International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

A Steganography Implementation Based On DCT Algorithm By Using MATLAB GUI Function

Er. Yashpal Lather[1], Er. Megha Goyal[2]

[1]Research Consultant, M.Tech (Optical Engg.), PGDIBO
[2]Assistant Professor, ECE Department, Galaxy Global Group of Institutions, Ambala, India

Abstract: ***Informally, steganography refers to the practice of hiding secret messages in communications over a public channel so that an eavesdropper (who listens to all communications) cannot even tell that a secret message is being sent. In contrast to the active literature proposing new concrete steganographic protocols and analysing Laws in existing protocols, there has been very little work on formalizing steganographic notions of security, and none giving complete, rigorous proofs of security in a satisfying model. This paper USED A image domain algorithms available for implementing Image Steganography. Covert communication is taking place by encrypting the password for information to be protected. The intended receiver will decrypt the information using that password.***

Keywords: Image Steganography, Graphic User Interface, Project Work, Future Work

1. Introduction

Although steganography is an ancient subject, the modern formulation of it is often given in terms of the prisoner's problem proposed by Simmons, where two inmates wish to communicate in secret to hatch an escape plan. Steganography is the science of hiding information. Whereas the goal of cryptography is to make data unreadable by a third party, the goal of steganography is to hide the data from a third party. Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. 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 this, is called steganography. Steganography 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. In image steganography the information is hidden exclusively in images.

2. Image Steganography

As stated earlier, images are the most popular cover objects used for steganography. In the domain of digital images many different image file formats exist, most of them for specific applications. For these different image file formats, different steganographic algorithms exist.

To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image. This numeric representation forms a grid and the individual points are referred to as pixels. Most images on the Internet consists of a rectangular map of the image's pixels (represented as bits) where each pixel is located and its colour. These pixels are displayed horizontally row by row. The number of bits in a colour scheme, called the bit depth, refers to the number of bits used for each pixel. The smallest bit depth in current colour schemes is 8, meaning that there are 8 bits used to describe the colour of each pixel. Monochrome and greyscale images use 8 bits for each pixel and are able to display 256 different colours or shades of grey. Digital colour images are typically stored in 24-bit files and use the RGB colour model, also known as true colour . All colour variations for the pixels of a 24-bit image are derived from three primary colours: red, green and blue, and each primary colour are represented by 8 bits. Thus in one given pixel, there can be 256 different quantities of red, green and blue, adding up to more than 16-million combinations, resulting in more than 16-million colours . Not surprisingly the larger amount of colours that can be displayed, the larger the file size.

Image and Transform Domain: Image steganography techniques can be divided into two groups: those in the Image Domain and those in the Transform Domain. Image – also known as spatial – domain techniques embed messages in the intensity of the pixels directly, while for transform – also known as frequency – domain, images are first transformed and then the message is embedded in the image. Image domain techniques encompass bit-wise methods that apply bit insertion and noise manipulation and are sometimes characterized as "simple systems". The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format. Steganography in the transform domain involves the manipulation of algorithms and image transforms. These methods hide messages in more significant areas of the cover image, making it more robust. Many transform domain methods are independent of the image format and the embedded message may survive conversion between lossy and lossless compression.

3. Graphic User Interface

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding

programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed.

The GUI components can be menus, toolbars, push buttons, radio buttons, list boxes, and sliders—just to name a few. GUIs created in MATLAB® software can group related components together, read and write data files, and display data as tables or as plots.

The GUI contains:

- An axes component.
- A pop-up menu listing three data sets that correspond to MATLAB functions: peaks, membrane.
- A static text component to label the pop-up menu.
- Three buttons that provide different kinds of plots: surface, mesh, and contour.

When we click a push button, the axes component displays the selected data set using the specified type of 3-D plot.

4. Project Work

This includes processing of image Data hiding, it means you are now able to hide your data in pixels of image. As it is known to all of persons that has already gone through some of basic image processing that your image has its color part as a combination of Red, Green and Blue values and also your brightness factor is related to it. We can use any of these available values, as in our case we gone through the blue part, we hide our data in byte values of pixels values for colour, As by using 'imread' function you can get any image in matrix form and that matrix will contain three different arrays, First array will be of Red, second of Green and third of Blue respectively. So to save our data like if you have to save 'Hello' in an image you will firstly read it in matrix form and then read last or you can say third matrix and after reading third matrix its dimensions will depends on the image size, as per according to image's height and width.

