ISSN 2348 - 7968

DATA HIDING ON IMAGE BY USING PARTICLE SWARM OPTIMIZATION AND HISTOGRAM MODIFICATION

Ajin.M (1),Prabu.T(2) Parthiban.P(3)

[1]PG Scholar, Department of ECE, Nehru Institute of Eng & Tech
Coimbatore, Tamilnadu, India

[2]Assistant Professor, Department of ECE, Nehru Institute of Eng & Tech
Coimbatore, Tamilnadu, India

[3]Assistant Professor
Department of ECE, Nehru Institute of Eng & Tech
Coimbatore, Tamilnadu, India

Abstract

Steganography is used to hide a secret message within a cover image, thereby yielding a stegoimage such that even the trace of the presence of secret information is wiped out. This project proposes a novel prediction based reversible steganographic scheme based on image Inpainting. First, reference pixels are chosen by using Particle Swarm Optimization technique. Then, In order to generate a prediction image, the image Inpainting technique based on partial differential equations is introduced, which has similar structural and geometric information as the cover image. Finally, by using the two selected groups of peak points and zero points, the histogram of the prediction error is shifted to embed the secret bits reversibly. Since the same reference pixels can be exploited in the extraction procedure, the embedded secret bits can be extracted from the stego image correctly, and the cover image can be restored lossless. Through the use of the PSO algorithm and the inpainting predictor, the prediction accuracy is high, and more embeddable pixels are acquired. Thus, this project provides a greater embedding rate and better visual quality compared with recently reported methods.

Keywords: *Steganography, inpainting technique, Particle Swarm Optimization, cover image*

I. Introduction

Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement is called Steganography. It is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. Today Steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels. Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, Steganography focuses on keeping the existence of a message secret. Two their technologies that are closely related to steganography are watermarking and fingerprinting. In watermarking and fingerprinting the fact that information is hidden inside the files may be public knowledge – sometimes it may even be visible while in steganography the imperceptibility of the information is crucial.

Reversible steganography, also called reversible data hiding [1], has been studied extensively in recent years, especially for digital images. The property of reversibility means that the original form of the image, before the secret bits were embedded, can be recovered completely after the embedded bits are extracted. Reversible data hiding can be used for medical, military, and legal applications, which do not allow any modification in the digital representation of the cover image due to the risk of misinterpretations. In general, there are two main categories of reversible data hiding methods for images, i.e., methods based on difference expansion and methods based on histogram shifting.

In 2003, Tian proposed a reversible data hiding method based on difference expansion [1]. In his work, the cover image was divided into a series of non overlapping, neighboring pixel pairs, and the difference of each pixel pair was doubled. Then, the doubled difference was either kept reserved or modified according to the parity of the embedding secret bit. Ni *et al* presented a method based on histogram shifting to embed secret data reversibly [2]. The peak point of the image histogram was selected and the pixel values in the range from its right one to the zero point were increased by one to create one vacant histogram bin for embedding. The number of secret bits that could be embedded depended on the pixel number of the peak point in the histogram. The information of the peak point and zero point was required in the procedure of extracting the embedded data and recovering the cover image. Tai *et al* introduced a binary tree structure that can be utilized to predetermine the peak point used for embedding secret messages [3]. Consequently, only the level of the binary tree must be shared by the sender and the receiver. Recently, many researchers have proposed reversible data hiding methods based on the prediction mechanism; these methods are extensions of the method based on difference expansion and the method based on histogram shifting. The key idea of the prediction-based method is that the prediction process is conducted first to estimate the cover image pixels, and the prediction error, i.e., the difference between the cover image and the prediction result, is used to embed the secret data by difference expansion or histogram shifting. The consistency of the prediction results in the embedding and extracting procedures ensures the correctness of the extraction of the secret bits and the recovery of the cover image. Thodi and Rodriguez indicated that the cover image can be divided into its prediction result and the corresponding prediction error [4]. The predictor they used was a low-complexity algorithm with an inherent edge detection mechanism. The prediction error was expanded according to the embedding data and combined with the prediction result to produce the stego image. Hong and Chen tried several interpolation techniques, such as bilinear interpolation and bi-cubic interpolation, to predict the cover image according to the chosen reference pixels. Then, they shifted the histogram of prediction error to embed the secret data. In 2012, Chuan Qin, Chin-Chen Chang presented a paper on which the reference image get selected by using adaptive distribution characteristics of the image. In order to increase the computational speed there need another algorithm which this paper is going to try fully in this article [5].

