



Inspiring Excellence

Final Thesis Report

Image Encryption and Decryption using Computer Generated Holography

Group Members

Md. Asif Istiaque

13101122

Md. Mainul Islam Mahi

14101183

Syed Aousaf Hasan

12201061

Supervisor:

Dr. Md. Ashraful Alam

Assistant Professor

**Department of Computer Science and Engineering
BRAC University, Dhaka, Bangladesh.**

Declaration

We hereby declare that this is an original report written by us with our own findings and has not been published or presented in parts or as a whole for any other previous degree. Resources and materials by other researchers used as guidelines for our research are duly mentioned in reference citations.

Signature of the Authors:

Signature of the Supervisor:

Md. Asif Istiaque

Dr. Md. Ashrafal Alam
Assistant Professor
Dept. of Computer Science and
Engineering
BRAC University

Md. Mainul Islam Mahi

Syed Aousaf Hasan

Acknowledgement

To start off, we would like to express our heartiest gratitude towards the Almighty Allah and His many blessings which is why we are healthy and alive. Secondly, we would like to share our sincere gratitude to our advisor Dr. Md. Ashraful Alam for his constant motivation, support, and immense knowledge to our research. It would not have been possible without his constant guidance in all phases of our research.

We are grateful to our parents from the heart who always support us day and night and help us in every step of the journey. And we appreciate our friends who supported and encouraged us along the way.

And finally, we extend our gratitude to BRAC University for giving us the opportunity and helping us with necessary resources to conduct this thesis.

Abstract

Holography is an optical technique to generate the interference pattern of the light field created from the interference of two coherent light sources, one being scattered from an object and another reference light being sent directly into the holographic plate. This interference pattern is then used in a 3D holographic display to view the holographic representation of the object. This is done by passing the same reference light through the interference pattern fringe in the 3D display. Computer Generated Holography (CGH) is a simulation technique to generate the same interference pattern without the object being present. Usually this is done by taking an image or a 3D representation of the object and using numerical techniques to simulate the interference pattern of the light field. CGH has many applications and one of them is image encryption and decryption. This paper describes how to generate a CGH of an image, how to overcome the general problems, namely the image overlapping problem during the reconstruction of the image from the interference pattern and how to encrypt an image and then decrypt it using the properties and principles of CGH.

Table of Contents

Declaration	i
Acknowledgement	ii
Abstract	iii
List of Figures	3
List of Tables	5
Chapter 1	
Introduction	6
Chapter 2	
Literature Review	8
2.1 Related Works.....	8
2.1.1 Fourier Transform.....	9
2.1.2 Discrete Fourier Transform.....	11
2.1.3 DFT in image processing.....	13
2.1.4 Fast Fourier Transform.....	13
2.2 Encryption.....	14
2.2.1 Image Encryption.....	16
Chapter 3	
Fundamentals Of CGH	17
3.1 In-line Holography.....	17
3.2 Off-line or Off-axis Holography.....	18
3.3 Digital Holography.....	20
3.4 Computer Generated Holography.....	21
3.5 Architecture of CGH.....	22

Chapter 4

Proposed Methodology	24
-----------------------------------	-----------

Chapter 5

Experimental Part.....	26
5.1 Solving the overlap problem	28
5.1.1 Part 1-Scaling the input image	31
5.1.2 Part 2-Reconstruction	32
5.2 Encryption and Decryption	34
5.2.1 Encryption	34
5.2.2 Decryption	37

Chapter 6

Results and Discussions	40
6.1 Solution to the overlapping problem	40
6.2 Encryption and Decryption	42
6.2.1 Encryption	42
6.2.2 Decryption	42
6.3 Discussion.....	45

Chapter 7

Conclusion	47
References	48

List of Figures

1(a): Recording of a Traditional Hologram	6
2.1.1(a): Time versus pressure wave.....	9
2.1.1(b): Rotating a wave around a circle	11
2.1.2(a): Discrete Fourier Transform results.....	12
2.2(a): Process of Encryption.....	15
2.2(b): Process of Decryption using the encryption key.....	15
3.1(a): In-line Fresnel Hologram Setup	17
3.2(a): Off-line Fresnel Hologram Setup	19
3.3(a): Digital Holography using CCD Camera	20
3.5(a): Flowchart of the architecture of computer generated holography	22
5.0(a): Flowchart of hologram generation and reconstruction.....	27
5.1(a): Input Image.....	28
5.1(b): Hologram	29
5.1(c): Reconstructed image	29
5.1(d). Input Image	29
5.1(e). Reconstructed Image with overlapping.....	30
5.1(f). Manually cropped input image	30
5.1(g). Reconstruction of manually cropped image.....	31
5.1.1(a): Flowchart of extending the matrix of input image	32
5.1.2(a): Flowchart of the reconstruction.....	33
5.1.2(b). Input Image	33
5.1.2(c). Reconstructed Image	33
5.2.1(a). Input Image	34
5.2.1(b): Flowchart for the encryption of input image	35
5.2.1(c). Input.....	36
5.2.1(d). Cipher Image	36
5.2.2(a): Flowchart for decryption of the image using the encryption key.....	37
5.2.2(b): Breakdown of the encryption key	38
5.2.2(c). Reconstructed Image	38

5.2.2(d). Reconstruction with wrong key	39
6.1(a) Input image.....	40
6.1(b) Input image	40
6.1(c) Input image.....	40
6.1(d) Reconstruction with overlapping	41
6.1(e) Reconstruction with overlapping	41
6.1(f) Reconstruction with overlapping	41
6.1(g) Reconstruction after solving the overlapping problem	41
6.1(h) Reconstruction after solving the overlapping problem	41
6.1(i) Reconstruction after solving the overlapping problem	41
6.2.1(a) Cipher images	42
6.2.1(b) Cipher images	42
6.2.1(c) Cipher images	42
6.2.2(a) Decryption with the correct Encryption key	43
6.2.2(b) Decryption with the correct Encryption key	43
6.2.2(c) Decryption with the correct Encryption key	43
6.2.2(d) Decryption with the wrong key	43
6.2.2(e) Decryption with the wrong key	43
6.2.2(f) Decryption with the wrong key.....	43
6.3(a) Magnitude Spectrum of the cipher image	45

List of Tables

6.1(a) Steps of the algorithm43

Chapter 1

Introduction

Holography is the art of presenting a 3D image on a 2D surface. Holograms are generated by recording the intensity and phase of a light. A photograph only records the intensity of light but a hologram records both the intensity and phase. For this reason, when it is printed on a holographic plate the object appears to be three dimensional. A traditional hologram is generated by recording the light scattered from an object when the object is illuminated from a coherent laser source. A reference light from the same source is then taken and when both the light waves are superimposed on each other then it produces a holographic pattern. Which then if projected on a holographic plate a 3D visualization of the object is generated.

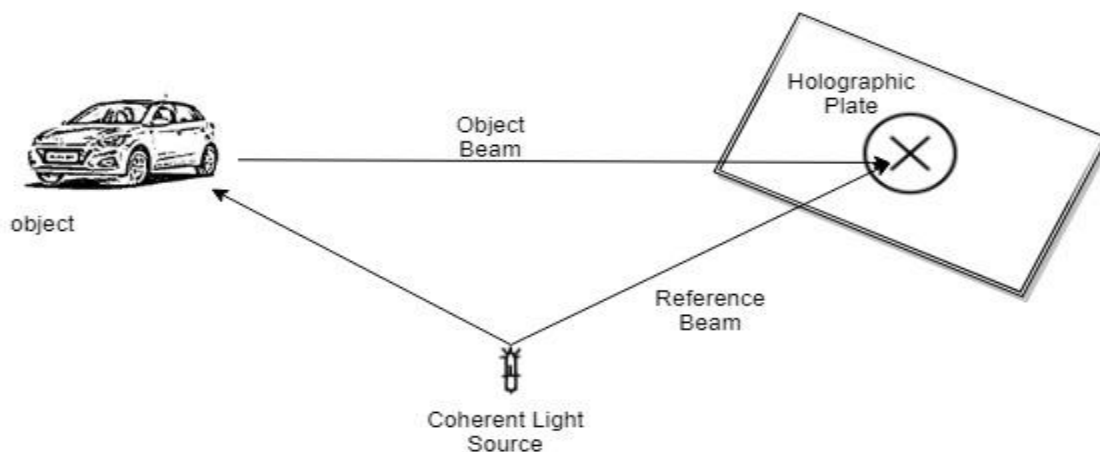


Figure 1(a): Recording of a Traditional Hologram

A computer-generated hologram which is popularly known as CGH is a bit different from traditional holography. For a digital hologram the presence of the real object is not necessary and the intensity of the coherent source is mathematically calculated. The interference pattern forms the two waves, which are then calculated by algorithms and thus a final holographic pattern is generated.

Holographic interference pattern can be used in image encryption. An image can be broken down and the spatial domain of the image can be converted into frequency domain using different algorithms. If the far field pattern of the image is printed then it cannot be decrypted

by anyone as it is nothing other than some scattering pattern. If the far field amplitude scattered from the object and the reference light can be calculated then the image can be perfectly reconstructed. In this case the reference wave can work as the encryption key in reconstruction of the image. Without the calculation of the reference light the image cannot be reconstructed.

So, by calculating the far field amplitude of the reference beam it can be added with the far field amplitude of the light scattered from the object to obtain the final image. But as the object is not physically present here the scattered light from the object is calculated by using mathematical models.

Chapter 2

Literature Review

2.1 Related Works

Computer Generated Holography is a diverse field and various works have been done in this field in various domains. Numerous papers have also been published about the utilities of CGH in various fields. The use of Fourier Transform to mathematically simulate the propagation of light was first proposed by Brown and Lohmann in a paper published in 1966 titled “Complex spatial filtering with binary masks” [1]. This numerical simulation technique created a breakthrough in CGH and various other studies onwards followed the same technique. The use of CGH in image encryption and decryption has been discussed in a paper by James Wu in 2007 titled “Computer Generated Hologram”. In this paper the use of Arnold Transformation was discussed to hide or embed an image into a host image in the Fourier Transform domain. In this process the hidden image becomes unrecognizable after a few iterations of the Arnold Transform and the resultant pixel array size was always smaller than the host image so it was easy to embed into the host image [7]. Another paper titled “Image Encryption Based on Interleaved Computer-Generated Holograms” discussed an encryption method where two CGH are encoded into one hologram by using interleaved patterns of the pixel columns. Other numerous works have been done in the field of image encryption using CGH but these are the prominent ones that we have used as our point of references [8].

The crucial part of our encryption algorithm consists of the use of Fourier Transform, namely Discrete Fourier Transform to convert the image from the spatial domain to the frequency domain and calculate the far field amplitude of the reference and object wave. A brief overview of Fourier Transform, Discrete Fourier Transform and general encryption algorithms is given below.

2.1.1 Fourier Transform

Fourier Transform is an important tool in mathematics that is used to decompose or separate a function or a wave into the components that make up the function. This is a high-level definition of Fourier Transform and not intuitive to understand. To understand Fourier Transform clearly we need to first realize the problem that it solves. Suppose we have a pure sound wave (A) which has 440 beats per second which means if we measure the air pressure as a function of time then the wave would make 440 oscillations each second. A lower pitch note like D has the same graph but with a different oscillation [9]. When these two sound waves are mixed the resultant wave becomes a complex mixture of these two waves that does not have any resemblance to the original waves.

