
**KAUNAS UNIVERSITY OF TECHNOLOGY**
**FACULTY OF INFORMATICS**

**Deepak Barua**

# Steganographic Algorithms and Application of DNA

Master's Degree Final Project

**Supervisor**
Assoc. prof. dr. Armantas Ostreika

**KAUNAS, 2017**

**KAUNAS UNIVERSITY OF TECHNOLOGY**
**FACULTY OF INFORMATICS**

# Steganographic Algorithms and Application of DNA

Master's Degree Final Project
**Informatics (code 621I10003)**

**Supervisor**
(parašas) Assoc. prof. dr. Armantas Ostreika
(date)

**Reviewer**
(parašas) Assoc. prof. dr. Tomas Blazauskas
(date)

**Project made by**
(parašas) Deepak Barua
(date)

**KAUNAS, 2017**

**AUTHENTICATION**


KAUNAS UNIVERSITY OF TECHNOLOGY

Faculty of Informatics

(Faculty)

Deepak Barua

(Student's name, surname)

621I10003 Informatics

(Title and code of study programme)

"Steganographic Algorithms and Application of DNA"

**DECLARATION OF ACADEMIC INTEGRITY**

20 17 March 9

Kaunas

I confirm that the final project of mine, **Deepak Barua**, on the subject "**Steganographic Algorithms and Application of DNA**" is written completely by myself; all the provided data and research results are correct and have been obtained honestly. None of the parts of this thesis have been plagiarized from any printed, Internet-based or otherwise recorded sources. All direct and indirect quotations from external resources are indicated in the list of references. No monetary funds (unless required by law) have been paid to anyone for any contribution to this thesis.

I fully and completely understand that any discovery of any manifestations/case/facts of dishonesty inevitably results in me incurring a penalty according to the procedure(s) effective at Kaunas University of Technology.

*(name and surname filled in by hand)* *(signature)*

**SUMMARY**

DNA Steganography is a relatively new entry in the Steganography field and even DNA Storage is relatively new as data storage scientists are still figuring out a fast and efficient method to store data in DNA molecule which has relatively large storage capacities in comparison current day commercial magnetic or solid-state storage devices like magnetic hard drive or solid state drives, hence there is a need to evaluate a futuristic Steganography technology like DNA Steganography with the traditional established Steganography techniques like DCT.

In this scientific paper, three major Steganographic algorithms have been analysed, one DWT based Egypt, DCT and DNA Substitution which are based on some complex mathematical properties of encoding data.

The mathematical properties of these algorithms are used to store data in the medium and in this paper the efficiency of each algorithm to store secret data is evaluated by determining the PSNR value for each algorithm is determined to evaluate the performance of the algorithms.

After some basic experimental testing, the experiment shows significant difference in the signal storage quality and computational complexity of each algorithm and that there is a need to choose a specific algorithm based on the Steganographic application.

There are some other implications which can be derived from the analysis of the experimental data and that is there is extensive corruption of the data embedded into the video stream which is why there is a need to record the same data into each frame of the video to maintain redundancy.

DNA Steganography still has a long way to go before it can compete commercially with traditional Steganographic algorithms.

Thus, the goal of this research paper which is to evaluate the performance of each Steganography algorithm like DCT, Egypt and DNA Substitution has been achieved and a statistical basis to differentiate the performance of each Steganographic algorithm has been established.

**CONTENTS**

**LIST OF TABLES**

**LIST OF FIGURES**

**TERMS AND ABBREVIATIONS DICTIONARY**

DCT – Discrete Cosine Transform

DWT – Discrete Wavelet Transform

LSB – Least Significant Bit

PSNR – Peak Signal to Noise Ratio

MPEG – Motion Picture Experts Group

JPEG – Joint Picture Experts Group

UML – Unified Modelling Language

MJ2 – Motion JPEG 2000

AVI – Audio Video Interleaved

MP4 – Motion Picture Experts Group version 4

PNG – Portable Network Graphics

GIF – Graphic Interchange Format

DNA – Deoxyribonucleic Acid

BPN – Bits per nucleotide

EBI – European Bioinformatics Institute

## ĮNTRODUCTION

Steganography is a less well known technique used for information security despite being in use for 1000‘s of years [7]. The term steganography is coined from the two Greek words “stegano” and “graphia” meaning “covered” and writing respectively.

In simple terms steganography is the practice of concealed communication where the presence of the message itself is a secret [2].

Despite the early origins of stenographic techniques in the time of the Greek empire, the first literature to quote the word “steganography” appeared only in the 17th century by Johannes Trithemius in Frankfurt 1606, specifically in the first two volumes he published called “Polygraphia” and “Steganographia” which specifically discuss cryptography and steganography [3].

Steganography is useful when you want to send extremely sensitive information to a colleague without detection by a third party who has access to the same communication channel, there are two methods to address this issue.

The first method involves encrypting the message so no one else can read it but the major drawback in using this method is that the third party will be aware that a secret message is being transmitted and he/she might try to intercept and decode the secret message.

The second method is to hide the very fact that any secret message is being transmitted, some of the older techniques used for achieving this objective were by using invisible ink or by integrating the true message in another message.

The first method is based on Cryptography which is used to secure communications and the second method Steganography is used to establish a secret communication link.

People generally decide to use Steganography when they want to hide their true intensions for communicating with another person in a more traditional communication scenario, or example two people exchanging pictures of cars with one another is perfectly innocuous and will not arouse any suspicion of clandestine activity but a third-party observer can notice the patterns in the history of communication between the two entities and deduce that some clandestine activity is being coordinated.

The procedure for initiating a Steganography session usually starts with the two entities deciding on a using a Steganographic algorithm for the information exchange session, the Steganography algorithm is a mathematical formula used to place bits of the secret message data in another file and the same algorithm is used to extract the original data bits from the file in the reverse decoding process.

The mathematical formulas used in Steganography are generally not very complex and can be easily executed by a computer system.

The traditional Steganography algorithms generally encode the secret message into images or video files but DNA Steganography uses the biological properties of DNA sequences such as complementary rules of combination of each nucleotide in the DNA sequence.

Mostly DNA Steganography does not deal with actual biological DNA but instead it deals with the software representation of a DNA sequence stored in the computer memory to perform various Steganography operations.

In this paper, the Substitution method for DNA Steganography which has been found to be most efficient compared to other DNA Steganography algorithms is used as the primary DNA Steganography algorithm.

There are approximately 163 million publicly available DNA sequences in the European Bioinformatics Institute database making it very difficult to guess the DNA sequence used for performing Steganography [17].

DNA storage is also a new developing field in computer science and recently some computer scientists could store large datasets like an operating system, movie, and some other files into a DNA sequence.

Theoretically there is enormous storage potential in a gram of DNA which has the capacity to store approximately 215 petabytes of data.

**Problems and Relevance**

There are various traditional Steganographic algorithms such as LSB, DCT and Egypt, each of which have their own advantages and disadvantages like LSB is simple to implement but easy to detect, DCT is a more complex mathematical function to perform on an image but more computationally intensive and Egypt is the most computationally complex of the algorithms.

DNA steganography is relatively new this field and has its own idiosyncrasies, thus an effective yardstick is needed to measure the performance and efficiency of each algorithm and to evaluate which algorithm can be used for any future practical Steganographic application that might be built.

**The aim and tasks**

This research paper will attempt to create a comparative study of different steganography algorithms available currently in the public domain.

The basic tasks were to be performed for realizing the final goal are

- Initially there was an analysis of various traditional image and video Steganography algorithms and watermarking techniques with the aim of gaining insight into how traditionally Steganography was implemented.
- Find suitable MATLAB implementations of the various Steganographic algorithms.
- Modify the source code to enable it to run on a small device like a laptop for demonstration purposes.
- Create a statistical data collection package to collect runtime information about each algorithm that was used for the evaluation of each algorithm.

**Work structure**

The report is divided into seven chapters. In the Introduction section, the existing problem is reviewed starting with an overview of the fundamental purpose the research paper. In Section 1 the various Steganographic algorithms are analysed and analytical findings are listed. Section 2 documents the designs and implementation of the system. Section 3 details the experimental testing that has been carried out on the solution. In Section 4 the results of the experimental testing are presented and further research that could be carried out within the field of DNA steganography is discussed. In Section 5 the summary of the references to the literature that is used to conduct our research is listed and finally in Section 6 the annexes like the collected test data and a user guide for other researchers to use the steganography application is attached in the end of the paper.

## 1. ANALYSIS OF STEGANOGRAPHIC ALGORITHMS

The problem analysis of current Steganographic techniques should provide a detailed comparative analysis of the advantages and disadvantages of each Steganographic technique.

A perfect example of an image processing algorithm adapted to Steganography is the Discrete Cosine transform which is traditionally used to compress the image files into JPEG format but in the various papers analysed DCT is a good substitute for the more complex Fourier transforms which use complex numbers which are computationally harder to solve than DCT which uses real numbers.

There are several implementations of DCT available in MATLAB including an implementation of Fast DCT which is six times faster than conventional Fourier Transforms but fundamentally the principles have not changed much, the DCT also has high energy compaction property which is superior to any know transform with a fast-computational algorithm.

The next Steganography algorithm which was considered in this paper is Egypt which is based on DWT (Discrete Wavelet Transform) which in turn in related to the Haar-DWT frequency domain based transform, the technique uses a pixel scanning method in the horizontal and vertical bands representing the various parts of the frequency domain.

The four frequency bands in the frequency domain for Haar-DWT have various properties but the demonstration elaborated below is restricted to the low frequency band to store the actual image data and the rest of the higher frequency bands can be used for Steganography thus this Steganography technique has a large storage capacity.

This optimization technique utilized in Haar-DWT is the main reason image quality is maintained while ensuring high capacity for storing secret data using Steganography.

