

## Protecting Data by Improving Quality of Stego Image based on Enhanced Reduced difference Expansion

Maurice Ntahobari<sup>1</sup>, Tohari Ahmad<sup>2</sup>

<sup>1,2</sup>Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia

<sup>1</sup>Groupe Scolaire de Gishubi, Gisagara District, Southern Province, Rwanda

---

### Article Info

#### Article history:

Received Jan 31, 2018

Revised Apr 2, 2018

Accepted Apr 20, 2018

---

#### Keyword:

Data hiding

Data protection

Information security

Secret data

Visual quality

---

### ABSTRACT

In this era of internet development, security of information sharing is the main problem faced by human being. Data hiding technique is one of the solutions. However, hiding credential information within a multimedia file such as image reduces its visual quality. Therefore, unauthorized users may suspect the existence of secret data within that image. In the past years, various data hiding algorithms have been developed by researchers to overcome the problem of high distortion of image after data embedding process. Achieving a high quality stego image, however, is still a challenging problem. In this paper, we proposed a new data hiding algorithm based on different expansion. It aims to enhance the quality of stego image for a given payload size. The new algorithm is evaluated on various medical images. Thereafter, the experimental results show that the visual quality is improved; and increasing the embedding capacity leads to more noises. Therefore, a better choice of base point and a reduced difference expansion affect the quality of stego image.

*Copyright © 2018 Institute of Advanced Engineering and Science.*

*All rights reserved.*

---

### Corresponding Author:

Maurice Ntahobari,

Department of Informatics,

Institut Teknologi Sepuluh Nopember,

Surabaya, Sukolilo, 60111, Jawa Timur, Indonesia.

Email: mntahobari@gmail.com

---

## 1. INTRODUCTION

In this information technology age, transferring files between users has been a need for everyone, anytime and anywhere. Public network that is normally used for such communication is however insecure. This means that everything being transferred through the internet can be intercepted and changed. If the modified message is forwarded to the receiver, the message integrity and confidentiality are violated. Hence, data should be protected during the transmission process. As one of the methods to protect data, data hiding technique hides secret information within a multimedia file such as image, video, audio or text [1]. In the simplest form, it was implemented thousands years ago before the information technology was developed [2]. Furthermore, this technique was also used in the Second World War [3].

Watermarking technique is a kind of spatial domain-based data hiding approach that uses the same technique as data hiding. In the spatial domain, secret data are hidden within the redundancy information in a cover media. This is different from transformed domain where secret information are hidden in coefficients obtained by transforming spatial values to frequency coefficients by using dimensional transform methods such as discrete cosine transform (DCT), discrete wavelet transform (DWT) or integer wavelet transform (IWT). These coefficients are then utilized for hiding secret data in transformed domain data hiding [4]. Nevertheless, in watermarking-based method the hidden secret data are related to the cover media for its copyright identification [5]. In this research field, high visual quality for high embedding capacity is a challenging problem.

The high distortion of original media causes the stego to be suspected to carry secret data, which may provoke the adversaries in doing their best to break the embedding algorithm. Due to the common appearance of original and stego media, unauthorized users who intercept the data may not realize that a secret message is being transmitted. In general, after a secret message has been embedded to the cover, the resulted stego medium is sent over a public network to the destination where the secret message and the cover are extracted and recovered accordingly. In literatures, data hiding techniques have been proposed by various researchers to maintain high quality of stego image for a given embedding capacity after secret data embedding, for example in [6] and [7]. In [8], a Difference Expansion (DE) based data hiding algorithm is proposed. As one of spatial domain techniques, this algorithm exploits redundancy information between pixels to hide a secret message. In [9], a high embedding capacity for low distortion of host image was developed where an RGB image was used and secret bits are inserted based on pixel block classification. In general, it is a challenging problem to develop a data hiding method that supports both high visual quality and payload capacity because these two factors are inversely proportional [10], [11].

In this paper, we propose a new base point in the computation of differences between pixels of image in combination with a new reduced difference expansion to minimize the difference before secret data embedding process takes place. Hence, the quality of stego image is enhanced for a given payload capacity.

This paper is organized in 5 sections. In Section 1, we give an overview of the research problems. In Section 2, we describe the literature study on data hiding based on difference expansion method. Section 3 explains the proposed method while in Section 4, the evaluation of the method and its results are presented which are then compared with previous methods. We end up the work in Section 5 by providing the conclusion remarks and the future works.