The rest of this paper is organized as follows. Section II describes our proposed reversible steganographic scheme, including the strategy for choosing reference pixels, the inpainting based pixel prediction, and the procedures of embedding, extraction, and recovery. Experimental results and comparisons are given in Section III and Section IV concludes this paper.

II. Proposed scheme

In our scheme, fewer reference pixels are chosen in the smooth regions of the cover images, while more reference pixels are chosen in the complex regions. This strategy of choosing reference pixels can produce a greater number of possible embeddable pixels and avoid more distortion caused by the embedding procedure. According to the chosen reference pixels, the PDE-based inpainting algorithm can effectively generate the prediction image that has the similar structural and geometric information as the cover image. In order to further achieve higher hiding capacity and less distortion, two groups of peak points and zero points are selected from the histogram of prediction error to embed the secret bits by the shifting operation. The flowchart of the embedding procedure is shown in Fig. 1. The extraction and recovery procedures are approximately the reverse process of the embedding procedure.

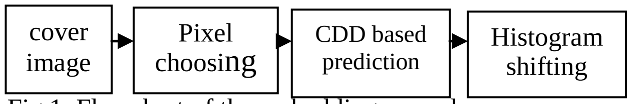

Fig.1. Flowchart of the embedding procedure.

A. PSO Algorithm

PSO algorithm is a population-based search algorithm based on the simulation of the social behavior of birds within a flock. In PSO, each single solution is a “particle”[6]. All of the particles have fitness values which are evaluated by the objective function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the personal and global best particles. The swarm is initialized with a group of random particles and it then searches for optima by updating through iterations. In every iteration, each particle is updated by following two “best” values. The first one is the best solution of each particle achieved so far. This value is known as p^{best} solution. Another one is that, best solution tracked by any particle among all generations of the swarm. This best value is known as g^{best} solution. These two best values are responsible to drive the particles to move to new better position. After finding the two best values, a particle updates its velocity and position with the help of the following equations;

$$v_{id}^{k+1} = v_{id}^{k} + c_1 r_1^k (pbest_{id}^k - x_{id}^k) + c_2 r_2^k (gbest_d^k - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad (2)$$

In this equality, v_{id}^k and x_{id}^k stand for separately the speed of the particle “i” at its “k” times and the d-dimension quantity of its position; $pbest_{id}^k$ represents the d-dimension quantity of the individual “i” at its most optimist position at its “k” times. $gbest_d^k$ is the d-dimension quantity of the swarm at its most optimist position. In order to avoid particle being far away from the searching space, the speed of the particle created at its each direction is confined between $-v_{dmax}$, and v_{dmax}. If the number of v_{dmax} is too big, the solution is far from the best, if the number of v_{dmax} is too small, the solution will be the local optimism; c_1 and c_2 represent the speeding figure, regulating the length when flying to the most particle of the whole swarm and to the most optimist individual particle. If the figure is too small, the particle is probably far away from the target field, if the figure is too big, the particle will maybe fly to the target field suddenly or fly beyond the target field. Usually, c_1 is equal to c_2 and they are equal to 2; r_1 and r_2 represent random fiction, and 0-1 is a random number. The selected reference pixels in the cover image are used to generate the prediction image and the detailed prediction process is described in the following section.

