Proceedings of the Fourth IASTED International Conference
VISUALIZATION, IMAGING, AND IMAGE PROCESSING
September 6-8, 2004, Marbella, Spain

A LOW-DEGRADATION STEGANOGRAPHY MODEL FOR DATA HIDING IN MEDICAL IMAGES

Pouria Mortazavian*, Mohammad Jahangiri* and Emad Fatemizadeh**
*BsC, Biomedical Engineering Department, Science & Research Campus
Islamic Azad University, Tehran
Biomedical Engineering Department, Golzar Yekome Av, Adl Blv, Poonak Sq
Tehran, Iran
**Assistant Prof. Biomedical Engineering Department, Science & Research Campus
Islamic Azad University, Tehran
P-Mortazavian@ bme-azad.org, M-Jahangiri @ bme-azad.org, emad@ipm.ir

ABSTRACT
Textual labels overlaid on medical images provide significant information which should be unavailable to un-authorized persons who have access to the image.

This paper proposes “image steganography” as a means to hide this valuable information inside the image without losing it.

Conventional template matching is used to find the text on the image and extract it, the text is subsequently removed from the image and In the next step, a steganographical method is proposed by the authors to embed the text into the image, so that the textual data will be present in the image but visually undetectable.

In the proposed steganographical scheme, the data is embedded into the mean value of a number of pixels. But since any degradation in medical images could reduce the clinical value of the image, keeping the image alterations to a negligible amount becomes of great concern , thus the maximum modification on each pixel is limited by the algorithm.

Implementation of this algorithm on many medical images shows that a reasonable amount of data can be embedded into an image without perceptible alterations besides having a satisfying robustness against noise.

KEY WORDS
Medical images, Image Steganography, data hiding.

1. Introduction

With the development of network technologies, digital medical images can be transmitted conveniently over the networks by means of standard protocols (DICOM, PACS) for consultation, educational, and various other purposes. On the other hand, Image tags (e.g. age, sex, imaging conditions, etc. which are useful for classification and interpretation of the image) that appear on most medical images, contain patient’s personal information or other significant records which should be kept secure from the access of second–opinion physicians, medical researchers, students, and other legitimate users who may be in contact with the image itself.

Thus this paper uses a template matching technique to find, eliminate and extract the textual information, and after transforming the extracted text to binary data, uses a robust, low degradation steganography method, to embed the binary data into the image which the textual information has been eliminated from.

These subsequent steps are shown in fig.9.

The template matching technique used for character recognition uses cross-correlation between a “standard font” template and blocks from the original image[1]. This paper focuses on the steganographical scheme, specially proposed for medical images.

Steganography is the art of communicating a “message” by embedding it into a signal which is used as a “cover” in a way that the cover seems innocuous (and -in the case of medical images- unchanged).This “cover signal” can be of type audio, video, still imagery etc. In this case, the “cover” is a medical image and the “message” is the text extracted from the original image.

Throughout this paper definitions are used with the following concepts:

- Original image: Unprocessed input image which contains the textual information on it.
- Cover image: Original image with the textual labels excluded.
- Stego-image: Cover image with the data embedded into.
- Message: Textual information extracted from the original image together with some information about the location of each character on the original image.
- Payload: Maximum amount of characters that can be embedded into the image.
- Steganalyze: Any effort towards discovering the existence of an embedded message in the stego-image.
- Imperceptibility: Innocuousness of the stego-image.

- Sender: Person who creates the stego-image from the original image.
- Receiver: Any person authorized to extract the message from the stego-image.

There are three significant facts to be considered in image steganography, 1)high payload, 2)high robustness against noise and steganalyze, and 3)imperceptibility. Different methods have been proposed for different applications where one of the above-mentioned facts becomes of greater significance. For example in LSB based methods,[2], having a significant amount of payload is the major purpose while in Jpeg – Jsteg [3] robustness against steganalyze is the first priority.

Since the cover image in this paper is a medical image, the amount of degradation in the cover image due to embedding data is the main concern. In addition to low degradation this method has a great robustness against noise and a practicable amount of payload.

