R.F. Hassan
Computer Science, Department University, of Technology, Iraq
surorh@yahoo.com

M.S. Mohammed
Computer Science, Department, University of Technology, Iraq
mayalrabi@yahoo.com

Received on: 15/11/2015
Accepted on: 29/12/2016

Information Hiding Using Geographic Information System (GIS) Vector File

Abstract-*There are different techniques for securing data like cryptography and information hiding (steganography and watermarking) which has received more attention and faced many challenges. In this paper, an efficient digital steganography method has been proposed, where the Geographic Information System (GIS) files used as a cover media. This method depends on hiding text file in a map vector coordinate using ESRI (Environmental Systems Research Institute) Shape file, which stores the geometry of the digital features as sets of vector, coordinates. The method is based on changing the value of unspecific order bits depending on an Input location. Since we are interested in maximizing capacity and ensure robustness requirements. Exploiting the advantage of double percentage number capacity in the 2Dimension vector file was one of the main goals of this research. A Steganography techniques requirement was satisfied since changing maps did not raise any suspicion, while they do not alter the original data content.*

Keywords-*Geographic information system; steganography; map vector coordinate, ESRI Shape file format*

How to cite this article: R.F. Hassan and M.S. Mohammed, "Information Hiding Using Geographic Information System (GIS) Vector File," *Engineering and Technology Journal*, Vol. 35, Part B, No. 2, pp. 182-188, 2017.

1. Introduction

With progress of technology and use of computer in many branches of life and work. This technology require the dvelopment of information security. The discussed in information security is the exchange of information through a cover media using steganography [1].

Steganography is used to hide information in the cover flier or media in a way that other person will not notice the presence of data and could not access to the secret information. . This is a main distinction between steganography and the other methods of exchanging information because, for example, in cryptography, anyone can notice the coded information but they will not be able to understand that information. However, the most steganography was used to hide information on (images, video, texts file, sounds and music). In addition, using the steganography with the other technical has improved the information security such as copyright, preventing e-document forging etc. [2].

Figure 1 shows the Steganography techniques that are used with the different ways that can use this technology [3]. On the other hand, the term (GIS) represents geographical and spatial data it is a digital maps. Each map represents a set of features stored as couples <*attribute*, *value* > for each position in the map. These features may be organized into different layers [4].

Figure 2 illustrates the representation of (GIS) as a collection of many and different layers. In this figure, the map is composed of four layers: town's layer, a rivers layer, a regions layer and a spatial coordinate system layer. Each of them can be visualized together or separately. In this paper represent a low distortion, high capacity, secure, and robust steganography technique with 2D vector coordinate map, is used an efficiently to hide information where shape files of ESRI GIS are used to hid text file or data represented in a binary form [6]. The rest of this paper is organized as follows. In section 2, related works in section 3 are described the main proposal method including the information embedding process and extraction procedure is presented, in section 4. Experimental results are shown, in section 5 the error analysis after hiding data in cities file format, followed by section 6. H.W and S.W, and finally the conclusion and suggestions for future work in section 7.

https://doi.org/10.30684/etj.2017.138666
2412-0758/University of Technology-Iraq, Baghdad, Iraq

