P - 1644

IMAGE COMPRESSION USING DCT AND DWT

A PROJECT REPORT

Submitted by

LAKSHMIPRIYA.S **71202205021**

ROSALINE BAGYAM SHEELA.S **71202205037**

SHANMUGA PRIYA.S **71202205044**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

KUMARAGURU COLLEGE OF TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2006

P - 1644

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"IMAGE COMPRESSION USING DCT AND DWT"** is the bonafide work of **"LAKSHMIPRIYA.S, ROSALINE BAGYAM SHEELA.S, SHANMUGA PRIYA.S"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.Gopalsamy	**Ms.N.Suganthi**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	Sr. Lecturer
Dept of Information Technology,	Dept of Information Technology,
Kumaraguru College of Technology,	Kumaraguru College of Technology,
Coimbatore – 641006.	Coimbatore – 641006.

The candidates with University Register Nos 71202205021, 71202205037, 71202205044 were examined by us in the project viva – voce examination held on ..27-4-2006...

INTERNAL EXAMINER	EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our sincere thanks to **Dr.K.Arumugam, B.E. (Hons), M.S. (USA), MIE.,** Correspondent, Kumaraguru College of Technology and the management, for providing us this opportunity to undertake this project work.

We express our profound gratitude to **Dr.K.K.Padmanabhan B.Sc. (Engg)., M.Tech., Ph.D.,** Principal, Kumaraguru College of Technology, Coimbatore, for permitting us to undergo a project work.

We are greatly indebted to **Dr.G.Gopalsamy, Ph.D.,** Professor and Head of the Department of Information Technology, for the immense support he has provided us throughout our project.

We extend our sincere thanks to our project coordinator **Prof.K.R.Baskaran, B.E., M.S.,** Assistant Professor, Department of Information Technology, for his constant support and encouragement.

We would like to express our heart felt thanks to our project guide, **Ms.N.Suganthi, M.E.,** Sr Lecturer Department of Information Technology, for her everlasting guidance and untiring help throughout our project.

We like to express our deep sense of gratitude to our families and friends who helped us in the successful completion of the project.

DECLARATION

We,

LAKSHMIPRIYA.S	**71202205021**
ROSALINE BAGYAM SHEELA.S	**71202205037**
SHANMUGA PRIYA.S	**71202205044**

Declare that the project entitled **"IMAGE COMPRESSION USING DCT AND DWT",** submitted in partial fulfillment to Anna University as the project work of Bachelor Of Technology (Information Technology) Degree, is a record of original work done by us under the supervision and the guidance of **Ms.N.Suganthi, M.E.,** Sr Lecturer, Department of Information Technology, Kumaraguru College of Technology, Coimbatore.

Place: Coimbatore

Date:

[Lakshmipriya.S]

[Rosaline Bagyam Sheela.S]

[Shanmuga Priya.S]

Project Guided by	Head Of the Department
[Ms.N.Suganthi, M.E.,]	**[Dr.G.Gopalsamy, Ph.D.]**

ABSTRACT

The Project work aims at compressing a monochrome image using Discrete Cosine Transform(DCT) and Discrete Wavelet Transform(DWT). The results of compression are compared to find the efficient transforming technique for the images.

In DCT, the input image is divided into blocks and DCT is applied for each block. By using Butterworth filter, filtering is done to reduce noise. The output of the filter is the codebook using which quantization is done. The position of the minimum values are taken and the index matrix is formed. This matrix is compressed. During decompression, the original image is reconstructed by applying inverse DCT.

In DWT, eight level decomposition is applied to the input image using Haar filter. As a result four coefficients namely, approximation, horizontal, vertical and diagonal are obtained as a matrix. The matrix is divided into blocks and DWT is applied for each block to get the codebook. Then quantization is done. The index matrix is formed and it is compressed. During decompression, the original image is reconstructed by applying inverse DWT.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SRS - Software Requirement Specification
DWT - Discrete Wavelet Transform
DCT - Discrete Cosine Transform
PSNR - Peak Signal to Noise Ratio

1. INTRODUCTION

Uncompressed data requires considerable storage capacity and transmission bandwidth. Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, demand for data storage capacity and data-transmission bandwidth continues to outstrip the capabilities of available technologies. The recent growth of data intensive multimedia-based web applications have not only sustained the need for more efficient ways to encode signals and images but have made compression of such signals central to storage and communication technology.

1.1 Existing System

The discrete cosine transform (DCT) helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). It transforms a signal or image from the spatial domain to the frequency domain

Despite all the advantages of compression schemes based on DCT namely simplicity, satisfactory performance, and availability of special purpose hardware for implementation, these are not without their shortcomings. Since the input image needs to be ``blocked," correlation across the block boundaries is not eliminated. This results in noticeable and annoying ``blocking artifacts" particularly at low bit rates. The image quality degrades because of the artifacts.