B.Pixel prediction based on image inpainting

After all reference pixels in the cover image have been decided, the non reference pixels are predicted by the image inpainting technique to generate the prediction image. The inpainting-based prediction process is totally dependent on the reference pixels of the cover image. Image inpainting, also known as image retouching, is a kind of technique that can fill in or remove the chosen regions of digital images seamlessly. Recently, some researchers have applied inpainting techniques in image repairing, image compression, and de-interlacing. Image inpainting is different from conventional image restoration in which the regions to be restored contain both noise and useful information. In image inpainting, however, the missing or damaged areas generally contain no useful information. Therefore, the task is to generate or create image regions that initially do not exist at all, based on the available information in the close neighborhood. Currently, there are several categories of inpainting methods, and the category that is the most widely used is based on PDE. By iteratively solving the numerical representation of a PDE, the inpainting methods manage to propagate the information of gray values smoothly from surrounding areas into region W to be inpainted along a specific direction. The inpainting process is terminated when the gray values in the computation domain reach a steady state.

In this work, we borrow the idea of treating the problem of pixel prediction from image inpainting, in which the useful information of reference pixels is used to estimate the non-reference pixels appropriately. A curvature driven diffusion (CDD)-based inpainting method [7] with a third-order PDE is utilized to implement pixel prediction. This method, in fact, represents an anisotropic diffusion process based on a total variation model, and the intensity of diffusion is related to the curvature κ of the pixels for prediction.

$$\frac{\partial P(x,y,t)}{\partial t} = div[\frac{\varphi|k|}{|\nabla P(x,y,t)|}\nabla P(x,y,t)] \quad \forall (x,y)\in\Omega \tag{3}$$

$$k = div[\frac{\nabla P}{|\nabla P|}] = \frac{P_{xx}P_y^2 - 2P_xP_yP_{xy} + P_{yy}P_x^2}{(P_x^2+P_y^2)^{\frac{3}{2}}} \tag{4}$$

$$\varphi(\omega) = \omega^{\lambda}, \omega > 0, \lambda \geq 1 \tag{5}$$

where t is the time index, Ω is the set of all (x, y) pairs that satisfy $Q(x, y) = 1$, and div(・) is the divergence operator. When we solve the PDE in (8) using the finite difference method, the useful information of reference pixels, which are in the surroundings of the non reference pixels and correspond to $Q(x, y) = 0$, can be propagated into region Ω successfully. The solution P_e is the final prediction result of the cover image.

Because the CDD-based inpainting model satisfies the connectivity principle of human visual perception, the prediction image P_e can reflect the structural and geometric information of the original cover image effectively. Therefore, good similarity between the cover image and the prediction image can be achieved. We also compared the prediction result of CDD inpainting with the gradient adjusted prediction (GAP) method. In the GAP method, the two topmost rows and the two leftmost columns are kept unchanged, and each of the residual pixels is predicted progressively in raster-scanning order by the seven neighboring pixels in its up and left regions.

C. Embedding Procedure

During the embedding procedure, the difference image E, i.e., the prediction error between P and P_e, is modified according to the distribution of its histogram and the embedding secret bits [4]. Then, the modified difference image E' is added to the prediction image P_e to generate the final stego image P_s. The detailed steps of the embedding procedure are listed as follows.

Step 1) After the prediction image Pe is obtained using the mask image Q and the CDD-based inpainting described in Sections II-A and II-B, the difference image E can be easily calculated as

$$E = P - P_e. \tag{6}$$

Step 2) Generate the histogram H_e of the difference image E. Denote two abscissa values of H_e, corresponding to the highest peak point and the second highest peak point, as α_1 and α_2, respectively. Thus, there are two cases according to the relationship between α_1 and α_2

Case I : if $\alpha_1 - \alpha_2 > 0$
Case II : if $\alpha_1 - \alpha_2 < 0$.

Step 3) If H_e matches Case I, from $\alpha 1$ to its right, search the first appearing zero point denoted as β_1 ($\alpha_1 < \beta_1$), and from α_2 to its left, search the first appearing zero point denoted as β_2 ($\beta_2 < \alpha_2$). Otherwise, search β_1 from α_1 to its left ($\beta_1 < \alpha_1$) and search β_2 from α_2 to its right ($\alpha_2 < \beta_2$).

