IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 9, Issue 2, Ver. III (Mar - Apr. 2014), PP 07-13
www.iosrjournals.org

Image Compression Using 1-D,2-D Dct And 3-D Discrete Cosine Transform

R.R.A.S.Narasimha Reddy, T.Madhu

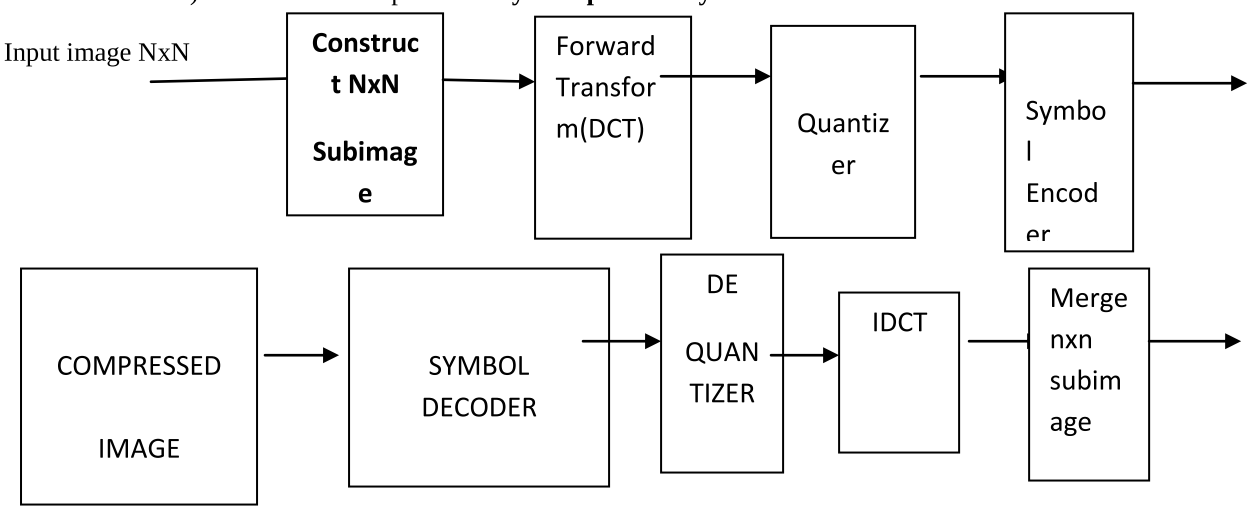
Abstract: *Image compression is a widely addressed researched area. Mamny compression standards are in place. But still there is a scope for high compression with quality reconstruction. In this paper we are going to perform image compression for better quality reconstruction by using Various types of Discrete Cosine Transform such as 1-D,2-D and 3-D transforms. In this paper we are going to compare the results for each type of compression technique. An extensive experimentation has been been carried out to arrive at the conclusion.*

I. Introduction

Image compression is the art /scienceof efficientlycoding digital images to reduce the number of bits required in representing an image. The purpose of doing sois to reduce the storage and transmission costs while maintaining good quality. Compressing an imageis significantly different than compressing raw binary data. Of course, general purpose compression programs can be used to compress image, but the resultis less than optimal.this is because images have certain stastical propertieswhich can be exploited by encoders specifically designed for them. Image compression is achieved by exploiting redundancies in the image,which could be spatial, spectral, or temporal redundancy. Spatial redundancy is due to the correlation between neighboring pixels. Typically, a compressed image when decoded to reconstruct its original form will be accompanied by some distortion. The efficiency of a compression algorithm is measured by its data compressing ability, the resulting distortion and as well by its implementation complexity
The complexity of data compression algorithm is a particular importantant consideration in their hardware implementation. Depending upon the reconstructed image , to be exactly same as the original or some unidentified loss may be incurred following two techniques for compression exists
Lossy compression system whose techniques can be used in images where some of the finer details in the image can be sacrified for the sake of saving a little or more bandwidth or storage space.

Lossless Compression system which aim at minimizing the bitrate of the compressed output without any distortion of the image. The decompressed bit stream is identical to original bit stream.

In this paper we are using following image compression model which make use of DCT(Discrete Cosine Transform) which is an example of **Lossy Compression system**



The above model comprises of the following 4 operations

· Sub image decomposition
. Transformation
. Quantization
. Encoding

An NxN image first is sub divided into sub images of size nxn which are then transformed to generate $(N/n)^2$ nxn subimage transform arrays. The goal of transformation is to decorrelate the pixels of each sub image or to pack as much
information as possible into the smallest no of transform coefficients.The quantization process eliminates or more coarsely quantizes the coefficients that carry least information.These coefficients have the smallest impact on reconstructed sub image quality the encoding process terminates by coding the quantized coefficient.Any or all the transform encoding steps are adapted to local image content called adaptive transform coding or fixed for all sub images called non adaptive transform coding.