Disadvantages of DCT

- Only spatial correlation of the pixels inside the single 2-D block is considered and the correlation from the pixels of the neighboring blocks is neglected

- Impossible to completely decorrelate the blocks at their boundaries using DCT
- Undesirable blocking artifacts affect the reconstructed images or video frames. (high compression ratios or very low bit rates)
- Scaling as add-on additional effort
- DCT function is fixed can not be adapted to source data
- Does not perform efficiently for binary images (fax or pictures of fingerprints) characterized by large periods of constant amplitude (low spatial frequencies), followed by brief periods of sharp transitions

1.2 Proposed System

Over the past several years, the wavelet transform has gained widespread acceptance in signal processing in general, and in image compression research in particular. In many applications wavelet-based schemes (also referred as subband coding) outperform other coding schemes like the one based on DCT. Wavelet transform decomposes a signal into a set of basis functions. These basis functions are called *wavelets*

Wavelet coding schemes avoid blocking artifacts. Wavelet-based coding is more robust under transmission and decoding errors, and also facilitates progressive transmission of images. Because of their inherent multiresolution nature, wavelet coding schemes are especially suitable for various applications.

. Wavelet-based coding on the other hand provides substantial improvement in picture quality at low bit rates because of overlapping basis functions and better energy compaction property of wavelet transforms. Because of the inherent multiresolution nature, wavelet-based coders facilitate progressive transmission of images thereby allowing variable bit rates. *scalability* and *tolerable degradation* are important.

Advantages of DWT

- No need to divide the input coding into non-overlapping 2-D blocks, it has higher compression ratios avoid blocking artifacts.
- Allows good localization both in time and spatial frequency domain.
- Transformation of the whole image introduces inherent scaling
- Better identification of which data is relevant to human perception higher compression ratio
- Higher flexibility

2. SYSTEM REQUIREMENT ANALYSIS

2.1 Project Definition

The project "Image Compression Using DCT and DWT" aims in reducing the size of the input image using DCT and DWT concepts. This is used to reduce the bandwidth consumption and increase the speed of the network through which the image is transferred.

2.2 Project Plan

The requirement phase deals with the sequence of activities in producing the Software Requirement Specification (SRS) document. It must be ensured that all the requirements of the software are elicited and analyzed. In other words the needs of the system are identified.

In the problem analysis phase, the current system is analyzed by study of the existing materials and the changes to be made in the proposed system are decided upon. A clear understanding of the needs of the system must be framed that leads to actual specification. MATLAB is selected as the most suitable language for the implementation of the project.

After the analysis phase, the design phase commences in which the various modules and functionalities are identified. The complete system flow of control and data are identified and depicted in the form of diagrams.

Next the implementation phase is taken care of in which the design is translated into code. Each module is coded separately and finally integrated to form the entire system. Care is taken to make the code easily understandable by future users.

In the testing phase each module is tested thoroughly and finally the integrated modules are tested together to ensure the correct working of the entire system. Testing is also done to ensure that the product satisfies the specified requirements and set criteria.

2.3 Software Requierment Specification

2.3.1 Introduction

2.3.1.1 Purpose

The purpose of this document is to specify the requirements of project "Image Compression Using DCT and DWT ". It describes the interfaces for the system.

2.3.1.2 Scope

SRS forms the basis for agreement between the client and the supplier and what the software product will do. It also provides a reference for the validation of the final project.

Any changes made to the SRS in the future will have to go through formal change approval process.

2.3.2 Abbreviation

SRS - Software Requirement Specification
DWT - Discrete Wavelet Transform
DCT - Discrete Cosine Transform
PSNR - Peak Signal to Noise Ratio

2.3.3 Intended Audience and Reading Suggestions

The document is mainly intended for developers, users, documentation writers and project managers. The information can be understood correctly and easily by start reading from the project scope.

2.3.4 Perspective Overall Description

2.3.4.1 Product

This product tells about the concepts of DCT and DWT. This also gives better understanding about the processes of image compression and decompression.

2.3.4.2 Product Features

This project achieves the following goals

- Better clarity
- Scalability
- Minimum data loss
- Reduces bandwidth usage
- Increases transmission speed

2.3.5 Operating Environment

Hardware Specification

Processor	:	Pentium IV 2.4 Ghz
Ram	:	128 MB
Hard Disk Drive	:	20 GB

Software Specification

Operating System	:	Windows 2000
Software	:	MATLAB 6.5

2.3.6 Design and Implementation Constraints

There are not any specific items or issues that will limit the options available to the developers.

2.3.7 Assumptions and Dependencies

The product is developed assuming that bandwidth consumption is more and the speed of the network goes slow. Compression of the image is therefore necessary before sending it through the network.

