ECE 5256

Project 8: Wavelet transform

Keefe Kamp

April 5, 2023

1 Part 1

For the first three parts of this project the shape of different kinds of wavelets are looked at. For the first part a signal was decomposed using the `wavedec()` command. I then replaced the C array with an array with a zero array with a single 1 at the index of 300 to investigate the level 1 wavelet. This was done for 3 different wavelets.

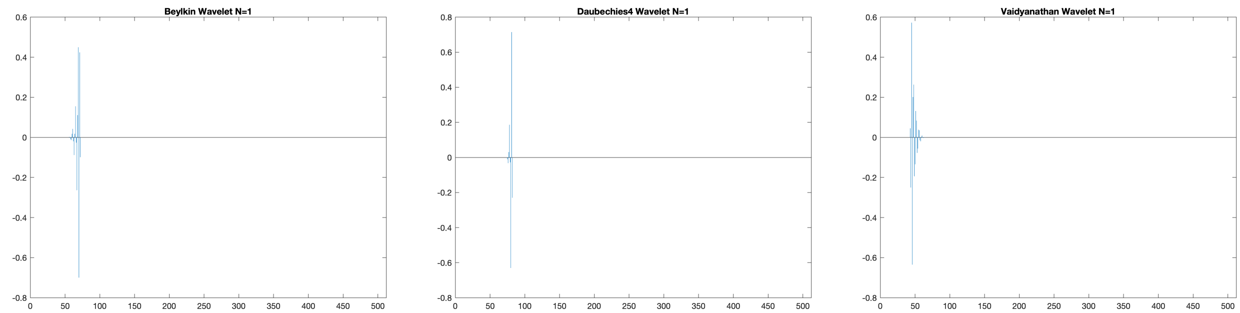

Figure 1: The three level 1 wavelets

2 Part 2

For the second part the level 2 wavelets were looked at using a similar procedure to the process outlines in section 1 but the 1 was placed at an index of 245 for the level 2 wavelet.

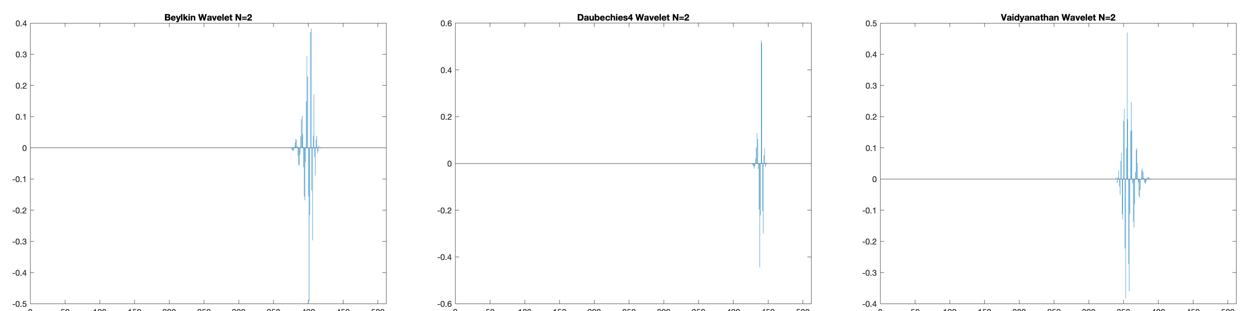

Figure 2: The three level 2 wavelets

3 Part 3

For the third part the level 3 wavelets were looked at using a similar procedure to the process outlines in section 1 but the 1 was placed at an index of 120 for the level 3 wavelet.