## 2. RESEARCH METHOD

In [8], a data hiding algorithm that uses difference expansion was proposed. The algorithm hides secret information within the difference between two adjacent pixels. Here, the difference and average are computed then a message is embedded within the difference while the average is used during the creation of stego pixels and in the extraction process. The algorithm is presented as follow. Suppose, the cover image is divided into blocks comprising two successive pixels  $p_n, p_{n+1}$ . Let  $h_n, h'_n$  be the difference without secret message and difference with secret message embedded, respectively; while  $p'_n$  is the pixel whose difference contains the secret data  $s_i$ . In the paper,  $n$  takes the value 0 to 1 because a block is composed of two pixels and only a single difference is computed which means  $n$  for difference is subjected to be 1. The index  $i$  is an index for secret message that takes 1 because only a single bit from a secret message is embedded within a difference. In addition,  $\bar{x}$  denotes the average between those two pixels. The stego pixels are obtained through the processes from Equations (1) to (5).

$$h_n = p_{n+1} - p_n \quad (1)$$

$$\bar{x} = \left\lfloor \frac{p_n + p_{n+1}}{2} \right\rfloor \quad (2)$$

$$h'_n = 2 \times h_n + s_i \quad (3)$$

$$p'_n = \bar{x} + \left\lfloor \frac{h'_{n+1}}{2} \right\rfloor \quad (4)$$

$$p'_{n+1} = \bar{x} - \left\lfloor \frac{h'_n}{2} \right\rfloor \quad (5)$$

In the extraction process, the difference between stego pixels is computed. Then the least significant bit function is applied to extract the secret message. Thereafter, the original difference is calculated which in turn is used to compute original pixels. First, the difference between stego pixels is calculated using  $h'_n = p'_{n+1} - p'_n$ . Secondly, the secret message is taken from each of the differences as shown in  $s_i = LSB(h'_n)$ . After getting the secret message, the original difference is obtained by using Equation (6) where the difference computed from the stego image is right shifted. The original pixel are given back by using Equations (7) and (8) where the former is used to compute the original first pixel while the latter is used to get the second one. This demonstrates that difference expansion is a reversible data hiding method where both secret message and cover are extracted and recovered respectively without any damage.

$$h_n = \left\lfloor \frac{h'_n}{2} \right\rfloor \quad (6)$$

$$p_n = \bar{x} + \left\lfloor \frac{h_n+1}{2} \right\rfloor \quad (7)$$

$$p_{n+1} = \bar{x} - \left\lfloor \frac{h_n}{2} \right\rfloor \quad (8)$$

The method suffers to underflow or overflow issue. In all grayscale images, overflow happens if the resulted pixel value is greater than 255 ( $p_n > 255$ ), while underflow happens if it is less than 0 ( $p_n < 0$ ). To ensure that the problem is avoided, the Equation (9) is verified.

$$0 \leq \bar{x} + \left\lfloor \frac{h'_n+1}{2} \right\rfloor \leq 255 \text{ and } 0 \leq \bar{x} - \left\lfloor \frac{h'_n}{2} \right\rfloor \leq 255 \quad (9)$$

In this method, a pixel block may have one of three possible characteristics: expandable, changeable or non-changeable. Here, a block is considered as expandable if Equation (10) is satisfied and as changeable if equation (11) is fulfilled. The non-changeable blocks are ignored because they cause either an overflow or underflow problem. Many algorithms have been proposed to improve the algorithm either by increasing quality of stego image [10], payload capacity [12], [13] or both [14], [15].

$$|2 \times h_n + s_i| \leq \min(2(255 - \bar{x}), 2\bar{x} + 1) \quad (10)$$

$$|2 \times \left\lfloor \frac{h_n}{2} \right\rfloor + s_i| \leq \min(2(255 - \bar{x}), 2\bar{x} + 1) \quad (11)$$

In [12], Alattar proposed a data hiding method based on quad pixels block that aims to improve the embedding capacity. To make this, the block size increases from  $1 \times 2$  to  $2 \times 2$  pixels. This time, the method is able to carry three secret bits in a single pixel block instead of one secret bit. The difference integer transform is computed by using Equation (12) where the first difference is considered as the average of all four pixels in a single block.

$$\begin{cases} h_0 = \left\lfloor \frac{p_0+p_1+p_2+p_3}{4} \right\rfloor \\ h_1 = p_1 - p_0 \\ h_2 = p_2 - p_1 \\ h_3 = p_3 - p_2 \end{cases} \quad (12)$$

