Distributed AMELSB Replacement Method for Text Hiding

Dr. Yossra H. Ali
University of Technology
Bagdad/Iraq
Yossra_h_a@yahoo.com

Ahmed Y. Yousif
University of Technology
Bagdad/Iraq
atds@yahoo.com

Tayseer S. Atia
University of Technology
Bagdad/Iraq
tsamastersc@yahoo.com

Abstract

Nowadays, daily communications of all kinds over the Internet have become incredibly popular. However, message transmissions over the Internet still have to face all kinds of security problems. While aim of cryptography is protecting the content of messages, steganography is the technique for hiding additional information in cover data. This paper, proposed method based on Adaptive Minimum Error Least Significant Bit (AMELSBR) but by distributed this error among the gray level components (red(R),green(G),blue(B)) of each pixel, where aim to reduce color variation among pixels when k* message bits embedded in the pixel tacking benefit that the gray level component (R,G,B) for each pixel have the same value , so that R component used to embed the value of k* and both the G and B component used to embed the actual message bits. This method speedup the extraction process so that it is not necessary to reverse the hiding procedure to extract embedded message. Finally, it is not necessary to use control information about the hiding procedure so that the stego cover will be used with full capacity to store the message only.

طريقةAMELSBR الموزعة لاخفاء النصوص

د. يسرى حسين علي احمد يونس يوسف تيسير سلمان عطية

في الوقت الحاضر، جميع انواع الاتصالات اليومية عبر الانترنت اصبحت شائعة بالرغم من ذلك ارسال الرسائل عبر الانترنت لا يزال يواجه المشاكل الامنية، طالما ان الهدف من التجفير هو حماية الرسائل فان اخفاء البيانات هي تقنية اخفاء معلومات اضافية في في غطاء من البيانات .في هذا البحث تم اقتراح طريقة اخفاء بالاعتماد على طريقة(AMELSBR) ولكن بواسطة توزيع هذا الخطا بين مكونات المستوى الرمادي (الاحمر، الاخضر، والازرق)ى لكل نقطة وهي تهدف الى تقليل اختلاف الالوان بين نقاط الصورة عندما يتم تضمين مجموع من البت في الرسالة ، باخذ ميزة مكونات المستوى الرمادي (الاحمر،الاخضر،الازرق) لكل نقطة بكونها تمتلك نفس القيمة ، حيث ان المكون الاحمر يستخدم لتضمين مقدار البت المراد تضمينها وكل من المكونين الاخضر والازرق تستخدم لاخفاء البيانات الفعلية من الرسالة.

هذه الطريقة تسرع عملية ا لاستخلاص ، حيث انه لا توجد ضرورة لعكس عملية الاخفاءلغرض استخلاص البيانات المضمنة في الرسالة ، كما ان غطاء الصورة يستخدم لتضمين بت الرسالة فقط وليس هناك حاجة لمعلومات توجيه حول عملية الاخفاء في الغطاء.

Keywords: Steganography, stego cover, MELSBR, DA-AMELSBR

1. Introduction

The escalation of communication via computer network has been linked to the increasing use of computer aided steganography. Steganographic methods usually hide ciphered messages in other, harmless- looking data in such a way that a third person can not detect or even prove this process. [1]

Information hiding represents a class of process used to embed data into various forms of media such as images audio, text or 3D models.

The embedded data should be invisible to a human observer. The term hiding can refer to either making the information imperceptible or keeping the existence of the information secret [2].

Steganography is the art and science of communicating in a way, which hides the existence of the communication in contrast to cryptography. Where the enemy is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by cryptosystem .the goal of steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second secret message present [3].

In this paper, a steganography method will embed messages in a cover-image and create a stego-image. Those stego-images produced using distributed minimum error LSB replacement method.

2. Minimum-Error LSB Replacement Method (MELSBR) [4]

In grayscale image, there are total 256 levels to represent the intensity of each pixel. If we were to embed k ($k<8$) bits of message in a pixel, directly replacing the k-LSBs of the pixel will introduce less error than replacing any other k-bits, and the maximum error is $2k -1$.

