Steganography Using 32X32 Quantization Table

A Thesis submitted in partial fulfillment of the Requirements for the award of degree of

Master of Engineering
In
Electronics Instrumentation and Control

Submitted by
TARA BANSAL
(Roll No. 801151028)

Under the Guidance of
Ms. Ruchika Lamba
Lecturer

Department of Electrical and Instrumentation Engineering
Thapar University
(Established under the section 3 of UGC act, 1956)
Patiala, 147004, Punjab, India

July 2013

DECLARTION

I hereby certify that the work which is being presented in the thesis entitled, **"Steganography on Color Images using 32x32 Quantization Tables"** in partial fulfillment of the award of degree of **Master of Engineering in Electronics Instrumentation and Control** submitted in the Electrical and Instrumentation Engineering Department, Thapar University, Patiala is an authentic record of my own work under the supervision of Ms. Ruchika Lamba, Lecturer, Department of the Electrical and Instrumentation Engineering, Thapar University, Patiala, Punjab.

Date: 12/07/13

Tara Bansal

TARA BANSAL
801151028

I certify that the above statement made by the student is correct to the best of my knowledge and belief.

Ruchika

Ms. Ruchika Lamba
Lecturer
Department of the Electrical and
Instrumentation Engineering,
Thapar University, Patiala, Punjab.

Countersigned By

S. Ghosh

Dr. Smarajit Ghosh
Head of Department
Department of the Electrical and
Instrumentation Engineering,
Thapar University, Patiala, Punjab.

Dr. S.K. Mohapatra
Dean of Academic Affairs
Thapar University, Patiala
Punjab

ABSTRACT

In the present electronic communication scenario, data security is one of the major challenges. After the World War II, the need for a secure and robust communication between the communicating entities has increased due to the fear of terrorism. The publishers of digital audio and video are worried of their works being corrupted by illegal copying or redistribution, hence it is of primary importance to protect information. Cryptography is the method to hide secret data by scrambling so that it is unreadable, however it does not assure security and robustness as the hacker can obviously guess that there is a confidential message passing on from the source to the destination. Steganography is concealed writing and is the scientific approach of inserting the secret data within a cover media such that the unauthorized viewers do not get an idea of any information hidden in it.

Steganography is an alternative to cryptography in which the secrete data is embedded into the carrier in such way that only carrier is visible which is sent from transmitter to receiver without scrambling. The combination of cryptography and steganography provide high level security to the secret information. Cover image is known as carrier image and is the original image in which the secret data i.e., the payload is embedded. The unified image obtained after embedding the payload into the cover image is called the stego image. The recent boom in IT industry facilitates embedding data and security issues effectively.

Steganography is hiding private or secret data within a carrier in invisible manner. Steganography refers to the information that has been concealed inside a digital picture, video or audio file. This paper is based on JPEG quantization table modification. Firstly, the cover image is divided into 32*32 blocks and DCT is applied on each block. The number of payload LSB bits is embedded into DCT coefficients of the cover image based on the values of DCT coefficients. Secondly, IDCT is applied to produce the stego image which is identical to cover image. Then, the watermarked image is transmitted over the public channel. Our basis objective is to work on the color images with three planes and to work out on the Capacity, PSNR AND MSE values.

ACKNOWLEDGEMENT

During my M.E study at Thapar University, Patiala, Punjab, I have been fortunate to receive the valuable suggestions, guidance and support from my mentors, colleagues, family and friends.

First of all, I would like to express my most sincere gratitude to my supervisor **Ms. Ruchika Lamba**. She has been a wise and trusted guide throughout the entire process. Her guidance helped me to solve engineering problems and improve my communication skills. I am thankful to her that she has full confidence in my ability; much of this work would have not been completed without her vision and encouragement.

I am thankful to **Dr. Smarajit Ghosh**, Head of Department, Electrical and Instrumentation Engineering Department, Thapar University, Patiala for his encouragement and support. I am thankful to all the faculty members and staff members of department of Electrical and Instrumentation Engineering, Thapar University for their support during my academic years.

This section will look incomplete if I fail to thank almighty God and all my near and dear friends and my family members who stood beside me, understood my academic goals and helped me to achieve it. Last but not least by any means, my heartiest thanks to all the persons, who made me what I am today; word fails to express my feelings for them.

Tara Bansal

TARA BANSAL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

Steganography is concealed writing and is the scientific approach of inserting the secret data within a cover media such that the unauthorized viewers do not get an idea of any information hidden in it. Cover image is known as carrier image and is the original image in which the secret data i.e., the payload is embedded. The unified image obtained after embedding the payload into the cover image is called the stego image. The recent boom in IT industry facilitates embedding data and security issues effectively. Image Steganography includes several techniques of hiding the payload within the cover image. The most popular hiding techniques are Spatial Domain based Steganographic Techniques and Transform Domain based Steganographic Techniques.

Spatial domain based steganography includes the Least Significant Bit (LSB) technique, and Bit Plane Complexity Steganographic (BPCS) technique. Spatial domain based steganography includes the Least Significant Bit (LSB) technique [1], and Bit Plane Complexity Steganographic (BPCS) technique. Transform [2] is used for hiding the secret message into the higher frequency coefficient of the wavelet transform while leaving the lower frequency coefficient sub band unaltered. The various ways of using transform domain techniques in steganography are the following:

(i) The cover image is transformed into frequency domain and LSBs of transformed domain are replaced by the spatial domain payload bit stream.

(ii) The payload is converted into frequency domain and the coefficients are embedded into the spatial domain LSBs of cover image.

(iii) Both the cover and payload are transformed into frequency domain of embedding process.

Steganography has various useful applications. However, like any other science it can be used for ill intentions. It has been propelled to the forefront of current security techniques by the remarkable growth in computational power, the increase in security awareness by, e.g., individuals, groups, agencies, government and through intellectual pursuit. Steganography's ultimate objectives, which are undetectability, robustness (resistance to various image processing methods and compression) and capacity of the hidden data, are the main factors that separate it from related techniques such as watermarking and cryptography. It provides a state-of-the-art review and analysis of the different existing methods of steganography along with some common standards and guidelines drawn from the literature. It concludes some recommendations and advocate for the object-oriented embedding mechanism.

1.2 Motivation

The basic problem with the work which is being performed is that the previous work has been achieved on gray scale image. A gray scale image is basically a one plane image. The color images have been taken for any further processing and it is desired to achieve a better PSNR and MSE on these images. The basic methodology would also change to achieve a better PSNR. Quantization on 32*32 plan segmentation has been applied as per the requirement. The three plane segmentation is because of the three color formats of a color image that is red green and blue.

The proposed technique is based on Segmentation, Discrete Cosine Transform and Watermarking. The cover image is divided into 8*8 blocks and DCT is applied on each block. The number of payload MSB bits is embedded into DCT coefficients of the cover image based on the values of DCT coefficients. Finally the watermarked image is transmitted over the public channel.

1.3 Objective of the Thesis

The objective of the work carried in this dissertation is as follows:

- To study Steganography using 8x8 and 16x16 Quantization implemented on standard test images.
- To study and implement Steganography using 32x32 Quantization on colored test images.
- Comparison between 256x256 and 512x512 pixel images based on three parameters namely, MSE, PSNR and capacity.

1.4 Organization of the thesis

Chapter 1 gives the brief introduction and overview of the work. It highlights the objective that is desired to be achieved.

Chapter 2 gives an introduction to steganography which is sufficient for the purpose of engineering applications. It also presents the history, categories and techniques used for steganography.

Chapter 3 discusses Quantization in detail.

Chapter 4 discusses Steganography using 32x32 Quantization table.

Chapter 5 of the thesis discusses the MATLAB in Graphical User Interface (GUI).

Chapter 6 contains the results obtained by the steganography using 32x32 Quantization.

Chapter 7 discusses the concluding remarks of the thesis and also explores the future aspect of this work.

CHAPTER 2

STEGANOGRAPHY

2.1 Importance of Steganography

Steganography or Stego as it is often referred to in the IT community, literally means, "Covered writing" which is derived from the Greek language. Steganography is defined as 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 a 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 message present.

In a digital world, Steganography and Cryptography are both intended to protect information from unwanted parties. Both Steganography and Cryptography are excellent means by which to accomplish this but neither technology alone is perfect and both can be broken. It is for this reason that most experts would suggest using both to add multiple layers of security. Steganography can be used in a large amount of data formats in the digital world of today. The most popular data formats used are .bmp, .doc, .gif, .jpeg, .mp3, .txt and .wav mainly because of their popularity on the Internet and the ease of use of the steganographic tools that use these data formats. These formats are also popular because of the relative ease by which redundant or noisy data can be removed from them and replaced with a hidden message.

Steganographic technologies are a very important part of the future of Internet security and privacy on open system such as the Internet. Steganographic research is primarily driven by the lack of strength in the cryptographic systems by their own and the desire to have complete secrecy in an open-systems environment. Many governments have created laws that either limit the strength of crypto systems or prohibit them completely. This has been done primarily for fear by law enforcement not to be able to gain intelligence by wire taps, etc. This unfortunately leaves the majority of the Internet community either with relatively weak and a lot of the times breakable encryption algorithms or none at all. Civil liberties advocates fight this with the argument that

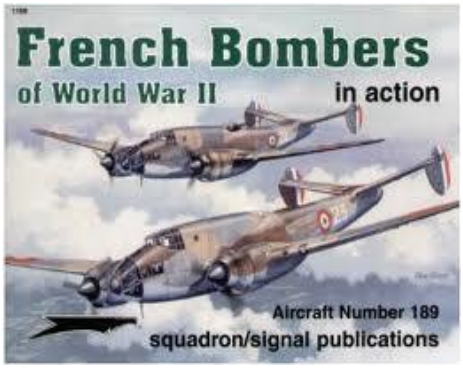
“these limitations are an assault on privacy”. This is where Steganography comes in picture. Steganography can be used to hide important data inside another file so that only the parties intended to get the message knows that a secret message exists. To add multiple layers of security and to help sub side the “crypto versus law”, it is a good practice to use Cryptography and Steganography together. Neither Cryptography nor Steganography are considered “turnkey solutions” to open systems privacy, but using both technologies together can provide a very acceptable amount of privacy for any one connecting to and communicating over these systems.

2.2 History of steganography

Steganography has been widely used, including the recent historical times and the present day. Possible permutations are endless and some known examples are cited.

Hidden messages within wax tablets in ancient Greece, people wrote messages on the wood, and then covered it with wax upon which an innocent covering message was written. Hidden messages on messengers body shown in Figure 2.2.(a) were also used in ancient Greece. Herodotus shown in Figure 2.2.(b) tells the story of a message tattooed on a slave's shaved head, hidden by the growth of his hair, and exposed by shaving his head again. The message allegedly carried a warning to Greece about Persian invasion plans. This method has obvious drawbacks such as delayed transmission while waiting for the slave's hair to grow, and its one-off use since additional messages requires additional slaves. In WWII, the French Resistance sent some messages written on the backs of couriers using invisible ink as shown in Figure 2.2.(c). Hidden messages on paper written in secret inks, under other messages or on the blank parts of other messages. Messages are written in Morse code on knitting yarn and then knitted into a piece of clothing worn by a courier. Messages are written on the back of postage stamps. During and after World War II, espionage agents used photographically produced microdots to send information back and forth. Microdots shown in Figure 2.2.(d) were typically minute, approximately less than the size of the period produced by a typewriter. WWII microdots needed to be embedded in the paper and covered with an adhesive (such as collodion). This was reflective and thus detectable by viewing against glancing light. Alternative techniques included inserting microdots into slits cut into the edge of post cards [3] .During World War II, a spy for Japan in New York City, Velvalee

(a)



(c)

(b)

(d)

Figure2.2 History of Steganography [3]

Dickinson, sent information to accommodation addresses in neutral South America. She was a dealer in dolls, and her letters discussed how many of this or that doll to ship. The stego text was the doll orders, while the concealed "plaintext" was itself encoded and gave information about ship movements, etc. Her case became somewhat famous and she became known as the Doll Woman. Cold War Counter-Propaganda In 1968, crew members of the USS Pueblo (AGER-2) intelligence ship held as prisoners by North Korea, communicated in sign language during staged photo opportunities, informing the United States they were not defectors but rather were being held captive by the North Koreans. In other photos presented to the US, crew members gave "the finger" to the unsuspecting North Koreans, in an attempt to discredit photos that showed them smiling and comfortable [3].

2.3 Mechanism of Steganography

The basic block diagram representation for steganography mechanism is shown in the Figure 2.3

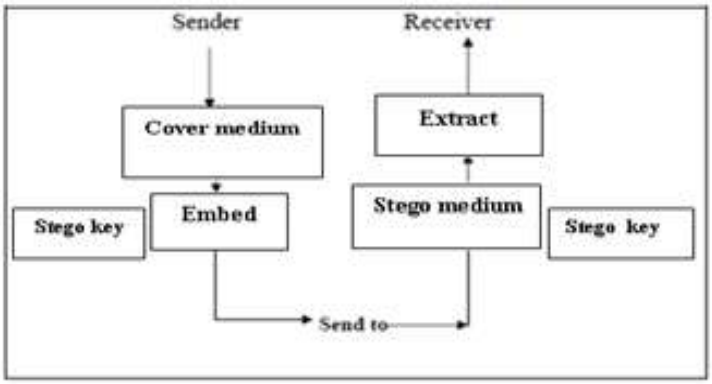


Figure2.3 Steganography mechanism [4]

The above figure shows a simple representation of the generic embedding and extraction process in steganography. In this example, a secret data is being embedded inside a cover image to produce the stego image. A key is often needed in the embedding process. The embedding procedure is done by the sender by using the proper stego key. The recipient can extract the stego cover image in order to view the secret data by using the same key used by the sender. The stego image should look almost identical to the cover image [4].

2.4 Categories of Steganography

Almost all digital file format scan be used for steganography, but the formats that are most suitable are those with a high degree of redundancy. Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display. The redundant bits of an object are those bits that can be altered without the alteration being detected easily. Image and audio files especially comply with this requirement, while research has also uncovered other file formats that can be used for information hiding Figure 2.4 shows the categories of file formats that can be used for steganography

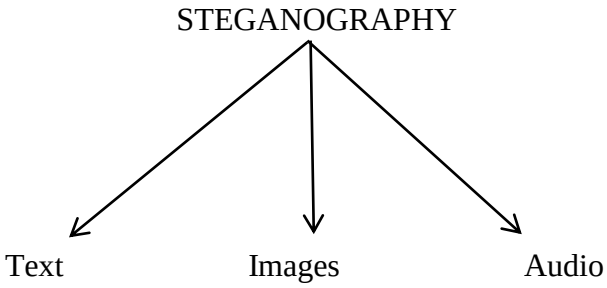


Figure 2.4 Categories of Steganography [5]

2.4.1 Text Steganography

Text steganography can be achieved by altering the text formatting, or by altering certain characteristics of textual elements (e.g., characters). The goal in the design of coding methods is to develop alterations that are reliably decodable (even in the presence of noise) yet largely indiscernible to the reader. These criteria, reliable decoding and minimum visible change, are somewhat conflicting; herein lies the challenge in designing document marking techniques. The document format file is a computer file describing the document content and page layout (or formatting), using standard format description languages such as PostScript2, TeX, @off, etc. It is from this format file that the image - what the reader sees - is generated. The three coding techniques that Nosrati et al., [5] propose illustrate different approaches rather than form <an exhaustive list of document marking techniques. The techniques can be used either separately or jointly. Each technique enjoys certain advantages or applicability as discussed in the following section.

2.4.1.1 Line-Shift Coding

This is a method of altering a document by vertically shifting the locations of text lines to encode the document uniquely. This encoding may be applied either to the format file or to the bitmap of a page image. The embedded code word may be extracted from the format file or bitmap. In certain cases this decoding can be accomplished without need of the original image, since the original is known to have uniform line spacing between adjacent lines within a paragraph [5].

2.4.1.2 Word-Shift Coding

This is a method of altering a document by horizontally shifting the locations of words within text lines to encode the document uniquely. This encoding can be applied to either the format file or to

the bitmap of a page image. Decoding may be performed from the format file or bitmap. The method is applicable only to documents with variable spacing between adjacent words. Variable spacing in text documents is commonly used to distribute white space when justifying text. Because of this variable spacing, decoding requires the original image or more specifically, the spacing between words in the un-encoded document [5].

2.4.1.3 Feature Coding

This is a coding method that is applied either to a format file or to a bitmap image of a document. The image is examined for chosen text features, and those features are altered, or not altered, depending on the code word. Decoding requires the original image, or more specifically, a specification of the change in pixels at a feature. There are many possible choices of text features; here, Nosrati et al., [5] choose to alter upward, vertical end lines - that is the tops of letters, b, d, h, etc. These end lines are altered by extending or shortening their lengths by one (or more) pixels, but otherwise not changing the end line feature. There is another form of text steganography which is defined by Chapman et al. as the text steganography is a method of using written natural language to conceal a secret message.

2.4.2 Image Steganography

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

The most widely used technique today is hiding of secret messages into a digital image. This steganography technique exploits the weakness of the human visual system (HVS). HVS cannot detect the variation in Luminance of color vectors at higher frequency side of the visual spectrum. A Picture can be represented by a collection of color pixels. The individual pixels can be represented by their optical characteristics like 'bright ness', 'Chroma' etc. Each of these characteristics can be digitally expressed in terms of 1s and 0s [6].

For example: A 24-bit bitmap will have 8 bits, representing each of the three-color values (red, green, and blue) at each pixel. If we consider just the blue there will be 2^8 values of blue. The difference between 11111111 and 11111110 in the value for blue intensity is likely to be undetectable by the human eye. Hence, if the terminal recipient of the data is nothing but human

visual system (HVS) Then the Least Significant Bit (LSB) can be used for something else other than color information.

This technique can be directly applied on digital image in bitmap format as well as for the compressed image format like JPEG. In JPEG format, each pixel of image is digitally coded using discrete cosine transform (DCT). The LSB of the encoded DCT components can be used as the carriers of hidden message [6].

2.4.3 Audio Steganography

In audio steganography, secret message is embedded into digitized audio signal which result slight altering of binary sequence of the corresponding audio file. There are several methods are available for audio steganography.

2.4.3.1 LSB Coding

Sampling technique followed by Quantization converts analog audio signal to digital binary sequence. In this technique LSB of binary sequence of each sample of digitized audio file is replaced with binary equivalent of secret message [6].

2.4.3.2 Phase Coding

Human Auditory System (HAS) can't recognize the phase change in audio signal as easy it can recognize noise in the signal. The phase coding method exploits this fact. This technique encodes the secret message bits as phase shifts in the phase spectrum of a digital signal, achieving an inaudible encoding in terms of signal-to- noise ratio [5].

2.4.3.3 Spread Spectrum

There are two approaches are used in this technique: the direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS). Direct-sequence spread spectrum (DSSS) is a modulation technique used in telecommunication. As with other spread spectrum technologies, the transmitted signal takes up more bandwidth than the information signal that is being modulated. Direct-sequence spread-spectrum transmissions multiply the data being transmitted by a "noise" signal. This noise signal is a pseudorandom sequence of 1 and −1 values, at a frequency much higher than that of the original signal, thereby spreading the energy of the original signal into a

much wider band. The resulting signal resembles white noise. However, this noise-like signal can be used to exactly reconstruct the original data at the receiving end, by multiplying it by the same pseudorandom sequence (because $1 \times 1 = 1$, and $-1 \times -1 = 1$). This process, known as "de-spreading", mathematically constitutes a correlation of the transmitted Pseudorandom Noise (PN) sequence with the receiver's assumed sequence. For de-spreading to work correctly, transmit and receive sequences must be synchronized. This requires the receiver to synchronize its sequence with the transmitter's sequence via some sort of timing search process.