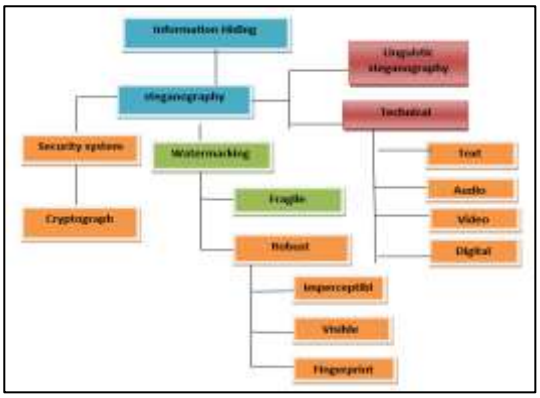

Figure 1: Different branches of steganography

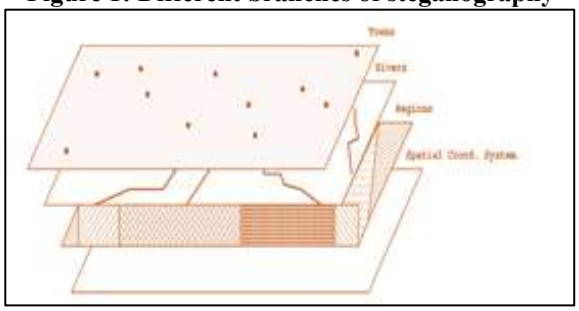

Figure 2: The digital map is built from a set of layers [5]

2. Related Work

There is many hiding information schemes for 2D models that have been proposed. These schemes are embed the messages by relying on additive or substitutive schemes.

Muttoo1 and Kumar suggested a method to produce map digest and it's embedding in the map preserves all the basic properties of the original map in the watermarked map [7].

Kasto presented the implementation of a vector stenographic method; that secret message is hidden in plain drawing (AutoCAD drawing) with vector coordinate used as stego cover. This paper focuses on trading a few least significant bits of original floating-point values and replacing them with a secret text. Its implemented by transform each character in the text into numbers (using RSA algorithm), then stores it as part of floating-point numbers embedded with plain drawing [8].

Cayre and Macq their scheme extends and enriches one of the simplest techniques, called the triangle strip peeling sequence (TSPS). The basic idea behind the TSPS algorithm is the insertion of bits while moving on the mesh, and there is a blind scheme in the spatial domain, the key idea is to consider a triangle as a two-state geometrical object [9].

3. The Proposed Method

This paper constraint on the hiding messages within a vector data problem. Vector data use (X-Y) coordinates as an orders of points, lines, and regional boundaries on a maps used extensively by geographic information systems. Therefore, this search proposes a new method for hiding a secret message within (X-Y) coordinate in GIS maps.

This section describes in details the proposed method for efficient information hiding in a GIS Point type model in shape file, by using the most familiar file format (shape file) the spatial data format which has the file extension (*.shp) .

As a pre-process a suitable shape, file with a p oint layer should be selected first with the secrete text file message, the text message have to convert to a binary code after embed a padding like" aaaa" to indicate the end of the message.

Open the files in binary mode and read one bit (0, 1) from message and one byte from shape file depending on location byte (B_k) input from program then write result in a new file .this method called (2 bits per point) and other method which is 4 bits per point .this methods which read 2 or 4 bits from message in time and embed in the shape file then write the result in a new file.

The embedding operation is begin by selecting the specific byte location (B_k)from the X or Y coordinate value and changing it with the message bits, selecting 2 or 4 bits per point from the choosing layer. Since each point is in shape file represented by X, Y coordinates, which are represented by a double precision floating-point format (8 bytes or 64 bits), after each embedding the file pointer should be moved to, the next 8 bytes depend on given byte location (B_k). This method of embedding process will be performed without affecting the positional accuracy of the original data, where it deals with making tiny changes in vector coordinates of the plain vector map. The Algorithm named Unspecific byte Location (UBL) Depends on an Input Location from (0 to 6) byte, which read the GIS file as Binary file (0, 1) and embeds the secrete message also as a binary code is illustrated in Algorithm (2), it should be performed after algorithm (1) that explains the preprocess steps required before embedding and extraction process.