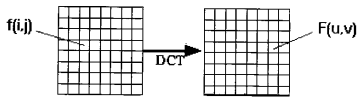
3. SYSTEM STUDY

3.1 DCT

DCT is an important achievement for the research community working on image compression. The DCT can be regarded as a discrete-time version of the Fourier-Cosine series. It is a close relative of DFT, a technique for converting a signal into elementary frequency components. Thus DCT can be computed with a Fast Fourier Transform (FFT) like algorithm in *O(n log n)* operations. Unlike DFT, DCT is real-valued and provides a better approximation of a signal with fewer coefficients.

The discrete cosine transform (DCT) helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality).

3.1.1 Spatial to Frequency Conversion of DCT



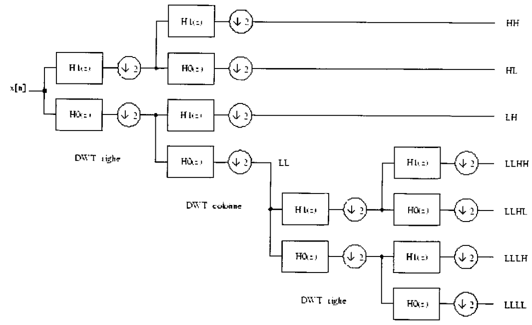
3.2 DWT

Wavelet transform decomposes a signal into a set of basis functions. These basis functions are called *wavelets* Wavelets are obtained from a single prototype wavelet *y(t)* called mother *wavelet* by *dilations* and *shifting*

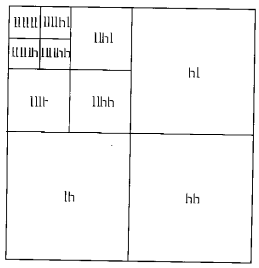
The wavelet transform is computed separately for different segments of the time-domain signal at different frequencies. Multi-resolution analysis:

analyzes the signal at different frequencies giving different resolutions MRA is designed to give good time resolution and poor frequency resolution at high frequencies and good frequency resolution and poor time resolution at low frequencies Good for signal having high frequency components for short durations and low frequency components for long duration.e.g. images and video frames

3.2.1 Decomposition



3.2.2 DECOMPOSITION - Example :



By reading from left to right, *l* or *h* letters tell us, we can see which filter was used for the analysis; with this convention the sub-images with two letters correspond to the first level of decomposition, the ones with 4 letters to the second level and so on.

3.3 Process

In DCT, the input image is resized into 256 x 256 and the image is divided into uniform blocks of size 8 x 8. DCT is applied to each of the blocks to get the image coefficients. Butterworth filter is then applied to the obtained coefficients. The result of which is then restructured to the size of 150. This is the old codebook. The restructuring size is taken as 150 because if the size is reduced further data loss occurs and the inclusion of values greater than this does not make any difference in the output. Butterworth filter is applied as it gives flat response. The new codebook is generated by repeating the same block process. Comparing the two codebooks the position of the minimum values are found and are put in the index matrix. Quantization process is then applied which gives the compressed image. The original image is reconstructed

by applying inverse DCT to the compressed image. PSNR value is calculated which depicts the clarity of the reconstructed image.

In DWT, the input image is resized into 256 x 256. Eight level decomposition is performed using the haar filter to obtain the four coefficients namely, approximation, horizontal, vertical, diagonal. A 256 x 256 matrix is formed as a result of this decomposition. The resultant matrix is divided into uniform blocks of size 8 x 8. DWT is applied to each of the blocks to get the image coefficients. The result of which is restructured to the size of 100. This is the old codebook. The new codebook is generated by repeating the same block process. Comparing the two codebooks the position of the minimum values are found and are put in the index matrix. Quantization process is then applied which gives the compressed image. The original image is reconstructed by applying inverse DWT to the compressed image. PSNR value is calculated which depicts the clarity of the reconstructed image.

By using the PSNR value and the image size before and after compression, DCT and DWT methods of compression are compared to know which of these two methods is better.

3.4 MATLAB – A CURTAIN RAISER

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran. The name MATLAB stands for MATrix LABoratory.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation and many others.

MATLAB is a high-level language and interactive environment that enables to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and FORTRAN. MATLAB can be used 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 our work. MATLAB code can be integrated with other languages and applications and be distributed.