In contrast, frequency-hopping spread spectrum pseudo-randomly retunes the carrier, instead of adding pseudo-random noise to the data, which results in a uniform frequency distribution whose width is determined by the output range of the pseudo-random number generator [6].

2.4.3.4 Echo Hiding

In this method the secret message is embedded into cover audio signal as an echo. Three parameters of the echo of the cover signal namely amplitude, decay rate and offset from original signal are varied to represent encoded secret binary message. They are set below the threshold of Human Auditory System (HAS) so that echo can't be easily resolved.

Video files generally consist of images and sounds, so most of the relevant techniques for hiding data into images and audio are also applicable to video media. In the case of Video steganography, sender sends the secret message to the recipient using a video sequence as cover media. Optional secret key 'K' can also be used during embedding the secret message to the cover media to produce 'stego-video'. After that, the stego-video is communicated over public channel to the receiver. At the receiving end, receiver uses the secret key along with the extracting algorithm to extract the secret message from the stego-object [5].

2.5 Techniques of Steganography

Following are different types of steganographic techniques that are available

1. Pure steganography

2. Public key steganography

3. Secret key steganography

2.5.1 Pure Steganography

Pure Steganography is a Steganography system that doesn't require prior exchange of some secret information before sending message; therefore, no information is required to start the communication process: The security of the system thus depends entirely on its secrecy.

The pure Steganography can be defined as the quadruple (C, M, D, and E) where:

C: the set of possible covers.

M: the set of secret massage with $|C| \geq |M|$.

E: $C \times M \rightarrow C$ the embedding function

D: $C \rightarrow M$ of the extraction function with the property that

D (E(c, m)) = m for all m Є M and c Є C

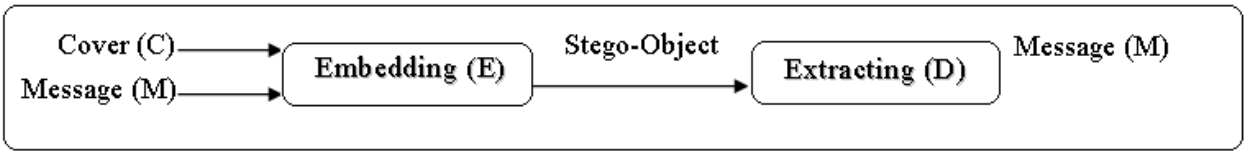


Figure 2.5.1 Pure Steganography Process [7]

2.5.2 Secret Key Steganography

A secret key Steganography system is similar to a symmetric cipher, where the sender chooses a cover and embeds the secret message into the cover using a secret key. If the secret key used in the embedding process is known to the receiver, he can reverse the process and extract the secret message [8].

Anyone who doesn't know the secret key should not be able to obtain evidence of the encoded information.

The secret key Steganography can be defined as the quintuple (C, M, K, DK, and EK) where:

C: the set of possible covers.

M: the set of secret message.

K: the set of secret keys.

EK: $C \times M \times K \rightarrow C$

With the property that DK (EK(c, m, k), k) = m for all m Є

M, c Є C and k Є K

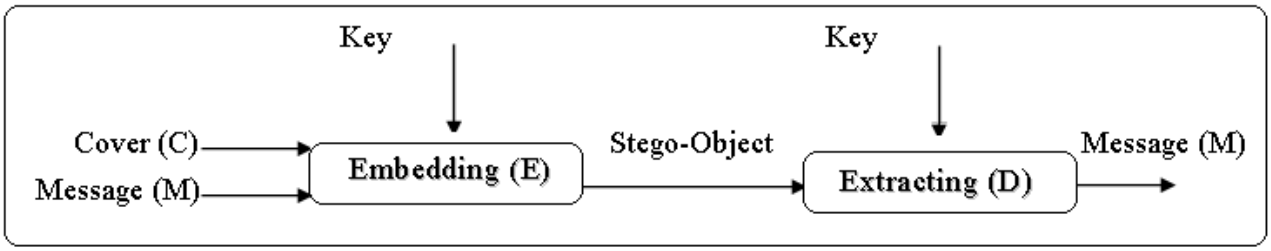


Figure2.5.2 Secret Key Steganography [7]

2.5.3 Public Key Steganography

Public key Steganography does not depend on the exchange of a secret key. It requires two keys, one of them private (secret), and the other public: The public key is stored in a public database, whereas the public key is used in the embedding process. The secret key is used to reconstruct the secret message [8]

One way to build a public key Steganography system is to use a public key crypto system. The sender and the receiver can exchange public keys of some public key cryptography algorithm before imprisonment. Public key Steganography utilizes the fact that the decoding function in a Steganography system can be applied to any cover, whether or not it already contains a secret message [8].

The public key Steganography relies on the fact that encrypted information is random enough to hide in plain sight. The sender encrypts the information with the receiver's public key to obtain a random-looking message and embeds it in a channel known to the receiver, thereby replacing some of the natural randomness with which every communication process is accompanied. Assume that both the cryptographic algorithms and the embedding functions are publicly known [7].

The receiver who cannot decide a priori if secret information is transmitted in a specific cover will suspect the arrival of message and will simply try to extract and decrypt it using his private key. If the cover actually contained information, the decryption information is the sender's message [9].

2.6 Literature Review

Luo et al., [10] proposed LSB matching revisited image steganography and edge adaptive scheme which can select the embedding regions according to the size of secret message and the difference between two consecutive pixels in the cover image. For lower embedding rates, only sharper edge regions are used while keeping the other smoother regions as they are. When the embedding rate

increases, more edge regions are released adaptively for data hiding by adjusting just a few parameters.

Mathkour et al., [11] compared the strengths and weaknesses of the existing techniques of steganography and implemented a new steganographic wizard based tool, which have been examined against several other tools like F5, S-Tools etc., for more robust and secure steganography.

Hong et al., [12] proposed a lossless steganography technique wherein the secret information is hidden inside the compressed Absolute Moment Block Truncation Coding image. Naji et al., [13] analyzed different steganographic techniques and weaknesses in the respective techniques.

Banoci et al., [14] presented Code Division Multiple Access Technique, where the embedding process is carried out by hiding secret image in each block of quantized DCT coefficients.

Mankun Xu et al., [15] proposed a technique to estimate the embedding rates of secret information in the Model Based steganography based on least square method.

Aos et al., [16] implemented a means of hiding the secret information in the Executable (.EXE) file, such that it is unrevealed to any anti-virus software, since anti-virus software secretly read the furtive data embedded inside the cover file.

Chen et al., [17] introduced an extension to the existing steganography to improve the capacity. At the receiver, template matching techniques are used to find the location of the object file.

Chang et al., [18] proposed a transform domain based adaptive data hiding method using Haar discrete wavelet transform. Most of the data was hidden in the edge region as it is insensitive to the human eye.

Cheiew et al., [19] proposed a scheme to estimate the length of hidden message through histogram quotient in Binary image embedded by using Boundary pixels Steganography technique.

Ramezani et al., [20] presented an Adoptive Steganography method with respect to image contrast thereby improving the embedding capacity of stego image contrast by selecting valid blocks for embedding based on average difference between the gray level values of the pixels in 2*2 blocks of non-overlapping spatially and their mean gray level.

Saed Sarreshtedari and Shahrokh Ghaemmaghami [21] proposed a high capacity image steganography in Discrete Wavelet Transform (DWT) domain.

HongmeiTang et al., [22] suggested a scheme for image encryption and steganography by encrypting the message with a combination of gray value substitution operation and position permutation and then it is hidden in the cover image.

CHAPTER 3

QUANTIZATION

3.1 Introduction

Quantization is the procedure of constraining something from a relatively large or continuous set of values (such as the real numbers) to a relatively small discrete set (such as the integers). The most powerful and quantization technique used for the image compression is vector quantization (VQ).The vector quantization algorithms for reducing the transmission bit rate or storage have been extensively investigated for speech and image signals. Image vector quantization (VQ) includes four stages: vector formation, Training set selection, codebook generation and quantization. The first step is to divide the input image into set of vectors .The Subset of vectors in the set is later chosen as a training sequence. The codebook of code words is obtained by an iterative clustering algorithm. Finally, in quantizing an input vector, closest code word in the codebook is determined and corresponding label of this code word is transmitted. In this process, data compression is achieved because address transmission requires fewer bits than transmitting vector itself. The concept of data quantization is extended from scalar to vector data of arbitrary dimension. Instead of output levels, vector quantization employs a set of representation vectors (for one dimensional case) or matrices (for two dimensional cases). Set is defined as "codebook" and entries as "code words". Vector quantization has been found to be an efficient coding technique due to its inherent ability to exploit the high correlation between the neighboring pixels [23].

Vector Quantization (VQ) is one of the techniques based on the principle of block coding that have long been used to compress media in order to make efficient use of network bandwidth and data storage space. The code words of the codebook are used to substitute the closest pixel block of the image during the compressing. The resultant compressed image is a table of indexes of the code words. Some steganographic methods for VQ compressed images have been reported in the recent years .A common feature of the methods is that they all partition the codebook into a number of groups or clusters and then embed the secret message by replacing the code word indexes of the compressed image with those of the same group / cluster selected according to the corresponding secret bits For example with a cluster of 8 (= 23) code words, each code word can embed 3 bits of the secret message. If the sequence of the secret bits isM102, (or 112), the second (or third) code

word is used to replace the original code word. The receiving end of the stego-image needs to have the same clustering of the same codebook. The secret message is extracted by concatenating the position (in binary form) of the received code words in their groups / clusters. Therefore, we can see that the greater the cluster, the greater the embedding capacity. However, the greater a cluster is, the greater the variance among the code words in the group becomes. It means the average embedding distortion can be greater because the possibility that a code word get replaced with a more distant code word is higher. So striking a balance between embedding capacity and distortion is important but, unfortunately, not trivial. The feasibility resides in the optimality of the codebook clustering algorithm [24].

3.2 Discrete Cosine Transform

The discrete cosine transform (DCT) helps to separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT is similar to the discrete Fourier transform: it transforms a signal or image from the spatial domain to the frequency domain as shown in the Figure 3.2

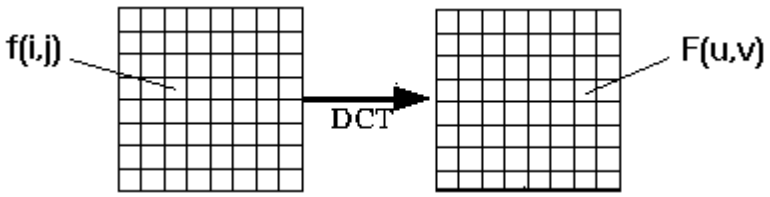


Figure 3.2 Transformation of function into DCT [25]

A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical in these applications; for compression, it turns out that cosine functions are much more efficient (as described below, fewer are needed to approximate a typical signal), whereas for differential equations the cosines express a particular choice of boundary conditions [25].

In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT"; its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sine transforms (DST), which is equivalent to a DFT of real and odd functions, and the modified discrete cosine transforms (MDCT), which is based on a DCT of overlapping data [25].

3.3 Bit Length Replacement Steganography Based on DCT Coefficients (BLSDCT)

The cover image is segmented into 8*8 blocks and DCT is applied on each block. The numbers of payload MSB bits are embedded into DCT coefficients of the cover image based on the values of DCT coefficients. The following are the techniques used in this method:

3.3.1 Embedding Technique

The payload is embedded into the cover image by segmentation, DCT and coherent bit length which are shown in the Figure 3.3.1. The cover image is color or gray scale of any size and format. If the cover image is color then convert into gray scale image and corresponding pixel intensity values. The gray scale cover image pixel intensity varies from zero to 255. During the payload embedding process the intensity values of cover image may exceed lower and higher level limits which results in difficulty to retrieve the payload at the destination. Hence the cover image pixel intensity values are limited to lower15 and upper 240instead of zero and 255. The cover image is segmented into 8x8 matrices. The DCT is applied on each 8x8 block to get DCT coefficients which are used to hide the payload Most Significant Bit (MSB) based on the DCT coefficient values of the cover image. Transform each 8x8 matrix into frequency domain using 2D-DCT. Using DCT on 8*8 sub blocks has an advantage of less computation time for embedding as well as security to payload increases compared to applying DCT to whole cover image.

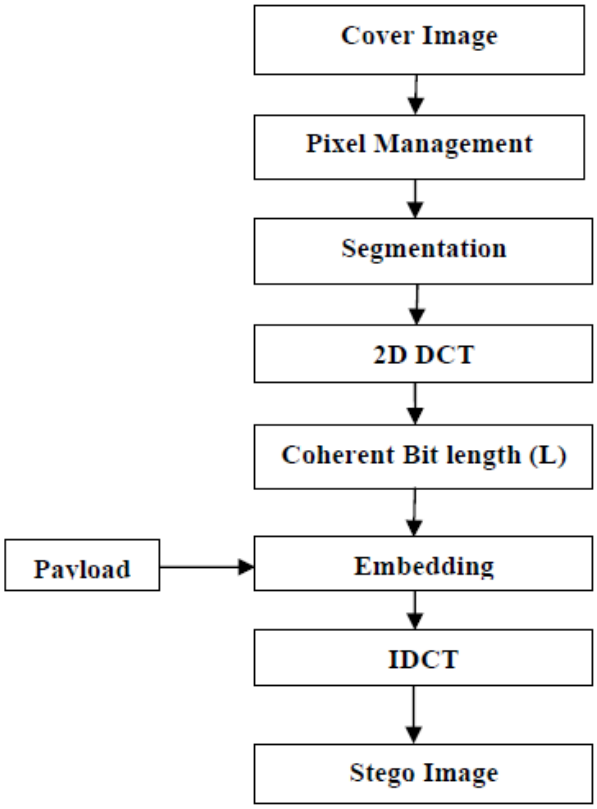


Figure 3.3.1 Block Diagram of BSLDCT Embedding Technique [26]

The length L, which determines the number of LSBs of each DCT coefficients (C0) of cover image that can be used to hide the payload MSB bits and is calculated according to the conditions given below:

If $C0 \geq 2^5$; L=5

If $2^4 \leq C0 \leq 2^5$; L=4

If $2^3 \leq C0 \leq 2^4$; L=3

Else L=2

The conditions to determine L also serves as secret key to retrieve the payload at the destination.

The spatial domain payload bits are embedded into the DCT coefficients of the cover image coherently based on the values coefficients. Four MSBs of each spatial domain payload pixel are embedded into the cover image DCT coefficients in a continuous manner depends on the value of L to derive the stego image in DCT domain. The stego image in the transform domain is converted to

the spatial domain by applying IDCT. The stego image obtained is similar to the cover image and the difference is not perceptible by the human eye. This image is transmitted to the destination over the open channel [26].

3.3.2 Retrieval Technique

The payload is retrieved from the stego image by adapting reverse process of embedding technique is as shown in Figure 3.3.2. The image received at the destination over the open channel is the stego image. Any intruder interfering in the transmission process will only be able to read the stego image and cannot extract the secret image embedded in it as the payload bits are embedded coherently. The stego image is segmented into 8x8 blocks to ensure proper retrieval of payload.

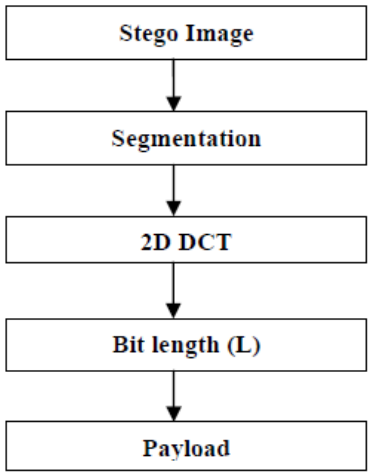


Figure 3.3.2 Block Diagram of BSLDCT Retrieval Technique [26]

The 8x8 sub blocks of stego image are transformed into frequency domain to generate DCT coefficients using 2D DCT. At the receiver L is determined based on the DCT coefficient values similar to the conditions of embedding technique. A spar matrix is initially taken as payload output matrix. The payload bits are extracted into this output matrix based on the value of L to get back the payload.

3.4 Quantization Tables

The JPEG standard uses 8x8 quantization tables, but it does not specify default or standard values for quantization tables. Specifying the quantization values is left up to the application. However, the JPEG standard provides a pair of quantization tables (luminance and chrominance) as examples tested empirically and found to generate good results as shown in Table I for luminance.

Table I. The standard (8×8 blocks) luminance quantization table in JPEG [27]

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Dividing this quantization Table I by 2, we get a new quantization table as shown in Table II.

Table II. The scaled quantization table (scale factor = 2) [27]

8	6	5	8	12	20	26	31
6	6	7	10	13	29	30	28
7	7	8	12	20	29	35	28
7	9	11	15	26	44	40	31
9	11	19	28	34	55	52	39
12	18	28	32	41	52	57	46
25	32	39	44	52	61	60	51
36	46	48	49	56	50	52	50

Using this new quantization table generates reconstructed images almost identical to the source image. Therefore, this table is used with Jpeg-JSteg [28] method. Since the values of these tables could be an arbitrary choice, some researchers modified these quantization tables for their research purposes. For example Table III, the modified version of Table II, has been used within Chang et al. method [28]. 8x8 quantization tables apart, there are no samples for larger quantization tables in the JPEG standard.

Table III. The modified quantization table [27]

8	6	5	8	1	1	1	1
6	6	7	1	1	1	1	28
7	7	1	1	1	1	35	28
7	1	1	1	1	44	40	31
1	1	1	1	34	55	52	39
1	1	1	32	41	52	57	46
1	1	39	44	52	61	60	51
1	46	48	49	56	50	52	50

Miano [29] states that "If you are implementing a JPEG encoder you can come up with your own scaling or use any other method you want for generating quantization values". Therefore, a quantization table can arbitrarily be generated. Consequently, 16x16 quantization table as shown in Table IV has been produced by simulating and stretching the scaled quantization table as shown in Table II. For embedding purposes, the middle frequencies of the produced quantization table were set to be 1 as shown in Table V.

3.5 Embedding and extracting procedure of JPEG Based steganography

The procedure of embedding a secret message in a cover image for JPEG-based steganography is illustrated in the left side of Figure 3.5. It can be described as follows:

- The message (M) to be embedded in the cover image is randomly generated.
- The cover image is divided into non-overlapping blocks of 16x16 pixels and then the DCT is used to transform each block into DCT coefficients.
- The DCT coefficients are scaled by the modified 16x16 quantization table as shown in Table IV. In this quantization table, the values of (1) represent the middle frequencies to be used for embedding (121 values). The quantized
- DCT coefficients of each block are rounded to the nearest integers and then set in zigzag scan order.
- The least two-significant bits of each middle frequency coefficient in the quantized DCT blocks are modified to embed two secret bits.
- The JPEG entropy coding (DPCM, Run-Length coding, and Huffman coding) is applied to compress these resultant blocks, and then the JPEG file is obtained.

