2024 Spring VLSI DSP Homework Assignment #2

Due date: 2024/3/31

Q1. LMS filter design

For a least mean square (LMS) adaptive filter, assume the filter is of the form finite impulse response (FIR) and m-tap long (i.e., with m coefficients $b_0 \sim b_{m-1}$ for $x(n) \sim x(n-m+1)$). Given an input signal consisting of 2 frequency components

$s(n) = \sin(2\pi * n/16) + 0.5 * \cos(2\pi * n/4)$

develop an adaptive low pass filter design

Set the target as a low pass filter to remove the high frequency component $\cos(2\pi * n/4)$ and use $\sin(2\pi * n/16)$ as the desired (or training) signal for LMS adaptation.

- write a Matlab code to simulate the LMS based adaptive filtering. Calculate the RMS (root mean square) value of the latest 16 prediction errors
  (i.e., $r(n) = \text{sqrt}((e^2(n) + e^2(n-1) + \ldots + e^2(n-15))/16)$ and the adaptation is converged if this value is less than 5% of RMS (root mean square) value of the desired signal, which equals 0.05/sqrt(2).
- Please determine the minimum filter length m and the step size μ so that the adaptation can converge in no more than 5000 iterations. The initial values of the filter coefficients $b_0 \sim b_{m-1}$ are set to $1/m$.
- Show the plot of "r(n)" versus "n" and indicate when the filter converges, i.e. how many training samples are required
- Show the plot of filter coefficients $b_i(n)$, for $i = 0 \sim m-1$, versus "n" and see if the values of filter coefficients remain mostly unchanged after convergence
- Apply a 64-point FFT to the impulse response of the converged filter and verify the filter is indeed a low pass one. Note that the input vector to the 64-point FFT is
  (b_0, b_1, ….., b_{m-1}, 0,0,…..,0) with $64-m$ trailing zeros.
- Conduct simulation with a sufficiently large number of samples to see how small the value of "r" can be (the convergence bias)

Q2. Discrete Wavelet Transform

For a discrete wavelet transform (DWT) adopting (9/7) filters, i.e. the <u>low pass</u> filter $h(i)$ is 9-taped and the <u>high pass</u> filter $g(i)$ is 7-taped. Both filters are liner phased and have symmetric coefficients. The filter coefficients are given in Table 1. For a corresponding inverse discrete wavelet transform, the <u>low pass</u> filter $q(i)$ is 7-taped and the <u>high pass</u> filter $p(i)$ is 9-taped. The filter coefficients are given in Table 2.

Table 1. Analysis filter coefficients for the floating point 9/7 filter

Analysis Filter Coefficients		
i	Lowpass Filter h_i	Highpass Filter g_i
0	0.852698679009	-0.788485616406
±1	0.377402855613	0.418092273222
±2	-0.110624404418	0.040689417609
±3	-0.023849465020	-0.064538882629
±4	0.037828455507	

Note: the high pass and low pass filter notations here are opposite to those in the lecture note

Table 2. Synthesis filter coefficients for the floating point 9/7 filter

Synthesis Filter Coefficients		
i	Low pass Filter q_i	High pass Filter p_i
0	0.788485616406	-0.852698679009
±1	0.418092273222	0.377402855613
±2	-0.040689417609	0.110624404418
±3	-0.064538882629	-0.023849465020
±4		-0.037828455507

a) For a 512×512 gray scale image (will be provided along with the homework assignment), please conduct a 2-D 3-level DWT transform (as shown in Figure 1) and show the transformed result. Then conduct a 2-D 3-level IDWT to convert it back. Please compare if the reconstructed image (after IDWT) is same as the original image by calculating its PSNR value.

b) By setting all three level 1 sub-bands HL1, LH1 and HH1 coefficients to zeros and perform IDWT. See how the reconstructed image is different from the original one and calculate its PSNR value.

