

Improving Prediction Error Expansion based Data Hiding Method for Securing Confidential Data

Hendro E. Prabowo, Tohari Ahmad

Abstract—Development of information technology has supported the delivery of multimedia document such as images, audio or video that can be applied in many fields of activity. Nevertheless, this implementation requires security treatment for sensitive data. Reversible Data Hiding methods are an option to meet this requirement. One of them which can be used is Prediction Error Expansion. This method embeds secret data into expanded prediction error which is a difference between pixel for data embedding and pixel for prediction. To improve the quality of stego image, the Pixel-Value-Ordering concept is applied when embedding data. This approach is further developed by utilizing an expanded prediction error value of 0, so that it can increase the capacity of data embedding. This method, however, only uses smooth block and is not suitable for block pixels whose size is more than 2×2 . In this paper, we propose a new embedding concept that can utilize another block type (e.g. rough type). Furthermore, we also propose a 2 bit embedding concept with prevention, to keep the stego image quality still good. The experimental result shows the average of stego image quality increases by 4.6 dB with capacity 5-10 times bigger than that of the previous method.

Index Terms—data security, data hiding, information security, multimedia data.

I. INTRODUCTION

NOWADAYS, information technology has been able to transmit data in various forms of data, such as image, audio and video. This advantage can be utilized to support human activity in different fields. Nevertheless, its implementation in sensitive data (e.g. military, financial and medical) requires special treatment to manage information in terms of data security and reliability [1]. Reversible Data Hiding (RDH) has become one of preferred techniques to use. This is because RDH has an ability to embed data into a digital object and extract it without harming the original object [2].

RDH should embed data without changing the digital object being used (carrier), since changes of the carrier may attract attention of unauthorized users (attackers) to hack or retrieve the embedded data [3]. Therefore, the problem of minimizing changes of the carrier is one of the main focus in this technique. In further research, RDH is divided into several groups, such as lossless compression, difference expansion (DE) and histogram shifting (HS).

Lossless compression is a technique that uses least significant bits (LSB) to embed data in the compressed image which is used as a carrier [4], [5]. A relatively poor stego image quality (the result of embedding process) and low embedding capacity have made this technique less efficient to

use. A different concept from lossless compression is used by DE which embeds data within the difference between neighboring pixels. It is proposed by Tian [6] that the stego image quality depends on the magnitude of the difference. The more similar the value of neighboring pixels, the higher quality of the stego image [7]. This concept has been expanded to various types of cover data, such as image [8] and audio [3]. Differently, histogram shifting (HS) uses histogram which is a statistical component of the carrier image to embed data. Developed by Ni et al. [9], HS embeds data by employing the peak and lowest point of histogram. The capacity of the payload which can be embedded, however, relies on the number of these peak and lowest points of the histogram. HS is further developed in several methods such as integer-to-integer transformation [10], [11], [12] and prediction-error expansion (PEE) [13], [14], [15], [16].

PEE is firstly developed by Thodi and Rodriguez [17] by predicting the possible changing of pixel value and embed data into it. It gives a good result that is represented by a small noise of the stego image according to its peak signal to noise ratio (PSNR). Because of it, PEE is one of the most important methods in the RDH research. Then, Sachnev et al. [18] improve PEE by implementing a sorting process to calculate the prediction-error value. Other development is carried out by Li et al. [19] by using pixel selection to embed data adaptively and to insert more data on the smooth pixel blocks. In other research, Li et al. [20] use pixel-value-ordering to define a pixel as a prediction error value calculator and pixel that is used for data embedding. This concept utilizes the maximum and minimum pixel on each block. However, in this research, secret data are only embedded in pixels whose prediction-error is 1. So, some pixels are not used in the process. Peng et al. [21] ameliorate this weakness by proposing a new concept to calculate prediction-error value, so that the embedding capacity can be increased.

However, this method only employs smooth pixel blocks with the maximum and minimum values to embed data. This concept is less effective for pixel blocks whose size is more than 2×2 . For example, if the size of the carrier is 512×512 with 4×4 block size, then there are only 16,384 blocks. If each block uses only the largest (maximum) and smallest (minimum) pixel values for data embedding, its capacity is only 32,768 bits by assuming that all blocks are smooth.

Based on that problem, we propose a method by exploring rough blocks for embedding data which is done by grouping similar pixels. Furthermore, we also propose a concept of reduction error expansion (REE) for data embedding, that each pixel is able to embed 2 bits of data without generating a substantial noise to the stego image.

The rest of the paper is structured as follows. Section 2 explains the previous research. Section 3 presents the

Manuscript received March 20, 2018; revised July 20, 2018. This work was supported in part by Institut Teknologi Sepuluh Nopember, Indonesia.

Hendro E. Prabowo was with Institut Teknologi Sepuluh Nopember, and now is with Universitas Negeri Semarang, Indonesia.

Tohari Ahmad is with Department of Informatics, Institut Teknologi Sepuluh Nopember, Kampus ITS, Surabaya, Indonesia: corresponding author, phone: +62315939214, email: tohari@if.its.ac.id (<http://orcid.org/0000-0002-3390-0756>).

proposed method. Section 4 provides the experimental results and the conclusion is drawn in Section 5.

II. RELATED WORKS

In this section, we briefly discuss some previous research. These references have become a basic concept for developing our proposed data hiding method.

A. Pixel Value Ordering

Pixel Value Ordering (PVO) [20] is a reversible method which embeds secret data by dividing the carrier into non-overlapping blocks. Each block is sorted in ascending to obtain $(P_{\sigma(1)}, \dots, P_{\sigma(n)})$ with $n = r \times c$ is the block size and $P_{\sigma(n)}$ is the pixel in the block. In this method, every block only uses 2 pixels for embedding: $P_{\sigma(1)}$ and $P_{\sigma(n)}$. The embedding process in $P_{\sigma(n)}$ is as follows:

- 1) Calculate the expanded prediction error for $P_{\sigma(n)}$ using (1).

$$d_{max} = P_{\sigma(n)} - P_{\sigma(n-1)} \quad (1)$$

- 2) For data embedding, d_{max} is modified using (2) where $b \in \{1, 0\}$ is the payload to be embedded.

$$d'_{max} = \begin{cases} d_{max}, & \text{if } d_{max} = 0 \\ d_{max} + b, & \text{if } d_{max} = 1 \\ d_{max} + 1, & \text{if } d_{max} > 1 \end{cases} \quad (2)$$

- 3) The new pixel value $P'_{\sigma(n)}$ can be calculated using (3).

$$P'_{\sigma(n)} = P_{\sigma(n-1)} + d'_{max} = \begin{cases} P_{\sigma(n)}, & \text{if } d_{max} = 0 \\ P_{\sigma(n)} + b, & \text{if } d_{max} = 1 \\ P_{\sigma(n)} + 1, & \text{if } d_{max} > 1 \end{cases} \quad (3)$$

A similar concept is also used to embed data to $P_{\sigma(1)}$. From this process, we know that secret data will be embedded if the expanded prediction error is 1. Additionally, in every block, the maximum data can be embedded is 2 bits.

B. Pixel Value Ordering Improvement

In this method [21], Peng et al. intend to increase the probability of data which can be embedded into pixels. Based on that case, they propose (4) which is actually developed based on (1).

$$d_{max} = P_u - P_v \quad (4)$$

Here, $u = \min(\sigma(n), \sigma(n-1))$, $v = \max(\sigma(n), \sigma(n-1))$, $\sigma(n)$ is $P_{\sigma(n)}$'s index and $\sigma(n-1)$ is $P_{\sigma(n-1)}$'s index. While (2) for data embedding is changed to (5) and new pixel value can be obtained using (6).

