ABSTRACT

The objective of steganography is the art of hiding the existence of data in another transmission medium to achieve secret communication. It does not replace cryptography but rather boosts the security using its obscurity features.

In this project secret data is embedded within skin region of images that will provide an excellent secure location for data hiding. For this skin tone detection is performed using HSV color space. Secret data embedding is performed using frequency domain approach DWT. Secret data is hidden in one of the high frequency sub-band of DWT by tracing skin pixels in that sub-band.

Different steps of data hiding are applied by cropping an image interactively. Cropping results into enhanced security than hiding data without cropping i.e. in whole image, so cropped region works as a key at decoding side.

This project is designed to develop secure communication by using steganography technique. Implementing this project in military application will improve the quality of secret communication.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

- **HSV** Hue, Saturation and Values
- **DWT** Discrete Wavelet Transform
- **LSB** Least Significant Bit
- **HVS** Human Visual System
- **RGB** Red, Green and Blue
- **IDWT** Inverse Discrete Wavelet Transform
- **PSNR** Peak Signal to Noise Ratio
- **MSE** Mean Square Error
- **GUI** Graphical User Interface
- **OPA** Optimum Pixel Adjustment
- **JPEG** Joint Photographic Experts Groups
- **PNG** Portable Network Group

CHAPTER 1

INTRODUCTION

In this highly digitalized world, the internet serves as an important role for data transmission and sharing. However, since it is a world wide and publicized medium, some confidential data might be stolen, copied, modified, or destroyed by an unintended observer. Therefore security problems are essentially an issue. So to overcome security problems cryptography is used. Although encryption achieves certain effects, they make the secret messages unreadable and unnatural or meaningless. These unnatural messages usually attract some unintended observer's attention. This is the reason a new steganography method is modified and implemented in this project.

1.1 STEGANOGRAPHY

Steganography means to hide secret information into image. Digital images are ideal for hiding secret information. An image containing a secret message is called a cover image.

First, the difference of the cover image and the stego - image should be visually unnoticeable. The embedding itself should draw no extra attention to the stego - image so that no hackers would try to extract the hidden message illegally.

Second, the message hiding method should be reliable. It is impossible for someone to extract the hidden message if she/he does not have a special extracting method and a proper secret key.

1.1.1 STEGANOGRAPHY IN HISTORY

Steganography comes from Greek and means “covered writing.” The ancient Greeks wrote text on wax-covered tablets. To pass a hidden message, a person would scrape off the wax and write the message on the underlying wood. He/she would then once again cover the wood with wax so it appeared unused.

1.1.2 STEGANOGRAPHY IN DIGITAL AGE

Steganography is the art of secret communication. Its purpose is to hide the very presence of communication as opposed to cryptography whose goal is to make communication unintelligible to those who do not posses the right keys.

Digital images, videos, sound files, and other computer files that contain perceptually irrelevant or redundant information can be used as “covers” or carriers to hide secret messages.

After embedding a secret message into the cover-image, a so-called stego-image is obtained. It is important that the stego-image does not contain any easily detectable artifacts due to message embedding.

For example, one should not use computer art, charts, images with large areas of uniform color, images with only a few colors, and images with a unique semantic content, such as fonts.

1.1.3 CRYPTOGRAPHY VS STEGANOGRAPHY

Cryptography is the science of encrypting data in such a way that no body can understand the encrypted message, whereas in steganography the existence of data is conceived means its presence cannot be noticed.

The information to be hidden is embedded into the cover object which can be text, image, audio or video so that the appearance of cover object doesn’t vary even after the information is hidden.

Information to be hidden + cover object = stego object.

To add more security the data to be hidden is encrypted with a key before embedding. To extract the hidden information one should have the key. A stego object is one, which looks exactly same as cover object with hidden information.

1.1.4 STAGANOGRAPHY VS WATERMARKING

Watermarking is another branch of steganography it is mainly used to restrict the piracy in digital media in steganography the data to be hidden is not at all related to the cover object, here our main intention is secret communication.

In watermarking the data to be hidden is related to the cover object it is extended data or attribute of the cover object, here our main intention is to stop piracy of digital data. Steganography is a very powerful tool because, as the stated above, it can be very difficult to detect.

1.2 STEGOSYSTEM

The stegosystem is conceptually similar to the cryptosystem.

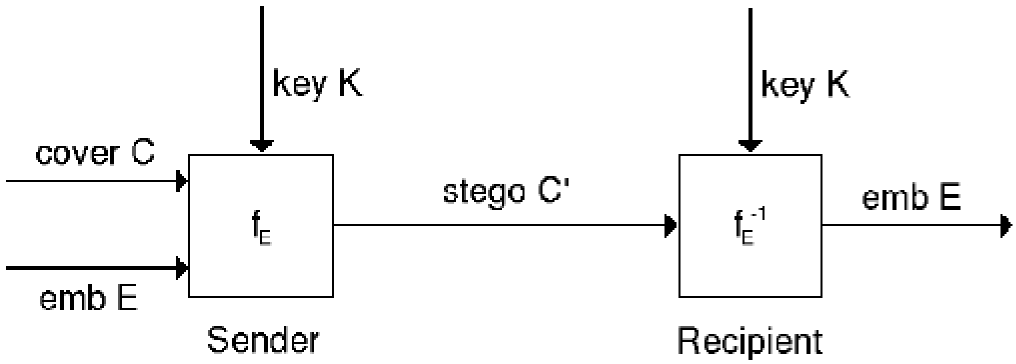


Figure 1.1 Block diagram for stegosystem

emb: The message to be embedded. It is anything that can be represented as a bit stream (an image or text).

cover: Data/Medium in which emb will be embedded.

stego: Modified version of the cover that contains the embedded message,

emb.key: Additional data that is needed for embedding & extracting.

f_E: Steganographic function that has cover, emb & key as parameters.

Steganography refers to the science of “invisible” communication. Unlike cryptography, where the goal is to secure communications from an eavesdropper, steganographic techniques strive to hide the very presence of the message itself from an observer.

The main goal of steganography is to communicate securely in a completely undetectable manner.

Here is a graphical version of the stegosystem:

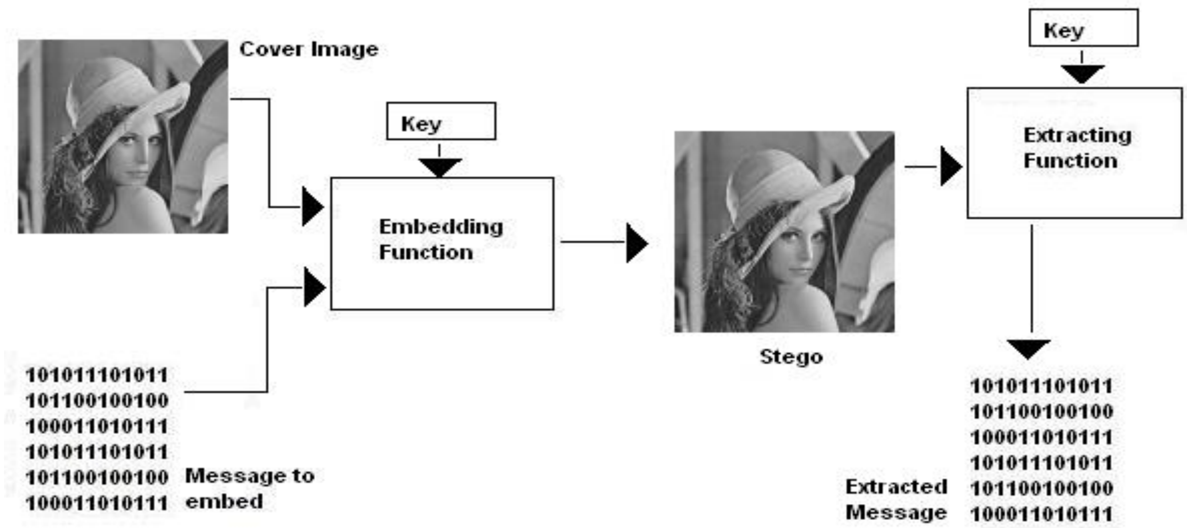


Figure 1.2 Graphical version of stegosystem

1.3 EXISTING METHOD & ITS DRAWBACKS

Steganography in Spatial Domain

- This is a simplest steganographic technique that embeds the bits of secret message directly into the least significant bit (LSB) plane of the cover image.
- In a gray level image, every pixel consists of 8 bits.
- The basic concept of LSB substitution is to embed the confidential data at the rightmost bits (bits with the smallest weighting) so that the embedding procedure does not affect the original pixel value greatly.

Drawbacks of Existing Method

- This method is easy and straightforward but this has low ability to bear some signal processing or noises.
- And secret data can be easily stolen by extracting whole LSB plane.

CHAPTER 2

LITERATURE REVIEW

Proposed method introduces a new method of embedding secret data within skin as it is not much sensitive to HVS. This takes advantages of biometric features such as skin tone, instead of embedding data anywhere in image, data will be embedded in selected regions.

At first skin tone detection is performed on input image using HSV color space. Secondly cover image is transformed in frequency domain. This is performed by applying Haar-DWT. Finally secret data embedding is performed in one of high frequency sub-band by tracing skin pixels in that band.

Before performing all steps cropping on input image is performed and then in only cropped region embedding is done, nor in whole image. Cropping enhances security since cropped region works as a key at decoding side.