1. 1. Previous work

Several image steganography techniques have been proposed. The most common approaches modify the least significant bits of each pixel (LSB). In [2] two such techniques are described. In another category of image steganographical methods, the frequency components are modified to embed the data [4],[5] while others exploit HVS (Human visual System) characteristics for embedding data[4],[6]. In [7], an image steganographical method (Spread Spectrum Image Steganography) is presented in which the data is modulated in a waveform noise and consequently added to the cover image. There are many image steganography techniques that are strongly correlated with the format of the cover image. S-Tools [8], Stego [9], and Fredrick’s method [10] are designed for palette based images, and Jpeg – Jstego [3] is used for JPEG compressed images. Consequently format conversion of these stego-images will destroy the whole hidden message. A steganographical method similar to ours is proposed in [11], where the mean of the grey-levels of a sub-block in the cover image is modified in a way that it represents the bit embedded in the sub-block. In other word the mean of a sub-block shows that the embedded bit is either 1 or 0.This method yields fine results in most cases, but is not readily applicable to medical images, since:

1. Block effect is noticeable in the uniform part of the image (eg. Background in MRI images) despite the effort made to reduce the block effect.
2. A “pre-processing” is necessary to limit the dynamic range of grey-level values to[K,255-K]. Where K is a positive integer. This pre-processing causes unacceptable modifications to the image (e.g. Increasing the overall brightness in the background or eliminating some fine details)

Our new method resolves these disadvantages by replacing other solutions instead of pre-processing and by completely solving the block effect problem.

1. 2 Paper Structure

In the next section our proposed medical image steganography model will be described. Some experimental results (subjective and objective) and performance analysis will be presented in Section 3 to show algorithm efficiency. In Section 4, conclusions will be demonstrated.

2. The Proposed Steganographic Model

In this steganographical model the first step is to create the “cover image” and the “message” to be embedded into the cover image. For this reason a conventional template matching technique is used to identify and locate the characters overlaid on the original image. Note that the original image is a grey-scale image which should be transformed to a binary image prior to using the template matching technique for character recognition. The “message” is constructed at this point by the identified characters. Afterwards the located characters are eliminated from the image by replacing the grey-level values to create the cover-image. Fig1 shows an overall block diagram of different parts of the process.

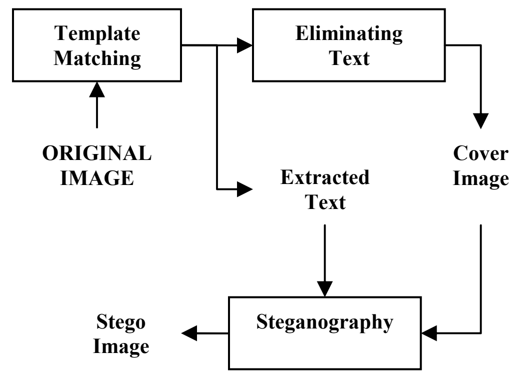


Fig1 : An overall block

2.1 Constructing the Stego - Image

The constructing process is shown in the block diagram of Fig.2.Each of the blocks is described in the following subsections.

2.1.1 Embedding Data

The embedding scheme is based on the fact that the mean value of a number of pixels is a robust attribute in stego-images affected by noise. So embedding data in this attribute will result in acceptable robustness against noise i.e. if a D-bit data string is to be embedded into the image the image should have D blocks and the mean value of the i th block should be modified such that it could represent the i th bit of the data string.

The embedding process is shown in the block diagram of Fig3.

The inputs to the "Data Embedding" block are:

1. **Data (i) (D (i)):** The i th bit of the binary data string. Note that the message has been encoded by an Error Correcting Code (ECC) prior to entering to the "Data Embedding" block (Fig. 2). A Hamming (7, 4) code is used as the ECC in our scheme. This ECC is only utilized in order to increase noise robustness and does not play a key role in this scheme. The output of this ECC Encoding block, is hereafter referred to as "Data String".
2. **Cover-Block (i) (CB (i)):** A n by n block of pixels of the image which the data is to be embedded into. As can be seen in Fig.2, This image is a shuffled version of the cover image and will be referred to as "suffled_image" from now on. Shuffling is performed using a pseudo random sequence which is generated by the PRSG (Pseudo-Random Sequence Generator) block (Fig.2) seeded with numeric key which is used as a "password"- shared between the sender and the receiver.