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

$$P'_{\sigma(n)} = P_{\sigma(n-1)} + |d'_{max}| = \begin{cases} P_{\sigma(n)} + b, & \text{if } d_{max} = 0 \\ P_{\sigma(n)} + 1, & \text{if } d_{max} > 1 \\ P_{\sigma(n)} - b, & \text{if } d_{max} = 0 \\ P_{\sigma(n)} - 1, & \text{if } d_{max} < 0 \end{cases} \quad (6)$$

Similar to the previous research, $b \in \{1, 0\}$ is data to be embedded and $P'_{\sigma(n)}$ is the new pixel value. This modification is also implemented in minimum pixel ($P'_{\sigma(1)}$) which generates (7)-(9) with $s = \min(\sigma(1), \sigma(2))$, $t = \max(\sigma(1), \sigma(2))$ and $P'_{\sigma(1)}$ is the new pixel value.

$$d_{min} = P_s - P_t \quad (7)$$

$$d'_{min} = \begin{cases} d_{min} + b, & \text{if } d_{min} = 1 \\ d_{min} + 1, & \text{if } d_{min} > 1 \\ d_{min} - b, & \text{if } d_{min} = 0 \\ d_{min} - 1, & \text{if } d_{min} < 0 \end{cases} \quad (8)$$

$$P'_{\sigma(n)} = P_{\sigma(n-1)} + |d'_{min}| = \begin{cases} P_{\sigma(n)} + b, & \text{if } d_{min} = 0 \\ P_{\sigma(n)} + 1, & \text{if } d_{min} > 1 \\ P_{\sigma(n)} - b, & \text{if } d_{min} = 0 \\ P_{\sigma(n)} - 1, & \text{if } d_{min} < 0 \end{cases} \quad (9)$$

For extracting secret data and restoring pixel values on the maximum pixel ($P'_{\sigma(n)}$), Peng et al. [21] give following definition, where d'_{max} is calculated by using (4):

- If $d'_{max} > 0$, then we know that $P'_u > P'_v$. The probability of d'_{max} , embedded data and original value can be described as follows:
 - If $d'_{max} \in \{1, 2\}$, then the embedded data is $b = d'_{max} - 1$ and the original pixel value is $P_{\sigma(n)} = P'_u - b$;
 - If $d'_{max} > 2$, then there is no embedding and the original pixel value is $P_{\sigma(n)} = P'_u - 1$.
- If $d'_{max} \leq 0$, then we know that $P'_u \leq P'_v$. The probability value of d'_{max} , embedded data and original value can be obtained by using following conditions:
 - If $d'_{max} \in \{0, -1\}$, then the embedded data is $b = -d'_{max}$, and the original pixel value is $P_{\sigma(n)} = P'_v - b$;
 - If $d'_{max} < -1$, then there is no embedding data and the original pixel value can be restored by using $P_{\sigma(n)} = P'_v - 1$.

For the minimum pixel, we can use following conditions, where d'_{min} is obtained by using (7):

- If $d'_{min} > 0$, then we know that $P'_s > P'_t$. The probability of d'_{min} is:
 - If $d'_{min} > 0$, then the embedded data is $b = d'_{min} - 1$ and the original pixel value is $P_{\sigma(1)} = P'_t + b$;
 - If $d'_{min} > 2$, then there is no embedding data and the original pixel value can be achieved by using $P_{\sigma(1)} = P'_t + 1$
- If $d'_{min} \leq 0$, then we know that $P'_s \leq P'_t$ and the probability of d'_{min} is :

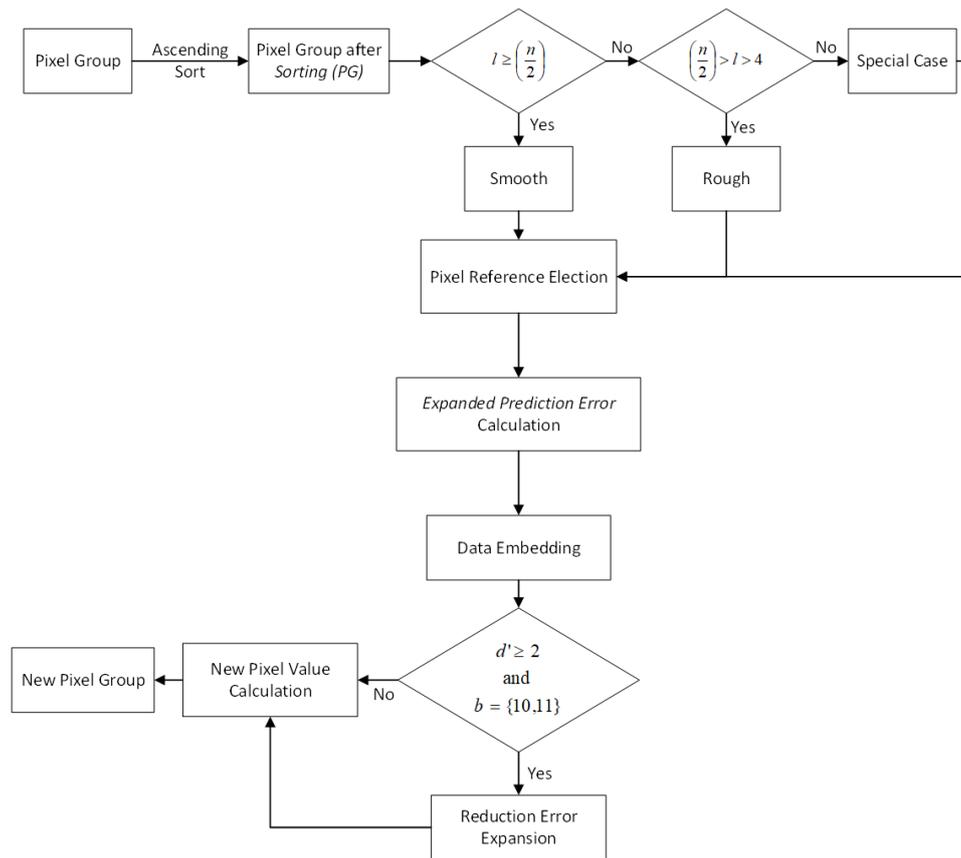


Fig. 1. Data embedding process

- If $d'_{min} \in \{0, -1\}$, then the embedded data is $b = -d'_{min}$ and the original pixel value is $P_{(1)} = P'_s + b$;
- If $d'_{min} < -1$, then there is no embedding data and the original pixel value is $P_{(1)} = P'_s + 1$.

Based on Peng et al. improvement [21], PVO is able to embed data into expanded prediction error whose value is either 1 or 0. This indirectly increases the embedding capacity from [20].

C. Adaptive Pixel Embedding Strategy on PVO

To increase the embedding capacity, Weng et al. [22] develop PVO by proposing a concept which is able to select pixels for both embedding and predicting based on the correlation value between a block pixel and its neighbors. The respective correlation level divides blocks to smooth and rough. In the smooth block, the correlation level comprises three groups: G_1 , G_2 and G_3 . Each group has an ability to carry various numbers of data. That is, G_1, G_2, G_3 can embed up to 6, 4, 2 bits of data, respectively. For data embedding process, G_1 uses $P_{\sigma(4)}$ and $P_{\sigma(n-3)}$ as the predictor in the minimum (i.e. $P_{\sigma(3)}, P_{\sigma(2)}, P_{\sigma(1)}$) and maximum pixels (i.e. $P_{\sigma(n-2)}, P_{\sigma(n-1)}, P_{\sigma(n)}$), respectively. In G_2 , Weng et al. [22] utilize $P_{\sigma(3)}$ and $P_{\sigma(n-2)}$; while in G_3 , they use $P_{\sigma(2)}$ and $P_{\sigma(n-1)}$ as the predictor pixel. In more details, G_3 is a block which is used by Peng et al. [21] and Li et al. [20].

III. PROPOSED METHOD

This proposed method is intended to extend that of Peng et al. [21] by improving Weng et al. concept [22], i.e., grouping

pixel blocks based on the correlation level. This grouping is performed to pixels in a block adaptively based on the similarity of the pixel value. Furthermore, we design pixel predictor selection to make it more effective than that of [21], especially in terms of the embedding capacity.

Here, we use grayscale image as carrier (I) whose size is $W \times H$. This is then divided into non-overlapping blocks (B_1, \dots, B_N). Every block B_k ($k \in \{1, \dots, N\}$) is grouped using the adaptive pixel value grouping concept. Pixels that belong to a group are used for data embedding; on the contrary, non-grouped pixels are not used for the embedding. Before starting the embedding process, the groups are sorted and classified based on their pixel group member (l) and the number of pixel in a block (n). Then, the predictor pixel is selected and expanded predictor error is calculated.

Embedding process itself is divided into 2 types: embedding with reduction error expansion and embedding without embedding error expansion. This type is determined by the expanded prediction error modification result (d') and data to be embedded (b). In general, the embedding process can be seen in Fig. 1.