The procedure of extracting the embedded message from the JPEG file is illustrated in the right side of Figure 3.5. In the extracting procedure, the JPEG file (stego image) is entropy decoded using the coding tables (Huffman tables) located in the image header. As a result we get the blocks of quantized DCT coefficients modified according to the secret message. From each pre-defined middle frequency coefficient of each block we retrieve the least two-significant bits (secret bits). We put these retrieved bits in the same order of embedding to get the secret message (M).

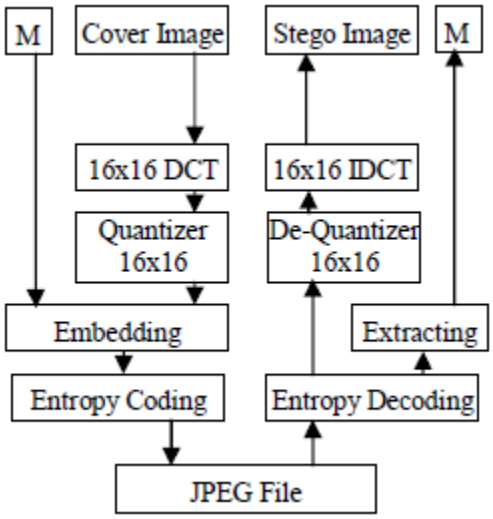


Figure 3.5 Block diagram of embedding (left) and extracting (right) procedures [27]

A novel steganographic method based upon JPEG compression and DCT transformation has been performed. Using gray-level cover images, transformation of DCT of non-overlapping blocks of 16x16 pixels has been done instead of non-overlapping blocks of 8x8 pixels. The transformed DCT coefficients were quantized by a modified 16x16 quantization table. Then, the secret data has been embedded within the middle frequency coefficients. The DCT transformations and the suggested 16x16 quantization table with Chang et al. embedding technique has been used for evaluation. Afterwards, comparing this method with Jpeg-JSteg method [28] and Chang et al. method [28] to show the improvements provided by this method

Table IV Suggested 16x16 quantization table [27]

16	8	7	6	6	1	1	1	1	1	1	1	1	1	1	1
7	7	6	6	1	1	1	1	1	1	1	1	1	1	1	30
7	6	6	1	1	1	1	1	1	1	1	1	1	1	30	28
6	8	1	1	1	1	1	1	1	1	1	1	1	32	35	29
8	1	1	1	1	1	1	1	1	1	1	1	32	35	32	28
1	1	1	1	1	1	1	1	1	1	1	35	40	42	40	35
1	1	1	1	1	1	1	1	1	1	35	44	42	40	35	31
1	1	1	1	1	1	1	1	1	35	44	44	50	53	52	45
1	1	1	1	1	1	1	1	31	34	44	55	53	52	45	39
1	1	1	1	1	1	1	31	34	40	41	47	52	45	52	50
1	1	1	1	1	1	30	32	36	41	47	52	54	57	50	46
1	1	1	1	1	36	32	36	44	47	52	57	60	60	55	50
1	1	1	1	36	39	42	44	48	52	57	61	60	60	55	51
1	1	1	39	42	47	48	46	49	57	56	55	52	51	54	51
1	1	41	46	47	48	48	49	53	56	53	50	51	52	51	50
1	43	47	47	48	48	49	57	57	56	50	52	52	51	50	50

3.6 Implementation of Modified 16×16 Quantization Table Steganography on Colour Images

The technique presents a novel steganographic method based on the JPEG quantization table modification. Instead of dividing cover image into 8×8 blocks, the cover image is divided into non-overlapping blocks of 16×16 pixels to embed secret information. Here colour images have been considered and the feasibility of data hiding has been investigated. Three performance parameters namely Capacity, MSE and PSNR have been compared on different sizes of standard test images. In comparison with Jpeg-JSteg and Chang et al. methods [28] based on the conventional blocks of 8x8 pixels the proposed method shows high performance with regard to embedding rate and PSNR of stego image. Furthermore, the produced stego-images are almost identical to the original cover images.

3.6.1 Algorithms

The embedding procedure contains five phases. The following are the five phases

- Colour image Pre-processing
- Message Encryption
- Message embedment
- Entropy encoding
- JPEG colour stego Image

The cover image and the secret message are given. Following are objectives of the embedding procedure:

(i) Embed the secret message into the cover image to derive the stego image for security.

(ii) Improve PSNR between cover image and stego image.

(iii)Enhance the stego capacity.

The following are the algorithms used in this method:

3.6.1.1 Embedding algorithm

In the embedding algorithm secret data is embedded into the cover image using segmentation into 16×16 non-overlapping blocks. The payload i-e secret data is embedded into the quantized DCT coefficients after quantization.

Table V 16×16 Modified Quantization Table [30]

8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	30
1	1	1	1	1	1	1	1	1	1	1	1	1	1	30	28
1	1	1	1	1	1	1	1	1	1	1	1	1	32	35	29
1	1	1	1	1	1	1	1	1	1	1	1	32	35	32	28
1	1	1	1	1	1	1	1	1	1	1	35	40	42	40	35
1	1	1	1	1	1	1	1	1	1	35	44	42	40	35	31
1	1	1	1	1	1	1	1	1	35	44	44	50	53	52	45
1	1	1	1	1	1	1	1	31	34	44	55	53	52	45	39
1	1	1	1	1	1	1	31	34	40	41	47	52	45	52	50
1	1	1	1	1	1	30	32	36	41	47	52	54	57	50	46
1	1	1	1	1	36	32	36	44	47	52	57	60	57	55	47
1	1	1	1	36	39	42	44	48	52	57	61	60	60	55	51
1	1	1	39	42	47	48	46	49	57	56	55	52	61	54	51
1	1	42	46	47	48	48	49	53	56	53	50	51	52	51	50
1	45	46	47	48	48	49	57	56	56	50	52	52	51	51	50

1. A cover image ‘C’ of any size like 256×256 is considered and any message such as character or strings is randomly generated for testing hiding algorithm.

2. Secret message is encrypted as data to be hidden it is in ASCII format which is converted to binary format.

3. Segmentation of cover image into blocks {C_1; C_2; C_3; . . . ; C_N/16× N/16}. Each C_i contains 16x16 pixels that are further transformed into DCT coefficients in transform domain.

4. DCT transforms each block C_i into DCT coefficient matrix Xi, where Xi = [a; b] = DCT (C_i [a; b]),

Where $1 \leq a; b \leq 16$ and $C_i = [a; b]$ is the pixel value in C_i.

5. Application of new 16×16 modified quantization table 'T' that generates 136 Quantized AC coefficients.

6. Two secret bits with LSB method are embedded into least two significant bits of AC coefficients which correspond to the value 1 in quantization table.

7. Entropy coding is applied on colour JPEG individual blocks of R, G and B which generates the required Compressed JPEG image file (Stego Image in Compressed Form).

The procedure is shown in the Figure 3.6.1.1

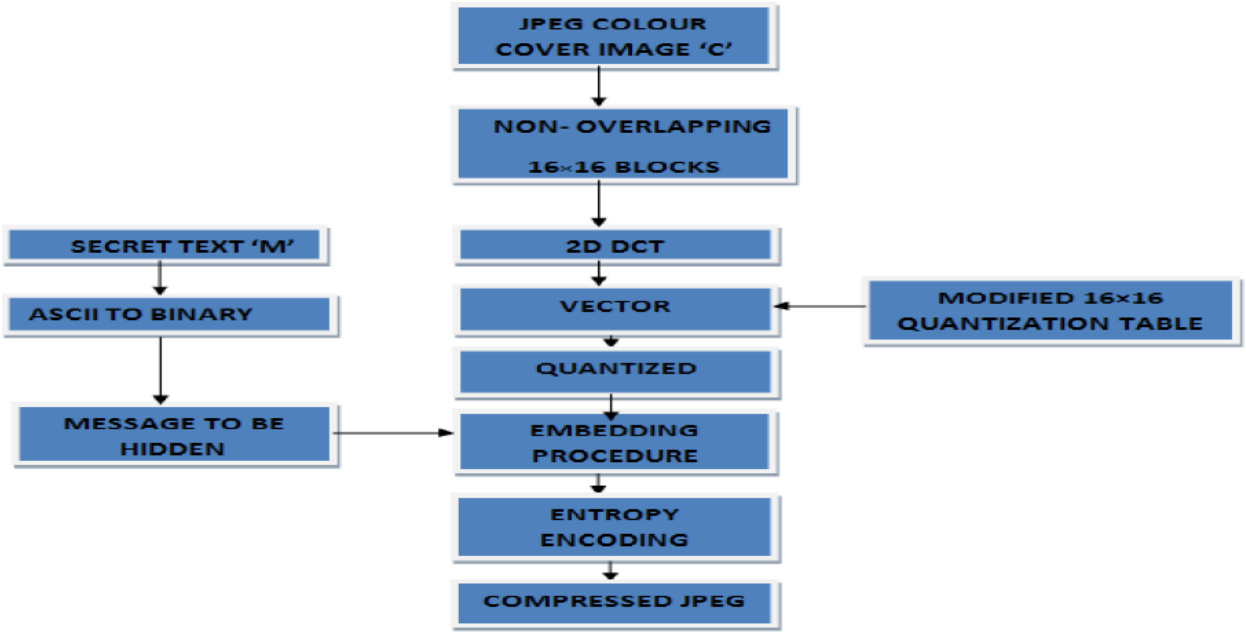


Figure 3.6.1.1 Embedding Procedure [30]

3.6.1.2 Extracting Procedure

The secret message is retrieved for the stego image by the adaptive reverse procedure of embedding.

1. Entropy decoding is done on the received JPEG image file.

2. Decoded block is followed by extraction of the secret message from least significant bits of 136 low and mid frequency coefficients .The message is decrypted to original ASCII format.

3. De-quantization using 16×16 quantization table is achieved.

4. Dequantized JPEG image is converted to spatial domain by implementing IDCT (Inverse Discrete cosine transform) segmented into 16×16 blocks.

5. Colored Stego Image obtained.

6. Secret Message 'M*'obtained.

M=Secret Text and M*=Extracted Secret Text

The procedure is shown in the Figure 3.6.1.2

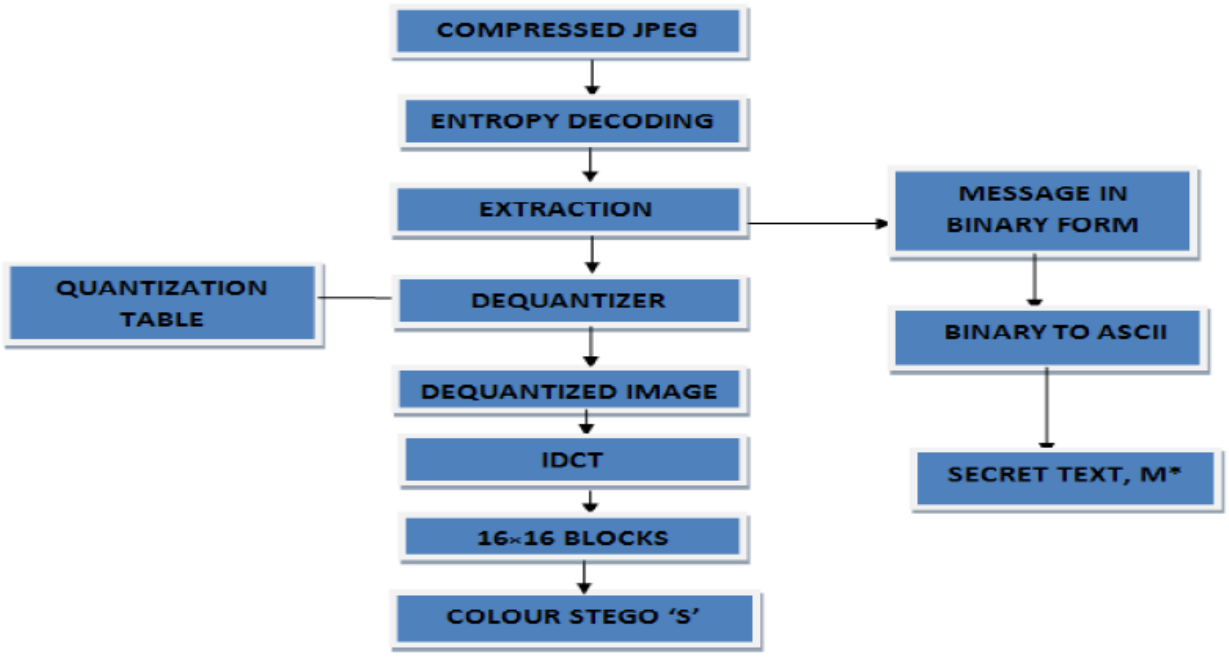


Figure 3.6.1.2 Extracting Procedure [30]

3.6.2 Comparison of Evaluation Parameters

Images of 256x256 and 512x512 pixel are taken and steganography using 16x16 quantization tables is applied on these different test images. Three evaluation parameters namely Hiding Capacity, PSNR, MSE are calculated and then these three parameters have been compared as shown in the Table VI. Table VI indicates that 512 × 512 pixel image has more PSNR and less MSE as compared to 256×256 pixel images. PSNR comparison is also shown between 256x256 and 512x512 pixel as shown in Figure 3.6.2(a). Table VII and Figure 3.6.2(b) shows comparison between steganography capacities on different sized test images. It has been found that capacity which is the amount of information embedding in colour images increases as the number of

modified quantized DCT coefficients increases. So, more data can be embedded using of 16×16 Quantization Tables as compared to 8×8 tables.

Table VI Comparison of Hiding Capacity, MSE AND PSNR in Steganography using 16x16 Quantization Table [30]

Image	Pixels	Hiding Capacity (bits)	MSE	PSNR
1.Lena	256X256	69632	0.0981	58.2122
	512X512	278528	0.0251	64.1282
2.Pepper	256X256	69632	0.0931	56.9715
	512X512	278528	0.0250	62.2132
3.Jet	256X256	69632	0.1771	55.6473
	512X512	278528	0.0324	63.0185
4.Baboon	256X256	69632	0.0944	58.3766
	512X512	278528	0.0240	64.3199

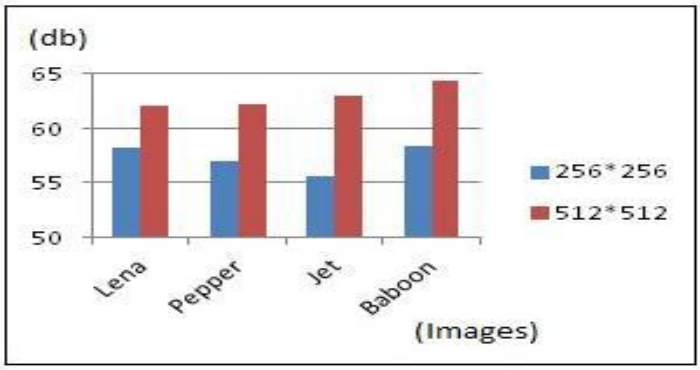


Figure 3.6.2(a) PSNR Comparison [30]

Table VII shows that our method has better capacity of embedding message bits in image than JSteg and Chang's. Since the DCT coefficients after the quantization are almost all zeros, the message capacity of Jpeg–JSteg is very much limited. A block can embed 136 × (417 × 417)/ (8 ×8) = 184757 secret bits into a cover image of 417 × 417 pixels.

Table VII Comparison of Steganography Capacity [30]

Method	Lena	Baboon
Proposed	184757 bits	184757 bits
Chang Method	141284 bits	141284 bits
Jpeg-Jsteg	49798 bits	53142 bits

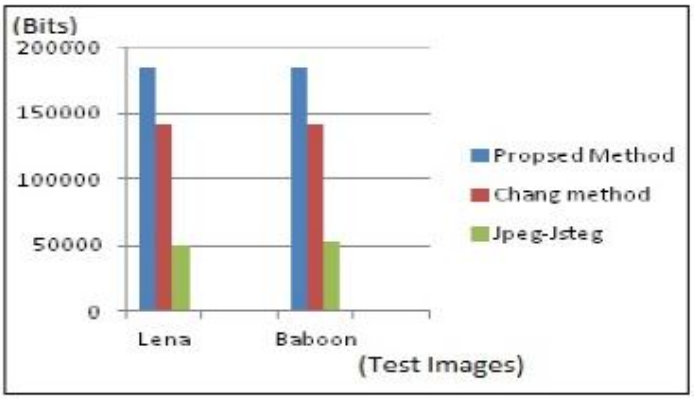


Figure 3.6.2(b) Capacity Comparison [30]

CHAPTER 4

STEGANOGRAPHY USING 32X32 QUANTIZATION TABLES

A lot of work has been already done in the case of steganography. Defining the bits is a very important step in case of pixeling. Different kind of pixeling has already been done. The previous pixeling work is done on gray scale images using 8x8 quantization tables [26] 16x16 quantization tables [30]. In this thesis we are working on colour images using 32x32 quantization tables.

4.1 Related Work

The proposed technique is based on Segmentation, Discrete Cosine Transform and Watermarking. The cover image is divided into 8*8 blocks and DCT is applied on each block. The number of payload MSB bits is embedded into DCT coefficients of the cover image based on the values of DCT coefficients. Finally the watermarked image is transmitted over the public channel.

- To segment the cover image into 8*8 blocks
- To apply DCT on each block to get the DCT coefficients
- To find the bit length to hide the data
- To embed the data in DCT coefficients according to the bit length
- To apply IDCT in order to get the stego image in spatial domain
- To add the watermark in the obtained stego image, to protect it from intruder
- To extract watermark and the stego image
- To evaluate the image using parameters like MSE, PSNR and capacity.

4.2 Proposed Work

4.2.1 Algorithms

The model uses an adaptive data hiding technique, where the number of payload bits *L* is embedded into the DCT coefficient of cover image based on the DCT coefficient of cover image in order to maximize the hiding capacity. The payload is embedded into the cover image by segmentation, DCT and adaptive bit length L.

4.2.1.1 Embedding Algorithm

The cover image is segmented into 8x8 matrices. The DCT is applied on each 8x8 block to get DCT coefficients which are used to hide the payload Most Significant Bit (MSB) based on the DCT coefficient values of the cover image.