2.1 BLOCK DIAGRAM

Embedding

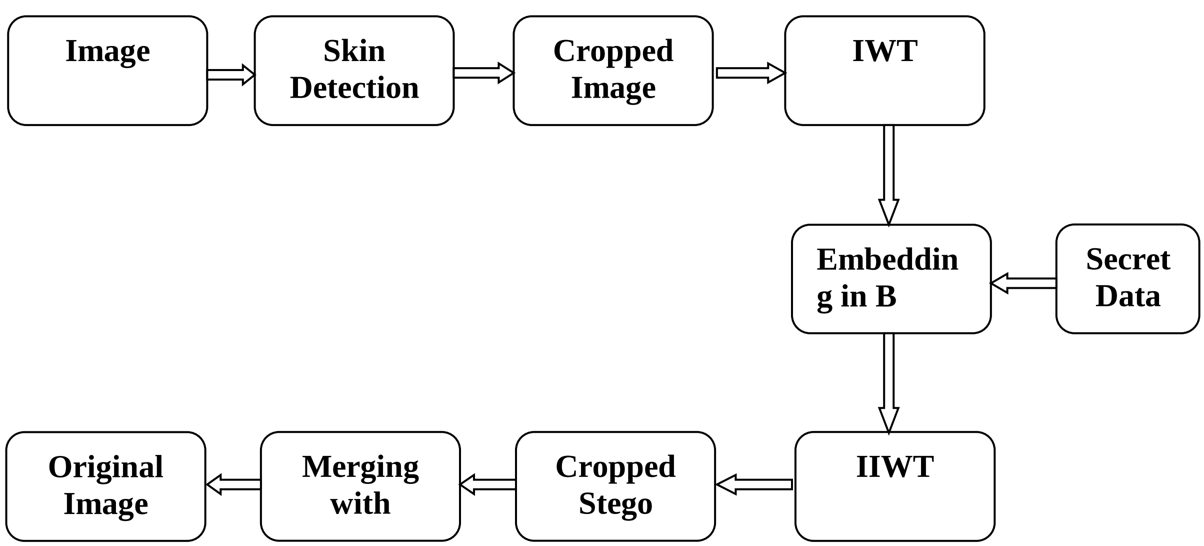


Figure 2.1 Block diagram for embedding process

Extraction

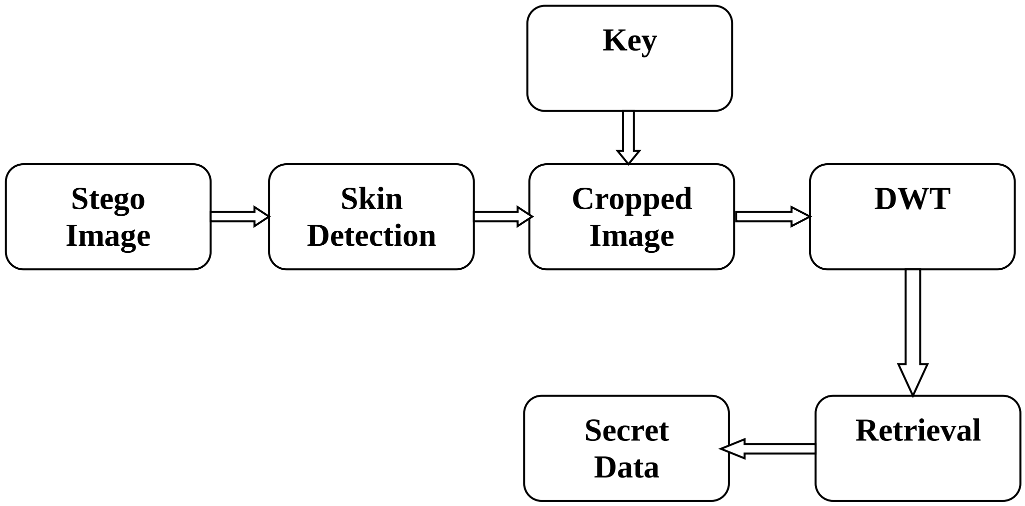


Figure 2.2 Block diagram for extraction process

2.1.1 ADVANTAGES

- By embedding data in only certain region (here skin region) and not in whole image security is enhanced.
- Also image cropping concept introduced, maintains security at respectable level since no one can extract message without having value of cropped region.
- It increases the quality of stego because secret messages are embedded in high frequency sub-bands which human eyes are less sensitive to.
- The proposed approach provides fine image quality.

2.2 HSV COLOR MODEL

The HSV stands for the Hue, Saturation and value based on the artists (Tint, Shade and Tone). The coordinate system in a hexacone in Figure 4 is a view of the HSV color model. HSV color wheel is used to pick the desired color.

Hue is represented by circle in the wheel. A separate triangle is used to represent saturation and value. The horizontal axis of the triangle indicates value and vertical axis represent saturation.

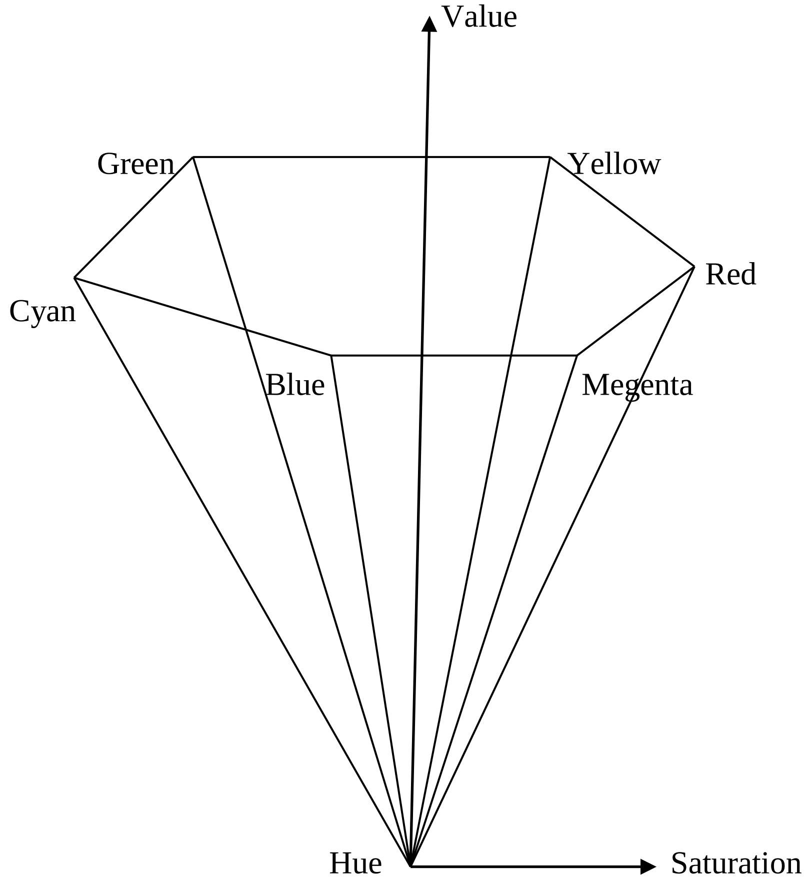


Figure 2.3 HSV coordinate system

HUE

In HSV, hue represents color. In this model hue is an angle from 0 to 360 degree.

Angle	Color
0 - 60	Red
60 - 120	Yellow
120 - 180	Green
180 - 240	Cyan
240 - 300	Blue
300 - 360	Megenta

Table 1

SATURATION

Saturation indicates the range of gray in the color space. It ranges from 0 to 100%. Sometimes the value is calculated from 0 to 1.

When the value is ‘0’, the color is gray and when value is ‘1’, the color is primary color. A faded color is due to lower saturation level, which means the color contains in gray.

VALUE

Value is the brightness of color and varies with color saturation. It ranges from 0 to 100. When the value is ‘0’ the color space will be totally black. With the increase in the value, the color space brightness up and shows various colors.

2.2.1 RGB TO HSV CONVERSION

The obtainable HSV color lies within a triangle whose vertices are defined by the three primary colors in RGB space:

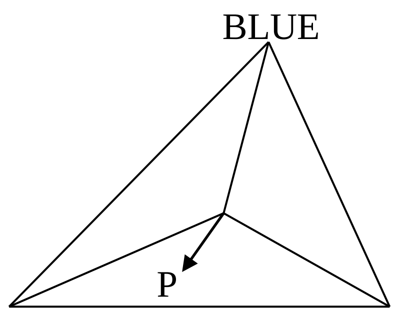


Figure 2.4 HSV to RGB color space

The hue of point P is the measured angle between the line connecting P to the triangle centre and line connecting RED point to the triangle center. The saturation of the point P is the distance between P and triangle center. The value (intensity) of the point P is represented as height on a line perpendicular to the triangle passing through its center. The grayscale points are situated onto the same line and the conversion formula is as follows:

$$\left.\begin{array}{l} H=\text{cos inverse}\left\{\dfrac{1/2[(R-G)+(R-B)]}{\sqrt{(R-G)\text{ pow }2+(R-B)(G-B)}}\right\} \\[2ex] S=1-\dfrac{3}{R+G+B}\,[\min(R, G, B] \\[2ex] V=1/3(R+G+B) \end{array}\right\} \quad (1)$$

2.3 SKIN COLOR TONE DETECTION

A skin detector typically transforms a given pixel into an appropriate color space and then uses a skin classifier to label the pixel whether it is a skin or a non-skin pixel. A skin classifier defines a decision boundary of the skin color class in the color space.

Important challenges in skin detection are to represent the color in way that is invariant or at least insensitive to changes in illumination. Another challenge comes from the fact that many objects in the real world might have skin-tone color. This causes the skin detector to have much false detection in the background.

The simplest way to decide whether a pixel is skin color or not, is to explicitly define a boundary. RGB matrix of the given color image converted into different color space to yield distinguishable region of skin or near skin-tone. Color space used for skin detection is HSV.

Sobottaka and pitas defined a face location based on HSV. They found that human flesh can be an approximation from a sector out of a hexagon with the constraints:

$S_{min}=0.23$, $S_{max}=0.68$, $H_{min}=0^{\circ}$ and $H_{max}=50^{\circ}$

2.4 IMAGE HISTOGRAMS

The histogram of an image is a plot or graph drawn between gray level values (0-255) in the X-axis and the number of pixels having the corresponding gray-levels in the Y-axis.

The histogram of a digital image with L total possible intensity levels in the range [0, G] is defined as the discrete function

$$h(r_k) = n_k$$

2.5 WAVELET TRANSFORM

The basic idea of the wavelet transform is to represent any arbitrary function as a superposition of a set of such wavelets or basis functions. These basis functions or baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by dilations or contractions (scaling) and translations (shifts). They have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Many new wavelet applications such as image compression, turbulence, human vision, radar, and earthquake prediction are developed in recent years. In wavelet transform the basis functions are wavelets. Wavelets tend to be irregular and symmetric. All wavelet functions, w (2kt - m), are derived from a single mother wavelet, w(t). This wavelet is a small wave or pulse like the one shown in Figure 2.5.