A. Adaptive Pixel Block Grouping

This concept is developed based on our previous research on adaptive pixel value grouping (APVG), which is performed by grouping pixels with similar value [23]. Each pixel group consists of a pixel group head (PG_h) and a pixel group body (PG_b) or a pixel group tail (PG_t). In this research, we restrict APVG operation to grouping pixels only in a block. If it is given a block $r \times c$, and $l = n$

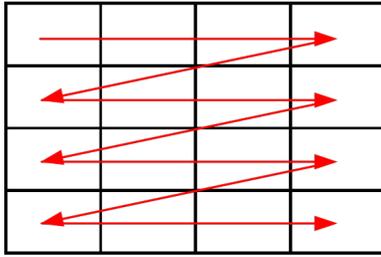


Fig. 2. The direction of pixel grouping process

as the maximum number of pixels in a pixel group (PG) with $n = r \times c$. The direction of grouping process is row by row as provided in Fig. 2. Then, every pixel group is sorted to obtain $(P_{\sigma(1)}, \dots, P_{\sigma(l)})$, before process of pixel group classification starts.

B. Pixel Group Classification

This classification aims to manage the quality of stego image remains good. The process divides pixel group into 3: smooth (PG_{smooth}), rough (PG_{rough}) and other (PG_{other}). This pixel group type is determined based on (10).

$$type = \begin{cases} smooth, & \text{if } l \geq \frac{n}{2} \\ rough, & \text{if } \frac{n}{2} > l > 4 \\ other, & \text{otherwise} \end{cases} \quad (10)$$

The result of classification affects the selection of the predictor pixel that used to calculate the expanded prediction error for every pixel in the pixel group.

C. Data Embedding

1) *Predictor Pixel Selection*: The predictor pixel is selected based on the pixel group type and the number of pixels in a group. Similar to [20] and [21], in some pixel group types, the predictor pixel is divided into 2: maximum predictor pixel and minimum predictor pixel. The minimum predictor pixel is used to calculate expanded prediction error (d_{min}) in the small pixel value. For the maximum predictor pixel, it is used to calculate expanded prediction error (d_{max}) in the large pixel value. To determine the index of predictor pixels for PG_{smooth} and PG_{rough} blocks, we employs (11) (12) and (13) (14), respectively; while index or predictor pixel in PG_{other} is determined using (15), (16). In addition, ref_{max} is index of maximum predictor pixel and ref_{min} is index of minimum predictor pixel.

$$ref_{max} = \begin{cases} \frac{l}{2} + 1, & \text{if } l \bmod 2 = 0 \\ \frac{l}{2}, & \text{if } l \bmod 2 = 1 \end{cases} \quad (11)$$

$$ref_{min} = \frac{l}{2} \quad (12)$$

$$ref_{max} = \begin{cases} \frac{l}{2} + 2, & \text{if } l \bmod 2 = 0 \\ \frac{l}{2} + 1, & \text{if } l \bmod 2 = 1 \end{cases} \quad (13)$$

$$ref_{min} = \frac{l}{2} - 1 \quad (14)$$

$$ref_{max} = \begin{cases} 3, & \text{if } l = 4 \\ 2, & \text{if } l = 3 \\ 1, & \text{if } l = 2 \end{cases} \quad (15)$$

$$ref_{min} = 2, \text{ if } l \in \{4, 3\} \quad (16)$$

An example of maximum and minimum predictor pixels for each PG is depicted in Fig. 3, where Fig. 3(a) and 3(b) are example of PG_{smooth} with l is even and odd number, respectively. For PG_{rough} , we can see in Fig. 3(c) as PG_{rough} with l is even number and Fig. 3(d) as PG_{rough} with l is odd number. While Fig. 3(e), 3(f) and 3(g) are examples of PG_{other} with l is 4, 3 and 2. In Fig. 3, light and dark gray in every pixel group represent the pixels that are selected as the predictor. The light one is the minimum predictor pixel ($P_{\sigma(ref_{min})}$) and the dark one is the maximum predictor pixel ($P_{\sigma(ref_{max})}$). It is shown that there are some pixel groups which only have one color pixel (e.g. Fig. 3(b) and 3(f)). This means that this pixel is used as $P_{\sigma(ref_{min})}$ and $P_{\sigma(ref_{max})}$ in the pixel group. Especially for Fig. 3(g) where PG_{other} has $l = 2$, we only utilize $P_{\sigma(ref_{max})}$ for the predictor pixel and the others for embedding data.

2) *Data Embedding and New Pixel Value Calculation*: The embedding data process is divided to some stages: expanded prediction error calculation, embedding data and reduction error expansion. Each stage is done twice to embed data in each part of the pixel group. We call this as maximum and minimum parts. The minimum part is pixels whose value is less than or equal to $P_{\sigma(ref_{min})}$. It is located on the left side of $P_{\sigma(ref_{min})}$ after sorting process. The maximum part is pixels whose value is greater than $P_{\sigma(ref_{max})}$ which is located on the right side of $P_{\sigma(ref_{max})}$. The explanation of maximum and minimum pixel parts in PG is presented in Fig. 4. In the case that PG_{rough} with $l = 2$, it only has 1 pixel for embedding (see Fig. 3(g)), so each stage is done only once, that is in maximum part. Pixels in PG which do not include in maximum or minimum parts are not used for embedding process and their values do not change.

Maximum-Modification

For each PG in P_k ($k \in 1, \dots, N$) that has maximum part, $P_{\sigma(ref_{max})}$ is used to predict the expanded prediction error value (d_{max}) in each pixel. It is done using (17), with (j) is pixel's index in the maximum part.

$$d_{max} = P_u - P_v; \begin{cases} u = \min(\sigma(j), \sigma(ref_{max})) \\ v = \max(\sigma(j), \sigma(ref_{max})) \end{cases} \quad (17)$$

The variable d_{max} is modified for embedding data using (5) where b is the data to be embedded whose value is $\{1, 2, 3, 4\}$, which is the conversion of $\{00, 01, 10, 11\}$. This value indicates that the data to be embedded is 2 bits in each pixel. To avoid d_{max} value changes too high, we propose a reduction function, called reduction error expansion (REE). This reduction process of d_{max} is done using (18).

$$d'_{max} = \begin{cases} d_{max} \bmod 2, & \text{if } |d_{max}| > 1, b = \{10, 11\} \\ d_{max}, & \text{otherwise} \end{cases} \quad (18)$$

Then, a new pixel value at maximum part denoted as $P_{\sigma(j)}$ which is calculated using (19) for d'_{max} which is not reduced. In the case d'_{max} is reduced, a new pixel value is obtained using (20).

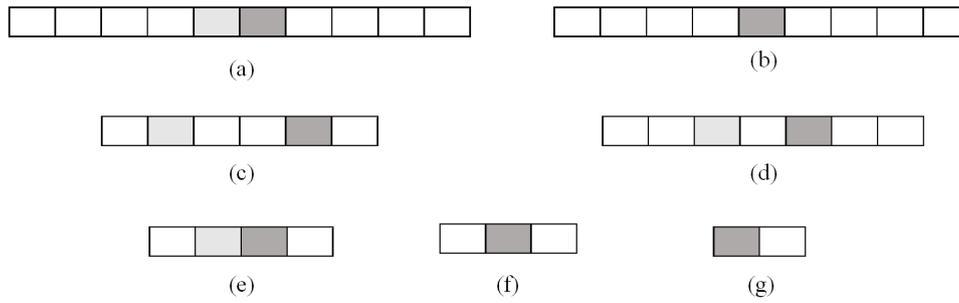


Fig. 3. Example of predictor pixel in every APVG type, where light and dark gray boxes represent minimum and maximum predictor pixels, respectively

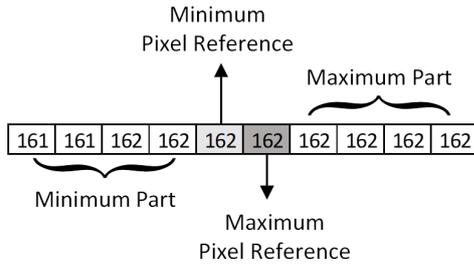


Fig. 4. Parts of a pixel group

of 4×4 . The PG is included in the smooth type and has two predictor pixels ($P_{\sigma(ref_{min})} = 162, P_{\sigma(ref_{max})} = 162$) with pixels index is 3 for $\sigma(ref_{min})$ and 5 for $\sigma(ref_{max})$. The number of data to embed is 16 bits which is divided into two groups, where each pixel get 2 bits of data. In Fig. 4, d'_{min} or d'_{max} which are represented by blue color pixels, is reduced firstly before new pixel value is calculated. The final result of embedding process shows that each pixel value should not be greater than $P_{\sigma(ref_{min})}$ for minimum part or less than $P_{\sigma(ref_{max})}$ for the maximum part. So, when the extraction process starts, the predictor pixel value does not change.

