3.1 INTRODUCTION

In this chapter, we present an overview of the research methodology of proposed research work. Experimental methodology is used to analyze different types of images and evaluate the performance of the algorithm. Text images of different sizes are checked based on the length of the text. Different types of image formats have been tested; most frequently used format of images like JPEG and BMP are checked and found working successfully.

We use MATLAB (MATrix LABoratory) as a development tool. It is used to check the images pixel values, recognizing colour channels etc. MATLAB is a high-level language for numerical computation, visualization and application development. It also provides an interactive environment for iterative exploration, design and problem solving. MATLAB consists of mathematical functions for linear algebra and statistics, Fourier analysis, filtering, optimization, numerical integration and solves ordinary differential equations as well as built-in graphics for visualizing data and tools for creating custom plots.

3.2 PROPOSED ALGORITHM AND FLOWCHART

We present algorithm to hide the image and text within image, we develop three algorithms with different fractals like Mandelbrot Fractal, Hilbert Curve and RGB channel of image respectively. The algorithm is developed by keeping in the mind that no one can easily detect the hidden message or image.

3.2.1 Steganography of Text and Images using Mandelbrot Fractal

3.2.1.1 Embedding Algorithm

In this technique we generate Mandelbrot fractal location on input image. Fractal is generated by performing a series of transforms on a line segment from (0, 0) to (1, 0).
3. RESEARCH METHODOLOGY

\[ f = \begin{bmatrix} a(p) & b(p) \\ c(p) & d(p) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e(p) \\ f(p) \end{bmatrix} \] (1)

The affine transform in eq. (1) shows the parametrical fractal transforms. The a, b, c and d are coefficients of the matrix multiplication do the scaling and rotation operations x and y respectively, followed by the addition of the translation, e and f, coefficients. All six coefficients are functions of the input geometric parameters, p, which govern the geometry of the generated fractal image. In order to plot a line segment in the final image, these operations must be performed twice, once to generate the starting coordinates of a line segment and once to generate the ending coordinates. These operations on a line segment are repeated in a recursive manner, until the specified number of iterations is reached [1].

In this approach we take three images showed in figure 11 for the purpose of steganography of text and images using Mandelbrot fractal. Image 11(a) is an original image which refers to as cover or carrier image. Tree image 11(b) is an image to embed within cover image and image 11(c) is an image of Mandelbrot Fractal which is used to identify location on image 11(a).

[Fig 11(a): Kido Image (Cover/Carrier) (b): Tree Image (Image to Hide) (c): Mandelbrot Fractal]
3. RESEARCH METHODOLOGY

First, image 11(c) of Mandelbrot Fractal is generated on kido image 11(a) to identify locations in kido image 11(a) after that kido image 11(a) is splitted into four parts. All pixel values are changed to 8 bit value 11111111 using bit manipulation technique and convert last two bit values to 0 using bitor and bitand operations and make the value 11111100.

Tree image 11(b) bit value is changed to 8 bit value 11111111 and these values are divided into four pairs 11,11,11,11. Now, first part of kido image 11(a) last two bits 00 are embedded with first pair 1 1 (1 and 2) of tree image 11(b). In the second part last two bits 00 are embed with second pair 1 1 (3 and 4), third part of last two bits 00 are embedded with third pair 1 1 (5 and 6) and the fourth part of last two bits 00 are embedded with fourth pair 1 1(7 and 8). The embedding process is done anti-clock wise to generate a new kid image 12(a) containing stegogramme which looks like original kido image 11(a).

![Kid Image (With Stegogramme)](Fig 12 (a): kid Image (With Stegogramme))
3. RESEARCH METHODOLOGY

Step 1: *Get the image “A”.*

Step 2: *Generate Mandelbrot Fractal to defined targeted region in Image.*

Step 3: *Change the Pixel value of the targeted region in an Image.*

Step 4: *Make the partition of image in four parts.*

Step 5: *Get the image ‘B’ to embed in image ‘A’*

Step 6: *Separate bits and make pair of two bits in four groups using bit manipulation.*

Step 7: *Join the pairs of image ‘B’ into the change value of image ‘A “anticlockwise to generate new image A₁ containing stegogramme₁”.*

Step 8: *End*

[Algorithm 1 – *Embedding algorithm to hide image and text using Mandelbrot Fractal*]
3.2.1.2 Flowchart for Embedding

[Flowchart 1: Embedding flowchart to hide image and text using Mandelbrot Fractal]

START

Get the image A

Generate Mandelbrot Fractal on image A to identify location

Get Image B and change the pixel value and make four pairs of two bits

Embed two pair bits of image B into image A anticlockwise until all pixels are padded to generate new image

Is the image embedded?

Y

Compare new embedded image with original image A

STOP

N
3.2.1.3 Extracting Algorithm

In the extraction process we take the embedded kid image 13(a) and recognize targeted region by generating Mandelbrot fractal image 13(b) on kid image 13(a).