In his method, the secret data embedding is performed in the same way as in [8] except that there are three differences to carry three secret bits. Thereafter, the inverse integers transform scheme to obtain original pixels is calculated using Equation (13).

$$\begin{cases} p_0 = h_0 - \left\lfloor \frac{h_1+h_2+h_3}{4} \right\rfloor \\ p_1 = h_1 + p_0 \\ p_2 = h_2 + p_1 \\ p_3 = h_3 + p_2 \end{cases} \quad (13)$$

In another research done to improve the visual quality of stego image, Liu *et al* [16] proposed to reduce the difference before the embedding process. The reduction is performed if and only if the difference is greater than 2 ( $h_n > 2$ ), otherwise it is not reduced, see Equation (14) where  $h''_n$  is the reduced difference. After getting  $h''_n$ , the embedding and extraction process have become the same as the method proposed in [8]. The location map (*loc*) specified in Equation (15) was used for the purpose of reversibility.

$$h''_n = \begin{cases} h_n & \text{if } h_n < 2 \\ h_n - 2^{\lfloor \log_2(h_n) \rfloor - 1} & \text{if } h_n > 2 \end{cases} \quad (14)$$

$$loc = \begin{cases} 1 & \text{if } 2^{\lfloor \log_2(h''_n) \rfloor} \neq 2^{\lfloor \log_2(h_n) \rfloor} \\ 0 & \text{if } 2^{\lfloor \log_2(h''_n) \rfloor} = 2^{\lfloor \log_2(h_n) \rfloor} \end{cases} \quad (15)$$

At the receiver side, the secret message bits are extracted from the embedded differences by using the *LSB(.)* function. Next, the original difference is recovered by using Equation (16). Once the value has been obtained, the original pixel is recovered using the same expression in [8].

$$h_n = \begin{cases} h''_n + 2^{\lfloor \log_2(h''_n) \rfloor - 1} & \text{if } loc = 0 \\ h''_n + 2^{\lfloor \log_2(h''_n) \rfloor - 1} & \text{if } loc = 1 \end{cases} \quad (16)$$

In [17], an improved quad and Reduced Difference Expansion (RDE) data hiding algorithm was presented. In this method, the quality of stego image has increased due to further reduction of difference between adjacent pixels. In addition, high payload capacity is maintained. Different from integer transform scheme in [12] and [18], the first difference was proposed to be 0 and the integer transform function was performed using Equation (17).

$$\begin{cases} h_0 = 0 \\ h_1 = p_1 - p_0 \\ h_2 = p_2 - p_1 \\ h_3 = p_3 - p_2 \end{cases} \quad (17)$$

In order to reduce image distortion after embedding process, difference reduction formula proposed in [16] is improved to get smaller differences, as depicted in Equation (18) where  $n$  varies from 0 to 3. That is, the reduction is performed for both negative and positive difference values except the difference that is equal to 0 or 1 which may cause either overflow or underflow.

$$h''_n = \begin{cases} h_n - 2^{\lfloor \log_2(h_n) \rfloor} & \text{if } h_n > 1 \\ h_n + 2^{\lfloor \log_2(h_n) \rfloor} & \text{if } h_n < -1 \end{cases} \quad (18)$$

The category of pixel block is slightly different from the previous one [16] because here, the expandable is classified into either RDE expandable or RDE non-expandable. In this case, if a block is RDE expandable, the difference is reduced before the embedding process; but if it is RDE non-expandable, then the embedding is performed directly to the original differences. The drawback of the method is that the similarity between the host image and stego image is still low. In the next section, we illustrate a new approach to improve this method.

### 3. IMPROVING THE QUALITY OF STEGO IMAGE

In this paper, to improve the quality of stego image we proposed a new data hiding approach based on the previous methods, especially that of Ahmad *et al* [17]. A new base point is defined and a reduced difference formula is proposed. That is, the data hiding approach is performed in two successive phases: embedding and extraction processes.