DWT based techniques also satisfy the basic requirements of any Steganography algorithm such as **Imperceptibility** which means that a human cannot distinguish between the original image and the image altered using Steganography techniques, **Security** which means the embedded message cannot be copied modified or deleted by any third party observer, **Robustness** which means that the Steganographic technique is resistant to noise introduced while transmitting or processing the image, **Statistically Undetectable** which means that it is extremely difficult to detect the embedded message using mathematical analysis of the image data and **Blind Detection** in which the message can be extracted from the image without the original image.

There is a proposal to integrate this Steganography technique into the JPEG2000 flow as that is also based on DWT.

Least Significant Bit the simplest Steganography technique which can be applied to any kind of media file in the computer system, it can be used to modify image, video, and even sound files, but it is highly vulnerable to even slight modification of the container files, it is very easy to detect if an image or video file has been altered using LSB Steganography and there are more complex automated Steganalytic techniques which can be used to detect the presence of a secret message in the container format.

The LSB Steganography technique is thus flawed but still useful for simple Steganography applications like Watermarking for video or image broadcasting which is very popular in the Television and Broadcasting industry like cable television or Direct to Home broadcasting.

Humanity is currently producing vast amounts of data creating the need for better storage devices, DNA based storage is a new concept which allows the storage of large amounts of data, DNA Steganography is on the forefront of steganography algorithms to hide secret data in host carrier.

The basic concepts of DNA Steganography are based on the natural protein sequences of the cell. In molecular biology, the information is stored in the deoxyribonucleic acid or DNA of the cell.

Some of the data hiding properties of DNA Steganography is based on the biological properties of natural DNA sequences.

In fact, the DNA Steganography techniques discussed in this paper can be implemented on a biological DNA sequence but it is practically difficult and the resulting mutations are hard to control.

The main potential use of DNA steganography could be in replacing the current key exchange algorithm proposed by **Diffie**–**Hellman**, so the unencrypted channel could be used to transmit the key for information exchange.

### 1.1. The purpose of analysis

The main purpose of analysing the DNA Steganography with other steganographic techniques and to understand their inner workings to find a suitable one for developing future applications.

### 1.2. The subject of research and problem areas

The main subject of this research is the quality of each Steganographic algorithm, this will depend on the PSNR value for each Steganographic technique given the same input [1].

The major problem areas are design and implementation of each algorithm and recording the PSNR values of each algorithm given the same input text to hide.

### 1.3. The study object Analysis

The main object of our study is to analyse Steganographic algorithms such as DCT, Egypt and DNA Steganography.

A MATLAB implementation of these algorithms is used to analyse their advantages and disadvantages.

- **Discrete Cosine Transform**

DCT is a very popular mathematical transform which is applied to various forms of image processing for example in the JPEG image format the DCT transforms are used in place of Fourier transforms which reduce the computational complexity of compressing pictures while maintaining a high level of image quality [6].

As per research papers using DCT in place of Fourier transforms could theoretically result in a six times image processing speed increases.

The basic Discrete Cosine Transform of a discrete function $f(i)\ \ where\ i = 0,1,2 \ldots\ldots, N-1$ is represented by

$$\boldsymbol{F(k) = \frac{2c(k)}{N} \sum_{i=0}^{N-1} f(i) cos \left[\frac{(2i+1)k\pi}{2N}\right]} \qquad \textbf{(1), [6]}$$

Where k = 0, 1, 2……., N-1

The opposite process of discrete cosine transform is the inverse transform which is the process that is executed to get back the image data matrix block that is used to restore the image data in pixel form.

$$\boldsymbol{F(i) = \sum_{k=0}^{N-1} c(k) F(k) cos \left[\frac{(2i+1)k\pi}{2N}\right]} \qquad \textbf{(2), [6]}$$

Where i =0, 1, 2.....N-1

And where

$c(k) = \frac{1}{\sqrt{2}}$ For k=0

=1 for k=1, 2, 3…...N-1

There are some more improved DCT algorithms in the market like FDCT (Fast Discrete Cosine Transform) which provide some more improvements over the basic formula specified in the

above section but for the purposes of this research paper the basic algorithm which demonstrates the advantages and disadvantages of DCT in a favourable manner is a good fit.

- **Egypt**

Egypt uses DWT mathematical transform to encode the Steganographic data into the image file, The Haar-DWT which is the simplest of the DWT transforms uses a two-step process to process the images files.

**Step 1:**

Scan the pixels from the left to right in the horizontal direction and then perform addition and subtraction operations on the neighbouring pixels.
By continuing with the above specified operations until all the rows of pixels are processed the two parts of the processed image pixels are extracted.
One part consists of the all the pixel sum operations which represents the low frequency part of the image pixels and the other part which consists of the collection of pixel differences which consists of the high frequency part of the image pixels.

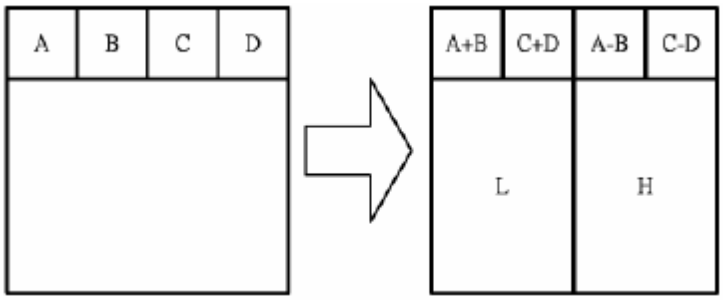


**Figure 1: Horizontal Operation on first row [5]**

**Step 2:**

In step 2 of the DWT image processing process the image pixels are scanned from top to bottom in a vertical manner, the operations performed are the same as in the first step addition and subtraction but the results are stored with results of the addition operation on top and the results of the subtraction operation are stored on the bottom of all the resulting calculations [5].

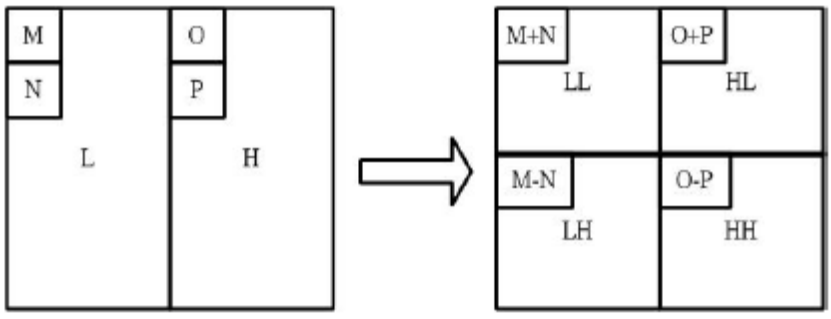


**Figure 2: The Vertical Operation[5]**

The above Figure 2 demonstrates that the image pixel data is split into 4 bands and the LL band represents the low frequency band which contains most of the original image data leaving the rest of the bands free for modification by the Steganographic algorithm to store the secret message data.

The Figure 3 below gives a good example of DWT transformed images look like.


**Figure 3: (a) Original Image (b) Result after first order 2D Haar DWT [5]**

- **Least Significant Bit**

LSB embedding in palette images is a very popular Steganographic technique as it is computationally very simple to implement, however it is very vulnerable to Steganalytic detection.

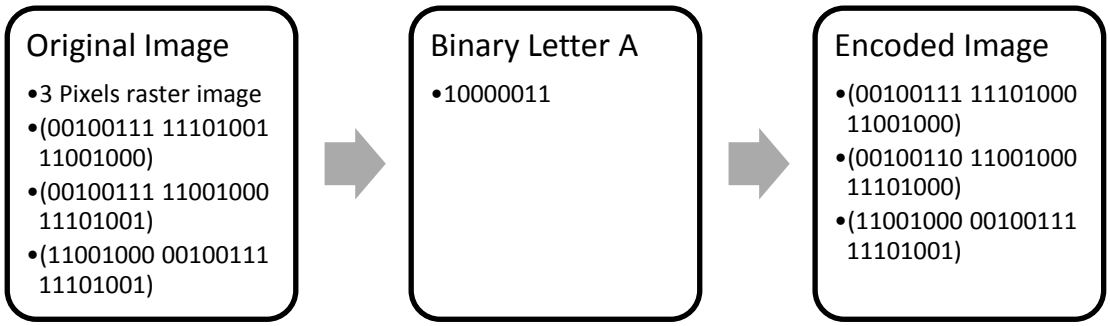


**Figure 4: LSB encoding of the word "Hello" in container data [12] [18]**

LSB steganography algorithm can be easily implemented for any simple container format of image or audio data by modifying a part of the stored representative data in the container format.
In the case of palette based image files as in our specific implementation the LSB of the three byte RGB colour representation is modified with the secret message, in this case the subsequent modification of the image data is not noticeable to a human being.
As in the case of uncompressed and lossless image compression formats which were used in our Steganographic implementation the exact representation of the image data is preserved which makes the LSB manipulation easy to perform on the given image data.
But in the case of compressed image data, the least significant bits are usually discarded to facilitate compression hence the secret message might be lost during any form of lossy compression.
This make LSB steganography unsuitable for storing secret information in data that is going to be compressed.
Palette based images like PNG and GIF are examples of lossless compressed formats, once the data in these container formats are decoded the information can be used for storing Steganographic data [10].

JPEG is a best example of a lossy image file container format, since JPEG uses DCT transform domain to store the image data and there is some negligible data loss because of the lossy compression used in this container format.

- **DNA Steganography**

DNA Steganography is an innovative steganography technique, it does not have much practical application currently and is still in the research phase.
The basic concepts of DNA Steganography are derived from biology, the genetic information is stored in deoxyribonucleic acid which is known and DNA in cells.
DNA is made up of four nucleotides which are thymidine(T), cytidine(C), guanosine(G), Adenosine(A).
The bases are connected by a DNA strands which are sugar components and phosphate groups, the DNA strands determine the direction of the DNA sequence.