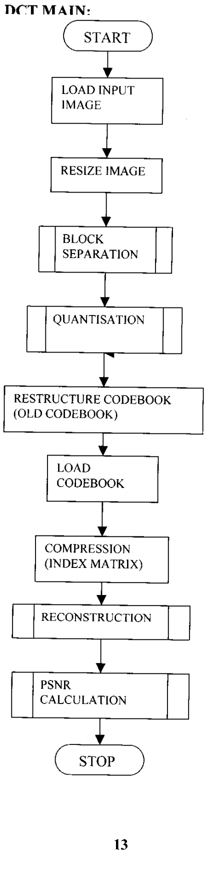
Key Features

- 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

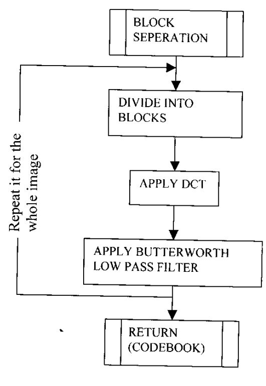
4.SYSTEM DESIGN

4.1 PROCESS DESIGN

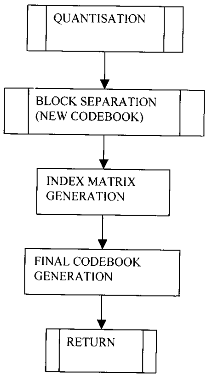
DCT MAIN:



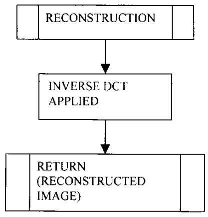
BLOCK SEPARATION:



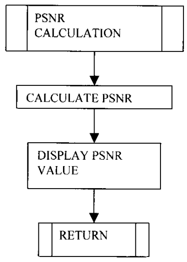
QUANTISATION:



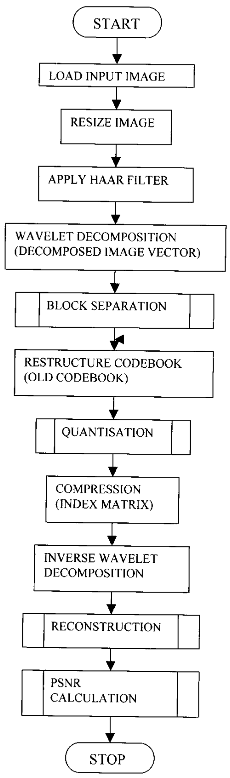
RECONSTRUCTION:



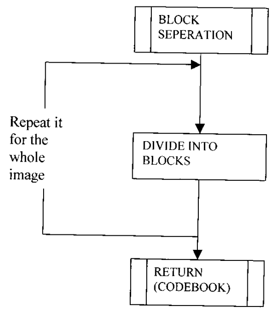
PSNR CALCULATION:



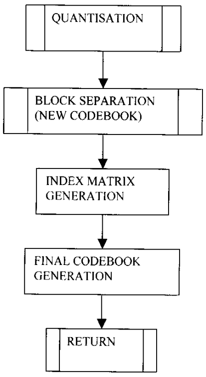
DWT MAIN:



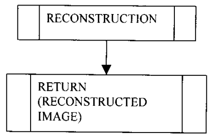
BLOCK SEPARATION:



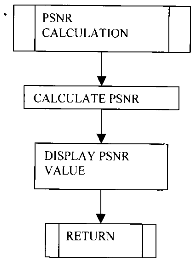
QUANTISATION:



RECONSTRUCTION:



PSNR CALCULATION:



5. PRODUCT TESTING

Testing can be separated into Verification & Validation and Functional & Structural testing.

5.1. UNIT TESTING

The testing of a single program, module or code. Each module of our project were executed separately and checked for the correct working of the module. The block separation module divides the given input image into blocks of same size. In the quantization module the appropriate filter is used to get the image components. The reconstruction module reconstructs the original image from the compressed one without any change in the clarity of the image. The modules were tested separately and the intended output for the modules were obtained.

5.2. VERIFICATION

Verification methods are used to ensure the system(software, hardware) compiles with an organization standard and processes relying on review. In our system the modules were tested for correct functionality.

5.3. VALIDATION TESTING

Validation ensures that the system operates according to the plans by executing real-life function. Validation is done for the modules.

5.3.1. Block Separation Module Validation

Block separation module takes the resized image as input and divides the image into blocks of same size. If the image does not exist then the corresponding error message is displayed.

5.3.2. Quantization Module Validation

The output of the block separation module is taken as input and the appropriate filter is applied to get the image components which are then quantized. If the input blocks are not of appropriate size then the corresponding error message is displayed. If the suitable filter is not applied then the corresponding error message is displayed.

5.3.3. Reconstruction Module Validation

The compressed image is taken as input and the original image is reconstructed. The reconstruction process should be just the inverse of the compression process, if not error is displayed.

5.4. INTEGRATION TESTING

The testing of related modules, program, validates that multiple parts of the system interact according to the plan. The various modules in our project are integrated, such that block separation, quantization and reconstruction are done correctly without any change in the clarity of the image.

