Enhancing DE-based Data Hiding Method by Controlling the Expansion

Dwi S. Angreni, Tohari Ahmad
shinta14@mhs.if.its.ac.id, tohari@if.its.ac.id
Departement of informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Jawa Timur 60111 Indonesia

Abstract- **Communication by transferring data using computer networks have an important role nowadays. There are many activities which require sending or receiving data from one host to another. However, it can be a problem if the data is compromised during the transmission. In order to resolve the problem, steganography is used to secure the data. It is a data hiding method which can hide sensitive data in media so that unauthorized parties may not be suspicious or interested in attack. The previous reversible data hiding methods have been able to recover the image data and to improve the capacity of embedding. However, there are some problems that cannot be resolved such as overflow and underflow which result to decreasing the quality. In this paper, we propose a method which is able to overcome these problems that each pixel value on the cover image is expandable. The experimental results show that the PSNR and the capacity is higher than existing methods.**

I. INTRODUCTION

In this era, sending data using computer networks is very needed. There are many objectives for someone, to send data to others. However, it can be a problem if the data is attacked during this data transfer. This attack can harm and disturb the privacy of users. There are various data that need a protection, such as medical data, military data, banking transaction and financial report [1].

There are two kinds of techniques that often used to secure data. These two techniques is called cryptography and steganography. There are some differences between those two methods. The former is a method which can change data into unreadable or meaningless form. While, the latter hides data into a cover media, such as image, audio or video. The transformation that occurs in data caused by cryptography can be easily determined. This can make other parties interested in breaking the code. Wheres, steganography hides data in the media whose result may not be recognized by unauthorized parties [2] [3] [4].

Steganography is classified into two types: reversible and irreversible. When we hide secret data without storing the recovery information, then the stego image will be distorted permanently. It means image can not be restored to the original form which is called irreversible. The least significant bit (LSB) is one type of data hiding which irreversibly change the LSB of the host image with the secret data. Besides that, if we have the recovery information, then original media can be recovered completely to the original form. This process is called reversible. For some situation, permanent distortion which occurs in stego image cannot be accepted. For example, when hiding secret data into medical images or paintings arts, a distorted image may lead to an incorrect diagnosis in medical or incorrect appraisement in work of art. Therefor, reversible data hiding methods is needed to be able to solve the above problems [5].

There are many reversible data hiding methods based on integer-transform such as difference expansion (DE) [6] proposed by Tian. Lou et al. in [7] proposes reduce difference expansion called RDE, which adds a reduced function to increase the expansion. Alattar [8] proposes quad difference expansion (QDE) by adding further Tian algorithm. This is done by embedding secret bits in each quad adjacent pixels. Ahmad et al. [9] propose a reversible data hiding method that combines [7] and [8]. In [10], Holil et. al propose a method based on the quad smoothness [11] and generalized difference expansion [12].

In general, the above existing methods have been able to recover the image data and to improve the capacity of the secret dat which can be embedded. However, there are some problems that can not be resolved by these methods such as overflow and underflow. Therefore, we propose a method which is able to overcome these problems so that each pixel value in the cover image is expandable.

The rest of the paper is structured as follows. Section 2 represent the existing methods. Section 3 describes our proposed method. Section 4 show the experimental result and the conclusion is given in section 5.

II. RELATED WORKS

In this section, we will explain the previous methods, which have a relationship with our proposed method. The methods will write into a couple subsection, as shows below.

A. *Difference Expansion*

Difference Expansion (DE) is a reversible data hiding technique introduced by Tian in [6]. This technique provides an embedding space on the difference value that we obtained between pairs of pixels. DE is using an 8-bit grayscale images which consists pairs of pixel $(x, y), x, y \in Z, \ni 0 \leq x, y \leq 255$.

The embedding process works by calculating the integer average value l and integer difference h as specified in (1) with the invers transform of x and y as shown in (2).