The matrix will contains values i.e. result of imread like shown below

{122, 122,144,123,205,55,35,88,22,225,125,223,……}, for red
{142,182, 104,193,105,255,035,188,22,25,15,33,……}, for green
{ 22, 12, 44,13,05,54,135,188,122,25,125,223,……}, for blue

Use blue one:

{ 22, 12, 44,13,05,54,135,188,122,25,125,223,……}, for blue
'Hello' = {72 101 108 108 111},
So new value for blue one is
{ 5,72, 101, 108,108,111,54,188,122,25,125,223,……}, for blue

Stego key = 5

Stego key is useful for decoding time. In our case it tells us how many values has to read during decoding time.

For all process we divide it I to steps and named it as hider and decoder. Hider file is a GUI which contain a user friendly environment for the user to hide its data in image which should be select by the user too by load image button than there is Pre Analysis which gives results as how many bytes can be hide in selected image, After that you have to load text file by pressing load text button, this will load text file and after pressing hide button this will add text file to your image, Right side status of your adding process will be shown to you. Status will show processing, Not Done or Done. After getting 'Done' your hider can exit, now copy your image file with named as 'code_image.png' or bys selecting path of your code image by browser after pressing load image same as you did in 'hider'.

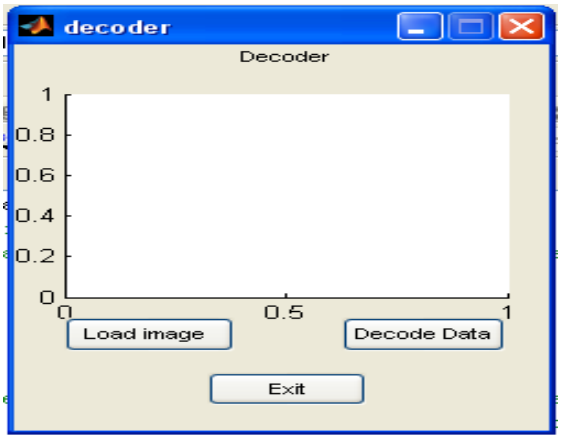

Figure 1: Decoder

In decoder figure you have three buttons Load image, Decode Data and Exit. On pressing load image you got a browser window in that you can select your image file by selecting its path. The path you will give must contain an image, so image will be shown on your decoder axes. After pressing decode data button your decoded.txt file will be there in same folder where your code of decoder is placed, after you receive decoded text file you can exit and your text file will contain the data entered by you or user on encoder or hider side.

The steps for making any GUI:

All these filters are made with the help of Mat Lab. The first step is to make a Front look for project which is called "GUI" as Graphical User Interface .As for this just open Mat Lab and then open a new 'GUI' file from file menu option, you will got this figure:

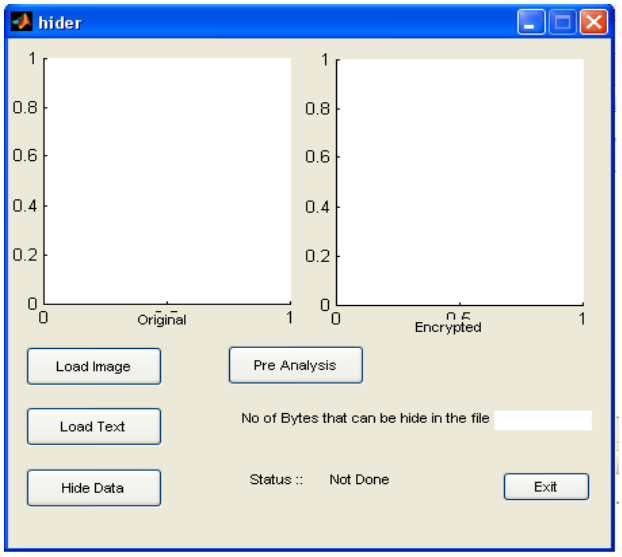

Figure 2: Main Hider

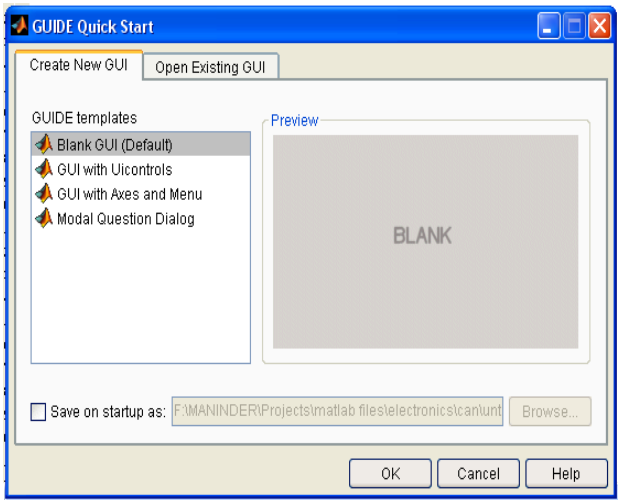

Figure 3: New GUI File

Now select 'Blank GUI(Default)' option in GUIDE templates and then ok after pressing ok you will get this figure:

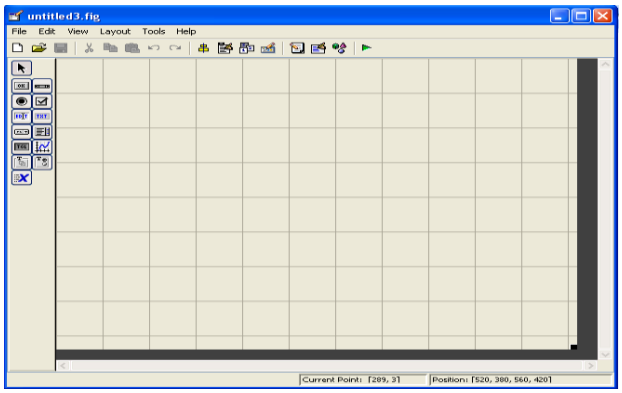

Figure 4: Blank GUI

On the left side of your GUI file there are tool box like switches that you can use accord to your requirement. Like if you need push button there are push button, if your requirement is popup menu or drop down box you can use that if you need some static text like you just need to write the name of figure just go for it, if you need some edit box, main requirement of your edit box is during your run time or you can say during project running like you have an image and you go for some changes in it than you can save your image with new name that you will enter during project running time. In short all u need, have to use. Its example is given in next step.

Now select suitable option i.e. Switches, Edit box, drop down boxes any other that you think required according to your projects requirements as below:

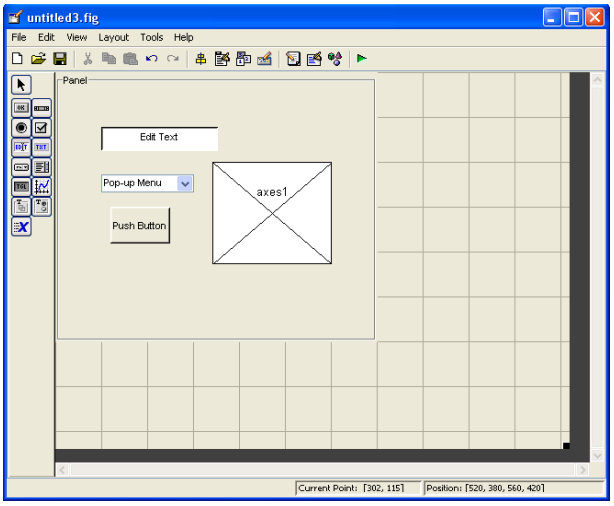

Figure 5: During Adding Required Featured Buttons

It is by just clicking on required switch or tabs and then click on desired location. After that just right click on your button or panel to set the properties etc.

After completing code u will have your GUI like one below.

