A HIGH CAPACITY 3D STEGANOGRAPHY ALGORITHM

A PROJECT REPORT

Submitted by

S.RAGHAVAN 71206205038

E.MANOJ RAJAN 71206205302

J.GOKUL MURALI 71206205014

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641006

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2010

BONAFIDE CERTIFICATE

Certified that this project report entitled "A High Capacity 3D Steganography Algorithm" is the bonafide work of

S.RAGHAVAN	**71206205038**
E.MANOJ RAJAN	**71206205302**
J.GOKUL MURALI	**71206205014**

who carried out the project work under my supervision.

Mrs.N.Rajathi, M.E.,
Assistant Professor,
Department of Information Technology,
Kumaraguru College of Technology,
Chinnavedampatti Post,
Coimbatore-641006.

Dr.L.S.Jayashree,Ph.D.,
Head of department,
Department of Information Technology,
Kumaraguru College of Technology,
Chinnavedampatti Post,
Coimbatore-641006.

Submitted for viva-voce examination held on 19.4.2010.

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

The exhilaration achieved on successful completion of any task should be shared with the people behind the venture. At the onset, we thank the management of our college for having provided the excellent facilities to carry out the project.

We express our deep gratitude to our Principal Dr.S.Ramachandran Ph.D., for ushering us in the path of triumph.

We are always thankful to our beloved Dean, Department of Computer Science and Engineering, Dr.S.Thangasamy Ph.D, whose consistent support and enthusiastic involvement helped us a great deal.

We convey our sincere thanks to our project coordinator Dr.L.S.Jayashree Ph.D, professor for her invaluable assistance.

We are greatly indebted to our beloved guide Mrs.N.Rajathi,M.E, Assistant Professor, Department of Information Technology for her excellent guidance and timely support during the course of this project.

We also feel elated in manifesting our deep sense of gratitude to all the staff and lab technicians in the Department of Information Technology

ABSTRACT

Steganography is the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message; this is in contrast to cryptography, where the existence of the message itself is not disguised, but the meaning is obscured. The advantage of steganography over cryptography alone is that messages do not attract attention to themselves, to messengers, or to recipients.

We present a very high-capacity and low-distortion 3D steganography scheme. Our steganography approach is based on a novel multilayered embedding scheme to hide secret messages in the vertices of 3D polygon models. Experimental results show that the cover model distortion is very small as the number of hiding layers ranges from 7 to 13 layers. To the best of our knowledge, this novel approach can provide much higher hiding capacity than other state-of-the-art approaches, while obeying the low distortion and security basic requirements for steganography on 3D models.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

The various abbreviations used in this project report are

ASCII	American Standard Code for Information Interchange
BMP	Windows Bitmap
FBI	Federal Bureau of Investigation
GIF	Graphics Interchange Format
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bit
MSB	Most Significant Bit

LIST OF TABLES

INTRODUCTION

1. Introduction

1.1 General:

Steganography is a technique to hide secret messages in a host media called cover media. The advantage of steganography over cryptography is that messages do not attract attention to attackers and even receivers. Steganography and cryptography are often used together to ensure security of the secret messages. For example, many previous steganography approaches use the secret key (i.e., idea borrowed from cryptography) to produce better protection of the information if the stego object arouses suspicion. Therefore, these approaches can become more secure and can be potentially useful to some security-demanding applications such as military intelligence. Watermarking is another popular technique to hide messages and it is usually used for providing ownership on copyrighted multimedia material and for detecting originators of illegally made copies. Therefore, an effective watermarking method must be robust against a variety of attacks. In contrast to watermarking, steganography prefers to hide information as much as possible and requires cover media with distortion as little as possible.

In the past, most of the effort on steganography has been concentrated on various media data types such as an image, an audio file, or even a video file. However, there is relatively little steganography work on 3D models. With the development of 3D hardware, 3D computing or visualization has become much more efficient than ever. This leads to the widespread use of 3D models in various applications such as digital archives, entertainment, Web3D, MPEG4, and game industry. Therefore, 3D models are good candidates and rich resources to serve as the innocuous looking hosts for hiding other types of digital content.

1.2 Objective

In this work, we proposed a very high-capacity and low-distortion 3D steganography scheme. Our steganography approach is based on a novel multilayered embedding scheme to hide secret messages in the vertices of 3D polygon models. Experimental results show that the cover model distortion is very small as the number of hiding layers ranges from 7 to 13 layers. To the best of our knowledge, this novel approach can provide much higher hiding capacity than other state-of-the-art approaches, while obeying the low distortion and security basic requirements for steganography on 3D models.

LITERATURE REVIEW

2. LITERATURE REVIEW

2.1 Existing system:

3D model watermarking technology has been actively studied in the past, but only a few works have addressed information hiding for covert communication. In this work we presented a blind information hiding scheme based on a substitution procedure in the spatial domain. This scheme is derived from the well-known quantization index modulation (QIM). The basic idea is to represent a triangle as a two-state, i.e., state "0" or "1," geometrical object depending on what bit value is to be hidden. They adopted a technique called the triangle strip peeling sequence (TSPS) presented to determine the data embedding order for security considerations. In the existing system, the authors introduced a multilevel embedding procedure extending the scheme proposed for expanding the hiding capacity. They propose three embedding levels called sliding, extending, and rotating to embed data based on slightly shifting the vertex position. This method can provide about three times the capacity when compared to the previous existing hiding schemes. In addition to a multilevel embedding procedure, they proposed a 3D model representation rearrangement procedure to hide more data (9 bits/vertex).