II. Concept Of Discrete Cosine Transform

We will now describe the image transform that has attractedthe most attention in recent years. The DCT is the most widely used transform in a class of image coding systems known as transform coders. To achieve digital representation of an image, the image must be converted to a set of binary integers, which in turn can be operated upon to recover the picture with a minimum possible degradation. The fact that , even though a fine sampling and quantization of a image are essential for desirable subjective quality of a digital picture from the view point of a stastician the information in the picture can be conveyed quite adequately with out all these variables. On the other hand, one and the adverse effect that would have on the the subjective quality of the picture., which leads to the next step that involves exploring the possibility of transforming these samples to a new set of variates that well have an varying degree of contributing to both the informations content subjective of the picture. Then one can discard the less significant of these variables without affecting the stastical information content of the picture causing a severe degradation in the subjective quality of the resultant picture. , which is basically what is achieved by using the DCT., which affords the energy compaction property that has been elaborated as before. As a result of employing this transform there is unequal distribution of the stastical information of the picture. Most of the information becomes concentrated towards the low-freequency region and hence coefficients away from the origin may be completely discarded or quantized using a fewer number of bits. This process is called **COMPRESSION**

III. Mathematical Representation Of Discrete Cosine Transform

1-D DCT

Let x(n) denote an N-point sequence that is zero outside the range $0 \le n \le N-1$ for N-point sequence, the DCT is also N-points long and is given by the relation,

$$C_x(k) = \sum_{n=0}^{N-1} 2x(n) \cos\left\{\frac{\pi k(2n+1)}{2N}\right\} \quad \text{for } 0 \le k \le N-1$$

$$= 0 \quad \text{otherwise}$$

If x(n) is real, $C_x(k)$ is also real and if x(n) is complex $C_x(k)$ is also complex
In order to compute back the N-point x(n) sequence we make use of the IDCT
(Inverse DCT) relationship is given by

$$X(n) = 1/N \sum_{k=0}^{N-1} W(k) Cx(k) \; \cos\left\{\frac{\pi k(2n+1)}{2N}\right\} \text{ for } 0 \le n \le N-1$$

$$= 0 \text{ otherwise}$$

2-D DCT

The two dimensional DCT relationship is used to evaluate the DCT of a 2-D sequence namely an image or a picture. Let x(n,n2) denote a 2-D sequence of N1xN2 points that is zero outside the range for $0 \le n1 \le N-1$, for $0 \le n2 \le N-1$

For this particular sequence, the DCT is also N1xN2 points that is zero outside the range for $0 \le n1 \le N-1$, for $0 \le n2 \le N-1$ is given by

$$C(u,v) = \alpha(u)\alpha(v) \sum_{N1=0}^{N-1} \sum_{N2=0}^{N-1} x(n1,n2) \cos\{2n1+1)\, u\pi/N\} \; \cos\{2n1+1)\, v\pi/N\}$$

& the corresponding IDCT relationship is given by\

$$X(n1,n2) = \sum_{N1=0}^{N-1} \alpha(u)\alpha(v) \sum_{N2=0}^{N-1} x(n1,n2) \cos\{2n1+1)\, u\pi/N\} \; \cos\{2n1+1)\, v\pi/N\}$$

Where

$\alpha(u) = 1/\sqrt{n}$ for u=0

$\sqrt{2/N}$ for u=1,2,…….,N-1

α(u)=1/$\sqrt{n}$ for v=0
$\sqrt{2/N}$ for v=1,2,…….,N-1

3-D DCT

The three dimensional DCT relationship is used to evaluate the DCT of a 3-D sequence namely an image or a picture. Let x(n1,n2,n3) denote a 3-D sequence of N1xN2 xN3points that is zero outside the range for 0 ≤ n1≤ N-1, for 0 ≤ n2≤ N-1 for 0 ≤ n3≤ N-1

For this particular sequence, the DCT is also N1xN2 points that is zero outside the range for 0 ≤ n1≤ N-1, for 0 ≤ n2≤ N-1 , 0 ≤ n3≤ N-1 is given by