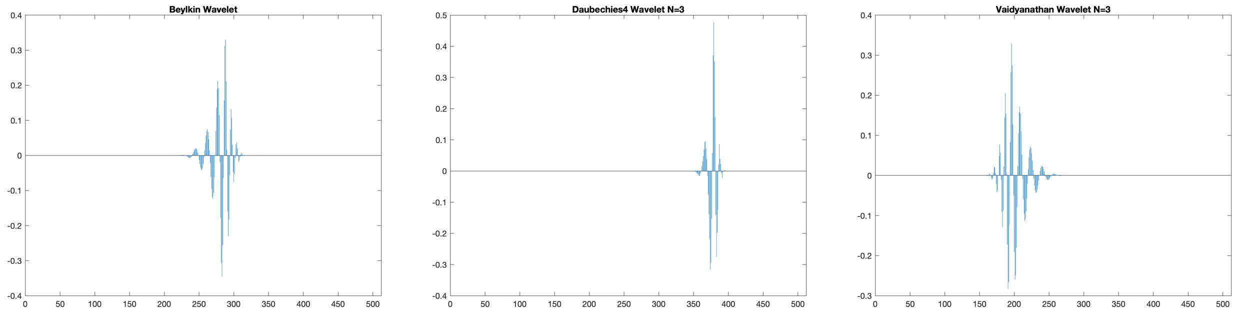

Figure 3: The three level 3 wavelets

4 Part 4

For this part of the project the BayesShrink method for reducing noise is looked at. To do this we first preform a wavelet decomposition on the source image. After this we threshold each of the sub-bands using $T = \frac{\sigma_i^2}{\sigma}$. After the threshold was preferred the images was reconstructed useing the command. The input noisy image and the denoised image can be seen in the figures. The noisy image is on the left and its denoised image is on the right.

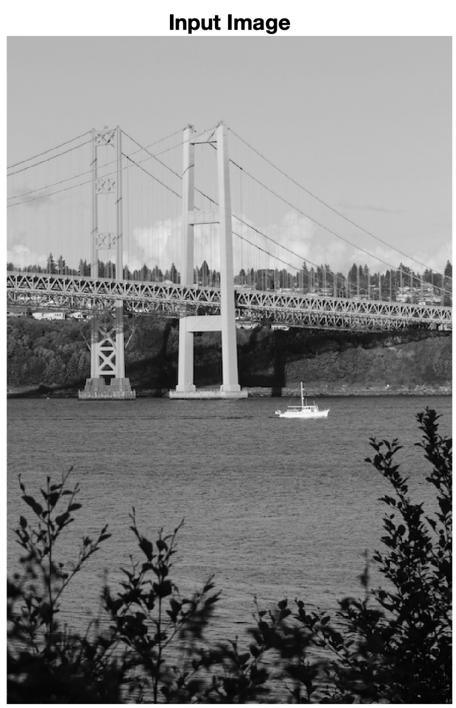

Figure 4: The input used for this part of the project.