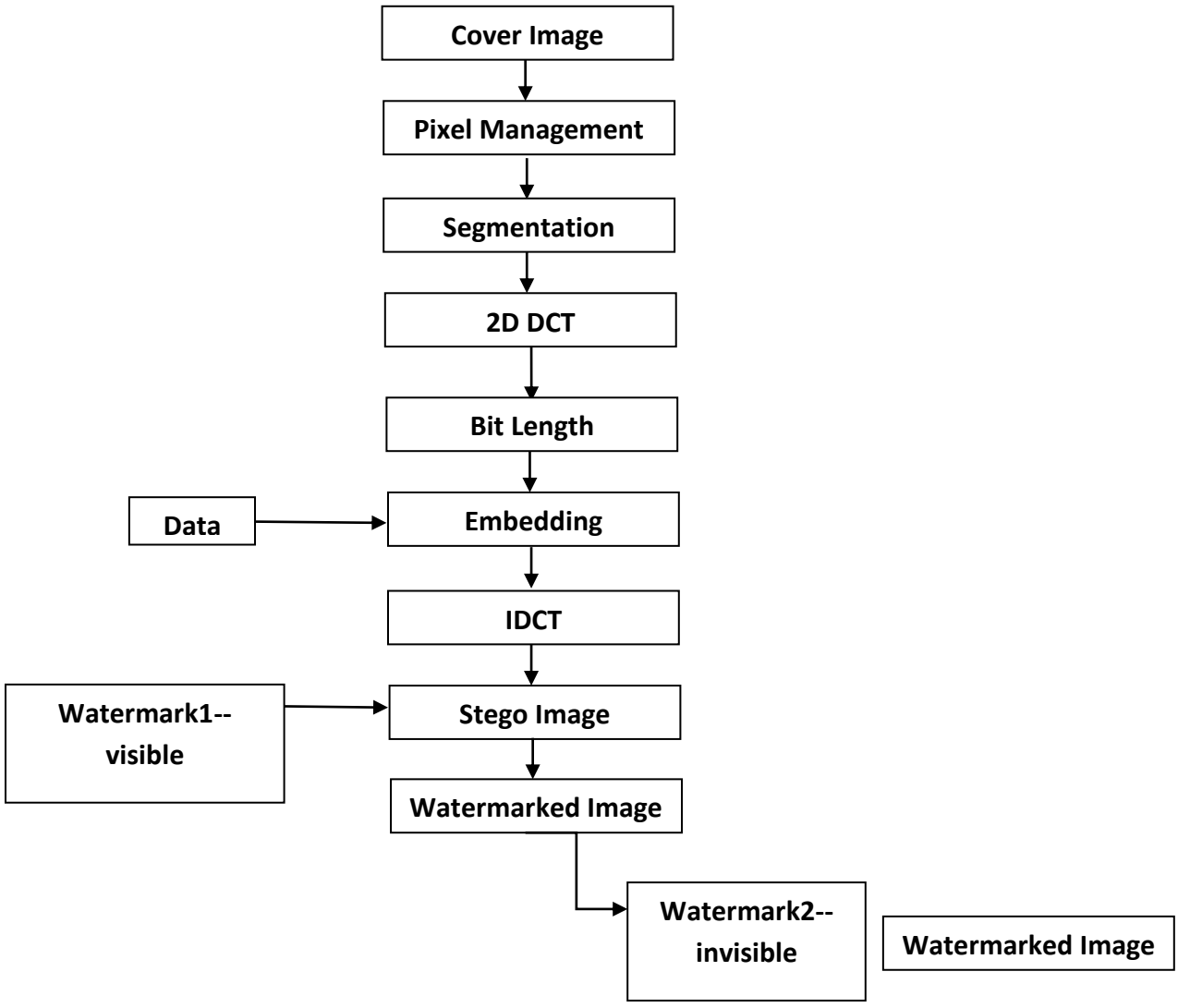


Figure 4.2.1.1 Block Diagram of Embedding Technique

Transform each 8x8 matrix into frequency domain using 2D-DCT. Four MSBs of each payload pixel are embedded into the cover image DCT coefficients in a continuous manner depends on the value of *L* to derive the stego image in DCT domain. The stego image in the transform domain is converted to the spatial domain by applying IDCT. The stego image obtained is similar to the cover

image and the difference is not perceptible by the human eye. The watermark is added to the stego image to make it protected. This watermarked image is transmitted to the destination over the open channel.

4.2.1.2 Retrieval Algorithm

The watermarked image is received at the destination over the open channel. Any intruder interfering in the transmission process will only be able to read the watermarked image and cannot extract the secret data embedded in it. The watermarked will be removed first of all to get the stego image, and then this stego image is segmented into 8x8 blocks to ensure proper retrieval of data. The 8x8 sub blocks of stego image are transformed into frequency domain to generate DCT coefficients using 2D-DCT.

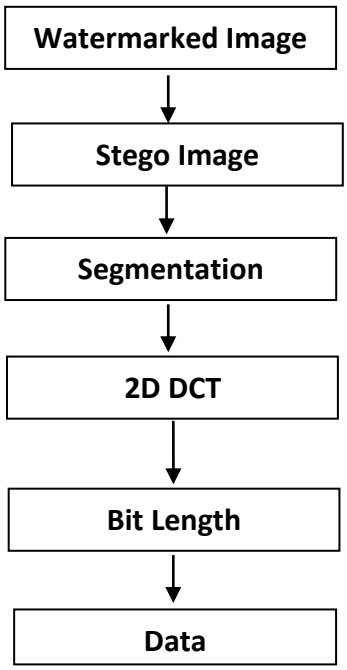


Figure 4.2.1.2 Block Diagram of Retrieval Technique

4.3 Proposed Solution

Due to the processing and bit wise steganography, the PSNR could be better using the following encoding. A color image could be considered as nxm array, where each element (x, y), ordered triple of quantized values is assigned that represents the intensity of each color component.

Therefore, a color space can be considered as a 3-D vector space, where each pixel is associated with an ordered triple of color components. Generally, a color image function can be defined by:

$$I^K = \begin{cases} C\,0\,(x, y) & \text{if } k=0 \\ C\,1\,(x, y) & \text{if } k=1 \\ C\,2\,(x, y) & \text{if } k=2 \end{cases} \quad \ldots\ldots\ldots (1)$$

The problem of best transforming an original image into a new one with only *J* colors can be considered as a vector quantization problem. In this case, the quantization space is the color space and the vectors, which should be quantized, have elements the color components C0 (x, y), C1 (x, y), and C2 (x, y). Let *N*(x, y) denotes the neighboring region of each pixel (usually the pixels of a 3*3 or 5* 5 or 32*32 mask). The intensities of each color component *c*0 (x, y), *c*1 (*x, y*), and *c*2*(x, y)* are related with the intensities of neighboring pixels, in a way that *L* extra features *c*3(x, y), *c*4(x, y), K, *c* (x, y) 2+*L* to be assigned to each input vector (*X*, r). The use of extra features affects the quantization process making easier the splitting of solid color classes. No restrictions are implied for the type of the local features. However, the features must represent simple spatial characteristics, such as min, max, entropy, median values of the neighboring masks. The extra features are obtained by using the max operator.

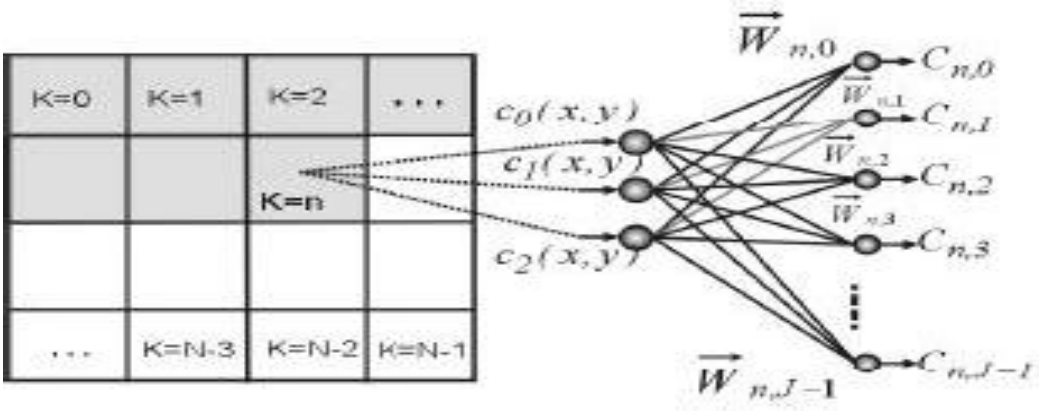


Figure 4.3 Image Splitting

The developed algorithm consists of two stages. In the first stage, the initial image is split into a pre-defined number (*N*) of windows as shown in Figure 4.3.

According to the structure of the algorithm, an input vector *X* of arbitrary dimension is mapped into one dimensional discrete map. The algo rithm is competitively trained according to the weight update function:

$$W_J^{T+1} = \begin{cases} a.(W_J - X) & \text{if } |c-i| < d \\ 0 & \text{Otherwise} \end{cases} \quad \ldots\ldots\ldots (2)$$

4.4 Steps for Steganography using 32x32 Quantization

Firstly, the image is uploaded and pixel management is performed on the uploaded image. Secondly, the image is quantized using 32x32 vector quantization. Thirdly, three colors namely red, green, blue are extracted from the embedded image and then discrete cosine transform (DCT) is applied on it for finding zeroes and ones. Lastly, the image is retrieved by the inverse process i.e. by using inverse discrete cosine transform (IDCT). The stego image is obtained which is identical to the original image. The flow chart of steps of steganography using 32x32 Quantization is shown in Figure 4.4

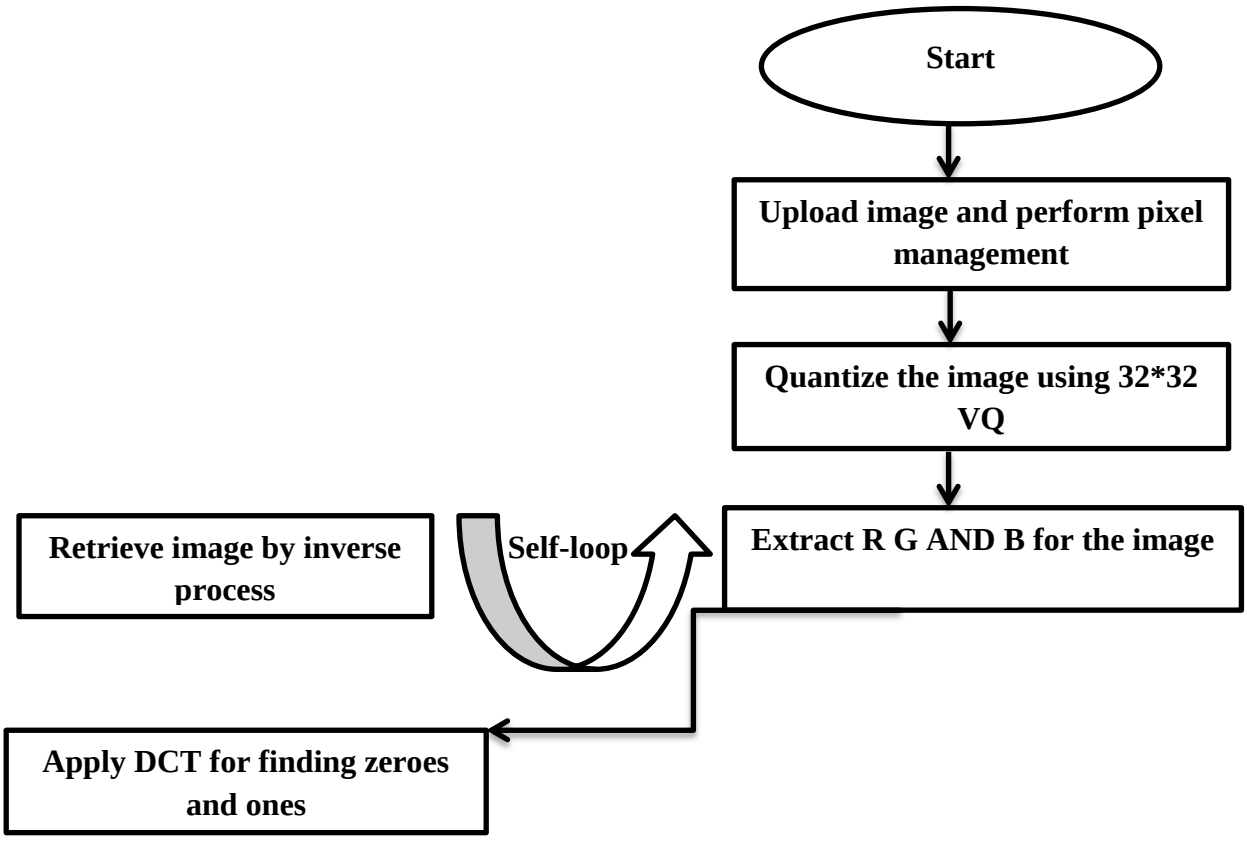


Figure4.4 Flow Diagram of Steganography using 32X32 Quantization

CHAPTER 5
MATLAB GRAPHICAL USER INTERFACE

5.1 Introduction

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.

5.2 Working of GUI

Most GUIs wait for their user to manipulate a control, and then respond to each action in turn. Each control, and the GUI itself, has one or more user-written routines (executable MATLAB code) known as callbacks, named for the fact that they "call back" to MATLAB to ask it to do things. The execution of each callback is triggered by a particular user action such as pressing a screen button, clicking a mouse button, selecting a menu item, typing a string or a numeric value, or passing the cursor over a component. The GUI then responds to these events. User, as the creator of the GUI, provides callbacks which define what the components do to handle events. This kind of programming is often referred to as event-driven programming. In the example, a button click is one such event. In event-driven programming, callback execution is asynchronous, that is, it is triggered by events external to the software. In the case of MATLAB GUIs, most events are user interactions with the GUI, but the GUI can respond to other kinds of events as well, for example, the creation of a file or connecting a device to the computer.

Although user can provide a callback with certain data and make it do anything user wants, user cannot control when callbacks will execute. That is, when user GUI is being used, user have no

control over the sequence of events that trigger particular callbacks or what other callbacks might still be running at those times. This distinguishes event-driven programming from other types of control flow, for example, processing sequential data files.

5.3 Laying Out a GUI

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools simplify the process of Laying out and programming GUIs. The GUIDE Layout Editor enables you to populate a GUI by clicking and dragging GUI components — such as buttons, text fields, sliders, axes, and so on — into the layout area. It also enables to create menus, context menus, and a toolbar for the GUI. Other tools, which are accessible from the Layout Editor, enable to size the GUI, modify component look and feel, align components, set tab order, view a hierarchical list of the component objects, and set GUI options

5.4 Programming a GUI

When user save we GUI layout, GUIDE automatically generates an M-file that user can use to control how the GUI works. This M-file provides code to initialize the GUI and contains a framework for the GUI callbacks—the routines that execute in response to user-generated events such as a mouse click. Using the M-file editor, user can add code to the callbacks to perform the functions which user wants.

5.5 Opening a New GUI in the Layout Editor

- Start GUIDE by typing guide at the MATLAB prompt. This displays the GUIDE Quick Start dialog shown in the following figure.
- In the Quick Start dialog, select the **Blank GUI (Default)** template. Click **OK** to display the blank GUI in the Layout Editor, as shown in the Figure 5.5

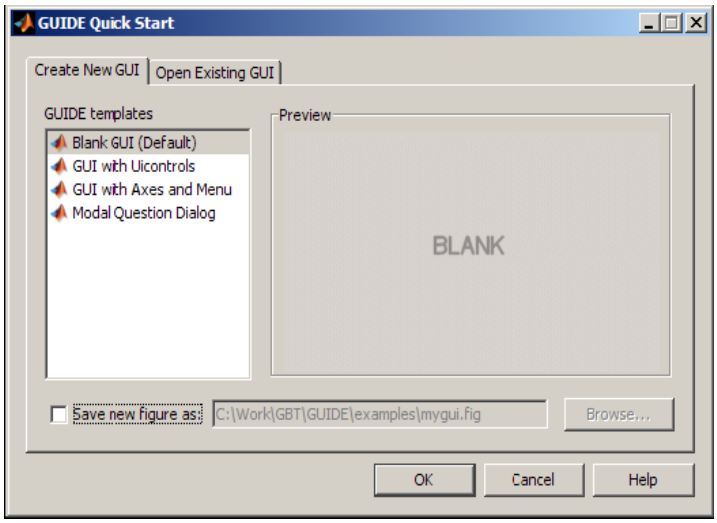


Figure5.5 Quick Start Dialog Box

5.6 Setting the GUI Figure Size

Set the size of the GUI by resizing the grid area in the Layout Editor. Click the lower-right corner and drag it until the GUI is approximately 3 inches high and 4 inches wide. If necessary, make the window larger as shown in the Figure 5.6

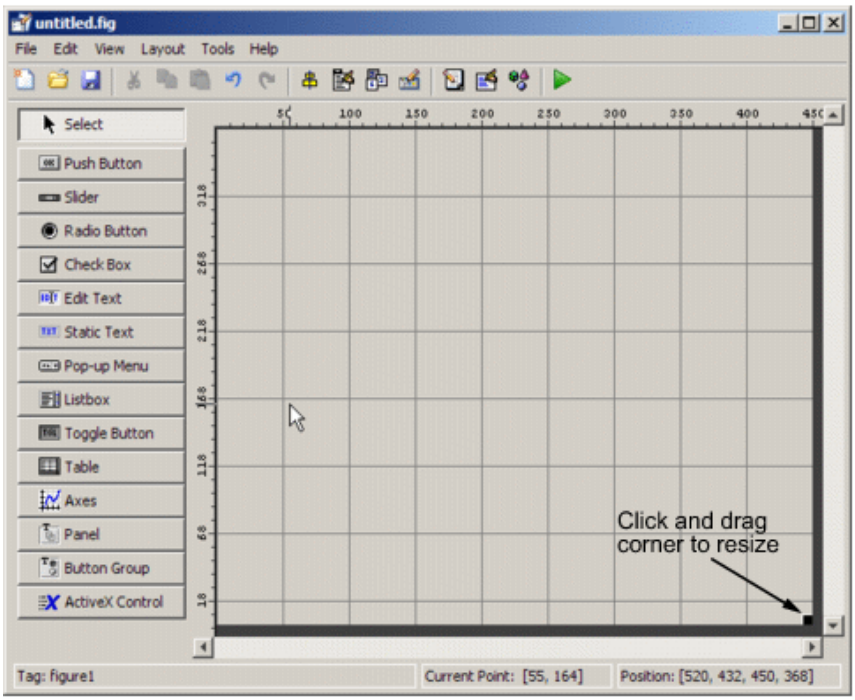


Figure 5.6 Resizing the GUI

5.7 Available Components

The component palette at the left side of the Layout Editor contains the components that user can add to your GUI. The components are shown in the Figure 5.7. User can display it with or without names.

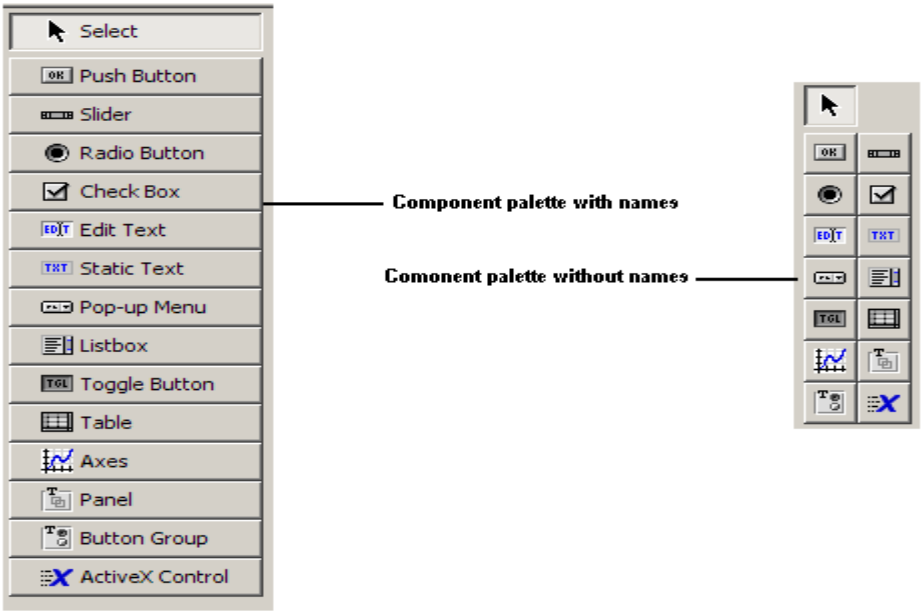
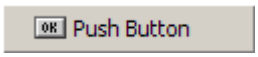


Figure5.7 Components of GUI

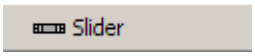
When user first opens the Layout Editor, the component palette contains only icons. To display the names of the GUI components, select Preferences from the File menu, check the box next to Show names in component palette, and click OK.