5.5. FUNCTIONAL AND STRUCTURAL TESTING

5.5.1. Functional Testing

Functional testing is called Black Box Testing because of no knowledge. They verify all validation tests and inspect how the system performs. Each module is tested for expected output, without tracing the path of the code. Block separation module takes the image as input and divides it into blocks. Quantization module takes the blocks as input and gives the quantized image as output. The reconstruction module takes the compressed image as input and gives the original image as output.

5.5.2. Structural Testing

Structural testing is called White Box Testing because the internal logic of the system is known. In our project, the block separation, quantization and reconstruction of image according to the correct format is tested by tracing the code structurally .The validation of the input file, image size validation, block size validation, components validation and reconstructed image validation are all done in structural testing.

5.6. BLACK BOX TESTING

The purpose of black box testing is to check out the inputs and the expected output for the system. The input is a bmp or jpg file that is to be compressed. The output is the reconstruction of the original image from the compressed image.

5.7. WHITE BOX TESTING

The purpose of white box testing is to check out whether the logical operation of the system works well. Here the processing of the input is considered. A threshold frequency is set and the values which are lower than this limit are made to zero and the values higher than the threshold values are considered for compression. The decompression operation involves the inverse of the compression operation. This is done for the proper reconstruction of the original image.

6. FUTURE ENHANCEMENT

In the proposed system, only monochrome image is applied for compression. This project can be extended for color images and video.

7. CONCLUSION

The compression method using DCT is used in many applications but still it has many disadvantages: clarity of the reconstructed image is poor as the PSNR value is 7.13dB, high compression of the input image leads to data loss, the input image is not highly compressed. The file size of the original image is 192KB and that of compressed image is 65KB. All these disadvantages can be overcome by using DWT : the clarity of the reconstructed image is better than using DCT as the PSNR value is found to be high of 25.16dB , data loss due to high compression of the input image is not as high as in the case of using DCT for compression, the input image is highly compressed. The file size of the original image is 257KB and that of compressed image is 65KB.

8. APPENDIX

8.1 SOURCE CODE

DCT

Main:

```
clear;
close all;
load lena;
X=lena;
wd=256;
Z=imresize(X,[wd wd]);
imag1=Z;
wd1=size(imag1,1);
wd2=size(imag1,2);
book1=block(imag1,16,16);
% Code Book Formation
a=book1(1:150);
b=a;
% Initial Code Book
for i=1:2
b=LBGforimage(a,imag1);
a=b;
end
save codebook.mat b;
load codebook;
% Compression of Image
imax=length(book1);
N=length(b);
for i=1:imax
   datablock=book1(i).data;
   dist=[ ];
   for j=1:N;
   v=b(j).data;
   S=sum(sum((datablock-v).^2));
   dist=[dist S];
   end
[y,ind]=min(dist);
Q(i)=ind;
end
%Decompresssion of Image
```

```
imax=length(Q);
final=zeros(wd1,wd2);
for i=1:imax
final=reconstruct(final,i,b,Q(i));
end
figure,imshow(final);
imwrite(final,'decom.bmp');
psnrval(im2double(uint8(final)),im2double(X));
```

Block:

```
function r=block(y,xw,yw)
% Design Of Butterworth Low Pass Filter
[n,Wn] = buttord(.1,.99999,1,2);
[b,a] = butter(n,Wn);
xmax=size(y,1);
ymax=size(y,2);
imax=xmax/xw;
jmax=ymax/yw;
k=0;
for i=1:imax
  for j=1:jmax
    k=k+1;
    x(k).data=y(((i-1)*xw+1):i*xw,((j-1)*yw+1):j*yw);
    % Applying DCT
    d(k).data=dct(x(k).data);
    % Low Pass Filtering
    r(k).data=filter(b,a,d(k).data);
  end
end
return
```


LBGforimage:

```
function newcodebook = LBGforimage(oldcodebook,u)
C=oldcodebook;
N=length(C);
k=size(C(1).data);
%N is the size of the codebook
%k is the dimension of the quantizer
M=block(u,k(1),k(2));
m=length(M);
%m is number of blocks in training sequence
for i=1:m;
databIock=M(i).data;
dist=[ ];
  for j=1:N;
  v=C(j).data;
  S=sum(sum((datablock-v).^2));
  dist=[dist S];
  end
[y,ind]=min(dist);
Q(i)=ind;
end

for j1=1:N;
r=(Q==j1);
  if length(r)>0
    imax=length(r);
    sum1=0;
    for i=1:imax
      if r(i)==1
      sum1=sum1+M(i).data;
      end
    end
    C(j1).data=sum1/sum(r);
  else
  end
end
newcodebook=C;
```