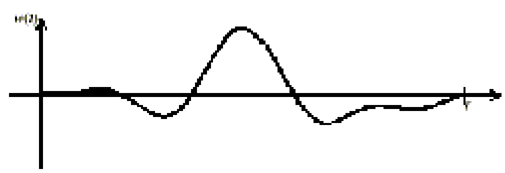


Figure 2.5 Mother wavelet

2.5.1 DISCRETE WAVELET TRANSFORM

Calculating wavelet coefficients at every possible scale is a fair amount of work, and it generates an awful lot of data. If the scales and positions are chosen based on powers of two, the so-called dyadic scales and positions, then calculating wavelet coefficients are efficient and just as accurate. This is obtained from discrete wavelet transform (DWT).

2.5.2 WAVELET RECONSTRUCTION

The reconstruction of the image is achieved by the inverse discrete wavelet transform (IDWT). The values are first up sampled and then passed to the filters. This is represented as shown in Figure 2.6.

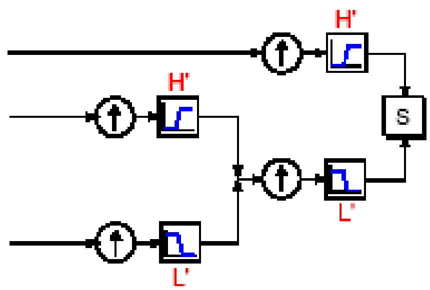


Figure 2.6 Wavelet Reconstruction

The wavelet analysis involves filtering and down sampling, whereas the wavelet reconstruction process consists of up sampling and filtering. Up sampling is the process of lengthening a signal component by inserting zeros between samples.

2.5.3 RECONSTRUCTING APPROXIMATION AND DETAILS

It is possible to reconstruct the original signal from the coefficients of the approximations and details. The process yields a reconstructed approximation which has the same length as the original signal and which is a real approximation of it.

The reconstructed details and approximations are true constituents of the original signal. Since details and approximations are produced by downsampling and are only half the length of the original signal they cannot be directly combined to reproduce the signal.

It is necessary to reconstruct the approximations and details before combining them. The reconstructed signal is schematically represented as in Figure 2.7

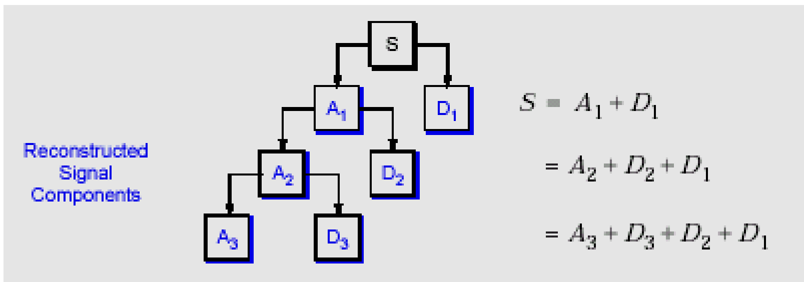


Figure 2.7 Reconstruction signal sampling

2.6 HAAR-DWT

The frequency domain transform applied in this project is Haar-DWT, the simplest DWT. A 2-dimensional Haar-DWT consists of two operations: One is the horizontal operation and the other is the vertical one. Detailed procedure of a 2-D Haar – DWT are described as follows:

Step 1: At first, scan the pixels from left to right in horizontal direction. Then, perform the addition and subtraction operation on the neighboring pixels.
Store the sum on the left and the difference on the right.

The pixel sums represent the low frequency part (denoted as symbol L) while the pixel differences represent the high frequency part of the original image (denoted as symbol H).

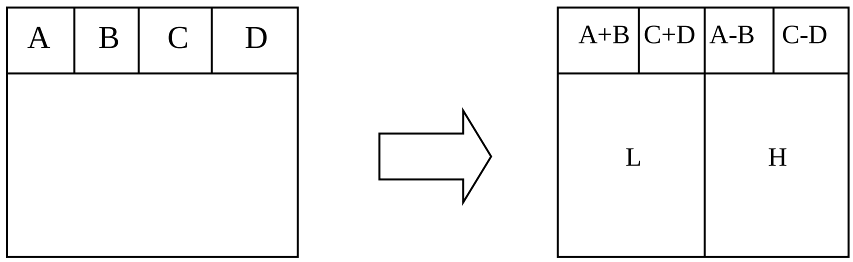


Figure 2.8 First Process in Haar-DWT

Step 2: Secondly, scan the pixels from top to bottom in vertical direction. Perform the addition and subtraction operation on neighboring pixels and then store the sum on the top and the difference on the bottom as illustrated in figure 2.9. Repeat this operation until all the columns are processed. Finally we will obtain 4 sub-bands denoted as LL, HL, LH and HH respectively. The LL sub-band is the low frequency portion and hence looks very similar to the original image.

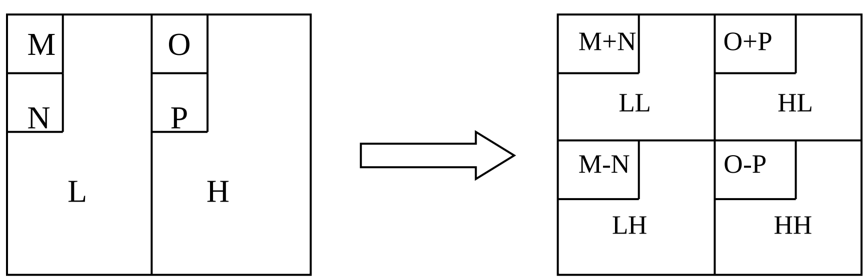


Figure 2.9 Second Process in Haar-DWT

2.6.1 2-D TRANSFORM HEIRARCHY

The 1-D wavelet transform can be extended to a two-dimensional (2-D) wavelet transform using separable wavelet filters.

With separable filters the 2-D transform can be computed by applying a 1-D transform to all the rows of the input, and then repeating on all of the columns.

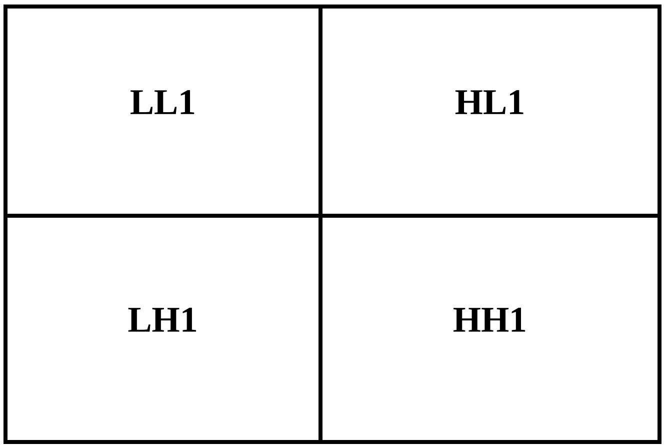


Figure 2.10(a) One – level decomposition

The original image of a one-level (K=1), 2-D wavelet transform, with corresponding notation is shown in Figure 2.10(a). The example is repeated for a three-level (K =3) wavelet expansion in Figure 2.10(b). In all of the discussion *K* represents the highest level of the decomposition of the wavelet transform.

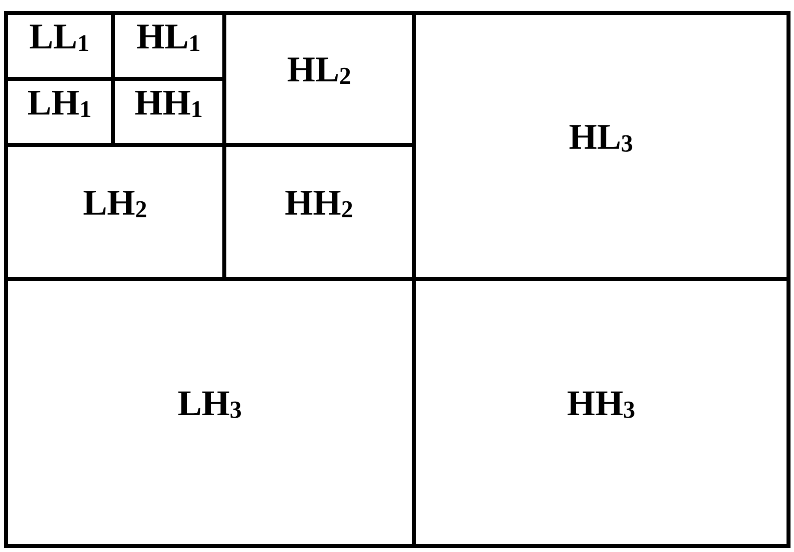


Figure 2.10(b) Three level decomposition

The 2-D sub-band decomposition is just an extension of 1-D sub-band decomposition. The entire process is carried out by executing 1-D sub-band decomposition twice, first in one direction (horizontal), then in the orthogonal (vertical) direction. For example, the low-pass sub-bands (Li) resulting from the horizontal direction is further decomposed in the vertical direction, leading to LLi and LHi sub-bands.

Similarly, the high pass sub-band (Hi) is further decomposed into HLi and HHi. After one level of transform, the image can be further decomposed by applying the 2-D sub-band decomposition to the existing LLi sub-band. This iterative process results in multiple “transform levels”.

An example of three-level decomposition into sub-bands of the image is illustrated in Figure 2.10(c).

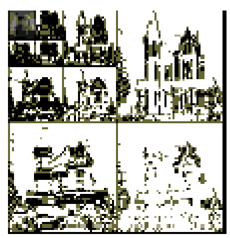


Figure 2.10(c) Three-level image decomposition

To obtain a two-dimensional wavelet transform, the one-dimensional transform is applied first along the rows and then along the columns to produce four sub-bands: low-resolution, horizontal, vertical, and diagonal.

(The vertical sub-band is created by applying a horizontal high-pass, which yields vertical edges.) At each level, the wavelet transform can be reapplied to the low-resolution sub-band to further decorrelate the image. illustrates the image decomposition, defining level and sub-band conventions used in the algorithm. The final configuration contains a small low-resolution sub-band.

In addition to the various transform levels, the phrase level 0 is used to refer to the original image data.

2.6.2 WAVELET COMPUTATION

In order to obtain an efficient wavelet computation, it is important to eliminate as many unnecessary computations as possible.

The one-dimensional wavelet transform is computed by separately applying two analysis filters at alternating even and odd locations. The inverse process first doubles the length of each signal by inserting zeros in every other position, then applies the appropriate synthesis filter to each signal and adds the filtered signals to get the final reverse transform.

2.6.3 ALGORITHMS AND TRANSFORMATIONS

Another steganography method is to hide data in mathematical functions that are in compression algorithms. Two functions are Discrete Cosine Transformation (DCT) and Wavelet Transformation. The DCT and wavelet functions transform data from one domain into another. The DCT function transforms that data from a spatial domain to a frequency domain.