$$P'_{\sigma(j)} = P_{\sigma(ref_{max})} + |d''_{max}| \quad (19)$$

$$P'_{\sigma(j)} = P_{\sigma(j)} + |d''_{max}| \quad (20)$$

Minimum-Modification

The proposed improvement at the maximum part also implementable in the minimum one. Expanded prediction error value (d'_{min}) from minimum can be obtained using (21), where $\sigma(i)$ is the index of pixel in the minimum part which is used for data embedding.

$$d_{min} = P_s - P_t, \begin{cases} s = \min(\sigma(i), \sigma(ref_{min})) \\ t = \max(\sigma(i), \sigma(ref_{min})) \end{cases} \quad (21)$$

The data embedding can be done using (8) as in [21] with b as that being used in the maximum part. The REE process is also applied in it and can be done using (22):

$$d''_{min} = \begin{cases} d'_{min} \bmod 2, & \text{if } |d'_{min}| > 1, b = \{10, 11\} \\ d'_{min}, & \text{otherwise} \end{cases} \quad (22)$$

The value of d''_{min} is used to calculate the new pixel value ($P'_{\sigma(i)}$) using (23) for d''_{min} which is not the result of the reduction process and using (24) for d''_{min} which is the result of the reduction process.

$$P'_{\sigma(i)} = P_{\sigma(ref_{min})} + |d''_{min}| \quad (23)$$

$$P'_{\sigma(i)} = P_{\sigma(i)} + |d''_{min}| \quad (24)$$

An example of the data embedding process is provided in Fig. 5 with PG 's member is 10 pixels ($l = 10$) from block

D. Data Extraction and Pixel Restoration

Similar to the data embedding process, data extraction begins with dividing stego image into block whose size is $r \times c$, where its value is equal to block size during embedding data. Each block is developed based on the location map that previously we have obtained. Pixels in the block which are marked with PG_h, PG_b and PG_t are clustered together to form a pixel group. While pixels that are marked as non-pixel grouping (NPG) are not grouped. Each group is then classified and sorted. Next, the predictor pixel is selected. In general, all these steps have been explained in the previous sub-section. If each pixel group has obtained its predictor pixel, we can extract data and restore the original pixel value on stego image in these two following ways: pixels with REE and pixels without REE.

Pixel with REE

The data extraction process starts by calculating the value of d''_{max} and d''_{min} for each pixel in the group. This process can be done using (25) and (26) which is an improvement of [21].

$$d''_{max} = P'_u - P'_v, \begin{cases} u = \min(\sigma(j), \sigma(ref_{max})) \\ v = \max(\sigma(j), \sigma(ref_{max})) \end{cases} \quad (25)$$

$$d''_{min} = P'_s - P'_t, \begin{cases} s = \min(\sigma(i), \sigma(ref_{min})) \\ t = \max(\sigma(i), \sigma(ref_{min})) \end{cases} \quad (26)$$

The embedded data b are taken using (27), (28). This value is then converted into binary to get the original message value.

$$b = \begin{cases} 3, & \text{if } d''_{max} \in \{-1, 1\} \\ 2, & \text{if } d''_{max} \in \{0, 2\} \end{cases} \quad (27)$$

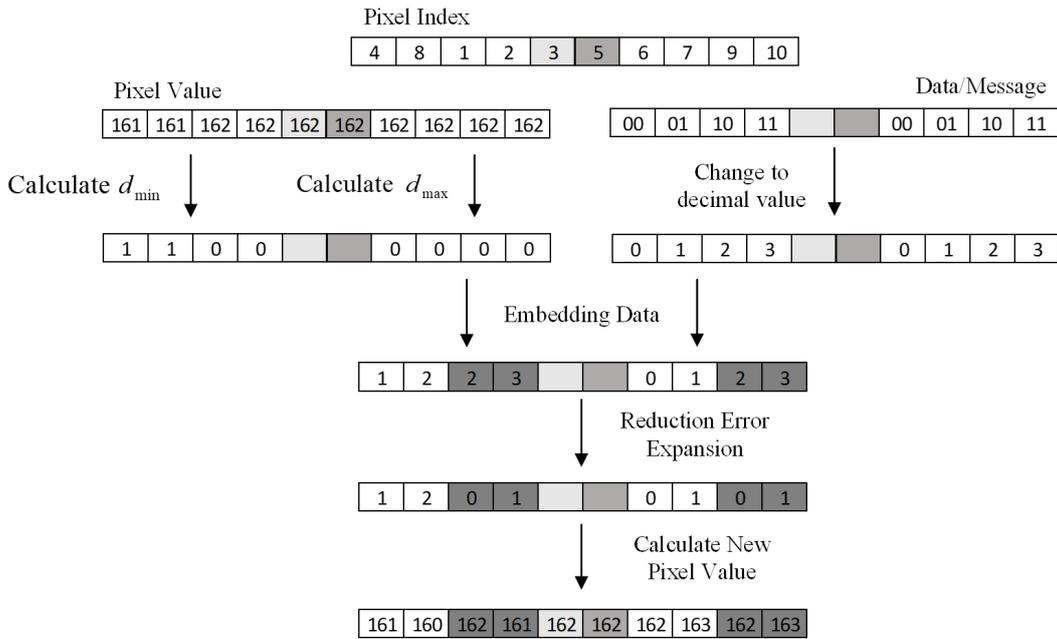


Fig. 5. Example of embedding data, where light and dark gray represent minimum and maximum predictor pixels, very dark gray and white represent pixels with and without REE, respectively

$$b = \begin{cases} 3, & \text{if } d''_{min} \in \{-1, 1\} \\ 2, & \text{if } d''_{min} \in \{0, 2\} \end{cases} \quad (28)$$

In order to obtain the original pixel value, d'_{max} and d'_{min} are calculated using (29) and (30). These are required for restoring the original pixel values using (31) and (32) for maximum and minimum parts, respectively.

$$d'_{max} = \begin{cases} d''_{max} - 1, & \text{if } d''_{max} > 0 \\ -d''_{max}, & \text{if } d''_{max} \leq 0 \end{cases} \quad (29)$$

$$d'_{min} = \begin{cases} d''_{min} - 1, & \text{if } d''_{min} > 0 \\ -d''_{min}, & \text{if } d''_{min} \leq 0 \end{cases} \quad (30)$$

$$P_{(j)} = P'_{(j)} - d'_{max} \quad (31)$$

$$P_{(i)} = P'_{(i)} + d'_{min} \quad (32)$$

Pixel without REE

Pixels without reduction error expansion (REE) are those generated from d_{max} or d_{min} without the reduction process. Therefore, extraction process proposed by Peng et al. [21] can be implemented. The resulted data are converted into binary to obtain the original data value. For restoring original pixels, we can use (33) and (34) to improve that of Peng et al. [21].

$$P_{(j)} = \begin{cases} P'_u - b, & \text{if } d''_{max} > 0 \text{ and } d''_{max} \in \{1, 2\} \\ P'_u - 1, & \text{if } d''_{max} > 2 \\ P'_v - b, & \text{if } d''_{max} \leq 0 \text{ and } d''_{max} \in \{-1, 0\} \\ P'_v - 1, & \text{if } d''_{max} < -1 \end{cases} \quad (33)$$

$$P_{(i)} = \begin{cases} P'_t + b, & \text{if } d''_{min} > 0 \text{ and } d''_{min} \in \{1, 2\} \\ P'_t + 1, & \text{if } d''_{min} > 2 \\ P'_s + b, & \text{if } d''_{min} \leq 0 \text{ and } d''_{min} \in \{-1, 0\} \\ P'_s + 1, & \text{if } d''_{min} < -1 \end{cases} \quad (34)$$

An example of extraction process of pixel groups in a stego image is presented in Fig. 6. Here, light and dark gray depict the selected predictor pixels; while very dark gray represents a pixel of the stego image resulted from REE, and the white pixels are stego image pixel without REE process. The final extraction result shows that the payload data can be fully extracted and the original pixel values are restored according to their pixel index.