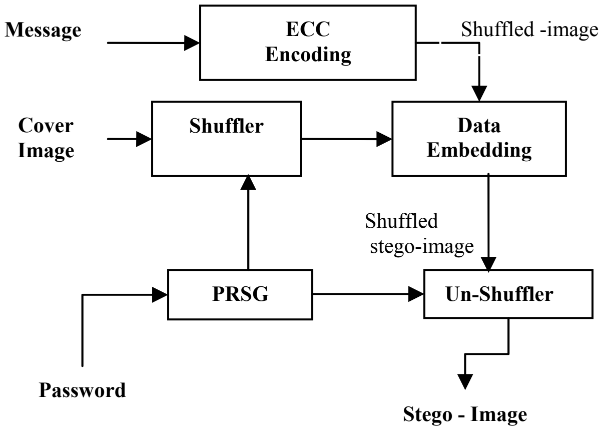


Fig 2 : Constructing process

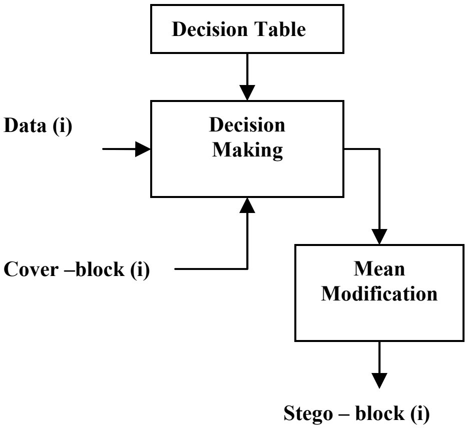


Fig. 3: Embedding

The Data Embedding block modifies the mean value of CB (i) such that the new value represents D (i).This new value is defined in Decision Making block according to D(i) and the Decision Table.

The "Decision Table" used both in the embedding module (at the sender side) and the extracting module (at the receiver side) contains information indicating what data bit is embedded in the mean value of a block.

Decision Table generation: Except for boundary areas, the numeral spectrum of mean values is divided into many equal sized intervals-each of length "k". The binary symbols of "0" and "1" are then assigned to the intervals successively.

The mean value of CB(i) will be adjusted to the centre of a certain interval with the binary symbol same as D(i). If the mean value of a block is not changed to another interval during transmission (e.g. by noise etc.), the embedded data bit (D (i)) can be extracted correctly in the receiver side. Hence the length of the intervals (K) determines the tolerance against noise. An example of a "Decision Table" with K=2 is shown in Fig.4.

Mean Spectrum	0 **1** 2^{-}	2 **3** 4^{-}	4 **5** 6^{-}	6 **7** 8^{-}	…
Binary Symbol	**0**	**1**	**0**	**1**	…

Fig4: Decision Table Example

Let:

1. $M_b(j)$ be the j th interval center with the binary symbol "b" assigned to (e.g. In Fig.2 m_l (1) =3 and m_0 (2) =5).
2. M(i) be the mean value of CB (i).
3. $M_d(i)$ be the destination mean value for CB(i) which is defined by the "Decision making" block.

Then the Decision Making process can be explained by equation 1.

$$M_d\,(i) = \underset{j=1,\,2,\,3\,\ldots}{Arg\ min}\ \{|M_b(j)-M(i)|\} \qquad (1)$$

Where: $b = D\ (i)$

Mean Modification: Once M_d (i) has been determined from (1) the grey-level values of the pixels in CB(i) should be modified such that the mean of their new values would be M_d (i).A simple way to modify the mean value of CB(i) is to add ΔM(i),defined in (2) ,to each of the pixels in CB(i).

$$\Delta M(i) = M_d\,(i)\text{-}\ M\,(i) \qquad (2)$$

That is:

$$G_{ij\text{-}new} = G_{ij\text{-}old+}\ \text{round}\ [\Delta M\ (i)] \qquad (3)$$

Where $G_{ij\text{-}old}$ is the old value of the grey-level of the j th pixel in CB(i) and $G_{ij\text{-}new\ is}$ it's new value.