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor, \;\; h = x - y \tag{1}$$

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor, \;\; y = l - \left\lfloor \frac{h}{2} \right\rfloor \tag{2}$$

There are some problems that often arise, when the embedding process is performed. Those are overflow and underflow. The former is a condition when the result of modified pixel value is more than 255, while the latter is a condition when the result of modified pixel value is less than 0. To prevent these, the embedding result must satisfy (3) and (4).

$$|h| \le 2\lfloor 255 - l \rfloor, \text{and } |h| \le 2l + 1 \tag{3}$$

$$\begin{cases} |h| \le 2(255 - l), \text{if } \; 128 \le l \le 255 \\ \;\; |h| \le \; 2l + 1, \; \text{if } \;\; 0 \le l \le 127 \end{cases} \tag{4}$$

After that, the embedding of the secret bit b in the difference value h is carried out. The new difference value h' is obtained after secret bit b is embedded to h as specified in (5).

The original pair of pixel values $x'$and y' are calculated by using (6).

$$h' = 2\,h + b \tag{5}$$

$$x' = l + \left\lfloor \frac{h'+1}{2} \right\rfloor, \quad y' = l - \left\lfloor \frac{h'}{2} \right\rfloor \tag{6}$$

Extraction and recovery process almost have the same steps with the embedding process. Extract the LSB of the difference value to obtain the secret bit b. The difference value is obtained from modified pairs of pixel x' and y'. Both calculations is shown in (7).

$$b = LSB(h')\;, \text{and } \; h = \left\lfloor \frac{h'}{2} \right\rfloor. \tag{7}$$

Finally, the original pixel pair value (x, y) is can be obtained by following the same steps as in embedding process.

B. *Reduced Difference Expansion*

Reduced difference expansion (RDE) [7] is a reversible data hiding method that proposed by Lou et al. in 2009. This method improve [6] by adding transformation function to reduce the difference value. The reduced difference value, which is called h' is calculated using (8).

$$h' = \begin{cases} h & \text{if } h' < 2 \\ h - 2^{\lfloor log_2 h \rfloor - 1} & \text{otherwise} \end{cases} \tag{8}$$

The location map is required to recover the original pixel value. It is constructed when the original difference value has not changed. The size of the location map is the same to the number pairs of pixel values. If the difference value h is 0 or 1, then it is not reduced and the location map value is 0. On the other hand, if h =2 and h'=1, then the location map value is 1. Location map is specified in (9).

$$\text{Location Map} = \begin{cases} 0 \text{ if } 2^{\lfloor log_2 h' \rfloor} = 2^{\lfloor log_2 h \rfloor} \; or \; h' = h \\ \quad 1 \text{ if } 2^{\lfloor log_2 h' \rfloor} \ne 2^{\lfloor log_2 h \rfloor} \end{cases} \tag{9}$$

For example, we assume that difference value h from a pair of pixels is 5, average value l is 202 and secret bit value b is 1. Then reduced difference value computed by $h' = h - 2^{\lfloor log_2 h \rfloor - 1} = 5 - 2^1 = 3$. Secret bit b is embedded to h' by applying (5), so that we obtain $h'' = 2\,h + b = 2 \text{ x } 3 + 1 = 7$. Then, the location map value is set 1, based on (9).

Furthermore, apply (10) to calculate new pair of pixels, x' and y'.

$$\begin{aligned} x' &= l + \left\lfloor \frac{h''+1}{2} \right\rfloor = 202 + \left\lfloor \frac{7+1}{2} \right\rfloor = 206 \\ y' &= l - \left\lfloor \frac{h''}{2} \right\rfloor = 202 - \left\lfloor \frac{7}{2} \right\rfloor = 199. \end{aligned} \tag{10}$$

$$h \begin{cases} h' + 2^{\lfloor log_2 h' \rfloor - 1} & \text{if Location map} = 0 \\ h' + 2^{\lfloor log_2 h' \rfloor} & \text{if Location map} = 1 \end{cases} \tag{11}$$