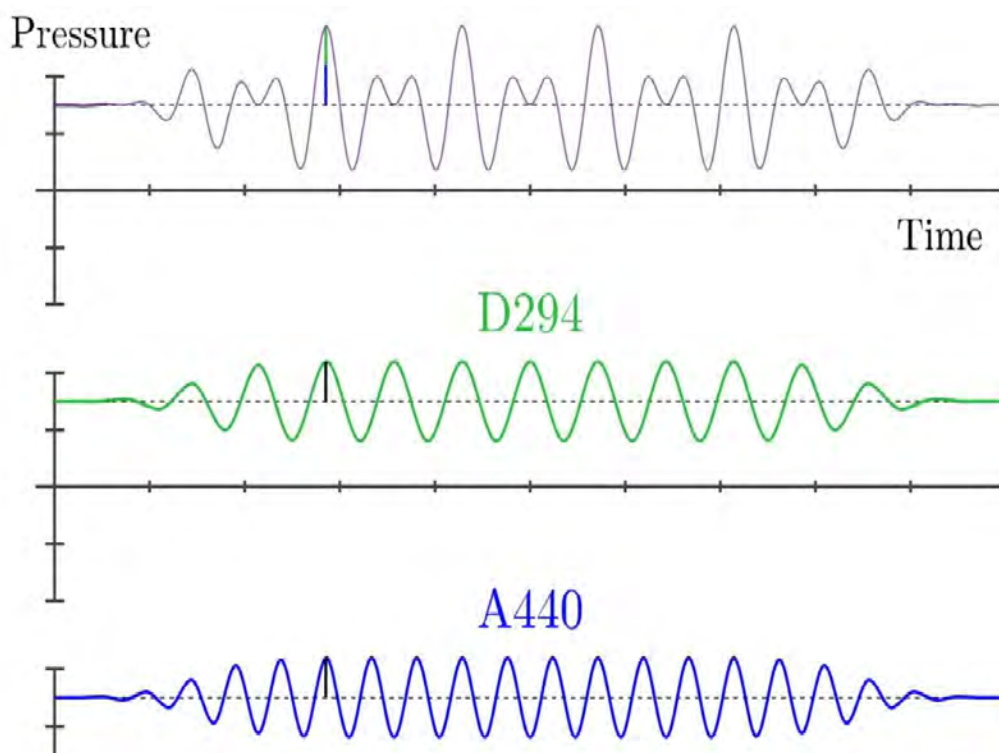


Figure 2.1.1(a): Time versus pressure wave

At any point in time the resultant wave forms from the interference of these two waves. The amplitude of the resultant wave is the highest when the peak of the two waves match and it is caused by constructive interference. The amplitude is lowest when the waves cancel out each other completely and it is caused by destructive interference. This wave can be made even more complex by adding one, two or even hundreds of pure waves and indeed the music that we hear today are actually a mixture of these waves [10]. But a piece of music comes with a

lot of unwanted noise and a practical problem is to filter out the unwanted noise from the music. One approach is to decompose the complex wave into all the waves that make it up and then keep only the waves that are needed. Fourier Transform helps us achieve that goal.

Mathematical Formula

The Fourier Transform of a function $f(x)$ looks like the following [9]:

$$F(k) = \int_{-\infty}^{+\infty} f(x)e^{-2\pi ikx} dk \quad (1)$$

Similarly, the inverse Fourier Transform is defined as follows [9]:

$$f(x) = \int_{-\infty}^{+\infty} F(k)e^{2\pi ikx} dk \quad (2)$$

Here,

$f(x)$ is the function in the time or spatial domain.

$F(k)$ is the function in the frequency domain.

e is the Euler's constant which approximately equals to 2.71828

i is the unit imaginary number.

An important part of the Fourier Transform consists of the Euler's formula. Euler's formula tells us that if you take e to the power of some number x times i then you would reach x amount of units counter clockwise of a circle of radius one. So $e^{2\pi i}$ would give us one full rotation of the circle [9]. The notion of Euler's formula and circular paths come into play because we need to think of the complex wave shapes as circular paths and try to wrap the wave shape around a circle.

From the formula the part $f(x)e^{-2\pi ikx}$ tells us that we are taking the original complex wave $f(x)$ and wrapping it around a circle. The figure below can be used to get an intuitive idea about wrapping a graph of a wave around a circle [9].

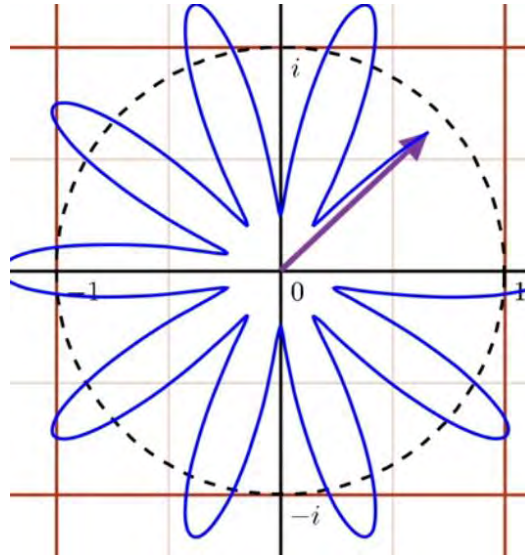


Figure 2.1.1(b): Rotating a wave around a circle

The rotating vector gets scaled up and down according to the value of the function $f(x)$ because of the multiplication. The negative sign ensures that the rotation happens clockwise instead of counter clockwise. Taking an integral of the whole functions gives us the center of mass of the wound up graph and this centre of mass spikes when the rotating frequency matches the actual frequency of one the waves that make up the complex wave. The limit positive infinity to negative infinity indicates that we are not taking a finite portion of the graph but considering all possible finite time intervals and the limit when that finite time interval reaches infinity [10].

The result of the Fourier Transform is a function if plotted will give spikes on the frequencies that make up the original function.

2.1.2 Discrete Fourier Transform

Discrete Fourier Transform is a variant of the Fourier Transform that is used on data that is discrete and periodic. For example, in digital signal processing the data is discrete and finite in length. The DFT converts an input signal of N points into an output signal of two points. The input signal can be in the time domain or the spatial domain. For an image the input signal will be in the spatial domain. The output will be in the frequency domain, the amplitudes of the sine and cosine waves.[11]

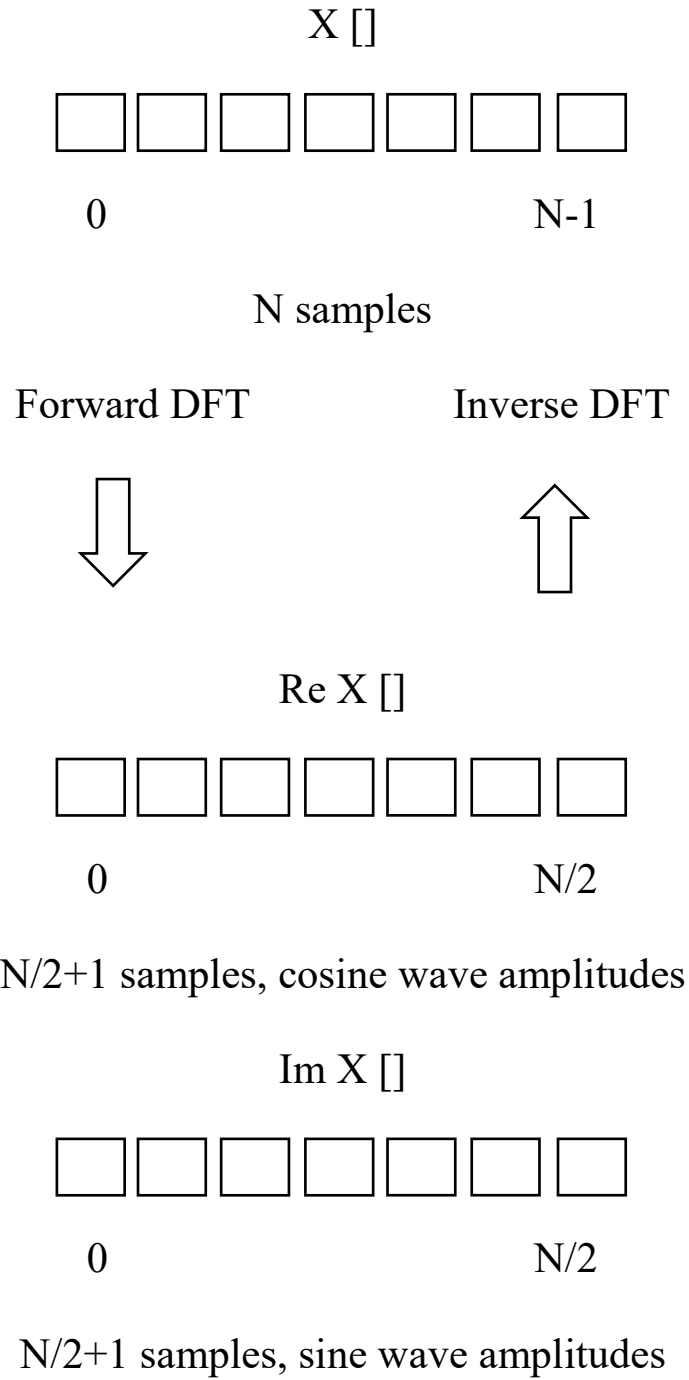


Figure 2.1.2(a): Discrete Fourier Transform Results

2.1.3 DFT in image processing

If we take a square image of size $\times N$, the two-dimensional DFT formula is [11]

$$F(k, l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) e^{-i2\pi(\frac{ki}{n} + \frac{lj}{n})} \quad (3)$$

Here $f(i, j)$ is the image in the spatial domain and $F(k, l)$ is each point in the Fourier domain.

The result Fourier image can be re constructed in the spatial domain by using the inverse Fourier Transform [11]

$$f(a, b) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} F(k, l) e^{i2\pi(\frac{ka}{n} + \frac{bj}{n})} \quad (4)$$

To save computational power the double sum calculation can be omitted and the DFT formula can be separated into two summations and then added back together to form the final DFT. Even with these savings DFT can be slow since it has N^2 time complexity. This can be reduced to $N \log_2 N$ by applying the Fast Fourier Transform [2]. Fast Fourier Transform is sometimes confused as another theoretical extension or application of the Fourier Transform but in reality, it is just a clever and faster way to calculate the values of the Discrete Fourier Transform. Details and applications of the Fast Fourier Transform have been discussed in the later parts of this paper.

2.1.4 Fast Fourier Transform (FFT):

Fast Fourier Transform of FFT is a way of calculating the Discrete Fourier Transform. Applying the Discrete Fourier Transform directly increases the computation time by $O(N^2)$ so a better way of calculation the Discrete Fourier Transformation was shown by J.W Cooley and John Turkey in their paper in 1965 and the algorithm is popularly known as the Cooley-Turkey algorithm. The Fast Fourier Transform (FFT) is able the compute the Discrete Fourier Transform (DFT) of an array of size N in time $O(N \log(N))$ [2]. Fast Fourier Transform

uses the divide and conquer strategy to recursively break down the DFT into some smaller DFT's. [9]

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i 2\pi k n/N} \quad (5)$$

Here let $eqn(1)$ be a Discrete Fourier Transform. It is then broken down into two terms which looks much similar to two Discrete Fourier Transforms [2].

$$= \sum_{m=0}^{\frac{N}{2}-1} x_{2m} \cdot e^{-i2\pi k (2m)/N} + \sum_{m=0}^{\frac{N}{2}-1} x_{2m+1} \cdot e^{-i 2\pi k (2m+1)/N} \quad (6)$$

Here in $eqn(6)$ the $eqn(5)$ has been split into two terms which looks like they are two smaller DFT's. One of the terms consist of the odd numbered values and the other consists of the even numbered values. Here each term is consisted of $(N/2) * N$ computations.