5.7.1 Push Button



Push buttons generate an action when clicked. For example, an OK button might apply settings and close a dialog box. When user clicks a push button, it appears depressed; and when user releases the mouse button, the push button appears raised.

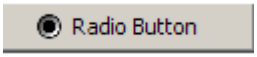
5.7.2 Slider



Sliders accept numeric input within a specified range by enabling the user to move a sliding bar, which is called a slider or thumb. Users move the slider by clicking the slider and dragging it, by

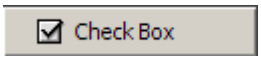
clicking in the trough, or by clicking an arrow. The location of the slider indicates the relative location within the specified range.

5.7.3 Radio Button



Radio buttons are similar to check boxes, but radio buttons are typically mutually exclusive with in a group of related radio buttons. That is, when user selects one button the previously selected button is deselected. To activate a radio button, click the mouse button on the object. The display indicates the state of the button. Use a button group to manage mutually exclusive radio buttons.

5.7.4 Check Box

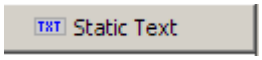


Check boxes can generate an action when checked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices, for example, displaying a toolbar.

5.7.5 Edit Text

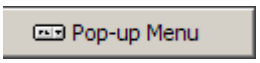
Edit text components are fields that enable users to enter or modify text strings. Use edit text when you want text as input. Users can enter numbers but you must convert them to their numeric equivalents.

5.7.6 Static Text



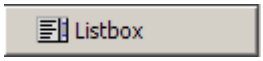
Static text controls display lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively.

5.7.7 Pop-Up Menu

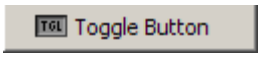


Pop-up menus open to display a list of choices when users click the arrow.

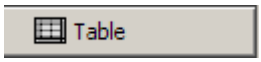
5.7.8 List Box



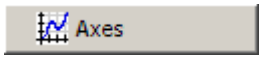
List boxes display a list of items and enable users to select one or more items.

5.7.9 Toggle Button 

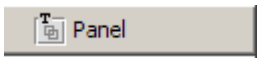
Toggle buttons generate an action and indicate whether they are turned on or off. When user clicks a toggle button, it appears depressed, showing that it is on. When user releases the mouse button the toggle button remains depressed until you click it a second time. When user does so, the button returns to the raised state, showing that it is off. Use a button group to manage mutually exclusive toggle buttons.

5.7.10 Table 

Use the table button to create a table component. Refer to the suitable function for more information on using this component.

5.7.11 Axes 

Axes enable your GUI to display graphics such as graphs and images. Like all graphics objects, axes have properties that user can set to control many aspects of its behavior and appearance. See Using Axes Properties in the MATLAB Graphics documentation and commands such as the following for more information on axes objects: plot, surf, line, bar, polar, pie, contour, and mesh. See Functions — By Category in the MATLAB Function Reference documentation for a complete list.

5.7.12 Panel 

Panels arrange GUI components into groups. By visually grouping related controls, panels can make the user interface easier to understand. A panel can have a title and various borders. Panel children can be user interface controls and axes as well as button groups and other panels. The position of each component within a panel is interpreted relative to the panel. If user moves the panel, its children move with it and maintain their positions on the panel.

5.7.13 Button Group 

Button groups are like panels but are used to manage exclusive selection behavior for radio buttons and toggle buttons.

5.7.14 ActiveX® Component

ActiveX components enable you to display ActiveX controls in your GUI. They are available only on the Microsoft® Windows® platform. An ActiveX control can be the child only of a figure, i.e., of the GUI itself. It cannot be the child of a panel or button group. See ActiveX Control in this document for an example. See Creating COM Objects in the MATLAB External Interfaces documentation to learn more about ActiveX controls.

5.8 Add Components to the GUIDE Layout Area

1. Place components in the layout area according to your design

- Drag a component from the palette and drop it in the layout area.
- Click a component in the palette and move the cursor over the layout area. The cursor changes to a cross. Click again to add the component in its default size, or click and drag to size the component as you add it.

Once user has defined a GUI component in the layout area, selecting it automatically shows it in the Property Inspector. If the Property Inspector is not open or is not visible, double-clicking a component raises the inspector and focuses it on that component.

2. Assign a unique identifier to each component. Do this by setting the value of the component Tag properties. See Assign an Identifier to Each Component for more information.

3. Specify the look and feel of each component by setting the appropriate properties. The following topics contain specific information.

- Define User Interface Controls
- Define Panels and Button Groups
- Define Axes
- Define Tables
- Add ActiveX Controls

Example of a GUI in the Layout Editor is shown in the Figure 5.8.

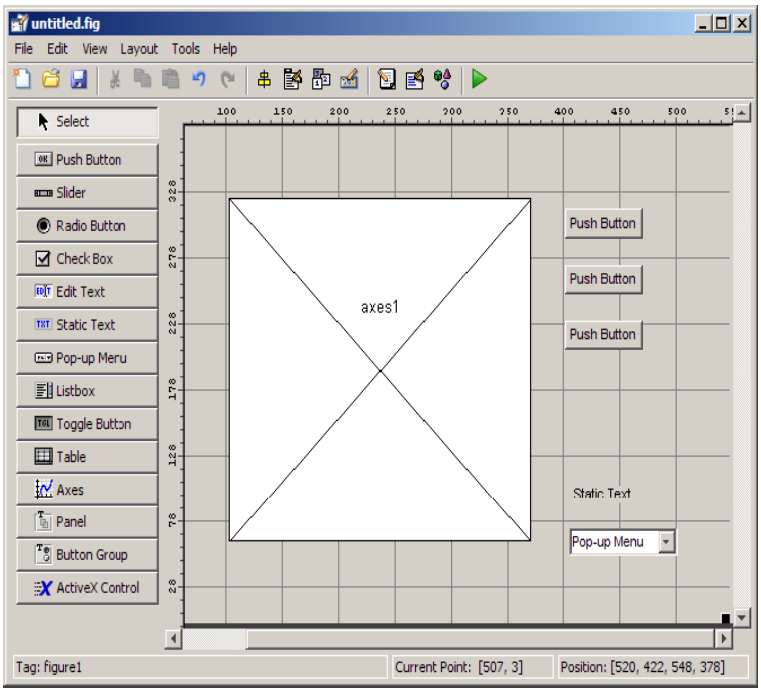


Figure 5.8 Example of GUI

5.9 Alignment Tool

The Alignment Tool enables user to position objects with respect to each other and to adjust the spacing between selected objects. The specified alignment operations apply to all components that are selected when user press the Apply button.

The alignment tool provides two types of alignment operations:

- Align — align all selected components to a single reference line.
- Distribute — Space all selected components uniformly with respect to each other.

Both types of alignment can be applied in the vertical and horizontal directions. In many cases, it is better to apply alignments independently to the vertical and horizontal using two separate steps.

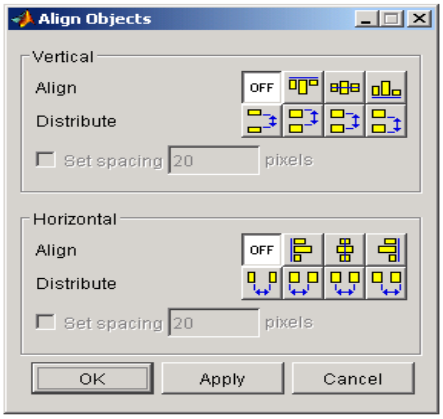


Figure 5.9 Alignment Dialog Box

5.9.1 Align Options

There are both vertical and horizontal align options. Each option aligns selected components to a reference line, which is determined by the bounding box that encloses the selected objects. For example, the following Figure 5.9.1 of the layout area shows the bounding box (indicated by the dashed line) formed by three selected push buttons.

All of the align options (vertical top, center, bottom and horizontal left, center, right) place the selected components with respect to the corresponding edge (or center) of this bounding box.

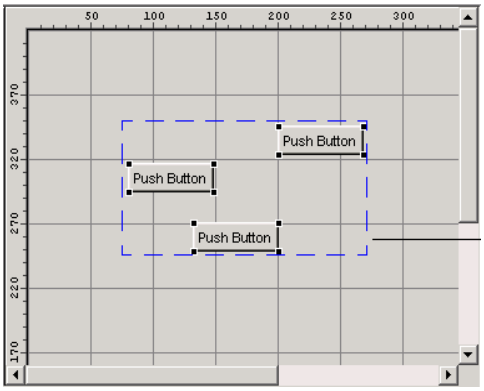


Figure5.9.1 Bounded Box

5.9.2 Distribute Options

Distributing components adds equal space between all components in the selected group. The distribute options operate in two different modes:

- Equally space selected components within the bounding box (default)
- Space selected components to a specified value in pixels (check Set spacing and specify a pixel value)

Both modes enable user to specify how the spacing is measured, as indicated by the button labels on the alignment tool. These options include spacing measured with respect to the following edges:

- Vertical — inner, top, center, and bottom
- Horizontal — inner, left, center, and right

5.10 Property Inspector

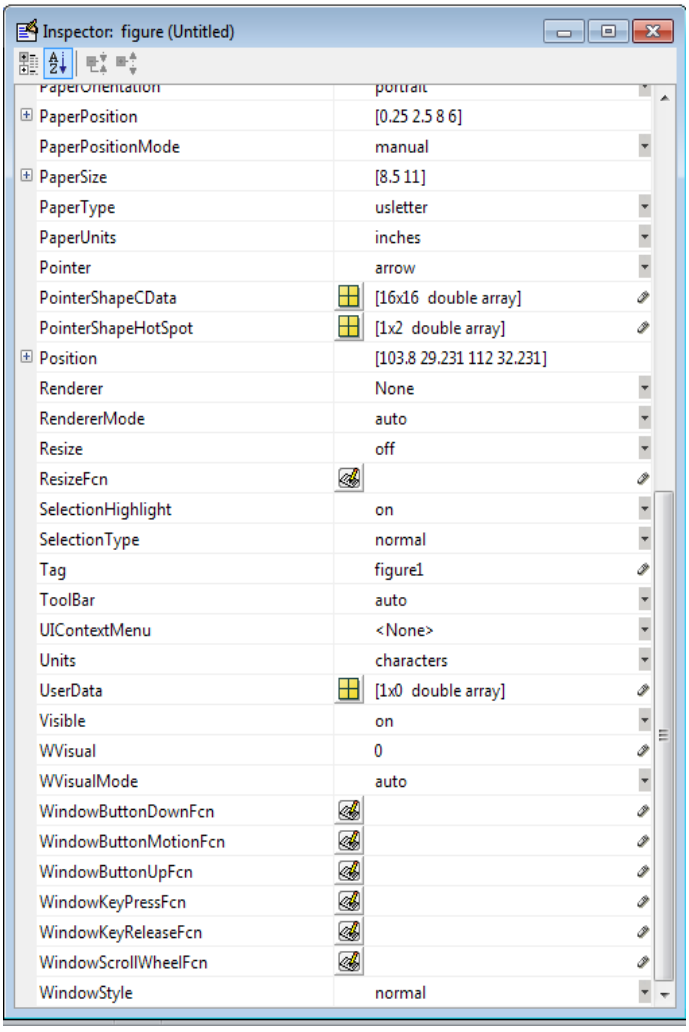


Figure5.10 Property Inspector Window

In GUIDE, as in MATLAB generally, user can see and set most components' properties using the Property Inspector. To open it from the GUIDE Layout Editor, do any of the following: Select the component user wants to inspect, or double-click it to open the Property Inspector and bring it to the foreground

- Select Property Inspector from the View menu
- Click the Property Inspector button 

The Property Inspector window opens, displaying the properties of the selected component.

5.11 Grid and Rulers

The layout area displays a grid and rulers to facilitate component layout. Grid lines are spaced at 50-pixel intervals by default and you can select from a number of other values ranging from 10 to 200 pixels. User can optionally enable snap-to-grid, which causes any object that is moved close to a grid line to jump to that line. Snap-to-grid works with or without a visible grid.

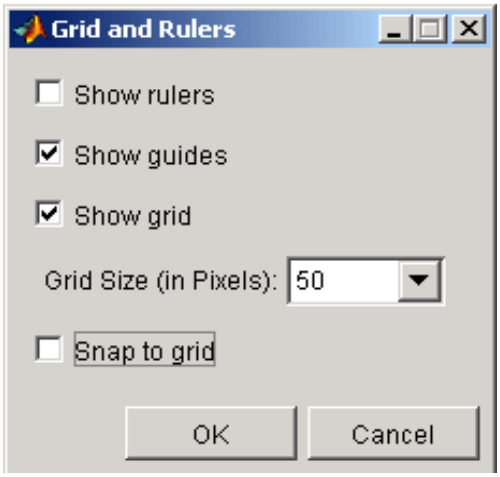


Figure 5.11 Grids and Rulers Dialog Box

Use the Grid and Rulers dialog box shown in Figure 5.11 (select Grid and Rulers from the Tools menu) to:

- Control visibility of rulers, grid, and guide lines
- Set the grid spacing
- Enable or disable snap-to-grid

5.12 Guide Lines

The Layout Editor has both vertical and horizontal snap-to guide lines. Components snap to the line when user moves them close to the line. Guide lines are useful when user wants to establish a reference for component alignment at an arbitrary location in the Layout Editor.

5.12.1 Creating Guide Lines

To create a guide line, click the top or left ruler and drag the line into the layout area as shown in Figure 5.12.1

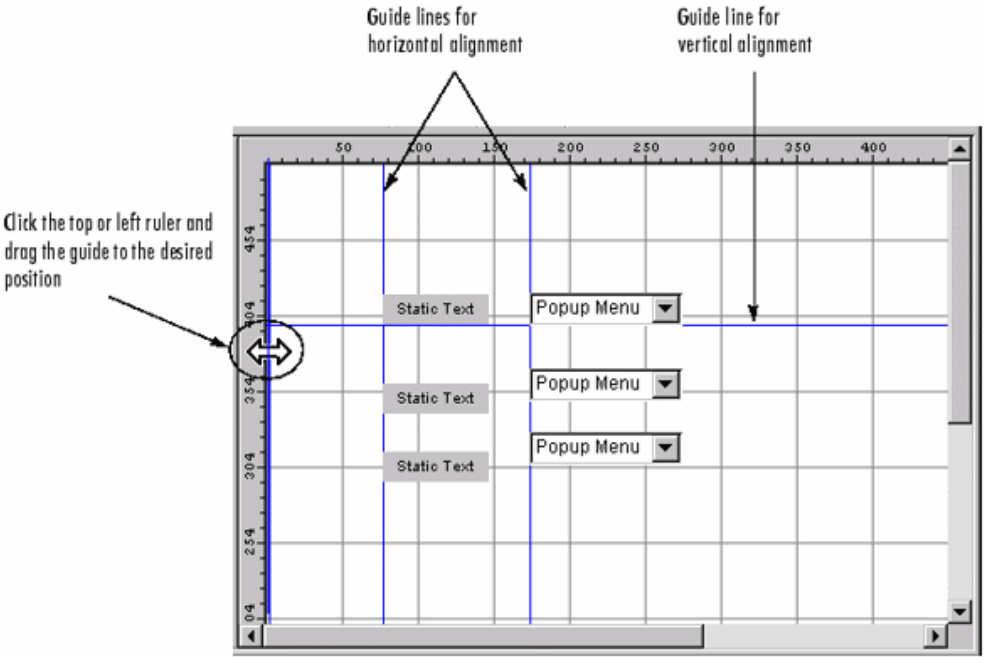


Figure5.12.1 Creation of Guide Lines

CHAPTER 6
RESULTS AND DISCUSSION

6.1 Evaluation Parameters

Three evaluation parameters have been calculated in this technique. They are PSNR, MSE and Hiding Capacity

6.1.1 Peak Signal to Noise Ratio

Peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

PSNR is most commonly used to measure the quality of reconstruction of lossy compression codecs (e.g., for image compression). The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs, PSNR is an approximation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality, in some cases it may not. One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content.

The PSNR is defined as shown in Equation 3:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned} \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

6.1.2 Mean Square Error (MSE)

It is defined as the square of error between cover image and stego image. The distortion in the image can be measured using MSE [12]. It is calculated using Equation 4:

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

I(i,j) = The intensity value of the pixel in the cover image.

K(i,j) = The intensity value of the pixel in the stego image.

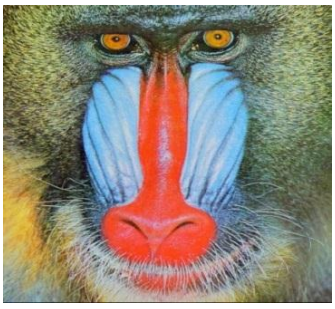
m*n = Dimensions of image.

6.1.3 Hiding Capacity

It is the size of the data in a cover image that can be modified without deteriorating the integrity of the cover image. The steganographic embedding operation needs to preserve the statistical properties of the cover image in addition to its perceptual quality. Capacity is represented by bits per pixel (bpp) and the Maximum Hiding Capacity (MHC) in terms of percentage [30].