The DCT function:

$$F(u,v) = \frac{\Lambda(u)\Lambda(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\frac{(2i+1)\cdot u\pi}{16}\cdot\cos\frac{(2j+1)\cdot v\pi}{16}\cdot f(i,j)$$

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & for\ \xi = 0 \\ 1 & otherwise \end{cases}$$

To Encode the Hidden Data:

- Take the DCT or wavelet transform of the cover image.
- Find the coefficients below a certain threshold.
- Replace these bits with bits to be hidden (can use LSB insertion).
- Take the inverse transform.
- Store as regular image.

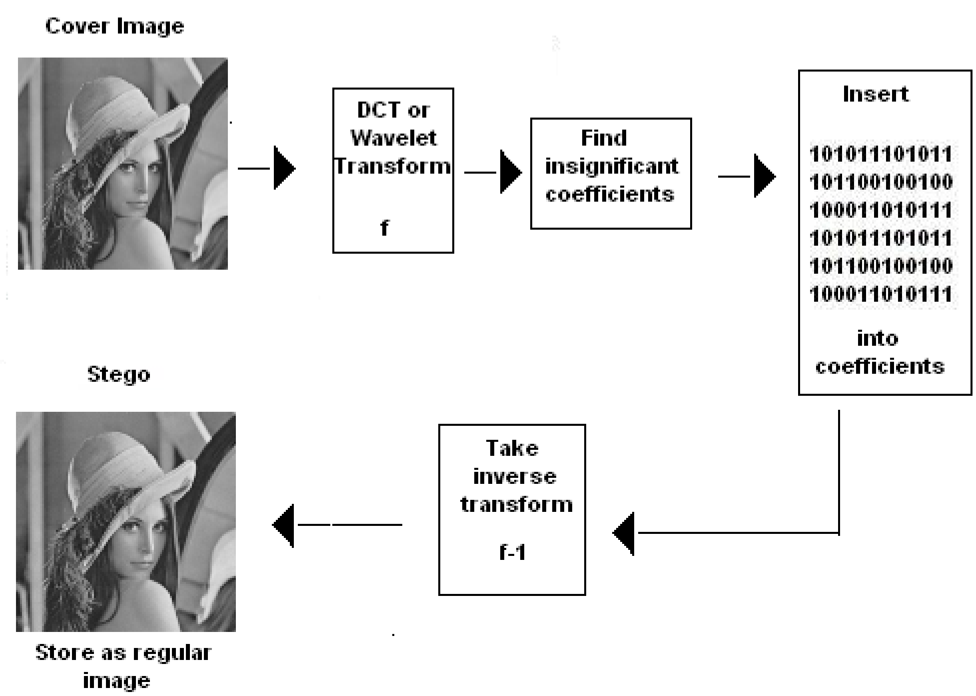


Figure 2.11 Block Diagram for Encoding

- To Decode the Hidden Data:

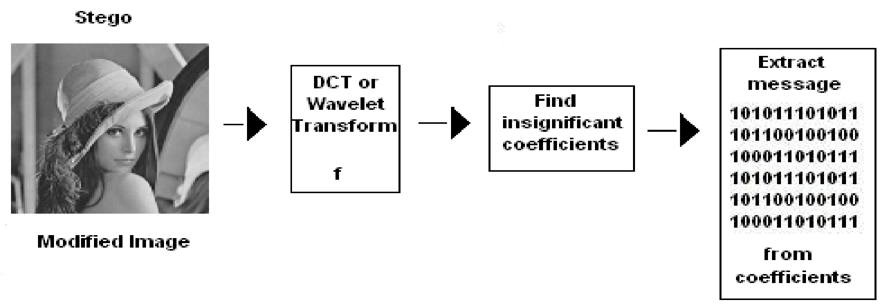


Figure 2.12 Block Diagram for Decoding

- Take the transform of the modified image.
- Find the coefficients below a certain threshold.
- Extract bits of data from these coefficients.
- Combine the bits into an actual message.

2.7 LSB SUBSTITUTION

The most frequently used steganography method is the technique of LSB substitution. In a gray-level image, every pixel consists of 8 bits. One pixel can hence display 2^8=256 variations. The weighting configuration of an 8-bit number is illustrated in figure.

Figure 2.13 LSB Substitution

The basic concept of LSB substitution is to embedded the confidential data at the rightmost bits (bits with the smallest weighting) so that the embedding procedure does not affect the original pixel value greatly. The mathematical representation for LSB method is:

Xi'=Xi-Xi mod 2^k + Mi

Xi'=i th pixel value of stego-image

Xi=original cover image

K= number of LSB to be substituted

The extraction process is to copy the k-rightmost bits directly. Mathematically the extracted message is represented as

Mi=Xi' mod 2^k

An ideal steganographic technique embeds message information into a carrier image with virtually imperceptible modification of the image. The objective of steganography is a method of embedding an additional information into the digital contents, that is undetectable to listeners.

2.7.1 ADVANTAGES OF LSB

A major advantage of the LSB algorithm is it is quick and easy. There has also been steganography software developed which work around LSB color alterations via palette manipulation. LSB insertion also works well with gray-scale images.

2.8 EMBEDDING PROCESS

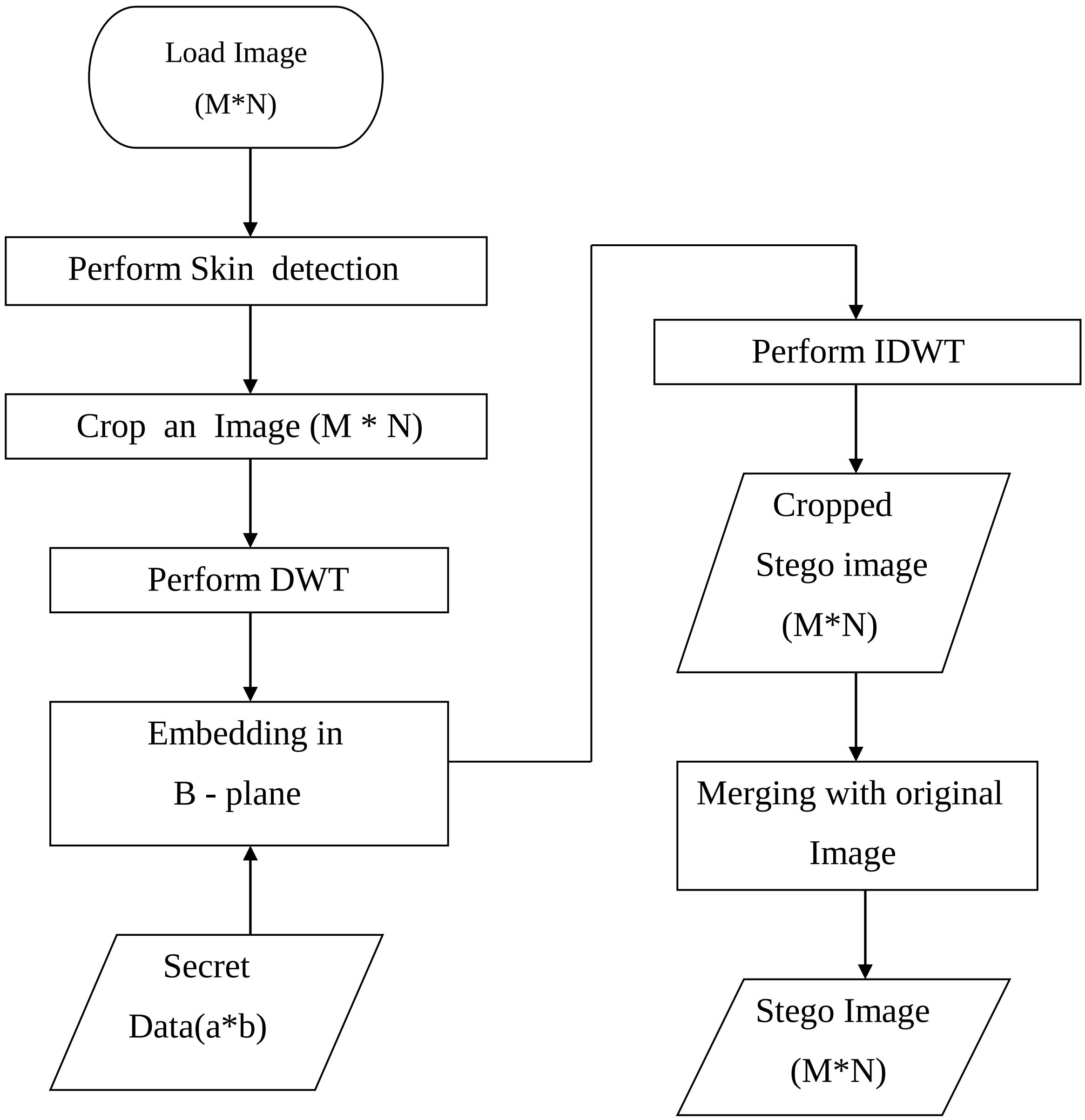


Figure 2.14 Flow chart for Embedding process

Step 1: Once image is loaded, apply skin tone detection on cover image. This will produce the mask image that contains skin and non skin pixels.

Step 2: Cropping is performed interactively on mask image. After that the original image is also cropped of same area. Cropped area must be in exact square form and it should contain the skin region.

Step 3: Apply DWT to only cropped area not whole image. Payload of image to hold the secret data is determined based on number of pixels present in one of high frequency sub-band in which data will be hidden.

Step 4: Secret data is embedded in one of sub-band that we obtained earlier by tracing skin pixels in that sub-band. Other than low frequency any high frequency sub-band can be selected for embedding. Because embedding in low frequency sub-band affects image quality greatly.

Step 5: Perform IDWT to combine 4 sub-bands.

Step 6: Cropped stego image is merged with the original image to get the stego image.

2.9 EXTRACTIOsN PROCESS

Stego image is input to extraction process. We must need the value of cropped area to retrieve data. Suppose cropped area value is stored in 'rect' variable that is same as in encoder. So this acts as a key at the decoder side. All steps of decoder are opposite to encoder**.** By tracing skin pixels in high frequency sub-band of DWT secret data is retrieved.