C(u,v) = α(u)α(v)$\sum_{N1=0}^{N-1} \sum_{N2=0}^{N-1} x(n1,n2,n3)\cos\{2n1+1)u\pi/N\}\ \cos\{2n1+1)v\pi/N\}\ \cos\{2n1+1)w\pi/N\}$

& the corresponding IDCT relationship is given by\

X(n1,n2,n3)=
$\sum_{N1=0}^{N-1} \alpha(u)\alpha(v)\alpha(w) \sum_{N2=0}^{N-1} x(n1,n2)\cos\{2n1+1)u\pi/N\}\ \cos\{2n1+1)v\pi/N\}\ \cos\{2n1+1)w\pi/N\}$

Where
α(u)=1/$\sqrt{n}$ for u=0
$\sqrt{2/N}$ for u=1,2,…….,N-1

α(v)=1/$\sqrt{n}$ for v=0
$\sqrt{2/N}$ for v=1,2,…….,N-1
α(w)=1/$\sqrt{n}$ for w=0
$\sqrt{2/N}$ for w=1,2,…….,N-1

IV. softwaresimulation & results

In this section we are giving MATLAB Code for image compression using 1-D,2-D 3-D DCT

1-D DCT

```
        f=imread('cameraman.tif');
       f1=im2double(f);
 N=input('enter theno of sample points of the given discrete time sequence');
 for i=1:N
   x1(i)=f1(i,j);
end
figure; imshow(x1);
for k1=1:N
        temp=0;
           for i=1:N
    temp=temp+x1(i).*(sqrt(2./N).*cos(2*pi*((2*k1)+1))./(i*N));
            end
         y(k1)=temp;
     end
    %b=numberof bits used to represent each value
b=5;
min=-1; max=1;
range=max-min;
delta=range./(2^b-1);
%quantize decimal values
y_quant=quant(y,delta);
y_int=y_quant./delta;
y_int_off=y_int+2^(b-1);
int_off=y_int_off;
binaryvect_intermed=[0];
for i=1:length(int_off)
```

```
    convert_compleated=i./length(int_off)*100;
    for j=b:-1:1
        if(int_off(i)>=2^(j-1));
            int_off(i)=int_off(i)-2^(j-1);
            binaryvect_intermed=[binaryvect_intermed 1];
        else
            binaryvect_intermed=[binaryvect_intermed -1];
        end
 end
end
binaryvect=zeros(length(binaryvect_intermed)-1);
binaryvect1=binaryvect_intermed(2:length(binaryvect_intermed));
%figure;imshow(x2);
figure;imshow(y);
figure;imshow(binaryvect);
d3=uencode(binaryvect,2);
figure;imshow(d3);
d4=udecode(d3(1:256),2);
%b=numberof bits used to represent each value
b=5;
min=-1; max=1;
range=max-min;
delta=range./(2^b-1);
d4_bin=(d4+1)./2;
a6=d4_bin(1:256);
intvect_intermed=[0];
for i=1:length((d4)./b)
    intval=0;
    for j=1:b
        intval=intval+a6((i)*b+(j-1))*(2.^(b-j));
    end
    intvect_intermed=[intvect_intermed intval];
end
intvect=zeros(length(intvect_intermed) -1);
intvect=intvect_intermed(2:length(intvect_intermed));
intvect_off=intvect-(2^(b-1)-1);
decimalvect=intvect_off*delta;
d5=idct(decimalvect);
figure;imshow(d4);
figure;imshow(decimalvect);
figure;imshow(d5);
```