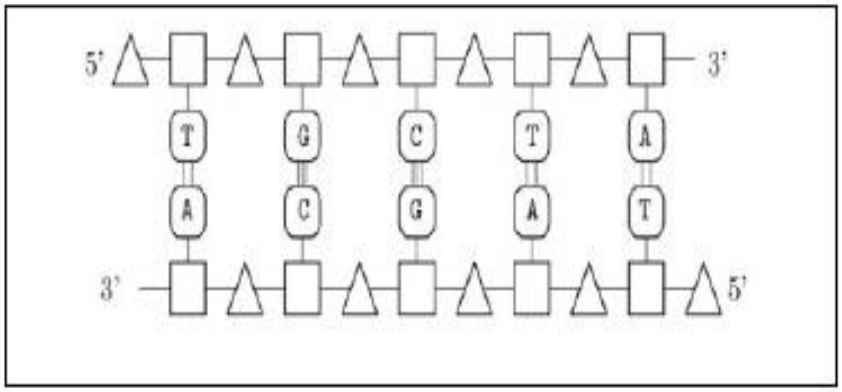


**Figure 5: DNA Strands[13]**

Hydrogen bonds connect each DNA strand to make the DNA a double strand, this enables the nucleotides to make a bond between A and T or G and C, this complementary rule is known as Watson-Crick base pairing.
C and G are bonded with a triple hydrogen bond whereas A and T are bonded with double hydrogen bond.
These complementary rules lead to the DNA double DNA strands twisting and forming the traditional DNA double helix that everyone is familiar with [13].
The first proposed scheme in software based data hiding for DNA Steganography is a lossless compression based information hiding scheme, the scheme begins by formatting the DNA sequence using a lossless compression algorithm and the secret message is appended to the end of the compressed DNA sequence to form a bit stream.
A 16-bit header is appended to the compressed file to indicate the size of the compressed data stream and as a final touch to increase the obfuscation of the data stream the entire data stream is converted back into nucleotide format represented as bits in the data storage device.
The second proposed DNA Steganography method adopts a difference expansion technique to conceal a secret bit in two neighbouring words which is categorized as capable of storing a bit or not based on a secret location map, this location map and collected LSB' s is compressed to form a bit stream which are used to extract the original words and correspondingly the secret embedded message in the data stream.
In the end, all the bit streams are converted to a new DNA sequence which can be used for transmission.
Based on our observations of real DNA sequences a special property of DNA sequences was inferred which will be used as a part of this paper and the property is that there is almost no difference between a real DNA sequence and a fake generated DNA sequence.

Another useful snippet of information is that there are approximately 163 million publicly available DNA sequences in the EBI (European Bioinformatics Institute) nucleotide sequences database, this is very useful for implementing DNA Steganography algorithms.
Based on the above two facts three basic DNA Steganography algorithms were designed, all these DNA Steganography algorithms secretly select a reference sequence S from the EBI database (or any public DNA sequence database) and only the sender and receiver are aware of this reference DNA sequence, the sender transforms the DNA sequence S to S' by embedding the secret message M into the DNA sequence S.
This new DNA sequence S' is transmitted by the sender to the receiver using some data communication channel, the receiver get the DNA sequence S' and proceeds to extract the secret message M from S' and in the process, regain the original DNA sequence S.
The sender and receiver use two data representation schemes which must remain secret to maintain the security and integrity of the DNA Steganography system, the first scheme is the binary coding rule which transforms the nucleotide letter sequence A, C, G and T to binary and also support the reverse decoding from binary to nucleotide letter's.
The second data representation scheme used by any DNA Steganography algorithm is the complementary rule wherein each nucleotide letter is assigned a complement denoted by C(x) such as ((AC)(CG)(GT)(TA)) where C(A)=C.
In this paper, it is safe to assume that the secret message M is in binary format and the size of the DNA sequence is represented as |S|.

**DNA Steganography Algorithm**

There are different types of DNA sequences available in websites like EBI (European Bioinformatics Institute), the algorithm uses the EBI database to extract 163 million DNA base sequences.
There are two main rules to the DNA steganography process, the first one being that the DNA coding technology is kept a secret between the sender and receiver, DNA coding technology is the process of converting binary data to a DNA string.

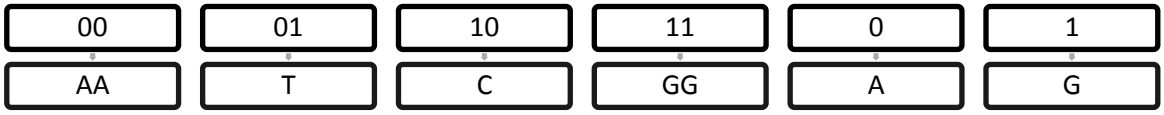


**Figure 6:DNA Coding Rules [13]**

Another rule is that the DNA complementary rule is kept secret in the sender and receiver side.

| | | | |
|---|---|---|---|
| AT | TC | CG | GA |
| AT | TG | GC | CA |
| AC | CT | TG | GA |
| AC | CG | GT | TA |
| AG | GT | TC | CA |
| AG | GC | CT | TA |

**Figure 7:Six Complementary rules for a DNA sequence [13]**

**Encoding Algorithm**

The first step in the encoding process is to extract the reference sequence from the EBI database, for this example the reference sequence is

**S= CGTATCGAATCGATGCAGAT**

Then the secret message to be sent is included

**M = '10001101'**

This is combined with a sequence of random numbers generated like

**A= {1, 3, 4, 6, 9, 11, 13, 16}**

The algorithm then follows the procedure highlighted in Figure 8 where if the primary index value is equal to secondary index random number and message bit at the secondary index is value 1 then a new encoded sequence is created which is the complementary value of the original sequence value. If the random number in the secondary index is equal to the primary index value and the message value at secondary index is 0 the procedure is to just copy the original genetic sequence to the encoded sequence value and if none of these conditions are satisfied, then the procedure is to double complement the original sequence value and append it to the final encoded sequence [13].

```
S=s1+s2+s3+…+sm = CGTATCGAATCGATGCAGAT
M=m1+m2+m3+…+mp= '10001101'
A= {A1+A2+A3+…+Ap} = {1, 3, 4, 6, 9, 11, 13, 16}
function S'=hide(S,M,A)
x= size(S,2); // get the length of reference sequence
y= size(A,2); // get the length of set of random number
for i=1:x
   for j=1:y
        if (i== A(j) && M(j)==1 )
            S'(i)= C(s(i)) // use complementary rule
        else if (i== A(j) && M(j)==0 )
            S'(i)= s(i);
         else
             S'(i)= C( C(s(i))) // use complementary rule
   end
end
end
```

**Figure 8: Encoding Algorithm for DNA Steganography [13]**

**Decoding Algorithm**

The Decoding algorithm is straight forward if the original sequence is the same as the encoded sequence then the plaintext message is of value 0 and if the encoded text is complement of the original sequence then the plaintext message value is 1 thus the original secret message is decoded [13].

```
function S'=extract(S, S')
i=1;
j=1;
x= size(S',2);
for i=1:x
       if (s(i)==s(i'))
              M(j)= 0;
              j=j+1;
       else (s'(i)== C(s(i)))
              M(j)= 1;
              j=j+1;
       end
end
```

**Figure 9: Decoding Algorithm for DNA Steganography [13]**

**Strength of proposed DNA Steganography Algorithm**

In the DNA Steganography algorithm which is selected for representation in this paper there are six main complementary rules for each DNA sequence and the main strength of DNA Steganography is the variety of public domain DNA sequences available from the EBI database.

There are approximately 163 million DNA sequences available in the database with the added option of using DNA sequences from other public DNA databases thus it is virtually impossible for an attacker to detect a secret message in a DNA sequence and it is also very difficult to guess the correct reference DNA sequence which was used for embedding the secret message.

The probability of detecting the existence of the secret message in a DNA sequence from 163 million DNA sequences is

$$\frac{1}{1.63 \times 10^8} \times \frac{1}{6} \quad \textbf{[13]}$$

This works out to approximately **$1.022 \times 10^{-9}$** which is an extremely low probability rate.

### 1.4. Existing methods for resolution analysis

Based on the various research papers [1][2][6] perused there is no existing comparison of signal to noise ratio between traditional Steganography algorithms and DNA Steganography algorithms.

### 1.5. The Aim Objectives and advantages pursued

The main objectives of this project are to find the advantages and disadvantages of each Steganographic algorithm compared with DNA Steganography.

### 1.6. Upwards the decision definition

Based on our PSNR analysis the best fit algorithm for pursuing in future Steganographic projects can be decided.

### 1.7. The Analytical findings

Based on our research into the various Steganographic algorithms our preliminary findings suggest that Least Significant Bit Steganographic algorithm is the least computationally expensive as it just involves inserting some extra bits of information in already defined data structure of the video codec, Discrete Cosine Transform is a relatively moderate image compression algorithm which can store extra Steganographic data in the signal data, it is four times faster than compressing the image using fast Fourier transforms, the main advantage being DCT uses the real number space to store the data.
The Egypt algorithm uses both spatial and temporal domains to store the information, hence the ability to store more data for the same amount of video information as DCT [12].

The DNA Steganography substitution algorithm has good capacity to store binary information on a large scale as DNA strands can have millions of DNA sequences grouped together and it provides a good obfuscation of the information to be transmitted.

## 2. REQUIREMENTS AND DESIGN

The requirements and design specified below are for a MATLAB implementation of various Steganographic algorithms like DCT and Egypt.

The following sections provide a detailed description of the various operations performed by the MATLAB functions to encode and decode the secret message into video files using the Steganographic algorithm specified in the implementation of the function.

### 2.1. Requirements specification

- Build or find some library in MATLAB that can embed given input text data into a Steganographic image.
- DCT, LSB and Egypt algorithms must be supported
- Instead of using an advanced video codec like MPEG-2 it is more prudent to use something simple like mj2 which consists of video frames with basic JPEG compression.
- The output of the decoder implemented must calculate the PSNR for each video frame encoded with the Steganographic data [1].