The following should be considered about (2) and (3):

1. This method uniformly modifies all pixels in CB(i), therefore it will cause considerable block effects especially in smooth regions of an image. As can be noticed in Fig.2 the data is embedded into blocks of the "shuffled" image in our scheme, so the block effect appears only in the shuffled stego-image and by un-shuffling this image, the actual stego-image is constructed in which the block effects have disappeared
2. The value of $\Delta M(i)$ -contrary to digital image grey-level spectrum- is not a discrete value therefore the rounded value of $\Delta M(i)$ should be added to each pixel and this imposes unwanted modification on the resulting mean value.
3. This simple way of modifying the grey-levels may result in values outside the grey-level dynamic range. This problem can be addressed by pre-processing the pixels to limit the dynamic range of grey-level values to [K,255-K] but this yields unacceptable modifications to the image (e.g. Increasing the overall brightness in the background or eliminating some fine details).
4. The alteration done on the pixels is not optimal in the sense that some pixels might be altered more than the minimum necessary amount. For example consider a block of 16 pixels in which $\Delta M(i)=0.875$, while the destination mean can be reached by only modifying 14 of the pixels, by using equation 3 all 16 pixels will be altered and this means more perceptibility in the resulting stego-image.

As mentioned above, the block effect problem is addressed by shuffling the image prior to embedding the data, but the three other problems still remain. In order to address these limitations, the algorithm shown in Fig. 5 is proposed by the authors for "Mean Modification" instead of (3). In this figure it is supposed that $M(i)>M_d(i)$ (i.e. $\Delta M(i)<0$) so the values of the pixels in CB(i) should be decreased, and:

- N1: Is the total number of modifiable pixels (number of pixels with values higher than $\Delta M(i)$)
- ΔG_T: Is the total amount of alteration necessary for the sum of the pixels' grey-levels so that the mean of the block will approach $M_d(i)$:

  $$\Delta G_T = |\Delta M(i)|.N \qquad (4)$$

  Where N is the total number of pixels in the block.
- $M_{d2}(i)$, $\Delta M_2(i)$, N2 and ΔG_{T2} are successively equivalents of $M_d(i)$, $\Delta M(i)$, N1 and ΔG_T in the switch mode

In simple words -in the above mentioned situation ($\Delta M(i)< 0$)- this algorithm decreases the grey-level value of modifiable pixels successively until the total necessary alteration is met, but in the same time it prevents the maximum alteration of each pixel from exceeding a threshold value of 2K. If a pixel is to be altered more than this threshold, the algorithm enters the "Switch mode". In this mode the destination mean is changed to the nearest $M_b(j)$ greater than M(i) which is denoted by M_{d2}. Consequently, $\Delta M_2(i)>0$ and this time the pixel values should be increased so the algorithm increases the grey-level values of the pixels which are modifiable in the "switch mode" -pixels with grey-level values less than $[255-\Delta M_2(i)]$- one by one until the total amount of modification reaches ΔG_{T2} which means that the mean of the block has reached M_{d2}.

It can be shown that if the algorithm enters the switch mode, the maximum alteration of each pixel remains less than 2K. So no pixel alters more than 2K in either mode. Additionally, since the pixels whose values reach the boundary of the grey-level dynamic range (non-modifiable pixels) will no longer be processed, the algorithm does not result in grey-level values outside the dynamic range. Note that this advantage is gained without having to "pre-process" the image to limit the dynamic range. A flowchart of the algorithm is shown in Fig.5.

In the cases where M (i) <M_d (i), a similar algorithm is utilized to increment the pixel values – or decrement them in case the algorithm enters the "switch mode".