6.2 Results

For performance analysis, Cover Image (CI) of 256x256 and 512x512 pixel are considered. The payload image is embedded into the cover image to get the quantized image by using discrete cosine transform (DCT), then de-quantized image is obtained by using inverse DCT and at last the stego image is obtained which is equal to original image. These cover images are Baboon (1), Koala (2), Lena (3) Penguins (4) Pepper (5) shown in Figure 6.2



Baboon(1)

Koala(2)

Lena(3)

Penguins(4)

Pepper(5)

Figure 6.2 Cover images

6.2.1 GUI Windows showing steganography using 32x32 Quantization Tables

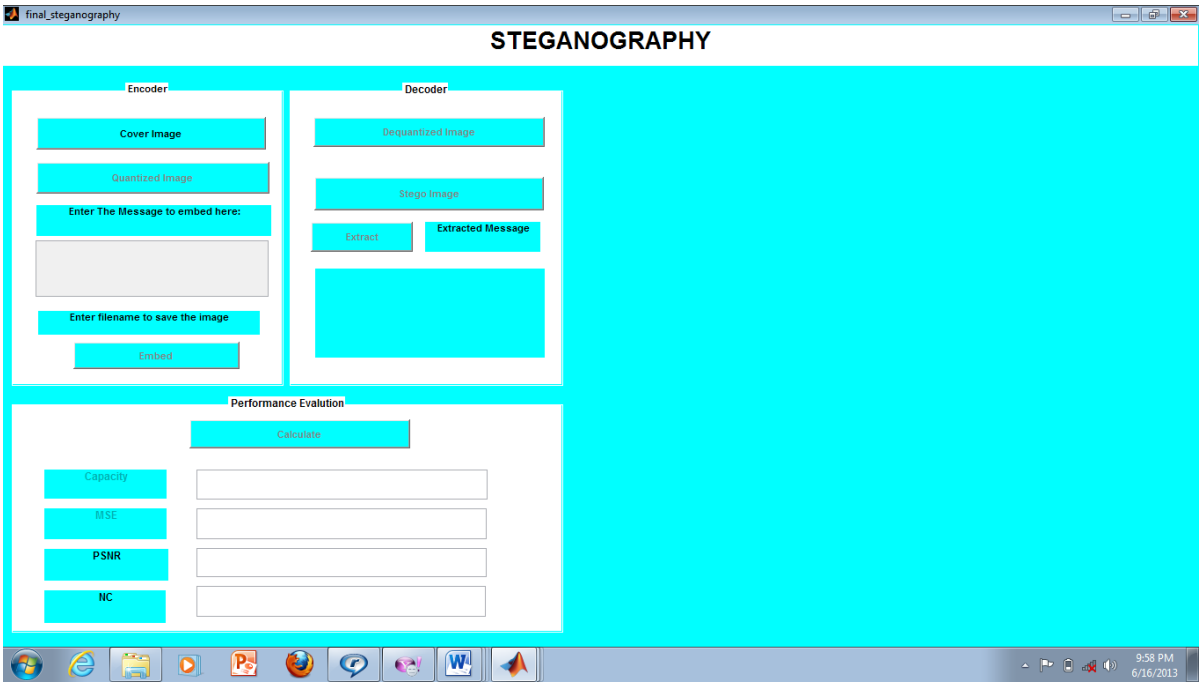


Figure 6.2.1.1 GUI for steganography using 32x32 Quantization Table

Here we are uploading the image file which is to be passed for the quantization.