In the total 256 gray levels, there are 2 (8-k) gray levels with the same value in the k least significant bite as the k message bits. To reduce the embedding error, we should select the one that has minimum-error with the original gray level to replace the pixel level. To reach the aim, a simple way is provided. It will adjust (k+l)th LSB, and check it's embedding error. And then select the gray-scale with less

embedding error to replace the original ones. Fig (1) illustrates the adjusting method, which contains two steps and is called minimum-error LSB replacement method (MELSBR). Using the MELSBR method, the maximum error can be restricted to $2^{(k-1)}$.

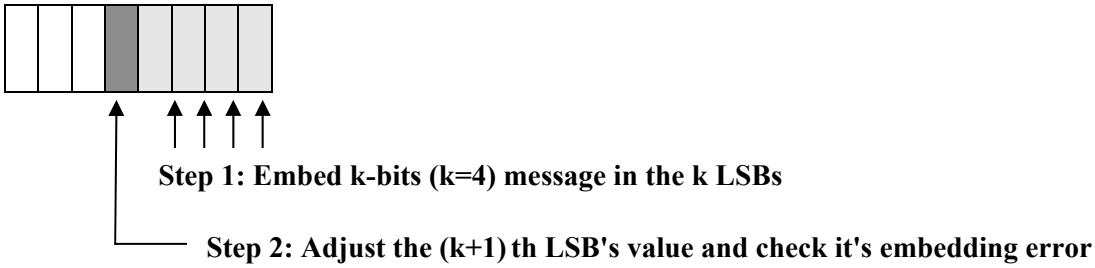


Figure (1) Two Steps of MELSBR method

3. Adaptive MELSBR Method [4]

This method works by tacking advantage of local characteristics of cover image base on the MELSBR.

The upper bound (U) of embedding capacity for each pixel in the cover image is evaluated,

$$U = \lfloor \log_2(X) - 1 \rfloor \quad (1)$$

where X is the gray level of pixel x, the embedding capacity K of pixel x is defined as the minimum number of bits to store the value D minus 1,

$$K = \lfloor \log_2(D) \rfloor \quad (2)$$

D for each pixel is equal to the difference between the maximum and minimum gray level of pixel x neighborhood as shown in Figure (2).

$$D = \max\{a, b, c, d\} - \min\{a, b, c, d\} \quad (3)$$

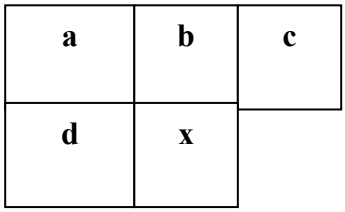


Figure (2) the mask for evaluating the gray variation in the neighbors of pixel "x"

The number of bits pixel can embed k* = minimum (U,K).

Finally, to avoid embedding message in local area a scatting method is provided. To scatter a message a random number with value in [0,1] is generated for each pixel to decide weather the pixel is used to embed Message bits by compare its value against P , where P = AM/ C , embedding ratio, amount of message, predictive embedding capacity respectively.

4. The Proposed Distributed AMELSPR Method

In order to reduce the amount of color variation between pixels when K* message bits embedded in the pixel, and tacking advantage that the gray level component (R,G,B) for each pixel have the same value for DA_MELSBR method was proposed.

This method works by compute the embedding capacity for each pixel k*, and distribute k* bits from message between G and B component for the selected pixel. R component will mark k* value for this pixel, since the total number of gray level is 256 used to represent the intensity of each pixel of gray scale image and since k* value is the minimum between U and K , log function reduce the value of the base parameter, the maximum value can be computed is log 256, so k* don't exceed 2-bit or 3-bit, the maximum value of bits pixels 6 and the 7^{th} bit is the adjust(k+1) LSB value. So any changing make for this bit change the color value which makes color variation between pixels color is noticeable.

For the above reasons, R component is reserved to store the actual number of bits in another component (G & B), G and B store the same number of bits, if k* is odd number then simple pad will be used.