The average l and difference value h' is required to extract the secret data from the stego image. Apply (1) to obtain average value l. By contrast, difference value h' can be calculated by using (11).

$$l = \left\lfloor \frac{206+199}{2} \right\rfloor = 202, h'' = 206 - 199 = 7 \tag{12}$$

Then the secret data b can be extracted by taking the LSB of h'' as shown in (13).

$$b = LSB(h'') = LSB(7) = 1 \tag{13}$$

Compute h' as defined in (14).

$$h' = \left\lfloor \frac{h''}{2} \right\rfloor = \left\lfloor \frac{7}{2} \right\rfloor = 3 \tag{14}$$

The location map is required, to recover the original difference value. Because the location map we obtained is 1, then the original difference h is calculated as $h = h' + 2^{\lfloor log_2 h' \rfloor} = 3 + 2^1 = 5$. Finally, by using h, the original pairs of pixel values can be recovered as shown in (15).

$$\begin{aligned} x &= l + \left\lfloor \frac{h+1}{2} \right\rfloor = 202 + \left\lfloor \frac{5+1}{2} \right\rfloor = 205 \\ y &= l - \left\lfloor \frac{h}{2} \right\rfloor = 202 - \left\lfloor \frac{5}{2} \right\rfloor = 200 \end{aligned} \tag{15}$$

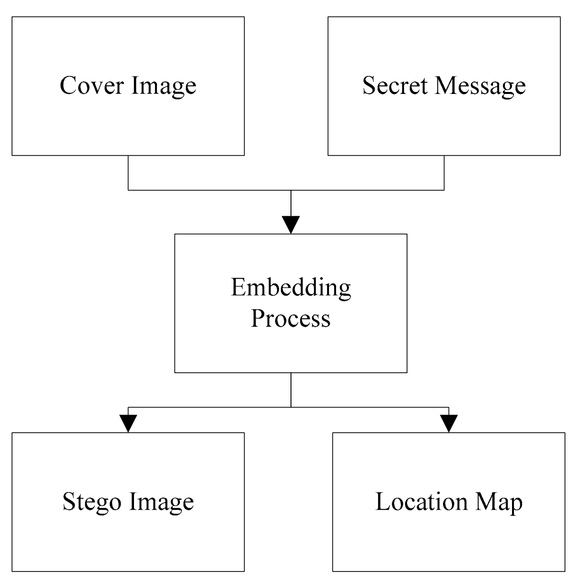

Fig. 1. Embedding Process

III. PROPOSED METHOD

There are some improvement that we introduce in this paper e.g. we use one pixel to embed one bit secret message, extend some functions that have been used in the existing methods and add some new conditions. In this section, we explain the algorithm more details.

A. Embedding Process

The embedding process in this study is basically similar with DE and RDE, which conduct it in the difference value from pairs of pixel. In this study, we propose to conduct the embedding process in the difference that obtained from pixel and random value. The random value make one pixel can be embedded with one secret bit. It makes this method can increase the capacity as much as two times the existing method.

The integer difference h and average l are defined as follows (16).

$$h = x - R, \qquad l = \begin{cases} \left\lfloor \frac{x+R}{2} \right\rfloor & \text{if } h \le 1 \\ \left\lceil \frac{x+R}{2} \right\rceil & \text{if } h > 1 \end{cases} \tag{16}$$

where x denotes the pixel value and R denotes random value. The random value can be the same or different for embedding and recovery process.

The average function has two conditions. These two conditions are mean to adjust the average with every difference value that might occurs.

To increase the expansion, we should reduce the difference value. We introduce an improvement of reduction function as in (17).

$$h' = \begin{cases} |h| & \text{if } h = 0 \text{ } or \text{ } h = 1 \\ \left\lceil \frac{|h|-1}{2} \right\rceil & \text{otherwise} \end{cases} \tag{17}$$

where h' denotes the reduce difference value. This function is another form of reduce function that based on [7]. It is simple and the result is smaller than the existing function. It uses the absolute of h to obtain the positive integer value. This function is mainly used to control the original difference value so that the expansion increase.