### 2.2. The project, a formal description

A Detailed system diagram of the proposed implementation should give an overall design view of the project.

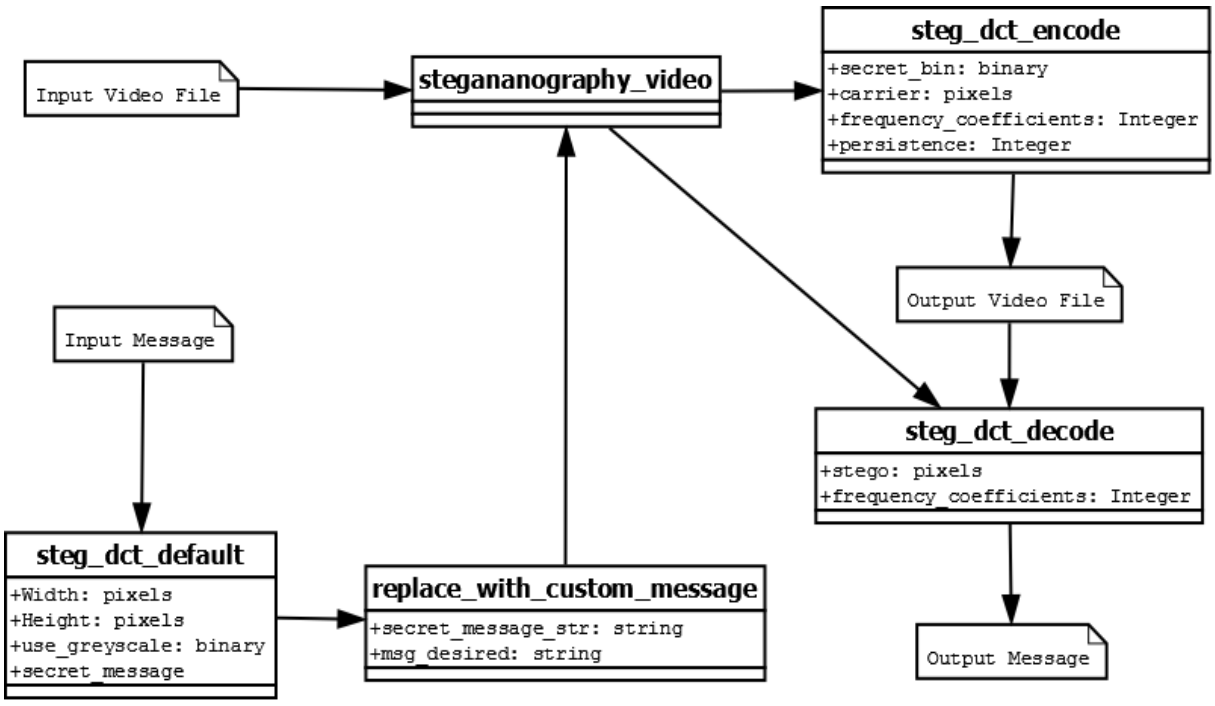


**Figure 10: System Diagram for Steganography Video Implementation**

The above System diagram in Figure 10 represents the main functions that encode the given secret message into the given input video file.

## FUNCTION FOR VIDEO STEGANOGRAPHY

The program beings with this function which acts like the main function in this MATLAB program, this function is the main program that in run and this is the function which calls all the other functions in this system design.

It contains all the necessary conditional statements to collect the data to be processed and call the respective functions to either encode or decode the given input text as per the algorithm specified by the user.

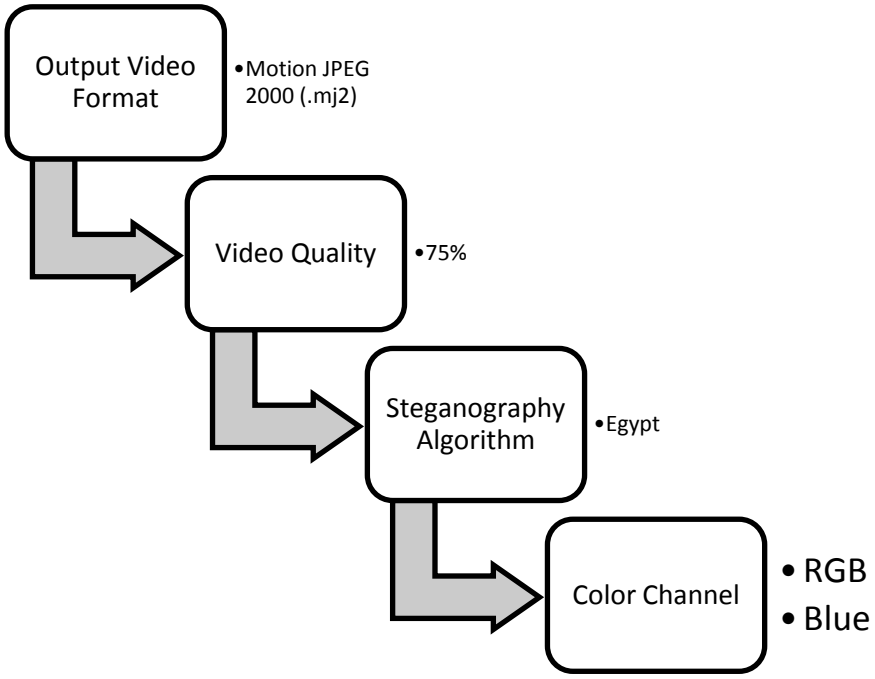


**Figure 11: Description of Video Steganography Function [8]**

## FUNCTION FOR DEFAULT DCT STEGANOGRAPHY ALGORITHM

This function is specific to the DCT algorithm and can be customized to the current algorithm which the user has specified to execute in the main steganography video function.

Here in this function the main constraint is introduced with the message length that can be encoded in the image is calculated.

It depends on the width and height of the image frame sent the algorithm

Thus, it is observed from Figure 11 that one bit is encoded in every 8x8 pixel block of the JPEG image frame and this function the secret message is converted from a string to a binary format.

$$Maximum\ Message\ Length\ = width\ / 8\ *\ height\ / 8 \quad \textbf{(3)}$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 52 | 66 | 74 | 74 | 89 | 89 | 89 | 90 |
| 34 | 67 | 75 | 75 | 90 | 90 | 90 | 91 |
| 54 | 68 | 76 | 76 | 91 | 91 | 91 | 92 |
| 66 | 69 | 77 | 77 | 92 | 92 | 92 | 93 |
| 44 | 70 | 78 | 78 | 93 | 93 | 93 | 94 |
| 66 | 71 | 79 | 79 | 94 | 94 | 94 | 95 |
| 77 | 72 | 80 | 80 | 95 | 95 | 95 | 96 |
| 88 | 73 | 81 | 81 | 96 | 96 | 96 | 97 |

**Figure 12: Picture Pixel Block (8x8 Matrix)**

**FUNCTION FOR CUSTOM MESSAGE**

This function is used the process the input secret message so that the custom message that is input into the previous function is of the correct length so that it can processed by the encoder function.

This is important as each image block can store only 1 bit of information.

$$Secret\ Message\ String\ =\ 1\ to\ Maximum\ Message\ Length \quad \textbf{(4)}$$

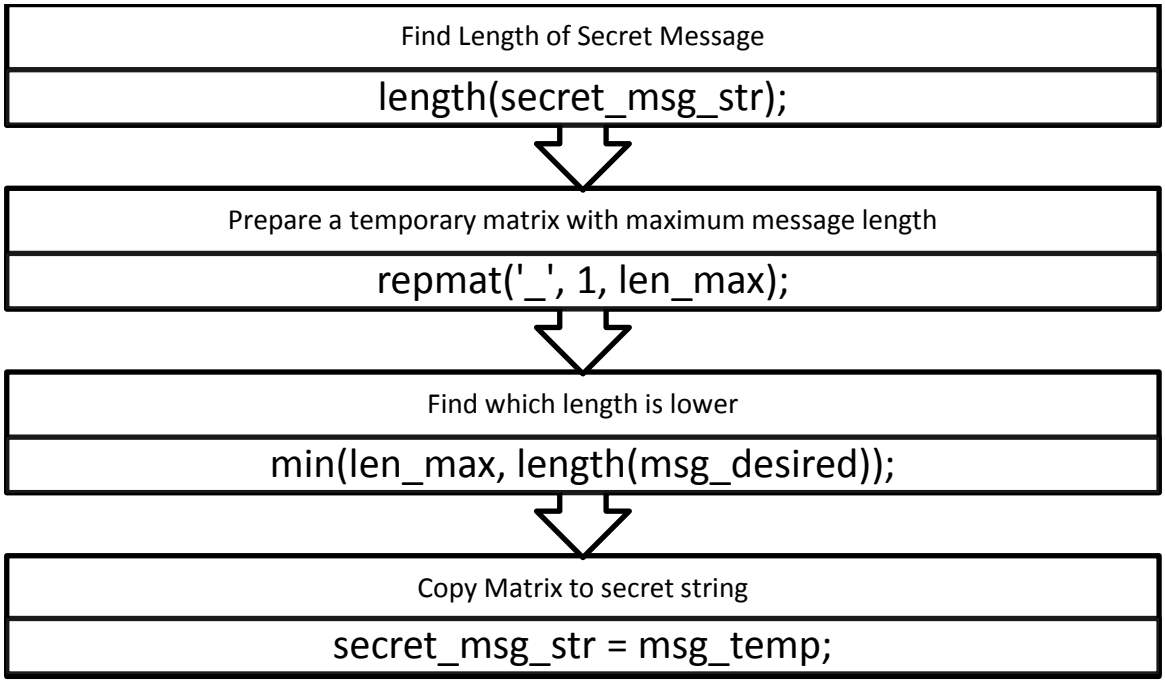


**Figure 13: Replace with Custom Message Function [8]**

**FUNCTION FOR STEGANOGRAPHY DCT ENCODING**

This is the main processing function for the DCT Steganographic algorithm and in the first step a block (8x8) pixels is extracted from the image frame and DCT is applied to the same.