E. Location Map (LM)

Location Map (LM) in this research is used as a marker for pixel group formation, overflow/underflow values and pixels which are obtained from the REE process as shown in Table I. In this case, each marker has two possible values: 1 and 0. This LM is not embedded to the stego image but it is transmitted separately. It aims to keep the quality of stego image always good as we have proved in [24].

In general, our research can be further depicted in Table II, which also provides the difference between existing methods and ours.

IV. RESULTS AND DISCUSSION

In this section, the proposed method is evaluated by comparing it with the reference. For this purpose, we use 512×512 gray scale images taken from [25] and medical image obtained from [26]. The secret payload is binary generated from Matlab *rand()* function. The example of testing images can be found in Fig. 7, which shows 'Baboon' image before embedding, after embedding and after extraction. From the

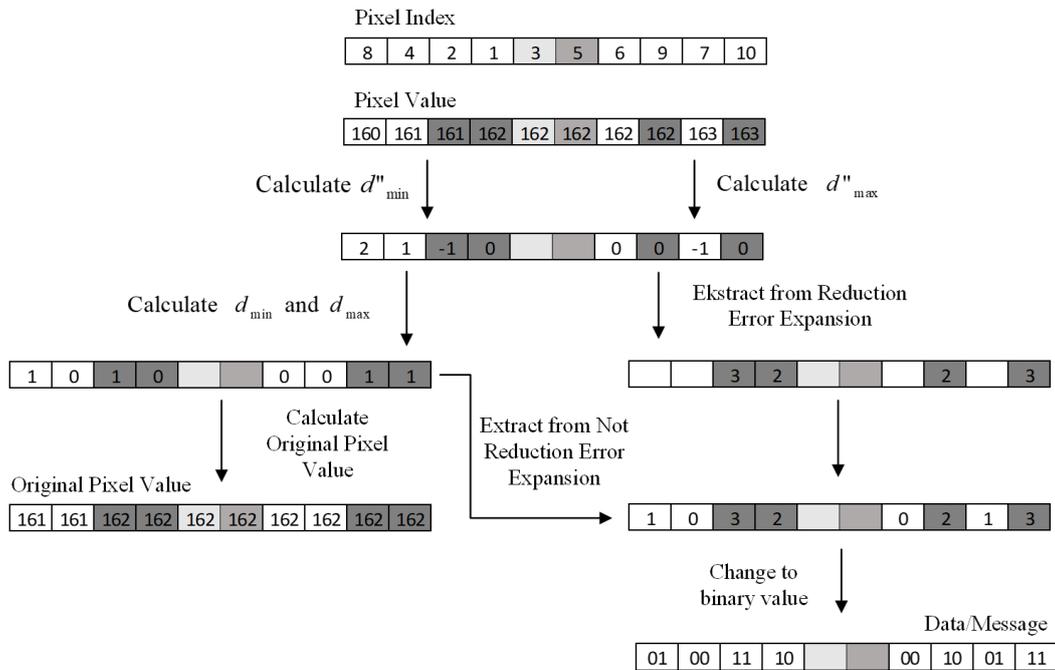


Fig. 6. Example of extraction process, where light and dark gray represent minimum and maximum predictor pixels, very dark gray and white represent pixels of stego image with and without REE, respectively

TABLE I
LOCATION MAP VALUE

LM Value	Pixel Grouping		Overflow/Underflow		Reduction Error Expansion	
	PG_h, PG_b	PG_t, NPG	Yes	No	Yes	No
	1	0	1	0	1	0

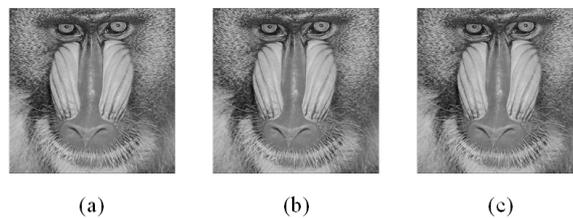


Fig. 7. An example image before embedding, after embedding and after extraction. (a) Cover image taken from [25], (b) Stego image, (c) Reconstructed image

experiment, we find that subjectively the stego images look very similar to the respective cover.

A. Capacity and Quality of Stego Image

We use two parameters (i.e. quality and capacity) to compare the capabilities of proposed and reference methods. In this case, the quality of stego image is measured using peak signal to noise ratio (PSNR), where higher PSNR value means better quality of the stego image. As for capacity measurement, we embed the maximum size can be accommodated by the cover image.

Tables III - X show the comparison of the stego image quality between the proposed method and Peng et al. [21]. In this experiment, we use blocks of 4×4 with 1Kb, 5Kb, 10Kb, 20Kb, 30Kb, 40kb, 50Kb and 100Kb of data, respectively. The PSNR value depicted in those tables is the highest one, obtaining from some possible threshold/embedding levels (1-254). It also presents that the

proposed method provides better stego image quality than [21] for all carrier images. In average, the quality of stego image increases approximately 4.6 dB for those payload size. This is caused by the reduce of the number of pixels used for hiding the secret. The most prominent change in PSNR values is represented by 'Baboon' whose pixel values are more varied (rough images) than others, which is 7.63 dB and 10.84 dB for 1Kb and 5Kb data, respectively. Interestingly, by using [21], 'Baboon' can only accommodate less than 10Kb of data, so it fails when 10Kb of data are employed (see Table V). This PSNR increase is provided in Fig. 8.

In general, Tables III - X also depict that by rising the capacity, the PSNR goes down. This condition applies for both [21] and the proposed method. On the contrary, the number of embed-able pixels grows according to the respective capacity, i.e. a half and full of the capacity for the proposed method and [21], respectively. For example, the proposed method and [21] explore correspondingly 500

TABLE II
COMPARISON OF GENERAL CHARACTERISTICS BETWEEN SOME EXISTING AND THE PROPOSED METHODS

Stage	Li et al. [20]	Peng et al. [21]	Weng et al. [22]	Proposed Method
Type of blocks	smooth, rough	smooth, rough	high, moderate, low correlation	smooth, rough, other
Capacity per blocks	smooth: 2 bits rough: not used	smooth: 2 bits rough: not used	high: 6 bits moderate: 4 bits low: 2 bits	adaptive, depending on the number of pixels in the group
Payload	1 bit per pixel: {0,1}	1 bit per pixel: {0,1}	1 bit per pixel: {0,1}	2 bits per pixel: {11,10,01,00}
Expanded diff. error (max)	$d_{max} = P_{\sigma(n)} - P_{\sigma(n-1)}$	$d_{max} = P_u - P_v$ $u = \min(\sigma(n), \sigma(n-1))$ $v = \max(\sigma(n), \sigma(n-1))$	$d_{max} = P_u - P_v$ $u = \min(\sigma(j), \sigma(ref))$ $v = \max(\sigma(j), \sigma(ref))$	$d_{max} = P_u - P_v$ $u = \min(\sigma(j), \sigma(ref))$ $v = \min(\sigma(j), \sigma(ref))$
Expanded diff. error (min)	$d_{min} = P_{\sigma(1)} - P_{\sigma(2)}$	$d_{min} = P_s - P_t$ $s = \min(\sigma(1), \sigma(2))$ $t = \max(\sigma(1), \sigma(2))$	$d_{min} = P_s - P_t$ $s = \min(\sigma(i), \sigma(ref))$ $t = \max(\sigma(i), \sigma(ref))$	$d_{min} = P_s - P_t$ $s = \min(\sigma(i), \sigma(ref))$ $t = \max(\sigma(i), \sigma(ref))$

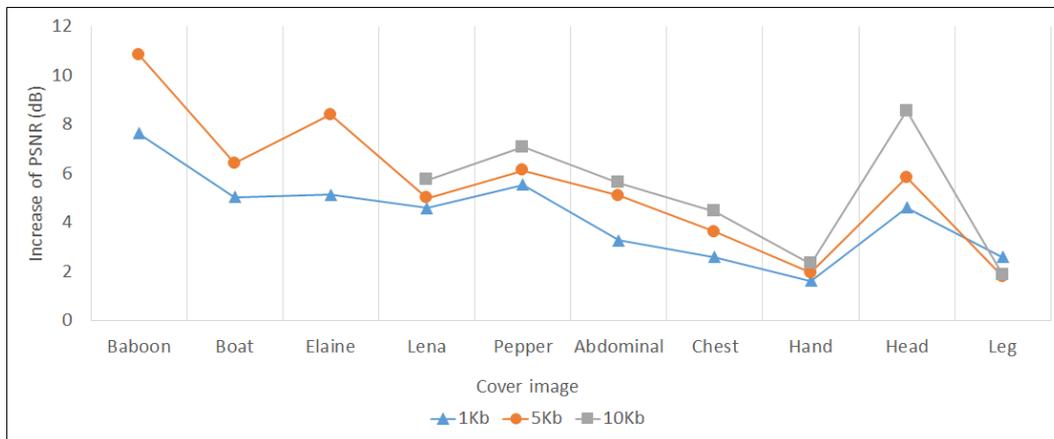


Fig. 8. The increase of PSNR by implementing [21] and the proposed method for each cover image