2.2 Proposed system:

In this work, the geometric characteristics of 3D models to provide high-capacity data hiding. Usually, there is a trade-off between capacity, invisibility, and robustness in steganography techniques. Capacity and invisibility are more important than robustness in the steganography system. Therefore, we aim at maximizing data hiding capacity while limiting distortion of cover models in a lower bounded value. To the best of our knowledge, our novel approach can hide much higher bit rates/vertex (i.e., 21 to 39 bits) than other previous state-of-the-art methods in steganography for 3D polygon models. . The vertex distortion of a covert model can be limited to an upper bound I=2, where I is the width of a partitioning interval. The capacity and distortion of our approach depends mainly on the number of partitioning intervals. We optimize the number of partitioning intervals in terms of capacity and distortion.

HARDWARE & SOFTWARE REQUIREMENTS

2.4 HARDWARE & SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS

Processor	:	Pentium IV
Speed	:	Above 500 MHz
RAM capacity	:	2 GB
Floppy disk drive	:	1.44 MB
Hard disk drive	:	200 GB
Key Board	:	Samsung 108 keys
Mouse	:	Logitech Optical Mouse
CD Writer	:	52x LG
Printer	:	DeskJet HP
Motherboard	:	Intel
Cabinet	:	ATX
Monitor	:	17" Samsung

SOFTWARE CONFIGURATION

Operating System	:	Windows XP and above
Front end used	:	MATLAB

2.5 Software Specification:

MATLAB:

MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

KEY FEATURES

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces

- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft Excel

DETAILS OF METHODOLOGY INVOLVED

3. Details of methodology involved

The design phase is a multi step process which focuses on system creation with the help of user specifications and informations gathered in the above phases. It is the phase where the system requirements are translated to operational details.

3.1 INPUT DESIGN

Input design is the part of overall system design which requires very careful attention. Often the collection of input data is the most expensive part of the system, in terms of both the equipment used and the number of people involved; it is the point of most contact for the users with the computer system; and it is prone to error. If data going into the system are incorrect, then the processing and output will magnify these error.

In this system inputs are given in two ways, the existing users can directly enter into the system using login form, and new users have to register all their details in the registration form provided.

Input design is the very important part in the project and should be concentrated well as it is prone to error. The data that are to be inserted are to be inserted with care as this plays a very important role. In order to get the meaningful output and to achieve good accuracy the input should be acceptable and understandable by the user.

3.2 OUTPUT DESIGN

Output design plays a very important role in a system. Getting a correct output is a task that has to be concentrated, as a system is validated as a correct one only if it gives the correct output according to the input.

Here in this project in all the three days of inductions if the employee has completed all his/her input, then the output shows the status as completed or his status will be pending.