Now since the range of k is $0 < k < N$ and the range of n is $0 < n < M$ and that is equivalent to $N/2$ so we need to perform only half the computation for each sub problem. This process is applied recursively until the array is small enough that it is not beneficial anymore. Thus, the DFT can be computed very fast using this method [2].

$$= \sum_{m=0}^{\frac{N}{2}-1} x_{2m} \cdot e^{-i2\pi k m/(\frac{N}{2})} + e^{-i 2\pi k/N} \sum_{m=0}^{\frac{N}{2}-1} x_{2m+1} \cdot e^{-i 2\pi k m/(\frac{N}{2})} \quad (7)$$

2.2 Encryption

Encryption is the process of encoding data in such a way that it can only be visible to authorized person. The people who are not authorized will not be able to view the encrypted information. An encrypted data is only visible after it has been decrypted using a 'encryption key' which is generated by an algorithm. The data or information is called 'plaintext' before the encryption. The algorithm which has been used to encode the information is called 'cipher'. After information the encrypted data is called the 'ciphertext'. The ciphertext can only be read after the information has been decrypted using the encryption key [25].

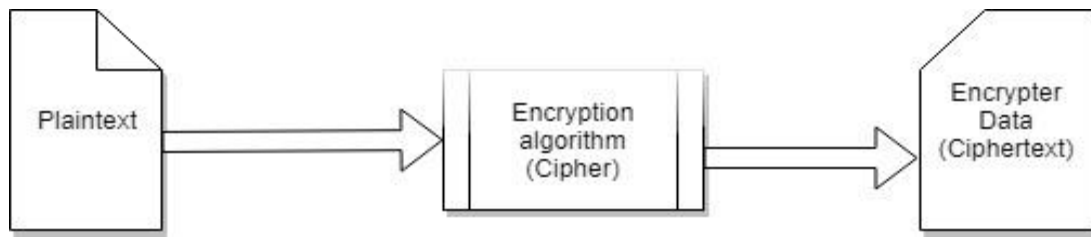


Figure 2.2(a): Process of Encryption

Encryption is one of the most important methods that provide security to information. There are two types of encryptions the symmetric-key ciphers and the public key cryptography.

In the symmetric-key ciphers there is only one key. The key is known as the secret key. This process uses only one key for the encryption of the data. When someone encrypts the information he/she provides a key or a key is generated for the encryption of that data. For the decryption that particular key needs to be used. For decryption the data the party who is encrypting the data must provide the key to the one who is decrypting the data.

The second process of encryption is the Asymmetric cryptography or in other words it is known as public key cryptography. The public key cryptography uses two keys but the two keys are mathematically linked. The two keys are the private key and the public key. The public key can be shared but the private key needs to be kept hidden. Both the private and the public key is needed to decrypt the information.

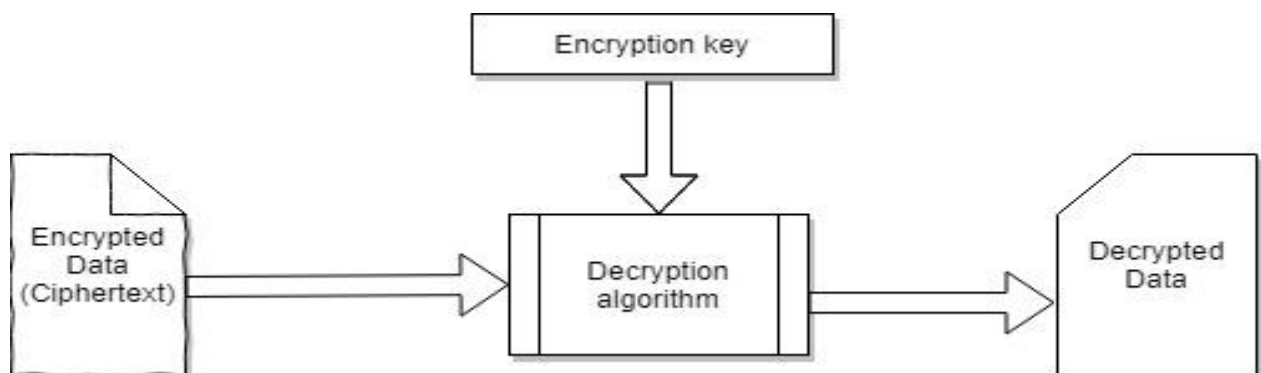


Figure 2.2(b): Process of Decryption using the encryption key

2.2.1 Image Encryption

Encryption is an integral part of almost every aspect of modern computing and information security. Image encryption is done on individual pixel values of an image. The pixels are scrambled or the values of the pixels are changed in such a way that the encrypted image or the ciphertext looks like a texture of some kind [24]. It has many real life practical utilities like embedding smaller images on a larger host image or using an encrypted image as a security stamp. There are many traditional algorithms that deal with image encryption and decryption in general. The traditional algorithms that deal with image encryption mostly treat images as matrices of pixels and use Gaussian elimination or similar matrix operations to alter the image states. The CGH method of image encryption and decryption on the other hand is totally different from the traditional algorithms. In the CGH way images are transformed from the spatial domain to the frequency domain and then the interference pattern of the object wave from the image and a reference wave is calculated [11]. The interference pattern is usually treated as the encryption of the image and only the original reference wave that was used to calculate the interference pattern can be used to reconstruct the original image from the pattern. So the reference wave can be treated as a key for decryption. But this process is way simpler because one plain wave with a point intensity of any pixel will be able to decrypt the image. So further encryption of the image is necessary for better security.

Chapter 3

Fundamentals of CGH

3.1 Inline Holography

In 1948, Dennis Gabor came up with “A new microscopic principle”, which he termed ‘holography’ [3]. In-line holography with spherical waves, as originally proposed by D. Gabor, is the simplest realization of the holographic method, working without lenses. Although its applications have been limited, however, until recently owing to the fact that reconstruction of the object image with another wave (light or electron) is not practical. In-line holography with electrons was brought back into scene when stable field emission tips that ended in a single atom, was created, eventually creating an intense point source for coherent electrons in the energy range from roughly 10 to 300 eV, i.e., for wavelengths from 2 to 0.5 Å [22].

Digital holographic microscopy (DHM) or Digital in-line holographic microscopy (DIHM) is different from other methods, because it does not record the projected image of the object. On the contrary, the information originating from the object in the form of light wave front is recorded in a digital method as a hologram. From this, the object image is calculated by a computer using a numerical algorithm [4].

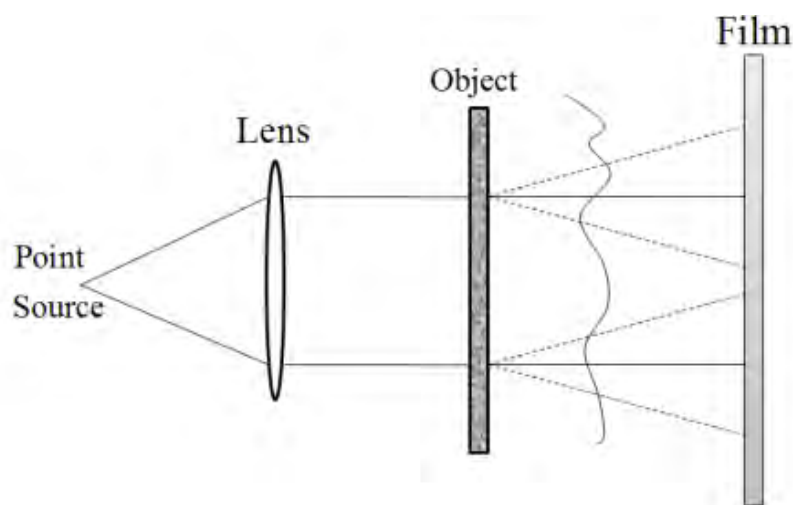


Figure 3.1(a): In-line Fresnel Hologram Setup

Above is a picture of In-line optical setup for writing Fresnel holograms. A point source of monochromatic light is collimated by a lens and the resulting collimated beam illuminates the semi-transparent object [12]. The film records the Fresnel diffraction pattern of the field emerging from the object. Similar to phase contrast microscopy, the light passing through a semi-transparent object consists of the scattered (U_1) and un-scattered field (U_0). At some distance z , which is behind the object, an intensity distribution generated by the interference of these two fields is recorded by the detector, [23]

$$\begin{aligned}
 I(x, y) &= |U_0 + U_1(x, y)|^2 \\
 &= |U_0|^2 + |U_1(x, y)|^2 + U_0 \cdot U_1^*(x, y) + U_0^* \cdot U_1(x, y) \quad (8)
 \end{aligned}$$

We assume a linear response to intensity associated with the photographic film, it is found that the transmission function has the form

$$t(x, y) = a + bI(x, y) \quad (9)$$

Where a , b are constants. The important information regarding the object is in the transmission function, t . This, in short, is the principle of writing an in-line Fresnel hologram [23].

3.2 Off-line or Off-axis holography

Leith and Upatnieks were the pioneers who worked with off-line holography. Off-axis or off-line electron holography is typically performed with highly elliptical illumination. Off-axis electron holography needs highly coherent illumination hence electron dose rates are typically low and exposure times are very similar to that of in-line holography [5]. The setup for off-axis holography encodes all spatial frequencies with equal strength.

Many of the shortcomings of in-line holograms have been overcome by going to an off-axis geometry that allows the various image components to be separated, and also allow opaque subjects to be front-illuminated.

The principal of off-line holography is described in short below:

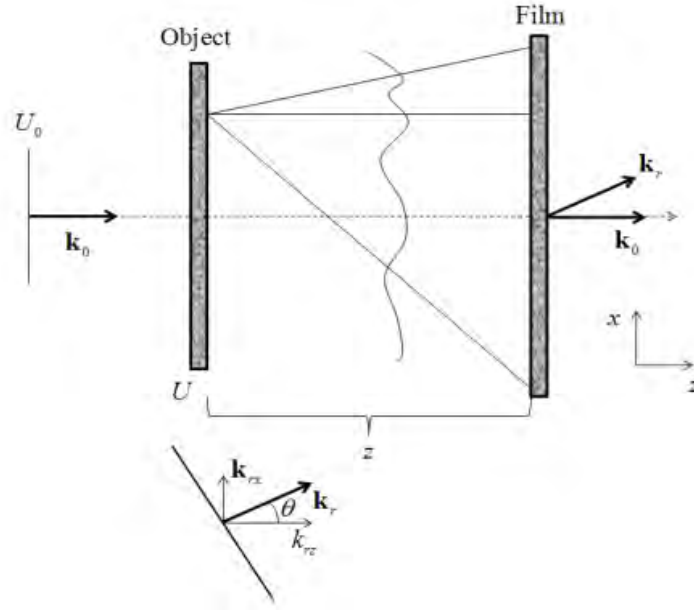


Figure 3.2(a): Off-line Hologram Setup

In the above figure, k_0 is the wave vector of the incident and k_r is the wave vector of the reference fields. Here, the object is illuminated by a monochromatic plane wave, U_0 , and the transmitted field reaches the photographic film at a distance z . The field distribution across the film, i.e. the Fresnel diffraction pattern, $U_F(x,y)$, is a convolution between the transmission function of the object, U , and the Fresnel diffraction kernel

$$U_F(x, y) = U(x, y) * e^{\frac{ik_0(x^2+y^2)}{2z}} \quad (10)$$

The reference field here, which is θ is the angle with respect to object beam.

Here, the reference field, U_r , is delivered at an angle θ (“off-axis”) with respect to the object beam. The total field at the film plane is

$$\begin{aligned} U_t(x, y) &= U_F(x, y) + |U_r|. e^{ik_r.r} \\ &= U_F(x, y) + |U_r|. e^{i(k_{rx}.x+k_{rz}.z)} \end{aligned} \quad (11)$$

The resulting transmission function related with the hologram is proportional to the intensity [23].

