A Reversible Data Hiding Algorithm for Radiological Medical Images

Z. Jezabel Guzmán Zavaleta[1], Claudia Feregrino Uribe[2],
José Alberto Martínez Villanueva[3], René Cumplido[4]
National Institute for Astrophysics, Optics and Electronics, INAOE, MEXICO
{zguzman[1], cferegrino[2], jamartinez[3], rcumplido[4]}@inaoep.mx

Abstract— **In this paper we introduce a reversible data hiding algorithm for medical images. Using the advantages of some proved methods and the characteristics of radiological medical images we obtain a large embedding capacity with minimum distortion of the original image using an easy control for recovering the hidden data and the original image.**

Key words— **Reversible Data Hiding, Radiological Medical Images.**

I. Introduction

Data hiding is a form of steganography that embeds data into digital media for the purpose of identification, annotation and copyright [1].

The classical application of data hiding is secret communication. Data hiding is used in the context of privacy, providing anonymity for content providers on the World Wide Web, as a mechanism for device-to-device communication, for embedding arbitrary meta-data into images, for embedding alternative channels into television broadcast, etc [2].

Reversible data hiding is a particular kind of data hiding. Reversibility is the characteristic that the method can recover exactly the original host signal (without any distortion) upon extraction of the embedded information. For that reason, reversible (lossless) data hiding is widely used on sensitive imagery such as deep space exploration, military reconnaissance and medical diagnosis. Many emerging applications within sensitive imagery are in the health domain. There are many kinds of medical diagnosis studies in different subareas of health; however, most of them are radiological-related, around 70%, thus security alternatives suited for radiological images are required to guarantee the patient information.

Recently, more and more reversible data hiding methods have been proposed and some authors have presented overviews of those methods [3, 4]. However, not all the methods are suitable for medical images, that contain homogeneous regions, and the insertion produces several distortions on the embedding host.

In this paper, we propose a new reversible data hiding technique, suitable for radiological medical images, using the advantage of some proved methods, which can embed a significant amount of data while keeping high visual quality. The rest of the paper is organized as follows: in section 2 a general overview of some proved methods is presented, followed by the proposed method in section 3. Some experimental results are shown in section 4 and finally section 5 concludes.

II. Reversible Data Hiding Methods

Different methods for reversible data hiding are available. The most recent methods that are suitable for medical images, in spite of presenting many advantages, they have some disadvantages; some of them have a low embedding capacity, others high distortion, or in other cases, they require to save a large amount of data for the exact recovering and most of the publications, do not show results with medical images.

Based on the published results, we have chosen three distinctive reversible data hiding algorithms. We tested those methods with radiological medical images and we show their advantages and disadvantages.

A. Lee Algorithm

Lee *et al* [5], propose a reversible image authentication technique based on watermarking. Their method is quite simple and it produces a small distortion of the image by utilizing the difference between adjacent pixel values.

The algorithm scans the image, obtaining the difference between pairs of pixels, the odd-line and the even-line. Depending on the difference value, it can embed data or not, that is, if the difference is equal to -1 or 1 then it embeds a bit, in this case if the bit to be embedded is '0' then the difference value remains equal, otherwise the difference value changes to -2 or 2; if the difference value is equal to 0 then it does nothing, otherwise, it shifts the differences histogram. The equation 1 summarizes how the histogram modification is done.

In order to generate a watermark they combine the hash of the image with a binary logo image using the bit-wise XOR operation.