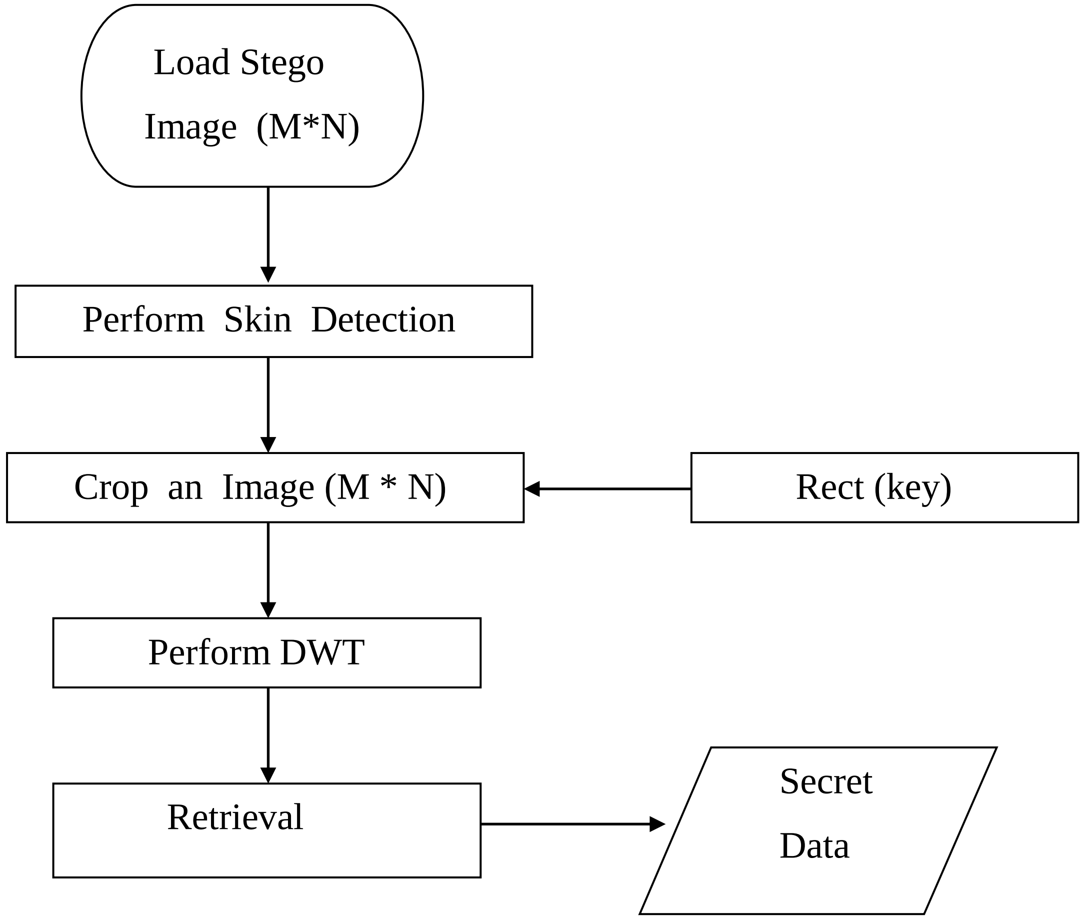


Figure 2.15 Flow for extraction process

QUALITY OF STEGO IMAGE

PSNR = 10 log (255^2 / MSE)

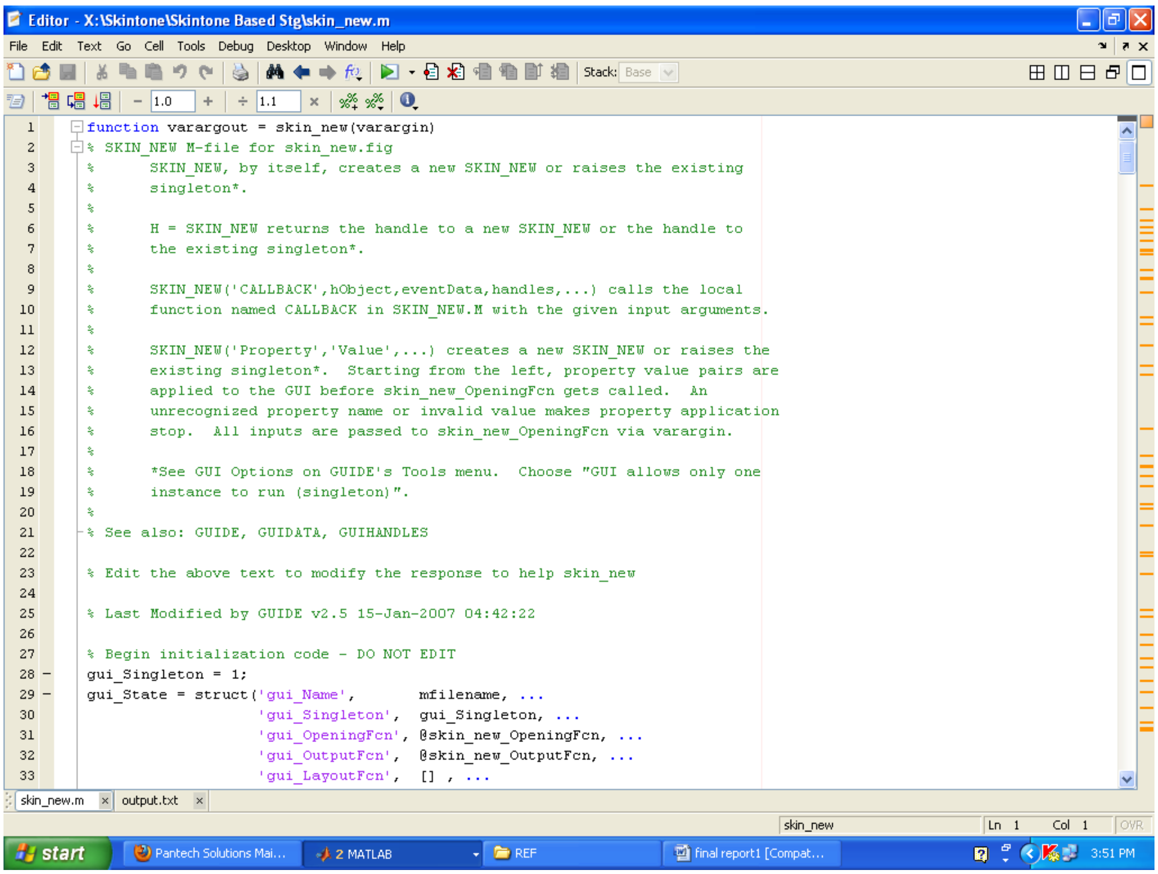
$$MSE = (1/(M*N)) \sum_{i=1}^{M} \sum_{j=1}^{N} (input - output)^2$$

2.10 APPLICATIONS

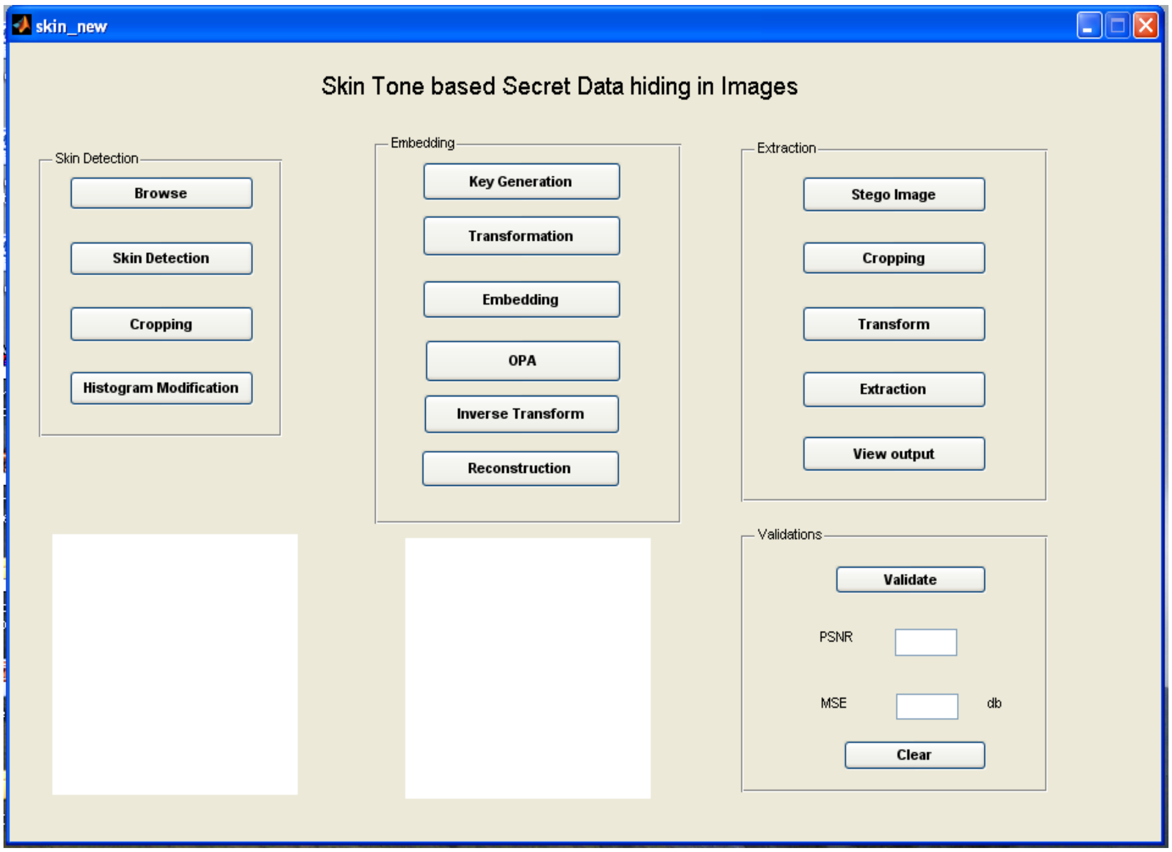
- Confidential communication and secret data storing.
- Protection of data alteration.
- Access control system for digital content distribution.
- Media database system.
- Copyright Protection.
- Military application.