3.3 Digital Holography

By Digital Holography, we usually understand the acquisition and processing of holograms with a digital sensor array, normally a CCD camera or a similar device. Image rendering, construction or reconstruction of object data is performed numerically from digitized interferograms. Digital holography offers a means of measuring optical phase data and typically delivers three-dimensional surface or optical thickness images. There are a number of methods of recording and processing of digital holography such as, off-axis or off-line configuration, phase-shifting holography, frequency-shifting holography, multiplexing of holograms etc. Scientists registered holographic image in a digital camera which in turn feeds the information to a computer program. The numerical reconstruction process permits immediate quantitative access to intensity and phase. Topographic measurements accurate down to the nanometres scale and an immediate real-time follow-up surveillance of the sample evolution [13]. Nowadays the term ‘Digital Holography’ has become common in the optics field and is used to describe not only methods to build holographic images from physically recorded holograms but also processes used to construct holograms from scratch from virtual objects using a computer.

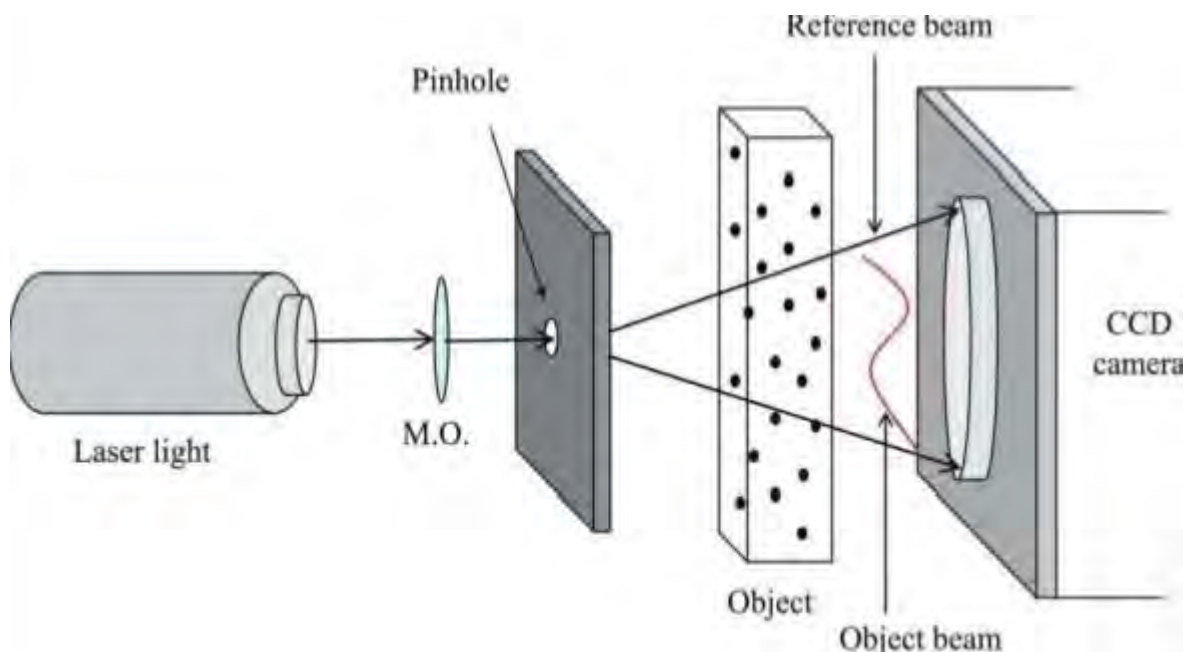


Figure 3.3(a): Digital Holography using CCD Camera

3.4 Computer Generated Holography

Unlike traditional holograms, the computer-generated holograms do not need a real object to be present. The object can be an image or a computer-generated design. The first developed computer-generated hologram was a binary hologram and it was developed by A. Lohmann [1]. He developed a technique by which a large image drawn by a computer output device made of only black and white parts were reduced photographically to reconstruct the hologram. A. Lohmann was the first to show that for constructing a hologram the presence of that object is not necessary. The object can be calculated mathematically and the output waves can be formed.

After that the process of digitally holography has undergone development and changes and now more advanced device and equipment are being used for developing a computer-generated hologram. The process is quite simple but as the real object is not present the object needs to be calculated in mathematical terms. For a computer-generated hologram we need two rays as the traditional holograms.

In traditional holograms a light from a point source is incident on the object and the diffraction is calculated and it is superimposed with the reference beam and the output beam gives the interference pattern for our result hologram. The diffraction from the object is calculated by mathematical terms. As we do not have any real object we need to have a formula for calculating the diffraction. For our experiment we have used Discrete Fourier Transformation for calculating the far field amplitude for the object beam. But as the Discrete Fourier Transformation is a bit slow we have used an improved fast version of the Discrete Fourier Transformation which is called Fast Fourier Transformation or in other words we call it FFT [2]. The Discrete Fourier Transformation goes through a Cooley - Turkey algorithm and thus gives the Fast Fourier Transformation (FFT). The Fast Fourier Transformation increases the computation speed by a several times and thus makes it easy with our calculations.

We use this Fast Fourier Transformation (FFT) to calculate the far field amplitude of the object wave. After we have calculated the object wave we need a reference wave from the same point source. We create the reference wave from the point source mathematically by

running our algorithm. Then we apply the Fast Fourier Transformation (FFT) on the reference wave to find the far field amplitude of our reference wave [15].

According to the formula of generating holograms we need to superimpose the two object wave and reference wave and the resulting wave will be our hologram interference pattern. But the object and the point source of light are not physically present so we compute both mathematically. We compute the superimposed wave mathematically and thus we get the final holographic pattern. The pattern is then reconstructed on a holographic plate with lasers and then reconstructed to form the actual holographic image.

3.5 Architecture of CGH

A holographic pattern can be obtained from an image using the computer generated. There are several steps in computing the interference pattern from the input image matrix. Several algorithms are then applied on the interference pattern to obtain the final reconstructed image. A flow chart of the working principles of computing a computer generated hologram is shown below [16]:

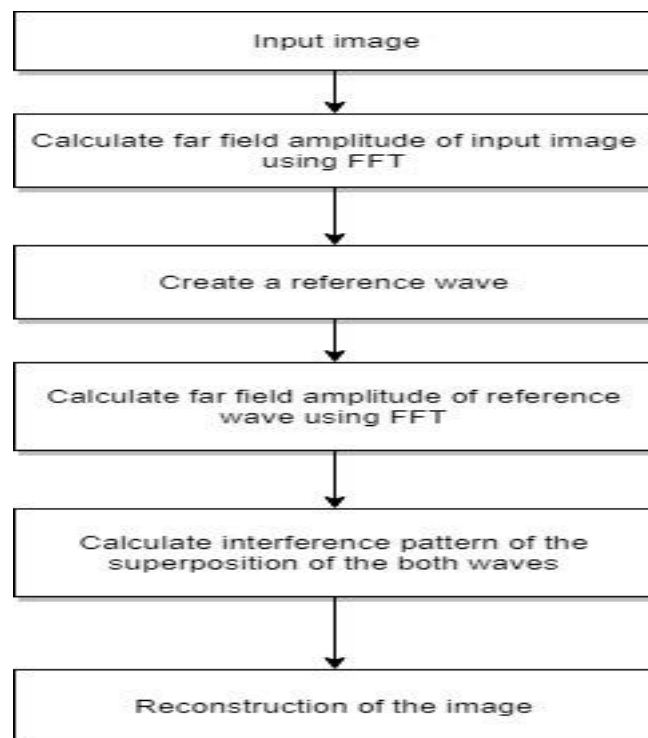


Figure 3.5(a): Flowchart of the architecture of computer generated holography

Below are the steps described for generating a CGH

Step 1:

In the first step the image is broken down into matrix and then the far field amplitude is calculated by Discrete Fourier Transform. Then the low intensity components are shifted into the centre of the image using another function.

Step 2:

In this step a reference wave is created. For creating the reference matrix the area of the input image is calculated and a matrix is created which is then filled with zeros. One point in the image is illuminated because we are assuming light coming from a point source. For computational ease the point is considered to be the centre of the reference wave.

Step 3:

Fourier Transform operation is performed on the reference wave and in the same way like the object wave the low intensity components are shifted to the centre.

Step 4:

This step is considered as the superposition of the waves. The far field amplitude of the reference wave and the object wave is added and we get the final image array. The absolute value of the object image is subtracted by matrix subtraction. The result is the holographic interference pattern.

Step 5:

This step is the reconstruction step. In this step Fast Fourier Transform is applied on the matrix that has been obtained from subtracting the object wave from the interference matrix. Then by calculating the absolute value of the matrix the reconstructed image can be obtained. By in this step the image obtained by reconstruction is overlapped by the same image of opposite orientation. This is because two images are created during this process.

Chapter 4

Proposed Methodology

We are proposing a method of encrypting and decrypting holographic interference pattern by using the principle of digital Holography using python as our programming language. For this firstly we will be creating our very own python library for generating computer generated holograms. The whole process will complete in four stages.

Firstly, we will generate a holographic pattern and the reconstructed final image by using python and a computer vision library openCV. This will allow us to take any input image and convert the matrix into a holographic matrix which can later be reconstructed by our algorithm.

Secondly, while reconstructing the image one of the major problems is that two images of opposite orientation will be formed and they will overlap. So the supplied image needs to be cropped and the dimensions of the images need to be supplied for the creation of the reference beam. So because of this overlap in the final reconstruction we will not be able to see the exact image, rather we will see two images overlapping on each other. We are also suggesting a solution for all of these problems so that it will work on images on any size. We are providing a method with which the reconstructed image will not be overlapping. Our algorithm will automatically rescale the image and a reference beam will be calculated according to that size of the image.

We have selected Discrete Fourier Transform (DFT) method for converting our image from a spatial domain into a frequency domain. But the Discrete Fourier Transform algorithm increases the computation time so we are using a better implementation of the algorithm which is known as Fast Fourier Transform (FFT). With python libraries we will apply Fast Fourier Transform (FFT) on our image and the reference beam to find the far field amplitude and to calculate the amplitude of the wave when the superposition of the object wave and the reference wave occurs. We have chosen python as our programming language because it is free and open source. Though most of the work in this field is done in MATLAB but MATLAB is closed and proprietary. So, at times it is not possible to dig deep into the methods and the algorithms. Python is fast and reliable with a huge library support. Python also has more than one library dealing with the Fast Fourier Transform (FFT) algorithm. We

will be also using openCV and numpy libraries of python for processing our images and to calculate the FFT.

Thirdly, after we figure out how to reconstruct the image without overlapping, we will apply our encryption on the image. We are suggesting a single key or symmetric cryptography technique for encryption of the image. For this we have to take the reference wave matrix and fill it with randomly created variables. For the holographic pattern we need a blank reference wave with point intensity because we are assuming that the light is coming from a coherent point source. So, if we will generate a reference wave with random pixels and we will note down the pixels and store them as our variables and we will take intensity variable which will be greater than any of the random variables we will take. Then we will convert them into string and from that we will generate our 'encryption key'. We will then add the reference wave that we have calculated and the object wave and create an encrypted holographic image pattern which is the cipher image. Now if we reconstruct this cipher image we will see nothing just some black lines or some random pattern.

Lastly, we are suggesting a decryption method from the pattern we have generated and the key. Our decryption algorithm will first calculate the intensity and the pixel variables from the key. Then we have generated an algorithm which can take that reference wave with random pixel values and with the help of the encryption key it will convert this to a second reference key wave. Then we find the far field amplitude of that second reference key wave. Now after we add that second key wave to our encrypted pattern, we will get the holographic pattern of our input image. Now if we reconstruct the image we will get our original input image.