3.3 Modules Description:

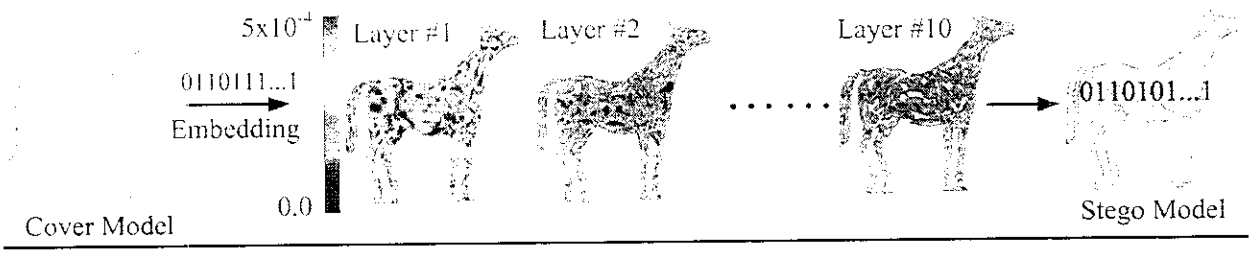


Fig 3.1. An illustration of multilayer information hiding

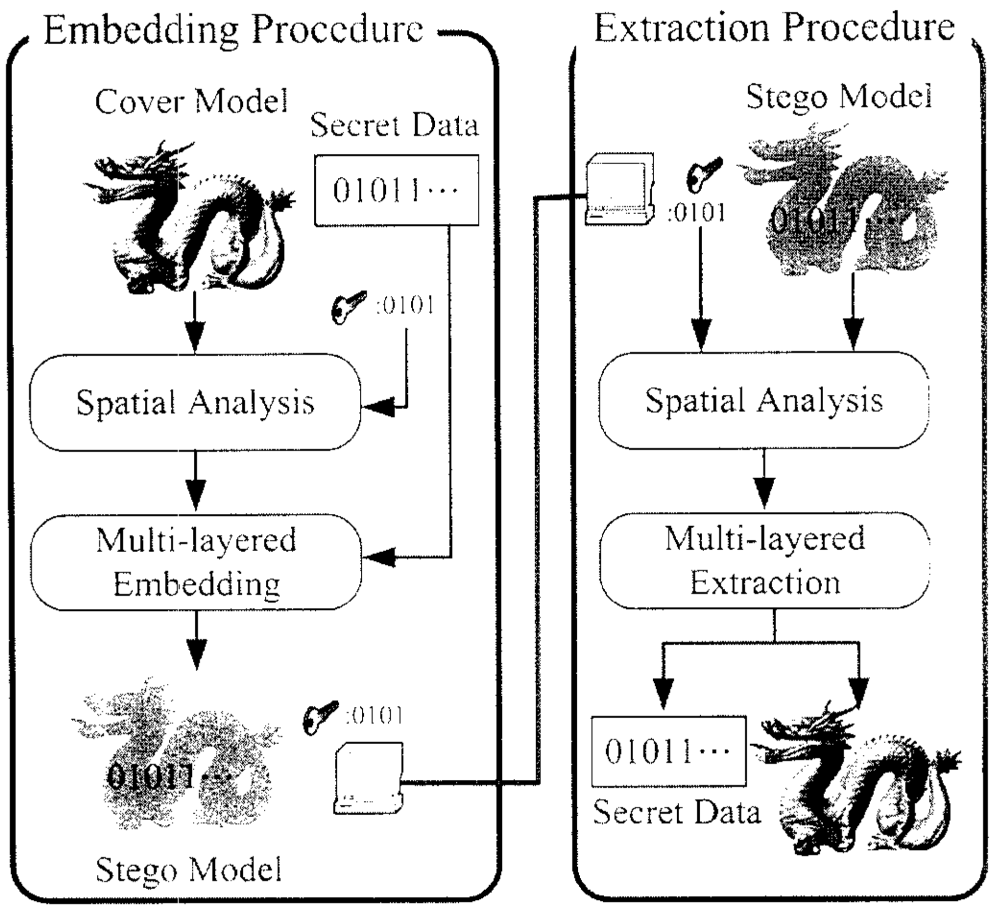


Fig 3.2. The 3D steganography procedure block diagram

3.3.1 Single-Layered Embedding Scheme

First, three vertices of a cover model denoted as Va, Vb, and Vc are selected. They are also called end vertices in this paper. These three end vertices are determined using the PCA of the cover model. Given the first and second principal axes, we orthogonally project all vertices onto these two axes. The vertices that fall on the two extreme ends of the first principal axis are selected as the end vertices Va and Vb. The vertex that fall on the furthest extreme end of the second principal axis is selected as the third end vertex Vc. If an end vertex has multiple candidates, we simply select the nearest candidate (nearest to the principle axis) as the end vertex and slightly shift the other candidates in order to uniquely define an end vertex.

3.3.2 Multilayered Embedding Scheme

It is very easy to extend the single-layered embedding scheme to the multilayered version by directly adding more layers, as follows: Like the single-layered version, two-state region subsets are arranged in an interleaved manner for each layer in our multilayered embedding scheme. However, we make a slight modification: the state regions in the even layers are shifted to the right by I=2. Using this arrangement, the moving direction of a vertex is bidirectional. Vertices are swung forward and backward in the half interval of a state region when embedding payloads on them. In this manner, we can potentially hide more data on each vertex in multilayers and do not enlarge the cover model distortion. The lower bounded distortion is limited to I=2.

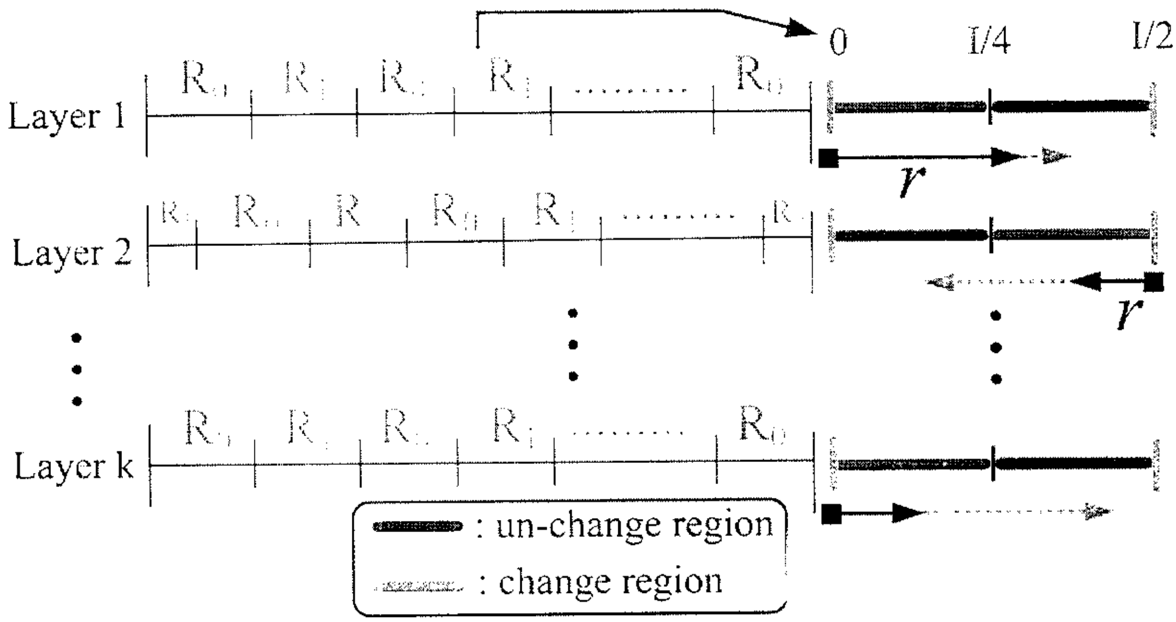


Fig 3.3. An illustration of the multilayer embedding scheme.