TABLE III
QUALITY OF THE STEGO IMAGES OF THE PROPOSED AND PENG ET AL. METHODS [21] USING 1 KB DATA

Image	PSNR (dB)		Embedded Pixel		Shifted Pixel	
	Prop.	[21]	Prop.	[21]	Prop.	[21]
Baboon	75.56	67.93	500	1000	246	2255
Boat	76.04	71.01	500	1000	185	864
Elaine	77.49	72.36	500	1000	59	503
Lena	76.15	71.57	500	1000	169	699
Pepper	75.82	70.29	500	1000	206	1107
Abdominal	78.39	75.13	500	1000	93	35
Chest	77.62	75.03	500	1000	111	47
Hand	76.99	75.37	500	1000	102	7
Head	79.37	74.76	500	1000	69	82
Leg	77.97	75.38	500	1000	34	6

TABLE IV
QUALITY OF THE STEGO IMAGES OF THE PROPOSED AND PENG ET AL. METHODS [21] USING 5 KB DATA

Image	PSNR (dB)		Embedded Pixel		Shifted Pixel	
	Prop.	[21]	Prop.	[21]	Prop.	[21]
Baboon	68.42	57.58	2500	5000	1212	25620
Boat	68.85	62.43	2500	5000	939	7200
Elaine	69.85	61.46	2500	5000	475	9643
Lena	69.02	64.03	2500	5000	869	4200
Pepper	68.8	62.67	2500	5000	1003	6666
Abdominal	72.19	67.09	2500	5000	274	789
Chest	71.26	67.65	2500	5000	313	380
Hand	70.17	68.23	2500	5000	371	17
Head	72.36	66.52	2500	5000	285	1254
Leg	70.00	68.21	2500	5000	435	32

and 1000 pixels for 1Kb, 2500 and 5000 pixels for 5Kb, 5000 and 10000 pixels for 10Kb. It is worth noting that this condition applies for the smaller payload size (i.e., less than 10Kb); while for the bigger payload, [21] is not able to hold it at all, as shown in Tables VII - X.

The results of next evaluation is provided in Table XI

which shows the capacity comparison between the proposed method and [21]. It can be inferred that the highest capacity is achieved with block size of 4×4 by [21]. Using the same threshold/embedding level, the proposed method provides a higher capacity, which is able to increase 5-10 times from those of [21]. This happens because, in the proposed method,

TABLE V
QUALITY OF THE STEGO IMAGES OF THE PROPOSED AND PENG ET AL. METHODS [21] USING 10 KB DATA

Image	PSNR (dB)		Embedded Pixel		Shifted Pixel	
	Prop.	[21]	Prop.	[21]	Prop.	[21]
Baboon	65.55	-	5000	-	2236	-
Boat	65.79	-	5000	-	1964	-
Elaine	66.43	-	5000	-	1334	-
Lena	65.98	60.25	5000	10000	1773	11027
Pepper	65.84	58.77	5000	10000	1957	17575
Abdominal	68.66	63.05	5000	10000	691	3381
Chest	68.22	63.77	5000	10000	707	2089
Hand	67.54	65.23	5000	10000	475	49
Head	60.92	60.92	5000	10000	495	9294
Leg	67.09	65.22	5000	10000	828	67

TABLE VI
QUALITY OF THE STEGO IMAGES OF THE PROPOSED AND PENG ET AL. METHODS [21] USING 20 KB DATA

Image	PSNR (dB)		Embedded Pixel		Shifted Pixel	
	Prop.	[21]	Prop.	[21]	Prop.	[21]
Baboon	62.55	-	20000	-	4447	-
Boat	62.8	-	20000	-	3966	-
Elaine	63.09	-	20000	-	3307	-
Lena	62.98	-	20000	-	3623	-
Pepper	62.82	-	20000	-	3968	-
Abdominal	65.76	-	20000	-	1281	-
Chest	65.58	-	20000	-	1177	-
Hand	64.64	62.09	20000	20000	878	532
Head	66.11	-	20000	-	1112	-
Leg	64.27	61.95	20000	20000	1409	877

TABLE VII
QUALITY OF THE STEGO IMAGES OF THE PROPOSED AND PENG ET AL. METHODS [21] USING 30 KB DATA

Image	PSNR (dB)		Embedded Pixel		Shifted Pixel	
	Prop.	[21]	Prop.	[21]	Prop.	[21]
Baboon	-	-	-	-	-	-
Boat	61.1	-	15000	-	5808	-
Elaine	61.23	-	15000	-	5428	-
Lena	61.21	-	15000	-	5467	-
Pepper	61.05	-	15000	-	5951	-
Abdominal	64.15	-	15000	-	1797	-
Chest	64.11	-	15000	-	1655	-
Hand	63.08	-	15000	-	1079	-
Head	64.35	-	15000	-	1645	-
Leg	62.69	-	15000	-	1800	-

each pixel holds 2 bits of data. This means that with the same number of pixels, the proposed method is able to increase at least two times of capacity than [21].

B. Block Size and Embedding Level

In further evaluation, we increase the block size and embedding level to determine their effect on stego image quality, whose results is provided in Fig. 9. In this case, the embedding level and the size of experimental data are 1 and 10 Kb, respectively.

It is shown that with block size of 5×5, some grayscale

TABLE VIII
QUALITY OF THE STEGO IMAGES OF THE PROPOSED AND PENG ET AL. METHODS [21] USING 40 KB DATA

Image	PSNR (dB)		Embedded Pixel		Shifted Pixel	
	Prop.	[21]	Prop.	[21]	Prop.	[21]
Baboon	-	-	-	-	-	-
Boat	59.87	-	20000	-	7451	-
Elaine	59.9	-	20000	-	7445	-
Lena	59.96	-	20000	-	7256	-
Pepper	59.79	-	20000	-	7853	-
Abdominal	62.90	-	20000	-	2301	-
Chest	62.85	-	20000	-	2294	-
Hand	61.82	-	20000	-	1208	-
Head	63.04	-	20000	-	2186	-
Leg	61.43	-	20000	-	2327	-

TABLE IX
QUALITY OF THE STEGO IMAGES OF THE PROPOSED AND PENG ET AL. METHODS [21] USING 50 KB DATA

Image	PSNR (dB)		Embedded Pixel		Shifted Pixel	
	Prop.	[21]	Prop.	[21]	Prop.	[21]
Baboon	-	-	-	-	-	-
Boat	58.87	-	25000	-	9566	-
Elaine	58.9	-	25000	-	9424	-
Lena	58.99	-	25000	-	9053	-
Pepper	58.8	-	25000	-	9869	-
Abdominal	61.92	-	25000	-	2827	-
Chest	61.87	-	25000	-	2853	-
Hand	60.88	-	25000	-	1320	-
Head	62.00	-	25000	-	2770	-
Leg	60.48	-	25000	-	2824	-

TABLE X
QUALITY OF THE STEGO IMAGES OF THE PROPOSED AND PENG ET AL. METHODS [21] USING 100 KB DATA

Image	PSNR (dB)		Embedded Pixel		Shifted Pixel	
	Prop.	[21]	Prop.	[21]	Prop.	[21]
Baboon	-	-	-	-	-	-
Boat	-	-	-	-	-	-
Elaine	-	-	-	-	-	-
Lena	-	-	-	-	-	-
Pepper	-	-	-	-	-	-
Abdominal	58.89	-	50000	-	5186	-
Chest	58.97	-	50000	-	5751	-
Hand	57.93	-	50000	-	2497	-
Head	59.08	-	50000	-	5228	-
Leg	57.53	-	50000	-	5340	-

images generate the best stego image quality, except 'Boat' image. Differently, the best stego image for medical is achieved by various block sizes. For example, the 'Head' image with block size 5×5 generates the best stego image, but with 10×10 the quality goes down; on the contrary, 'Abdominal' image provides the best quality with 10×10.