Step 4) After obtaining α_1, β_1, α_2, and β_2, conduct raster scanning for the pixels $E(x, y)$ that correspond to $Q(x, y) = 1$ in the difference image. If H_e matches Case I, each scanned $E(x, y)$ is modified to $E'(x, y)$ using (13) in order to fulfill the embedding procedure. Otherwise, if H matches Case II, $E'(x, y)$ can also be acquired using (6), which changes (8) by substituting α_1 for α_2, α_2 for α_1, β_1 for β_2, and β_2 for β_1. It should be noted that each binary secret bit S for hiding is only embedded when the scanned $E(x, y)$ is equal to $\alpha 1$ or $\alpha 2$.

$$E'(x,y) = \begin{cases} E(x,y)+1, & \text{if } E(x,y)=\alpha 1 \text{ and } S=1 \text{ or if } \alpha 1 < E(x,y) < \beta 1 \\ E(x,y)-1, & \text{if } E(x,y)=\alpha 2 \text{ and } S=1 \text{ or if } \beta 2 < E(x,y) < \alpha 2 \\ E(x,y), & \text{if } E(x,y)=\alpha 1 \text{ or } \alpha 2, \text{ and } S=0 \text{ or otherwise} \end{cases}$$

subject to $Q(x, y) = 1$ (7)

$$E'(x,y) = \begin{cases} E(x,y)+1, \text{ if } E(x,y)=\alpha 2 \text{ and } S=1 \text{ or} \\ \quad \text{if } \alpha 2 < E(x,y) < \beta 2 \\ E(x,y)-1, \text{ if } E(x,y)=\alpha 1 \text{ and } S=1 \text{ or} \\ \quad \text{if } \beta 1 < E(x,y) < \alpha 1 \\ E(x,y), \text{ if } E(x,y)=\alpha 1 \text{ or } \alpha 2, \text{ and } S=0 \text{ or} \\ \quad \text{otherwise} \end{cases}$$

subject to $Q(x, y) = 1$. (8)

Step 5) Add the modified E' to the prediction image P_e to produce the final stego image P_s.

$$P_s = P_e + E' \quad (9)$$

It should be noted that if β_1 or β_2 do not exist in H_e mentioned in Step 3, zero points should be created. For example, in Case I, if zero point β_1 does not exist, the locations of the cover pixels corresponding to the lowest bin of H_e on the right side of $\alpha 1$ are recorded by a binary matrix Γ_z, and then the lowest bin is modified to zero and its abscissa value is used as the zero point β_1. In the extraction and recovery procedures, the binary matrix Γ_z can be utilized to localize and restore the cover pixels corresponding to the original lowest bin of H_e on the right side of α_1. Additionally, in order to avoid underflow and overflow problems, we should record the locations of the cover pixels, whose values equal 0 or 255 and their corresponding prediction errors [8]-[10] are in the ranges of two groups of peak points and zero points in H_e, and then modify their values to 1 and 254, respectively. The recorded information is stored in a binary matrix Γ_f which can assist in recovering these pixels to their original values, i.e., 0 and 255 in the extraction and recovery procedures. Usually, the two binary matrices, i.e., Γ_z and Γ_f , that served as the location map are sparse, and they can be compressed easily using a run-length coder. We can concatenate the compressed results of Γ_z and Γ_f into a binary sequence Γ.

After the above steps have been implemented, the embedding procedure is finished and the stego image P_s is obtained. The total number of embedded secret bits, i.e., total hiding capacity, is equal to the number of the difference image pixels that satisfy $E(x, y) = \alpha_1$ or α_2 and $Q(x, y) = 1$. In fact, the embedding procedure is equivalent to shifting the histogram of the difference image between the two groups of peak point and zero point, i.e., α_1, β_1 and α_2, β_2, by one level at most. Consequently, only the corresponding cover image pixels $P(x, y)$ are increased or decreased by one gray level atmost to form the stego image P_s. Therefore, the visual quality of the stego image is satisfactory.

D. Extraction And Recovery Procedures

During the extraction procedure, the embedded bits can be extracted, and the cover image can be recovered lossless. To guarantee the success of extraction and recovery, the parameters, μ, T_1, T_2, α_1, β_1, α_2, β_2, Γ, must be transmitted to the receiver side through the secure channel. Suppose that the size of the stego image P_s is not changed during transmission. Because the same strategy for choosing reference pixels is utilized and since the reference pixels are intact during the embedding procedure, the same prediction image P_e can be acquired from the stego image P_s by applying CDD-based inpainting. The detailed steps of the extraction and recovery procedures are listed as follows.