#### 3.1. The embedding process

Similar to other reduced difference expansion algorithms proposed in [12], [16] and [17], the embedding process starts by splitting image into pixel blocks. Also, as in [12] and [17], we use the block of 4 pixels and differences between adjacent pixels are computed based on fixed pixel value in each block. In this work, the fixed pixel is located at the top right in each block. In a  $2 \times 2$  pixel block, the used fixed pixel is at the second position on the top right corner. The proposed integer transform scheme is shown in Equation (19). Here, the differences are computed by subtracting the fixed pixel from each pixel in a block. Thereafter, the differences are saved into a vector as:  $h_n = (h_0, h_2, h_3)$ . In Equation (19), the second difference is equal to 0 ( $h_1 = 0$ ) because the pixel is fixed ( $p_1 - p_1$ ).

$$\begin{cases} h_0 = p_0 - p_1 \\ h_1 = 0 \\ h_2 = p_2 - p_1 \\ h_3 = p_3 - p_1 \end{cases} \quad (19)$$

After computing the differences, the characteristic of blocks is considered in order to reduce the difference before embedding; otherwise, the payload is embedded directly to the difference without reduction. Possible characteristics are expandable, changeable or non-changeable. A difference is reduced if it is categorized as expandable which is subdivided into RDE expandable and non-RDE expandable. Here, RDE expandable is reduced for positive and negative value using Equation (20) while non-RDE expandable is not reduced. This is similar to changeable pixel blocks, where the embedding is done at the original

difference; on the other hand, non-changeable pixel blocks are ignored due to overflow and underflow issues. The reduction is done as follow. If the difference between two pixels is greater than 1 ( $h_n > 1$ ), then the first expression of Equation (20) is used; if ( $h_n < -1$ ), then the second expression is considered. After the difference reduction, secret data are embedded.

$$\begin{cases} h''_n = h_n - \left( 2^{\lfloor \log_2(h_n) \rfloor - 1} + \left\lfloor \sqrt{2^{\lfloor \log_2(h_n) \rfloor}} \right\rfloor \right), & \text{if } h_n > 1 \\ h''_n = h_n + \left( 2^{\lfloor \log_2(|h_n|) \rfloor - 1} + \left\lfloor \sqrt{2^{\lfloor \log_2(|h_n|) \rfloor}} \right\rfloor \right), & \text{if } h_n < -1 \end{cases} \quad (20)$$

$$\forall n \in \mathbb{R}^+, 0 \leq n \leq 3$$

The location map is defined to identify embedded pixel blocks as:  $loc = loc_1, loc_2, loc_3, loc_4, loc_5$  where the bits 0, 1, -1 are assigned to the pixel block categories: expandable, changeable and non-changeable respectively. As stated previously, an expandable pixel block is classified as expandable RDE and non-expandable RDE. For expandable RDE,  $loc_2$  is assigned the value 1 ( $loc_2 = 1$ ), while for expandable non-RDE, it is given the value 0 ( $loc_2 = 0$ ). Here, the embedding is performed using (21) for expandable pixel block. Then, the location map from  $loc_3$  until  $loc_5$  are assigned the value 1, if and only if the first part of Equation (22) is verified for positive differences greater than 1 ( $h_n > 1$ ). Moreover, the location map from  $loc_3$  to  $loc_5$  is given the value 0, if and only if the second part is verified. This is done for negative differences that are less than -1 ( $h_n < -1$ ). These location map values are used to identify the block carrying secret data. For the expandable non RDE, the embedding is performed on the original difference without reduction and the location map  $loc_1, loc_2$  are given the value 1 and 0 respectively.

$$\begin{cases} h'_0 = 2 \times h_0 + s_0 \\ h'_2 = 2 \times h_2 + s_1 \\ h'_3 = 2 \times h_3 + s_2 \end{cases} \quad (21)$$

$$\begin{cases} h_n = h''_n \pm \left( 2^{\lfloor \log_2(h'') \rfloor - 1} + \left\lfloor \sqrt{2^{\lfloor \log_2(h'') \rfloor}} \right\rfloor \right) \\ h_n = h''_n \pm \left( 2^{\lfloor \log_2(|h'') \rfloor} + \left\lfloor \sqrt{2^{\lfloor \log_2(|h'') \rfloor}} \right\rfloor \right) \end{cases} \quad (22)$$

For changeable pixel blocks, the secret data embedding is done using Equation (23). Then, the pixel block is identified using the location map by considering the original difference as follow. If  $h_n$  is odd, then the location map from  $loc_3$  to  $loc_5$  is assigned the value 1; if the original difference  $h_n$  is even, the location map from  $loc_3$  until  $loc_5$  is assigned the value 0. Different from [12], the overflow or underflow issue is alleviated by using the Equation (24) which must be checked before the construction of new pixel (stego pixel) using (25) which later gives stego image. Here, the pixel at the second position is not changed and it is added to each embedded difference.