The embedding of the secret bit can be represented in (18).

$$h'' = \begin{cases} 2h' - b & \text{if } h < 0 \\ 2h' + b & \text{if } h \ge 0 \end{cases} \tag{18}$$

where h'' is the difference value after we embed the secret message bit b. Equation (5) is extended by adding a new condition. The purpose of this improvement is to maintain the even pixel value that has $h < 0$. Then, for the other condition, we use the original function in [6] which aim to control the odd pixel value with $h \ge 0$.

The invers transform can be defined as (19), where x' denotes new pixel that obtained after embedding process. In this paper, we improve the original function in [6] by forming (19). Equation (19) is mean to restrict the expansion that can cause overflow and underflow problems.

To recover the pixel value correctly, we need recovery information, that is location map. In this study, location map is not measured based on wheather the pixel is expandable or not like in [6]. On the other hand, it is based on the original pixel value condition. If the original pixel value is odd, then the location map value is 1. The location map is 0, if the original pixel value is even. The location map can be defined as follows (20).

$$x' = \begin{cases} l - \left\lceil \frac{h''+1}{2} \right\rceil, \text{ if } x = 2n+1 & \multirow{2}{*}{if h < 0} \\ l - \left\lceil \frac{h''}{2} \right\rceil, \text{ if } x = 2n & \\ l + \left\lfloor \frac{h''-1}{2} \right\rfloor, \text{ if } x = 2n+1, & \text{if h} \ge 0 \\ l + \left\lfloor \frac{h''}{2} \right\rfloor, \text{ if } x = 2n & \end{cases} \tag{19}$$

$$\text{Location Map} = \begin{cases} 1, & \text{if } x = 2n+1 \\ 0, & \text{if } x = 2n \end{cases} \tag{20}$$

The size of location map is the same with the number of pixel in cover image. The secret messsage is converted into binary bits and embedded into cover image. Then, we obtain two files, namely, stego-image and location map. We can see the flowchart of embedding process in Fig. 1.

For a simple example in embedding process, we summarize the steps as follows :

1. Choose random value, example $R = 300$ and pixel value x =255 and secret data bit $b = 0$.
2. Calculate the difference value.
   $$h = x - R = 255 - 300 = -45.$$
3. Then, the average value.
   $$l = \left\lfloor \frac{x+R}{2} \right\rfloor = \left\lfloor \frac{255+300}{2} \right\rfloor = 277.$$
4. The reduced difference value is obtained as follow.
   $$h' = \left\lceil \frac{|h|-1}{2} \right\rceil = \left\lceil \frac{|-45|-1}{2} \right\rceil = 22$$
5. Then, the secret data bit b is embedded in h' to obtain h''
   $$h'' = 2h' - b = 2 \text{ x } 22 - 0 = 44$$
6. Calculate the new pixel value x' by using condition in (13), because the x value is an even number, so the new pixel value is
   $$x' = l - \left\lceil \frac{h''+1}{2} \right\rceil = 277 - \left\lceil \frac{44+1}{2} \right\rceil = 277 - 23 = 254$$
7. The new pixel is 254 and the location map is 1. Later, the location map is used to recover the original pixel value.

B. Extaction and Recovery Process

In order to obtain the secret message and recover the original image correctly. The first, we have to extract the secret bit b using (21).

$$b = LSB(x') \tag{21}$$

where x' is pixel value that has been modified. The second, recover the original image, by calculating h and l value using (16). In this step, we add new function that shown in (22). The steps that we use in embedding and recovery process are quite similar. The difference between both process is in the input of information. In embedding process we use secret bit b as input while in recovery process we use location map LM. By using