Algorithm (1): preprocess operation.
In put: Secrete message as Text file, and shape file contain point layer.
Output: secrete message as a binary code, and reaching the X and Y coordinates of each point in the layer
Begin
Step 1: Read the secrete text message.
Step 2: Add "aaaaa" to the end of the message, for the extraction purpose.
Step 3: Convert each character of message to its equivalent binary code.
Step 4: Specify number of bits (2 or 4) to be hiding.
Step 5: Open the shape file and Access the file header, Reading (100) bytes, i.e., moving the file pointer (100) Bytes.
Step6: Reading (8) bytes, which is (4) bytes for record number, and the next (4) bytes for record length.
Step7: Reading (4) bytes for the shape type.
Sttep8: The next (8) Bytes for X_Value and (8) Bytes for Y_value (X,Y are Double Precision format) .
End.

Algorithm (2): Embedded operatiot
Input: Text file as Secrete message, and shape file as a cover file.
Output: setegocover file.
Begin
Step 1: Call Algorithm (1): preprocess Algorithm.
Step 2: loop until message = nothing.
Step 3: Select 2 or 4 bits from the secrete message (0, 1).
Step 4: Input byte number location = K (0 to 6).
Step 5: Move the file pointer to selecting byte (k).
Step 6: Embed the bits in the selecting byte (k).
Step 7: Move the pointer (7-k) bytes the rest of 8 byte
step 8: Loop
End.

Algorithm (3): Extract operation
Input: stegocover file
Output: secret message
Begin
Step 1: Open the stegocover file.
Step 2: Call Accessing Algorithm (2)
Step 3: Str = nothing
Step 4: Loop until Str = "aaaaa"
Step 5: Move the file pointer to byte number (k).
Step 6: Extract the select bits from byte number (k).
Step7: Concatenate the extract bits
Step 8:.Convert bits to it is correspond char.
End.

4. Experimental Results

For showing that the proposed technique is very efficient, secure, and applicable, some experiment had been implemented and the results were examined, since the X_Y format is Double Precision that mean each one has (64 bits - 8 Bytes) length that lead to wide rang can be used to hide. The byte location number input (B_k) value should be between 0 to 6, by using the fraction part only; the distortion is increased by increasing B_k from 0 to 6. In addition, this Algorithm has high capacity, because each point can hide 2, 4, or 8 bits as will be shown in the following example.

Example 1.
The first 4 points X_Y _Coordinates is from Shape file (Cities. shp) which is Point type file

Original Values before hide process		
NO	X	Y
1	-122.466903686523	48.7440490722656
2	-109.679100036621	48.5438194274902
3	-122.629402160645	48.4923896789551
4	-122.314498901367	48.4217300415039

Table 1: The Original first 4 points X_Y Values

Precision floating Point Format the formula to calculate X or Y is as the following formula:

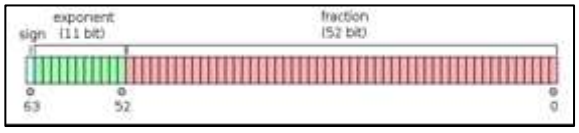

Figure 3: Double Precision format

$$(-1)^{sign}\left(1+\sum_{i=1}^{52} b_{52-i}2^{-i}\right)\text{x } 2^{e-1023} \quad (1)$$

[10]

Table 2: Original Values in Binary form

Original Values in Binary form before hide process		
NO	X	Y

1	11000000010111101001110111100001	01000000010010000101111100111101
	11000000000000000000000000000000	00000000000000000000000000000000
2	11000000010110110110101101110110	01000000010010000100010110011011
	01100000000000000000000000000000	11100000000000000000000000000000
3	11000000010111101010100001001000	01000000010010000011111100000110
	00100000000000000000000000000000	10100000000000000000000000000000
4	11000000010111101001010000100000	01000000010010000011010111111011
	11000000000000000000000000000000	01000000000000000000000000000000

Now let say that the secrete message is [m=109=01101101] want to hide according to the proposed algorithm least significant bit (LSB), 2 bits per point and according to the Byte location input number k=0.