The process for recovering the embedded data is quite similar; first it scans the image (in the same order than the hiding method) and obtains the differences between each pair of pixels. If the difference is greater than 2 or less than -2 then the algorithm shifts back the histogram. If the difference is equal to 2 or -2 then it obtains a bit '1' of the hidden message and subtracts one to the odd-line pixel. If the difference is equal to 1 or -1, it obtains a

IEEE Catalog Number: CFP08827-CDR
ISBN: 978-1-4244-2499-3
Library of Congress: 2008903800

bit '0' of the hidden message and the odd-line pixel remains without changes.

$$I_S(i,2j+1)=\begin{cases} I(i,2j+1)+1 \text{ if } D(i,j)\geq 2 \\ I(i,2j+1)-1 \text{ if } D(i,j)\leq -2 \\ I(i,2j+1)+b \text{ if } D(i,j)=1 \\ I(i,2j+1)-b \text{ if } D(i,j)=-1 \\ I(i,2j+1) \text{ otherwise} \end{cases} \quad (1)$$

Where, $I(i,j)$ is the original image of size MxN pixels; $D(i,j) = I(i,2j+1) - I(i,2j)$; b is the bit to be embed '1' or '0'. Then I_S is the stego image (the modified image).

The advantages of this algorithm are: the easy control for recovering the original image and the embedded data; it does not need to save a location map or any other information; the PSNR (Peak-Signal-to-Noise-Ratio) of the watermarked image is greater than or equal to 51 dB, with adequate capacity for addressing many applications; this algorithm is useful for homogeneous images. However, authors do not suggest how they manage underflow and overflow problems that may exist.

B. Ni Algorithm

Ni *et al* [6] proposed a lossless data embedding technique that utilizes the zero or the minimum points of the histogram of an image and modifies the pixel grayscale values to embed data into the image. The complexity of their algorithm is low, and the PSNR between the original and the marked image is greater than 48 dB.

Ni's method first makes the histogram of the image and then checks where the histogram has a zero value, which is called a *zero point* (Z), and then it founds a *peak point* (P), the maximum value of the histogram. The equation 2 shows the embedding process.

The detection algorithm needs the Z and P values used. First, the image is scanned. If $Z<P$ then all pixels with value P indicate that the embedded data bit is '1' (they should not be modified) and pixels with value $P-1$ indicate a bit '0' and it needs to be increased by one. Then it shifts back all values between Z and P-2 (increasing by one). If $Z>P$ all pixels with value P indicate that the embedded bit is '0' (they should not be modified) and pixels with value $P+1$ indicate a bit 1 and it needs to be reduced by one. Then it shifts back all values between P+2 and Z (reducing by one).

Fallahpour *et al* [7] proposed an improvement over Ni's algorithm, they propose to divide the image into n-blocks and run the algorithm over each block, and also find more pairs of Z and P points for each block, increasing so the embedded capacity.

$$I_S(i,j)=\begin{cases} I(i,j)-1 \text{ if } Z<P \ \& \ I(i,j) > Z \ \& \ I(i,j)<P \\ I(i,j)+1 \text{ if } Z>P \ \& \ I(i,j) < Z \ \& \ I(i,j)>P \\ I(i,j) + b\text{-}1 \text{ if } Z<P \ \& \ I(i,i)=P \\ I(i,j) + b \text{ if } Z>P \ \& \ I_s(i,i)=P \\ I(i,j) \quad \text{otherwise} \end{cases} \quad (2)$$

Where, $I(i,j)$ is the original image of size MxN pixels; Z is the zero point; P is the peak point and b is the bit to be embed '1' or '0'. I_S is the stego image.

In the recovering process, both algorithms need to save the Z and P values. Fallahpour method also needs to save the number of blocks including the pairs of Z and P points for each block. If the algorithm does not find any zero value on the image histogram, then it chooses the minimum value for Z, and also needs to save the minimum value. However, it is easy to address that information.

C. Thodi Algorithm

Thodi *et al* [8] proposed a reversible watermarking algorithm for digital images. It embeds information into prediction errors of adjacent pixels using a MED Predictor (Medium Edge Predictor).

The value predicted and its prediction error is given by (3). Consider a pixel with value x and its context; the pixel to its right, bottom and bottom right.

$$\boldsymbol{x_p}=\begin{cases} \max(a,b) & \text{if } c\leq\min(a,b) \\ \min(a,b) & \text{if } c\geq\max(a,b) \\ a+b-c & \text{otherwise} \end{cases} \quad (3)$$

Where x is the current pixel, x_p is the predicted value and a,b,c are its context as shown.

x	b
a	c

Then the prediction error is

$$\boldsymbol{p_e = x - x_p}$$

The value of the prediction error p_e and the pixel intensity x determine which locations can be embedded by prediction error expansion. When the p_e is less than a threshold T and after embedding a bit there is neither underflow nor overflow then that location can be used for embedding a bit i using (4), all those locations are the set E.