Chapter 5

Experimental Part

For our experiment we have chosen python as our programming language. Python is a strong programming language and it has several advantages over Matlab because it is free and open source and we can look under the hood how the methods were written unlike Matlab which is closed and proprietary. The python libraries that we needed for our work are:

- numpy
- opencv
- matplotlib

Firstly we have implemented the process of developing a computer generated hologram and the reconstruction in python. Our algorithm will take the input image and generate a holographic pattern and a reconstructed image from that input image.

For breaking down the image into image matrix we have used the openCV library. We have used the numpy fft library for the calculation of the far field amplitude of the object and reference wave. After applying fft we need to take the low intensity components to the center of the image for the ease of calculation. We do that by the `fftshift()` function which is available in numpy. After calculation the far field amplitudes we add the both far field amplitude and we subtract the square of absolute value of the object wave, we get a final holographic pattern.

Now we can reconstruct the holographic patter. We apply Fourier Transform on the final holographic pattern and then shift the low intensity components to the centre. Then we find the square of absolute value of our reconstructed image and then we plot the matrix and get the final reconstructed image. In this process we will get two output, the first one is in straight orientation and the other is inverted. The two generated images will overlap with each other in this stage of the process.

The flowchart of our algorithm is shown below:

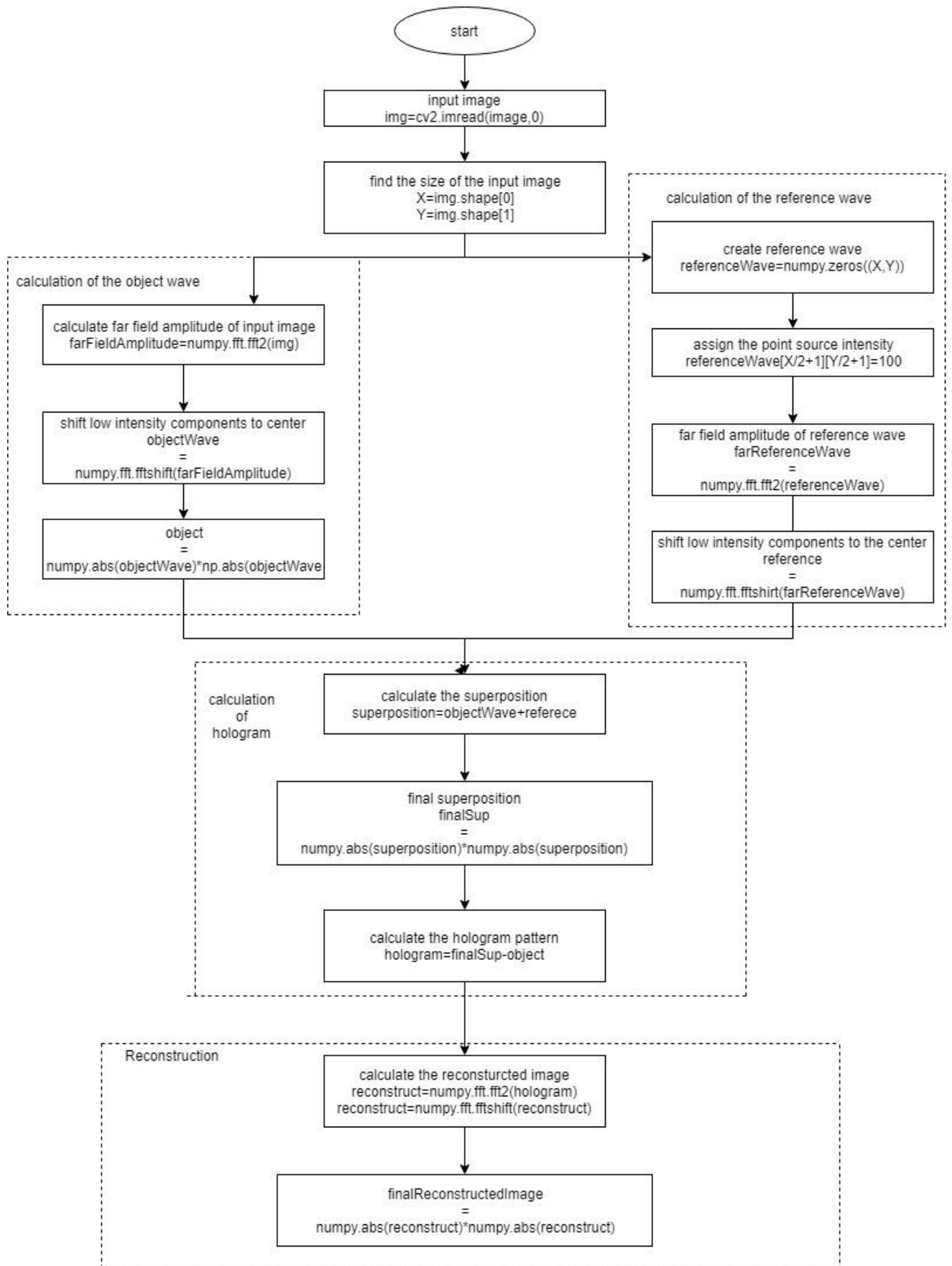


Figure 5.0(a): Flowchart of hologram generation and reconstruction

We have broken down our further experiment into the following section:

- Solve the overlapping problem
- Encrypting the given image
- Generate an encryption key by an algorithm
- Decrypting the image by using the encrypting key
- Reconstruction of the Decrypted image

5.1 Solving the overlap problem:

First an image is taken as an input. Our algorithm will return the holographic interference pattern as an output which can later be reconstructed to form the reconstructed holographic image.

For our first experiment let us take the following input image.

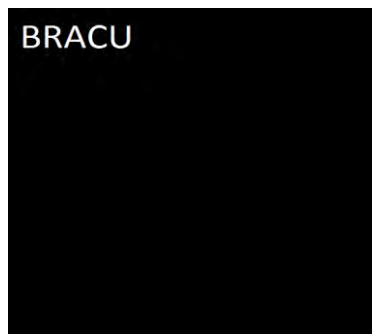


Figure 5.1(a) Input Image

Our algorithm will convert this image into a holographic pattern and then later reconstruct the image. Our algorithm returns the following results,

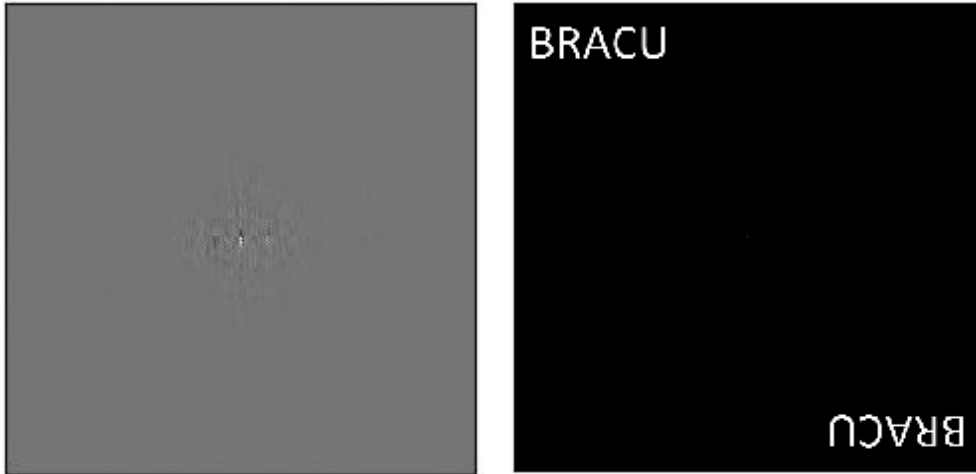


Figure 5.1(b): Hologram

Figure 5.1(c): Reconstructed image

From our algorithm we get the two outputs. Figure 5.1(b) shows the holographic interference pattern of figure 5.1(a). Figure 5.1(c) shows the reconstruction of figure one. But from figure 5.1(c) we see that it is not exactly like our original image. That is because by the theory of holography two images are created during reconstruction. The first one is straight and the second one is inverted. If this matter is not taken care of then parts of image might get overlapped. In our previous example we took a simple image mostly of black parts so the overlapping is not totally visible. Let, we look at another example to figure out what actually can happen when our reconstruction is overlapped. Let us now take an image of a car.



Figure 5.1(d). Input Image

We take image of a car in figure 5.1(d) and run our algorithm and the result is:



Figure 5.1(e). Reconstructed Image with overlapping

Figure 5.1(e) is the reconstructed image of figure 5.1(e) as we addressed earlier this overlapping causes a huge problem while dealing with full sized images. This can be solved manually by editing our input image by cropping it to a quarter now if we do that then it can be helpful but every time we need to manually edit an image which is not a very pleasant thing to do. Then again the output image will be two with one inverted on the other quarter then again we have to crop it manually. Let us look at one such example:



Figure 5.1(f). Manually cropped input image



Figure 5.1(g). Reconstruction of manually cropped image

Figure 5.1(f) is the image that we have manually cropped and gave it to our program as an input and the figure 5.1(g) is our output. Here the images are not overlapping but to get the actual holograph of a single image we again need to crop the image.

We have developed an advanced algorithm which extend the size of the image to retain the quality in such a way that the reconstructed image is not overlapping. The final reconstructed image will be exactly similar to our input image. Our algorithm will ensure the quality of the image and ensure that parts of the reconstructed image are not overlapping. Now we look at how our advanced algorithm works for:

Our algorithm has two parts first:

5.1.1 Part 1 (Scaling the input image):

In this part we take the input image. Now we can do one of the two operations either extend the image or crop it. Cropping might play a role in reducing the quality of our original image. So we extend the size of the image to double the size of the original image and the fill the newly formed extended pixels with 0.