2.11 SIMULATION RESULTS

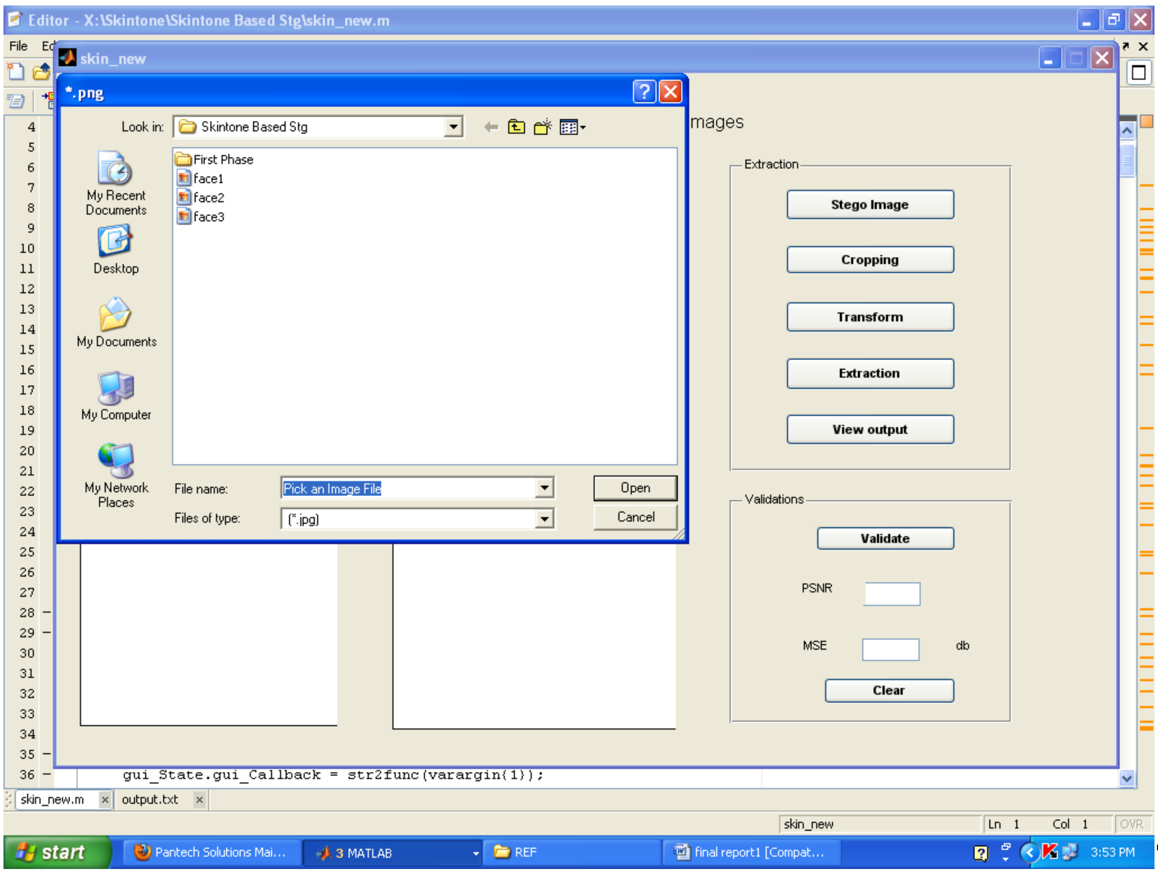
(1) CREATION OF GUI



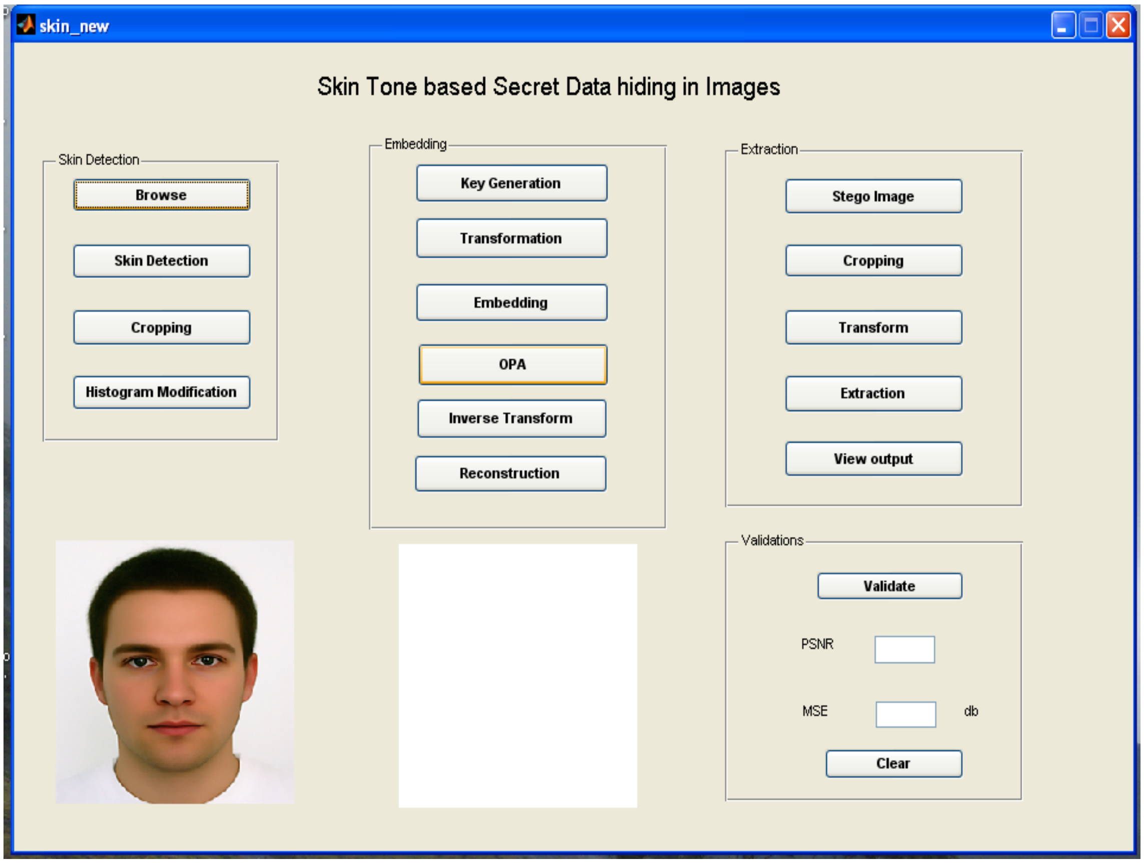
(2) CREATED GUI



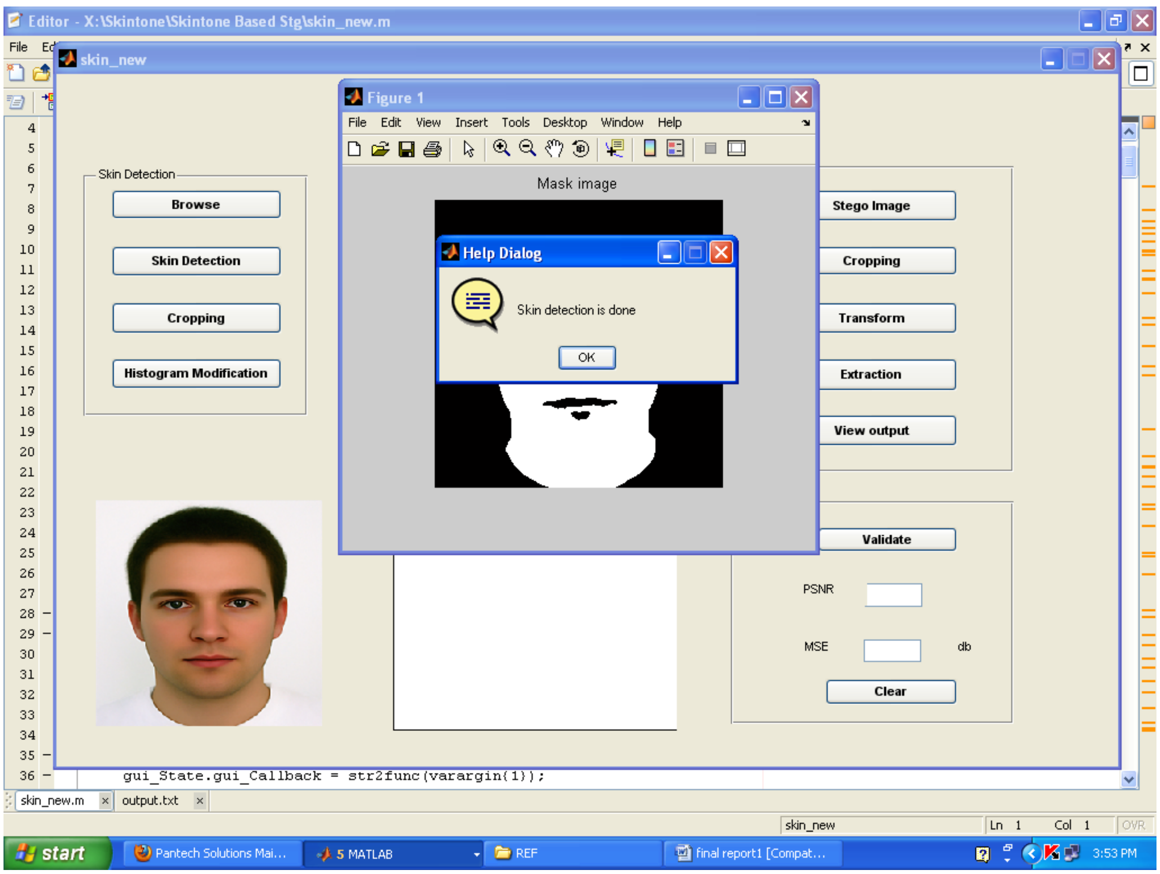
(3) SELECTION OF IMAGE FILE



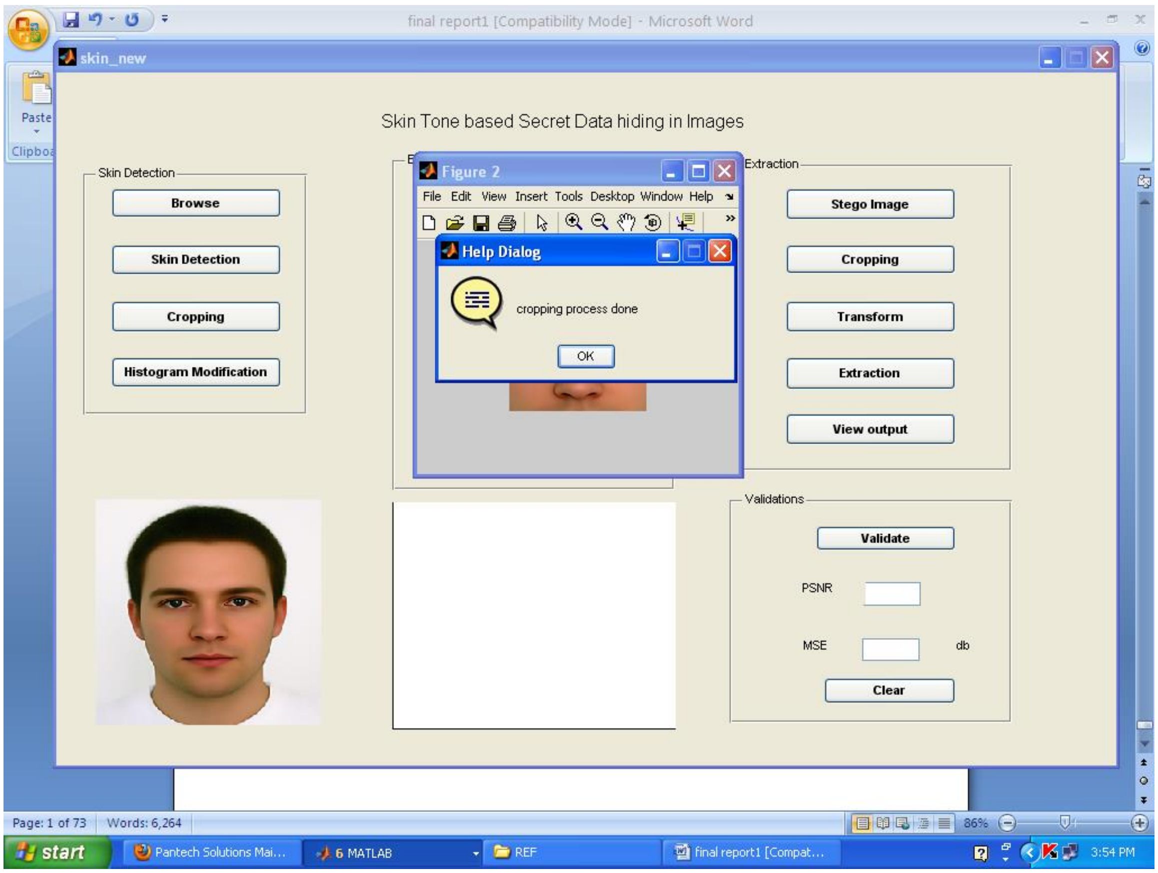
(5) SHOW A IMAGE USING GUI



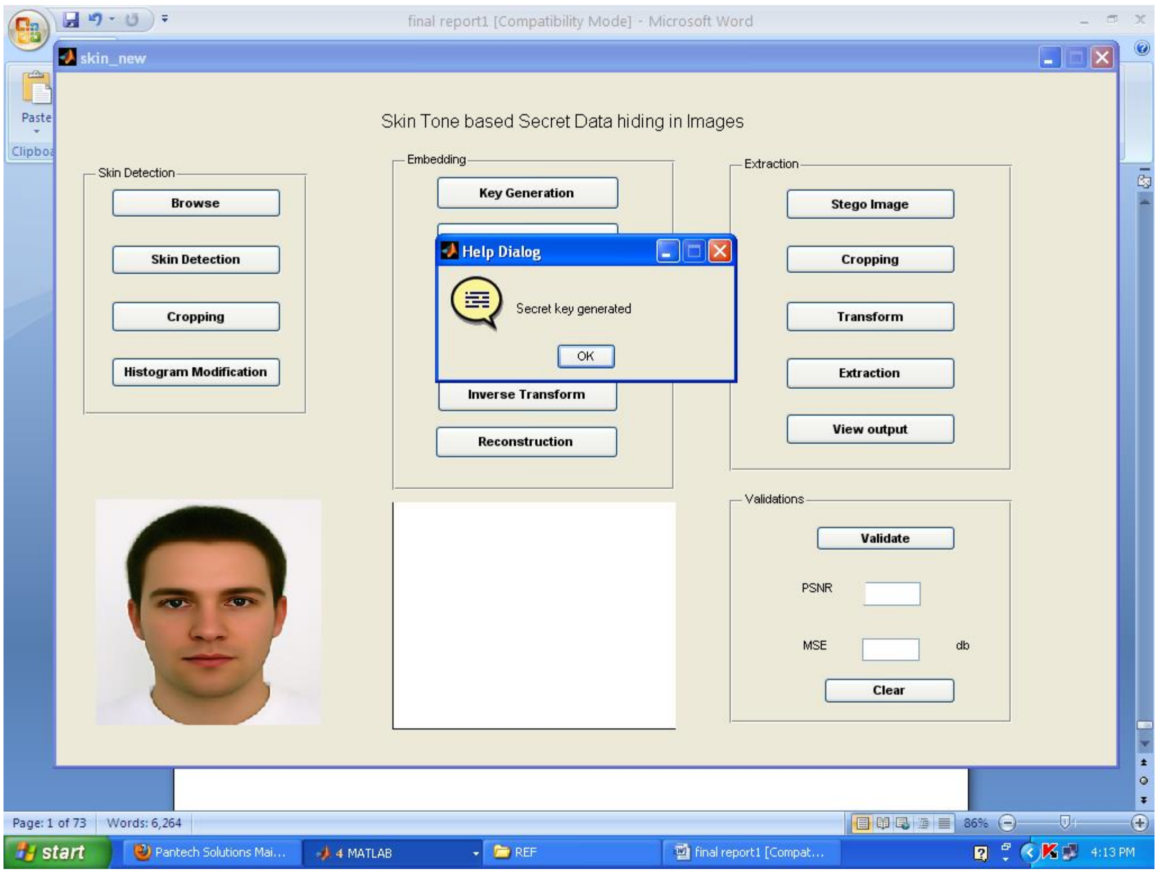
(6) SKIN DETECTED IN AN IMAGE



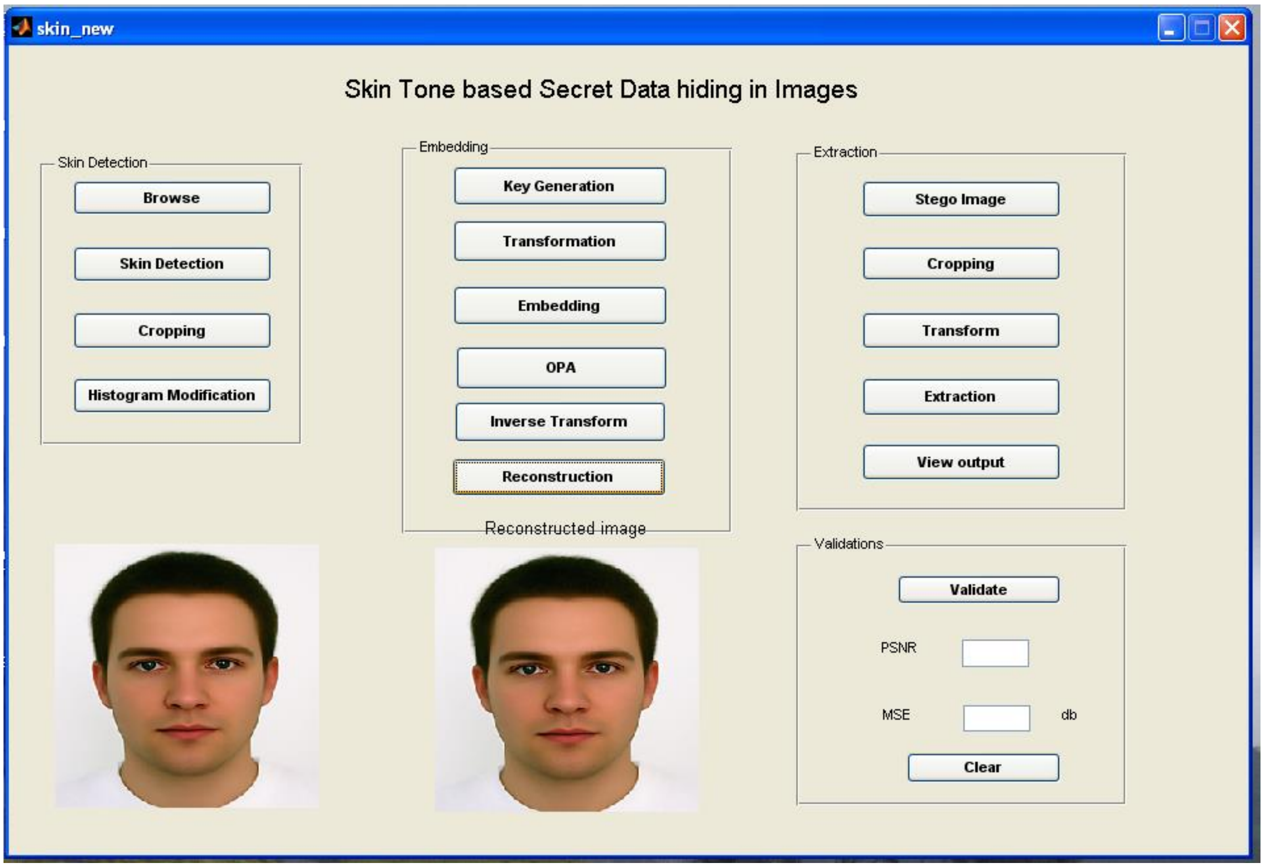
(7) CROPPED IMAGE



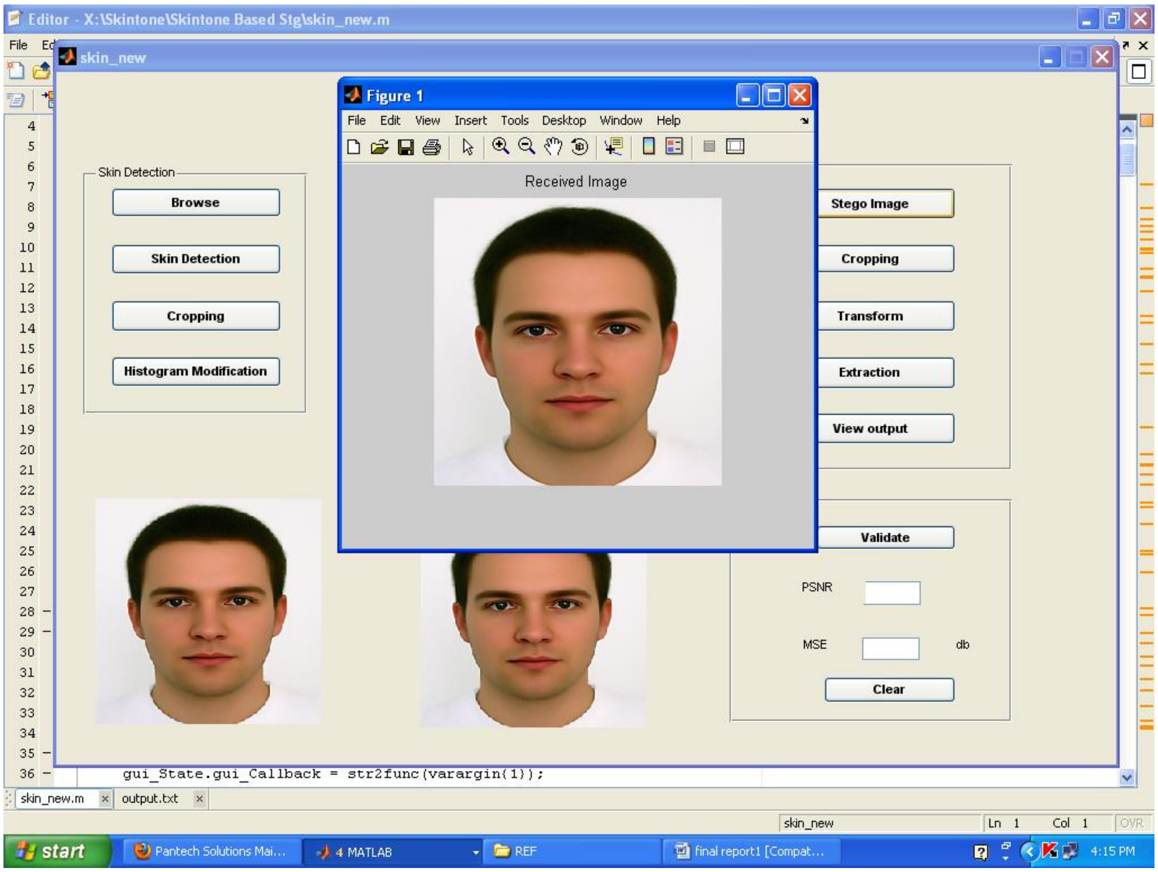
(8) SECRET KEY GENERATED IN AN IMAGE



(9)   STEGANOGRAPHY PROCESS DONE



(10) OUTPUT


Editor - X:\Skintone\Skintone Based Stg\skin_new.m
skin_new
Figure 1
Received Image
Skin Detection
Browse
Skin Detection
Cropping
Histogram Modification
Stego Image
Cropping
Transform
Extraction
View output
Validate
PSNR
MSE
db
Clear
gui_State.gui_Callback = str2func(varargin{1});
skin_new.m
output.txt
start
4:15 PM

CHAPTER 3

SOFTWARE SPECIFICATION

3.1 MATLAB 7.5

MATLAB (**mat**rix **lab**oratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, and Fortran.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

Variables

Variables are defined with the assignment operator, =. MATLAB is a weakly dynamically typed programming language. It is a weakly typed language because types are implicitly converted. It is a dynamically typed language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects, and that their type can change. Values can come from constants, from computation involving values of other variables, or from the output of a function.

Vectors/matrices

MATLAB is a "Matrix Laboratory", and as such it provides many convenient ways for creating vectors, matrices, and multi-dimensional arrays. In the MATLAB vernacular, a *vector* refers to a one dimensional ($1 \times N$ or $N \times 1$) matrix, commonly referred to as an array in other programming languages. A *matrix* generally refers to a 2-dimensional array, i.e. an $m \times n$ array where m and n are greater than or equal to 1. Arrays with more than two dimensions are referred to as multidimensional arrays.

MATLAB provides a simple way to define simple arrays using the syntax: *init*:*increment*:*terminator*.

Semicolons

Unlike many other languages, where the semicolon is used to terminate commands, in MATLAB the semicolon serves to suppress the output of the line that it concludes

Structures

MATLAB supports structure data types. Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names (field look-ups by name, field manipulations etc). Unfortunately, MATLAB JIT does not support MATLAB structures, therefore just a simple bundling of various variables into a structure will come at a cost.

