the embedding capacities of these images are 1.4 bpp, 0.85 bpp and 0.8 bpp respectively for BOSS, CalTech101 and standard image datasets in MLTC (*i.e.* $k = 3$). At these embedding capacities, CalTech101 and standard images provide better PSNR, as in Figs. 8 and 9. On the other hand, capacity achieved in the BOSS images in the triple cycle embedding process at embedding layer 4 is greater than 1.2. That is why, the detection rate by the gBL in the triple cycle process in BOSS image database has increased. Again, in BOSS image dataset, achieved capacity at embedding layer 4 by quadruple cycle is about 1.55 bpp which is in the range [1.21, 1.7]. That is why; the detection rate decreases in the quadruple cycle. Another exception is also noticeable in the MLSC scheme in the BOSS images for $L = 4$. There the detection rate is small because at $L = 4$, the embedding capacity of MLSC (when $k = 1$) is 0.4, as in Fig. 6, and in that embedding capacity, BOSS images provide better PSNR for single cycle. From that analysis it can be concluded that embedding into multi-layer making trade-off with multi-cycles will meet the demand for the higher embedding capacity as well as it will increase the stego image quality and resistance again statistical attacks.

As a concluding remark of that section it can be mentioned that the proposed multi-cycles embedding schemes provide more resistance than single cycle schemes when the demand for embedding capacity is large. Thus, MLMC schemes will exhibit more resistance against any statistical attacks.

6. Conclusion

Text and audio data related to investigations, interviews and various reports are usually large in volume. During the implantation of such type of massive data and in hybridized manner, data hiding scheme requires high embeddable space. Our proposed scheme will be a pathway to fulfill such higher capacity demand. The proposed multi-layer multi-cycle embedding scheme enhances both the payload and the stego image quality compared with these in multi-layer single cycle schemes. It enhances the embedding capacity at least by a factor of 2 and several times in higher embedding cycles and layers. It can conceal large volume of data as well as hybrid data into an image of reasonable size. It will serve the demand for larger embedding capacity of many applications including those in medical, forensic, military, law-enforcement agencies. It is also extendable to choose arbitrary number of embeddable error points in a side in the histogram during its data implantation without any further modification in its embedding rules. The freedom of defining the number of embeddable error points in a side, choosing predictor and its parameters, selecting embedding cycles and applying transposition cipher into the data bits has made that proposed reversible scheme more robust and secured. Besides, it exhibits more resistance against statistical attacks when large volume of data is embedded. For these reasons, that multi-cycle schemes will obviously be a notable contribution in the field of prediction errors based reversible data hiding arena.

In our future work, to enhance the stego image quality, we will devote our efforts to minimize the error shifting by modifying the data embedment rules and process. Then, multi-cycle embedding schemes will get more attentions of the researchers as well as the application developers.

Author's contributions

The first author, AHMK, is a PhD student of the department of Computer Science and Engineering of the Bangladesh University of Engineering and Technology. He is working under the supervision of second author, MMI. Hence, the whole work is supervised and guided by MMI. Mr. MMI has been consulted on all the way to the progress of the research work by the author AHMK. Mr. AHMK has completed the experiments and made the draft of the manuscript. Mr. MMI has revised the manuscript and given the final approval to submit it to that journal.

Conflict of interest

The authors do not have any economical interest from that article. The first author is a PhD student and working under the supervision of the second author. To meet the requirement for achieving the PhD degree, the first author has to publish his research works on ranked journals which are published by well recognized publishers. Therefore, the authors have chosen this journal to publish that work. Both the authors are aware of that submission. The first author is a fellow of ICT division of the Ministry of Post, Telecommunication and Information Technology of the Government of Bangladesh. However, the fellowship neither covers any publication charges, nor claims any financial interest from that research.

Acknowledgment

The research work is done by the support of Ministry of Post, Telecommunication and Information Technology of the Government of Bangladesh through the Information and Communication Technology Fellowship program. Hence, the authors are happy to acknowledge that remarkable support.

References

[1] Wien Hong, Adaptive reversible data hiding method based on error energy control and histogram shifting, Opt. Commun. 285 (2) (2012) 101–108.
[2] Hon Yin Leung et al., Adaptive reversible data hiding based on block median preservation and modification of prediction errors, J. Syst. Softw. 86 (8) (2013) 2204–2219.
[3] De-Sheng Fu et al., Reversible data hiding based on prediction-error histogram shifting and EMD mechanism, AEU-Int. J. Electron. Commun. 68 (10) (2014) 933–943.
[4] Zhenfei Zhao, Hao Luo, Zhe-Ming Lu, Jeng-Shyang Pan, Reversible data hiding based on multilevel histogram modification and sequential recovery, Int. J. Electron. Commun. (AEÜ) 65 (2011) 814–826.