IEEE Catalog Number: CFP08827-CDR
ISBN: 978-1-4244-2499-3
Library of Congress: 2008903800

$$p_e' = 2p_e + i$$
$$x' = x_p + p_e' = x + p_e + i \quad (4)$$

i is the bit to be embedded and
x' is the modified value

The set N_e contains all locations which $p_e > T$. Other set, U_e, contains the locations in E and N_e that can cause ambiguity at the decoder, that is, if the modified value (after embedding a bit with prediction error expansion) is in $(0,T\text{-}1]$ or is in $[L\text{-}1\text{-}T, L\text{-}1)$, where T is the threshold and L is the representative pixel intensities; for those cases the algorithm needs flag bits. The set U contains all locations that can not be changed and also need flag bits for being recognized. Because the algorithm needs flag bits for the lossless extraction process, it reserves some locations for embedding those extra bits.

The embedding algorithm proceeds as follows:

1. The image is preprocessed to separate the reserve region R of the region S, that is, the rest of the image; region R is going to be used for the extra bits and region S for embedding the bit stream. It has to save the LSB of every pixel on region R, called R_{LSB}
2. The bitstream B is formed by combining a payload Py, an end-of-payload indicator, EOP, and R_{LSB} as $B = Py \cup EOP \cup R_{LSB}$
3. For every pixel in region S (without the last row and column), it obtains the p_e, and then it selects the set it belongs to, E, N_e or $U = S - E - N_e$
   - If it belongs to E then embed a bit of the beginning of B and then drop that bit. If also it belongs to U_e append a flag bit '1' at the beginning of the bitstream B
   - If it belongs to N_e then shift right by T positions when $p_e \geq 0$ and shift left by T-1 when $p_e < 0$; in that way the decoder will can distinguish the embedded locations. If also it belongs to U_e append a flag bit '1' at the beginning of the bitstream B
   - Else (it belongs to U) it appends a flag bit '0' at the beginning of the bitstream B
4. If some bits remain in B then those bits have to be embedded in the LSB's of R

At the decoder, the watermarked image has to be scanned in reverse order than the embedding process. Because the decoder extracts the bit embedded and restores the image, it is important to restore the original pixel intensity value at the current location before proceeding to the next location.

1. Compute the prediction error value p_e of the current location:
   - If that location belongs to $U_e \cup U$ then the flag bit is dropped from the bitstream. If the bitstream is null, the flag bit is obtained from the LSBs of the pixels at the start of the image
   - If it belongs to $E\text{-}U_e$ then the embedded bit extraction and restoration are done
   - If it belongs to $N\text{-}U_e$ then restore the original value

The advantage of this algorithm is a great embedding capacity with minimum distortion of the image. However, the disadvantages are the necessity of control flag bits, and the preprocessing by reserving a region for extra bits, that increases the execution time of the algorithm with large images.

Other quite similar methods are Tian's algorithm [9] and Kallel's improved of Tian's method [10] that uses difference expansion for embedding. The main disadvantage of those methods is the large location map that has to be saved for the lossless recovering. Thodi *et al* [11], made a comparison between his own method versus Tian's method obtaining similar results. The advantage of Thodi's method over their own improved Tian's method is the use of flag bits to recover the image versus a large location map.

III. PROPOSED METHOD

We propose to use the main advantages of the examined methods and the characteristics of radiological medical images, with the aim of obtaining a large embedding capacity with minimum distortion of the image using an easy control for the exact recovering. We describe the proposed method in the next sections.

