Homework 8

David Bolin

December 1, 2005

1. Create a MATLAB function that inputs a signal, a lowpass and a highpass filter, and performs the wavelet transform. Apply it repeatedly to a piecewise linear continuous signal of your choice using the Daubechies-2 filters. Can you detect the discontinuity in the 1st derivative of the signal?

Solution

```
%This function decomposes the signal f into one highpass
%component and one lowpass component.

function [H,L] = wavetrans(f,l,h)

lf = length(f);
extsize = length(l)-1;
last = lf + extsize;
f = wextend('1D','spd',f,extsize);
Htemp = conv(f,h);
Ltemp = conv(f,l);
H = Htemp(extsize+1:2:length(Htemp)-extsize);
L = Ltemp(extsize+1:2:length(Htemp)-extsize);
```

The program makes a smooth extension of the signal on both sides, the lenght of the extension is the filter length minus one. The extended signal is then convolved with the filters, the extensions are removed, and the signals gets downsampled. I used the following program for applying the transform repeatedly to a signal.

```
%This program decomposes the signal f, n times.
%in each step we keep the lowpass component for
%further decomposition.

function High = waverecurs(f,l,h,n,x)
figure(1)
L = f;
for i=1:n
    [H,Lnew] = wavetrans(L,l,h);
    L = Lnew;
    X = linspace(0,x,length(H));
    plot(X,H)
    pause
end

High = H;
```

I used the following testfunction

```
f1 = linspace(1,2,256);
f2 = 2*f1;
f = [f1,f2(2:length(f2))];
p0 = (1+sqrt(3))/4;
p1 = (3+sqrt(3))/4;
p2 = (3-sqrt(3))/4;
p3 = (1-sqrt(3))/4;
l = [p0, p1, p2, p3];
h = [-p3, p2, -p1, p0];
```

See the last pages for figures. Figure 1 is the plot of the piecewise linear test signal, in figure 2, figure 3 and figure 4 we see that there are peaks in the highpass components in every step where the discontinuity in the derivative is.

2. Create a MATLAB function that checks whether a given pair of lowpass and highpass filters satisfies the conjugate quadrature rule.

Solution

I have written a program that checks if the two filters $h = [h_1, h_2, \ldots]$ and $g = [g_1, g_2, \ldots]$ are Conjugate Quadrature filters, i.e. the filter coefficients must satisfy

$$\sum_{k\in\mathbb{Z}} h_k \bar{h}_{k-2n} = \delta_{0n} \ \ \forall n \in \mathbb{Z}$$

$$\sum_{k\in\mathbb{Z}} h_k \bar{h}_{k-n} = (-1)^n \sum_{k\in\mathbb{Z}} g_k \bar{g}_{k-n} \ \ \forall n \in \mathbb{Z}$$

$$\sum_{k\in\mathbb{Z}} h_k \bar{g}_{k-2n} = 0 \ \ \forall n \in \mathbb{Z}$$

I found these formulas in the paper "notes on wavelets 2001" by gerd Grubb, found on the homepage www.math.ku.dk/ grubb/notes/wave/wave8.ps. We see that the coefficients in the filters must be normalized, i.e. the vectors h and g with the filter coefficients must have norm one. The program returns 1 if the filters are CQF and 0 if not.

```
function true  = cqrcheck(h,g)

N = length(h);
x = 1;

if h*h'- 1>1e-15
    x=0;
end

for n=1:N
    hbartemp = [zeros(2*n,1); h'];
    htemp = [h,zeros(1,2*n)];
    s = htemp*hbartemp;
    if s > 1e-15
        x = 0;
```

```
    end
end
for n=1:N
    hbartemp = [zeros(n,1); h'];
    htemp = [h,zeros(1,n)];
    hsum = htemp*hbartemp;
    gbartemp = [zeros(n,1); g'];
    gtemp = [g,zeros(1,n)];
    gsum = gtemp*gbartemp;
    if hsum - ((-1)^(n))*gsum > 1e-15
        x = 0;
    end
end

for n=1:N
    gbartemp = [zeros(2*n,1); g'];
    htemp = [h,zeros(1,2*n)];
    s = htemp*gbartemp;
    if s > 1e-15
        x = 0;
    end
end
true = x;
```

I tested the program using the haar filters, and the Daubechie 2 filters (normalized), and it works.