After this operation, the secret bit is inserted into one of the blocks and then the modified blocks are copied back into the blocks and an inverse DCT operation is applied to get back the image frame with the secret message bit encoded.

$$Block\ =\ Discrete\ Cosine\ Transform\ (Matrix\ of\ Image\ Pixels) \quad \textbf{(5)}$$

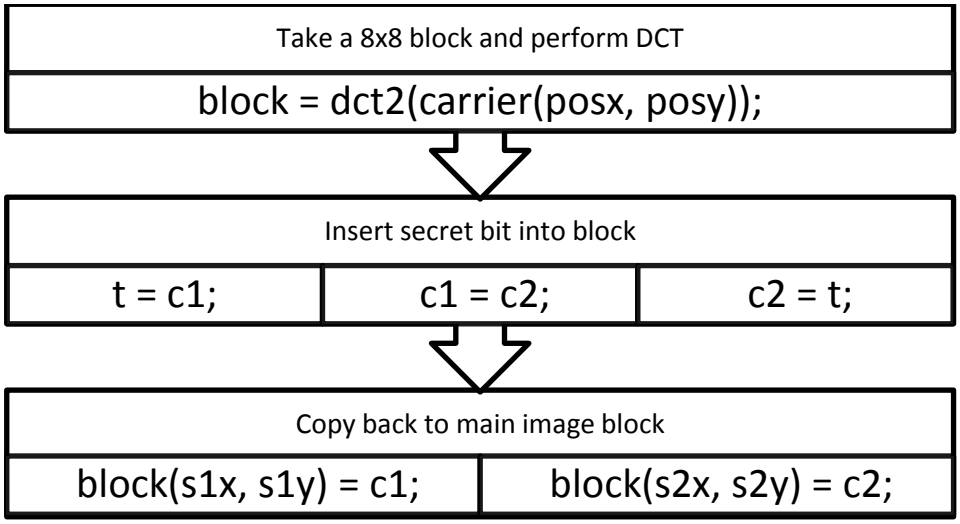


**Figure 14: Insert secret bit in DCT operation [8]**

$$Block = Inverse\ Discrete\ Cosine\ Transform\ (Matrix\ of\ Image\ Pixels) \quad \textbf{(6)}$$

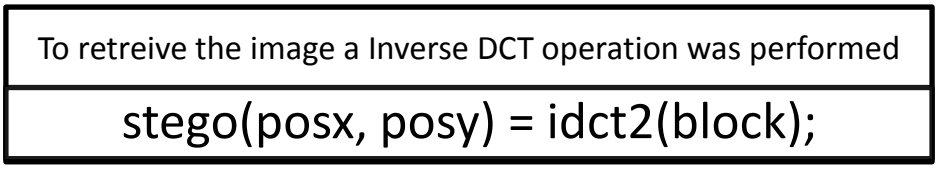


**Figure 15: Inverse DCT operation to retrieve image [8]**

**FUNCTION FOR STEGANOGRAPHY DCT DECODING**

This function is like the encode function and again a block (8x8) of pixels is extracted from the data and DCT is performed on this block to retrieve the same recorded values.

Thus, in this function the original encoded data can be retrieved and in turn the secret message which is the main objective of this project.

$$Block\ =\ Discrete\ Cosine\ Transform\ (Matrix\ of\ Image\ Pixels) \quad \textbf{(7)}$$

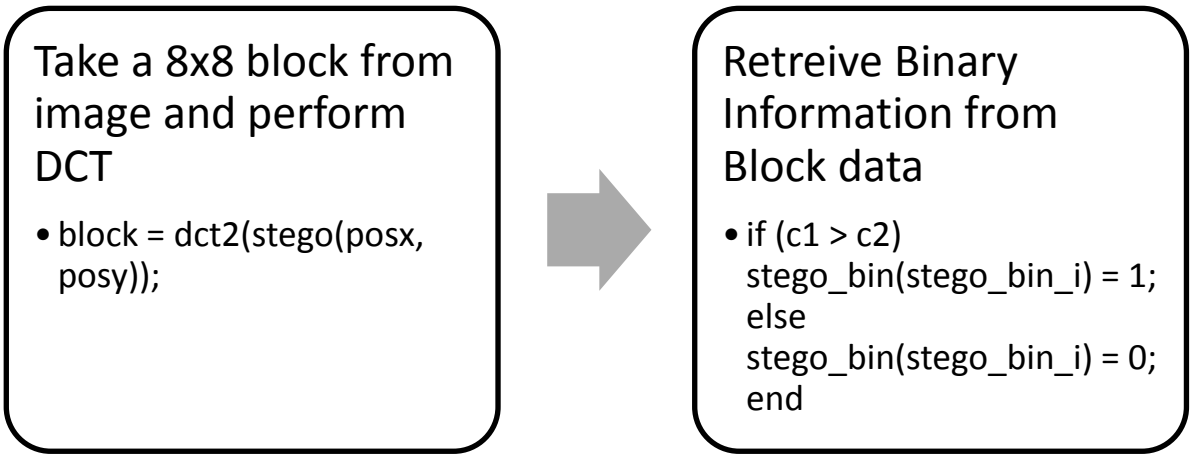


**Figure 16: Function for DCT decoding to retrieve secret message [8]**

Thus, the formal design and architecture of the project to be implemented is illustrated in the above section.

### 2.3. Hardware Requirements

For the basic demo of the Steganographic algorithms a basic laptop with MATLAB 2014 or above is needed.

But for a more advanced setup which can process a large quantity of video files in a short amount time and more professional setup with Linux Clusters as the main hardware platform is more suitable.

MATLAB has the parallel computing toolbox which is designed for performing MATLAB operations on a parallel scalable manner.

Some of the basic hardware requirements to operate this Linux Cluster are

- **Operating Systems**

  MATLAB performance is similar on Windows®, Mac OS® X, and Linux®, although differences can occur among platforms for the following reasons, the different compilers used in to build Mathworks products can result in different performance characteristics. There are various third party software developed for Mathworks which may have different configurations and running properties.

  The basic differences in the design of the operating systems result in different running characteristics.

  In general, performance differences in operating system releases (for example, between Windows 7 and Windows 8) are negligible.

- **Hardware Considerations**

  Each component of a typical computer configuration has an impact on MATLAB performance.

  - **Central Processing Unit (CPU)**

    Computers with **more CPU cores** generally outperform those with lower CPU cores. MATLAB automatically uses the natural parallelism of mathematical problems. But not all MATLAB functions are built or parallelism hence the speed up varies for different types of algorithms. The Parallel Computing Toolbox offers more in terms of functionality.

    MATLAB performance is dependent on the presence of floating-point hardware and this depends on the CPU core design structure in which some processors have independent Floating Point Units and in other designs a single FPU is shared with multiple cores, this severely affects performance.

    **Virtual cores** may provide some small performance gains but overall the performance will remain static. Intel CPUs with hyper-threading give the impression that the CPU has double the number of CPU cores. When using a tool such as Windows Task Manager, MATLAB may appear to be using only half the cores whereas the other half is being used for hyperthreading.

  - **Memory**

    MATLAB uses the computer memory system extensively so if the available physical memory runs out it starts using the virtual memory which results in thrashing. If MATLAB is not using the CPU fully, it might be an clear indication of thrashing. To detect thrashing on a Windows platform, use Windows Performance Monitor. On a Mac, use Activity Monitor.

    **MATLAB applications that use more than 3 GB of memory** (2 GB on some platforms) require the 64-bit version of MATLAB.

  - **Hard disk**

    **The hard disk speed is an important component of MATLAB start-up time**. Once MATLAB is running, disk speed is only important if file I/O is an important part of

the MATLAB application operations, or if your system is using virtual memory (see Memory section). For disk-intensive MATLAB applications a solid-state drive can be used to improve performance. Using a RAID array is also offers significant performance advantages.

- **Graphics Processing Unit (GPU) for display**
  MATLAB Graphics are rendered using OpenGL technology, so a graphics card with OpenGL support offers several advantages. It is a good practice to keep updated drivers for the graphics card to provide the best visual rendering.
- **Graphics Processing Unit (GPU) for computation**
  To **speed up computation**, Parallel Computing Toolbox leverages NVIDIA graphics cards with compute capability 2.0. See the compute capabilities of all NVIDIA graphics cards. MATLAB does not support accelerated computing using AMD or Intel GPUs now.
- **Benchmarking Your Program**
  MATLAB provides a **built-in benchmarking utility** called bench that provides a general overview of the performance of the MATLAB application, but it cannot reliably predict how any MATLAB application will run. Use the MATLAB functiontimeit to help produce reliable and repeatable performance benchmarks. Use gputimeit to benchmark GPU code [11].

### 2.4. Research Facility Users Analysis

No major research facility is required for this analysis, just a basic windows computer system with MATLAB 2014 installed and any user with MATLAB experience can execute the program.

In a more advanced system which can process video files at a faster rate for encoding or decoding the Steganographic data at a faster rate a Linux Cluster could be built for the specific purpose of processing the video files.

There is also need to include Python Language Interpreter for implementing the statistical data collection package.

### 2.5. Realization and Operational Description

In this section, the implementation and testing of our project software program is illustrated, the main program is a MATLAB file called steganography_video.m, in this file the parameters needed for running the various Steganographic operations, such as the output video format like Motion JPEG 2000 and output quality of the video file.

The steganography algorithm to be used like LSB, DCT or Egypt is also specified in this file which is used for testing the application and the colour channel to be used for encoding the data and colour scheme is specified in this file.

Before the encoding of the message data into the given video input file there is a need to specify an output image format as there are several options like AVI, MP4 and MJ2 and MJ2 format was chosen because of the minimal manipulation of the image frame into the video file.