Thodi algorithm scans the image and selects the best locations for the embedding process, generates less distortion using prediction error expansion; the challenge is to do it without ambiguity at the recovering process and without saving control bits. In medical images the pixel intensity values are commonly homogeneous; with that premise, we select the location with the minor difference between adjacent pixels as Lee algorithm. In that manner we embed data only at locations with prediction error values equal to 1 or -1. To avoid underflow or overflow problems, we suggest the use of a pre-processing step based on Ni histogram shifting.

In the embedding process, first at all, we make a pre-processing step where we eliminate the underflow and overflow problems. We obtain the image histogram and then we find the zero points on the histogram boundaries (Z_i and Z_L) and then we make a zero map. This process is explained in the next subsections. Once we eliminate any problem, we embed the bitstream B into the image.

IEEE Catalog Number: CFP08827-CDR
ISBN: 978-1-4244-2499-3
Library of Congress: 2008903800

A. Underflow and Overflow Problem

The proposed method utilizes prediction error values on the image and modifies pixel values slightly to embed data. With those modifications, it is necessary to manage any problem of underflow/overflow. We stretch the histogram of the image by removing any pixel value that, after some modification, can cause underflow or overflow as follows:

1. For a grayscale image *I(i,j)* of size *MxN* pixels, obtain the histogram of the pixel values of the image
2. Find the first and the last zero points (Z_i and Z_L) on the histogram (see Ni method). Save Z_i and Z_L values. If do not exist any zero values on the histogram then utilize the minimum values and also save the pixel locations of those values
3. Make the stretch. With every pixel of the image,
   - If $I(i,j) < Z_i$ then shift to the right that is, $I(i,j)+1$
   - If $I(i,j) > Z_L$ then shift to the left that is, $I(i,j)-1$
   - Otherwise do nothing
4. Form the bitstream appending the Zero Map (*ZM*), the payload (*Py*) to be embedded and an end of payload flag (EOP):

$$B = ZM \cup Py \cup EOP$$

B. The Zero Map

The zero-map, *ZM*, is needed for the lossless recovery process. We append the zero-map at the beginning of the bitstream to be hidden. In case the histogram does not have zero values then we take the minor value as a zero point at each boundary. The first two bits of the map are used to identify zero points, the next two represent the bit depth of the image and next bits are used for the zero point values (Z_i and Z_L). When the zero point is the minor value different from zero then we save the values of the zero points (K_i / K_L) and its location inside the image (K_i/K_L x BitDepht) guarantying the lossless recovering. The *ZM* is showed in figure 1.

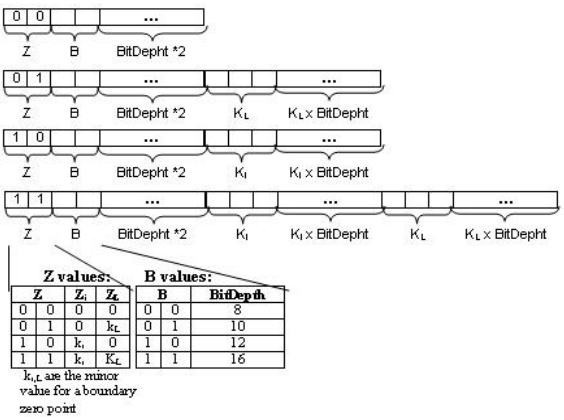


Figure 1: the Zero Map

C. Embedding Process

This is the process to embed a bitstream *B* into an Image *I*, obtaining the modified image I_S with the same dimensions as the original. The last row and the last column of the image must not be used for embedding purposes; those pixels are going to remain without changes for the exact recovery of the original image. A region of 2x2 is a pixel with value x and its context: the pixel in its right, bottom and bottom right.