CHAPTER 4

CONCLUSION

In this project we use skin region of images in DWT domain for embedding secret data. By embedding data in only skin region and not in whole image security is enhanced. Also image cropping concept is introduced, maintain security at respectable level since no one can extract message without having value of cropped region. Features obtained from DWT coefficient are utilized for secret data embedding. This also increases the quality of stego because secret message are embedded in high frequency sub-bands which human eyes are less sensitive to. According to simulation results, proposed approach provides fine image quality.

FUTURE SCOPE

- Integer wavelet transform will be used instead of wavelet transform.
- Secret data can be embede in color space component.
- Other plane also used instead of blue plane.

APPENDICES

PROGRAM

READ COVER IMAGE

```
[file,path] = uigetfile('*.jpg','*.png','Pick an Image File');
if isequal(file,0) | isequal(path,0)
    warndlg('User Pressed Cancel');
else
    a = imread(file);
    a=imresize(a,[256 256]);
    [r c p] = size(a);
   axes(handles.axes1);
    imshow(a);
    handles.a=a;
    guidata(hObject, handles);
end
```

SKIN DETECTION

```
a=handles.a;
HSV=rgb2hsv(a);
h = HSV(:,:,1);
s = HSV(:,:,2);
v = HSV(:,:,3);
[r c p]=size(a);
d=zeros(r,c);
```

```
       if ((h(i,j)<25)&((s(i,j)<0.68)& (s(i,j)>0.10)))
          d(i,j)=1;
    d = medfilt2(d,[3 3]);
     guidata(hObject, handles);
     helpdlg('Skin detection is done');
```

CROPPING

```
a=handles.a;
d=handles.d;
[x y]=find(d==1);
            Min_y = min(y)+30;
            Max_y = max(y) - 30;
            x_tot=[x y];
            x_cen= floor(length(x)/2);
            x_loc = x(x_cen);
            y_loc = y(x_cen);
            Inc_val = y_loc - Min_y;
            New_intr = x_loc-Inc_val;
             New_endr = x_loc+Inc_val;
             New_intc = y_loc-Inc_val;
             New_endc = y_loc+Inc_val
           Cropped_mask=d(x_loc-Inc_val:x_loc+Inc_val,y_loc-
Inc_val:y_loc+Inc_val);
           imshow(Cropped_mask,[]);
           [R C]=size(Cropped_mask);
```

```
M1=mod(R,2);
 if M1==1
     skin_mask = a(New_intr:New_endr-1,New_intc:New_endc-1,:);
 end
[s1 s2] = size(skin_mask);
figure;
 imshow(skin_mask);
 handles.skin_mask=skin_mask;
 guidata(hObject, handles);
 helpdlg('cropping process done');
 handles.a=a;
 handles.d=d;
 handles.New_intr=New_intr;
 handles. New_endr= New_endr;
 handles.New_intc=New_intc;
 handles.New_endc=New_endc;