So, R value can be 01,10, and 11 which represent the number of bit in each component G&B. '0' value will be used as an indicator and will be stored in the LSB bit from the R byte, the final values can be stored in R byte is 010,100, and 110. This value represented actually the number of bits in both G & B byte (2, 4 or 6).

In order to scatter the message, we use the following key,

Key= 1 – (X /256). If k< p then the pixel can be used to embed the message.

Variable used:

P embedding ratio

C predictive embedding capacity

AM amount of message

U upper bound

K embedding capacity K of pixel x

4.1 Embedding Algorithm

Step 1. Predict the embedding capacity C of the cover-image.

Step 2. Compute the embedding ratio with P-AM/C, AM is the amount of embedded message.

Step 3. Scan the cover image from the top-left to the bottom-right. For each pixel x in the message-embedded part, perform the following steps:

Step 4. Using Eqs. (1) and (2) to evaluate the embedding capacity (K) and the embedding upper bound (U) of x. Take K* = min(K, U)

Step 4.1 Embed the value ofk/2 in red byte

Step 4.2 Embed K*-bit message in the K*-least significant bits of x, with k/2 bits in green byte and k/2 in blue byte.

4.2 Extract Algorithm

The following algorithm describes the sequence of extracting process.

Step 1: open (stego cover, recover binary file, recover text file)

Step2: do while pixel position not equal to the end of hiding data

Step3: extract the value of k/2 from the red byte

Step4: extract the first k/2 bit of text from green byte and add it to the recover binary file

Step5: extract the remainder k/2 bit from the blue byte and add it to he recover binary file

Step6: convert the binary file into text file

5. Experimental Result

We have used the proposed method to embed different messages in different images as shown in Figure [3 a,b,c] and Figure[4 a,b,c] respectively, Figure[3.a] shows the selected message, Figure[3.b]

shows the binary representation of the embedded message and the internal implementation of the proposed algorithm where the value of R, G, B, K*,Bts, R', G', and B' parameters are evaluated for a selected gray scale stego cover, Figure[3.c] shows the resulted extracted binary representation and the retrieved text. Figure [4.a,b,c] shows the same details for color image, the minimum square error, signal to noise ratio, peak signal to noise ratio metrics are evaluated for each cover image, Table[1] lists different message size and different stego covers and the time for embedding and extracting process for each stego cover. From Figure(3.b) we can note that the change in value for green and blue component are small, but the change for value of the red component is greater than the other, this changes for gray scale image while in Figure(4.b) the changes for all the component are closer to each other.