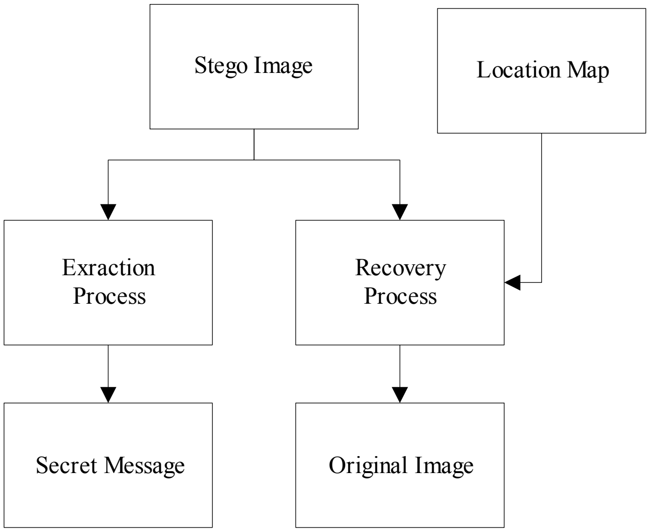

Fig. 2. Extraction and Recovery Process

difference and location map value, we can obtain the new difference value h'' in (22).

$$h'' = \begin{cases} 2h' - LM & \text{if } h < 0 \\ 2h' + LM & \text{if } h \geq 0 \end{cases} \quad (22)$$

where LM denotes the location map. The last, calculate the original pixel value by using (19). The flowchart of extraction and recovery process is shown in Fig. 2. Both of these process are conducted separately. The extraction process performed by taking the LSB of the pixel value, while recovery process is conducted to restored the original pixel value by using the location map value.

To extract and recover the original pixel value, we can use this following steps :

1. Any random value R are chosen. The random value do not have to be the same with the one that used in the embedding process, for example we choose $R = 200$ and pixel value x' =254 and the location map is 1.
2. Extract the secret bit $b = LSB(x') = LSB(254) = 0$.
3. Calculate the difference value $h = x' - R = 254 - 200 = 54$.
4. Then, calculate the average value l.
   $$l = \left\lfloor \frac{x+R}{2} \right\rfloor = \left\lfloor \frac{254+200}{2} \right\rfloor = 227.$$
5. Find the reduce difference value h'.
   $$h' = \left\lfloor \frac{|h|-1}{2} \right\rfloor = \left\lfloor \frac{|54|-1}{2} \right\rfloor = 27.$$
6. Then, the new difference value is calculated by using location map value.
   $$h'' = 2h' + LM = 2 \times 27 + 1 = 55.$$
7. Restore the original pixel value x by using (17). Because x value is an odd number, then the original pixel value is
   $$x = l + \left\lceil \frac{h''}{2} \right\rceil = 227 + \left\lceil \frac{55}{2} \right\rceil = 227 + 28 = 255.$$
8. Finally, we obtain the original pixel value x is 255 and the secret bit b is 0.

As we see our proposed method is reversible and can overcome the overflow and underflow problems, which the embedding result will not over 255 or under 0.