1. Eliminate the underflow/overflow problem in the image (explained before)
2. In an determined order (left to right and up to bottom),we scan the image *I(i,j)* where *i*= 0 to *M*-1 and *j*=0 to *N*-2 and with every pixel do:
   - If *i* is equal to *M*-1 or if *j* is equal to *N*-1 then $I_S(i,j) = I(i,j)$
   - Else, obtain the prediction error value p_e of *I(i,j)* and its context showed in (3),
   - If $p_e = 1$ then embed a bit *b* from the bitstream *B*, that is $I_S(i,j) = I(i,j) + b$
   - Else if $p_e = -1$ then embed a bit *b* from the bitstream *B*, that is $I_S(i,j) = I(i,j) - b$
   - Else if $p_e \geq 2$ then make a right shift, that is $I_S(i,j) = I(i,j) + 1$
   - Else if $p_e \leq -2$ then make a left shift, that is $I_S(i,j) = I(i,j) - 1$
   - Otherwise $I_S(i,j) = I(i,j)$

D. Extract and recovery process

The exact recovery process is inverse to the embedding process. In recovery process, it has to extract the embedded bits in the image in inverted order than they were hidden, i.e. if the embedding process was made in left to right and up to bottom order then the inverse process is going to be bottom to up and right to left order. It obtains the original image *I(i,j)* from $I_S(i,j)$ following the next steps:

1. Scan the image in reverse order and with every pixel of $I_S(i,j)$ (where *i*= *M*-2 to 0 and *j*=*N*-2 to 0) do:
   a. Obtain the prediction error value p_e of $I_S(i,j)$ and its context showed in (3),
      - If $p_e = 1$ or $p_e = -1$ then the embedded bit *b* was '0', append a '0' at the beginning of the recovered bitstream B_R and $I(i,j) = I_S(i,j)$
      - Else if $p_e = -2$ then the embedded bit *b* was '1', append a '1' at the beginning of the recovered bitstream B_R and $I(i,j) = I_S(i,j)+1$
      - Else if $p_e = 2$ then embedded bit *b* was '1', append a '1' at the beginning of the recovered bitstream B_R and $I(i,j) = I_S(i,j)-1$
      - Else if $p_e > 2$ then make a left shift, that is $I(i,j) = I_S(i,j) - 1$
      - Else if $p_e < -2$ then make a right shift, that is $I(i,j) = I_S(i,j) + 1$

IEEE Catalog Number: CFP08827-CDR
ISBN: 978-1-4244-2499-3
Library of Congress: 2008903800

- Otherwise $I(i,j) = I_S(i,j)$

After recovering the embedded bitstream it needs to decode the zero map from the beginning of B_R, see figure 1. Once it has the Z_I and Z_L values and if it is necessary the location map, it has to stretch back the histogram of the image I to recover the original image.

E. Other Improvements

In this method we use difference values equal to 1 and -1 for embedding data into images; however, we can use the specific characteristics of the radiological medical images to obtain more embedding capacity. One characteristic is that medical images have pixel intensities regularly homogeneous. Other characteristic is that the image perimeter is commonly black and the region of interest for the physician is the center of the image, then an alteration of the image perimeter would not alter the medical diagnostic. With these antecedents, we propose to use the slight modification of using black 2x2 regions of the image to hide data; i.e. if the context of a referred pixel is a black region and if the referred pixel is also black then we embed a bit on that pixel, else we shift its value to the right. This simple change may increase significantly the embedding capacity on this kind of images.

If higher hiding capacity is required, we can use a second process to hide data into the embedded image. In the recovery process, first we obtain the last hiding payload and then we recover the original embedded image, after this, we follow again all the steps for obtaining the first hiding payload and recover the original image, finally we concatenate both payloads to obtain the total embedded bits.

IV. Software Results

Even though the published hiding methods show interesting results, they do not provide any result on medical images, so we analyze their performance on some radiological images and we compare the results against our method. We use the PSNR and the hiding capacity metrics. The PSNR reflects the average error on the stego image. We use some medical gray scale images in DICOM format [12] and we hide the standard text *Alice,* from the Canterbury corpus [13], as a bitstream into each image, with a total of 149 KB.

In the tables 1, 2 and 3 we show the obtained results for different images, the labels represent the different methods.