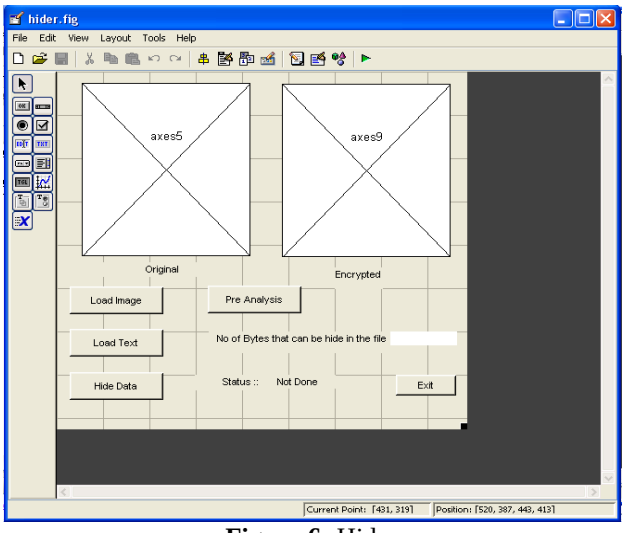

Figure 6: Hider

Now press run button from editor toolbar or RUN from debug button or simple Function 5(F5).You will have to save you fig file to save your front end .Just save and check the output after running it. You may ask to change directory.

This one is your output than open its saved file which is as of the same name of your '.fig' file. If you save your figure file with name 'PROJECT.FIG' than in same directory your 'PROJECT.M' file will be there , which is output of your Mat Lab processor.

5. Future Work

Due to time and computing limitations, we could not explore all facets of steganography and detection techniques. As you saw, we studied the power in our pictures to test for hidden data. Another method which we were unable to explore was to analyze the noise of the pictures. Adding hidden data adds random noise, so it follows that a properly tuned noise detection algorithm could recognize whether or not a picture had steganographic data or not.

6. Conclusion

We explored several steganography techniques and the various detection algorithms associated with them. By using the properties of the DCT and our understanding of the frequency domain we developed the zeros hiding method. Zeros hiding proved to be easier to analyze than bit-o-steg and can hide significantly more data. Unfortunately its ease of detection makes it a less secure method. After researching various techniques already implemented, we chose to improve upon one, thus creating our bit-o-steg method. Bit-o-steg can only hide data in coefficients that were not dropped, thus limiting the amount of data we can hide. However, it greatly enhances the effectiveness of the steganography since it uses a key, making it much more challenging to detect. In the end we found both effective, but the complexity of bit-o-steg makes it more promising. Detection of our methods was critical to the breadth of our project. By investigating the power in various components of our images we discovered how to detect data hidden via the zero hiding method. Detecting bit-o-steg required us to draw on past steganography research and statistically analyze the effects of this type of data hiding. The methods and accompanying detection schemes we developed broadened our understanding of steganography, which, unlike encryption, allows secret data to be traded hands without raising an eyebrow.

References

[1] Gary C.Kesler, “Steganography: Hiding Data Within Data”, September, 2001.
[2] Moskowitz, Ira S, “Information Hiding”, 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 2001 Proceedings (LNCS 2137).
[3] Johnson, Neil F., Duric, Zoran / Jajodia, Sushil G, “Information Hiding: Steganography and Watermarking”, Attacks and Countermeasures Advances in Information Security, Volume 1 2001.
[4] Katzenbeisser, Stefan Petitcolas, Fabien A. P “Information Hiding: Techniques for Steganography and Digital Watermarking”, 2000.
[5] M. J. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg, “Natural language watermarking and tamperproofing,”, Information Hiding: Fifth International Workshop, 9 November 2010.
[6] F. A. P. Petitcolas, ed., Lecture Notes in Computer Science 2578, pp. 196–212, Springer, October 2002.
[7] K. Bennett, “Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text,” Tech. Rep. TR 2004-13, Purdue CERIAS, May 2004.
[8] M. Topkara, G. Riccardi, D. Hakkani-Tur, and M. J. Atallah, “Natural language watermarking: Challenges in building a practical system,” in Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, January 2006.
[9] Grothoff, K. Grothoff, L. Alkhutova, R. Stutsman, and M. Atallah, “Translation-based steganography,” in Proceedings of Information Hiding Workshop (IH 2005), pp. 213–233, Springer, 2005