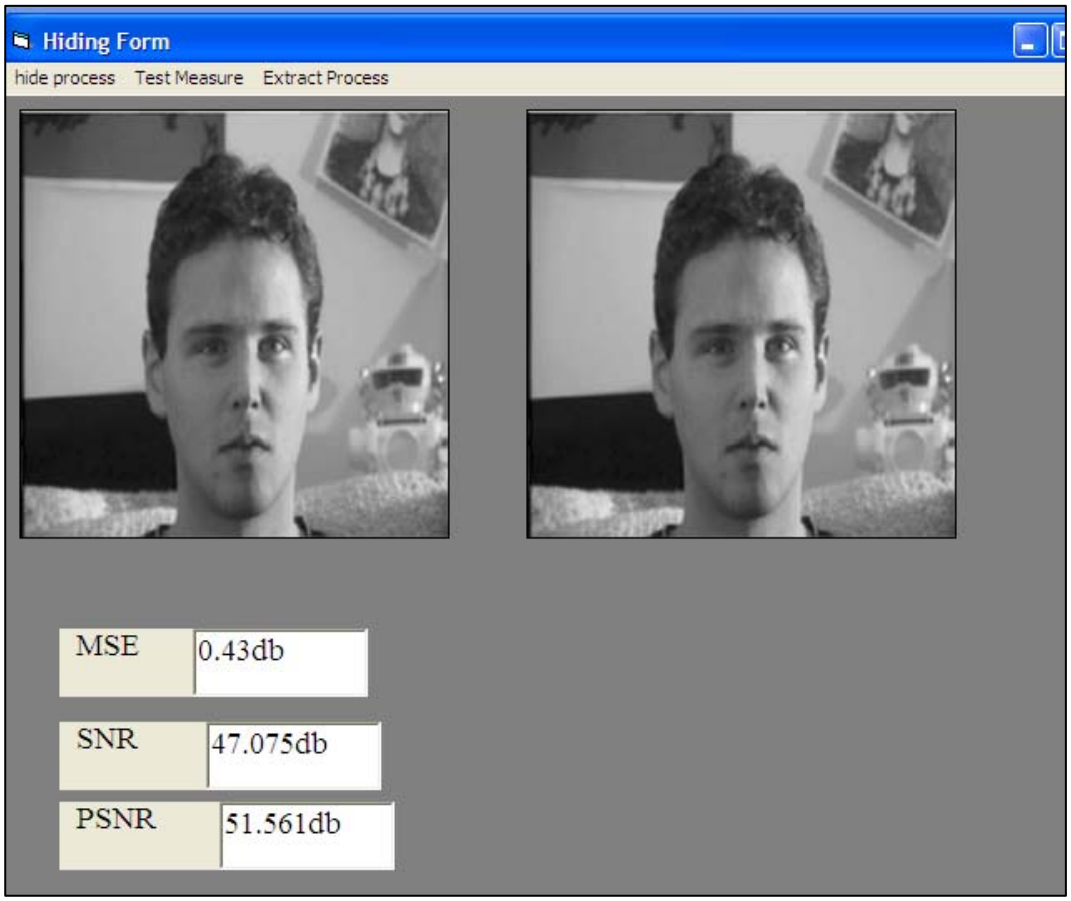
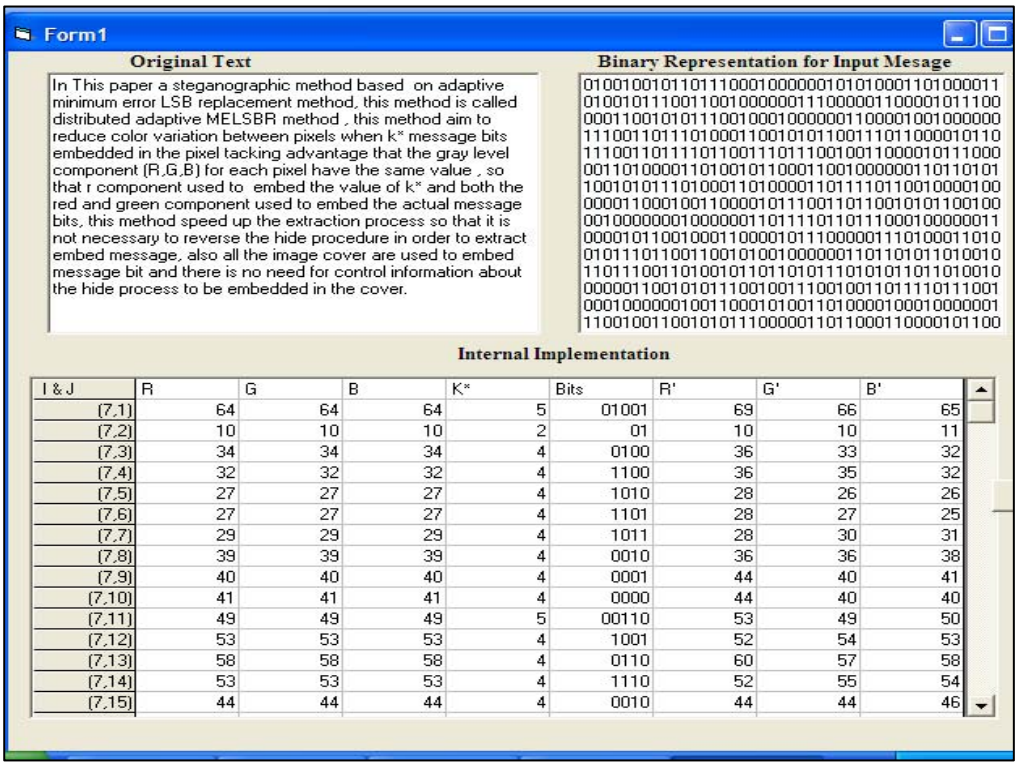


Figure (3.a) an experimental result of proposed method on gray scale image, MSE = 0.43, SNR = 47.075, PSNR = 51.561.