2-D DCT

```
f=imread('cameraman.tif');
f1=im2double(f);
N=input('enter theno of sample points of the given discrete time sequence');
for i=1:N
    for j=1:N
    x1(i,j)=f1(i,j);
end
figure; imshow(x1);
for k1=1:N
    for k2=1:N
       temp=0;
            for i=1:N
                for j=1:N

temp=temp+x1(i,j).*(sqrt(2./N).*cos(2*pi*((2*k1)+1))./(i*N)).*(sqrt(2./N).*
cos(2*pi*((2*k2)+1))./(j*N));
```

```
                 end
                y(k1,k2)=temp;
            end
    end
end
%b=numberof bits used to represent each value
b=5;
min=-1; max=1;
range=max-min;
delta=range./(2^b-1);
%quantize decimal values
y_quant=quant(y,delta);
y_int=y_quant./delta;
y_int_off=y_int+2^(b-1);
int_off=y_int_off;
binaryvect_intermed=[0];
for i=1:length(int_off)
    convert_compleated=i./length(int_off)*100;
    for j=b:-1:1
        if(int_off(i)>=2^(j-1));
            int_off(i)=int_off(i)-2^(j-1);
            binaryvect_intermed=[binaryvect_intermed 1];
        else
            binaryvect_intermed=[binaryvect_intermed -1];
        end
 end
end
binaryvect=zeros(length(binaryvect_intermed)-1);
binaryvect1=binaryvect_intermed(2:length(binaryvect_intermed));
%figure;imshow(x2);
figure;imshow(y);
figure;imshow(binaryvect);
d3=uencode(binaryvect,2);
figure;imshow(d3);
d4=udecode(d3(1:256),2);
%b=numberof bits used to represent each value
b=5;
min=-1; max=1;
range=max-min;
delta=range./(2^b-1);
d4_bin=(d4+1)./2;
a6=d4_bin(1:256);
intvect_intermed=[0];
for i=1:length((d4)./b)
    intval=0;
    for j=1:b
        intval=intval+a6((i)*b+(j-1))*(2.^(b-j));
    end
    intvect_intermed=[intvect_intermed intval];
end
intvect=zeros(length(intvect_intermed) -1);
intvect=intvect_intermed(2:length(intvect_intermed));
intvect_off=intvect-(2^(b-1)-1);
decimalvect=intvect_off*delta;
d5=idct(decimalvect);
figure;imshow(d4);
figure;imshow(decimalvect);
figure;imshow(d5);
```

3-D DCT

```
 1.%program for 3-D image data compression using 3-d DCT
f=imread('cameraman.tif');
f1=im2double(f);
N=input('enter  theno  of  sample  points  of  the  given  discrete  time
sequence');
for i=1:N
    for j=1:N
    x(i,j)=f1(i,j);
    end
end
for k=1:N
    x1(k)=f1(k);
end
x2=[x; x1];
figure; imshow(x);
figure; imshow(x2);
for k1=1:N
    for k2=1:N
         temp=0;
            for i=1:N
                for j=1:N

temp=temp+x2(i,j).*(sqrt(2./N).*cos(2*pi*((2*k1)+1))./(i*N).*sqrt(2./N).*co
s(2*pi*((2*k2)+1))./(j*N));
                    end
                end
            y(k1,k2)=temp;
        end
    end
for k1=1:N
    temp=0;
    for i=1:N
        temp=temp+x2(i).*(sqrt(2./N).*cos(2*pi*((2*k1)+1)./(j*N)));
    end
    y1(k1)=temp;
end
y2=[y;y1];
%b=numberof bits used to represent each value
b=5;
min=-1; max=1;
range=max-min;
delta=range./(2^b-1);
%quantize decimal values
Y2_quant=quant(y2,delta);
y_int=y_quant./delta;
y2_int_off=y2_int+2^(b-1);
int_off=y2_int_off;
binaryvect_intermed=[0];
for i=1:length(int_off)
    convert_compleated=i./length(int_off)*100;
    for j=b:-1:1
        if(int_off(i)>=2^(j-1));
            int_off(i)=int_off(i)-2^(j-1);
            binaryvect_intermed=[binaryvect_intermed 1];
        else
            binaryvect_intermed=[binaryvect_intermed -1];
        end
 end
end
binaryvect=zeros(length(binaryvect_intermed)-1);
```

```
binaryvect1=binaryvect_intermed(2:length(binaryvect_intermed));
%figure;imshow(x2);
figure;imshow(y2);
figure;imshow(binaryvect);
d3=uencode(binaryvect,2);
figure;imshow(d3);
d4=udecode(d3(1:256),2);
%b=numberof bits used to represent each value
b=5;
min=-1; max=1;
range=max-min;
delta=range./(2^b-1);
d4_bin=(d4+1)./2;
a6=d4_bin(1:256);
intvect_intermed=[0];
for i=1:length((d4)./b)
    intval=0;
    for j=1:b
        intval=intval+a6((i)*b+(j-1))*(2.^(b-j));
    end
    intvect_intermed=[intvect_intermed intval];
end
intvect=zeros(length(intvect_intermed) -1);
intvect=intvect_intermed(2:length(intvect_intermed));
intvect_off=intvect-(2^(b-1)-1);
decimalvect=intvect_off*delta;
d5=idct(decimalvect);
figure;imshow(d4);
figure;imshow(decimalvect);
figure;imshow(d5);
```

V. Conclusions

In this paper we suggest a method how to compress an digital image using various types of Discrete Cosine transform.