Firstly, extract last two bits 1 1 from each part of kid image 13(a). After getting all the values from different four groups of kid image 13(a) in anticlockwise manner and join them together to extract the hided tree image(extracted) 14 (a) and compare original image, extracted image and embedded image for the intensity of pixels.
3. RESEARCH METHODOLOGY

Step 1: Get the embedded image “A_1”.

Step 2: Generate Mandelbrot fractal to define the locations of stegogramme.

Step 3: Extract the embedded pixels and make four groups by changing the value using bit manipulation.

Step 4: Pad 1\textsuperscript{st} and 2\textsuperscript{nd} group and 3\textsuperscript{rd} and 4\textsuperscript{th} group.

Step 5: Merge all groups derived from previous step.

Step 6: Extract the actual message (text) or Image and cover image A_1.

Step 7: End.

[Algorithm 2 – Extracting algorithm to hide image and text using Mandelbrot Fractal]
3.2.1.4 Flowchart for Extraction

[Flowchart 2: Extracting flowchart to hide image and text using Mandelbrot Fractal]

START

Get the image $A_1$ containing stegogramme

Generate Mandelbrot Fractal on image $A_1$ to identify location

Get last two bits of Image $A_1$ and join them to get the embedded image

Does the new embedded image contain stegogramme?

Y

Compare original, embedded and extracted image

STOP

N

Find Embedded Locations
3.2.2 Steganography of Text and Images using Hilbert Curve

3.2.2.1 Embedding Algorithm

Hilbert curve is a space filling curve that visits every point in a square grid with power of 2. The basic elements of the Hilbert curves are a square with one open side and a vector that joins squares. The open side of square can be top, bottom, left or right. In addition, every square has two end-points, and each of these can be the entry point or the exit point. So, there are eight possible varieties of squares. Usually Hilbert curve uses only four types of squares and has direction of up, down, left or right. The first order Hilbert curve is just a single square. The second order Hilbert curve replaces that square by four squares, which are linked together by three joins. Every next order repeats the process or replacing each square by four smaller squares and three joins.

In this approach, we take three images for the purpose of steganography of text and images using Hilbert Curve. Manaal image 15(a) is an original image which refers to as cover or carrier image. Text image 15 (b) is an image to embed within cover image and image 15(c) is an image of Hilbert Curve which is used to identify location on Manaal image 15(a).

![Fig 15 (a): Manaal Image (Cover/Carrier) (b): Text Image (Image to Hide) (c): Hilbert Curve Image[15]](image)
3. RESEARCH METHODOLOGY

Hilbert Curve image 15(c) is generated on Manaal image 15(a) to identify locations in Manaal image 15(a) after that, all pixel values of Manaal image 15(a) are changed to 8 bit value 11111111 using bit manipulation technique and last four bit values are changed to 0 using bitor and bitand operations and make the value 11110000.

Now, text image 15(b) bit value is changed to 8 bit value 11111111 using bit manipulation technique. Then embed first four bits of text image 15(b) to last four bits of Manaal image 15(a) to generate new image 16(a) containing stegogramme which is looks like Manaal image 15(a)

[Fig 16 (a): Manaal Image (extracted)]
Step 1: Get the image “A”.
Step 2: Generate Hilbert Curve to identify targeted region in an Image.
Step 3: Change the Pixel value of Targeted Region in an Image.
Step 4: Get the image ‘B’ to embed.
Step 5: Change the pixel value of image ‘B’ using bit manipulation.
Step 6: Pad the pixel of image ‘B’ to image ‘A’ till all the pixels are embedded to generate stegogramme ‘A₁’
Step 7: End.

[Algorithm 3 – Embedding algorithm to hide image and text using Hilbert Curve]
3.2.2.2 Flowchart for Embedding

[Flowchart 3: Embedding flowchart to hide image and text using Hilbert Curve]

START

START

Get the image A

Generate Hilbert Curve on image A to identify location

Get Image B and change the pixel value

Embed four bits of image B into image A until all pixels are padded to generate new image

Is the image embedded?

Y

Compare new embedded image with original image A

STOP

N
3.2.2.3 Extracting Algorithm

In the extraction process, we take the embedded image 17(a) and recognize targeted region by generating Hilbert Curve image 17(b) and extract last four bits 1 1 1 1 (5678) of on image.

![Man Image (With Stegogramme) and Hilbert Curve](image1.png)

*Fig 17 (a): Man Image (With Stegogramme) (b): Hilbert Curve[15]*

After getting all the values from embedded image 17 (a), join them together so values are changed to 11111111 to extract the hidden text image 18(a) from man image 17(a) and compare original, embedded and extracted image.

Hello
How are You?
Please reply

![Text Image (extracted)](image2.png)

*Fig 18 (a): Text Image (extracted)*
Step 1: Get the embedded image ‘A₁’ containing Stegogramme.