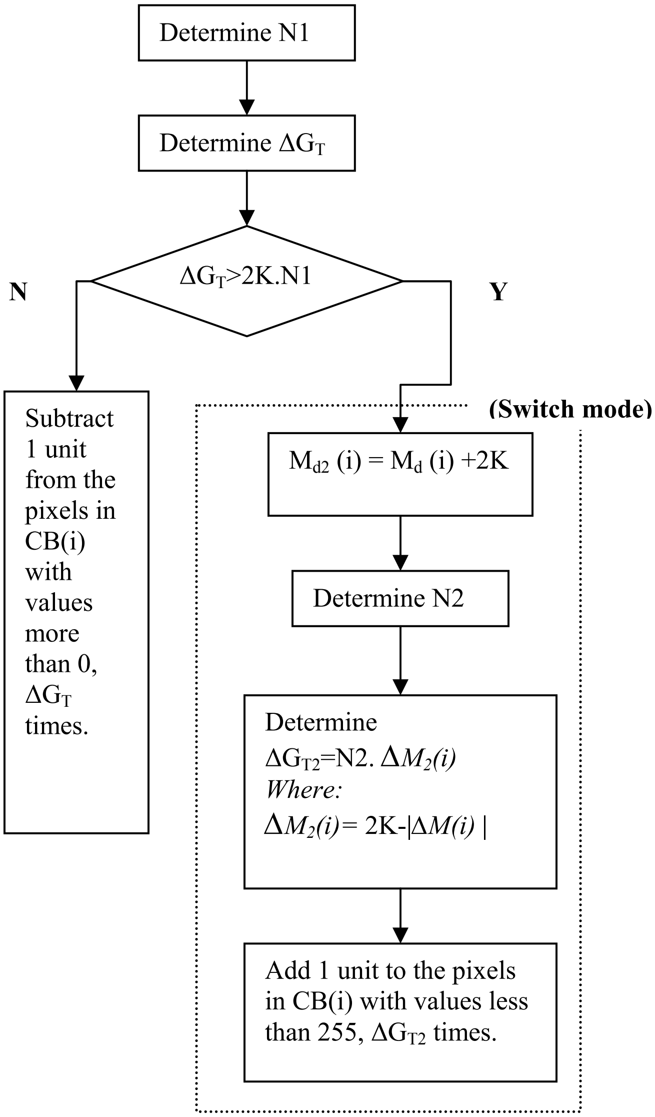

Determine N1
Determine ΔG_T
ΔG_T>2K.N1
N
Y
Subtract 1 unit from the pixels in CB(i) with values more than 0, ΔG_T times.
(Switch mode)
M_{d2} (i) = M_d (i) +2K
Determine N2
Determine ΔG_{T2}=N2. $\Delta M_2(i)$ Where: $\Delta M_2(i)$= 2K-|$\Delta M(i)$ |
Add 1 unit to the pixels in CB(i) with values less than 255, ΔG_{T2} times.

Fig. 5: Mean modification algorithm

The embedding process can be summarized as below:

```
1. Create the “message” and the “Cover-Image”
2. Shuffle the “Cover_image”
3. Let i=1
4. Take the i^th , N pixels (from pixel number (i-1)*N+1 to pixel number i*N ) as CB(i)
5. Determine M(i)
6. Determine $M_d(i)$. (Decision Making)
7. Run the “Mean Modification” algorithm(fig.5 or its equivalent in case $M (i) <M_d (i)$ )
8. Put the resulting pixels as pixels number (i-1)*N+1 to i*N in the “shuffled-stego-image”.
9. If there are still more data bits to embed
       Increment i and go to step 4
   Else
       Un-shuffle the “shuffled-stego-image” to get the final “stego-image”.
```

2.2 Message Extraction

Message extraction is the process of extracting the message from the stego-image at the receiver side. Using a shared password (between sender & receiver), the receiver can extract the message from the stego-image. This password enables the receiver-side to recreate the same sequence used in the sender-side for shuffling the image, by utilizing a PRSG .This sequence is used to reshuffle the stego-image. Afterwards the Decision Table is recreated (with the same ‘k’) and the ith message bit, b(i), is extracted from the ith block of the shuffled-stego-image (SB(i)) in the following manner:

“Let M_{sb}(i) be the mean value of SB(i), then b(i) is the binary symbol assigned to the interval in which M_{sb}(i) is located.”