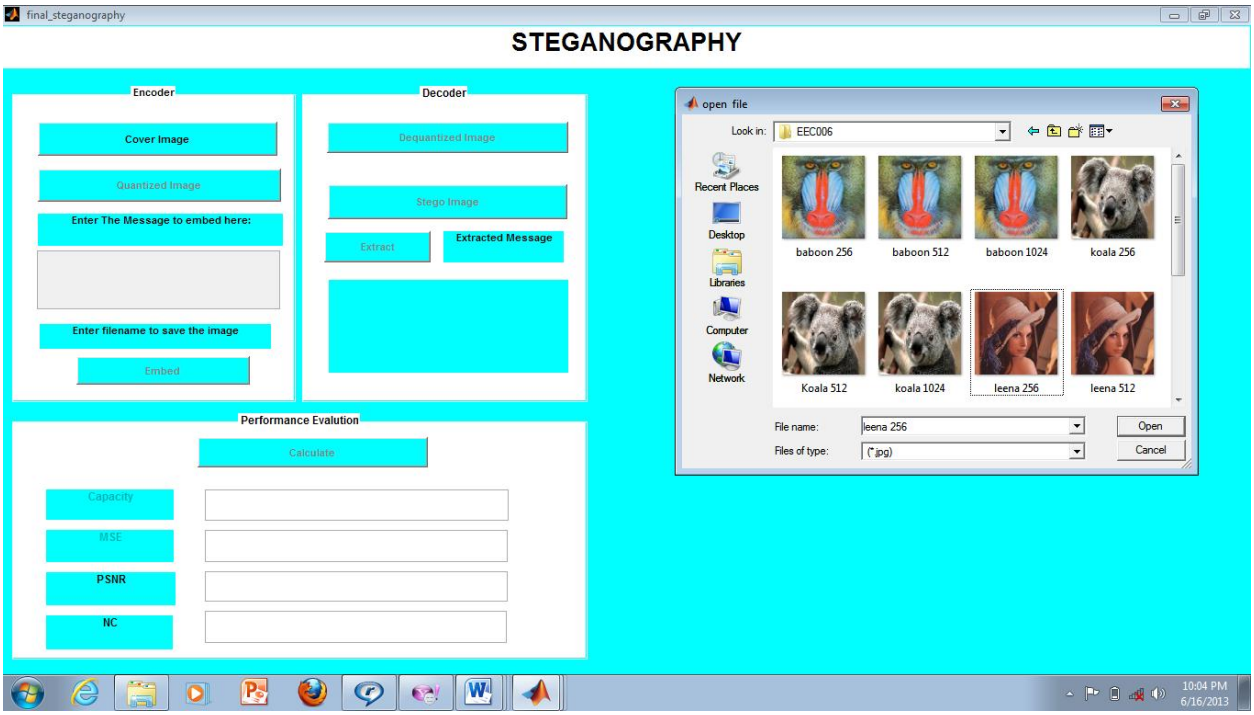


Figure 6.2.1.2 Uploading the Image file

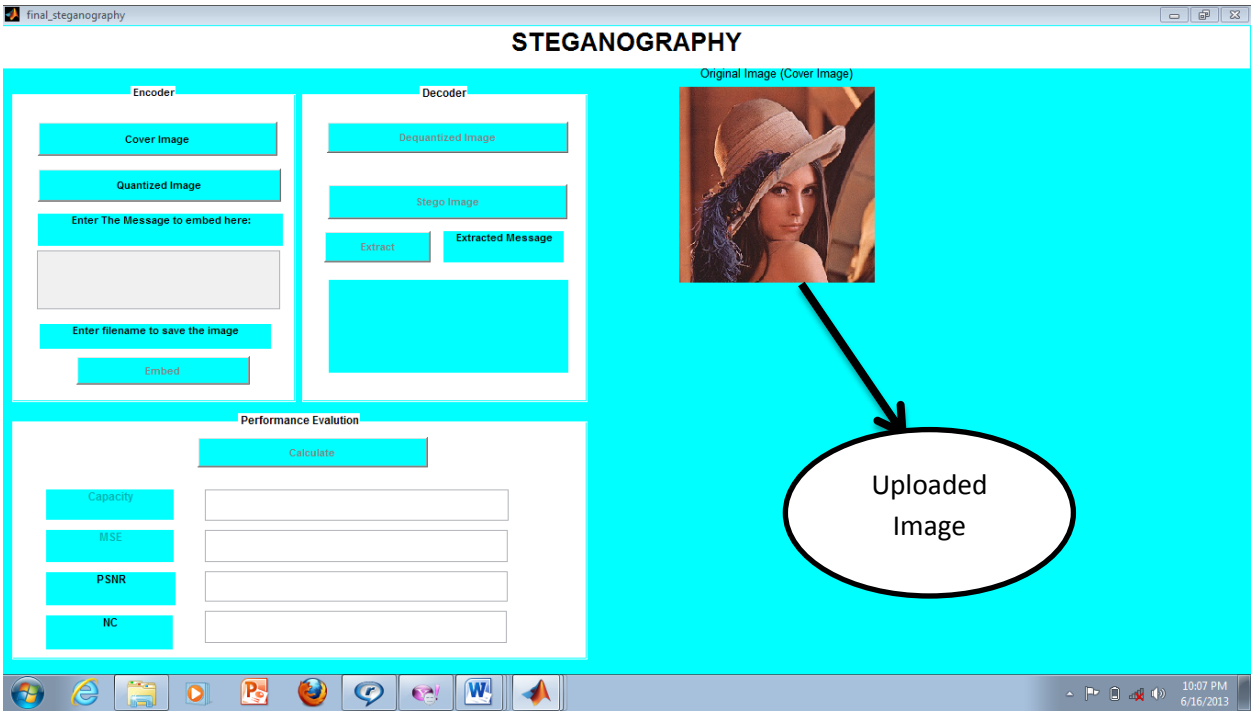


Figure 6.2.1.3 Uploaded Image

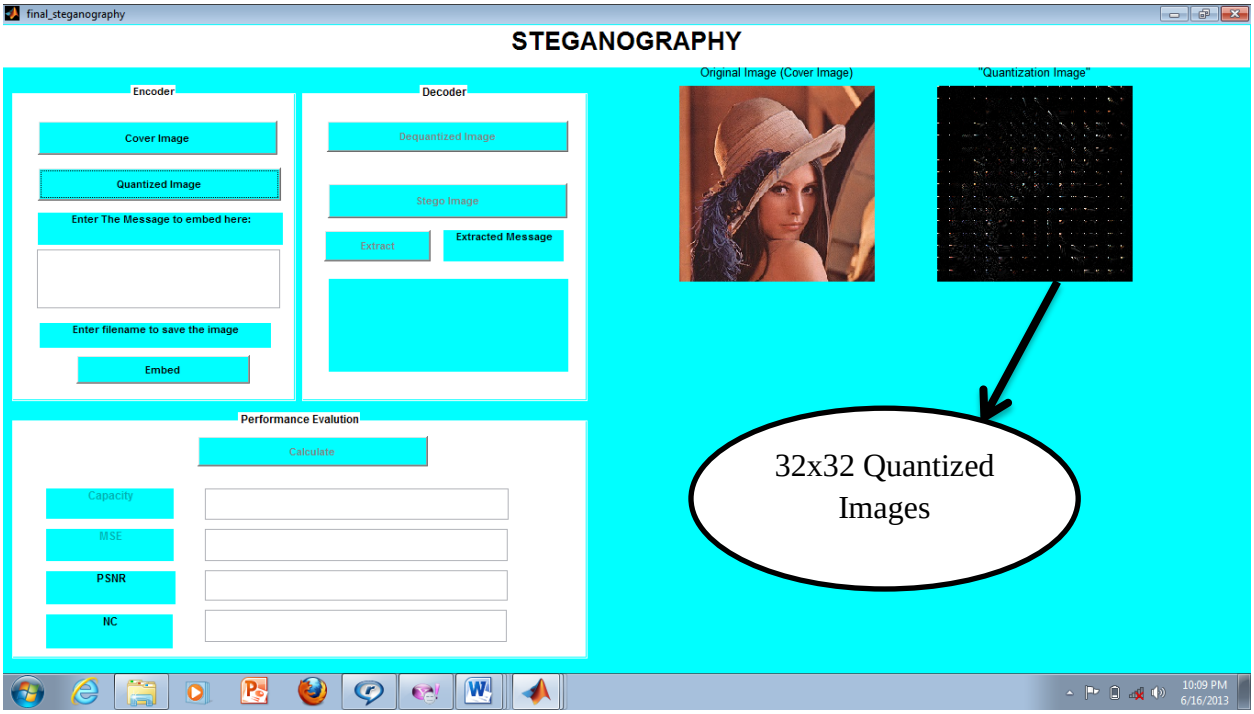


Figure 6.2.1.4 Quantized Image

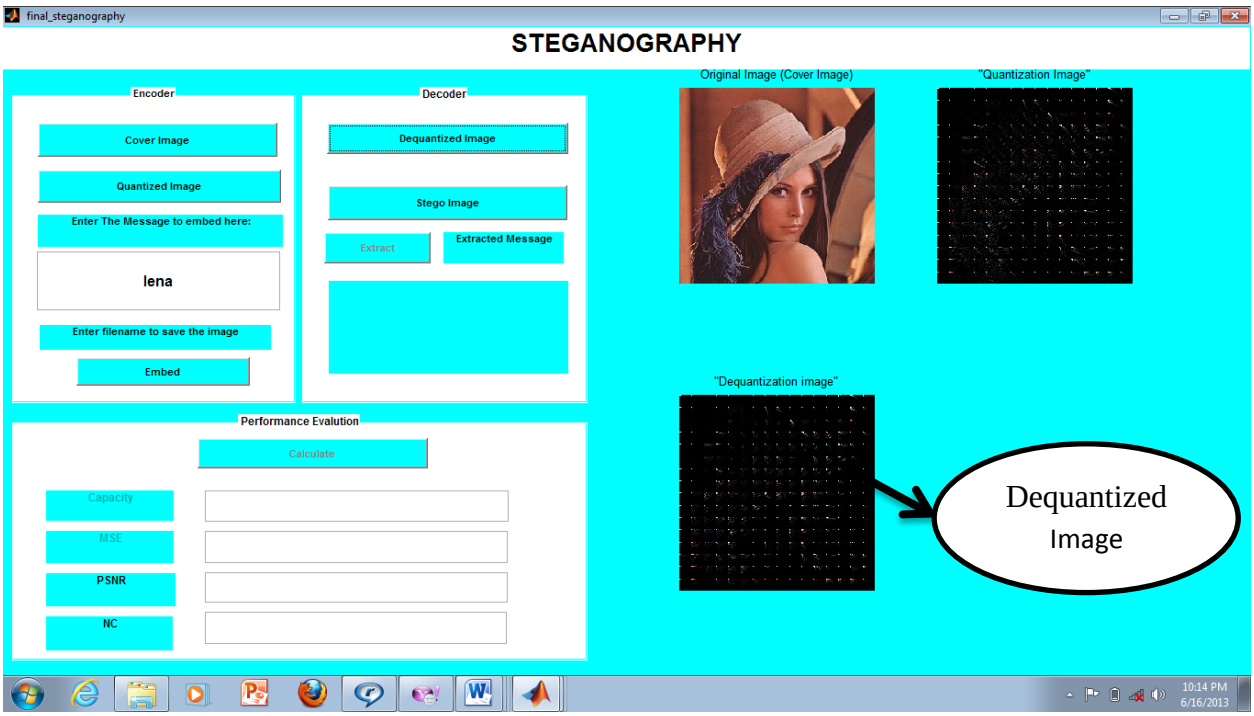


Figure 6.2.1.5 Dequantized Image

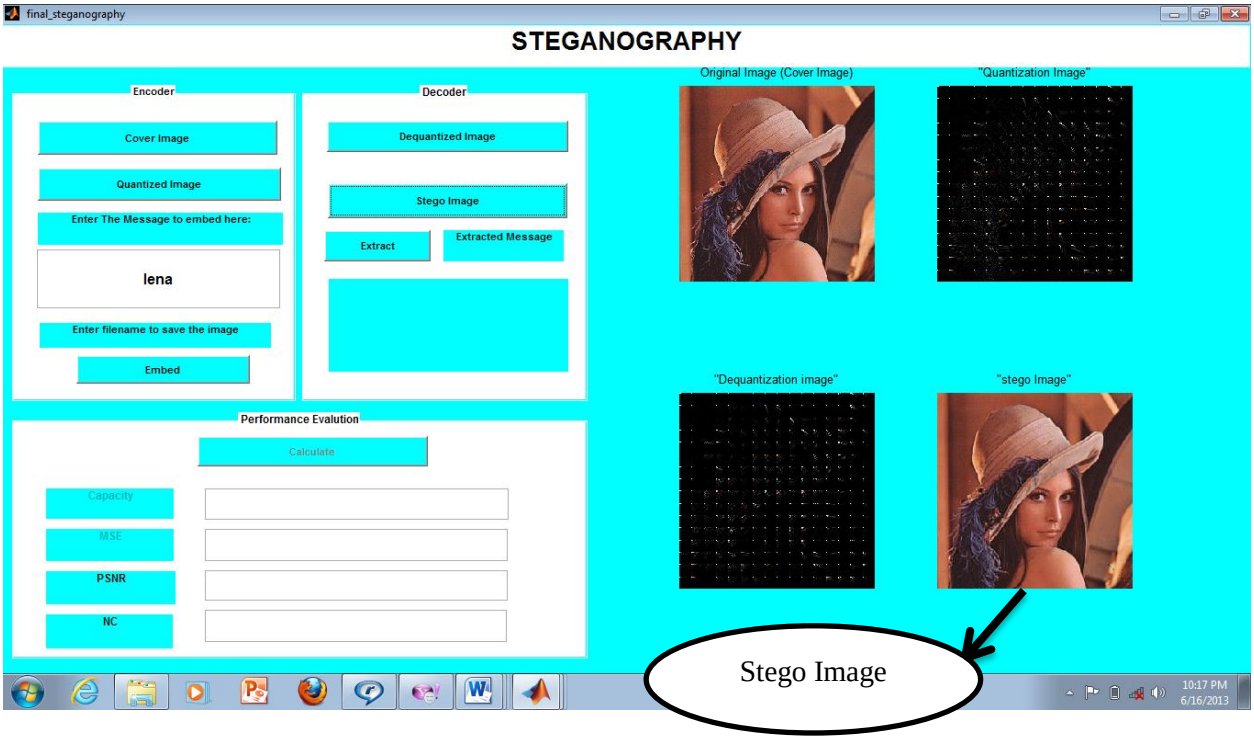


Figure 6.2.1.6 Stego Image

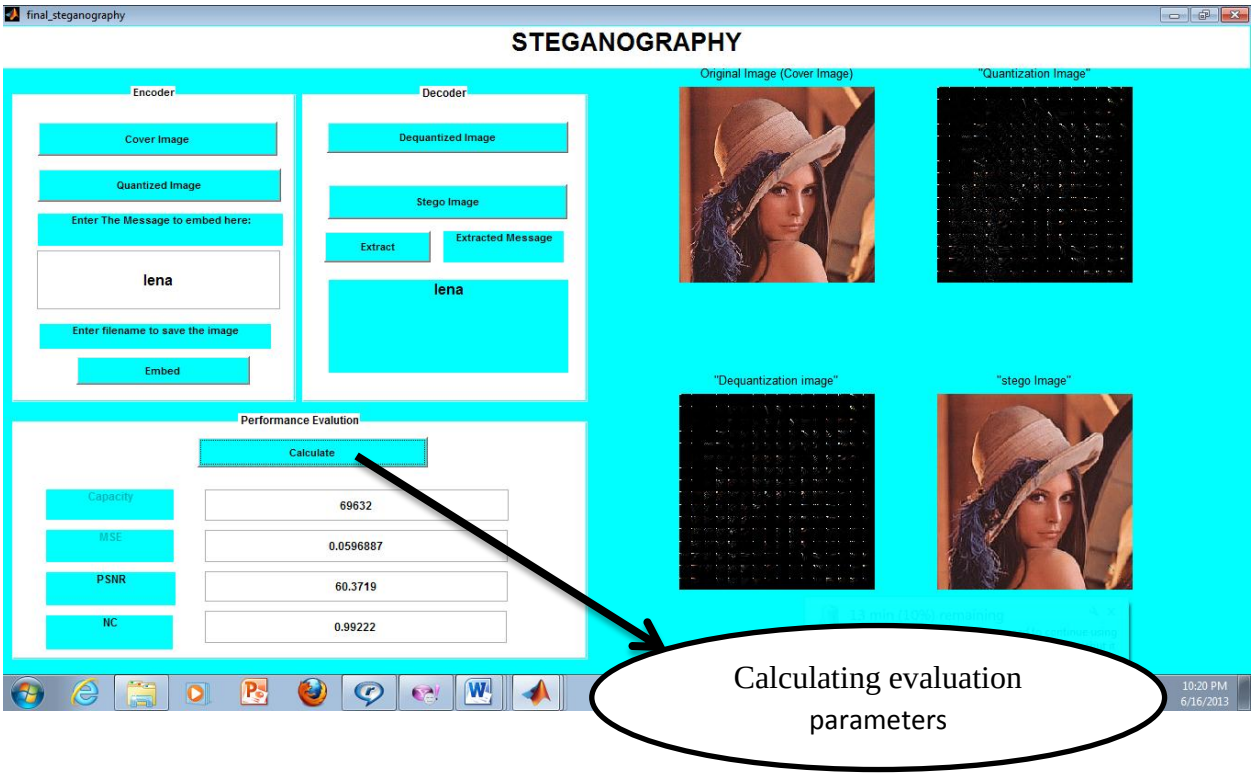


Figure 6.2.1.7 Evaluation Prameters

6.2.2 Steganography implemented on various images

6.2.2.1 Baboon Image

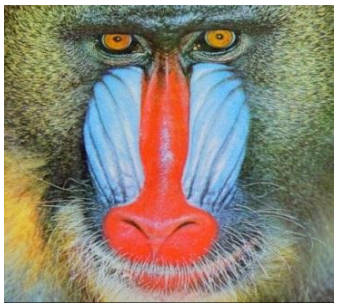


Figure 6.2.2.1 Baboon Image

(a) 256x256 Pixels

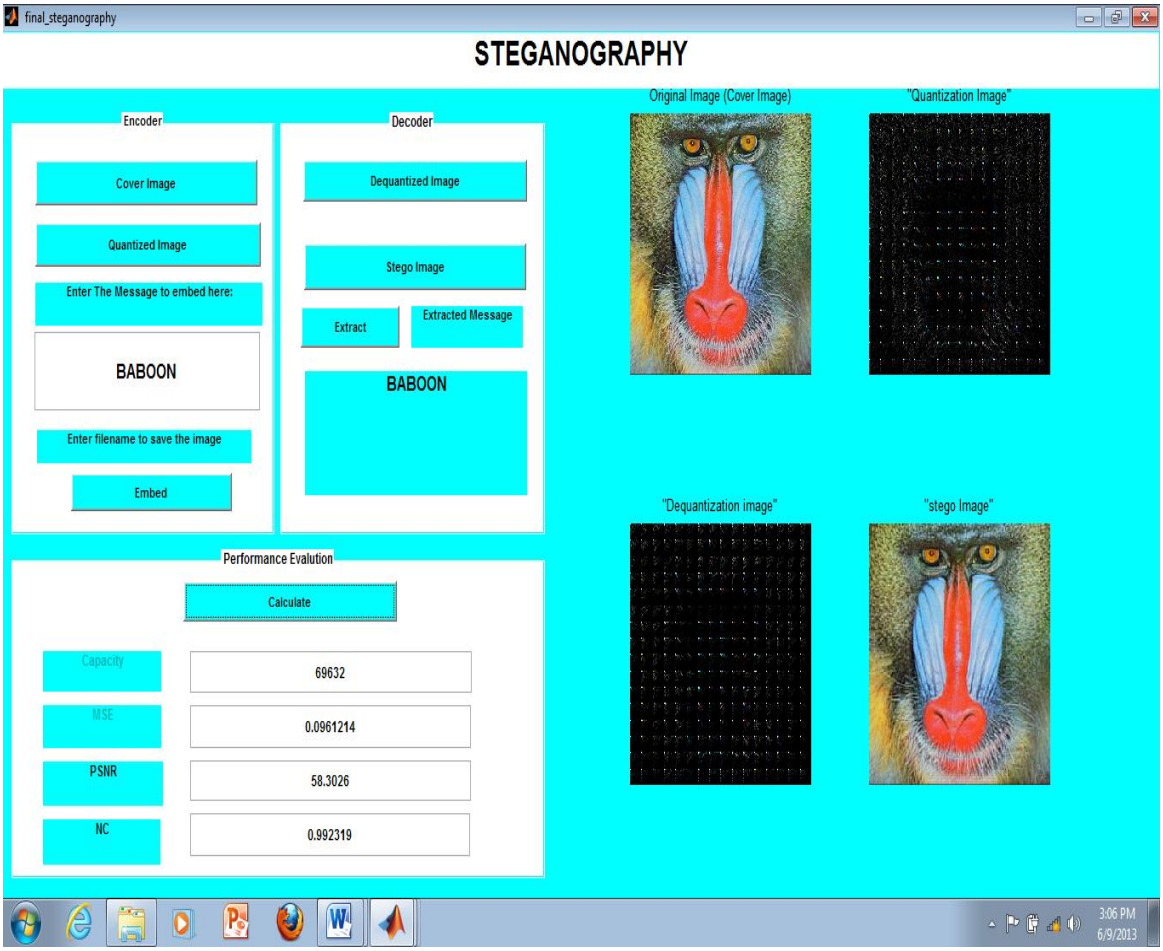


Figure 6.2.2.1(a) Steganography implemented on Baboon image (256x256 pixels)

(b) 512x512 Pixels

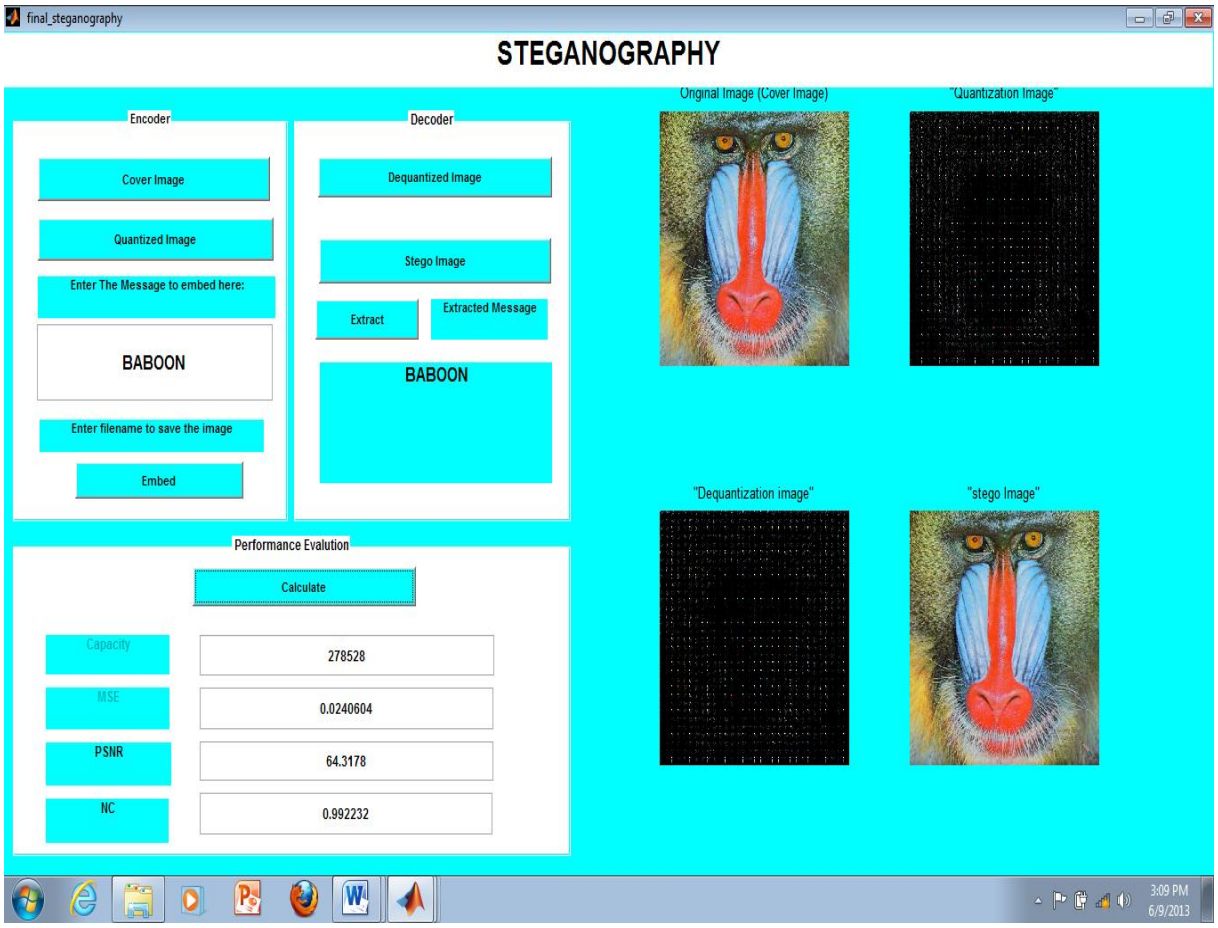


Figure 6.2.2.1(b) Steganography implemented on Baboon image (512x512 pixels)

6.2.2.2 Koala Image

Figure 6.2.2.2 Koala Image

(a) 256x256 Pixels

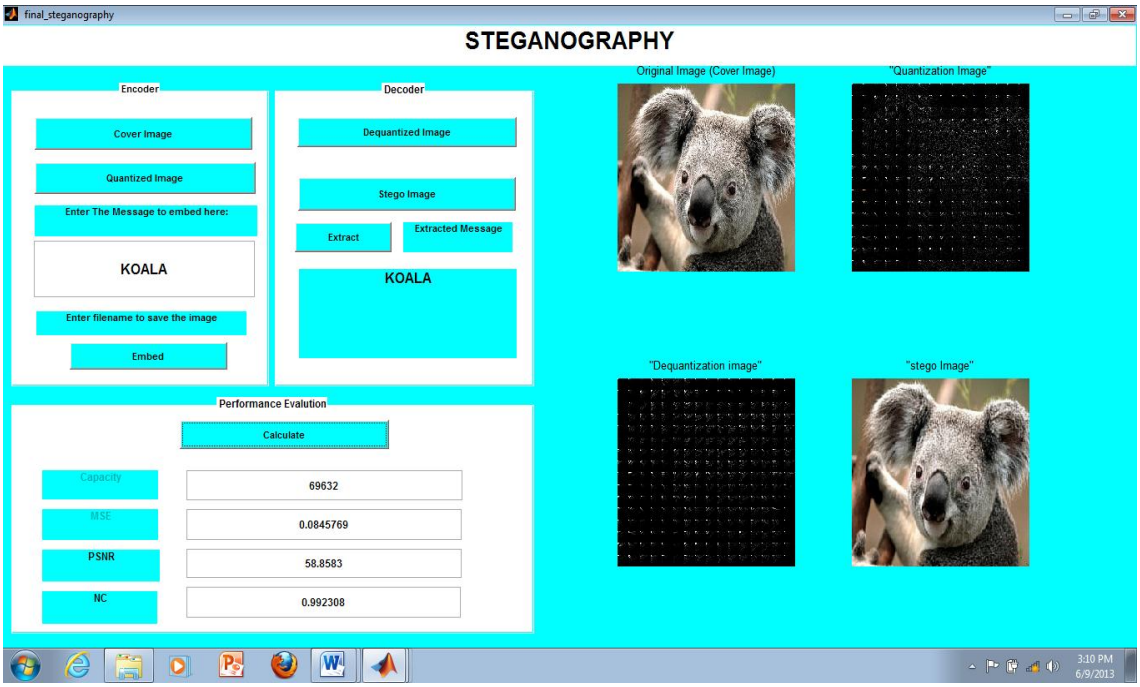


Figure 6.2.2.2(a) Steganography implemented on Koala image (256x256 pixels)

(b) 512x512 Pixels

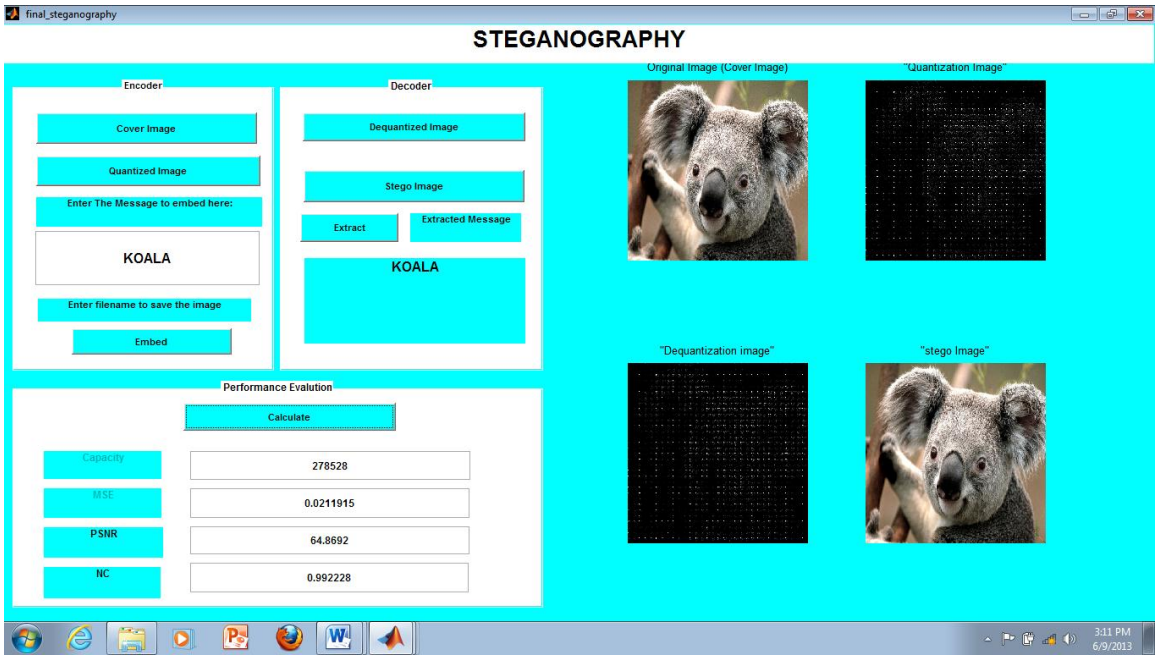


Figure 6.2.2.2(b) Steganography implemented on Koala image (512x512 pixels)

6.2.2.3 Lena Image

Figure 6.2.2.3 Lena Image

(a) 256x256 Pixel

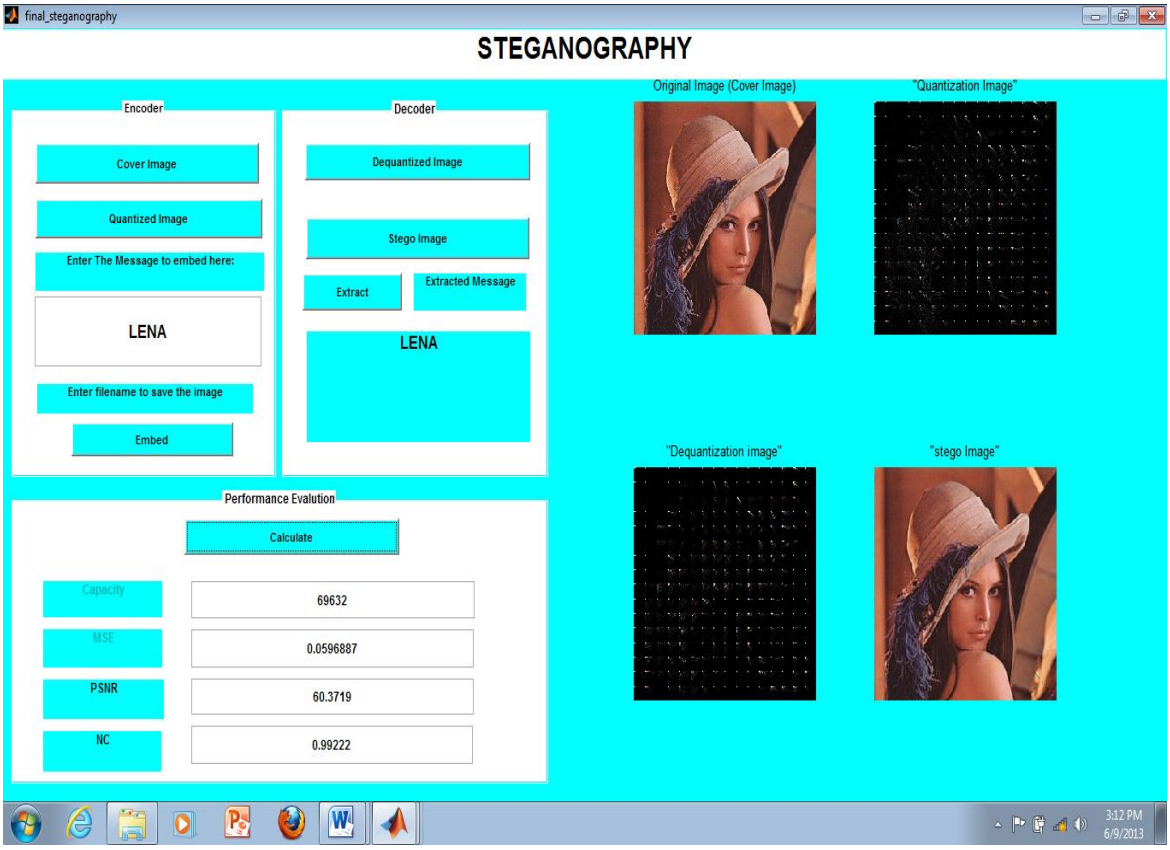


Figure 6.2.2.3(a) Steganography implemented on Lena image (256x256 pixels)

(b) 512x512 Pixels

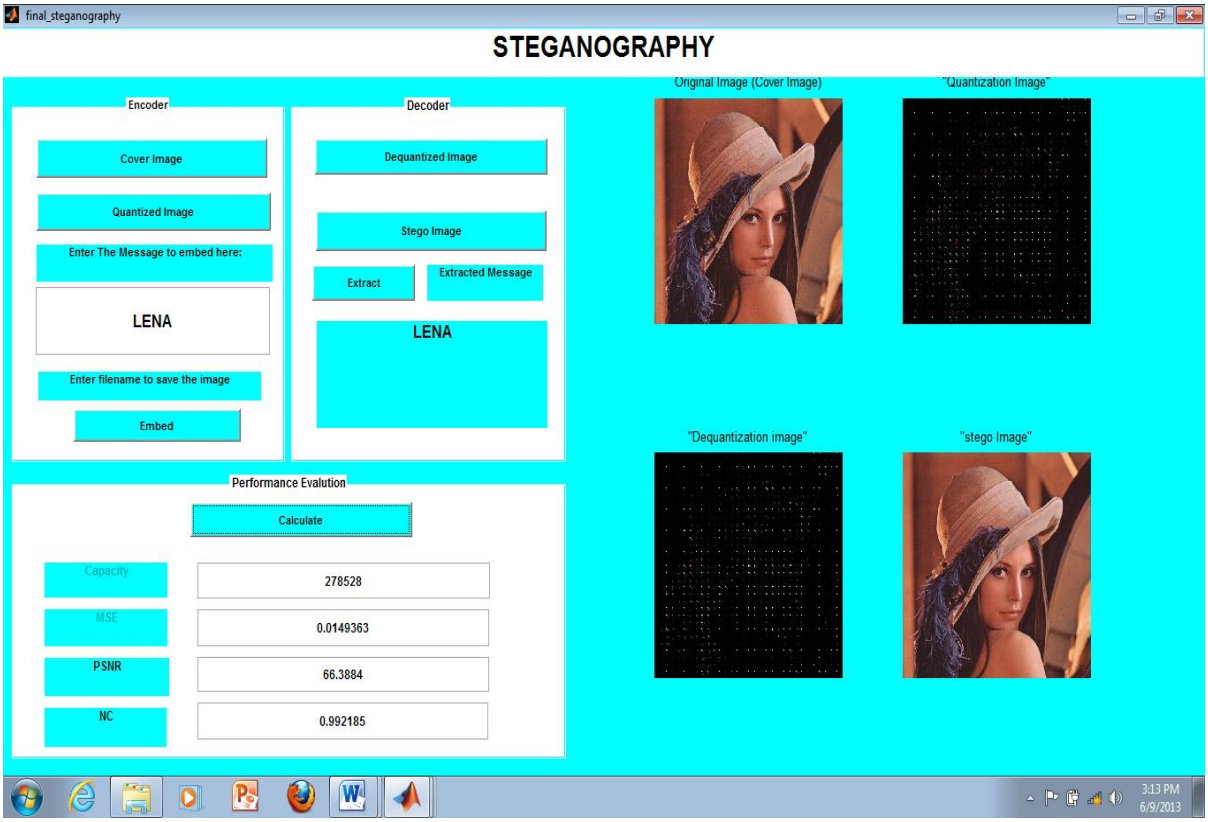


Figure 6.2.2.3(b) Steganography implemented on Lena image (512x512 pixels)

