LOSSLESS DATA HIDING FOR MEDICAL IMAGES WITH PATIENT INFORMATION

Sang-Kwang Lee, Seong-Jae Lim*, Young-Ho Suh*, and Yo-Sung Ho***

*Electronics and Telecommunications Research Institute (ETRI)
161 Gajeong-Dong, Yuseung-Gu, Daejeon 305-700, KOREA
**Gwangju Institute of Science and Technology (GIST)
1 Oryong-Dong, Buk-Gu, Gwangju 500-712, KOREA
{sklee, sjlim, syh}@etri.re.kr, hoyo@gist.ac.kr

ABSTRACT

This paper presents a lossless data hiding algorithm for medical images, where we embed the patient information into the segmented liver region of the CT image. This algorithm utilizes the characteristics of difference images and modifies pixel values slightly to embed a large amount of data while keeping high visual quality.

Index Terms— Lossless data hiding, biomedical image processing

1. INTRODUCTION

In order to protect the intellectual property rights, we can apply information hiding techniques in various application areas, such as broadcast monitoring, proof of ownership, transaction tracking, content authentication, copy control, and device control [1]. In recent years, various lossless data hiding techniques have been proposed for images. The data embedding can take place in the spatial domain [2][3], or in the transform domain [4][5]. Ni *et al.* [3] proposed a reversible data embedding technique, which utilizes the zero or the minimum point of the histogram and modifies pixel values to embed the data. It can embed more data than other existing lossless data hiding algorithms. However, gray-level values of the zero point and the peak point should be transmitted to the receiving side for data retrieval.

In this paper, we propose a lossless data hiding algorithm for medical images, which can embed more data than other lossless data hiding algorithms. After we perform automatic segmentation of the liver in abdomen CT images using a priori knowledge and a watershed algorithm based on the morphological filters [6], we embed the patient information, such as patient details, history and measured physiological signals prior to transmission and storage [7], into the segmented liver region. The proposed data hiding algorithm utilizes the characteristics of difference images from the liver region and modifies pixel values slightly to embed the data. It can embed as much data as Ni's algorithm and the PSNR values of the marked images are higher than 51.14 dB [8]. Moreover, the proposed scheme is quite simple and the execution time is rather short.

2. PROPOSED DATA HIDING ALGORITHM

The entire framework of the the proposed embedding scheme is illustrated in Fig.1.

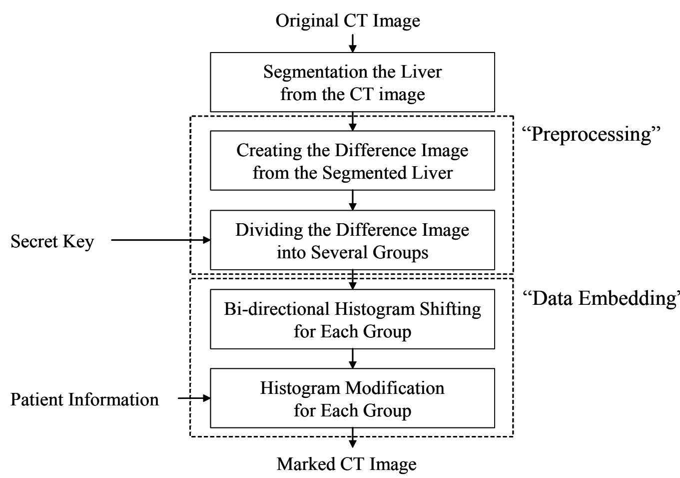
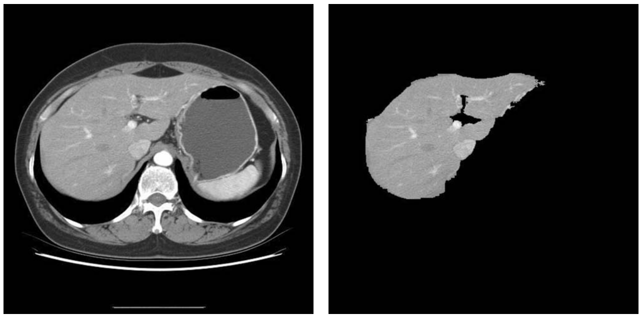


Fig. 1. Data embedding

2.1. Automatic Segmentation of the Liver in CT Images

First, we simplify the input CT image by using two-level thresholding based on general features and statistical information of livers. Then multiscale morphological filtering detects the initial liver boundary that is used to be the marker for a watershed algorithm. Lastly, an immersion-based watershed algorithm segments each region by detecting the watershed lines through the immersion of the surface from the minima [6]. Fig. 2 shows the segmented liver region from the CT image.