Step 2: Generate Hilbert Curve to define the locations of Stegogramme.

Step 3: Extract the embedded Pixels by changing the value using bit manipulation.

Step 6: Extract the actual message (text) or Image and cover image A₂.

Step 7: End.
3.2.2.4 Flowchart for Extraction

[Flowchart 4: Extraction flowchart to hide image and text using Hilbert Curve]

START

Get the image $A_1$ containing stegogramme

Generate Hilbert Curve on image $A_1$ to identify location

Get four bits of Image $A_1$ and join them to get the embedded image

Does new embedded image contain stegogramme?

Compare original, embedded and extracted image

STOP
3.2.3 Steganography of Text and Images using RGB Channel of Image

3.2.3.1 Embedding algorithm

Colour digital images are made of pixels, and pixels are made of combinations of primary colors. A Colour of Image belongs to three different channels called Red, Green and Blue respectively. In this algorithm, we differentiate all three channels of both images; tower image 19(a) and sunflower image 19(b) and generate Mandelbrot Fractal image 19(c) to identify the targeted locations for embedding.

![Fig 19 (a): Tower Image (Cover/Carrier) (b): Sunflower Image (Image to Hide) (c): Mandelbrot Fractal](image)

Tower image 19(a) value is changed to 8 bit 11111111 using bit manipulation technique and changed the last four bits to 0 using bitor and bitand and converted to 11110000 in all three channels.

Sunflower image 19(b) is split in all three channels Red, Green and Blue respectively to changed the value to 8 bit 11111111 using bit manipulation technique and joins the first four bit to last four bit values of tower image 19(a) in all three channels and merge them to generate a new Tower1 image 20(a) containing stegogramme and compare original and embedded image.
3. RESEARCH METHODOLOGY

(a) [Fig 20(a): Tower1 Image(With Stegogramme)]

- Step 1: Get the image “A”.
- Step 2: Split Image ‘A’ in three channels Red, Green and Blue Respectively.
- Step 3: Generate the Mandelbrot Fractal to defined location for embedding.
- Step 4: Change the Pixel value of targeted region on all three Channels.
- Step 5: Get the Image ‘B’ to embed.
- Step 6: Split Image ‘B’ in three channels Red, Green and Blue Respectively.
- Step 7: Change the Pixel value of all three channels.
- Step 8: Join the pixels of image ‘B’ with image ‘A’ as Channel wise using bit manipulation.
- Step 9: End

[Algorithm 5 – Embedding algorithm to hide image and text using RGB channel of Image]
3.2.3.2 Flowchart for Embedding

[Flowchart 5: Embedding flowchart to hide image and text using RGB channel of image]
3.2.3.3 Extracting Algorithm

In extraction process, we take the embedded image 21(a) and recognize the targeted region by generating Mandelbrot fractal image 21 (b).

![Tower Image (With Stegogramme)](image1) ![Mandelbrot Fractal](image2)

(Fig 21 (a): Tower1 Image (With Stegogramme) (b): Mandelbrot Fractal)

After that, extract last four bits from image 21(a) from all three channels and add them together to extract the hided sunflower image 22(a) and compare original, embedded and extracted image.

![Sunflower Image (extracted)](image3)

(Fig 22 (a): Sunflower Image (extracted))
3. RESEARCH METHODOLOGY

[Algorithm 6 – Extracting algorithm to hide image and text using RGB channel of Image]

Step 1: Get the embedded image ‘A1’ containing

Stegogramme.

Step 2: Generate Mandelbrot to define the locations of

Stegogramme.

Step 3: Split Image ‘A1’ in three channels Red, Green

and Blue respectively

Step 4: Change the pixel value of all three channels and

join all three channels to get the original

image.

Step 5: Extract the actual message (text) or Image and

cover image A2.

Step 6: Display and Compare image A2 with image A1

Step 7: End.
3.2.3.4 Flowchart for Extraction

[Flowchart 6: Extraction flowchart to hide image and text using RGB channel of image]

START

Get the image $A_1$ containing stegogramme and split them in three different channels

Red | Green | Blue

Generate Mandelbrot Fractal on image $A_1$ to identify location

Get four bits from Image $A_1$ Red channel  
Get four bits from Image $A_1$ Blue channel  
Get four bits from Image $A_1$ Green channel

Merge them to get the embedded image

Does new embedded image contain stegogramme?

Y

Compare original, embedded and extracted image

STOP

N

Find Embedded Value
3.3 SUMMARY

In this section we have described three different algorithms and flowcharts for embedding and extracting i.e. Steganography of Text and Image using Mandelbrot Fractal, Steganography of Text and Image using Hilbert Curve, Steganography of Text and Image using RGB Channel of Image respectively using MATLAB, we have generated gray scale and RGB histogram for different images to check the distribution of data and to analyze mean and standard deviation to check the intensity of pixels.
REFERENCES