The extracted message is decoded by ECC hamming decoder (7, 4).Once the message is decoded, characters eliminated in the embedding process will be revealed on their original locations.

The recovering process can be summarized as below:

```
1. Regenerate the “Decision Table”.
2. Regenerate the shuffling indices by using a PRSG seeded with the shared password.
3. Shuffle the “Stego-Image” to get the “Reshuffled-Stego-Image” .
4. Let i=1 .
5. Take the i^th , N pixels (from pixel number (i-1)*N+1 to pixel number i*N ) of the “Reshuffled-Stego-Image” as SB(i)
6. Determine the mean value of SB(i). ($M_{sb}$(i))
7. Determine b(i) as the binary symbol assigned – by the Decision Table- to the interval in which $M_{sb}$(i) lies.
8. If the message has not yet been completely extracted,
       Increment i and go to step 5
   Else
       Decode the extracted message and reveal the textual data on the Stego-Image to yield the “Recovered-Image”
```

3. Experimental Results

The proposed method was implemented on 40 CT images and 40 MRI images, all of size 640x512 with a bit depth of 8 (grey-levels ranging from 0 to 255). 9 full-load stego-images were generated using different parameters (N=16, 64,256 & K=2,3,4) for each image. Furthermore implementation of this algorithm on a number of RGB images results in suitable responds.

The algorithm was implemented on a typical PC with a Pentium IV (2GHz) processor, 512 MB RAM, using Matlab 6.1 software.

It is difficult to quantify how imperceptible the embedded data is. As a numeric measure to quantify the imperceptibility of the setego-images, peak-signal-to-noise ratio (PSNR) between each cover-image and its corresponding stego-image was calculated. Note that a higher PSNR means less perceptibility. Fig.7 shows the mean PSNR value over the entire image database vs. N and K. As can be seen in Fig.7, the PSNR decreases (perceptibility increases) with increasing “K”. This phenomenon can be easily explained by noting that a larger numeric value for “K” yields more alteration of the pixels. Also Fig.6 shows that imperceptibility does not vary much with N. This could have already been expected since the amount of alteration on pixel values is independent of “N”.

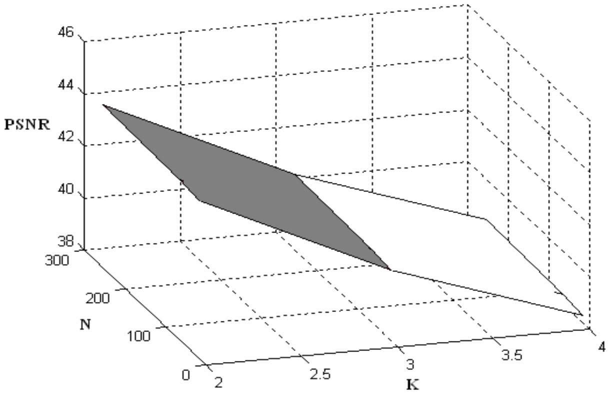


Fig.6: PSNR changes vs. N&K

Additionally a subjective test was undertaken to measure the imperceptibility of the stego-images regarding HVS. 30 male and female observers -aging from 18 to 44 - took part in the test. The observers were asked to compare five of the stego-images with their corresponding cover-images and assign a numeric value from 0 to 4 to each image according to the amount of difference they feel between the two images, with the following meanings:

- 0: No difference at all
- 1: Hardly any difference
- 2: Little difference
- 3: Moderate difference
- 4: High difference

The results of the subjective test, summarized in Table1, show that for almost all the parameter sets the stego-images have adequate imperceptibility. Furthermore Table1 confirms the claim that imperceptibility is related with “K” and almost independent of “N”.