Lee: represent the results for the Lee method
Ni: using the Ni improved method with 4 blocks
Thodi: using the Thodi method with threshold parameter of 3
P1: are the results with the Lee algorithm plus shift histogram eliminating underflow and overflow problems
P2: the proposed algorithm
P3: P2 and hiding data in black regions
P4: P3 with 2 embedding layers, i.e. we insert the data into the original image, and then we insert data again into the embedded image.

The methods were programmed with Matlab 7.4 R2007a. In the tables, we show the hiding capacity measured in bits and the control bits required for every method, the difference between them gives the real hiding capacity. The rows of underflow and overflow show the number of pixels with these problems. Also we measure the execution time in every method.

TABLE 1: IMAGE1 RESULTS

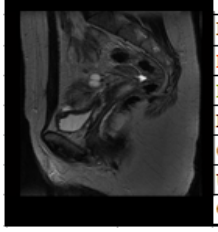
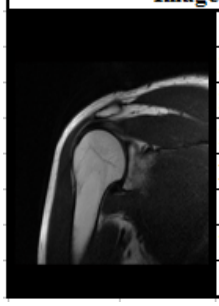
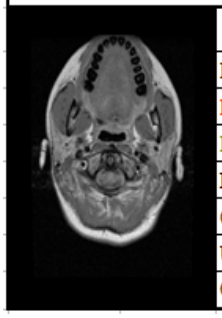
Image 1: MR-MONO2-12-an2.dcm 256x256x12							
	Lee	Ni	Thodi	P1	P2	P3	P4
Hiding Cap (bits)	467	8308	10309	467	1112	8762	13572
Real Capacity (bits)	467	7918	2467	439	1084	8734	13486
PSNR (dB)	75.91	81.68	65.48	74.85	72.36	72.23	65.99
ExecTime (seg)	0.01	0.08	22.09	0.05	0.09	0.09	0.27
Overflow (pixels)	0	0	0	0	0	0	0
Underflow (pixels)	256	0	0	0	0	0	0
Control (bits)	0	390	7842	28	28	28	86

TABLE 2: IMAGE2 RESULTS

Image 2: MR-MONO2-12-shoulder.dcm 1024x1024x12							
	Lee	Ni	Thodi	P1	P2	P3	P4
Hiding Cap (bits)	22357	226762	270013	22357	62707	278482	453165
Real Capacity (bits)	22357	226564	19367	22329	62679	278454	453079
PSNR (dB)	76.57	72.79	67.21	70.87	69.95	69.76	63.56
ExecTime (seg)	0.10	0.53	15222.36	0.47	1.46	1.56	3.21
Overflow (pixels)	0	0	0	0	0	0	0
Underflow (pixels)	2733	0	0	0	0	0	0
Control (bits)	0	198	250646	28	28	28	86

IEEE Catalog Number: CFP08827-CDR
ISBN: 978-1-4244-2499-3
Library of Congress: 2008903800

TABLE 3: IMAGE3 RESULTS

Image 3: brain_001.dcm 256x256x16							
	Lee	Ni	Thodi	P1	P2	P3	P4
Hiding Cap (bits)	1203	16291	18114	1203	2775	18085	27839
Real Capacity (bits)	1203	15901	1074	1167	2739	18049	27729
PSNR (dB)	100.80	98.31	91.44	95.00	94.08	93.90	87.65
ExecTime (seg)	0.01	0.23	55.04	0.02	1.45	1.53	3.34
Overflow (pixels)	0	0	0	0	0	0	0
Underflow (pixels)	254	0	0	0	0	0	0
Control (bits)	0	390	17040	36	36	36	110

The PSNR is given by (5); values over 36 dB in PSNR are acceptable in terms of degradation, which means no significant degradation is observed by the human eye [14].

$$PSNR = 10*\log_{10}\left(\frac{(MaxValue)^2}{MSE}\right)$$

$$MaxValue=(Bits-1)^2 \qquad (5)$$

$$MSE=\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(I_I(i,j)-I_E(i,j))^2$$

Where MaxValue is the maximum representative value; MSE is the Mean Squared Error, I_I is the original image and I_E is the watermarked image.