Step 1) after the same prediction image *Pe* is obtained from the stego image P_s by the same method used in the embedding procedure, the difference between P_s and P_e, i.e., E', can be acquired easily.

Step 2) According to the received α_1 and α_2, the case to which the embedding procedure belongs can be determined by (8).

Step 3) Conduct raster-scanning for the pixels $E'(x, y)$ that correspond to the locations of the nonreference pixels for the stego image P_s. If the embedding procedure belongs to Case I, (12) can be utilized to extract each embedded secret bit S and to modify each scanned $E'(x, y)$ into $E'_c(x, y)$.

It should be noted that each hidden binary bit S is extracted only when the scanned $E'(x, y)$ is equal to α_1, α_2, $\alpha_1 + 1$, or $\alpha_2 + 1$

$$E'_c(x,y) = \begin{cases} E'(x,y)+1, \text{ and } S=1, \text{if } E'(x,y)=\alpha_2-1 \\ E'(x,y)-1, \text{ and } S=1, \text{if } E'(x,y)=\alpha_1+1 \\ E'(x,y), \text{ and } S=0, \text{ if } E'(x,y)=\alpha_1 \text{ or } \alpha_2 \\ E'(x,y)+1, \text{ if } \beta_2 \le E'(x,y) < \alpha 2 \\ E'(x,y)-1, \text{ if } \alpha 1 < E'(x,y) \le \beta 1 \\ E'(x,y), \text{ otherwise} \end{cases} \quad (10)$$

$$E'_c(x,y) = \begin{cases} E'(x,y)+1, \text{ and } S=1, \text{ if } E'(x,y)=\alpha_1-1 \\ E'(x,y)-1, \text{ and } S=1, \text{ if } E'(x,y)=\alpha_2+1 \\ E'(x,y), \text{ and } S=0, \text{ if } E'(x,y)=\alpha_1 \text{ or } \alpha_2 \\ E'(x,y)+1, \text{ if } \beta 1 \le E'(x,y) < \alpha_1 \\ E'(x,y)-1, \text{ if } \alpha 2 < E'(x,y) \le \beta_2 \\ E'(x,y), \text{ otherwise.} \end{cases} \quad (11)$$

Step 4) Add E'_c to the prediction image *Pe* to produce the recovered image *Pc*.

$$P_c = P_e + E'_c \quad (12)$$

After the above steps have been implemented, all embedded secret bits S are extracted correctly, and the recovered image P_c is obtained. Actually, the extraction procedure is equivalent to shifting the histogram H'_e of E' back to the histogram H_e of E and E'_c is exactly equal to E. Thus, it can be found from (10) and (12) that the recovered image P_c is completely the same as the original cover image P.

According to the reference pixels that were chosen, the PDE-based inpainting algorithm using the CDD model can generate the prediction image effectively that has the similar structural and geometric information as the cover image. Through the use of the adaptive strategy for choosing reference pixels and the inpainting predictor, the accuracy of the prediction result was high, and larger numbers of embeddable pixels are acquired. To further reduce the distortion caused by embedding, two groups of peak points and zero points are selected from the histogram of the prediction error to embed the secret bits reversibly by the shifting operation. Because the same reference pixels can be exploited, the prediction result of the extraction procedure is the same as that of the embedding procedure. Thus, the embedded secret data can be extracted from the stego image correctly, and the original cover image can be recovered lossless. Compared with other schemes that have been reported recently, the proposed scheme has better performances with respect to the embedding rate and the visual quality of the stego image.

IV. Results And Discussions

Experiments were conducted on a group of gray-level images with different sizes. We used the embedding rate R to evaluate the pure hiding capacity shown as

$$R = \frac{L - 44 - |\Gamma|}{M \times N}(bpp) \tag{13}$$

where L is total number of embeddable pixels in the cover image, and bpp denotes bits per pixel.

The six standard, 512 × 512 images that were used for testing, including Lena, Lake, Barbara, Goldhill, Tiffany, and Peppers, are shown in below.