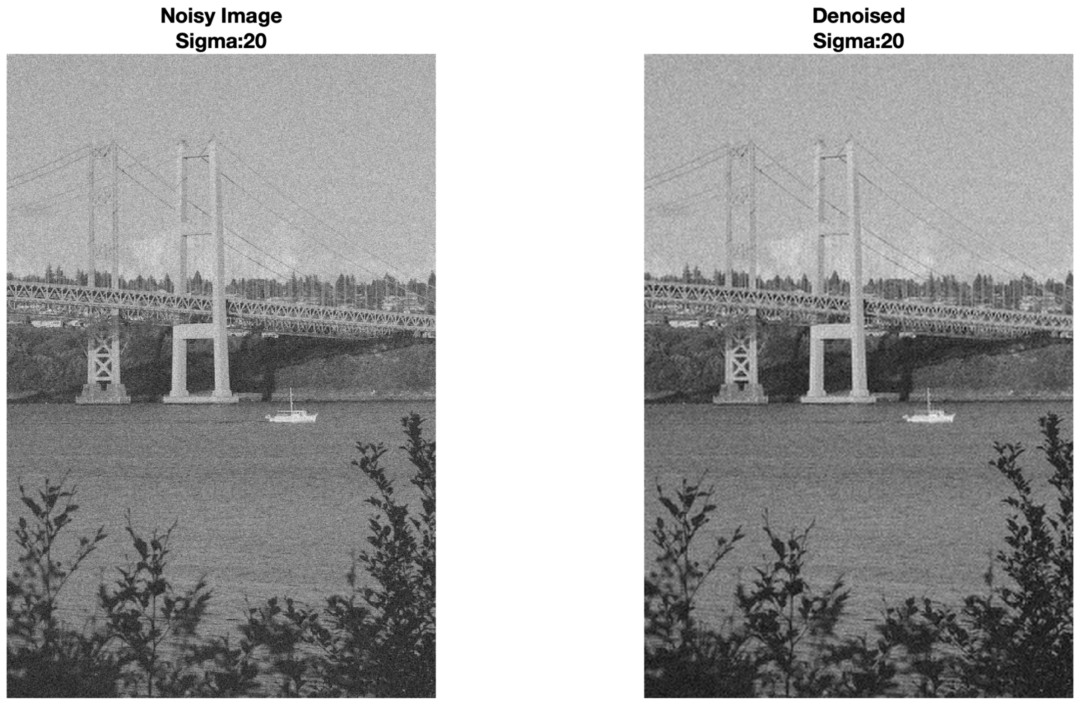

Figure 5: First run of the BayesShrink with $\sigma = 20$

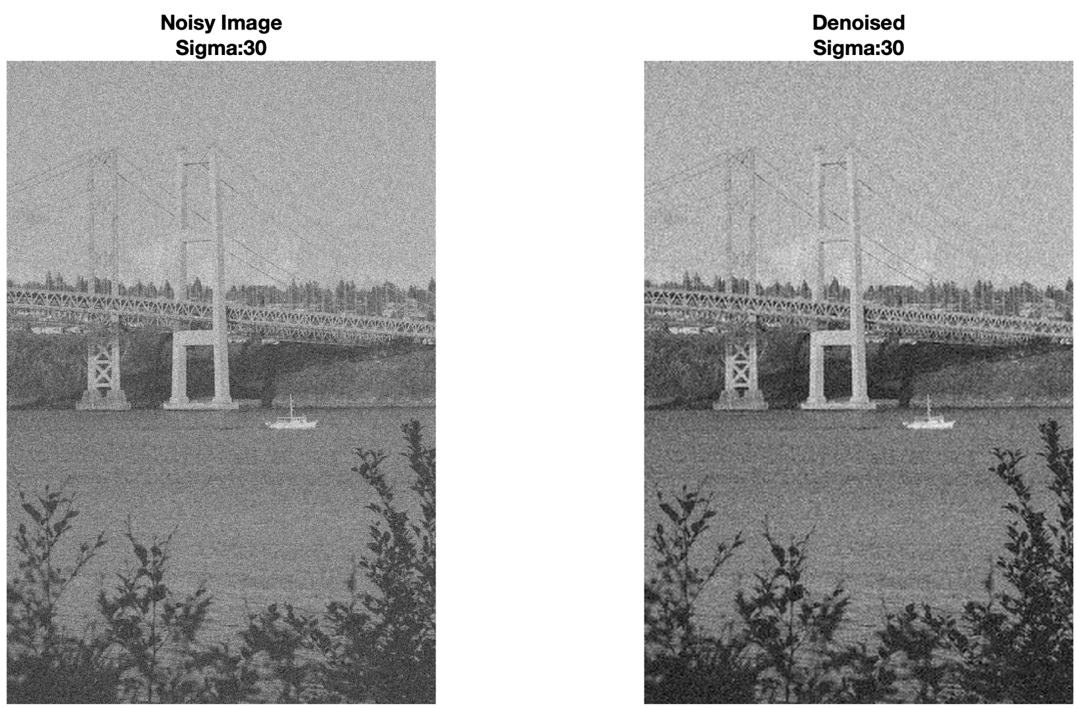

Figure 6: Second run of the BayesShrink method using $\sigma = 30$

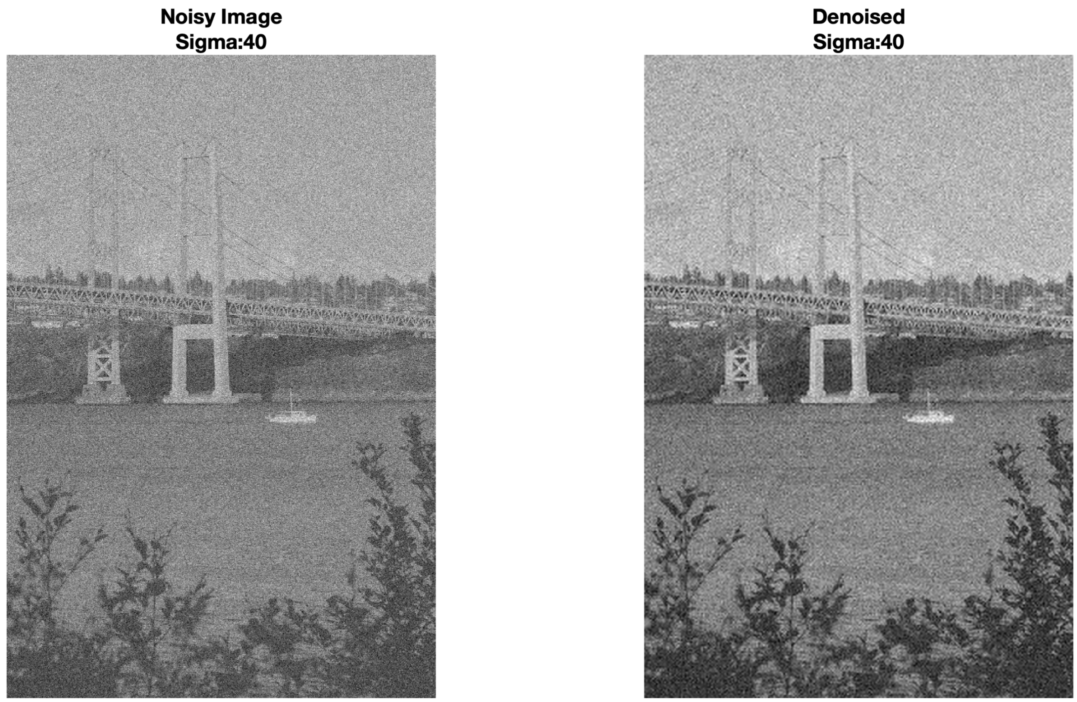

Figure 7: Second run of the BayesShrink method using $\sigma = 40$

A MATLAB Code

This is the code used for the coding portions of this project.

```
%Housekeeping commands
clear all
close all

Signal=ones(1,512); %input signal

[C,L]=wavedec(Signal,1,'beyl'); %wavelet decomp
%changing the C arary to have a 1 in the level 1 aread
C=zeros(size(C));
C(300)=1;
X=waverec(C,L,'beyl'); %wavetel recombination
%dispalying wavelt
figure
stem(X,'MarkerSize',0.1)
xlim([0,512])
title('Beylkin Wavelet N=1')
exportgraphics(gcf,'BeylkinN1.png','Resolution',300)

[C,L]=wavedec(Signal,1,'vaid');
C=zeros(size(C));
C(300)=1;
X=waverec(C,L,'vaid');
figure
stem(X,'MarkerSize',0.1)
xlim([0,512])
title('Vaidyanathan Wavelet N=1')
exportgraphics(gcf,'VaidyanathanN1.png','Resolution',300)

[C,L]=wavedec(Signal,1,'db4');
C=zeros(size(C));
C(300)=1;
X=waverec(C,L,'db4');
figure
```

```
stem(X,'MarkerSize',0.1)
xlim([0,512])
title('Daubechies4 Wavelet N=1')
exportgraphics(gcf,'Daubechies4N1.png','Resolution',300)

[C,L]=wavedec(Signal,2,'beyl');
C=zeros(size(C));
C(245)=1; % 1 in level 2
X=waverec(C,L,'beyl');
figure
stem(X,'MarkerSize',0.1)
xlim([0,512])
title('Beylkin Wavelet N=2')
exportgraphics(gcf,'BeylkinN2.png','Resolution',300)

