Khwopa Engineering College

Course Code :BEG 475 IP

Image Processing and Pattern recogntiion

Lab Report on Edge Detection and Image Compression

Author:
Rabi Raj Khadka

Roll Number:
700324

July 11, 2017

Contents

1 Theory

Edge Detection
Edge detection is a well-developed field on its own within image processing. Region boundaries and edges are closely related, since there is often a sharp adjustment in intensity at the region boundaries. Edge detection techniques have therefore been used as the base of another segmentation technique.

The edges identified by edge detection are often disconnected. To segment an object from an image however, one needs closed region boundaries. The desired edges are the boundaries between such objects or spatial-taxons.

Spatial-taxons are information granules, consisting of a crisp pixel region, stationed at abstraction levels within a hierarchical nested scene architecture. They are similar to the Gestalt psychological designation of figure-ground, but are extended to include foreground, object groups, objects and salient object parts. Edge detection methods can be applied to the spatial-taxon region, in the same manner they would be applied to a silhouette. This method is particularly useful when the disconnected edge is part of an illusory contour.

`Different Filters Used for Edge Detection are:`
- ▷ Gradient Filter
- ▷ Robert Filter
- ▷ Sobel Filter
- ▷ Prewitt Filter
- ▷ Canny Filter
- ▷ Laplace Filter

2 Code Description

Program for Edge Detection and Compression

```
%Gradient Filters
a=imread('img\neuromancer.jpg');
b=rgb2gray(a);
h=[1 0 -1;2 0 -2;1 0 -1]/9;
c=imfilter(b,h);
figure;
subplot(121),imshow(b),title('Original Image');
subplot(122),imshow(c),title('Gradient Filtered Image');
rob=[1 0;-1 0];
c=conv2(b,rob,'valid');
yrob=[1 -1;0 0];
ab=conv2(b,yrob,'valid');
figure;
subplot(121),imshow(c),title('Robert filter in x axis');
subplot(122),imshow(ab),title('Robert filter in y axis');
a=edge(b,'roberts');
ab=edge(b,'sobel');
abc=edge(b,'prewitt');
abcd=edge(b,'canny');
figure;
subplot(221),imshow(a),title('Robert filter');
subplot(222),imshow(ab),title('Sobel filter');
subplot(223),imshow(abc),title('Prewitt filter');
subplot(224),imshow(abcd),title('Canny filter');
%Laplace Filter
h=[0 1 0;1 -4 1;0 1 0];
x=imread('img\neuromancer.jpg');
x=rgb2gray(x);
x=im2double(x);
y=conv2(x,h,'valid');
h=fspecial('laplacian');
z=conv2(x,h,'valid');
figure;
subplot(121),imshow(y),title('Laplace filtering on image ');
subplot(122),imshow(z),title('Built in laplace filter ');
%DCT and IDCT for compression
xorg=imread('img\neuromancer.jpg');
```

```
x=rgb2gray(xorg);
F=dct2(x);
ff=idct2(F);
[r,c]=size(x);
DF=zeros(r,c);
DFF=DF;
IDF=DF;
IDFF=DF;
depth=4; N=8;
for i=1:N:r
    for j=1:N:c
        f=x(i:i+N-1,j:j+N-1);
        df=dct2(f);
        DF(i:i+N-1,j:j+N-1)=df;
        dff=idct2(df);
        DFF(i:i+N-1,j:j+N-1)=dff;
        df(N:-1:depth+1,:)=0;
        df(:,N:-1:depth+1)=0;
        IDF(i:i+N-1,j:j+N-1)=df;
        dff=idct(df);
        IDFF(i:i+N-1,j:j+N-1)=dff;
    end
end
A=DFF/255;imwrite(A,'DCTcompressed.jpg');
B=IDFF/255;imwrite(B,'IDCTcompressed.jpg');
figure;
subplot(241),imshow(x),title('Gray Scale Image');
subplot(242),imshow(F*0.01),title('Built-in DCT ');
subplot(246),imshow(ff/255),title('Built-in IDCT');
subplot(243),imshow(DF),title('DCT Block Image');
subplot(247),imshow(DFF/255),title('IDCT Block Image');
subplot(244),imshow(A),title('DCT Compressed Image');
subplot(248),imshow(B),title('IDCT Compressed Image');
subplot(245),imshow(xorg),title('Original Image');
```