Fig. 2. Six standard test images. (a) Lena. (b) Lake. (c) Barbara. (d) Goldhill. (e) Tiffany. (f) Peppers.

In the experiments, we utilized MATLAB 7.9 for implementing the whole project because MATLAB stores an intensity image as a single matrix, with each element of the matrix corresponding to one image pixel. The matrix can be of class double, in which case it contains values in the range [0,1], or of class uint8, in which case the data range is [0,255]. The elements in the intensity matrix represent various intensities, or gray levels, where the intensity 0 represents black and the intensity 1 (or 255) represents full intensity, or white.

(a) (b) (c) (d)

Fig.3.Reference pixels for cover image *Lena* with different levels. (a) Level = 2. (b)Level = 3. (c) Level = 4. (d) Level = 5

(a) (b) (c) (d)

Fig .4. Stego images for Lena with corresponding parameters in Fig 2. (a).PSNR = 36.258 (b).PSNR = 36.2615 (c).PSNR = 36.2615 (d).PSNR = 36.2614

However, if the reference pixels in smooth regions decrease too much, the accuracy of the prediction result deteriorates, which leads to the decrease of embedding rate R. Whenever the level of the pixels get increases, the mask images get clear view. But when we increase the pixel level, the number of embedding bits gets decreases. The computational time of this method is low compared to the previous methods. This method also produces better visual quality of the medical stego images than the schemes proposed earlier.

IV.Conclusion

Because this project utilized two zero points to decrease redundant distortion and because the inpainting based prediction technique here used has better accuracy, this project has a greater embedding rate and produces better visual quality than that of previous schemes. In the stage of prediction, this project only increase one for the prediction errors between the peak point and the zero point that is nearest to the peak point; therefore, the visual distortion of the proposed scheme may be slighter. On the other hand, though the embeddable pixels of this scheme becomes more with the increasing of the thresholds, the decreasing number of the reference pixels leads to the degradation of prediction accuracy and more distortion by shifting. Thus, it can be seen that Compared with other schemes that have been reported recently, this project has better performances with respect to the embedding rate and the visual quality of the stego image. Further improvements will include using the faster numerical method to solve the high order PDE of the inpainting model so that the pixel prediction can be implemented more efficiently.

REFERENCE

[1]. J. Tian, "Reversible data embedding using a difference expansion," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896, Aug 2003.

[2]. Z.C.Ni, Y.Q. Shi, N.Ansari, and W.Su, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362 May 2006.

ISSN 2348 - 7968

[3]. W.L.Tai, C.M.Yeh, and C.C.Chang, "Reversible data hiding based on histogram modification of pixel differences," IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 6, pp. 906–910, Jun. 2009.

[4]. D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, o.3,

[5]. Chuan Qin, Chin-Chen Chang, Ying-Hsuan Huang, and Li-Ting Liao, "An Inpainting-Assisted Reversible Steganographic Scheme Using a Histogram Shifting Mechanism", IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 7, july 2013

[6]. Dian Palupi Rini, Siti Mariyam Shamsuddin, Siti Sophiyati Yuhaniz,, " Particle Swarm Optimization: Technique, System and Challenges," International Journal of Computer Applications (0975 – 8887) Volume 14– No.1, January 2011

[7]. T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions,"J. Visual Commun. Image Representation, vol. 12, no. 4, pp. 436–449 2001.

[8]. Z.Zhao, H.Luo, Z.M.Lu, and J.S.Pan, "Reversible data hiding based on multilevel histogram modification and sequential recovery," Int. J. Electron. Commun., vol. 65, no. 10, pp. 814–826 , 2011.

[9]. Z. Zhao, H. Luo, Z. M. Lu, and J. S. Pan, "Reversible data hiding based on multilevel histogram modification and sequential recovery," *Int. J .Electron. Commun.* vol. 65, no. 10, pp. 814–826, 2011.

[10]. M. Fallahpour, D. Megias, and M. Ghanbari, "Reversible and high capacity data hiding in medical images," *IET Image Process.*, vol. 5, no.2,pp.190–197,2011.