Table.1: Subjective Test

K	N	Mean of opinions
2	16	0.79
	64	0.92
	256	0.76
3	16	0.83
	64	1.2
	256	1.16
4	16	1.56
	64	1.3
	256	1.4

In order to test the robustness of stego-images generated by this scheme against noise, Gaussian noise with zero mean and different variances was added to each image and consequently the embedded data was extracted from the resulting noisy stego-images. The bit-error-rate (BER) was calculated each time as the percentage of bits that have been recovered incorrectly after adding noise. Fig.7 shows the mean value of BER over the entire test database vs. the amount of noise added to the stego-image (mean value of the PSNR between the noisy and the non-noisy stego images) for three different amounts of K. Since a larger numeric value for K yields wider intervals in the Decision Table, more noise can be tolerated before the a block’s mean value gets out of it’s interval, as a result, a larger numeric value for “K” should result in higher noise robustness.

Fig. 8 is the same as Fig.7, only this time the results for different amounts of N are plotted. Obviously, when a block contains more pixels, its mean value is less vulnerable against noise. Thus a larger numeric value of N should yield more robustness against noise and this is exactly what Fig.8 shows.

The result of applying this image steganographical algorithm on a MRI image, is shown in fig.9b and 9c which are the “Cover-Image” and the “Stego-Image” respectively.

4. Conclusions:

A large numeric value for K increases robustness against noise but in the same time increases the “perceptibility” of the stego-image. On the other hand, a large numeric value for N also increases noise robustness but it decreases the payload. Depending on specific applications, a trade-off should be made between these parameters. In the case of medical images, N=64 and K=2 would be practicable yielding not only significant imperceptibility but also satisfying noise robustness and -usually- enough payload.

The greatest advantage of this method is that it makes only negligible alterations to the cover image, therefore the method is applicable for medical images without reducing their authenticity.

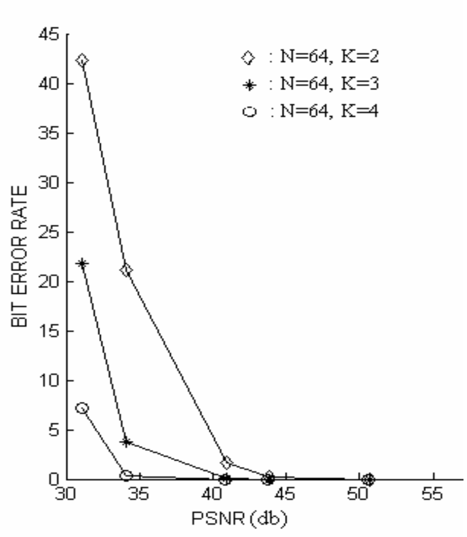


Fig.7: BER vs. Amount of noise added to the image (PSNR) in N=64 and K=2, 3, 4.