The quality of stego images is influenced by the number of shifted pixels (pixels whose expanded prediction error (d_{max}/d_{min}) is more than 1 or less than 0) on a certain block size. For example, as shown in Table XII, 'Lena' image whose block size is 5×5 gives better quality of stego image

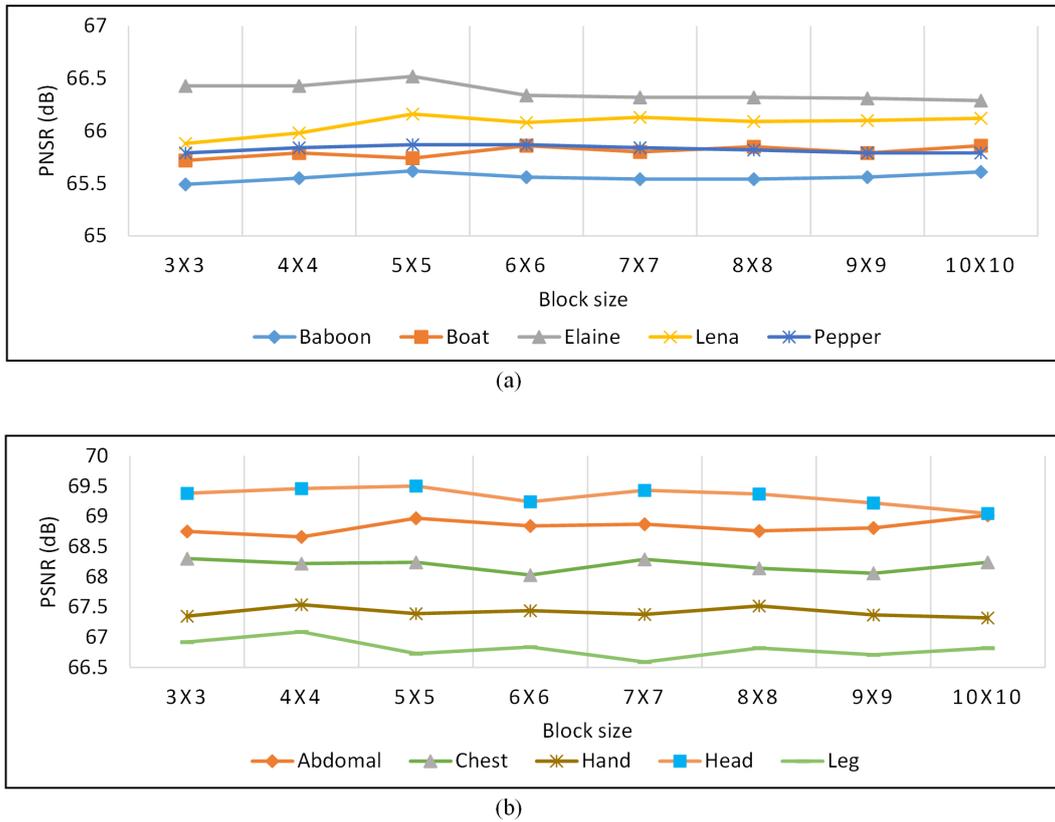


Fig. 9. The Effect of block size on the stego image quality where the embedding level is 1 and the payload size is 10 Kb, (a) Grayscale image, (b) Medical Image

TABLE XI
THE CAPACITY OF THE STEGO IMAGES OF THE PROPOSED METHOD AND THE PENG ET AL. METHOD [21]

Image	Capacity (bits)		Threshold/Embedding Level
	Prop.	[21]	
Baboon	31400	5300	195
Boat	65158	9582	204
Elaine	65518	8946	145
Lena	95692	12762	184
Pepper	81974	11035	193
Abdominal	235802	14121	103
Chest	289170	13723	127
Hand	299622	26519	125
Head	259938	8877	148
Leg	281190	26836	146

TABLE XII
THE EFFECT OF THE BLOCK SIZE ON THE QUALITY AND THE NUMBER OF SHIFTED PIXELS IN 'LENA' IMAGE

Block Size	PSNR (dB)	Shifted Pixel
3x3	65.88	1863
4x4	65.98	1773
5x5	66.16	1642
6x6	66.08	1681
7x7	66.13	1639
8x8	66.09	1681
9x9	66.1	1632
10x10	66.12	1642

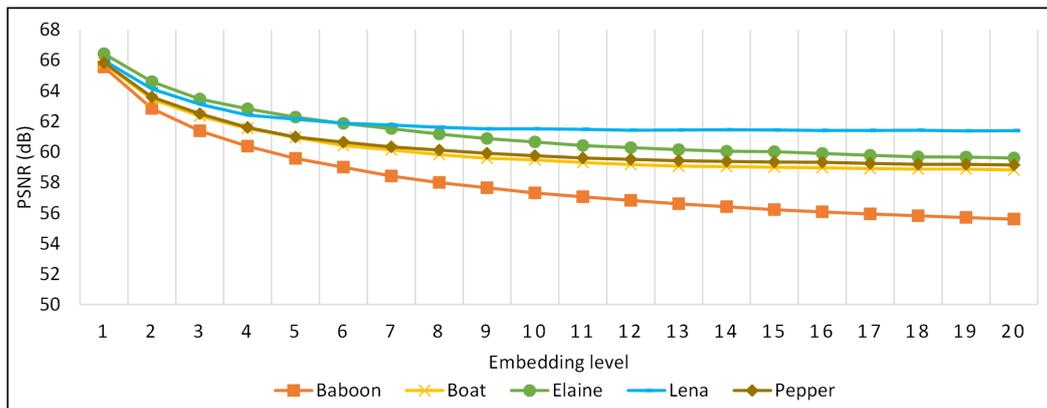
TABLE XIII
THE EFFECT OF THE BLOCK SIZE ON THE QUALITY AND THE NUMBER OF SHIFTED PIXELS IN 'ABDOMINAL' IMAGE

Block Size	PSNR (dB)	Shifted Pixel
3x3	68.75	658
4x4	68.66	691
5x5	68.97	644
6x6	68.84	664
7x7	68.87	665
8x8	68.76	673
9x9	68.81	625
10x10	69.02	644

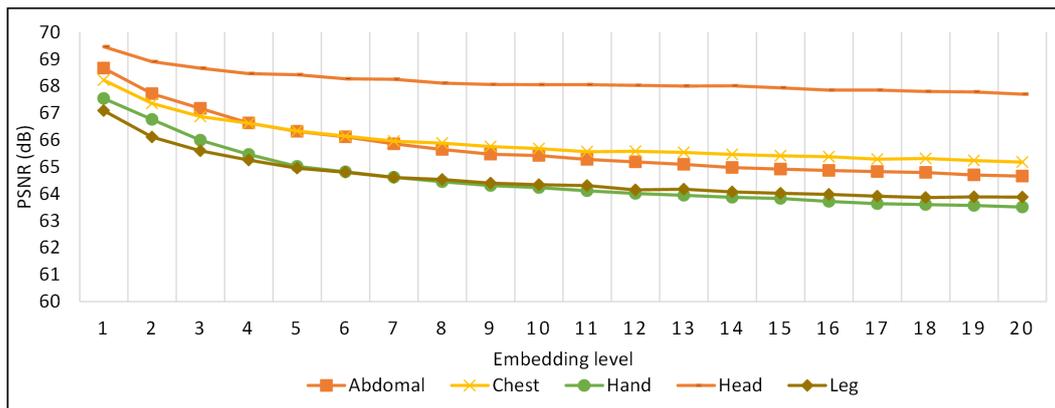
than 6x6. This is because the number of shifted pixels in a 6x6 block is higher than that in 5x5. Similarly, this also applies for 'Abdominal' image as depicted in Table XIII, where 10x10 generates better stego image than almost all others.