Reconstruct:

```
function imager=reconstruct(imagei,index,codebook,cindex)
imager=imagei;
wd1=size(imagei,1);
wd2=size(imagei,2);
b1=size(codebook(1).data,1);
b2=size(codebook(1).data,2);
p1=wd1/b1;
t1=mod(index,p1);
if t1==0 t1=p1; end
if mod(index,p1)~=0
  t2=fix(index/p1)+1;
else
  t2=fix(index/p1);
end
imager(((t2-1)*b1+1):t2*b1,((t1-
1)*b1+1):t1*b1)=idct(codebook(cindex).data);
return
```

psnrval:

```
function psnrval(A,B)
%input image
error = A - B;
decibels = 20*log10(1/(sqrt(mean(mean(error.^2)))));
disp(sprintf('PSNR = +%5.2f dB',decibels))
```


DWT

Main:

```
clear all;
close all;
clc;
warning off;
%image compression
lena=imread('lena.bmp');
sd=size(lena);
if(size(sd,2)==3)
  lena=rgb2gray(lena);
end
wd=256;
lena=imresize(lena,[wd wd]);
im=double(lena);
[nRow nColumn]=size(lena);
%level of decomposition
level=8;
type = 'haar';
[Lo_D,Hi_D,Lo_R,Hi_R]=wfilters(type);
[im_d, Sm] = f_DWT(im,level,Lo_D,Hi_D);
fprintf(' \n  wavelet decomposition is completed');
Z=im_d;
imag1=Z;
wd1=size(imag1,1);
wd2=size(imag1,2);
book1=block(imag1,8,8);
% Code Book Formation
a=book1(1:100);
% Initial Code Book
for i=1:10
b=LBGforimage(a,imag1);
a=b;
end
save codebook.mat b;

% Compression of Image
imax=length(book1);
N=length(b);
for i=1:imax
  datablock=book1(i).data;
  dist=[ ];
  for j=1:N;
```

```
v=b(j).data;
S=sum(sum((datablock-v).^2));
dist=[dist S];
end
[y,ind]=min(dist);
Q(i)=ind;
End

%Decompresssion of Image
%load woman;
imax=length(Q);
final=zeros(wd1,wd2);
for i=1:imax
final=reconstruct(final,i,b,Q(i));
end

%----------- Wavelet Reconstruction ----------------
imr = f_InvDWT(final, Sm, Lo_R, Hi_R,level);
fprintf('\n   wavelet reconstruction is completed \n');
imwrite(uint8(imr),'decom.bmp');
psnrval(im2double(uint8(imr)),im2double(lena));
figure,imshow(lena);
title('input image');

figure,imshow(uint8(im_d));
title(' wavelet decomposed image');
figure,imshow(uint8(final));
title(' vetor quantization decompressed image');
figure,imshow(uint8(imr));
title('output image');
```


f_DWT:

```
function [I_W , S] = f_DWT(I, level, Lo_D, Hi_D);

% Wavelet decomposition
%
% input:   I : input image
%          level : wavelet decomposition level
%          Lo_D : low-pass decomposition filter
%          Hi_D : high-pass decomposition filter
%
% output:  I_W : decomposed image vector
%          S : corresponding bookkeeping matrix
%

[C,S] = wavedec2(I,level,Lo_D,Hi_D);
S(:,3) = S(:,1).*S(:,2);       % dim of detail coef nmatrices
L = length(S);
I_W = zeros(S(L,1),S(L,2));

% approx part
I_W( 1:S(1,1) , 1:S(1,2) ) = reshape(C(1:S(1,3)),S(1,1:2));

for k = 2 : L-1
  rows = [sum(S(1:k-1,1))+1:sum(S(1:k,1))];
  columns = [sum(S(1:k-1,2))+1:sum(S(1:k,2))];
  % horizontal part
  c_start = S(1,3) + 3*sum(S(2:k-1,3)) + 1;
  c_stop = S(1,3) + 3*sum(S(2:k-1,3)) + S(k,3);
  I_W( 1:S(k,1) , columns ) = reshape( C(c_start:c_stop) , S(k,1:2) );

  % vertical part
  c_start = S(1,3) + 3*sum(S(2:k-1,3)) + S(k,3) + 1;
  c_stop = S(1,3) + 3*sum(S(2:k-1,3)) + 2*S(k,3);
  I_W( rows , 1:S(k,2) ) = reshape( C(c_start:c_stop) , S(k,1:2) );

  % diagonal part
  c_start = S(1,3) + 3*sum(S(2:k-1,3)) + 2*S(k,3) + 1;
  c_stop = S(1,3) + 3*sum(S(2:k,3));
  I_W( rows , columns ) = reshape( C(c_start:c_stop) , S(k,1:2) );
end
```


Block:

```
function x=block(y,xw,yw)
xmax=size(y,1);
ymax=size(y,2);
imax=xmax/xw;
jmax=ymax/yw;
k=0;
for i=1:imax
  for j=1:jmax
    k=k+1;
    x(k).data=y(((i-1)*xw+1):i*xw,((j-1)*yw+1):j*yw);
  end
end
return
```

