**Project 3**

UIN:01183457

**Task. 1. First, implement HT following the discussion in text on page 490 – 493 for N= 8. Reproduce Figure 6.18 from text.**

**Output:**

Chart, line chart

Description automatically generatedTable

Description automatically generated with low confidence

Calendar

Description automatically generated

Code:

Main Code:

k= generate\_haar(8);

Ndecimals = 2;

f = 10.^Ndecimals;

A = round(f\*k)/f

A=transpose(so); B=transpose(s1);C=transpose(s2);D=transpose(s3); E=transpose(s4); F=transpose(s5); G=transpose(s6);

H=transpose(s7);

first\_1= A\*so; second\_1=B\*so; third\_1=C\*so; four\_1=D\*so; fiv\_1=E\*so; six\_1=F\*so; sev\_1=G\*so; eig\_1=H\*so;

first\_2=A\*s1; second\_2=B\*s1; third\_2=C\*s1; four\_2=D\*s1; fiv\_2=E\*s1; six\_2=F\*s1; sev\_2=G\*s1; eig\_2=H\*s1;

first\_3=A\*s2; second\_3=B\*s2; third\_3=C\*s2; four\_3=D\*s2; fiv\_3=E\*s2; six\_3=F\*s2; sev\_3=G\*s2; eig\_3=H\*s2;

first\_4=A\*s3; second\_4=B\*s3; third\_4=C\*s3; four\_4=D\*s3; fiv\_4=E\*s3; six\_4=F\*s3; sev\_4=G\*s3; eig\_4=H\*s3;

first\_5=A\*s4; second\_5=B\*s4; third\_5=C\*s4; four\_5=D\*s4; fiv\_5=E\*s4; six\_5=F\*s4; sev\_5=G\*s4; eig\_5=H\*s4;

first\_6=A\*s5; second\_6=B\*s5; third\_6=C\*s5; four\_6=D\*s5; fiv\_6=E\*s5; six\_6=F\*s5; sev\_6=G\*s5; eig\_6=H\*s5;

first\_7=A\*s6; second\_7=B\*s6; third\_7=C\*s6; four\_7=D\*s6; fiv\_7=E\*s6; six\_7=F\*s6; sev\_7=G\*s6; eig\_7=H\*s6;

first\_8=A\*s7; second\_8=B\*s7; third\_8=C\*s7; four\_8=D\*s7; fiv\_8=E\*s7; six\_8=F\*s7; sev\_8=G\*s7; eig\_8=H\*s7;

**Associated Function:**

function [Hr]=generate\_haar(N)

if (N<2 || (log2(N)-floor(log2(N)))~=0)

error('The input argument should be of form 2^k');

end

p=[0 0];

q=[0 1];

n=nextpow2(N);

for i=1:n-1

p=[p i\*ones(1,2^i)];

t=1:(2^i);

q=[q t];

end

Hr=zeros(N,N);

Hr(1,:)=1;

for i=2:N;

P=p(1,i); Q=q(1,i);

for j= (N\*(Q-1)/(2^P)):(N\*((Q-0.5)/(2^P))-1)

Hr(i,j+1)=2^(P/2);

end

for j= (N\*((Q-0.5)/(2^P))):(N\*(Q/(2^P))-1)

Hr(i,j+1)=-(2^(P/2));

end

end

Hr=Hr\*(1/sqrt(N));

end

**Task.2 Next, apply this HT transform on the *Tree* image to obtain forward Haar transformed image and display the transformed image.**

**Output:**

A picture containing calendar

Description automatically generated

A picture containing text, electronics, display, screenshot

Description automatically generatedA picture containing text, electronics, screenshot

Description automatically generated

Code:

I=imread('4.1.06.tiff'); %Read in image

in\_img = rgb2gray(I);

x=double(in\_img);

vert=haar\_transform(x);

figure;

imagesc(vert);colormap(gray);axis image; title('Level 1 Coeffcient with Haar Wavelet');

vert\_m=haar\_transform(vert);

figure;

imagesc(vert\_m);colormap(gray);axis image;title('Level 2 Coeffcient with Haar Wavelet');

vert\_3=haar\_transform(vert\_m);

figure;

imagesc(vert\_3);colormap(gray);axis image;title('Level 3 Coeffcient with Haar Wavelet');

**Associated Function:**

function vert =haar\_transform(x)

k=x(:,1:2:end);

l=x(:,2:2:end);