I & J	R	G	B	K*	Bits	R'	G'	B'
(7,1)	64	64	64	5	01001	69	66	65
(7,2)	10	10	10	2	01	10	10	11
(7,3)	34	34	34	4	0100	36	33	32
(7,4)	32	32	32	4	1100	36	35	32
(7,5)	27	27	27	4	1010	28	26	26
(7,6)	27	27	27	4	1101	28	27	25
(7,7)	29	29	29	4	1011	28	30	31
(7,8)	39	39	39	4	0010	36	36	38
(7,9)	40	40	40	4	0001	44	40	41
(7,10)	41	41	41	4	0000	44	40	40
(7,11)	49	49	49	5	00110	53	49	50
(7,12)	53	53	53	4	1001	52	54	53
(7,13)	58	58	58	4	0110	60	57	58
(7,14)	53	53	53	4	1110	52	55	54
(7,15)	44	44	44	4	0010	44	44	46

Figure (3.b) original text, the binary representation of it, internal implementation of proposed algorithm for the last row in embedding process
R= original red value
G= original red value
B= original red value
K* = No of bit embedded

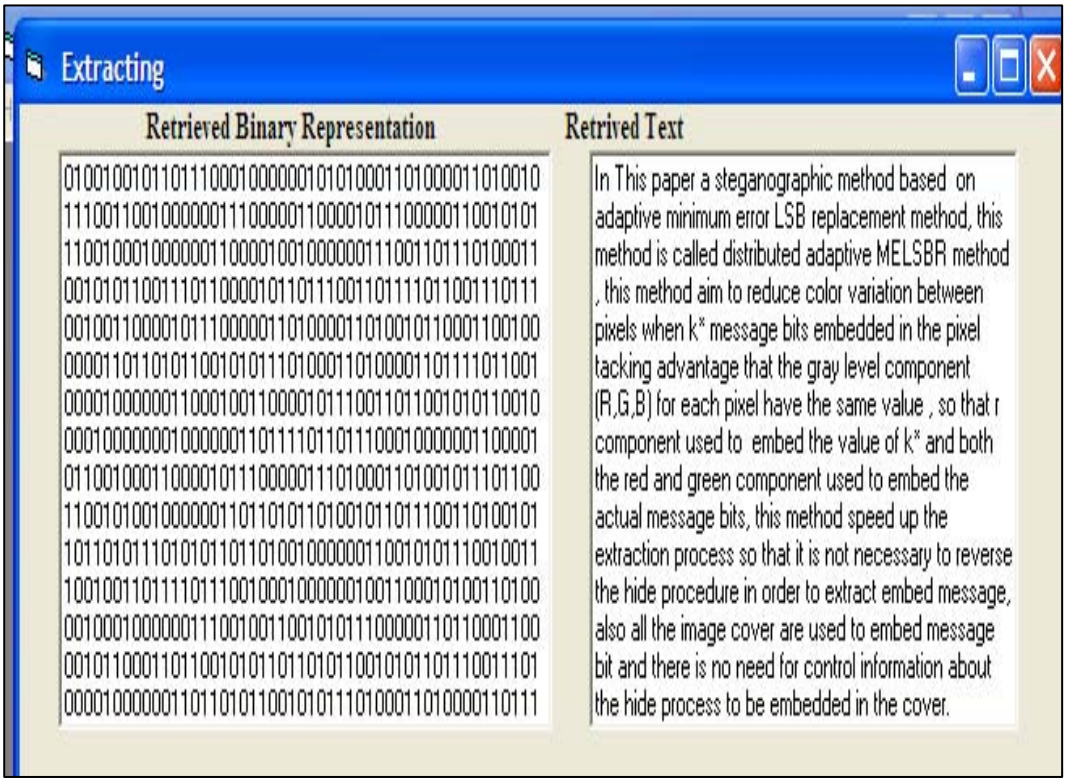


Figure (3.c) retrieved binary representation and retrieved text

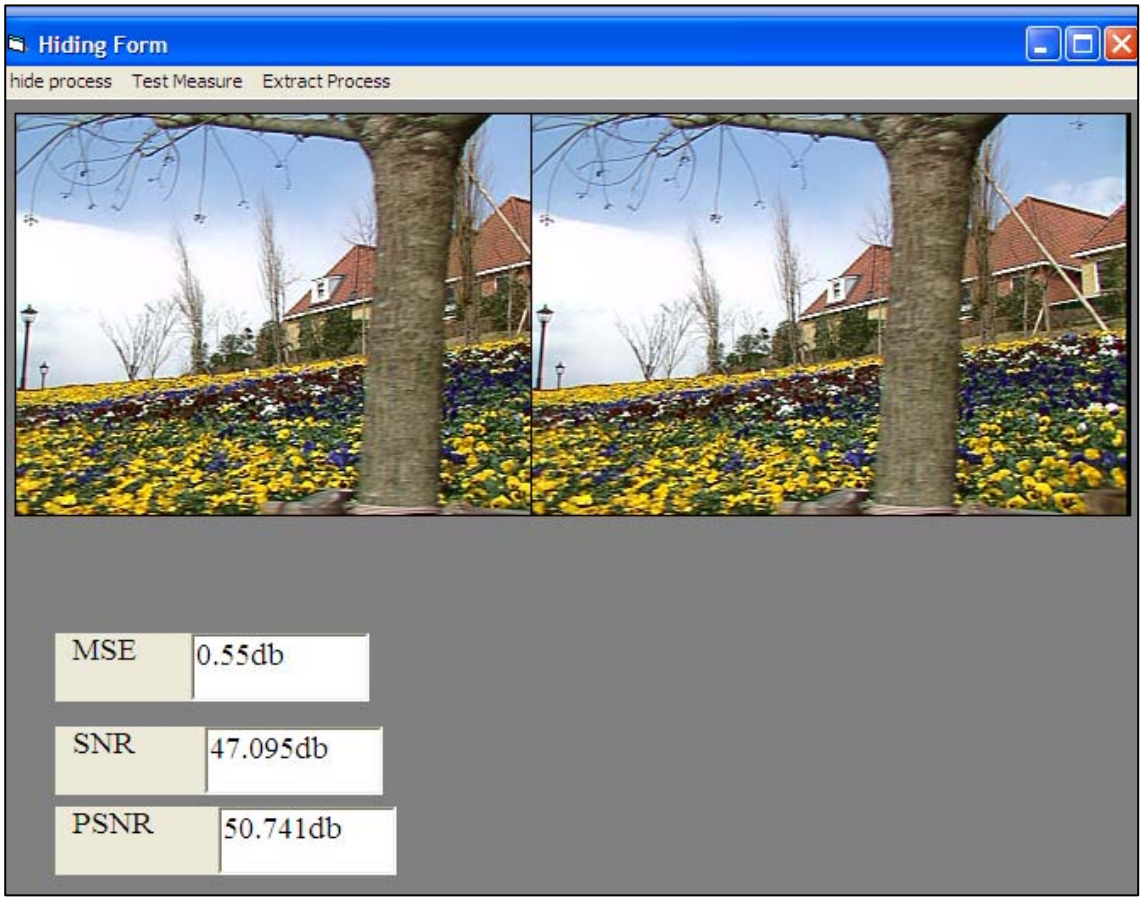
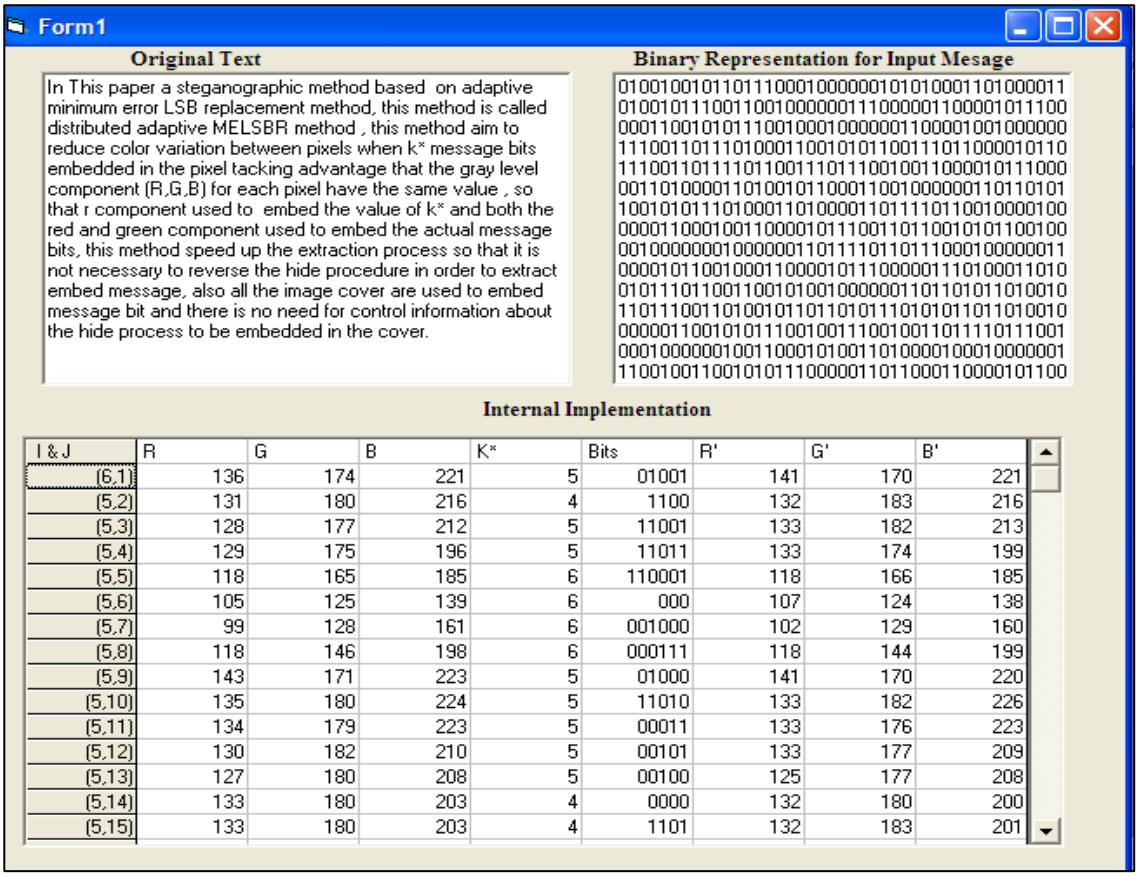