3.3.3 Multilayered Extraction Scheme

In the multilayered extraction processes, the payloads are extracted in a reverse embedding order, i.e., from the last layer to the first layer, that is, extracting the bit values in reverse vertex embedding order.

The overall multilayered extraction algorithm is processed as follows:

1. Calculate the variable di for each vertex vi.
2. Calculate the variable ri for each vertex.
3. Extract a bit value from vi.
4. Recover the vertex position on the previous layer.
5. Repeat steps 1-4 for the previous layer extraction until all payloads are completely extracted.

3.3.4 Performance analysis

Thus the performance of the single layered embedding scheme and the multilayered embedding scheme.

3D cover model	Available number of pixels	
	Single layered embedding	Multi layered embedding
Elephant	37672	396900
Dragon	26486	131292
Bird	51026	446880

Table 3.1. Single layer and multi layer embedding comparison

3Dcover models	Vertices	NO. of pixels available	PSNR1	PSNR2
Elephant	42321	1269570	93.56	117.73
Dragon	437645	1312920	92.75	99.71
Bird	34834	940464	100.57	108.68

Table 3.2.Embedding results of various 3D models

CONCLUSION

5. Conclusion

We have introduced a novel multilayered information hiding scheme for 3D polygon models. We optimized the number of partitioning intervals to obtain a good balance between distortion and capacity. The experimental results showed that the proposed method can provide much higher capacity than previous approaches. Currently, the proposed approach has the following limitations that will be solved in the near future. Perfectly smooth (i.e., sphere) or extremely small-size models are not suitable for selection as cover models because the hidden data might be easily observed after even a very small modification by any embedding method. As in the previous work, utilizing PCA to determine the vertex traverse list may potentially hinder the robustness of the proposed method. Although the end vertices and initial triangle obtained from cover and stego models are identical in all cases in our experiments, we cannot guarantee that both are always exactly identical. One simple solution is to simply project vertices on the x; y; and z-axes.

FUTURE SCOPE

6.Future Scope:

However, this approach cannot withstand similarity transformations. A better approach for determining vertex traverse list is required in the near future. Another limitation is that our approach cannot withstand certain malicious attacks such as smoothing, additional noise, nonuniform scaling, simplification, and vertices resampling. As a result, the proposed approach is not suitable for the applications of digital content protection and authentication. In the future,the security related aspects of this 3D steganography algorithm could be further strengthened using various other methods for hiding data such as frequency based methods etc.

CODING

6.Coding:

SINGLELAYER EMBEDDING:

```
function [Image,Max_pos]=single_embedd(Image,signals,text,key)

[row col dim]=size(Image);

siz=length(text);
N=8*siz;

zer=find(signals==0);   %%zero's position



bina_t = dec2bin(text,8);
bint = reshape(bina_t,1,N); %bint is in char
d = 48;
bi = bint - d;      %ascii

h=waitbar(0,'Embeddind the Text in Single layer image');

for wait=1:10000
  waitbar(wait/10000)
end
for in=1:N

   Image(zer(in))=bi(in);
```

```
end
close(h)

Image(row)=siz;

Z=row-100;

for po=1:N
   Image(Z+po)=zer(po);
end
Image(row-1)=key;

Max_pos=length(zer);
```

SINGLE LAYER EXTRACTION:

```
function [rectxt]=single_retrieval(Image,Key1)


[row col dim]=size(Image);


siz=Image(row);
Key=Image(row-1);


if Key==Key1
N=8*siz;
Z_re=row-100;


h=waitbar(0,'Retrieving  the Text from a Single layer image');


for wait=1:10000
  waitbar(wait/10000)
end
close(h)
for po_re=1:N


    zer_re(po_re)=Image(Z_re+po_re);


end


for out=1:N
  re_tx(out)=Image(zer_re(out));
end
```

```
rebi = (dec2bin(re_tx,1))';
rebin = reshape(rebi,siz,8);
rectxt = (bin2dec(rebin))';

else
   rectxt='enter the correct key';
end
```

MULTI LAYER EMBEDDING:

```
function [new_image,max_p]=multilayer_embedd(Image,txt,Key)




h=waitbar(0,'Text Embedding in a Multi layer Image');

for wait=1:10000
   waitbar(wait/10000)
end
close(h)
layer1=Image(:,:,1);
layer2=Image(:,:,2);
layer3=Image(:,:,3);
figure;
  imshow(layer1);title('Layer 1')
figure;
  imshow(layer2);title('Layer 2')
figure;
  imshow(layer3);title('Layer 3')



siz=length(txt);
N=8*siz;

dim=size(layer1);              %% converting to binary
bina_t = dec2bin(txt,8);
```

```
bint = reshape(bina_t,1,N);
d = 48;
bi = bint - d;


emb=round(N/3);


b_1=bi(1:emb);
for en=1:emb
  layer1(en)=b_1(en);
end


b_2=bi(emb+1:(emb*2));
for en=1:emb
  layer2(en)=b_2(en);
end


b_3=bi(((emb*2)+1):N);
for en=1:length(b_3)
  layer3(en)=b_3(en);
end


new_image(:,:,1)=layer1;
new_image(:,:,2)=layer2;
```

```
new_image(:,:,3)=layer3;

max_pix=numel(layer1);
max_pix1=numel(layer2);
max_pix2=numel(layer3);

max_p=max_pix+max_pix1-max_pix2;
figure;

  imshow(new_image)

  [row col dim]=size(new_image);title('Embedded Image')

  new_image(row)=siz;
  new_image(row-1)=Key;
```