[C,L]=wavedec(Signal,2,'vaid');
C=zeros(size(C));
C(245)=1; % 1 in level 2
X=waverec(C,L,'vaid');
figure
stem(X,'MarkerSize',0.1)
xlim([0,512])
title('Vaidyanathan Wavelet N=2')
exportgraphics(gcf,'VaidyanathanN2.png','Resolution',300)

[C,L]=wavedec(Signal,2,'db4');
C=zeros(size(C));
C(245)=1; % 1 in level 2
X=waverec(C,L,'db4');
figure
stem(X,'MarkerSize',0.1)
xlim([0,512])
title('Daubechies4 Wavelet N=2')
exportgraphics(gcf,'Daubechies4N2.png','Resolution',300)

[C,L]=wavedec(Signal,3,'beyl');
```

```
C=zeros(size(C));
C(120)=1; % 1 in level 3
X=waverec(C,L,'beyl');
figure
stem(X,'MarkerSize',0.1)
xlim([0,512])
title('Beylkin Wavelet')
exportgraphics(gcf,'BeylkinN3.png','Resolution',300)

[C,L]=wavedec(Signal,3,'vaid');
C=zeros(size(C));
C(120)=1; % 1 in level 3
X=waverec(C,L,'vaid');
figure
stem(X,'MarkerSize',0.1)
xlim([0,512])
title('Vaidyanathan Wavelet N=3')
exportgraphics(gcf,'VaidyanathanN3.png','Resolution',300)

[C,L]=wavedec(Signal,3,'db4');
C=zeros(size(C));
C(120)=1; % 1 in level 3
X=waverec(C,L,'db4');
figure
stem(X,'MarkerSize',0.1)
xlim([0,512])
title('Daubechies4 Wavelet N=3')
exportgraphics(gcf,'Daubechies4N3.png','Resolution',300)

%Housekeeping commands
clear all
close all

Bayes('beyl',20) %sigma of 20
Bayes('beyl',30) %sigma of 30
Bayes('beyl',40) %sigma of 40
```

```
function Bayes(wav,s)
    blankreigion=[0.0013 0.1523;0.6445 1]; %array for ratio of blank region

    Bridges=imread('Bridges.jpeg'); %Reading in input