3 Result and Discussion

Input image is converted to gray scale image and gradient filter is applied to it and then robert filter on x axis and y axis alone are applied. Different Built-in fillters available on MATLAB is applied to grayscale image and shown as below.
Another image is taken for applying the Discrete Cosine Transformation and IDCT and image is compressed and saved as another image.
Outputs

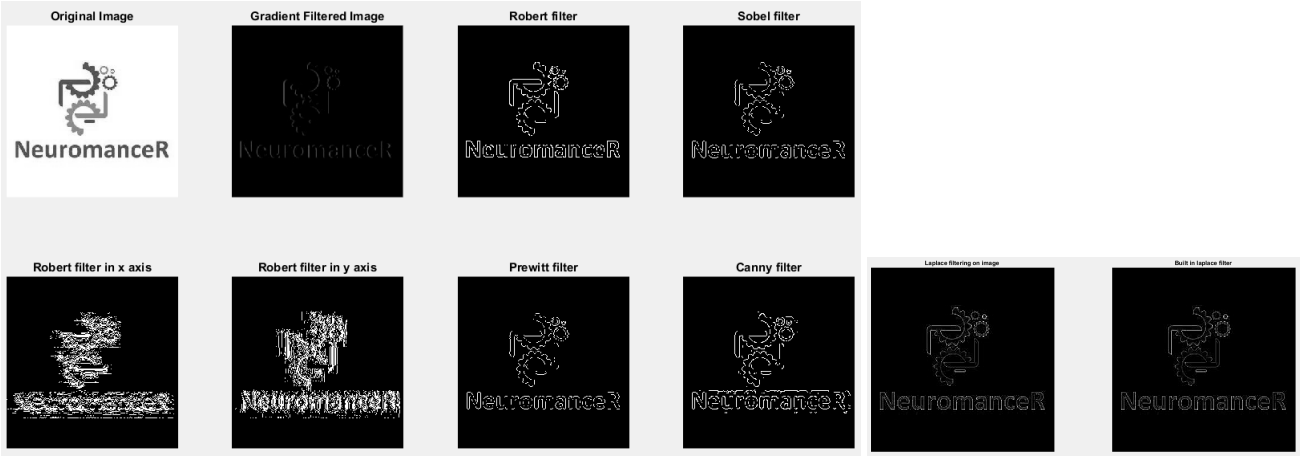

Figure 1: Edge Detection Filters

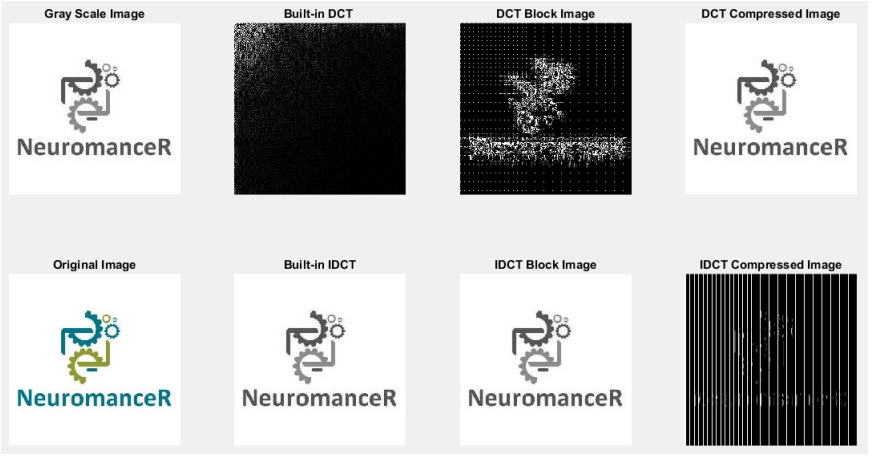

Figure 2: DCT and IDCT for Image Compression

4 Conclusion

Hence,
We are familiarized with edge detection of an image using different filters such as Sobel, Canny, Robert, Gradient,Laplace and Discrete Cosine Transformation and Inverse Discrete Cosine Transformation for image compression using the MATLAB application.