$$\begin{cases} h'_0 = 2 \times \left\lfloor \frac{h_0}{2} \right\rfloor + s_0 \\ h'_2 = 2 \times \left\lfloor \frac{h_2}{2} \right\rfloor + s_1 \\ h'_3 = 2 \times \left\lfloor \frac{h_3}{2} \right\rfloor + s_2 \end{cases} \quad (23)$$

After embedding phase, the remaining step is to construct the full stego image which is then sent together with location map to the receiver through a public network where the secret message is extracted and cover image is restored.

$$\begin{cases} 0 \leq h'_0 + p_1 \leq 255 \\ 0 \leq h'_2 + p_1 \leq 255 \\ 0 \leq h'_3 + p_1 \leq 255 \end{cases} \quad (24)$$

$$\begin{cases} p'_0 = h'_0 + p_1 \\ p'_1 = p_1 \\ p'_2 = h'_2 + p_1 \\ p'_3 = h'_3 + p_1 \end{cases} \quad (25)$$

**3.2. The extraction process**

In the extraction process, two entities are considered. These are the difference between two pixels of stego image and location map. The extraction process starts by dividing stego image into blocks of the same size as the size of a block used in the embedding process. After that, the difference between the pixels that makes stego pixel block is computed. Then, the secret message extraction is performed directly from the difference by taking the least significant bit (LSB) from each difference using equation  $s_i = LSB(h''_n)$  [12] where  $i$  takes values 0, 1, 2 (three bits to be extracted). The obtained difference from stego image is the one that has the secret message. Due to the difference reduction that has been performed during embedding process when the block was expandable RDE, this difference is right shifted to return to the reduced difference. The original differences are recovered using Equation (26).

$$h_n = \begin{cases} h''_n + 2^{\lfloor \log_2(|h''_n|) - 1 \rfloor} + \lfloor \log_2(h''_n) \rfloor \sqrt{h''_n} & \text{if } loc_3 \text{ to } loc_5 = 1 \\ h''_n + 2^{\lfloor \log_2(|h''_n|) \rfloor} + \lfloor \log_2(h''_n) \rfloor \sqrt{h''_n} & \text{if } loc_3 \text{ to } loc_5 = 0 \end{cases} \tag{26}$$

For non-RDE expandable, we have adapted the formula proposed by Alattar in [12] to recover the original differences using Equation (27). If the block is changeable and difference value is even, then the original difference is recovered using the first expression of Equation (28), while for the odd difference value the second part of Equation (28) is used. Here,  $h'_n$  is the embedded difference computed from stego image. Next, the original image is reconstructed using Equation (29).

$$h_n = \lfloor \frac{h'_n}{2} \rfloor \tag{27}$$

$$\begin{cases} h_n = \lfloor \frac{h'_n}{2} \rfloor \\ h_n = \lfloor \frac{h'_n}{2} \rfloor + 1 \end{cases} \tag{28}$$

$$\begin{cases} p_0 = h_0 + p'_1 \\ p_1 = p'_1 \\ p_2 = h_2 + p'_1 \\ p_3 = h_3 + p'_1 \end{cases} \tag{29}$$

In general, the comparison between our proposed method and the previous method [17] is illustrated in Table 1. The stages of embedding and extraction processes for each pixel block are depicted in Figure 1, in the form of algorithm.

Table 1. Comparison between the proposed and the previous Method of [17]