Note 1: the filter orders for analysis (DWT) is (low pass 9/ high pass 7) and the filter orders for synthesis is (low pass 7/ high pass 9). And the filter coefficients have the following relations $h(i) = (-1)^{i+1} p(i), \quad g(i) = (-1)^{i} q(i)$

Note 2: when performing down sampling in each octave, low pass filters always keep the **odd** numbered output data while high pass filters always keep the **even** numbered output data. In up-sampling process of IDWT, the discarded data are replaced with zeros and inserted to the output data stream.

Note 3: at the boundary of the image, you need to perform a symmetric extension to obtain the pixel values for $h(i),\ \ g(i),\ \ i<0$. The extension is illustrated in Figure 3.

Note 4: The PSNR is calculated as

$$\mathrm{MSE} = \frac{1}{\mathrm{M}\cdot\mathrm{N}}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(I(i,j)-\hat{I}(i,j)\right)^2$$

$$PSNR = 10\log_{10}\left(\frac{MAXI^2}{\mathrm{MSE}}\right)$$

Where $I(i,j)$ is the original image and $\hat{I}(i,j)$ is the reconstructed image after performing DWT and IDWT. MAXI is the maximum possible value of a pixel. If it's an 8-bit pixel, MAXI is 255. For a reconstructed image with a PSNR value greater than 50dB, it is considered visually lossless.