- **Basic Setup Operations**

  1. Choose the input secret message to be encoded like

     | “Hi this is Deepak” |
     |---|

  2. Choose video file in MP4 format like “bunny.mp4”

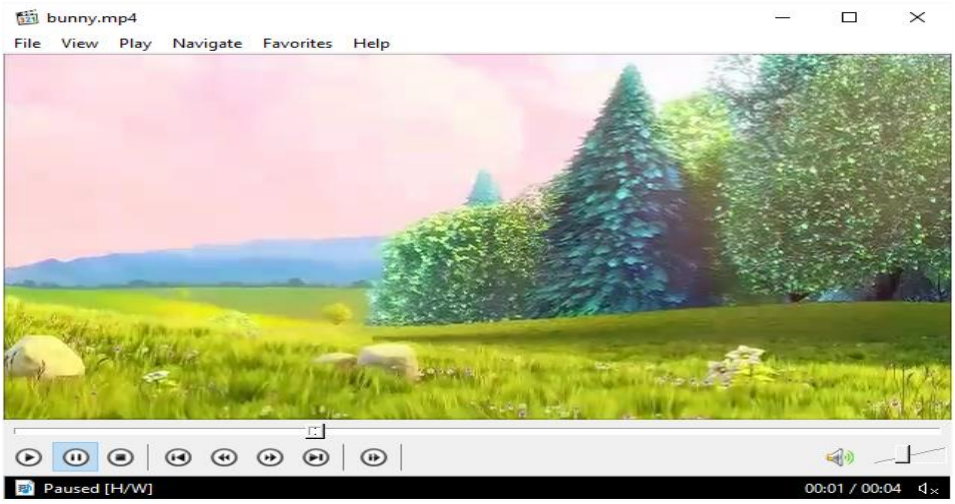


**Figure 17: Input Video File [19]**

3. Choose the right parameter's for encoding the message

```
%@@ Choose algorithm: LSB, DCT, ZK, WDCT, Fusion, Egypt
%@@ (not case sensitive)
algorithm = 'LSB';

%@@ Frames to use from the video
frame_start = 0;
frame_max = 10;

%@@ Which colour channel to use (1=r, 2=g, 3=b)
channel = 3;
```

**Source Code 1: steganography_video.m [8]**

4. After specifying the parameters, the MATLAB program is run to get the output video file with the secret message encoded into it and the PSNR data for each frame.

## 3. DNA STEGANOGRAPHY EXPERIMENTAL ANALYSIS

### 3.1. Experiment Planning

The basic experimental plan is to run the Steganographic algorithms like DCT, LSB and Egypt in a controlled software environment with the same input data and record the resulting data generated in this experiment like PSNR and percentage of message similarity to the original and derive the relevant conclusion from the data generated by the software [8].

After collecting the relevant PSNR data the mean average PSNR is evaluated for each algorithm, for e.g.

**13, 18, 13, 14, 13, 16, 14, 21, 13**

The mean is the usual average, so:

$(13 + 18 + 13 + 14 + 13 + 16 + 14 + 21 + 13) \div 9 = 15$

The experimental plan also includes a comprehensive test plan to ensure the software integrity of each Steganographic algorithm implementation.

In the test plan, the default message for each Steganographic algorithm is entered and decoded in such a way that the output of each algorithm is decipherable to some extent, if the decoded output of the Steganographic algorithm is too garbled we isolate that algorithm for further development and testing and not include the test data in our results.

For DNA Steganography, there is a requirement to calculate the Mean Squared Error based on the experimental data collected from which the PSNR value is evaluated.

$$\boldsymbol{PSNR = 10 \cdot log_{10}\left(\frac{MAX_I^2}{MSE}\right)} \textbf{ (7), [16]}$$

In this equation, the $MAX_I = 2^B-1$ where B is Bits per Sample.

$$\boldsymbol{MSE = \sum_{n}^{1}(Y' - Y)^2} \textbf{ (8), [16]}$$

### 3.2. Test Pattern Data and Results

After some testing with the input data mentioned in the previous section the following data patterns were recorded.

**Table1: Egypt Algorithm Test Data**

| Frame No | Similarity (%) | PSNR (dB) | Encode Time (s) | Decode Time (s) |
|---|---|---|---|---|
| 1 | 97.222 | 46.793 | 11.381 | 0.144 |
| 2 | 93.055 | 46.987 | 12.107 | 0.140 |
| 3 | 94.444 | 46.429 | 11.753 | 0.201 |
| 4 | 95.833 | 46.583 | 11.872 | 0.142 |
| 5 | 97.222 | 46.728 | 11.536 | 0.179 |
| 6 | 95.833 | 46.760 | 11.413 | 0.161 |
| 7 | 94.444 | 46.733 | 12.567 | 0.161 |
| 8 | 98.611 | 46.882 | 15.032 | 0.173 |
| 9 | 98.611 | 46.858 | 11.723 | 0.145 |
| 10 | 95.833 | 46.893 | 12.304 | 0.141 |

**Table2: DCT Algorithm Test Data**

| Frame No | Similarity (%) | PSNR (dB) | Encode Time (s) | Decode Time (s) |
|---|---|---|---|---|
| 1 | 71.888 | 33.218 | 3.606 | 1.443 |
| 2 | 72.333 | 33.284 | 3.416 | 1.438 |
| 3 | 73.888 | 33.359 | 3.262 | 1.662 |
| 4 | 74.166 | 33.383 | 3.864 | 1.398 |
| 5 | 74.222 | 33.371 | 3.722 | 1.435 |
| 6 | 74.166 | 33.423 | 3.746 | 2.037 |
| 7 | 74.555 | 33.442 | 4.426 | 1.692 |
| 8 | 74.611 | 33.429 | 3.500 | 2.012 |
| 9 | 74.888 | 33.456 | 3.359 | 1.459 |
| 10 | 74.944 | 33.425 | 3.145 | 1.835 |

### 3.3. Experimental Results

When the experiment was run, some of the results collected coincided with the theoretical expectations which were based on the various advantages and disadvantages of each Steganographic algorithm.

The PSNR data for DNA Steganography is calculated from existing Table 6 extracted from the research paper [14].

For example, when the assumed value of Y' is predicted value 1 then the MSE for 8 values is **0.3122** given the BPN values from Table 6.

From this MSE value we can calculate PSNR for DNA Steganography by if the $Max_I$ for the PSNR formula 7 is $2^1$-1 = 1.

Thus

$$10 \cdot log_{10}\left(\frac{1^2}{0.3122}\right) = \mathbf{360.96}$$

So, the PSNR calculated for DNA Steganography is **360.96.**

Since it is important to show the similarity of message recovery while decoding the various Steganography algorithms being tested, there is a need to calculate the similarity of messages in the DNA Steganography experiment from Table 6.

The Similarity percentage derived from Table 6 is shown in Table 3 below.

**Table 3: Similarity for DNA Steganography**

| Theoretical Data Limit | Actual Bits Encoded | Similarity (%) |
|---|---|---|
| 1 | 0.8 | 80 |
| 1 | 1 | 100 |
| 1 | 0.78 | 78 |
| 1 | 0.77 | 77 |
| 1 | 0.8 | 80 |
| 1 | 0.84 | 84 |
| 1 | 0.82 | 82 |
| 1 | 0.73 | 73 |

Calculating the average similarities for each algorithm the following values are calculated **Egypt: 96.111%, DCT: 73.966% and DNA Steganography: 81.75%.**