Method	Base point	Integer transform scheme	Reduced difference expansion	New pixel	Overflow /underflow prevention
Ahmad <i>et al</i> [17]	dyna- mic	$\begin{cases} h_0 = 0 \\ h_1 = p_1 - p_0 \\ h_2 = p_2 - p_1 \\ h_3 = p_3 - p_2 \end{cases}$	$h'' = \begin{cases} h_n - 2^{\lfloor \log_2(h_n) \rfloor} & \text{if } h_n > 1 \\ h_n + 2^{\lfloor \log_2(h_n) \rfloor} & \text{if } h_n < -1 \end{cases}$	$\begin{cases} p'_0 = p_0 \\ p'_1 = h'_1 + p'_0 \\ p'_2 = h'_2 + p'_1 \\ p'_3 = h'_3 + p'_2 \end{cases}$	---
Proposed method	static	$\begin{cases} h_0 = p_0 - p_1 \\ h_1 = 0 \\ h_2 = p_2 - p_1 \\ h_3 = p_3 - p_1 \end{cases}$	$h''_n = \begin{cases} h_n - \left( 2^{\lfloor \log_2(h_n) \rfloor - 1} + \lfloor \log_2(h_n) \rfloor \sqrt{h_n} \right), & \text{if } h_n > 1 \\ h_n + \left( 2^{\lfloor \log_2(h_n) \rfloor - 1} + \lfloor \log_2(h_n) \rfloor \sqrt{h_n} \right), & \text{if } h_n < -1 \end{cases}$	$\begin{cases} p'_0 = h'_0 + p_1 \\ p'_1 = p_1 \\ p'_2 = h'_2 + p_1 \\ p'_3 = h'_3 + p_1 \end{cases}$	$\begin{cases} 0 \leq h'_0 + p_1 \leq 255 \\ 0 \leq h'_2 + p_1 \leq 255 \\ 0 \leq h'_3 + p_1 \leq 255 \end{cases}$

```

The step by step for the embedding process
1. Read image
2. Read secret message and convert to (1,0) bits
3. Compute differences  $h_0, h_2, h_3$  equation (19)
4. If  $(2 \times h_n + s_i > 0 \ \&\& \ 2 \times h_n + s_n < 255)\%EX.$ 
5.   Set location map  $loc_1$  to 1
6.   if  $|h_0| \ || \ h_2| \ || \ h_3| \geq 2$ 
7.     Set location map  $loc_2$  to 1
8.     if  $h_0, h_2, h_3 > 1$ 
9.       %RDE Ex. For positive difference
10.      Apply RDE using equation (20) part 1
11.      Embed data using equation (21)
12.    else if  $h_0, h_2, h_3 < -1$ 
13.      %RDE Ex. for negative difference
14.      RDE using equation (20) part2
15.      Embed data using equation (21)
16.    end if
17.    if the first part equation (22) is met
18.      Set  $loc$  from  $loc_3$  to  $loc_2$  to 1
19.    else
20.      Set  $loc$  from  $loc_3$  to  $loc_5$  to 0
21.    end if
22.    else %RDE non expandable
23.      Set location map  $loc_4$  to 0
24.      Embed data using equation (21)
25.    end if
26.  else if  $(2 \times \lfloor \frac{h_n}{2} \rfloor + s_i > 0 \ \&\& \ 2 \times \lfloor \frac{h_n}{2} \rfloor + s_i <$ 
27.     $255)\%$  changeable
28.    Set location map  $loc_1$  to 0
29.    if  $\frac{h_0}{2}, \frac{h_2}{2}, \frac{h_3}{2} == 0.$  % even differences
30.      Set location map from  $loc_2$  to  $loc_5$  to 0
31.    else if  $\frac{h_0}{2}, \frac{h_2}{2}, \frac{h_3}{2} \neq 0.$  % odd differences
32.      Set location map from  $loc_2$  to  $loc_5$  to 1
33.    end if
34.    Embed data using equation (23)
35.    else % non changeable
35.   Set location map to -1. %  $loc_1$ 
36.   New pixel == original pixel
37.   Check over and underflow equation (24)
38.   Compute new stego pixel equation (25)
39.   end if
40.   Go to next block

Note: the phrase preceded with percentage sign (%) are
      comments
      EX: Expandable

The step by step for extraction process
1.   Compute  $h_0, h_2, h_3$  % from stego image
2.   if location map  $loc_1 == 1$  % expandable
3.     Extract data using  $s_i = LSB(h'_n)$ 
4.     Right shift the differences
5.     if location map from  $loc_3$  to  $loc_5 == 1$ 
6.       Restore differences equation (26) part 1
7.     else
8.       Restore difference equation 26 part 2
9.     end if
10.    if  $loc_1 == 1, loc_2 == 0$  % non RDE expandable
11.      Restore difference using equation (27)
12.    end if
13.    else if location  $loc_1 == 0$  % changeable
14.      Extract data using equation  $s_i = LSB(h'_n)$ 
15.      if Location map from  $loc_3$  to  $loc_5 == 0$ 
16.        Restore difference equation (28) part1
17.      else
18.        Restore difference equation (28) part 2
19.      end if
20.    else % non changeable
21.      Restore pixel block using (29)
22.    end if
23.    Go to the next block

Note: The phrase preceded with percentage sign (%) are comments