MULTILAYER EXTRACTION:

```
function multilayer_retrieving(new_image)



h=waitbar(0,'retrieving Text from a Multi layer Image');

for wait=1:10000
   waitbar(wait/10000)
end

close(h)
[row col th]=size(new_image);

siz=new_image(row);
Key=new_image(row-1);
layer1=new_image(:,:,1);
layer2=new_image(:,:,2);
layer3=new_image(:,:,3);

N=8*siz;

ret=N/3;

for re=1:ret
   r_1(re)=layer1(re);
end
```

```
for re=1:ret
  r_2(re)=layer2(re);
end

BB=N-(ret*2);
for re=1:BB
  r_3(re)=layer3(re);
end
 retxt=[r_1 r_2 r_3];
 rb_txt = (dec2bin(retxt,1))';
rbin = reshape(rb_txt,siz,8);
rectxt = (bin2dec(rbin))';
hitx=char(rectxt);

msgbox(hitx)
```

MAIN:

```
clc
close all;
clear all;

choice=0;
possibility=7;


while choice~=possibility,
   choice=menu('3D STEGNAOGRAPHY','INPUT','SINGLE LAYER
EMBEDDING','SINGLE LAYER EXTRACTING','MULTI LAYER
EMBEDDING','MULTI LAYER EXTRACTING','COMPARISSON','EXIT');

      if choice==1,
            [namefile,pathname]=uigetfile({'*.bmp;*.tif;*.jpg;*.fig;*.gif; *.png; *.pgm;
*.jpeg','IMAGE Files (*.bmp,*.tif,*.jpg,*.gif, *.png, *.pgm *.jpeg)'},'Choose an
Image');
            [image,map]=imread(strcat(pathname,namefile));



%              txt='hi';
%              Key=10;
             txt=input('Enter the text to hide::');
             Key=input('Enter the Key::');

      end
```

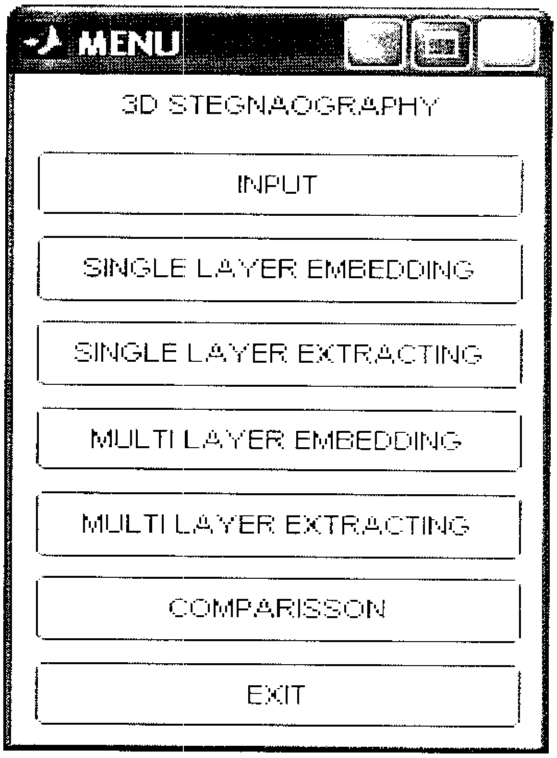
```
if choice==2,

            data1=double(image(:,:,1));
            [M N]=size(data1);
            mn=mean(data1,2);
            B=repmat(mn,1,N);
            data=data1-repmat(mn,1,N);

            covariance = 1 / (N-1) * data * data';
            [PC, V] = eig(covariance);
            V = diag(V);
            c=-1*V;
            [junk, rindices] = sort(c);
             rindices;
            V = V(rindices);
            PC = PC(:,rindices);
            signals = PC' * data;
            signals=round(signals);

      figure;
      imshow(image(:,:,1)),title('Single Layer-Input image')
```

```
    [Image,Max_pos]=single_embedd(data1,signals,txt,Key);
end

if choice==3,

    Key1=input('Enter the key to Retrieve::');
    %Key1=10;
    [rectxt]=single_retrieval(Image,Key1);

    hiddentxt=char(rectxt);
    msgbox(hiddentxt)

end



if choice==4,
        [new_image,max_p]=multilayer_embedd(image,txt,Key);
end


if choice==5

    multilayer_retrieving(new_image)
end
```

```
    if choice==6


        msgbox(num2str(Max_pos))
        msgbox(num2str(max_p))
        figure;
        plot(data1,abs(signals))
%hist(data1)
%           axis([0 300 0 300])
        title('single layer ')
        xlabel('total pixels in a image')
        ylabel('Available  pixels to embedded a text')



        figure;
        new_im=double(new_image(:,:,1)+new_image(:,:,2)+new_image(:,:,3));
         plot(new_im,abs(signals))
%hist(new_im)
        title('multi layer ')
        xlabel('total pixels in a image')
        ylabel('Available  pixels to embedded a text')
    end

    if choice==7
        close all;
    end

end
```