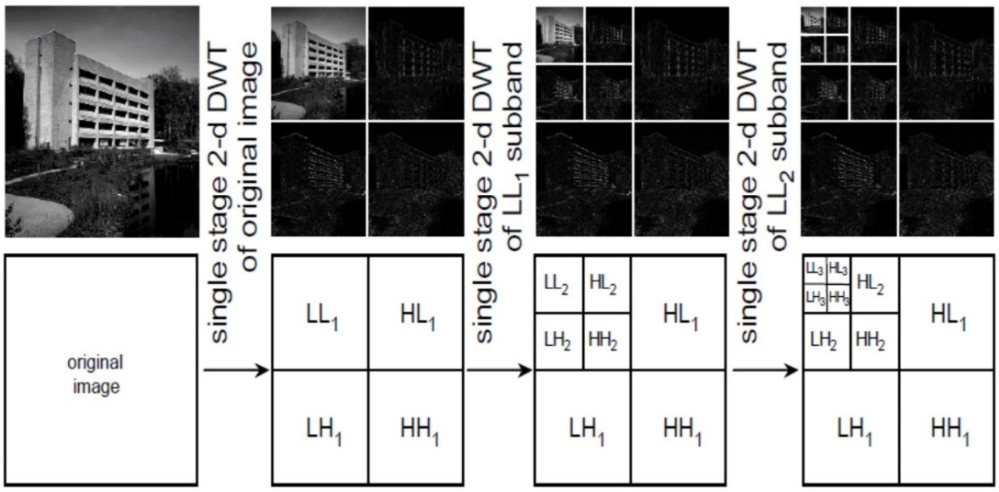
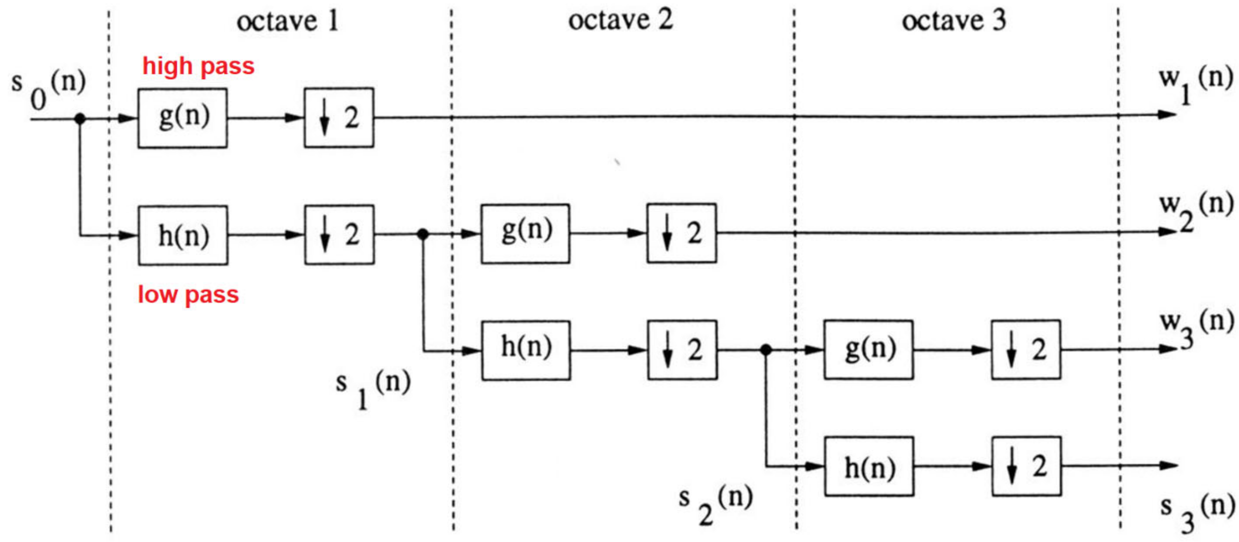
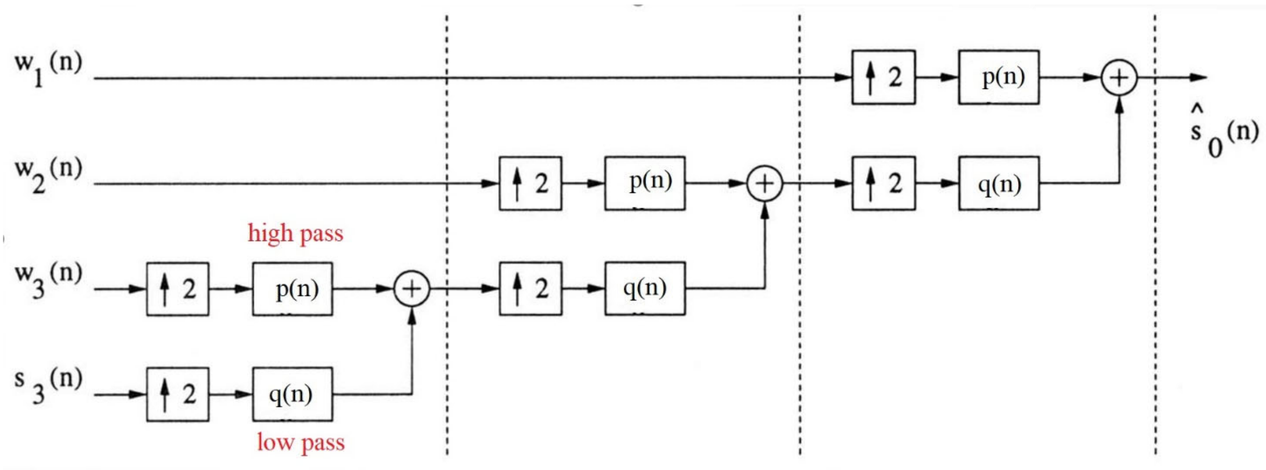


Figure 1. a 3-level DWT example



(a) One-dimensional 3-level DWT transform



(b) One-dimensional 3-level IDWT transform

Figure 2. 1-D 3-level DWT versus IDWT

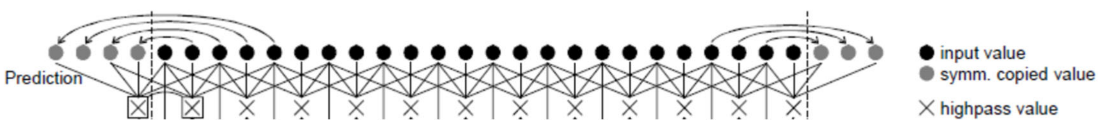


Figure 3. Symmetric extension scheme for boundary pixels