(a) Original CT image (b) Segmented liver region

Fig. 2. Automatic segmentation of the liver in CT images

2.2. Data Embedding

After liver segmentation, we embed the patient information into the segmented liver region. Before the data are embedded, the preprocessing for data embedding is performed as follows:

1. The segmented liver region is divided into two interlaced images: one consists of odd lines (odd-line image), the other even lines (even-line image). Each interlaced image has half the raster lines of the segmented liver region.

2. In order to obtain the difference image from the liver region, we subtract the even-line image from the odd-line image.

3. Pixels of the difference image are randomly divided into several groups. This grouping operation forms the secret key of the embedding algorithm.

We perform the data embedding procedure for each group independently. In this paper, we divide pixels of the difference image into two groups, Group A and Group B. In order to embed data, we modify pixel values of the odd-line image slightly. The data embedding for Group A is performed as follows:

1. We first generate the histogram of Group A indicated by $N_A(k), k_n \leq k \leq k_p$.

2. We utilize the difference values of -1 and 1 for the data embedding. To do this, we empty the difference values of -2 and 2 by bi-directional shifting in the histogram. If the difference value is less than or equal to -2, then subtracting by 1 from the odd-line image. If greater than or equal to 2, then adding by 1 to the odd-line image. Finally, the shifted histogram, $N_{As}(k), k_n - 1 \leq k \leq k_p + 1$, can be summarized as follows:

$$N_{As}(k) = \begin{cases} N_A(k+1) & \text{if } k_n - 1 \leq k < -2 \\ 0 & k = -2 \\ N_A(k) & k = -1, 0, 1 \\ 0 & k = 2 \\ N_A(k-1) & 2 < k \leq k_p + 1 \end{cases} \quad (1)$$

3. The whole difference image is scanned again. Once a pixel with the difference value of -1 or 1 is encountered, we check the data to be embedded. If the bit to be embedded is "1", we move the difference value of -1 to -2 (subtracting by 1 from the odd-line image) or 1 to 2 (adding by 1 to odd-line image). For instance, when a pixel with the difference value of -1 is encountered and the bit to be embedded is "1", the marked histogram, $N_{Ae}(k), k_n - 1 \leq k \leq k_p + 1$, is given by

$$\begin{cases} N_{Ae}(-1) = N_{Aep}(-1) - 1 \\ N_{Ae}(-2) = N_{Aep}(-2) + 1 \end{cases} \quad (2)$$

where $N_{Aep}(k)$ is the previous marked histogram. When a pixel with the difference value of 1 is encountered and the bit to be embedded is "1", the marked histogram is as follows:

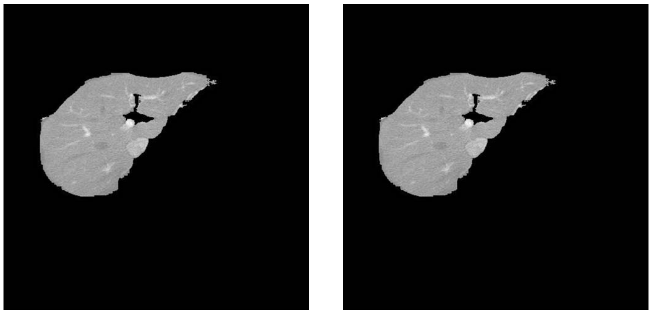
$$\begin{cases} N_{Ae}(1) = N_{Aep}(1) - 1 \\ N_{Ae}(2) = N_{Aep}(2) + 1 \end{cases} \quad (3)$$

4. If the bit to be embedded is "0", we skip the pixel of the difference image until a pixel with the difference value -1 or 1 is encountered. In this case, there is no change in the histogram, so it can be expressed as $N_{Ae}(k) = N_{As}(k)$.

Similarly, the data embedding for Group B is performed by the above procedure. It merely differs in that we utilize the difference values of -3 and 3 instead of -1 and 1 for the data embedding and empty the difference values of -4 and 4 instead of -2 and 2. In this way, we complete the data embedding procedure. Fig. 3 shows the result of the data embedding for the segmented liver region. It is seen from this figure that there is no visible degradation due to embedding in the marked liver region.

The capacity of this algorithm equals to the sum of the numbers of pixels with the difference values of -1 and 1 in Group A and -3 and 3 in Group B. For instance, in the experiment with the CT image of Fig. 2(a), the numbers of pixels with the associated difference values are 2,568, 3,936, 3,886, and 2,893, respectively. Hence, the capacity is 13,283 bits.