SCREENSHOTS

User interface:



Single layer embedding:

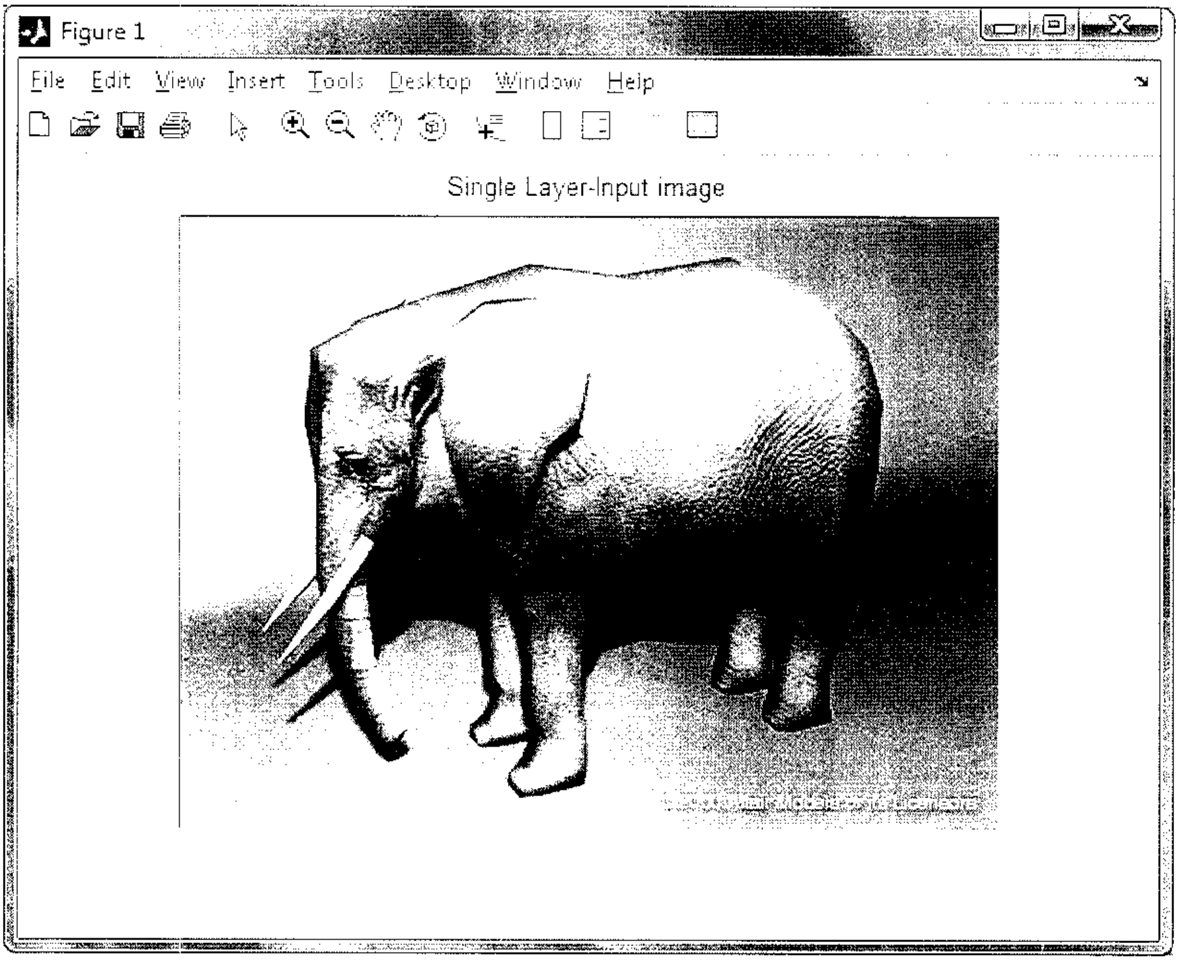


Fig A.1.The singlelayer input image.

User input:

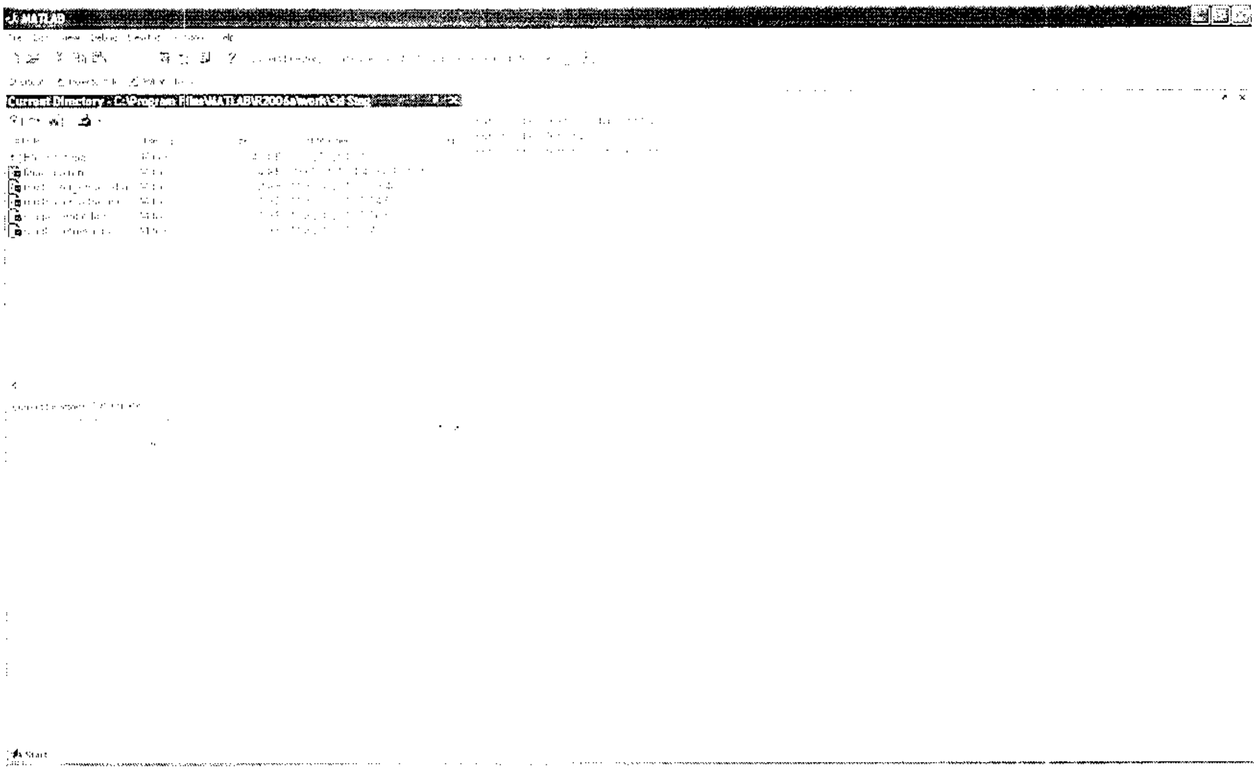
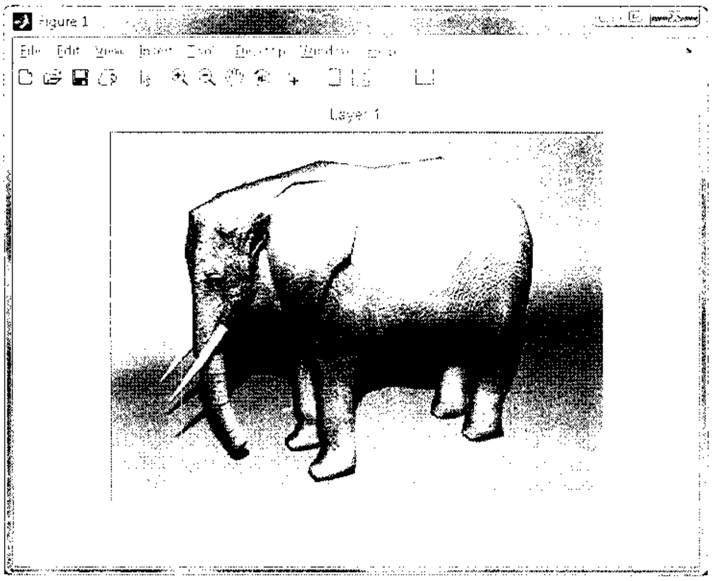
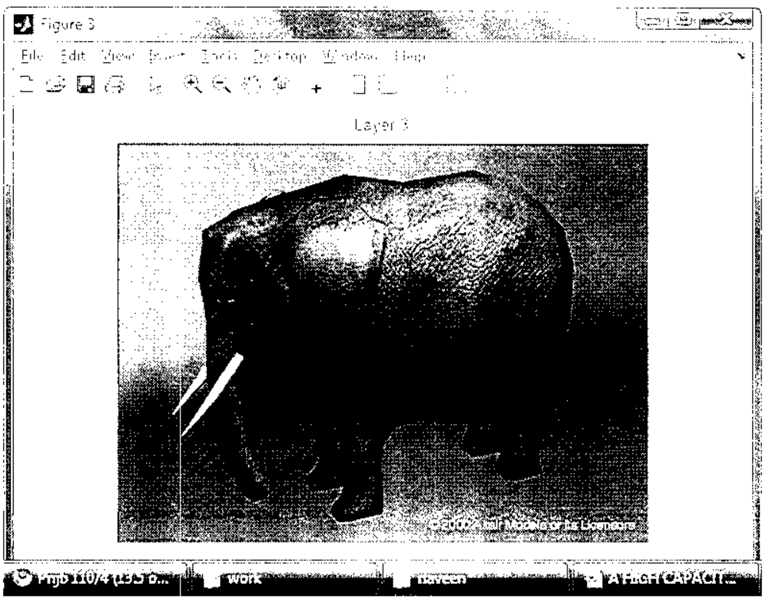
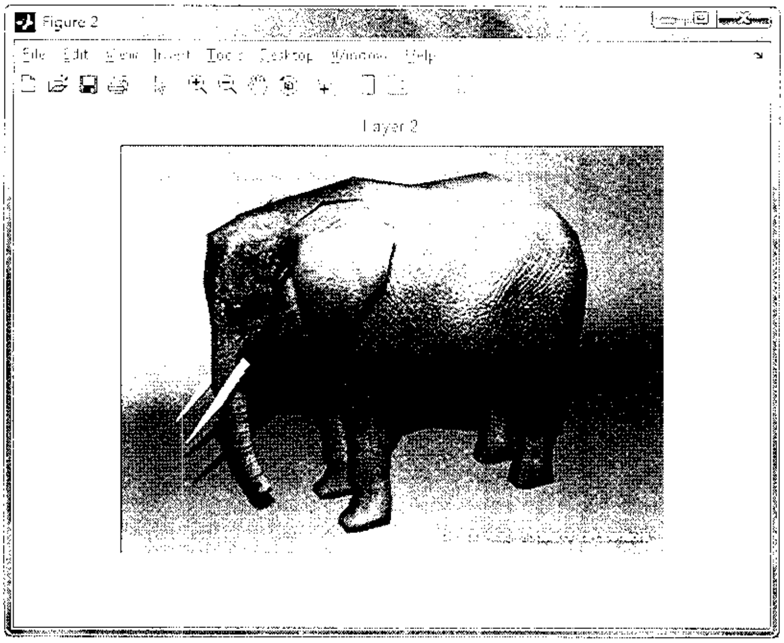