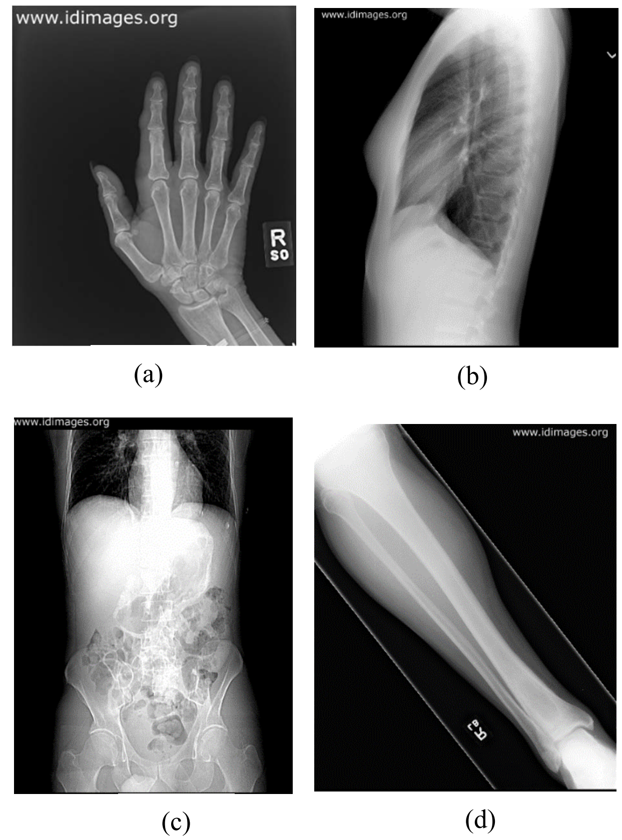

Fig. 3. Data Set Image : (a) hand, (b) lung, (c) abdominal, and (d) leg.

IV. Experimental Result

The experiment result that we have conducted is shown in this section. To conduct the experiment, we use medical image from [13] with various size that we can see in Fig. 3. The secret message we used is standar text in [14]. It contains a collection of paragraphs with length of 1500 bytes. In this experiment we evaluate the quality of the stego image using Peak Signal to Noise (PSNR). To obtain the PSNR, we have to know the Mean Squared Error (MSE) of the cover signal and the modified signal. MSE and PSNR can be defined by (23) and (24) respectively.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (P_i - Q_i)^2 \quad (23)$$

$$PSNR = 20_{log_{10}} \frac{\max(|P_i|)}{\sqrt{MSE}} \quad (24)$$

In additio, we also evaluate the capacity of the cover image, using bit per pixel (bpp). Bpp can be obtained by dividing the number of payload and cover image. The payload

We conduct the test using four medical images, i.e. abdominal, lung, hand and leg. The experiment is works in images with .png format because it is an extensible format for lossless image. At this stage, we will compare the PSNR and the capacity of the proposed method with DE.

The threshold that used in DE is the maximum threshold value. However, our method is not using threshold because the pixel changes after embedding, only increased or reduced by one. This led the use of threshold will not have much effect in the performance of the method. The stego-images obtained from the embedding process is shown in Fig. 4. As we see in Fig. 4, there is no noticeable distortion that shown in the stego image of our method.

The experimental result are depicted in Table I and II. Table I represent the capacity of the embedding process of four test images with DE and RDE. Based on the results obtained, the capacity of proposed method is better than other methods. The highest capacity value obtained from all test images is 1 bpp. On the other hand, the highest capacity value of those both methods is 0.498 bpp, that obtained from hand image.

The main factor which make the capacity of the proposed method can reach 1 bpp is becausein the experiment we did not include the location map with the payload. The location map is stored in a separate file with the stego image. Because of the location map that constructed from the embedding process has a small redundancy, this would make the location map is difficult to compress.

Table II shows the comparation result of the PSNR. The experimental result show that the proposed method also have better PSNR result than the existing methods. The highest PSNR value obtained is from lung image which it goes to 53.53 dB. In DE and RDE, the highest PSNR is also in lung image, which the PSNR value is 41.90 dB and 44.64 dB respectively.

From the experimental result, we can see that our proposed method can enhance the the quality and capacity of stego image. This happened because the even pixel is embedded with secret bit 0 then the value remains the same and if it is embedded with secret bit 1 then the value will increase by 1. While, if odd pixel value is embedded with secret bit 1 the the value will remains the same and if it is embedded with secret bit 0 then the value will decrease by 1. It makes the embedded pixel will not over 0 or 255. This method is also fit for multiple embedding process. Because no matter how much message can be embedded, the PSNR results is stable.

TABLE I.

THE EMBEDDING CAPACITY

Data Set	**Proposed**	**DE**[6]	**RDE**[7]
	Capacity (bpp)	Capacity (bpp)	Capacity (bpp)
abdominal	1	0.493	0.493
hand	1	0.498	0.498
lung	1	0.496	0.496
leg	1	0.495	0.495

TABLE II.