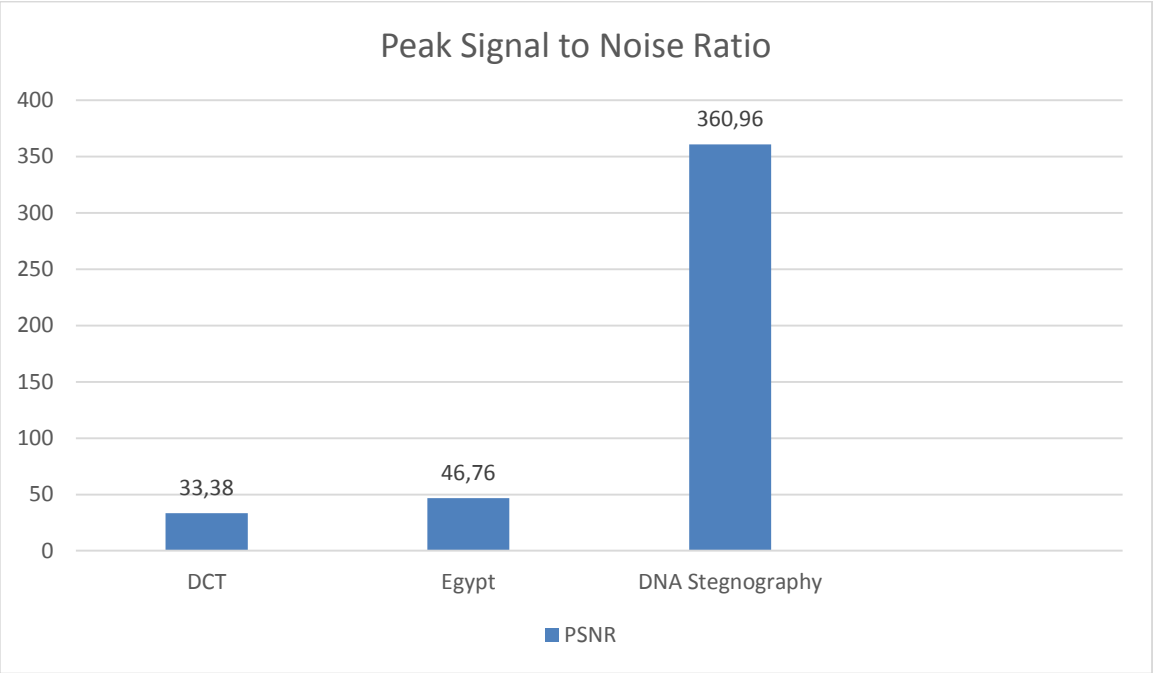


**Figure 18: Experimental Results**

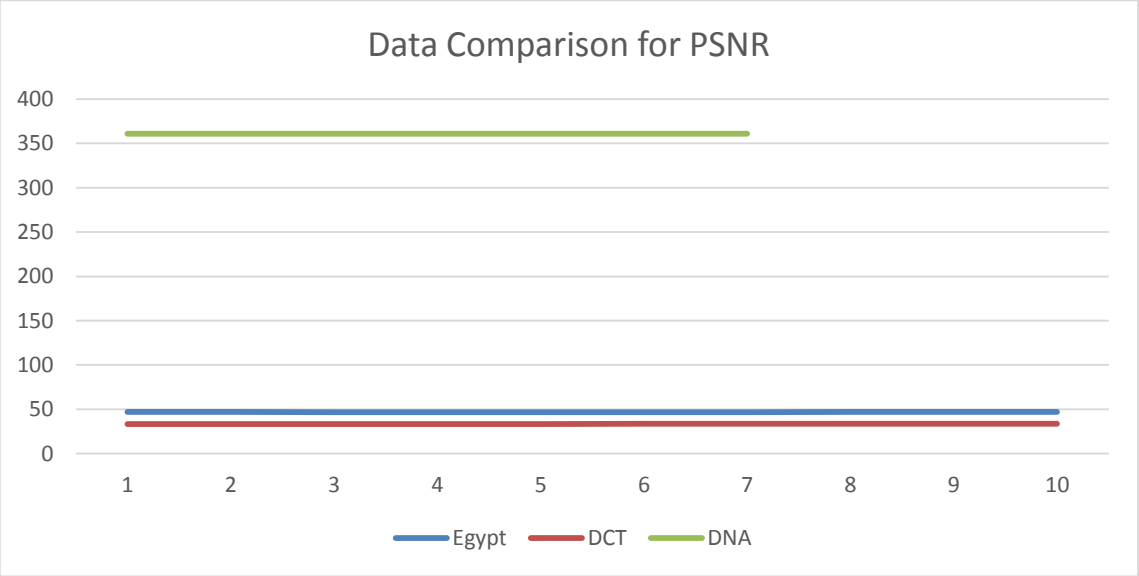


**Figure 19: Data Comparison for PSNR**

As observed in Figure 20 the variation in PSNR values for each algorithm on different runtime execution is extremely low.

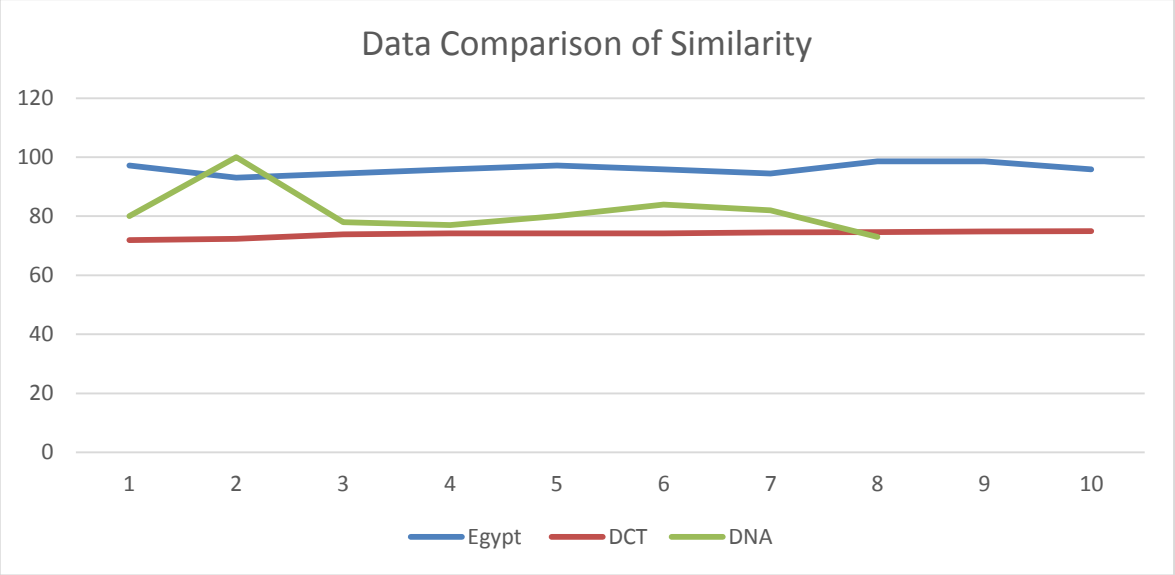


**Figure 20: Data Comparison of Similarity**

As observed from Figure 21 line graph the Similarity ratios are steady for Egypt and DCT but DNA Steganography seems to have some fluctuations depending on the runtime execution parameters.

### 3.4. Decision Operation and Analysis of properties and Quality Criteria for Rating

Least Significant Bit Steganographic algorithm is the least computationally expensive as it just involves inserting some extra bits of information in already defined data structure of the video codec, Discrete Cosine Transform is a relatively moderate image compression algorithm which can store extra Steganographic data in the signal data, it is four times faster than compressing the image using fast Fourier transforms, the main advantage being DCT uses the real number space to store the data [5].

The Egypt algorithm uses both spatial and temporal domains to store the information, hence the ability to store more data for the same amount of video information as DCT.

The DNA Steganography algorithm makes use of the best spatial data compression and since DNA sequences can be as large as DNA strands available it can be used to store and transmit large amounts of data.

## 4. SUMMARY AND CONCLUSIONS OF STEGANOGRAPHIC ALGORITHM ANALYSIS

- DNA Steganography is a more robust steganography algorithm with high PSNR value as compared to DCT and Egypt Steganography Algorithms.
- When the Similarity of the decoded message is compared to the original text it seems the Egypt algorithm is still holding the first place with a Similarity percentage of **96.111% as** compared to DNA Steganography which has a Similarity percentage of **81.75%.**
- Many practical DNA Steganography Applications are still in the research stage so the full extent of the DNA Steganography has not been explored [15].
- There is scope for combining DNA Steganography with other steganographic techniques like Image or Video Steganography to provide a further level obfuscation against an attacker.
- DNA Steganography has a lot of potential applications especially in Key Exchange systems where it can effectively replace Diffie-Hellman algorithm.
- Thus, in conclusion the initial aim of this research paper to evaluate the performance of each steganography algorithm like DCT, Egypt and DNA Steganography in practical conditions and in turn create a statistical basis to differentiate the performance of each Steganographic algorithm has been achieved.

## REFERENCES


[1] Almohammad, A., & Ghinea, G. (2010). Stego image quality and the reliability of PSNR. *2010 2nd International Conference on Image Processing Theory, Tools and Applications, IPTA 2010*, (1), 215–220. http://doi.org/10.1109/IPTA.2010.5586786

[2] Anderson, R. J., & Petitcolas, F. A. P. (1998). On the limits of steganography. *IEEE Journal on Selected Areas in Communications*, *16*(4), 474–481. http://doi.org/10.1109/49.668971

[3] Anshel, M., & Boklan, K. (2007). Review of "Introduction to Criptography with Coding Theory, 2nd Edition" by Wade Trappe and Lawrence Washington. *The Mathematical Intelligencer*, *29*(3), 66–69.

[4] Bhattacharya, S., Chattopadhyay, T., & Pal, A. P. A. (2006). A Survey on Different Video Watermarking Techniques and Comparative Analysis with Reference to H.264/AVC. *2006 IEEE International Symposium on Consumer Electronics*, 0–5. http://doi.org/10.1109/ISCE.2006.1689458

[5] Chen, P., & Lin, H. (2006). A DWT Based Approach for Image Steganography. *International Journal ofApplied Science and Engineering*, *Vol. 4*(No. 3), 275–290.

[6] Chen, W.-H., Smith, C. H., & Fralick, S. C. (1977). A Fast Computational Algorithm for the Discrete Cosine Transform. *IEEE Transactions on Communications*, *C*(9), 1004–1009. http://doi.org/10.1109/TCOM.1977.1093941

[7] Cole, E. (2003). *Hiding in Plain Sight : Steganography and the Art of*. *America*. http://doi.org/10.1016/j.tree.2005.05.010

[8] Hooke, S. (2016). Steganography. Retrieved from https://github.com/marbsydo/Steganography

[9] Kalker, T., Depovere, G., Haitsma, J., & Maes, M. (n.d.). A Video Watermarking System for Broadcast Monitoring, 31–40.

[10] Rhoads, G. (2000). Video steganography. *US Patent 6,026,193*, *00*(c), 0–3. http://doi.org/10.1109/PERVASIVE.2015.7087159

[11] System Requirements for MATLAB. (2016). Retrieved June 1, 2016, from http://www.mathworks.com/products/MATLAB/choosing_hardware.html

[12] Zlii, L. (2003). LSB Steganography Detection Algorithm, 2780–2783.

[13] M. Reza, N. Torkaman, P. Nikfard, N. S. Kazazi, M. R. Abbasy, and S. F. Tabatabaiee, "Improving Hybrid Cryptosystems with DNA Steganography," pp. 42–52, 2011.

[14] H. J. Shiu, K. L. Ng, J. F. Fang, R. C. T. Lee, and C. H. Huang, "Data hiding methods based upon DNA sequences," *Inf. Sci. (Ny).*, vol. 180, no. 11, pp. 2196–2208, 2010.

[15] Y. Erlich and D. Zielinski, "DNA Fountain enables a robust and efficient storage architecture."

[16] D. Salomon, G. Motta, and D. Bryant, *Data Compression: The Complete Reference*. Springer London, 2007.