LBGforimage:

```
function newcodebook = LBGforimage(oldcodebook,u)
C=oldcodebook;
N=length(C);
k=size(C(1).data);
%N is the size of the codebook
%k is the dimension of the quantizer
M=block(u,k(1),k(2));
m=length(M);
%m is number of blocks in training sequence
for i=1:m;
datablock=M(i).data;
dist=[ ];
  for j=1:N;
  v=C(j).data;
  S=sum(sum((datablock-v).^2));
  dist=[dist S];
  end
[y,ind]=min(dist);
Q(i)=ind;
end

for j1=1:N;
r=(Q==j1);
```



```
  if length(r)>0
    imax=length(r);
    sum1=0;
    for i=1:imax
      if r(i)==1
      sum1=sum1+M(i).data;
      end
    end
    C(j1).data=sum1/sum(r);
  else
  end
end
newcodebook=C;
```

Reconstruct:

```
function imager=reconstruct(imagei,index,codebook,cindex)
imager=imagei;
wd1=size(imagei,1);
wd2=size(imagei,2);
b1=size(codebook(1).data,1);
b2=size(codebook(1).data,2);
p1=wd1/b1;
t1=mod(index,p1);
if t1==0 t1=p1; end
if mod(index,p1)~=0
  t2=fix(index/p1)+1;
else
  t2=fix(index/p1);
end
imager(((t2-1)*b1+1):t2*b1,((t1-1)*b1+1):t1*b1)=codebook(cindex).data;
return
```

f_InvDWT:

```
function im_rec = f_InvDWT(I_W, S, Lo_R, Hi_R, level);
% Inverse wavelet decomposition
%
% input:   I_W : decomposed image vector
%          S : corresponding bookkeeping matrix
%          Lo_D : low-pass decomposition filter
%          Hi_D : high-pass decomposition filter
%          level : wavelet decomposition level
%
% output:   im_rec : reconstruted image
%
L = length(S);
m = I_W;
C1 = zeros(1,S(1,3)+3*sum(S(2:L-1,3)));

% approx part
C1(1:S(1,3)) = reshape( m( 1:S(1,1) , 1:S(1,2) ), 1 , S(1,3) );

for k = 2:L-1
   rows = [sum(S(1:k-1,1))+1:sum(S(1:k,1))];
   columns = [sum(S(1:k-1,2))+1:sum(S(1:k,2))];
   % horizontal part
   c_start = S(1,3) + 3*sum(S(2:k-1,3)) + 1;
   c_stop = S(1,3) + 3*sum(S(2:k-1,3)) + S(k,3);
   C1(c_start:c_stop) = reshape( m( 1:S(k,1) , columns ) , 1, c_stop-
c_start+1);
   % vertical part
   c_start = S(1,3) + 3*sum(S(2:k-1,3)) + S(k,3) + 1;
   c_stop = S(1,3) + 3*sum(S(2:k-1,3)) + 2*S(k,3);
   C1(c_start:c_stop) = reshape( m( rows , 1:S(k,2) ) , 1 , c_stop-c_start+1
);
   % diagonal part
   c_start = S(1,3) + 3*sum(S(2:k-1,3)) + 2*S(k,3) + 1;
   c_stop = S(1,3) + 3*sum(S(2:k,3));
   C1(c_start:c_stop) = reshape( m( rows , columns ) , 1 , c_stop-c_start+1);
end

if (( L - 2) > level)   %set those coef. in higher scale to 0
   temp = zeros(1, length(C1) - (S(1,3)+3*sum(S(2:(level+1),3))));
   C1(S((level+2),3)+1 : length(C1)) = temp;
end
```



```
S(:,3) = [];
im_rec = waverec2(C1,S, Lo_R, Hi_R);
```

psnrval:

```
function psnrval(A,B)
%input image
error = A - B;
decibels = 20*log10(1/(sqrt(mean(mean(error.^2)))));
disp(sprintf('PSNR = +%5.2f dB',decibels))
```