Table 3: New Values after hiding process

New Values in Binary form after hiding process		
NO	X	Y
1	11000000010111101001110111100001 11000000000000000000000000000000	01000000010010000101111100111101 00000000000000000000000000000001
2	11000000010110110110101101110110 01100000000000000000000000000001	01000000010010000100010110011011 11100000000000000000000000000000
3	11000000010111101010100001001000 00100000000000000000000000000001	01000000010010000011111100000110 10100000000000000000000000000001
4	11000000010111101001010000100000 11000000000000000000000000000000	01000000010010000011010111111011 01000000000000000000000000000001

Table 4: Calculating process of X,Y values for 2 bits per point and according to the byte location input number k=0.

Calculating X,Y values New Values after hiding process		
NO	X	Y
1	New(X)=Old(X)= -122.466903686523	New(Y)=Old (Y)+(+1)×2^{-52}×2^{e-1023} =48.7440490722656+(2^{-52} ×2^{5}) =48.7440490722656 +2^{-47} =48.7440490722656 +0.00000000000000710 ≅48.7440490722656
2	New(X)=Old (X)+(-1)×2^{-52}×2^{e-1023} = -109.679100036621 - (2^{-52}×2^{6}) = -109.679100036621 - (2^{-46}) = -109.679100036621 - 0.0000000000000142 ≅-109.679100036621	New(Y)=Old(Y)= 48.5438194274902

Since X-Y are Double Precision format and the max number of digits is 15 for this reason, the values not change because the addition value is very small and it will not effect.

Table 5: Original Values in Binary form

Original Values in Binary form before hide process		
NO	X	Y
1	11000000010111101001110111100001 11000000000000000000000000000000	01000000010010000101111100111101 00000000000000000000000000000000
2	11000000010110110110101101110110 01100000000000000000000000000000	01000000010010000100010110011011 11100000000000000000000000000000

Table 6: New Values after hiding process

New Values in Binary form after hide process		
NO	X	Y
1	11000000010111101001110111100001 11000000000000000000010000000000	01000000010010000101111100111101 00000000000000000000010000000000
2	11000000010110110110101101110110 01100000000000000000011000000000	01000000010010000100010110011011 11100000000000000000001000000000

Now let say the Message is [m=109=01101101] want to hide Using (4 bits per point) and according to the Byte (B_1) Location input number K=1

Example 3: Table 7 and Figures 4 and 5, calculating process of X, Y values for 4 bits per point and according to the Byte location input number k=1.

Table 7: Example of hiding 4 bits per point / Shape file point type

Hide/ Unhide 4 bits Per Point and Select Which Byte you Want to use to Hide Text 2 bit in X ,2 bit in Y Point Type (Cities.shp)				
In this example the Byte Number is 5 and the Message is [my account number is 12345] The first 4 points				
	X_Value Before hide	Y_Value Before hide	X_Value After hide	Y_Value After hide
1	-١٢٢.٤٦٦٩٠٣٦٨٦٥٢٣٤٣٦٨	٤٨.٧٤٤٠٤٩٠٧٢٢٦٥٦٢٥٦	-١٢٢.٤٥١٢٧٨٦٨٦٥٢٣٤٣٦٨	٤٨.٧٤٤٠٤٩٠٧٢٢٦٥٦٢٥٦
2	-١٠٩.٦٧٩١٠٠٠٣٦٦٢١٠٩٤٤	٤٨.٥٤٣٨١٩٤٢٧٤٩٠٢٣٦٨	-١٠٩.٦٧٩١٠٠٠٣٦٦٢١٠٩٤٤	٤٨.٥٣٦٠٠٦٩٢٧٤٩٠٢٣٦٨
3	-١٢٢.٦٢٩٤٠٢١٦٠٦٤٤٥٣٧٦	٤٨.٤٩٢٣٨٩٦٧٨٩٥٥٠٧٨٤	-١٢٢.٦٤٥٠٢٧١٦٠٦٤٤٥٣٧٦	٤٨.٤٩٢٣٨٩٦٧٨٩٥٥٠٧٨٤
4	-١٢٢.٣١٤٤٩٨٩٠١٣٦٧١٨٧٢	٤٨.٤٢١٧٣٠٠٤١٥٠٣٩٠٧٢	-١٢٢.٣١٤٤٩٨٩٠١٣٦٧١٨٧٢	٤٨.٤٢١٧٣٠٠٤١٥٠٣٩٠٧٢