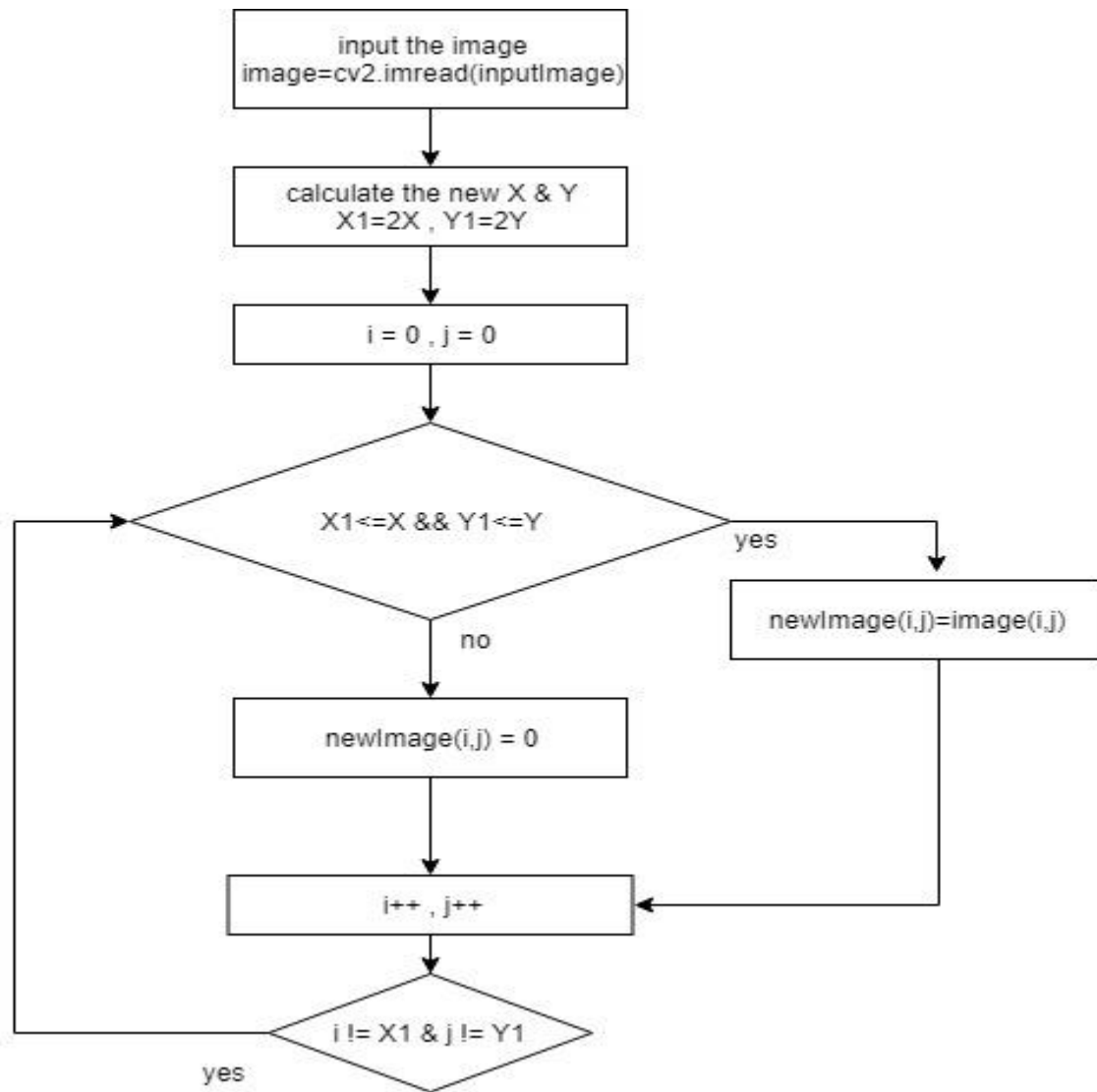


Figure 5.1.1(a): Flowchart of extending the matrix of input image

5.1.2 Part 2 (Reconstruction):

In second part of this algorithm comes after the reconstruction pattern has been figured out. When the reconstruction pattern is figured out the image is cropped to one fourth of its size. For doing that we take our input reconstructed array and reduce the X dimension by half and Y by half. The newly formed array when plotted will give the reconstructed image exactly like our input without any self-overlapping parts.

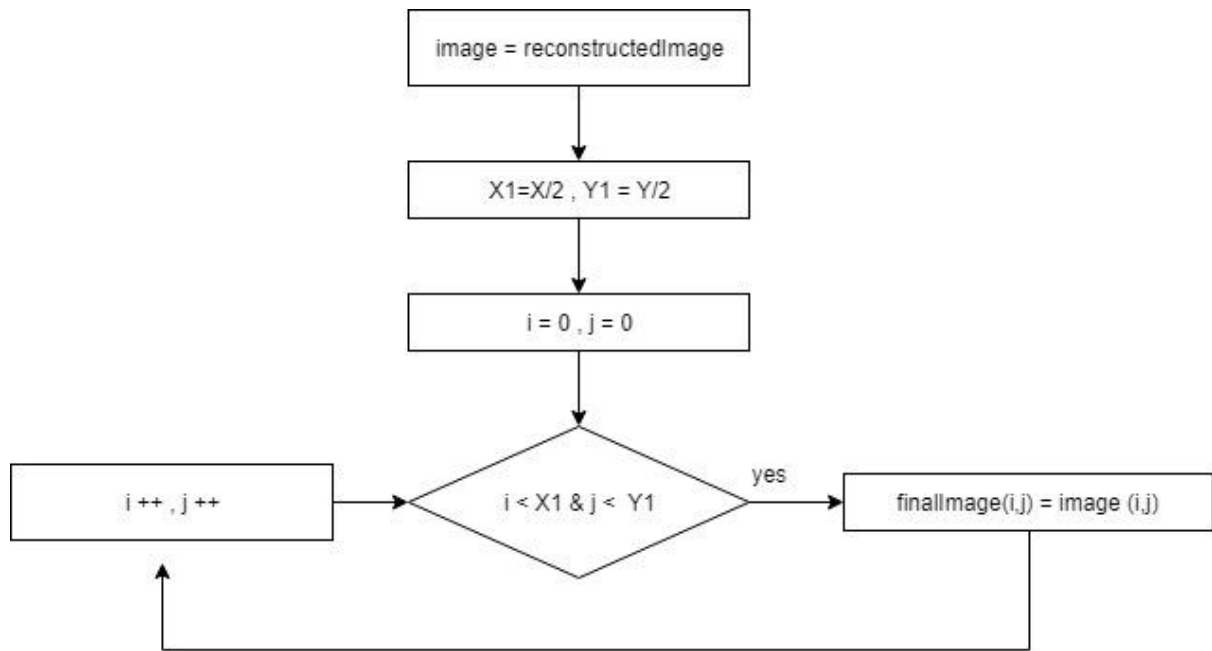


Figure 5.1.2(a): Flowchart of the reconstruction



Figure 5.1.2(b). Input Image



Figure: 5.1.2(c). Reconstructed Image

The figure 5.1.1(c) shows our original input image and 5.1.1(d) shows the reconstructed image. Here we can see that our algorithm has worked perfectly in reconstruction the original image with parts of it overlapping with itself.

5.2 Encryption and Decryption:

5.2.1 Encryption:

Now that we have figured a way to perfectly reconstruct the image from the given image now we will encrypt the image. For encryption let us take our previous example of the car as our sample plain image.



Figure 5.2.1(a). Input Image

Now by the general process of hologram generation we convert the image into grayscale pixel array and perform Fast Fourier Transform on the array. Then to take the DC components to the centre we perform a Fourier shift and get the image in frequency domain.

Now for encryption we have to perform operations on our reference wave. For a general hologram construction we took a black reference wave filled with zeros. But for encryption of the image we take some random variables

$$x_1, x_2, x_3, \dots, x_n.$$

Here n is the number of variables we are taking for the process of encryption. For our algorithm we have used five variables and the each of those variables has a random value which is generated by our cipher algorithm.

After we have got the values for the randomly generated variables we take another variable intensity which will be generated by our algorithm too such that, $Intensity > x$

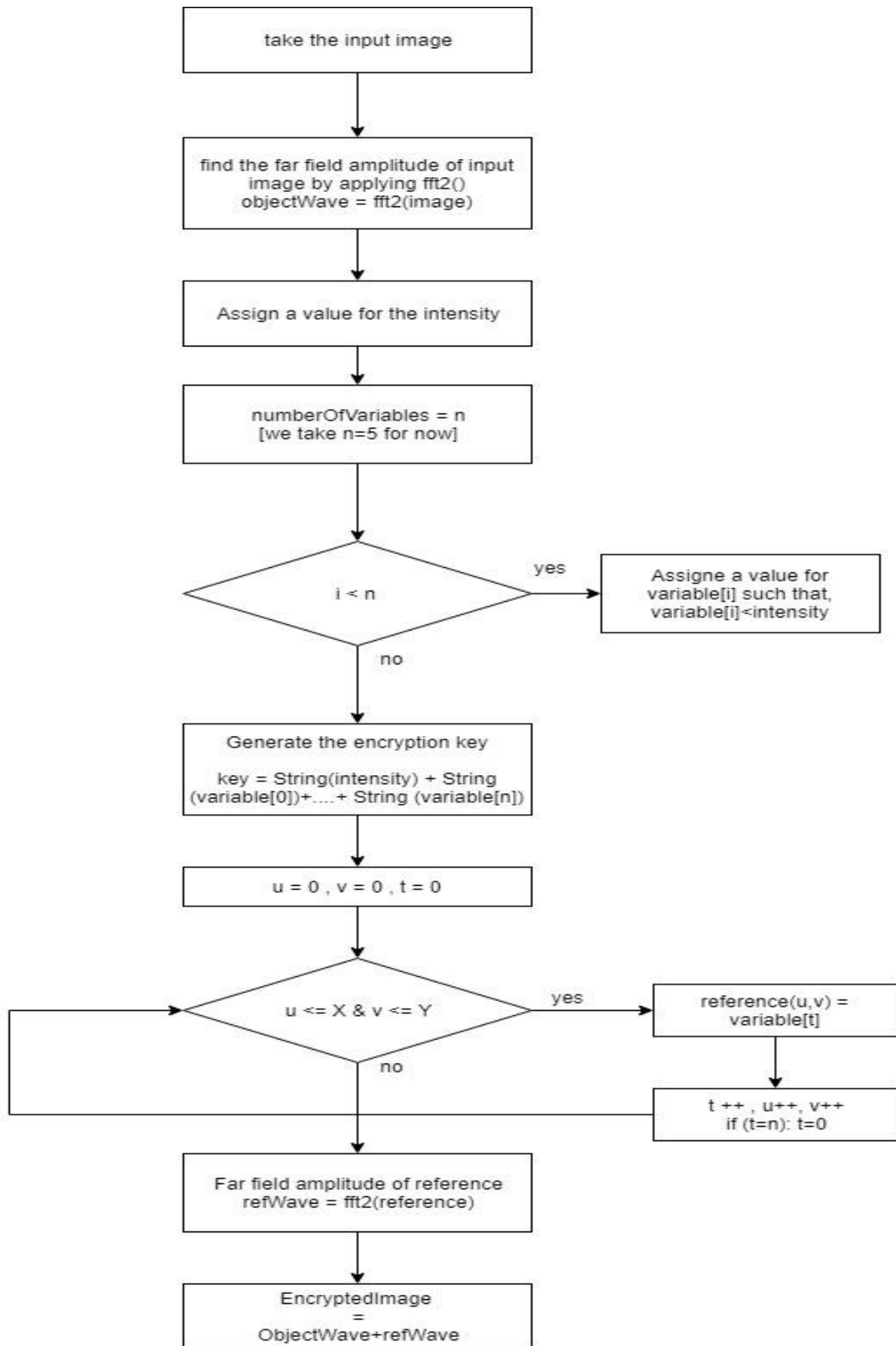


Figure 5.2.1(b): Flowchart for the encryption of input plane image

We are taking the intensity greater than our previously declared variables for the ease of future calculations.

Now we periodically fill our reference array with the five variables that we have just randomly generated. We are periodically filling that wave for a pattern so that we can have our cipher image.

Now we have to generate an encryption key. For the generation of our encryption key, our algorithm converts all the variables and the intensity to string and adds the values to a sequence of string. The sequence of string is our encryption key.

Here n is the number of variables used. In our case it is 5. Now for our input image figure (8) and for five reference variables a key is automatically generated. For this case the key that is generated randomly is *574637442715*.



Figure 5.2.1(c). Input

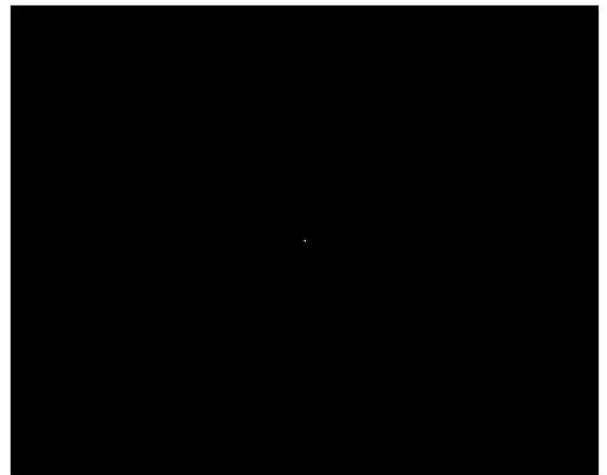


figure 5.2.1(d). Cipher Image

Figure 5.2.1(c) shows a grayscale version of the image that we have used and figure 5.2.1(d) is the encrypted image. To reconstruct this image for this encryption we need the encryption key.