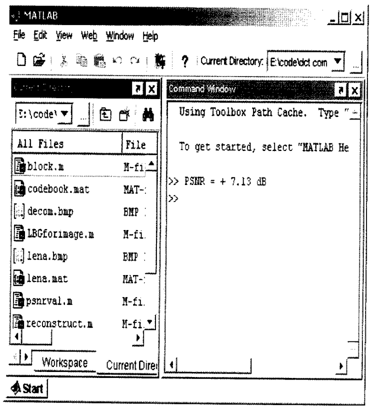

8.2 SAMPLE OUTPUT

DCT

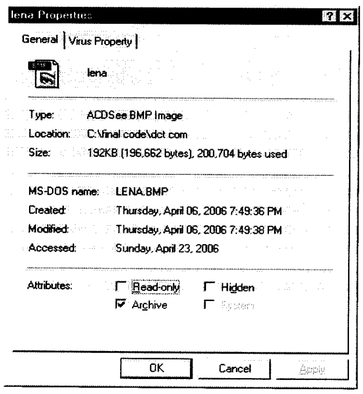
ORIGINAL IMAGE

OUTPUT IMAGE

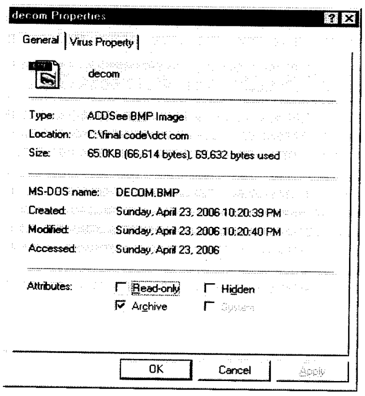
SCREEN DISPLAYING PSNR VALUE



FILE SIZE OF ORIGINAL IMAGE



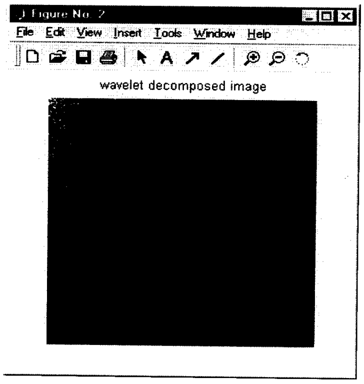
FILE SIZE OF COMPRESSSED IMAGE



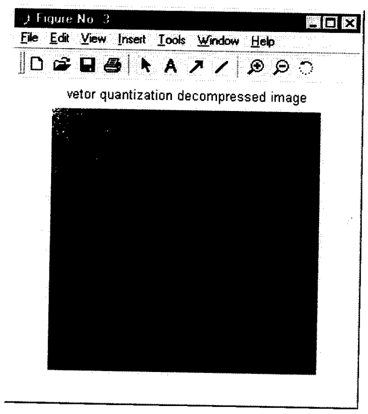
DWT

ORIGINAL IMAGE

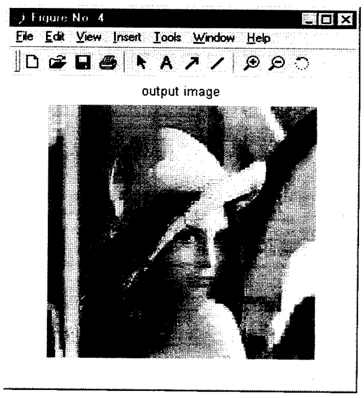
WAVELET DECOMPOSED IMAGE



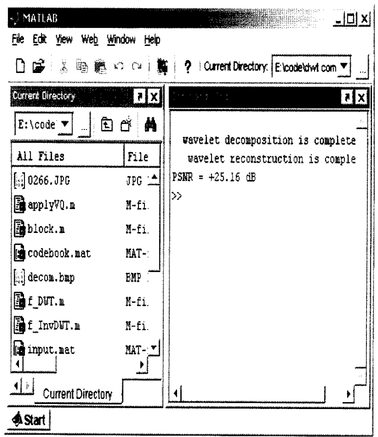
VECTOR QUANTIZATION DECOMPRESSED IMAGE



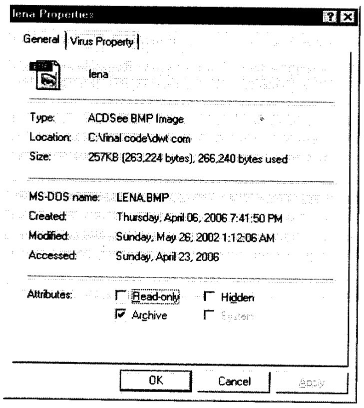
OUTPUT IMAGE



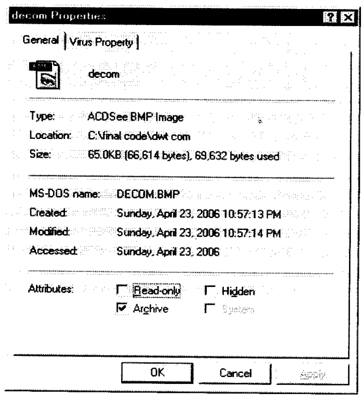
SCREEN DISPLAYING PSNR VALUE



FILE SIZE OF ORIGINAL IMAGE



FILE SIZE OF COMPRESSSED IMAGE



9. REFERENCES

1. Gonzolas, "Image Processing with MATLAB"
2. Huang Xian-Wu, Wang Jia-Jua, Li Jia-Hua, "Digital Image Processing and Compression Coding Technology University ChenDu Dec 2000
3. Kristian Sandberg, "Introduction to Image Processing in MATLAB1"
4. LideY.etal, "An Algorithm for Vector Quantization Design"