    Bridges=cast(im2gray(Bridges),'double'); %converting to graycale
    noise=cast(floor(randn(size(Bridges)).*s),'double'); %creating noise
    NoisyBridge=Bridges+noise; %adding noise

    %displaying noisy image
    figure
    imagesc(NoisyBridge)
    colormap("gray")
    title({'Noisy Image','Sigma:'+string(s)})
    axis off image
    exportgraphics(gcf,'NoiseS'+string(s)+'.png','Resolution',300)



    [C,L]=wavedec2(NoisyBridge,1,wav); %wavelt decomp

    [H1,V1,D1]=detcoef2('all',C,L,1); %coef detection
    A=appcoef2(C,L,wav,1); %approx finding

    sz=size(H1); %size of coef array
    loc=round(blankreigion.*sz'); %indexes of blank spot
    sh=std2(H1(loc(1,1):loc(1,2),loc(2,1):loc(2,2))); % sigma H
    sv=std2(V1(loc(1,1):loc(1,2),loc(2,1):loc(2,2))); % sigma V
    sd=std2(D1(loc(1,1):loc(1,2),loc(2,1):loc(2,2))); % sigma D

    %tresholding
    H1t=Thresh(H1,sh^2 /s);
    V1t=Thresh(V1,sv^2 /s);
    D1t=Thresh(D1,sd^2 /s);

    %wavelet recompasition
    BridgeOut=idwt2(A,H1t,V1t,D1t,wav);
```

```
    %displaying out image
    figure
    imagesc(BridgeOut)
    colormap('gray')
    axis image off
    title({'Denoised','Sigma:'+string(s)})
    exportgraphics(gcf,'DenoisedS'+string(s)+'.png','Resolution',300)

    function T=Thresh(A,t)
        sgn=sign(A); %getting sign of array
        B=abs(A)-t; % sombatracting thershold from abs of array
        %setting thessholdinged values to 0
        [x,y,~]=find(B<0);
        B(x,y)=0;
        T=B.*sgn; %reintroducing the sign
    end
end
```