Fig A.2.The user input screen.

Multilayer embedding:







Embedded image:

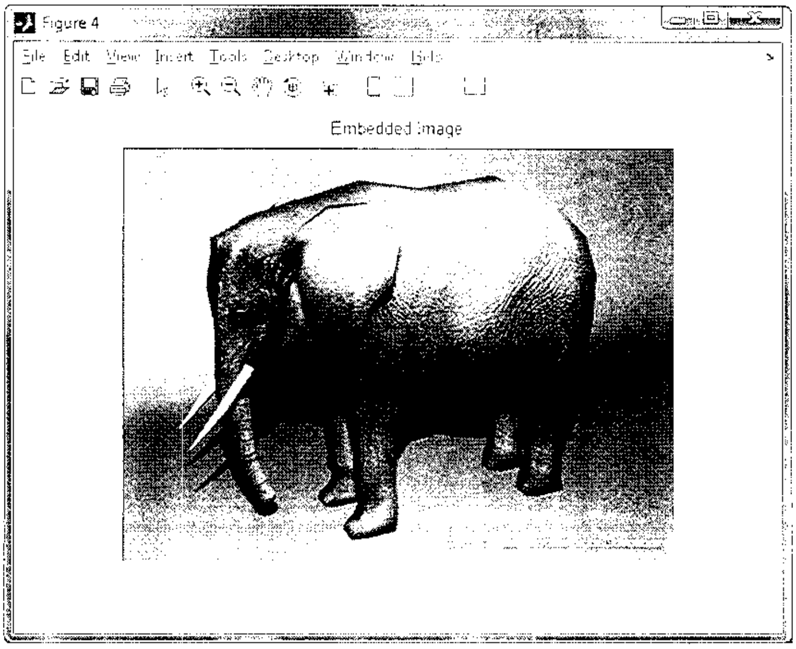


Fig A.3. The multilayer embedded image.

Extracted message:

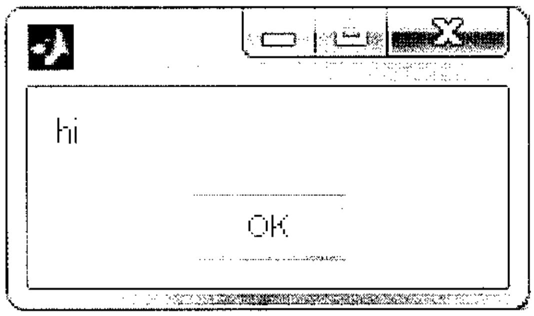


Fig A.4. The message box displaying the extracted message.

Comparison:

Fig A.5.Message boxes displaying the capacities of singlelayer and multilayer embedding.

REFERENCES

7.References:

[1] V.R. Doncel, N. Nikolaidis, and I. Pitas, "An Optimal Detector Structure for the Fourier Descriptors Domain Watermarking of 2D Vector Graphics," IEEE Trans. Visualization and Computer Graphics, vol. 13, no. 5, Sept./Oct. 2009.

[2] I.J. Cox, M.L. Miller, and J.A. Bloom, Digital Watermarking. Morgan Kaufmann, 2002.

[3] S. Katzenbeisser and F.A.P. Petitcolas, eds., Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, 2000.

[4] J. Fridrich, "Applications of Data Hiding in Digital Images," Tutorial for the ISSPA, pp. 22-25, Aug. 1999.

[4] O. Benedens and C. Busch, "Towards Blind Detection of Robust Watermarks in Polygonal Models," Proc. EUROGRAPHICS '00, pp. 199-209, Aug. 2000.

[5] O. Benedens, "Geometry-Based Watermarking of 3-D Polygonal Models," Proc. IEEE Computer Graphics and Applications, special issue on image security, vol. 19, no. 1, pp. 45-46, Jan./Feb. 1999.