Figure (4.a) an experimental result of proposed method on color image, MSE = 0.55, SNR = 47.095, PSNR = 50.741.



I & J	R	G	B	K*	Bits	R'	G'	B'
(6,1)	136	174	221	5	01001	141	170	221
(5,2)	131	180	216	4	1100	132	183	216
(5,3)	128	177	212	5	11001	133	182	213
(5,4)	129	175	196	5	11011	133	174	199
(5,5)	118	165	185	6	110001	118	166	185
(5,6)	105	125	139	6	000	107	124	138
(5,7)	99	128	161	6	001000	102	129	160
(5,8)	118	146	198	6	000111	118	144	199
(5,9)	143	171	223	5	01000	141	170	220
(5,10)	135	180	224	5	11010	133	182	226
(5,11)	134	179	223	5	00011	133	176	223
(5,12)	130	182	210	5	00101	133	177	209
(5,13)	127	180	208	5	00100	125	177	208
(5,14)	133	180	203	4	0000	132	180	200
(5,15)	133	180	203	4	1101	132	183	201

Figure (4.b) original text, the binary representation of it, internal implementation of proposed algorithm for the last row in embedding process
R= original red value
G= original red value
B= original red value
K* = No of bit embedded
R'= new red value
G'= new red value
B'= new red value