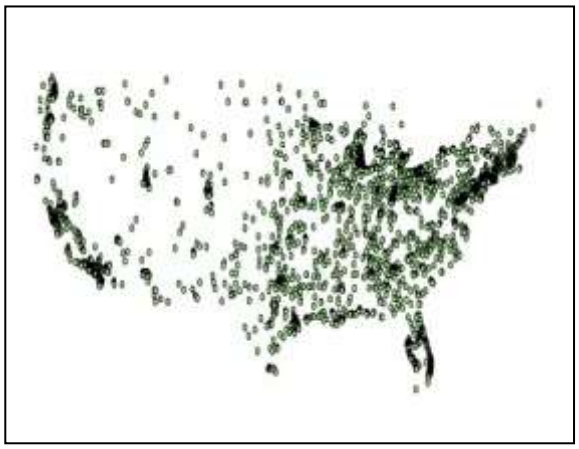

Figure 4: Cities Shape file before hiding 4 bits per point

Figure 5: Cities Shape file after hiding 4 bits per point

I. Error Analysis

Image, video and any digital media quality assessment occurs when you have to measure the degree of fidelity of an encoded copy of a picture or a digital media against their original version. It also occurs when you have to evaluate the performance of a compression algorithm and/or moreover, when you have to understand the degree of acceptable impairments that can affect an image or a digital media (in example, the errors introduced by a noisy transmission channel).

Quality assessment techniques are applied layer by layer in the case that the frames of the encoded copy and the original copy have to be perfectly and have to be in the same image format to the human eyes, in case of picture evaluation, the requisite is only the image format. In this work, it is considered only the X-Y vector, since the human eye is far more sensitive to the presence of noise and distortions in brightness.

II. Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR)

MSE and PSNR are the algorithms historically adopted in image processing in order to evaluate the performance of the codec of interest; They are closely linked to and borrowed from other contexts of signal processing. Although simple to implement and calculate, they show the side in different situations, so the findings cannot be considered always reliable. Nevertheless, their use continues to be predominant in the performance evaluation of any coding system. Let X and Y two arrays of size NxM, respectively representing the Y channel frame of reference (i.e. the original copy) and Channel frame of the encoded/impaired copy. The mean square error between the two signals is thus defined as:

$$MSE = \frac{1}{NxM}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[X(i,j) - Y(i,j)]^2 \quad (3)$$

The more Y is similar to X, the more MSE is small. Obviously, the greatest similarity is achieved when MSE equal to 0. PSNR is so defined as:

$$PSNR = 10 \log 10 \frac{12}{\text{MSE}}$$
(4)

L reflects the range of values that a pixel can take: for example, if the Y channel is encoded with a depth of 8bit, then L = 2^8 -1= 255. It's evident from the formula that the result is expressed in decibels. A small mean square error results in a high signal to noise ratio, if MSE tends to zero, then PSNR tends to infinity. Excellent values range from 30 to up depending to the dB, while an acceptable range in wireless transmission settles around 25dB [11].