THE QUALITY OF IMAGES

Data Set	**Proposed**	**DE**[6]	**RDE**[7]
	PSNR (dB)	PSNR (dB)	PSNR (dB)
abdominal	53.04	34.15	40.58
hand	51.15	35.23	40.40
lung	53.53	41.90	44.64
leg	52.98	37.98	40.19

V. CONCLUSION

In this paper we introduce a reversible data hiding method that mainly to overcome the overflow and underflow problems. The experimental result shows that the proposed method can

(a)

(b)

Fig. 4. (a) Original Image and (b) Stego Image of Proposed Method.

overcome those both problems. The PSNR and the capacity value is also significantly higher than the existing methods.

Future work may be conducted by increasing the number of secret bit that embedded to one pixel or adding location map to the payload.

REFERENCES

[1] T. Ahmad, H. Studiawan, H. S. Ahmad, R. M. Ijtihadie, and W. Wibisono, “Shared secret-based Steganography for protecting medical data,” in *Computer, Control, Informatics and Its Applications (IC3INA), 2014 International Conference on*, 2014, pp. 87–92.

[2] T. Ahmad and M. Holil, “Increasing the Performance of Difference Expansion-based Steganography when Securing Medical Data,” *SmartCR*, vol. 4, no. 4, pp. 307–322, 2014.

[3] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for data hiding,” *IBM Syst. J.*, vol. 35, no. 3.4, pp. 313–336, 1996.

[4] T. D. Kieu and C.-C. Chang, “A high stego-image quality steganographic scheme with reversibility and high payload using multiple embedding strategy,” *J. Syst. Softw.*, vol. 82, no. 10, pp. 1743–1752, 2009.

[5] C.-C. Lin, S.-P. Yang, and N.-L. Hsueh, “Lossless data hiding based on difference expansion without a location map,” in *Image and Signal Processing, 2008. CISP’08. Congress on*, 2008, vol. 2, pp. 8–12.

[6] J. Tian, “Reversible data embedding using a difference expansion,” *IEEE Trans Circuits Syst Video Techn*, vol. 13, no. 8, pp. 890–896, 2003.

[7] D.-C. Lou, M.-C. Hu, and J.-L. Liu, “Multiple layer data hiding scheme for medical images,” *Comput. Stand. Interfaces*, vol. 31, no. 2, pp. 329–335, 2009.

[8] A. M. Alattar, “Reversible watermark using difference expansion of quads,” in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP’04). IEEE International Conference on*, 2004, vol. 3, pp. iii–377.

[9] T. Ahmad, M. Holil, W. Wibisono, and others, “An improved Quad and RDE-based medical data hiding method,” in *Computational Intelligence and Cybernetics (CYBERNETICSCOM), 2013 IEEE International Conference on*, 2013, pp. 141–145.

[10] M. Holil and T. Ahmad, “Secret data hiding by optimizing general smoothness difference expansion based method,” *J. Theor. Appl. Inf. Technol.*, vol. 72, no. 2, pp. 155–163, 2015.

[11] C.-N. Lin, D. J. Buehrer, C.-C. Chang, and T.-C. Lu, “Using quad smoothness to efficiently control capacity–distortion of reversible data hiding,” *J. Syst. Softw.*, vol. 83, no. 10, pp. 1805–1812, 2010.

[12] A. M. Alattar, “Reversible watermark using the difference expansion of a generalized integer transform,” *Image Process. IEEE Trans. On*, vol. 13, no. 8, pp. 1147–1156, 2004.

[13] “Partners Infectious Disease Images - eMicrobes Digital Library - Home.” [Online]. Available: http://www.idimages.org/. [Accessed: 27-Feb-2016].

[14] “Lorem Ipsum - All the facts - Lipsum generator.” [Online]. Available: http://www.lipsum.com/. [Accessed: 25-Feb-2016].