```

Figure 1. The proposed embedding and extraction processes

#### 4. EXPERIMENTAL RESULTS AND DISCUSSION

In the implementation of our algorithm, we used 8 bits grayscale medical images of resolution 512×512 taken from [19]. In addition, to evaluate the performance of the proposed method, we also implemented the method in [17]. Furthermore, in the experiment, we use MATLAB programming. Thereafter, random binary bits of different sizes (5Kb, 10Kb and 20Kb) are generated using a random function. The visual quality is measured using Peak Signal to Noise Ratio (PSNR) and Means Square Error (MSE) in order to evaluate how much the original image is distorted after secret message embedding. The formula to compute PSNR and MSE are illustrated in Equations (30) and (31) respectively where  $N$  is the number of pixels,  $I_i$  is the original image and  $I'_i$  is the stego image.

$$MSE = \frac{1}{N} \sum_{i=1}^N (I_i - I'_i)^2 \quad (30)$$

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \quad (31)$$

The results of the experiment can be presented in Table 2, Table 3, Table 4 according to the size of secret message. That is, we use 5Kb, 10Kb, and 20Kb whose corresponding PSNR values are depicted in Table 2, Table 3 and Table 4, respectively. Due to different pixel intensity (brightness), smoothness and texture, the visual quality of stego image are different for the same payload capacity. In some cases, the cover are relatively similar to the stego images, as indicated by their PSNR values. In Table 2, where the secret message is smaller (5Kb), the performance is better than that of bigger size, as shown in Table 3 and Table 4. This shows that more payload leads to decreasing visual quality.

Table 2. PSNR Values when 5 Kb of secret message is embedded

Images	Ahmad <i>et al.</i> [17]	Proposed method
	PSNR(dB)	PSNR(dB)
Abdominal	40.1559	40.8684
Leg	40.0875	40.6171
Head	33.6507	34.7428
Hand	39.8570	40.6476
Chest	40.7396	42.4564
Average	38.89814	39.86646

Table 3. PSNR Values when 10 Kb of secret message is embedded

Images	Ahmad <i>et al.</i> [17]	Proposed method
	PSNR(dB)	PSNR(dB)
Abdominal	37.6805	38.1745
Leg	38.0537	38.9313
Head	31.9050	32.7272
Hand	37.5609	38.0336
Chest	38.0634	39.1532
Average	36.6527	37.4039

Table 4. PSNR Values when 20 Kb of secret message is embedded

Images	Ahmad <i>et al.</i> [17]	Proposed method
	PSNR(dB)	PSNR(dB)
Abdominal	37.6781	38.1718
Leg	38.0088	38.8323
Head	31.9048	32.7263
Hand	37.5563	38.0310
Chest	38.0617	39.0524
Average	36.6419	37.3627

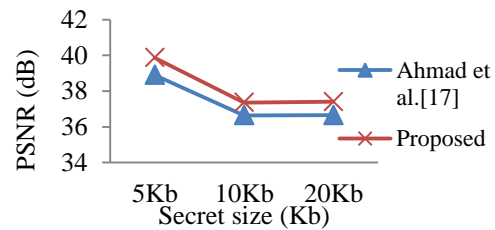


Figure 2. PSNR (average) comparison between the proposed and [17]

In both proposed and previous method in [17], the visual quality is better for chest image than other medical images, while the lower performance is obtained for the head image. The reason of this phenomenon can be described as follow. Some images have a large white color and some do not, which means the highest and the lowest value of pixels. Meanwhile, gray scale images have the pixel values between white and black. Therefore, in case many pixels tends to be white (head) or black (chest), this characteristic affects the computed difference, which also influences the visual quality of stego image after the embedding process.

The performance of two algorithms for all five medical images is evaluated based on the average of PSNR for each secret size, whose results are presented in Figure 2. Here, the results prove that the proposed method performs better for each size of secret message used. In addition, along with the increasing of the payload size, the visual quality decreases. As shown in those tables, the average of PSNR is 39.866 dB and 38.898 dB for the proposed and that of [17], respectively. If the secret message of 10Kb is hidden, we obtain 37.403 dB on average, while [17] is 36.652 dB. Next, ours is 37.362 dB, and [17] is 36.641 for 20Kb of secret data.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we have developed a new data hiding algorithm based on enhanced reduced difference expansion and fixed pixel value for each block of four pixels to solve the problem of high distortion of original image after bits embedding. Secret message of different sizes (5Kb, 10Kb and 20 Kb) was used to evaluate the proposed approach. The experimental results show that our proposed method performs better in terms of the visual quality of stego image. Moreover, by increasing the payload size, the quality of stego image decreases gradually. According to the results, we also notice that the embedding capacity and visual quality could be affected by the properties of image like smoothness, brightness and textures in addition to the choice of the basepoint of image pixel blocks.