Table 8: The error analysis for cities file format (point type)

Cities Shape File					
Message Size 10 sentences " my account is 12345 number"				= 2280 bits = 285 bytes = 285 char	
	X -coordinate			Y- coordinate	
Byte Number		MSE	PSNR	MSE	PSNR
2	2 bits per point	3.59E-10	136.3	1.98E-10	130.7
	4 bits per point	4.98E-10	134.9	4.09E-10	127.6
3	2 bits per point	9.19E-08	112.2	5.08E-08	106.6
	4 bits per point	1.27E-07	110.82	1.04E-07	103.5

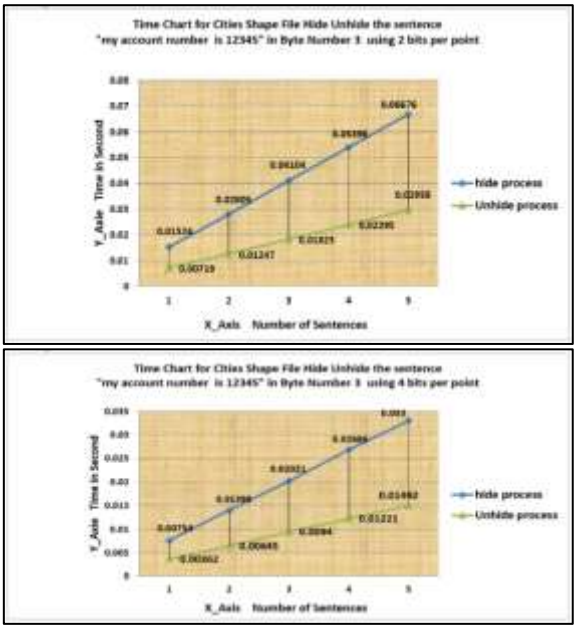

Figure 6: charts of time consuming before and after hiding 2 bits per point

From the two charts above The time consume to embed Message is bigger than the time to retrieve it because there is multi operations process like (Mod,-,+) and convert each bit from string form ("0","1") into numerical form (0,1) but in embedding process We have only mod operation regardless the other operations because it is same in both process Hide and UnHide. Also we see that the time is less by half ($^1/_2$) when we use the 4 bits per point method comparing with 2 bits per point because with the first method (2 bits per point) we need (n) number of main process which is n=length of the string (message) because the process is one bit per time but with (4 bits per point) we need ($^n/_2$) number of process since the process is 2 bits per time [12].

III. H.W and S.W and experimental results

We implemented the proposed technique using Microsoft Visual Basic six (VB6) programming language. We also performed experiments to validate the feasibility of our algorithms. Results were collected on a personal computer Laptop with 1.7 GHz processor and (1) MB memory. Also used an ActiveX Control MapWinGIS v4.8.8SR-32bit it which is combatable with VB6 to Edit Or view the GIS file (File.shp).

To view the GIS file need another 2 files (index file and database file) the index (Shx) file contain the offset address of points and the database (.dbf) file contain all data for the shape file such as (names, length…) it is FoxPro database for this reason used 2 API functions (Delete and Copy) to

delete old file and to copy these 2 files. Beside that used some other API functions like Query Performance Frequency to 'get number of counts/second (the frequency of the CPU) and Query Performance Counter to 'get current count number of the CPU, then the time equal (Query Performance Counter1-Query Performance Counter2)/Query Performance Frequency.

From Chart time found the 4 bits method is more efficient (less time consumer) than 2 bits. About security best to select an input between (0 to 6) and greater than zero because number 0 mean first byte (B0) least significance bit (LSB). Also the distortion start to show when use the byte number 5 or 6 that mean the best choice is in the middle.

From the Figure 7, 8 and 9 there is no different between Original file and the new one i.e the human's eyes can't recognize or Catch the difference between Old and new position. To view or show the X-Y values of the points of shape file in Decimal or binary form we used another API function called Shell Execute to Run Any Executable file since the result saved on Notepad file ones in X_Y Coordinates and second in Binary form Automatically.