[17] European Bioinformatics Institute DNA Database. (n.d.). Retrieved April 10, 2017, from http://www.ebi.ac.uk/services/dna-rna

[18] Johnson, N. F., & Jajodia, S. (1998). Exploring steganography: Seeing the unseen. IEEE Computer, 31(2), 26–34. http://doi.org/10.1109/MC.1998.4655281

[19] Goedegebure, S., & Goralczyk, A. (2017). Open Source Video. Retrieved from https://peach.blender.org/download/

## ANNEXES

Additional information and documents are not included in the main document. Accessories are optional, they are only given if needed.

### 1. Annex. Test Data

After some testing with the input data mentioned in the previous section the following data patterns were recorded.

**Table 4: Egypt Algorithm Test Data**

| Frame No | Similarity (%) | PSNR (dB) | Encode Time (s) | Decode Time (s) |
|---|---|---|---|---|
| 1 | 97.222 | 46.793 | 11.381 | 0.144 |
| 2 | 93.055 | 46.987 | 12.107 | 0.140 |
| 3 | 94.444 | 46.429 | 11.753 | 0.201 |
| 4 | 95.833 | 46.583 | 11.872 | 0.142 |
| 5 | 97.222 | 46.728 | 11.536 | 0.179 |
| 6 | 95.833 | 46.760 | 11.413 | 0.161 |
| 7 | 94.444 | 46.733 | 12.567 | 0.161 |
| 8 | 98.611 | 46.882 | 15.032 | 0.173 |
| 9 | 98.611 | 46.858 | 11.723 | 0.145 |
| 10 | 95.833 | 46.893 | 12.304 | 0.141 |

**Table 5: DCT Algorithm Test Data**

| Frame No | Similarity MATLAB (%) | PSNR (dB) | Encode Time (s) | Decode Time (s) |
|---|---|---|---|---|
| 1 | 71.888 | 33.218 | 3.606 | 1.443 |
| 2 | 72.333 | 33.284 | 3.416 | 1.438 |
| 3 | 73.888 | 33.359 | 3.262 | 1.662 |
| 4 | 74.166 | 33.383 | 3.864 | 1.398 |
| 5 | 74.222 | 33.371 | 3.722 | 1.435 |
| 6 | 74.166 | 33.423 | 3.746 | 2.037 |
| 7 | 74.555 | 33.442 | 4.426 | 1.692 |
| 8 | 74.611 | 33.429 | 3.500 | 2.012 |
| 9 | 74.888 | 33.456 | 3.359 | 1.459 |
| 10 | 74.944 | 33.425 | 3.145 | 1.835 |

The data represented in the table 6 and table 7 are taken from the research paper [14] and is based on the software testing of DNA Steganography algorithm presented in the paper.

**Table 6: DNA Steganography Test Data[14]**

| Sequence | Number of nucleotides | *capacity C* | *payload P* | $bpn = \frac{\|M\|}{C}$ |
|---|---|---|---|---|
| AC153526 | 200,117 | 200,117 | 0 | 0.80 |
| AC166252 | 149,884 | 149,884 | 0 | 1.00 |
| AC167221 | 204,841 | 204,841 | 0 | 0.78 |
| AC168874 | 206,488 | 206,488 | 0 | 0.77 |
| AC168897 | 200,203 | 200,203 | 0 | 0.80 |
| AC168901 | 191,456 | 191,456 | 0 | 0.84 |
| AC168907 | 194,226 | 194,226 | 0 | 0.82 |
| AC168908 | 218,028 | 218,028 | 0 | 0.73 |

**Table 7: DNA Stegnography Average BPN [14]**

| Method | Insertion | Complementary Pair | Substitution |
|---|---|---|---|
| Average *bpn* | 0.58 | 0.07 | 0.82 |

## 2. Annex. User Guide

The following steps enable the testing of DCT and Egypt steganography algorithms using MATLAB 2014.
Please adapt the same for any other steganography algorithm which is to be tested.

- Step 1
  - Open MATLAB
  - Open the steganography_video.m file
  - Change the following parameters
    - Output Video Type: MJ2
    - Output Quality: 75
    - Algorithm to be Used: ‘DCT’
  - Frames to use from video
    - frame_start = 0
    - frame_max = 10
  - Colour Channel to Use
    - channel = 3
  - Colour Space
    - colourspace = 'rgb'
  - Specify Input Video File Name
    - input_video_filename = [dir_input, 'bunny.mp4']

- Step 2
    - Insert message to encode in the algorithm default function that was specified in the previous step
        - For e.g.
            - case 'dct'
                - [secret_msg_bin, frequency_coefficients, persistence] = steg_dct_default(width, height, use_greyscale, 'Hi This is Deepak')
- Step 3
    - Run the steganography_video.m file and the encoded video file and an excel spreadsheet is created with the relevant PSNR statistical data
        - bunny_dct MJ2k.mj2
        - DCT_video_results.xls
- Step 4
    - If the command window was observed we will get the following output for each frame of the video file.
        - Frame 8 message: "Hi txis is deepak_"
        - Encode time: 11.200704s
        - Decode time: 0.178715s
        - PSNR: 46.882986
        - Message similarity (MATLAB): ~98.61%

These are the basic user guide instructions for using the steganography MATLAB program.

## 3. Annex. Source Code

Source Code 2 represents the initial MATLAB file that is run and it basically establishes the input and output formats and calls the required functions to perform the encoding and decoding process.

```
clc;
clear variables;
[dir_input, dir_output, dir_results] = steganography_init();

%@@ Output video, format and compression
%@@ 1 = Archival         (.mj2)
%@@ 2 = Motion JPEG AVI   (.avi)
%@@ 3 = Motion JPEG 2000  (.mj2)
%@@ 4 = MPEG-4           (.mp4)
%@@ 5 = Uncompressed AVI  (.avi)
profile_type = 3;

%@@ Video quality
%@@ NOTE: Only applicable to Motion JPEG AVI and MPEG-4
output_quality = 75;

%@@ Choose algorithm: LSB, DCT, ZK, WDCT, Fusion, Egypt
%@@ (not case sensitive)
algorithm = 'Egypt';

%@@ Frames to use from the video
frame_start = 0;
frame_max = 10;

%@@ Which colour channel to use (1=r, 2=g, 3=b)
channel = 3;

%@@ Which colour space to use ('rgb', 'hsv', 'ycbcr');
colourspace = 'rgb';

%@@ Whether the video is greyscale
use_greyscale = false;

%@@ Name of folder to store test results in
test_name = [algorithm, '_video'];

[dir_results_full, ~] = create_directory_unique([dir_results, test_name]);
output_csv_filename = [dir_results_full, test_name, '_results.csv'];

% Encode
% ======

%@@ Input video and output video. File extension is not required for output
%@@ because it is generated based upon the chosen format
```

**Source Code 1: steg_dct_default [8]**

Source Code 3 represents the DCT encoding algorithm in MATLAB which encodes the binary secret message within the transform domain, this operation is performed in 8x8 block by block mode with the 8x8 block representing the matrix of picture pixel values.

```
function [stego, bits_written, bits_unused] = steg_encode_dct(secret_bin, carrier, frequency_coefficients, persistence)
% steg_encode_dct Encodes binary data within the transform domain
% INPUTS
%   secret      - Stream of binary data to hide.
%   carrier     - Image within which to hide data.
%   persistence  - A higher value makes the hidden data more persistent
%                  i.e. Can survive more compression. 25 is recommended.
% OUTPUTS
%   stego       - Steganographically altered image.
%   bits_written - Number of written bits. If the same length as secret,
%                  then the process hid all data.
%   bits_unused  - Number of  of unused bits. If bigger than zero, then
%                  more data could have fit.

secret_bin_i = 1;
secret_length = numel(secret_bin);

% Tally the written and unused bits
bits_written = 0;
bits_unused = 0;

% If the data is not long enough, this bit is written instead
insufficient_bit = 0;

% Dimension of block that image will be split into
block_width = 8;
block_height = 8;

% Location of s1 and s2 within each block
% Comparing these values determines the binary value
s1x = frequency_coefficients(1,1);
s1y = frequency_coefficients(1,2);
s2x = frequency_coefficients(2,1);
s2y = frequency_coefficients(2,2);

[width height] = size(carrier);
stego = zeros(width, height);

grid_width = width / block_width;
grid_height = height / block_height;

for gx = 1:grid_width
   for gy = 1:grid_height
      cx = (gx-1) * block_width + 1;
      cy = (gy-1) * block_width + 1;
```

**Source Code 2: steg_dct_encode [8]**

Source Code 4 represents the DCT decoding process which is similar to the encoding process with the Steganography image split into 8x8 blocks and the secret binary value is retrieved from the by applying DCT again in the reverse process.

```
function [secret] = steg_decode_dct(stego, frequency_coefficients)
% steg_decode_dct Retrieves data encoded with steg_encode_dct
% INPUTS
%   stego  - Steganographic image to retrieve data from
% OUTPUTS
%   secret - Retrieved binary data

% Dimension of block that image will be split into
block_width = 8;
block_height = 8;

% Location of s1 and s2 within each block
% Comparing these values determines the binary value
s1x = frequency_coefficients(1,1);
s1y = frequency_coefficients(1,2);
s2x = frequency_coefficients(2,1);
s2y = frequency_coefficients(2,2);

[width height rgb] = size(stego);

grid_width = width / block_width;
grid_height = height / block_height;

stego_bin = zeros(1, grid_width * grid_height);
stego_bin_i = 1;

for gx = 1:grid_width
   for gy = 1:grid_height

      cx = (gx-1) * block_width + 1;
      cy = (gy-1) * block_width + 1;

      posx = cx:cx+block_width-1;
      posy = cy:cy+block_height-1;

      % Take the block and perform DCT
      block = dct2(stego(posx, posy));

      c1 = block(s1x, s1y);
      c2 = block(s2x, s2y);

      if (c1 > c2)
         stego_bin(stego_bin_i) = 1;
      else
         stego_bin(stego_bin_i) = 0;
```

**Source Code 3: steg_dct_decode [8]**