For further research, this method can be improved in terms of embedding capacity by considering the image properties during difference computation before the embedding process. So, more secret bits can be hidden several times in each block based on the embedding level of each pixel block.

## REFERENCES

- [1] A. K. Sahu and G. Swain, "A Review on LSB Substitution and PVD Based Image Steganography Techniques," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 2, no. 3, pp. 712-719, 2016.
- [2] C.-H. Huang, S.-C. Chuang and J.-L. Wu, "Digital invisible-Ink data hiding based on spread-spectrum and quantization techniques," *IEEE Transactions on multimedia*, vol. 10, no. 4, pp. 557-569, 2008.



- [3] L. Y. Por and B. Delina, "Information hiding: A new approach in text steganography," in *International Conference on Applied Computational Science*, Hangzhou, China, 2008.
- [4] V. A. Kuma, C. Dharmaraj and C. S. Rao, "A Hybrid Digital Watermarking Approach Using Wavelets and LSB," *International Journal of Electrical and Computer Engineering*, vol. 7, no. 5, pp. 2483-2495, 2017.
- [5] R. Thanki and K. Borisaga, "Security of Biometric Data Using Compressed Watermarking Technique," *International Journal of Electrical and Computer Engineering*, vol. 4, no. 5, pp. 758-766, 2014.
- [6] K.-H. Jung, "A high-capacity reversible data hiding scheme based on sorting and prediction in digital images," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13127-13137, 2017.
- [7] F. Y. Shih and X. Zhong, "Intelligent Watermarking for High-Capacity Low-Distortion Data Embedding," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 30, no. 5, 2016.
- [8] J. Tian, "reversible data embedding using difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, Aug 2003.
- [9] M. S. Arya, M. Rani and C. S. Bedi, "Improved Capacity Image Steganography Algorithm using 16-Pixel Differencing with n-bit LSB Substitution for RGB Images," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 6, pp. 2735-2741, 2016.
- [10] H. Yi, S. Wei and H. Jianjun, "Improved reduced difference expansion based reversible data hiding scheme for digital images," in *International Conference on Electronic Measurement & Instruments, Beijing, China, 2009*.
- [11] Z. Hua, D. Shoujian and Z. Daozhen, "A reversible watermarking scheme for vector drawings based on difference expansion," in *Computer-Aided Industrial Design & Conceptual Design*, Yiwu, China, 2010.
- [12] A. M. Alattar, "Reversible watermark using difference expansion of quads," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147-1156, 2004.
- [13] A. Arham, H. A. Nugroho and T. B. Adji, "Multiple layer data hiding scheme based on difference expansion of quad," *Signal Processing*, vol. 137, pp. 52-62, August 2017.
- [14] M. Liu, H. S. Seah, C. Zhu, W. Lin and F. Tian, "Reducing location map in prediction-based difference expansion for reversible image data embedding," *Signal Processing*, vol. 92, no. 3, pp. 202-206, 2015.
- [15] H.-C. Wu, C.-C. Lee, C.-S. Tsai, Y.-P. Chu and H.-R. Chen, "A high capacity reversible data hiding scheme with edge prediction and difference expansion," *Journal of Systems and Software*, vol. 82, no. 12, p. 1966-1973, August 2009.
- [16] C.-L. Liu, D.-C. Lou and C.-C. Lee, "Reversible data embedding using reduced difference expansion," in *Intelligent Information Hiding and Multimedia Signal Processing, Kaohsiung, Taiwan, 2007*.
- [17] T. Ahmad, M. Holil, W. Wibisono and R. M. Ijtihadie, "An improved quad and RDE-based medical data hiding method," in *Computational Intelligence and Cybernetics, Yogyakarta, Indonesia, 2013*.
- [18] M. Holil and T. Ahmad, "Secret Data Hiding by Optimizing General Smoothness Difference Expansion-Based Method," *Journal of Theoretical and Applied Information Technology*, vol. 72, no. 2, pp. 155-163, 2015.
- [19] "Partners infectious disease images- emicrobes Digital Library," April 2017. [Online]. Available: [www.idimages.org](http://www.idimages.org).