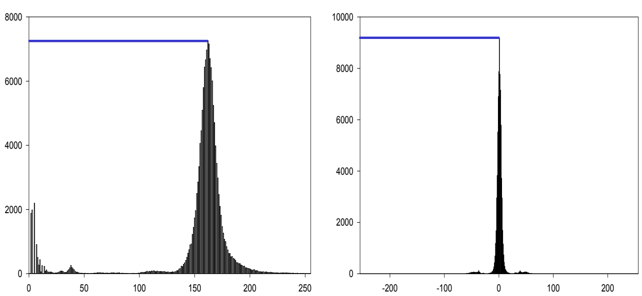
A large number of pixels of difference images have a tendency to be distributed around 0. Using this property of difference images, we can embed more data as compared to the image itself. Fig. 4 shows this characteristic property of difference images. The number of pixels with the peak point



(a) Segmented liver region (b) Marked liver region

Fig. 3. Result of the data embedding

in the histogram of the segmented liver region image of Fig. 2(b) is around 7,000. On the other side, the number of pixels with the peak point (the difference value of 0) in the histogram of the difference image from the segmented liver region is higher than 9,000. In the case of typical grayscale images such as Lena and Airplane image, this property of difference images is obvious. For Lena image, the number of pixels with the peak point in the histogram of the image is around 3,000. However, the number of pixels with the peak point in the histogram of the difference image from the image is around 15,000.



(a) Segmented liver image (b) Difference image

Fig. 4. Histogram characteristics of difference images

In the worst case, all pixels of odd-line image will be added or subtracted by 1, implying that MSE (mean squared error) is 0.5. Hence, the PSNR of the proposed scheme can be calculated as follows:

$$PSNR(dB) = 10 \log_{10} \frac{255^2}{MSE} \approx 51.14 \tag{4}$$

In short, the lower bound of the PSNR values of marked images are 51.14 dB. This result is much higher than other lossless data hiding algorithms.

2.3. Data Retrieval and Recovery

We retrieve the patient information from the segmented liver region and recover the original CT image exactly. Data retrieval and recovery process is shown in Fig. 5. This process is performed by the dual of data embedding described in Section 2.1 and 2.2.

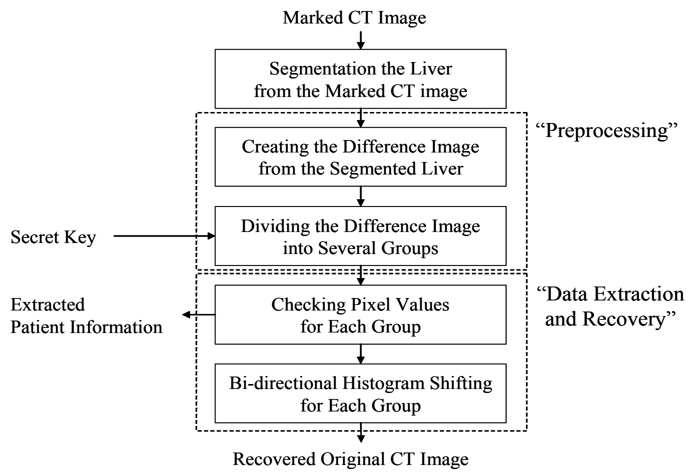


Fig. 5. Data retrieval and recovery

After we perform segmentation of the liver in the marked CT image, we divide the marked liver region into the odd-line image and the even-line image. In order to obtain the difference image from the segmented liver region, we subtract the even-line image from the odd-line image. Using the secret key, pixels of the difference image are randomly divided into Group A and Group B. And then we perform the data retrieval procedure for each group independently. The data retrieval and recovery procedure for Group A is performed as follows:

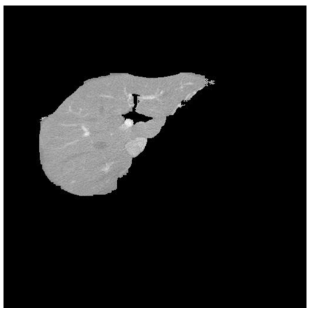
1. We generate the histogram of Group A indicated by $N'_{Ae}(k), k_n - 1 \leq k \leq k_p + 1$.
2. The whole difference image is scanned. If the pixel with the difference value of -1 or 1 is encountered, the "0" is retrieved. If the pixel with the difference value of -2 or 2 is encountered, the "1" is retrieved. In this way, the hidden data can be retrieved.
3. The whole difference image is scanned once again. If the difference value is less than or equal to -3, then adding by 1. If greater than or equal to 3, then subtracting by 1. The reconstructed histogram, $N_{Ar}(k)$, $k_n \leq k \leq k_p$, can be expressed as follows:

$$N_{Ar}(k) = \begin{cases} N'_{Ae}(k-1) & \text{if } k_n \leq k < -1 \\ N'_{Ae}(k-1) + N'_{Ae}(k) & k = -1 \\ N'_{Ae}(k) & k = 0 \\ N'_{Ae}(k) + N'_{Ae}(k+1) & k = 1 \\ N'_{Ae}(k+1) & 1 < k \leq k_p \end{cases} \tag{5}$$

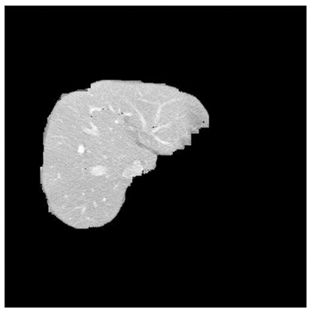
In this way, we can restore the original image without any distortion. Finally, $N_{Ar}(k) = N_A(k), k_n \leq k \leq k_p$. Similarly, the data retrieval for Group B is performed by the above procedure. For Group B, we check the difference values of -4, -3, 3, and 4 instead of -2, -1, 1, and 2.