 handles.s1 = s1;
handles.s2 = s2;
guidata(hObject, handles);
```

EMBEDDING

```
data1=fopen(file,'r');
F=fread(data1);
fclose(data1);
end
```

```
len=length(F);
count=1;
totalbits=8*len;
a=128;
k=1;
[r c]=size(Y);
for i=1:8:r-7;
    for j=1:8:c-7;
        block3=Y(i:i+7,j:j+7);
        for ii=1:8
            for jj=1:8;
                if orig_key(ii,jj)==1;
                    coeff=abs(block3(ii,jj));
                    [ block3(ii,jj),a,k,count]=bitlength(coeff,a,k,F,totalbits,count,len);
        Y(i:i+7,j:j+7)=block3;
        Y=abs(Y);
        if count>totalbits;
    if count>totalbits;
        break;
outpu_t=Y;
figure;
imshow(outpu_t,[]);
title('embedded image');
handles.outpu_t=outpu_t;
helpdlg('Process completed');
```

EXTRACTION

```
YY=handles.YY;
totalbits=handles.totalbits;
orig_key=handles.orig_key;
fil_e=YY;
a=128;
jjj=1;
count=1;
k=0;
[r c]=size(YY);
for i=1:8:r-7;
    for j=1:8:c-7;
        block9=fil_e(i:i+7,j:j+7);
        for ii=1:8
                for jj=1:8;
                if orig_key(ii,jj)==1;
                    coeff=abs(block9(ii,jj));
                    g=coeff;
                    if g>=64;
                        bits=6;
                        h=32;
                    elseif g<64 & g>=32;
                        bits=5;
                        h=16;
                    elseif g<32 & g>=16;
```

```
bits=4;
h=8;
elseif g<16
bits=3;
h=4;
end
l=bits;
 for iii=1:l;
    if bitand(g,h)==h;
        k= bitor(k,a);
    end
    count=count+1;
    a=a/2;
    h=h/2;
        if a<1;
        R(jjj)=k;
        fid=fopen('output.txt','wb');
fwrite(fid,char(R),'char');
        fclose(fid);jjj=jjj+1;
        k=0;
        a=128;
    end
helpdlg('Secret data was obtained in text file');
```

REFERENCES

[1] Ahmed E., Crystal M. and Dunxu H, (2009) "Skin Detection - a short Tutorial", Encyclopedia of Biometrics by springer - verlag Berlin Heidelberg.

[2] Chang, C.C., Chen, T.S and Chung, L.Z., (2002)"A steganographic method based upon JPEG and quantization table modification", Information Sciences, vol.[4], pp.i23-138.

[3] Hung-Ju Lin and Po-Yueh Chen , (2006) "A DWT Based Approach for Image Steganography", International Journal of Applied Science and Engineering,4,3:275-290.

[4] Katzenbeisser. S and petitcolas, F.A.P (2000) "Information hiding Technique for Steganography and Digital Watermarking" ,Artech House, INC.

WEBSITES:

[1] **www.wikipedia.org**

[2] **www.steganography.org**

[3] **www.fact-archive.com**

[4] **www.thefreedictionary.com/extraction**

[5] **ww.programming.itags.org/matlab/15108**