[5] Sim Ying Ong, Koksheik Wong, Kiyoshi Tanaka, A scalable reversible data embedding method with progressive quality degradation functionality, Signal Process.: Image Commun. 29 (1) (2014) 135–149.
[6] Kuo-Liang Chung, Yong-Huai Huang, Wen-Ming Yan, Wei-Chung Teng, Distortion reduction for histogram modification-based reversible data hiding, Appl. Math. Comput. 218 (9) (2012) 5819–5826.
[7] C.C. Lin, W.L. Tai, C.C. Chang, Multilevel reversible data hiding based on histogram modification of difference images, Pattern Recogn. 41 (12) (2008) 3582–3591.
[8] Lute Kamstra, Henk J.A.M. Heijmans, Reversible data embedding into images using wavelet techniques and sorting, IEEE Trans. Image Process. 14 (12) (2005) 2082–2090.
[9] Sagar Gujjunoori, B.B. Amberker, DCT based reversible data embedding for MPEG-4 video using HVS characteristics, J. Inform. Security Appl. 18 (4) (2013) 157–166.
[10] Ou Bo et al., Reversible data hiding based on PDE predictor, J. Syst. Softw. 86 (10) (2013) 2700–2709.
[11] Wei-Jen Yang et al., Efficient reversible data hiding algorithm based on gradient-based edge direction prediction, J. Syst. Softw. 86 (2) (2013) 567–580.
[12] Xiaoxiao Ma et al., High-fidelity reversible data hiding scheme based on multi-predictor sorting and selecting mechanism, J. Visual Commun. Image Represent. 28 (2015) 71–82.
[13] Hao-Tang Chan, Wen-Jyi Hwang, Chau-Jern Cheng, Digital hologram authentication using a hadamard-based reversible fragile watermarking algorithm, J. Display Technol. 11 (2) (2015) 193–203.
[14] Saiful Islam, Phalguni Gupta, Effect of morphing on embedding capacity and embedding efficiency, Neurocomputing 137 (2014) 136–141.
[15] Wafaa Mustafa Abduallah, Abdul Monem S. Rahma, Al-Sakib Khan Pathan, Mix column transform based on irreducible polynomial mathematics for color image steganography: a novel approach, Comput. Electr. Eng. 40 (4) (2014) 1390–1404.
[16] Wien Hong, Tung-Shou Chen, Chih-Wei Luo, Data embedding using pixel value differencing and diamond encoding with multiple-base notational system, J. Syst. Softw. 85 (5) (2012) 1166–1175.
[17] Wien Hong, Tung-Shou Chen, A novel data embedding method using adaptive pixel pair matching, IEEE Trans. Inform. Forensics Security 7 (1) (2012) 176–184.
[18] A.H.M. Kamal, M. Mahfuzul Islam, Facilitating and securing offline e-medicine service through image steganography, Healthcare Technol. Lett. 1 (2) (2014) 74–79.
[19] Xiang-yang Wang et al., J. Syst. Softw. 86 (2) (2013) 255–277.
[20] Wei-Jen Wang, Cheng-Ta Huang, Shiuh-Jeng Wang, VQ applications in steganographic data hiding upon multimedia images, IEEE Syst. J. 5 (4) (2011) 528–537.
[21] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (3) (2006) 354–362.
[22] Wien Hong, Tung-Shou Chen, A local variance-controlled reversible data hiding method using prediction and histogram-shifting, J. Syst. Softw. 83 (12) (2010) 2653–2663.
[23] Wei-Liang Tai, Chia-Ming Yeh, Chin-Chen Chang, Reversible data hiding based on histogram modification of pixel differences, IEEE Trans. Circuits Syst. Video Technol. 19 (6) (2009) 906–910.
[24] Zhenfei Zhao et al., Reversible data hiding based on multilevel histogram modification and sequential recovery, AEU-Int. J. Electron. Commun. 65 (10) (2011) 814–826.
[25] P.V. Sabeen Govind, M. Wilscy, A new reversible data hiding scheme with improved capacity based on directional interpolation and difference expansion, Proc. Comput. Sci. 46 (2015) 491–498.
[26] A.H.M. Kamal, M.M. Islam, Capacity improvement of reversible data hiding scheme through better prediction and double cycle embedding process, in: Proc. of IEEE Int Conference on Advance Networks and Telecommunication Systems, Kolkata, India, 2015, pp. 7–12.
[27] H. Luo, F.X. Yu, H. Chen, Z.L. Huang, H. Li, P.H. Wang, Reversible data hiding based on block median preservation, Inf. Sci. 181 (2) (2011) 308–328.
[28] J. Wang et al., An efficient reversible data hiding scheme using prediction and optimal side information selection, J. Vis. Commun. Image R 25 (6) (2014) 1425–1431.
[29] Z. Pan et al., Reversible data hiding based on local histogram shifting with multilayer embedding, J. Vis. Commun. Image R 31 (2015) 64–74.
[30] Chenggang Yan et al., Efficient parallel framework for HEVC motion estimation on many-core processors, IEEE Trans. Circuits Syst. Video Technol. 24 (12) (2014) 2077–2089.
[31] Tomas Pevny, Patrick Bas, Jessica Fridrich, Steganalysis by subtractive pixel adjacency matrix, IEEE Trans. Inform. Forensics Security 5 (2) (2010) 215–224.
[32] Jessica Fridrich, Jan Kodovsky, Rich models for steganalysis of digital images, IEEE Trans. Inform. Forensics Security 7 (3) (2012) 868–882.
[33] Panagiotis Andriotis, George Oikonomou, Theo Tryfonas, JPEG steganography detection with Benford's Law, Digital Invest. 9 (3) (2013) 246–257.
[34] Dongdong Fu, Yun Q. Shi, Wei Su, A generalized Benford's law for JPEG coefficients and its applications in image forensics, Electronic Imaging 2007, International Society for Optics and Photonics, 2007.