% haar\_matrix = 1/sqrt(2)\*[1 1; 1 -1] implementation

horizavg=(k+l)/sqrt(2); %along the row (avg of horizontal component) using [1 1] (Low frequencey)

horizdif=(k-l)/sqrt(2); %along the row( diff between horizontal component) using [1 -1](High Frequency)

horiz=[horizavg horizdif];

vertavg=(horiz(1:2:end,:)+horiz(2:2:end,:))/sqrt(2); % columwise using [1 1]

vertdif=(horiz(1:2:end,:)-horiz(2:2:end,:))/sqrt(2); % columwise using [1 -1]

vert=[vertavg; vertdif];

end

**Task.3. Obtain inverse transformed image and display the image.**

**Output:**

Graphical user interface, application, PowerPoint

Description automatically generated

**Code:**

% %Synthesis operation (if avg and diff is known, we can get the original

% image

inv\_1=inverse\_haar(vert\_3); %inverse transform from level 3 to level 2

inv\_2=inverse\_haar(inv\_1); %inverse transform from level 2 to level 1

inv\_3=inverse\_haar(inv\_2); %inverse transform from level 1 to original image

figure;

subplot (1,2,1);

imagesc(in\_img);colormap(gray);axis image;title('Original Image');

subplot (1,2,2);

imagesc(inv\_3);colormap(gray);axis image;title('Restored Image');

**Associated Function:**

function d=inverse\_haar(vert)

c=vert;

c(1:2:end,:)=( vert(1:end/2,:)+ vert(end/2+1:end,:) )/sqrt(2);

c(2:2:end,:)=( vert(1:end/2,:)- vert(end/2+1:end,:) )/sqrt(2);

% figure;

% imagesc(c);colormap(gray);axis image;

d=c;

d(:,1:2:end)= ( c(:,1:end/2) + c(:,end/2+1:end) )/sqrt(2);

d(:,2:2:end)= ( c(:,1:end/2) - c(:,end/2+1:end) )/sqrt(2);

end

**Task.4. Compare your original and transformed images and clearly discuss your findings.**

**Output:**

Root mean square error between the pixel values of restored image and original image is zero.



It is observed that with the increment in the co-efficient in haar transformed image the percentage of error decreases and by doing histogram analysis of transformed image, we see that in haar transformed image most of the co-efficient are concentrated around zero and hence with the increment of level we see the decomposition of low-frequency part of image.

A picture containing chart

Description automatically generated

Chart

Description automatically generated

**Code:**

figure;

hist(vert\_3(:),200);

cs=sort( abs ( vert\_3(:) ),'descend');

L=length( vert\_3(:) );

percentage=0.1:0.1:5;

MSE=[];

en=sum(cs.^2);

for p=percentage

K=ceil(p\*L/100);

error=sum( cs(K+1:end).^2);

MSE=[MSE error/en];

end

figure;

plot(percentage,MSE\*100);

xlabel('Percentage of Coefficient Kept')

ylabel('Percentage of error')

% Find the error in pixel values between Original and Restored Image

DIF=imsubtract(x,inv\_3);

mse=mean(mean(DIF.\*DIF));

rmse=sqrt(mse);

formatSpec = 'RMSE is %f\n';

fprintf(formatSpec,rmse);

**Task 5: Discuss how HT follows the properties of a basic wavelet transform.**

Fourier transform is to decompose the original function with a series of different frequencies of the cosine function. After the transformation is the coefficient of the original function at different frequencies of the positive cosine.

Wavelet transform uses a series of different scales to decompose the original function, and the transformation obtained is the coefficient of the original function under different scale wavelet.

Different wavelet transform decomposition by translation and scale, translation is to obtain the time characteristics of the original function, and the scale transformation is to obtain the frequency characteristics of the original function.



HAAR wavelet used here, the zoom or scale function is [1 1], the wavelet function is [1 -1]. It is the simplest wavelet.

Wavelets are basically basis function or kernel which is parametrized by translation and scaling parameter. Haar wavelets are basically square functions which can scaled and translated to obtain different time-frequency resolution. For example, from the haar transform matrix in Task:1, the first row is simply the DC value or we would say father wavelet because it has some average value. Whereas the second row we can see the average value is zero and it is called mother wavelet because all consequent wavelets are created from mother wavelet by using translation and scaling. For instance, the difference between fourth to seventh row of haar matrix is the translation of values.



Table

Description automatically generated with low confidence



Wavelets with high frequency and low scale provide high resolution in time and low resolution in frequency whereas wavelets with low frequency and high scale give good resolution in frequency and less resolution in time. Basically splitting in time means doubling in frequency.