For decrypting figure 5.2.1(d) we need to input this cipher image figure 5.2.1(d) into our decryption algorithm along with our encryption key.

5.2.2 Decryption:

From the encryption key the intensity and the variables that we have used is figured out.

The flowchart of the decryption algorithm is shown below:

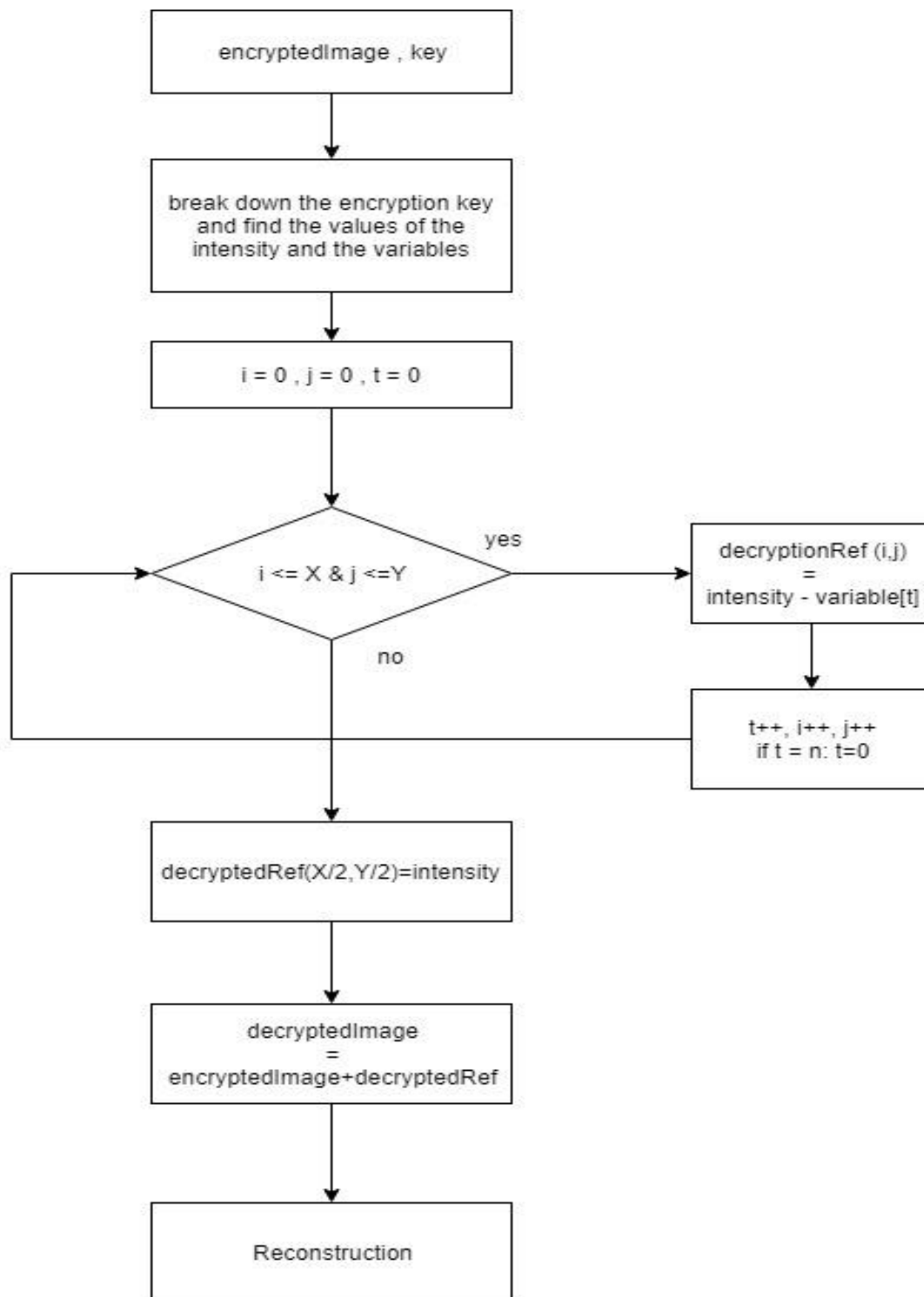


Figure 5.2.2(a): Flowchart for decryption of the cipher image using the encryption key

Below is a demonstration of how our encryption key is broken down into parts.

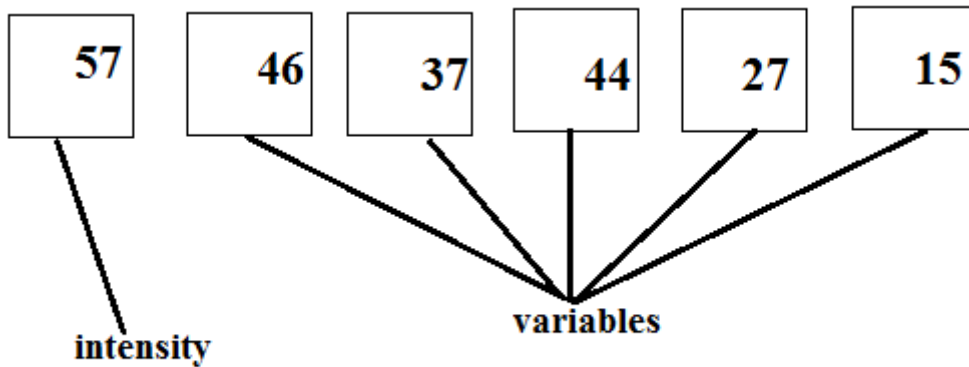


Figure 5.2.2(b): Breakdown of the encryption key

Now we have broken down the encryption key into parts. Now we will use these values to calculate our reference wave. With the provided encryption key we reconstruct our cipher image.



Figure 5.2.2(c). Reconstructed Image

Figure 5.2.2(c) is the decrypted reconstructed image that we have obtained from our cipher image. Now in case if we put the wrong encryption key then the decrypted image is shown below:

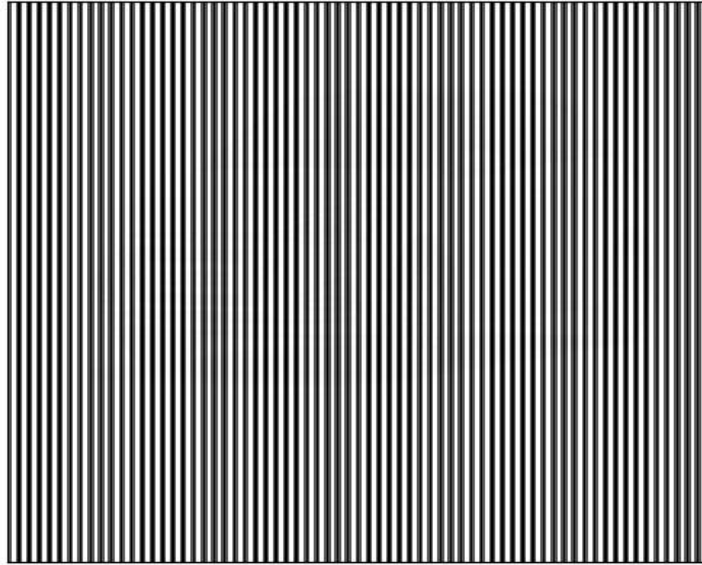


Figure 5.2.2(d). Reconstruction with wrong key

The above figure 5.2.2(d) shows the decryption of the cipher image using the wrong encryption key.

Chapter 6

Results and discussions

6.1 Solution to the overlapping problem

To compare between images we take three images and compare their results.



Figure 6.1(a)



Figure 6.1(b)



Figure 6.1(c)

Input images

Image 6.1(a) is a picture of a String with the writing BRACU. The second image, figure 6.1(b) is a numerical value and the third image figure 6.1(c) is the picture of a car. We have calculated how our algorithm behaves in case of these three samples.

First of all we checked the results before the overlapping problem was resolved. The images with the overlapping problems are shown below:



Figure 6.1(d)



Figure 6.1(e)



Figure 6.1(f)

Reconstruction with overlapping

Here we can see the results of these images before the overlapping problem was solved. Now we tested these images with our algorithm to see how they behaved after the overlapping problem was addressed.

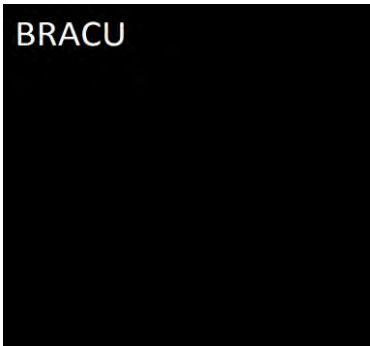


Figure 6.1(g)



Figure 6.1(h)



Figure 6.1(i)

Reconstruction after solving the overlapping problem

From the results we can clearly see that after we ran our algorithm the images were not overlapping anymore. Thus we can conclude that our algorithm worked perfectly in addressing the overlapping problem.

6.2 Encryption and Decryption

6.2.1 Encryption:

For encryption we will run our algorithm with the three images as plain image that we have previously used to show the overlapping problem. Now we run our encryption algorithm or cipher on these images and get the cipher image.

With each encryption we get an encryption key. The encryption key is further needed for decrypting these cipher images. After running our encryption algorithm or cipher we get the following cipher images with pattern of complex values. The encryption keys generated for the images Figure 6.1(a), Figure 6.1(b) and Figure 6.1(c) are 531233431945, 841513183218 and 853027482335 respectively. We used these keys to decrypt the cipher images.

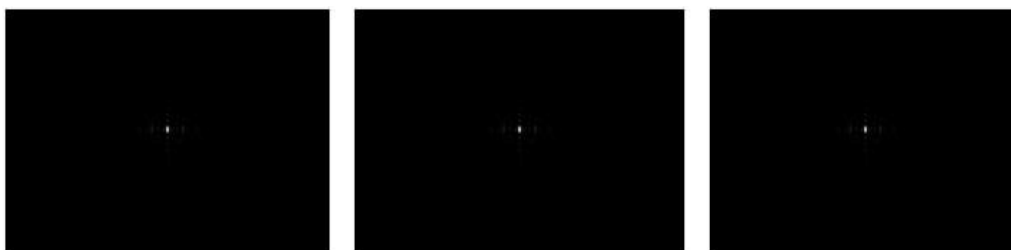


Figure 6.2.1(a)

Figure 6.2.1(b)

Figure 6.2.1(c)

Cipher images

Figure 6.2.1(a), 6.2.1(b) and 6.3.1(c) are the cipher images of the input plane images Figure 6.1(a) , Figure 6.1(b) and Figure 6.1(c) respectively.

6.2.2 Decryption:

We have used these keys to decrypt the images. At first we will try decrypting with the correct keys and then we will reshuffle the keys to see if we can decrypt the images.



Figure 6.2.2(a)



Figure 6.2.2(b)



Figure 6.2.2(c)

Decryption with the correct Encryption key

So here we have given the correct encryption key to and found the result which we have expected that is the correct reconstructed image. Now we will reshuffle the encryption keys and check if we can get the reconstructed image.