3. EXPERIMENTAL RESULTS AND ANALYSIS

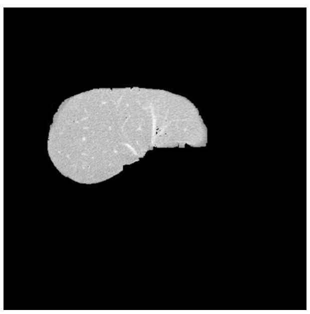
Fig. 6 shows the marked liver regions with the patient information with 5,152 bits added by the embedding algorithm described in Section 2.2. The hidden data cannot be perceptible in the marked liver region.



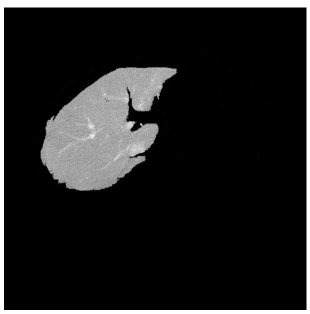
(a) CT1



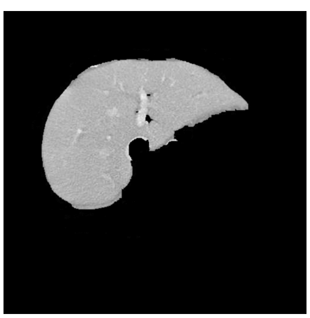
(b) CT2



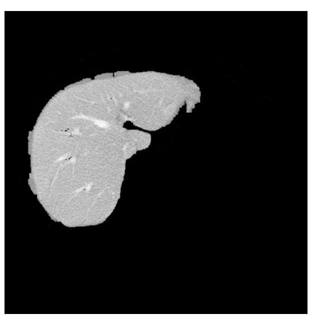
(c) CT3



(d) CT4



(e) CT5



(f) CT6

Fig. 6. Marked liver regions

Table 1 shows the PSNR values of marked CT images and the corresponding capacities. In our experiments, the PSNR values of all marked images are above 51.14 dB, as we theoretically proved in Section 2.2. On the other side, the PSNR of Ni's algorithm are higher than 48.13 dB. Thus, the PSNR values of marked images generated by the proposed scheme are much higher than almost all lossless data hiding algorithms.

Table 1. PSNR and capacity of the proposed scheme

CT images ($1024\times1024\times8$)	PSNR (dB)	Size of liver region (pixels)	Capacity (bits)
CT1	61.17	149,249	13,283
CT2	59.22	240,270	16,821
CT3	61.29	131,614	9,570
CT4	61.37	130,724	9,344
CT5	59.88	213,034	16,653
CT6	59.51	198,829	13,796

4. CONCLUSIONS

In this paper, we have described a high-capacity, low-distortion, and lossless data hiding algorithm for medical images. Lossless recovery of the original is achieved by using the histogram of the difference image. Experimental results show that the proposed algorithm can embed a large amount of data while keeping high visual quality. This algorithm embeds a patient information into the ROI in the CT image. Hence, it can be easily applied to invertible authentication for medical images.

5. REFERENCES

[1] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, 2001.

[2] J. Fridrich, M. Goldjan, R. Du, "Invertible authentication," *Proc. SPIE, Security and Watermarking of Multimedia Contents*, pp.197–208, 2001.

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

[4] B. Macq, F. Deweyand, "Trusted headers for medical images," *DFG VIII-D II Watermarking Workshop*, (1999)

[5] M. Goldjan, J. Fridrich, R. Du, "Distortion-free data embedding," *Proc. 4th Information Hiding Workshop*, pp. 27–41, 2001.

[6] S. Lim, Y. Jeong, C. Lee, and Y. Ho, "Automatic segmentation of the liver in CT images using the watershed algorithm based on morphological filtering," *Proc. SPIE, Medical Imaging*, pp.1658–1666, Feb. 2004.

[7] D. Anand, U. Niranjan, "Watermarking medical images with patient information," *Proc. ITAB-ITIS 2000* pp. 703–706, 1998.

[8] S. Lee, Y. Suh, and Y. Ho, "Lossless data hiding based on histogram modification," *PCM 2004*, LNCS 3333, pp. 340-347, Nov. 2004.