6.2.2.4 Penguins Image

Figure 6.2.2.4 Penguins Image

(a) 256x256 Pixels

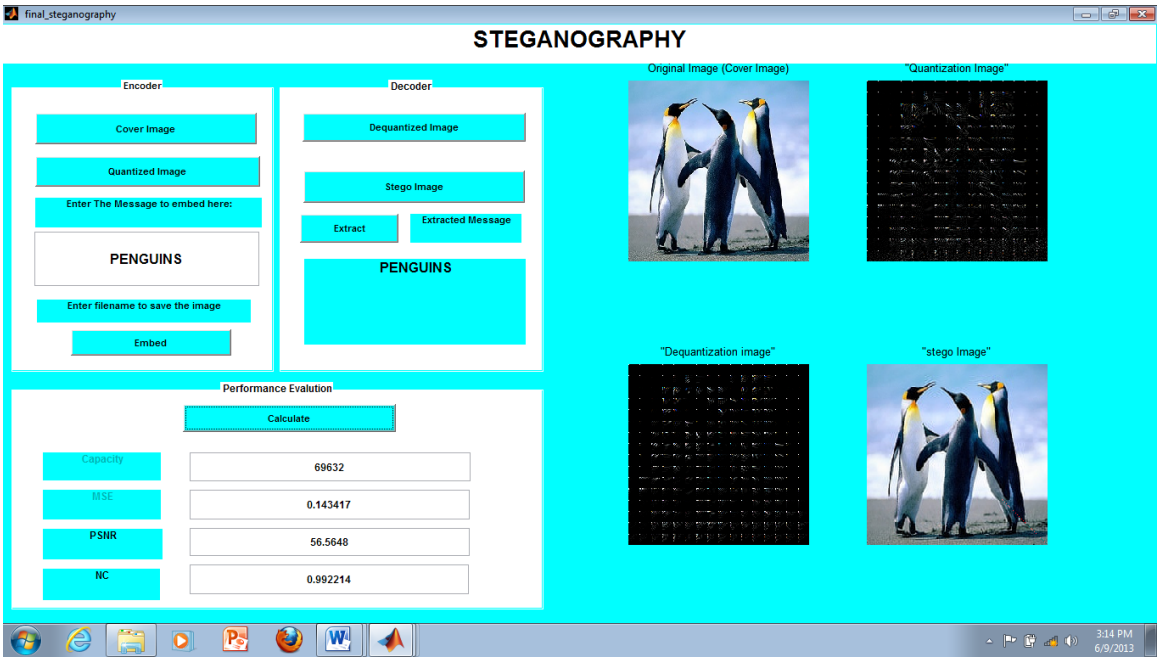


Figure 6.2.2.4(a) Steganography implemented on Penguins image (256x256 pixels)

(b) 512x512 Pixels

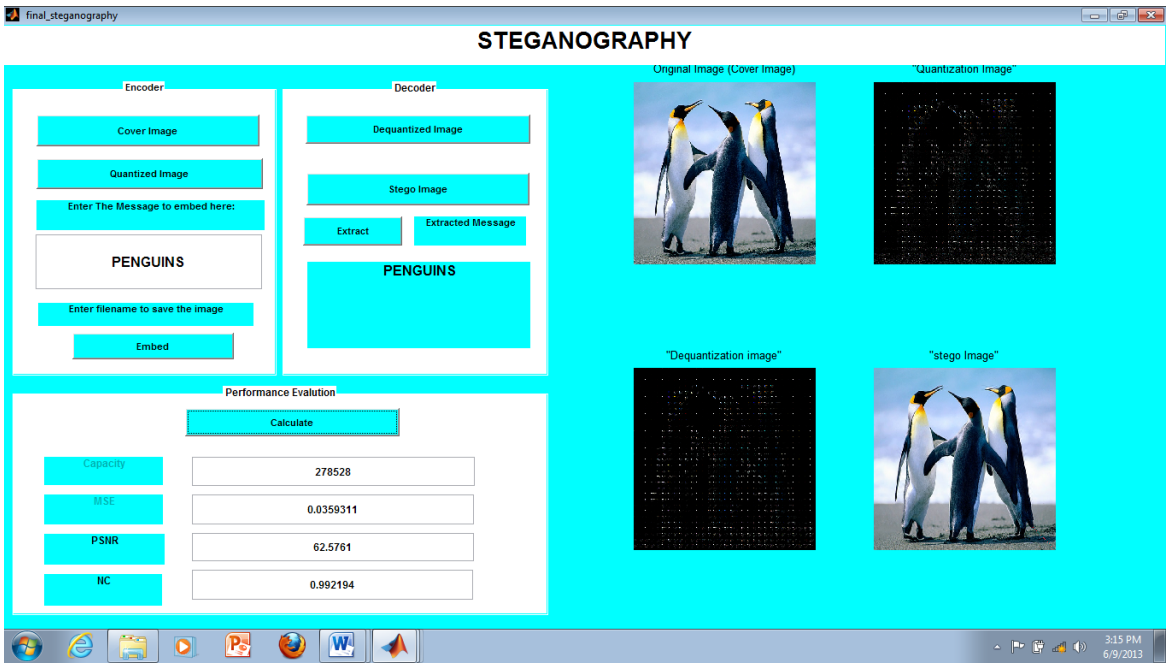


Figure 6.2.2.4(b) Steganography implemented on Penguins image (512x512 pixels)

6.2.2.5 Peppers Image

Figure 6.2.2.5 Peppers Image

(a) 256x256 Pixels

Figure 6.2.2.5(a) Steganography implemented on Peppers image (256x256 pixels)

(b) 512x512 Pixels

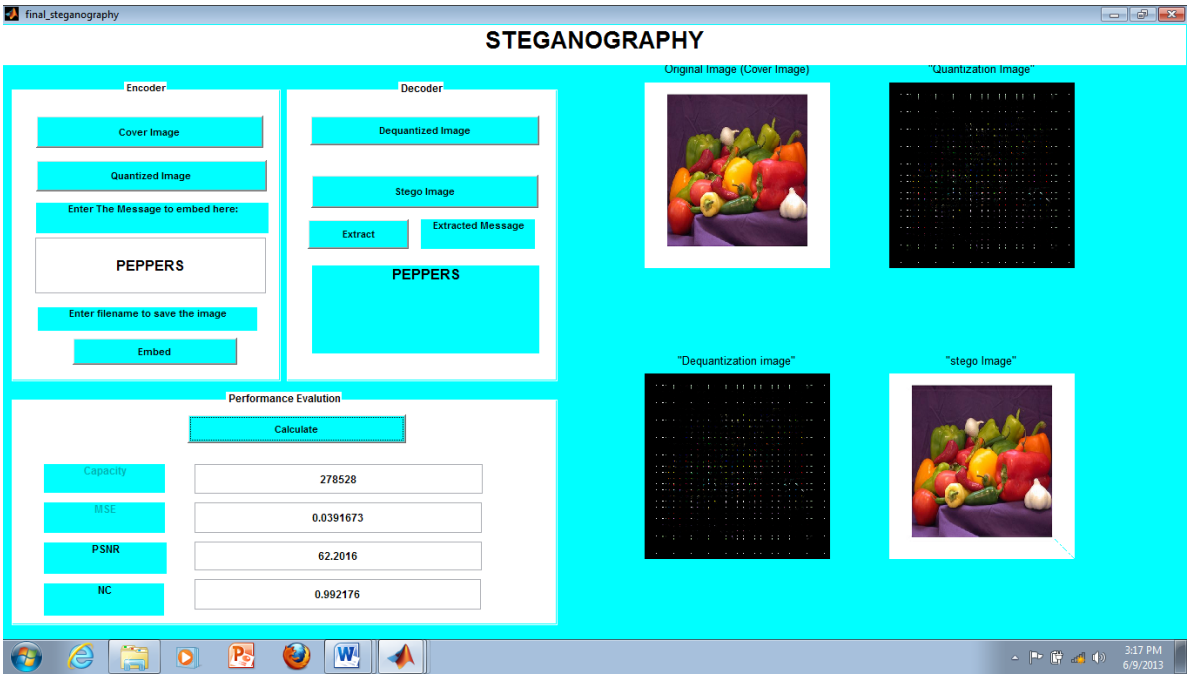


Figure 6.2.2.5(b) Steganography implemented on Peppers image (512x512 pixels)

6.2.3 Comparison of Evaluation Parameters using 32x32 Quantization Tables

Four color images namely Baboon, Koala, Lena, Penguins, Peppers of 256x256 and 512x512 pixel are taken and steganography using 32x32 quantization tables is applied on these different test images. Three evaluation parameters namely Hiding Capacity, PSNR, MSE are calculated and then these three parameters have been compared as shown in the Table VI. Table VI indicates that 512 × 512 pixel image has more PSNR and less MSE as compared to 256×256 pixel images. Comparison of Hiding Capacity, MSE AND PSNR in steganography using 16x16 Quantization Table has also been previously discussed in the section 3.6.2. Comparison of PSNR and MSE is also shown between 256x256 and 512x512 pixel in Figure6.3.3.1 and Figure 6.3.3.2.

Table VIII. Comparison of Hiding Capacity, MSE AND PSNR in Steganography using 32x32 Quantization Table

IMAGE	PIXEL	HIDING CAPACITY (Bits)	MSE	PSNR (db)
1.Baboon	256x256	69632	0.0961214	58.3026
	512x512	278528	0.0240604	64.3178
2.Koala	256x256	69632	0.0845769	58.8583
	512x512	278528	0.0211915	64.8692
3.Lena	256x256	69632	0.0596887	60.3719
	512x512	278528	0.0149363	66.3884
4.Peguins	256x256	69632	0.143417	56.5648
	512x512	278528	0.0359311	62.5761
5.Peppers	256x256	69632	0.156344	56.19
	512x512	278528	0.0391673	62.2016

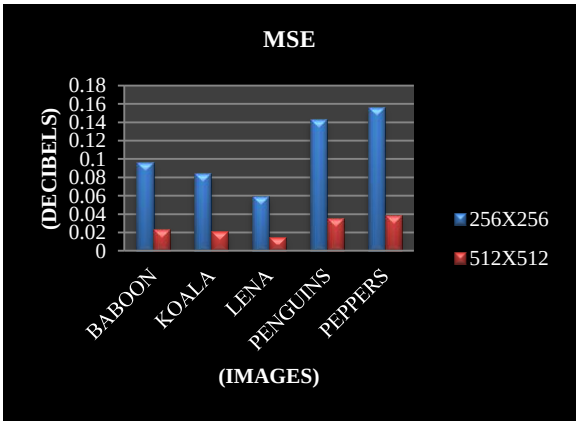


Fig 6.3.3.1 MSE Comparison

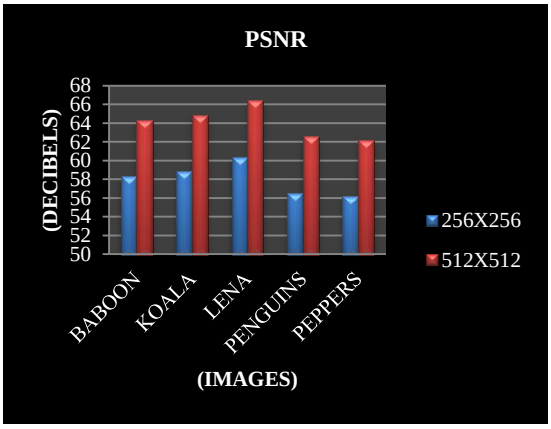


Fig 6.3.3.2 PSNR Comparison

CHAPTER 7

CONCLUSION

The proposed method with five colour images namely Lena, peppers, penguins and koala, and Baboon as steganographic covers has been implemented. The results are compared based on three parameters namely PSNR, MSE, Capacity on different test images using 32x32 Quantization. It has been found that 512x512 pixel images has more PSNR and Capacity as compared to 256x256 and 512X512 pixel image has less MSE as compared to 256x256 pixel image.

From the work which have been done, it can be concluded that 32 * 32 vector quantization is a very efficient technique for the image steganography if it is combined with DCT & IDCT technique. The only drawback in this current research work is that there is no looping method in the entire section through which the blocks can check out whether there is any other possibility to merge the image bits more, although the results which have come out from our method are quite effective but still modifications are always possible. In this thesis steganography has been done using 32x32 Quantization. The results are already available on 8x8 and 16x16 quantization and further in future this technique can be implemented through Neural Network too.

REFERENCES

[1] Vidyasagar M. Potdar, Song Han, Elizabeth Chang, "A Survey of Digital Image Watermarking Techniques" 3rd IEEE International Conference on Industrial Informatics, 2011

[2] Po-Yueh Chen and Hung-Ju Lin, "A DWT Based Approach for Image Steganography," International Journal of Applied Science and Engineering, pp. 275- 290, December 2012

[3] Hareendra's Blog, Coding Coadjutor, "History of Steganography"

[4] A. Swathi, Dr. S.A.K Jilani, "Video Steganography by LSB Substitution Using Different Polynomial Equations", International Journal of Computational Engineering Research (ijceronline.com) Vol 2 Issue 5, pp. 1620 – 1623, September 2012

[5] Masoud Nosrati, Ronak Karimi, Mehdi Hariri, "An introduction to steganography methods", World Applied Programming, Vol (1), No (3), August 2011, 191-195

[6] Soumyendu Das, Subhendu Das, Bijoy Bandyopadhyay, Sugata Sanyal, "Steganography and Steganalysis: Different Approaches"

[7] Zaidoon Kh. AL-Ani, A.A.Zaidan, B.B.Zaidan and Hamdan.O.Alanazi, "Overview: Main Fundamentals for Steganography", Journal of Computing, Volume 2, Issue 3, pp. 158 – 165, March 2010

[8] Hamid.A.Jalab, A.A Zaidan, B.B Zaidan, "New Design for Information Hiding with in Steganography Using Distortion Techniques", International Journal of Engineering and Technology (IJET)), Singapore, Vol 2, No. 1, ISSN: 1793-8236, Feb (2010)

[9] A.W. Naji, A.A.Zaidan, B.B.Zaidan, Ibrahim A.S.Muhamadi, "New Approach of Hidden Data in the portable Executable File without Change the Size of Carrier File Using Distortion Techniques", Proceeding of World Academy of Science Engineering and Technology (WASET),Vol.56, ISSN:2070-3724, pp. 493-497, 2010

[10] Weiqi Luo, Fangjun Huang, and Jiwu Huang, "Edge Adaptive Image Steganography Based on LSB Matching Revisited", IEEE Transactions on Information Forensics and Security, vol. 5, no. 2, pp. 201-214, 2010.

[11] Mathkour H, Al-Sadoon B and Touir A, "A New Image Steganography Technique", International Conference on Wireless Communications, Networking and Mobile Computing, pp.1-4, 2008

[12] Wien Hong, Tung-Shou Chen and Chih-Wei, "Lossless Steganography for AMBTC-Compressed Images," Congress on Image and Signal Processing, Vol.2, pp.13-17**,** 2008

[13] A W Naji, Teddy S Gunawan, Shihab A Hameed, B B Zaidan and A A Zaidan, "Stego-Analysis Chain, Session One", International Spring Conference on Computer science and Information Technology, pp. 405-409, 2009

[14] Vladimir Banoci, Gabriel Bugar and Dusan Levicky, "Steganography Systems by using CDMA Techniques", International Conference on Radio Electronica, pp.183-186, 2009

[15] Mankun Xu, Tianyun Li and Xijian Ping "Estimation of MB Steganography Based on Least Square Method", International Conference on Acoustics, Speech and Signal Processing, pp. 1509-1512, 2009

[16] Aos A Z, A W Nazi, Shihab A Hameed, Fazida Othman, B B Zaidan, "Approved Undetectable-Antivirus Steganography", International Spring Conference on Computer and Information Technology, pp. 437-441, 2009

[17] Mci-Ching Chen, Sos S Agaian and C L Philip Chen, "Generalized Collage Steganography on Images", International Conference on Systems, Man and Cybernetics, pp.1043-1047, 2008

[18] Bo-Luen Lai and Long-Wen Chang, "Adaptive Data Hiding for Images Based on Haar Discrete Wavelet Transform", Lecture Notes in Computer Science (LNCS), vol. 4319, 2006

[19] Kang Leng Chiew and Josef Pieprzyk, "Estimating Hidden Message Length in Binary Image Embedded by using Boundary Pixels Steganography", IEEE International Conference on Availability, Reliability and Security, pp. 683-688, 2010

[20] Mahdi Ramezani and Shahrokh Ghaemmaghami, "Adaptive Image Steganography with Mod-4 Embedding using Image Contrast," IEEE Consumer Communication and Networking Conference, pp. 243 – 246, 2010

[21] Saed Sarreshtedari and Shahrokh Ghaemmaghami, "High capacity Image Steganography in wavelet domain," IEEE Consumer Communication and Networking Conference, pp. 267 – 271, 2010

[22] HongmeiTang, GaochanJin, Cuixia Wu and Peijiao Song, "A New Image Encryption and Steganography Scheme," IEEE International Conference on Computer and Communication Security, pp. 60 – 63, 2010

[23] Nasser. N. Nasrabadi, Member, IEEE and Yushu Feng, student, Member, IEEE "A multilayer addresses vector quantization technique". IEEE Transactions on circuits and systems, Vol 37, Issue 7, pp. 912 – 921, July 1990

[24] Veerdeep Kaur Maan, Harmanjot Singh Dhaliwal, "Vector Quantization in Image Steganography", International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 4, April – 2013

[25] Manish Gupta, Dr.Amit Kumar Garg, "Analysis of Image Compression Algorithm Using DCT", International Journal of Engineering Research and Application, Volume 2, Issue 1, pp. 515 – 521, Jan - Feb 2012

[26] K B Shiva Kumar, K B Raja, R K Chhotaray, Sabyasachi Pattanaik, "Bit Length Replacement Steganography Based on DCT Coefficients", International Journal of Engineering Science and Technology Vol. 2(8), pp. 3561-3570, 2010

[27] Adel Almohammad Robert M. Hierons, "High Capacity Steganographic Method Based Upon JPEG", the Third International Conference on Availability, Reliability and Security

[28] C.C. Chang, T.S. Chen and L.Z. Chung, "A steganographic method based upon JPEG and quantization table modification", Information Sciences, vol. 141, pp. 123-138, 2002,

[29] J. Miano, "Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP", Addison-Wesley, 1999.

[30] Neha Batra, Pooja Kaushik**, "**Implementation of Modified 16×16 Quantization Table Steganography on Colour Images", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 10, pp. 244 - 250 October 2012.