5. Conclusion and Future Work

In this paper, we have presented an efficient digital steganography technique for 2D GIS file type (Point) our technique provides information hiding with efficiency, high capacity, security, low distortion, and automaticity. In future we can develop the Algorithm to be more efficient and change it by use GIS files different types like (polyline or Polygon) which become more secure and high capacity and more efficiency because the bit order number from 1 to 52 available to hide message. In addition, we can use permutation the message before the process. another idea is to read shape file and duplicate it (duplicate value of X,Y) or add some points and select a specific value of X,Y as cover for example even or odd order .finally the combination between Steganography and cryptography to be complete secure we recommend to use RSA ,DES or any crypto method to encrypt the secrete message befor hiding it in the spatial file format

Figure 7: Original map file (Cities .shp) for United States of America

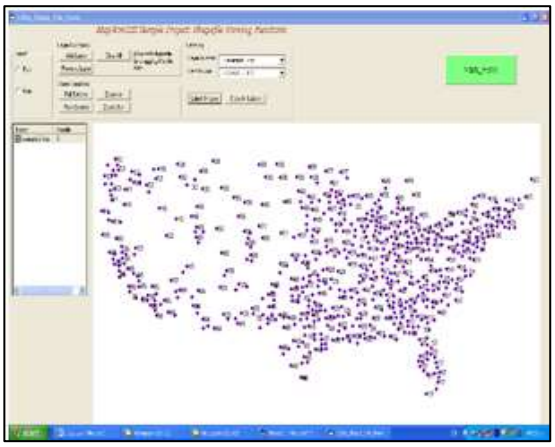

Figure 8: The New file after hide using ٢ bits per point using B_0 and sentence "my account number is 12345

Figure 9: The New file after hide using 4 bits per point using B_3 and sentence "my account number is 12345

References

[1]. S.C. Ajay and J.S, (Steganography an Art of Hiding Data) International Journal on Computer Science and Engineering Vol. 1, No. 3, 137-141, 2009.

[2]. J. Marks, M. Alex, M. Wang & Y. M. Cheng (An Efficient Information Hiding Algorithm for Polygon Models) 592 / EUROGRAPHICS, 2005.

[3]. S.M. Wang, F.M. Chen and K.W. Chen, (Using Reversible Steganography Algorithm to Embed

Metadata in Vector Maps) department of computer science and engineering… vol.3, no. 5, 2009.

[4]. M. C. and L.V, (Information Hiding for Spatial and Geographical Data) Mancini University of Rome "La Sapienza", Rome Italy), 2009.

[5]. K.W Chen., S.M. Wang, and C.M. Wang, (A Reversible Data Hiding Algorithm for Vector Maps) Information Security Conference 2007, Chiayi, Taiwan, 2007.

[6].Available http://www.digitalpreservation.gov/formats/fdd/fdd000280.shtml [ESRI Shape file]

[7]. P.C. and W. Zeng, (Image Adaptive Watermarking Using Visual Models) IEEE Journal on Selected Areas in Communications, Vol. 16, No. 4, 525-540, 1998.

[8]. N.F. Kasto, (Hide Information Using Plain Drawing) Department of Computer Science, College of Science, University of Baghdad. Journal of Science, Vol.48, No.1, 205-212, 2007.

[9].C.F. MACQ (Data Hiding on 3-D Triangle Meshes). IEEE Trans. Signal Processing 51, 4, 939-949, 2003.

[10]. D.A. Patterson and J.L. Hennessy, "Computer Organization and Design", University of California, Berkeley, Revised fourth edition, IEEE 754 double-precision binary floating-point format: binary64, 2010.

[11]. E. C. Personal Home, " Image and Video quality assessment – Part One: MSE & SNR" ,http://Emanuele colucci.com/image and video quality. Posted on (April 10, 2011.

[12]. D. Roberts, "Slopes and Equations of Lines" Oswego City School District Regents Exam Prep Center, 2010.

Author's biography

Rehab F Hassan, PhD of Computer Science. Asset prof. of Computer Science Department in University of Technology, Bagdad, Iraq.

May S. Mohammed, MSC of Computer Science, Assit Lecturer of Computer Science Department University of Technology, Baghdad, Iraq .