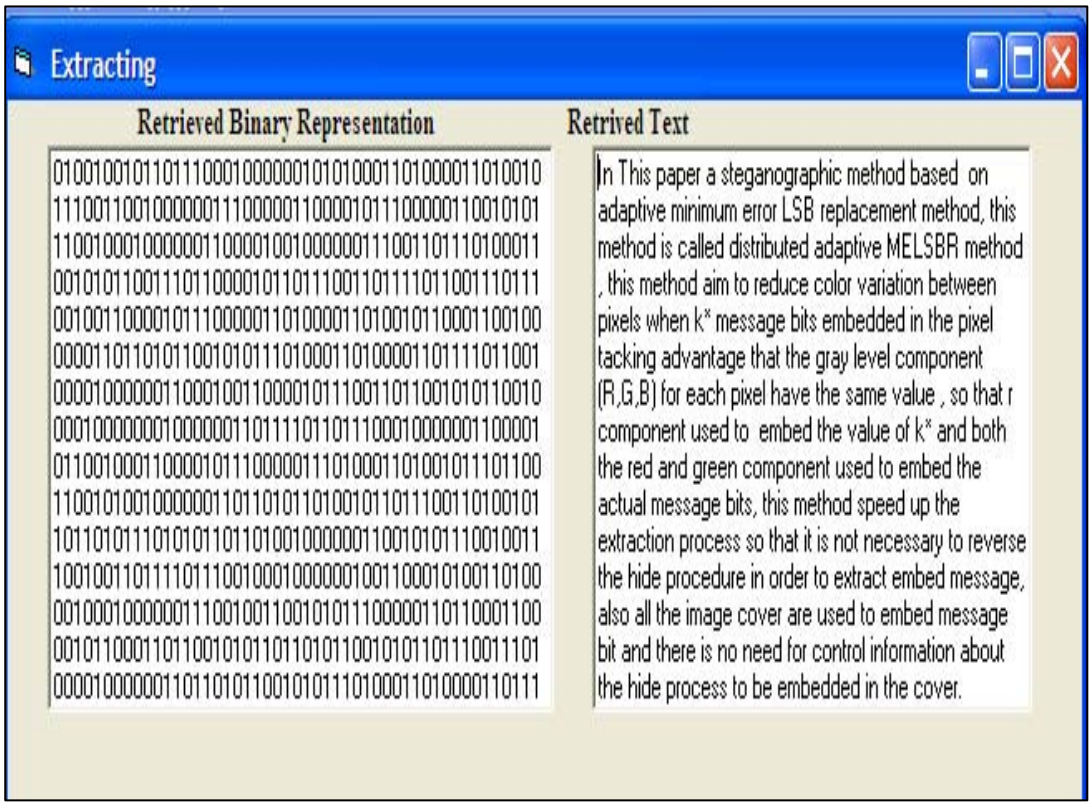


Figure (4.c) retrieved binary representation and retrieved text

Table (1) Message size, Time for Embedding and Extracting for Different Stego Covers

Stego cover	Message size In bit	Image Type	Time for hide process In second	Time for extract process In second
Face one	8840	Gray scale	8	6
lena	9200	Color image	9	7
renata	14789	Color image	5	5

6. Conclusion

In this paper we proposed steganographic method for hide text in image this method depend on the AMELSB method, the proposed method takes the advantage of gray level distribution and local characteristic of cover-image to embed maximal amount of image size and speed up, facility the process of extract and retrieve message where there is no control information needed to be hidden the cover-image.

7. References

1- Anders W., Gritta W., 1998, "Steganography in Video Conferencing System, LNCS 1525,pp.32-47.

2- Joshua Silman, 2001 “Steganography and Steganalysis: An Overview”, http://www.sans.org/reading.room

3- Stefan K. Enbcisser and Fabien A. Petitcolas, 2000 ,“Information Hiding Techniques for Stcganography and Digital Watermarking”, Artech house Inc,USA.

4. Yeuan-Kuen L. & Ling-Hwei C., 1999, ”An Adaptive Image Steganographic Model Based on Minimum-Error LSB Replacement”. http://debut.cis.nctu.edu.tw/Publications/pdfs/C14.pdf