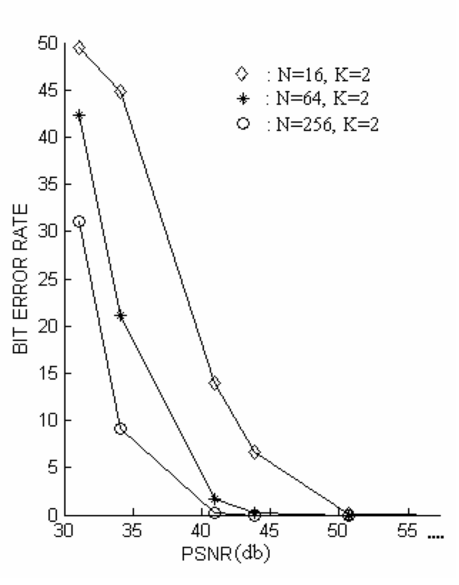
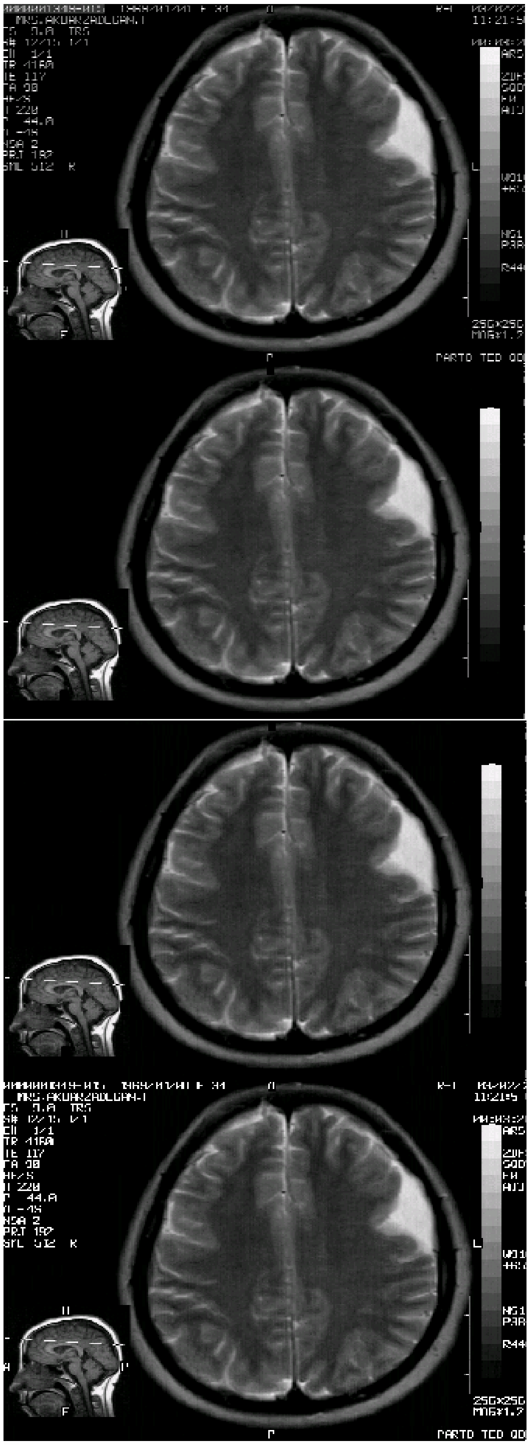


Fig.8: BER vs. Amount of noise added to the image (PSNR) in K=2 and N=16, 64, 256



(a) Original-Image

(b) Cover-Image

(c) Stego-Image

(d) Recovered-Image

Fig.9: Implementing the whole algorithm on an MRI Image, by removing the texts from the original image(a), the cover-image(b) is yielded, and the Stego-Image(c) is constructed by embedding the data into the Cover-Image by using the proposed steganographical algorithm. Finally the extracted textual data is revealed on the Stego-Image at the receiver side to yield the Recovered-Image(d)

References

[1] Rosenfeld, A. and Kak A.C., *Digital picture processing(*Academic Press, New York, N.Y., 1976)

[2]R.G Van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A Digital Watermark" *in Proc. 1994 IEEE Int. Conf. on Image Proc., Vol. II*,(Austin, TX), pp. 86-90, 1994.

[3] D.Upham,Jpeg-Jstego,Modification of the Independent JPEG Group.JPEG software (release 4) for 1-bit steganography in JFIF output file. ftp://ftp.funet.fi/pub/crypt/steganography

[4] M. D. Swanson, B. Zhu and A. H. Tewfik. "Robust Data Hiding for Images", *in IEEE Digital Signal Proc Workshop* (Loen, Norway) pp. 37-40, Sept. 1996.

[5] I. Cox, J.Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia" *Tech. Rep. 95-10 NEC Research Institute.*

[6] Raymond B. Wolfgang , Christine I. Podilchuk and Edward J. Delp partially supported by a grant from the AT & T Foundation.

[7] Lisa M. Mavel, Charles G. Bouncelet, Jr and Charles T. Retter, "Spread Spectrum Image Steganography", *IEEE Transaction on Image Processing, Vol. 8, NO.*8, *Aug.* 1999, pp 1075-1083

[8] A.Brown,"S-Tools",Shareware ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/s-tool4.zip (Version 4)

[9] Romana Machado, "Stego", Shareware http:/www.stego.com

[10] Jiri Fridrich, "A New Steganographic Method for Pallete-Based Images", in Proeedings of the *IS&T PICS Conf , Savannah ,Georgia , April 25-28,1999,*pp.285-289

[11] Y.K.Lee & L.H.Chen, "A Secure Robust Image Steganographic", Model, Supported in part by The National Science Council