The Thodi method shows a good hiding capacity performance without visible distortions on the medical image. One disadvantage is the increased execution time on big images because the processing time required to find the hiding locations. Other disadvantage is the great number of bits that it needs to save inside the image for the exact recovery, control bits diminishing the real hiding capacity. Lee method has an easy way to hide data and recover the image, the disadvantage is the poor hiding capacity compared with the other methods, as we can see in the tables. The P1 algorithm requires a few control bits for the exact recovering, however P1 eliminates the underflow and overflow problems that Lee method shows, with a slight more distortion.

With the P2 algorithm we obtained better results than the Lee method but with the P3 we can reach and exceed the other methods results. With P4 we show that we can increase even more the real hiding capacity without visible distortions on the image, also without underflow or overflow problems and recovering the same original image.

VI. CONCLUSION

We tested the mentioned methods (Lee, Ni and Thodi) with some medical images showing their performances. We proposed a new reversible method. Its main characteristics are based on others proved techniques. We reach and exceed the performance of the others tested methods, given an easy way to hide a large payload without visible distortions, with high levels of PSNR values, and with a simple solution to the underflow and overflow problem.

REFERENCES

[1] Information Hiding: First International Workshop, R. J. Anderson, Editor, Lecture Notes in Computer Science 1174, Isaac Newton Institute, Cambridge, England, Springer-Verlag May 1996.

[2] Bender W. et al. Application for Data Hiding. IBM Systems Journal, Vol. 39, Nos. 3&4, 2000, pp 547-568.

[3] Awranjeb M. An Overview of Reversible Data Hiding. International Conference on Computer and Information Technology (ICCIT) Dec 19-21,2003. Jahangirnagar University. Bangladesh, pp 75-79.

[4] Y. Hu, B. Jeon, Z. Lin, H.Yang. Analysis and Comparison of Typical Reversible Watermarking Methods. Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, 2006.c Springer-Verlag Berlin Heidelberg, pp. 333–347.

[5] Lee S-K, Suh Y-H, Ho Y-S. Reversible Image Authentication Based on Watermarking. IEEE International Conference on Multimedia & Expo (ICME) 2006, Canada, pp 1321-1324.

[6] Ni Z., Shi Y., Ansari N., and Su W., "Reversible data hiding," *Proc. ISCAS 2003*, vol. 2, pp. 912–915.

[7] Fallahpour M, Sedaaghi M. "High Capacity lossless data hiding based on histogram modification". IEICE Electronic Express, Vol. 4, No. 7, April 10, 2007, pp. 205-210.

[8] Thodi D.M. and Rodriguez J. J., "Prediction-error-based reversible watermarking," in *Proc. IEEE Conf. Image Processing*, Oct. 2004, pp. 1549–1552.

[9] Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. On Circuits and Systems for Video Technology. Vol. 13, No. 8, Aug. 2003, pp. 890-896.

[10] Kallel I., Bouhlel M.S, Lapayre J.C. Improved Tian's Method for Medical Image Reversible Watermarking GVIP Journal, Volume 7, Issue2, August 2007

[11] Thodi DM, Rodriguez J. Expansion Embedding Techniques for Reversible Watermarking. IEEE Transactions on Image Processing. Vol 16, No 3, March 2007

[12] S. Barré: Medical Image Samples:
http://barre.nom.fr/medical/samples/
and MATLAB Central File Exchange:
http://www.mathworks.ch/matlabcentral/fileexchange/loadFile.do?objectId=2762&objectType=FILE

[13] The Canterbury Corpus: http://corpus.canterbury.ac.nz/

[14] Xuanwen Luo, Qiang Cheng, Joseph Tan, "A Lossless Data Embedding Scheme For Medical in Application of e- Diagnosis," *Proceedings of the 25th Annual International Conference of the IEEE EMBS Cancun*, Mexico. September 17-21, 2003.

IEEE Catalog Number: CFP08827-CDR
ISBN: 978-1-4244-2499-3
Library of Congress: 2008903800

978-1-4244-2499-3/08/$25.00 ©2008 IEEE