In the next evaluation, the effect of the embedding level on the stego image is measured whose result is presented in Fig. 10. The amount of data used in this experiment is 10 Kb with 4x4 block size. It is found that the quality of both stego image groups (i.e. grayscale and medical images) decreases along with the increasing of the embedding level. Furthermore, 'Baboon' image sharply declines of around 5 - 10 dB, more than other images. This happens especially when the embedding level is below 8. Slightly different,

the decrease in medical images is more gradual. Moreover, 'Head' image has the lowest decrease of about 2 dB. It is stabilized at around 68 dB for the embedding level of more than 7. The other medical images, however, drop about 3



(a)



(b)

Fig. 10. The Effect of embedding level on the stego image quality with block size is 4×4 , payload size is 10 Kb, (a) Grayscale Image, (b) Medical image

TABLE XIV

THE EFFECT OF THE EMBEDDING LEVEL ON THE QUALITY AND THE NUMBER OF SHIFTED PIXELS IN 'LENA' IMAGE

Embedding Level	PSNR (dB)	Shifted Pixel
1	65.98	1773
2	64.12	4077
3	63.12	5772
4	62.4	7259
5	62.14	7915
6	61.88	8516
7	61.76	8815
8	61.61	9217
9	61.51	9495
10	61.51	9535
11	61.47	9651
12	61.42	9740
13	61.43	9712
14	61.44	9725
15	61.43	9759
16	61.4	9792
17	61.4	9812
18	61.41	9825
19	61.37	9856
20	61.38	9848

TABLE XV

THE EFFECT OF THE EMBEDDING LEVEL ON THE QUALITY AND THE NUMBER OF SHIFTED PIXELS IN 'ABDOMINAL' IMAGE

Embedding Level	PSNR (dB)	Shifted Pixel
1	68.66	691
2	67.72	1401
3	67.17	1841
4	66.63	2252
5	66.32	2577
6	66.12	2777
7	65.86	3029
8	65.65	3233
9	65.47	3406
10	65.42	3533
11	65.28	3639
12	65.19	3772
13	65.09	3891
14	64.98	3985
15	64.92	4100
16	64.87	4166
17	64.83	4237
18	64.79	4313
19	64.70	4408
20	64.66	4466

dB for 1 - 8 of embedding level and gently fall for higher levels. This indicates that higher embedding level means more pixels which join in a group. Also, rising the number of PG's increases the number of shifted pixels.

An example of the effect of the number of shifted pixels is depicted in Tables XIV and XV. It takes 'Lena' and 'Abdominal' images as the case. In general, the increase of the embedding level rises the number of shifted pixels which leads to reducing the PSNR value with small fluctuation. Additionally, from 1 to 5 of the embedding level, the changes are sharper than the next level values. It can also be inferred that the larger block size does not necessarily generate a better stego image. It is different from the effect of the embedding level.

Next, shifted pixels are not used for data embedding but their value are shifted by 1. In the case that the change of block size reduces the number of shifted pixels, the quality of stego image is better. This condition corresponds to [20] that fewer number of shifted pixels leads to a higher stego image quality.

V. CONCLUSION

This paper has proposed a data hiding method by designing a new embedding concept which explores a possible block type. Additionally, we also propose a 2 bit embedding concept with prevention, to maintain the quality of the stego image.

It has been depicted that the proposed method is able to generate better stego image quality with same block size and threshold/embedding level value. This is because the method is able to manage the pixels by the APVG approach. Similar pixels are grouping according to their embedding level for keeping the number of shifted pixels small.

In terms of capacity, the proposed method has at least two times bigger than the reference. It is because each pixel is able to carry 2 bits of data while the previous method is only 1 bit. Furthermore, embedding 2 bits in a pixel does not affect the stego image quality significantly.

REFERENCES

- [1] W. R. Simpson, and K. E. Foltz, "Secure Identity for Enterprises," *IAENG International Journal of Computer Science*, vol. 45, no.1, pp. 142-152, 2018.
- [2] A. Khan, A. Siddiqa, S. Munib and S. A. Malik, "A Recent Survey for Reversible Watermarking Technique," *Information Science*, vol. 279, no. 20, pp. 251-272, 2014.
- [3] M. B. Andra, T. Ahmad, and T. Usagawa "Medical Record Protection with Improved GRDE Data Hiding Method on Audio Files," *Engineering Letters*, vol. 25, no. 2, pp. 112-124, 2017.
- [4] J. Fridrich, M. Goljan, and R. Du, "Lossless Data Embedding-New Paradigm in Digital Watermarking," *EURASIP Journal on Applied Signal Processing*, vol. 2, pp. 185-196, 2002.
- [5] M. U. Celik, G. Sharma, A. M. Tekalp and E. Saber, "Lossless Generalized-LSB Data Embedding," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253-266, 2005.
- [6] J. Tian, "Reversible Data Embedding Using A Difference Expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, 2003.
- [7] B. Ou, X. Li, J. Wang and F. Pang, "High-fidelity Reversible Data Hiding Based on Geodesic Path and Pairwise Prediction-Error Expansion," *Neurocomputing*, vol. 226, pp. 23-34, 2017.
- [8] P. Maniriho, and T. Ahmad "Enhancing the Capability of Data Hiding Method Based on Reduced Difference Expansion," *Engineering Letters*, vol. 26, no. 1, pp. 45-55, 2018.
- [9] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible Data Hiding," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, 2006.
- [10] D. Coltuc, "Low Distortion Transformation for Reversible Watermarking," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 412-417, 2012.
- [11] F. Peng, X. Li and B. Yang, "Adaptiave Reversible Data Hiding Scheme Based on Integer Transform," *Signal Processing*, vol. 92, no. 1, pp. 54-62, 2012.
- [12] Y. Qiu, Z. Qian and L. Yu, "Adaptive Reversible Data Hiding by Extending Generalized Integer Transformation," *IEEE Signal Processing Letters*, vol. 23, no. 1, pp. 130-134, 2016.
- [13] J. Wang, J. Ni and Y. Hu, "An efficient reversible data hiding scheme using prediction and optimal side information selection," *Journal of Visual Communication and Image Representation*, vol. 25, no. 6, pp. 1425-1431, 2014.
- [14] I. -C. Dragoi and D. Coltuc, "Local-Prediction-Based Difference Expansion Reversible Watermarking," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1779-1790, 2014.
- [15] Z. Wang, J. Ding and Q. Pei, "A Novel Reversible Image Data Hiding Scheme Based on Pixel Value Ordering and Dynamic Pixel Block Partition," *Information Sciences*, vol. 310, pp. 16-35, 2015.
- [16] Y. Liu, L. Chen, M. Hu, Z. Jia, S. Jia and H. Zhao, "A reversible data hiding method for h.246 with shamir's (t,n)-threshold secret sharing," *Neurocomputing*, vol. 188, pp. 63-70, 2016.
- [17] D. M. Thodi and J. J. Rodriguez, "Expansion Embedding Technique for Reversible Watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721-730, 2007.
- [18] V. Sachnev, H. J. Kim, J. Nam, S. Suresh and Y. Q. Shi, "Reversible Watermarking Algorithm Using Sorting and Prediction," *IEEE Transactions on Circuits and System for Video Technology*, vol. 19, no. 7, pp. 989-999, 2009.
- [19] X. Li, B. Yang and Y. Zeng, "Efficient Reversible Watermarking Based on Adaptive Prediction-Error Expansion and Pixel Selection," *IEEE Transaction on Image Processing*, vol. 20, no. 12, pp. 3524-3533, 2011.
- [20] X. Li, J. Li, B. Li and B. Yang, "High-Fidelity Reversible Data Hiding Scheme Based on Pixel-Value-Ordering and Prediction-Error-Expansion," *Signal Processing*, vol. 93, no. 1, pp. 198-205, 2013.
- [21] F. Peng, X. Li and B. Yang, "Improved PVO-based Reversible Data Hiding," *Digital Signal Processing*, vol. 25, pp. 255-256, 2014.
- [22] S. Weng, J. S. Pan and L. Li, "Reversible data hiding based on an adaptive pixel-embedding strategy and two-layer embedding," *Information Sciences*, vol. 369, pp. 144-159, 2016.
- [23] H. E. Prabowo and T. Ahmad, "Adaptive Pixel Value Grouping for Protecting Secret Data in Public Computer Network," *Journal of Communications*, vol. 13, no. 6, 2018.
- [24] M. H. A. Al Huti, T. Ahmad and S. Djanali, "Increasing the capacity of the secret data using DE pixels block and adjusted RDE-based on Grayscale Image," in *International Conference on Information and Communication Technology and Systems (ICTS)*, pp. 225-230, 2015.
- [25] USC-SIPI, "The USC-SIPI Image Database," 1977. [Online]. Available: <http://sipi.usc.edu/database>.
- [26] S. P.H. Inc., "Partners Infectious Disaese Images eMicrobes Digital Library," 2002. [Online]. Available: <http://www.idimages.org/>.