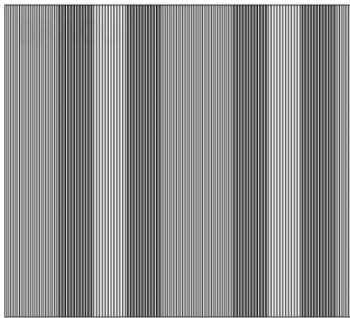


Figure 6.2.2(d)

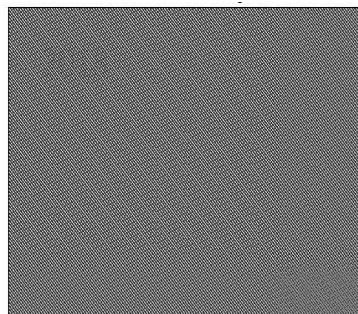


Figure 6.2.2(e)

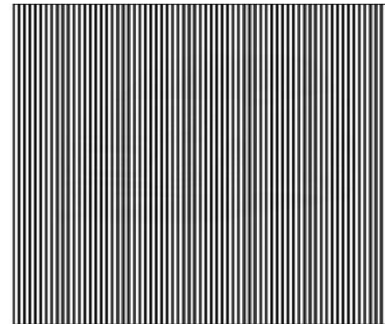



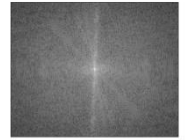
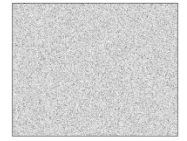
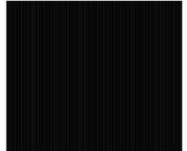

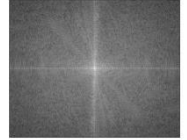
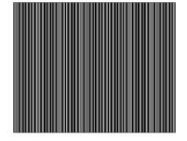



Figure 6.2.2(f)

Decryption with the wrong key

With the wrong encryption key we get the above result. So we can conclude that our algorithm has successfully been able to encrypt and decrypt holographic pattern from images.

A table is given with different steps involved in our procedure and the changes that the image and the image matrix which occurs during the steps are shown in the table.

Description	Reference pixel	Value	Plotted Image
Input Image	(120,120)	90	
Applying Fourier Transform on input image	(120,120)	(7517.24306782+7104.15931242j)	
Shifting low intensity components to the center (object wave)	(120,120)	(4259.51321617-11253.9490313j)	
Magnitude Spectrum of object wave	(120,120)	147.800927924	
Image generated with random pixels equal to the size of the input	(120,120)	35.0	
Applying Fourier Transform on the generated image (Encryption wave)	(120,120)	(0.000137052361546+2.88502964915e-05j)	
Cipher	(120,120)	(85.7964522232+1617.5074506j)	
Magnitude spectrum of cipher		147.800927699	
Second Reference wave	(120,120)	(132.146357604-452.254663253j)	
Decrypted Holographic pattern with key	(120,120)	-0.255771175027	


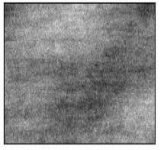
Reconstructed image with encryption key	(120,120)	7.47925277749e+18	
Reconstructed image with the wrong key	(120,120)	1.21299004045e+18	

Table 6.1 Steps of the algorithm

In the Table 6.1 the steps and the changes that occurs when we go from one step to the other is shown. We have taken the pixel $(X,Y) = (120,120)$ as the reference pixel. The values of the reference pixel after each operation are given. The change of the image matrix has also been plotted.

6.3 Discussion

The cipher images that we have generated appear as black because the intensity values of the Fourier image is too large to be displayed on the screen. We can view the magnitude spectrum of the image by applying logarithmic transformation on our cipher image. For example, if we take our cipher image 6.2.1(c) and apply logarithmic transform on that image we get,

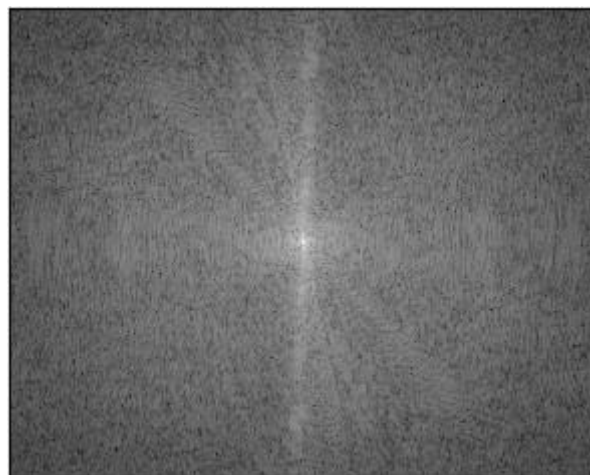


Figure 6.3(a): Magnitude Spectrum of the cipher image

This above figure 6.3(a) is the magnitude spectrum of the cipher image of the car that we have used.

Generally when we are reconstructing a holographic interference pattern the final result that we get is basically a hologram which consists of two images with overlapping parts. But we have further developed a scaling technique by which the final reconstructed image is the image itself and not the holographic image. Thus by our algorithm the input image is perfectly reconstructed. Because of our scaling algorithm the encryption and decryption of the image was possible.

The most famous image encryption algorithm uses a technique of scrambling the pixels and then the scrambled image is the cipher. This usually has a pattern of pixel scrambling and if the pattern can be figured out then with that pattern the encryption can be broken. In our algorithm we have taken the image from spatial domain to the frequency domain by applying Discrete Fourier Transform. This can be termed as one level of encryption. To re-transform the image from the frequency domain both the magnitude and phase information has to be preserved. If any of the information is scrambled here then re-transformation is not possible. We have used our second level of encryption here. We took our custom made reference wave where we have scrambled the pixels and took that reference wave to the frequency domain and then added it to the object wave which is in the frequency domain. Now the magnitude and phase data has been scrambled and from here the image cannot be re-transformed. From here we have generated the holographic pattern and as we said earlier because our scaling algorithm allows us to get the image by reconstructing the interference pattern.

Now what makes our encryption stronger than the other image encryption techniques is that while in the frequency domain it is not possible to find a pattern because the coordinates are so large and has no specific pattern. So our cipher image cannot be re-transformed, it has to go through the reconstruction process for getting back the original image. Our algorithm takes the encryption key and generated a second reference wave from that key and then takes the second reference wave to the frequency domain and then it is added to the cipher image. Then the cipher image can be reconstructed to find the original image. This whole process adds a two level encryption which is stronger than the general pixel scrambling technique.

Chapter 7

Conclusion

The above techniques are currently applicable in the context of 2D images. But the algorithm is general enough to take a 3D object as an input and encrypt the 3D object using the same technique, with minor tweaks in the code. The techniques described in this paper can also be used to overcome some basic problems which is faces during the reconstruction of image by Computer Generated Holography (CGH) technique such as the image overlapping problem that occurs during the reconstruction of the original image from a hologram. These techniques have a wide range of utilities in digital forensics and security domain. Image encryption or hiding is an important practical problem and the CGH technique described in this paper provide a computationally fast and inexpensive way to generate a strong encryption of an image. The encrypted images that is generated by this technique cannot be decrypted without the encryption key because by the application of Discrete Fourier Transform the image matrix gets one level of encryption and then we have applied a second level of encryption to this image. This technique can be used in a wide range of security applications in the future. Our holographic interference pattern generation algorithm can also be used to generated holograms from any given image and with the use of hologram generation instruments and lasers, these holograms can be printed in any holographic plate. Our algorithm is applicable only for grayscale images for now but in future it can be developed so that coloured images can be transformed into holographic interference pattern and can be encrypted using our technique.

References:

- [1] B. R. Brown, A. W. Lohmann, "Complex Spatial Filtering with Binary Masks", *Appl. Optics*, Vol 6, (1967), 1739-1748.
- [2] Cooley, James W.; Tukey, John W. (1965). "An algorithm for the machine calculation of complex Fourier series". *Math. Comput.* 19: 297–301. doi:10.2307/2003354
- [3] GABOR, D. (1948). A New Microscopic Principle. *Nature*, 161(4098), pp.777-778.
- [4] Jericho, M., Kreuzer, H., Kanka, M. and Riesenber, R. (2012). Quantitative phase and refractive index measurements with point-source digital in-line holographic microscopy. *Applied Optics*, 51(10), p.1503.
- [5] Leith, E. and Upatnieks, J. (1965). Holograms: Their Properties and Uses. *Optical Engineering*, 4(1).
- [6] Garcia-Sucerquia, J., Castañeda, R. and Medina, F. (2002). Fresnel–Fraunhofer diffraction and spatial coherence. *Optics Communications*, 205(4-6), pp.239-245.
- [7] Computer Generated Holography, James B Wendt.
- [8] Kong, D., Cao, L., Shen, X., Zhang, H. and Jin, G. (2018). Image Encryption Based on Interleaved Computer-Generated Holograms. *IEEE Transactions on Industrial Informatics*, 14(2), pp.673-678.
- [9] Auslander, L. and Grunbaum, F. (1989). The Fourier transform and the discrete Fourier transform. *Inverse Problems*, 5(2), pp.149-164.
- [10] Yang, D. (1990). Fast discrete Radon transform and 2-D discrete Fourier transform. *Electronics Letters*, 26(8), p.550.
- [11] Singh, S. and Singh, K. (2013). Image Change Detection by Means of Discrete Fractional Fourier Transform. *International Journal of Computer Applications*, 77(16), pp.16-20.
- [12] Gabor, D. (1966). Fundamentals and applications of holography. *Vacuum*, 16(6), p.313.
- [13] An Optical Interferometry System for Measuring Three-Dimensional Displacements. (2004). *Instruments and Experimental Techniques*, 47(5), pp.711-714.
- [14]] R.J. Collier, C.B.Burckhardt, and L.H.Lin, 'Optical Holography' (Academic, New York, 1971) (p-206-217).

- [15] Lee, W. (1970). Sampled Fourier Transform Hologram Generated by Computer. *Applied Optics*, 9(3), p.639.
- [16] Divya P.S.1, Sheeja M.K2 “A study and simulation of computer generated Hologram”, *International Journal of Advances in Engineering & Technology*, July 2013. ©IJAET, ISSN: 22311963
- [17] D. Kong, L. Cao, X. Shen, H. Zhang and G. Jin, "Image Encryption Based on Interleaved Computer-Generated Holograms," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 673-678, Feb. 2018. doi: 10.1109/TII.2017.2714261
- [18] M. K. Sheeja, P. T. Ajith Kumar, S. Nair Achuthsankar “Photopolymer-based holographic variable data storage system for security applications” *Proc. SPIE. 6352, Optoelectronic Materials and Devices 635224* (September 21, 2006) doi: 10.1117/12.689041
- [19] Sheeja M.K., Ajith Kumar P.T., Achuthsankar S. Nair, “Encrypted Fourier holographic data storage with variable data reference wave for optical information security”, *Proceedings of SPIE, Photonics Asia 2007, Vol. 6832, Beijing, China, Nov. 2007.*
- [20] Giuseppe A. Cirino, Patrick Verdonck, Ronaldo D. Mansano, José C. Pizolato Jr., Daniel B. Mazulquim and Luiz G. Neto (2011). “Digital Holography: Computer-Generated Holograms and Diffractive Optics in Scalar Diffraction Domain, Holography - Different Fields of Application”, Dr. Freddy Monroy (Ed.), ISBN: 978-953-307-635-5
- [21] J.W. Goodman, “Introduction to Fourier Optics”, San Francisco: McGraw Hill, (1968).
- [22] Jorge Garcia-Sucerquia, Wenbo Xu, Stephan K. Jericho, Peter Klages, Manfred H. Jericho, and H. Jürgen Kreuzer, “Digital in-line holographic microscopy”, 2006.
- [23] <http://light.ece.illinois.edu/ECE460/PDF/Holography.pdf>
- [24] Zhang, H. and Wang, R. (2012). Image Encryption Technology Based on Composite Chaotic System and Symmetric Encryption Algorithm. *Key Engineering Materials*, 500, pp.465-470.
- [25] PANG, L., LI, H., PEI, Q., LIU, Y. and WANG, Y. (2012). A Public Key Encryption Scheme with One-Encryption and Multi-Decryption. *Chinese Journal of Computers*, 35(5), pp.1059-1066.