The University Of Sheffield.

EEE6081 (EEE421)
Visual Information Engineering (VIE)
Topic 6: Wavelet Transforms -II

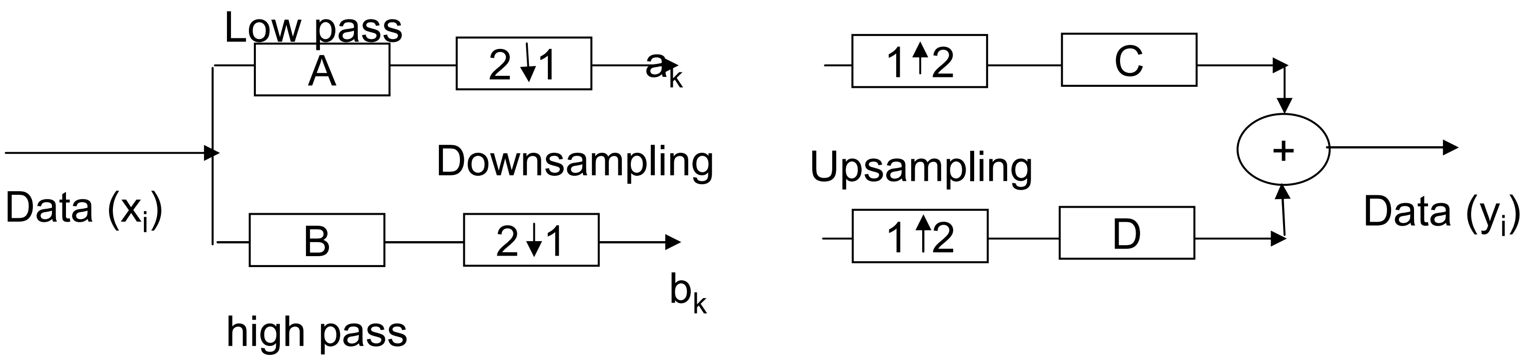
- Biorthogonal wavelets
  - Filter bank design (Revision)
  - Orthogonal filter banks
  - Biorthogonal filter banks
  - Examples
- Lifting based wavelet transform design & implementation
  - The lifting concept
  - Haar wavelet using lifting
  - 5/3 biorthogonal wavelet using lifting
  - Advantages of using lifting

- Additional Reading:
  - http://cm.bell-labs.com/who/wim/papers/#lifting
  - http://cm.bell-labs.com/who/wim/papers/papers.html#iciam95

Dr Charith Abhayaratne
Email: c.abhayaratne@sheffield.ac.uk
Office: F176 Tel: 25893

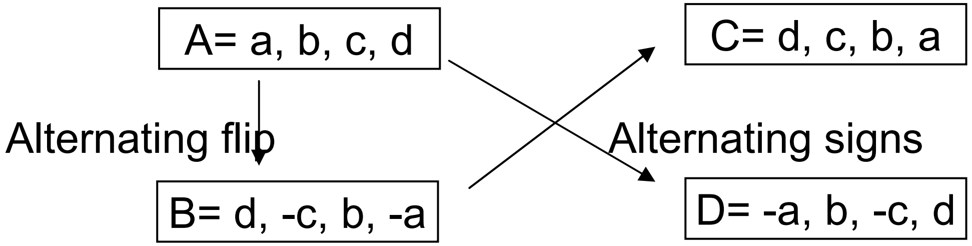
The University Of Sheffield.

Filter Banks (Revision)



Analysis filter bank

Synthesis filter bank



C(z)=B(-z) and D(z)=-A(-z)

Filter Bank design:
For filter A
Orthogonality condition for the filter bank:

$$\sum_i h_i h_{i+2k} = \delta_{0k}$$

Regularity condition for the filter bank:

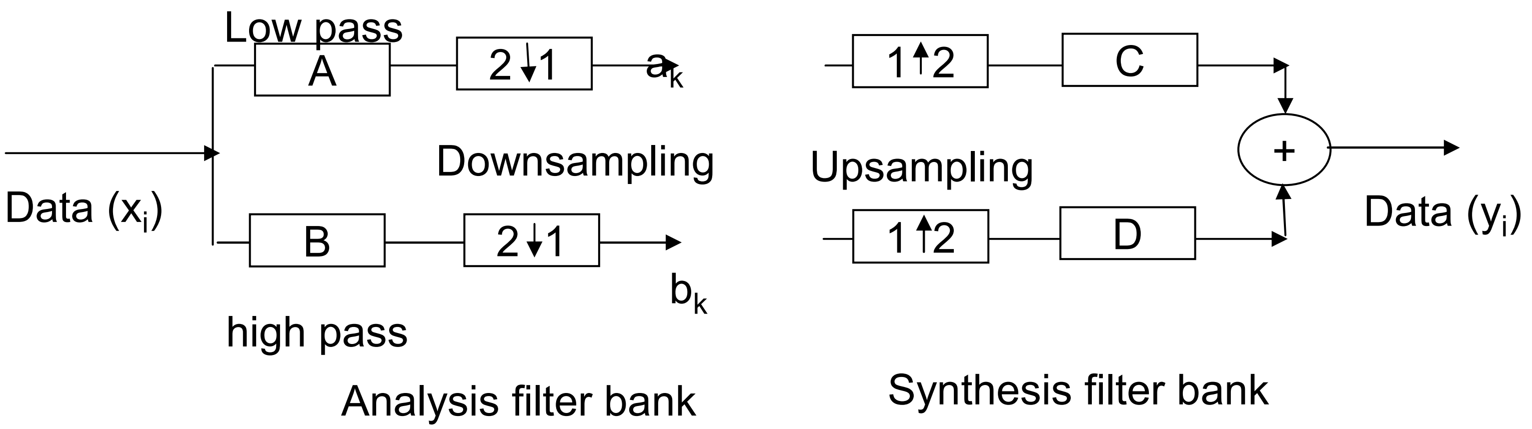
B is a high pass filter only its coefficients add up to zero. This and Perfect reconstruction condition mean,

$$\sum_i h_i = \sqrt{2}$$

Filter Banks

- We computed filter bank filter coefficients for N=2, N=3, and N=4.
- Not possible to design odd length filter banks.
- Can we make the filter symmetric?
  - i.e., h0=h3 and h1=h2 in N=4
- Filters in orthogonal filter banks can't be symmetric. Therefore, they have phase distortion (no linear phase response).
- A solution is Biorthogonal filter banks.
- In orthogonal filter banks, we got C by time reversing the filter A. If we designed filter A using the design constraints then we were able to get other two (B & D) by using the Perfect reconstruction (PR) conditions.

Biorthogonal Filter Banks



A and C do not have to be derived from the same filter. Instead we choose:

A= { h_0, h_1, h_2, …} and C = { f_0, f_1, f_2, …} .

But A and C must obey an orthogonality condition:

$$\boxed{\sum_i h_i f_{i+2k} = \delta_{0k}}$$

$$k = 0: \quad \sum_i h_i f_i = 1$$

$$k \neq 0: \quad \sum_i h_i f_{i+2k} = 0$$

Biorthogonal Filter Banks

Regularity conditions should also be satisfied:

$$\sum_i h_i = \sqrt{2}$$

B and D are a high pass filters only its coefficients add up to zero.

$$\sum_i (-1)^i h_i = 0$$

$$\sum_i (-1)^i f_i = 0$$

This and Perfect reconstruction condition mean:

C(z)=B(-z) and D(z)=-A(-z)

Then B and D are in the form:

D= { $-h_0$, h_1, $-h_2$, …} and B = { f_0, $-f_1$, f_2, …}

Biorthogonal Filter Banks

In Biorthogonal filter banks, filters can be either odd or even length.

We can also make filters symmetric.

Different lengths for A and C can be chosen.

In the designing process, A and C are assumed to be the same length, but leading and trailing coefficients in pairs could be zero.

Exercise: Write down design equations for 6/4 filter bank: (i.e., length of A is 6 and length of C is 4.)

Biorthogonal Filter Banks

Exercise: Filters A and C for the odd length filter bank 9/7 is shown below in integer format:

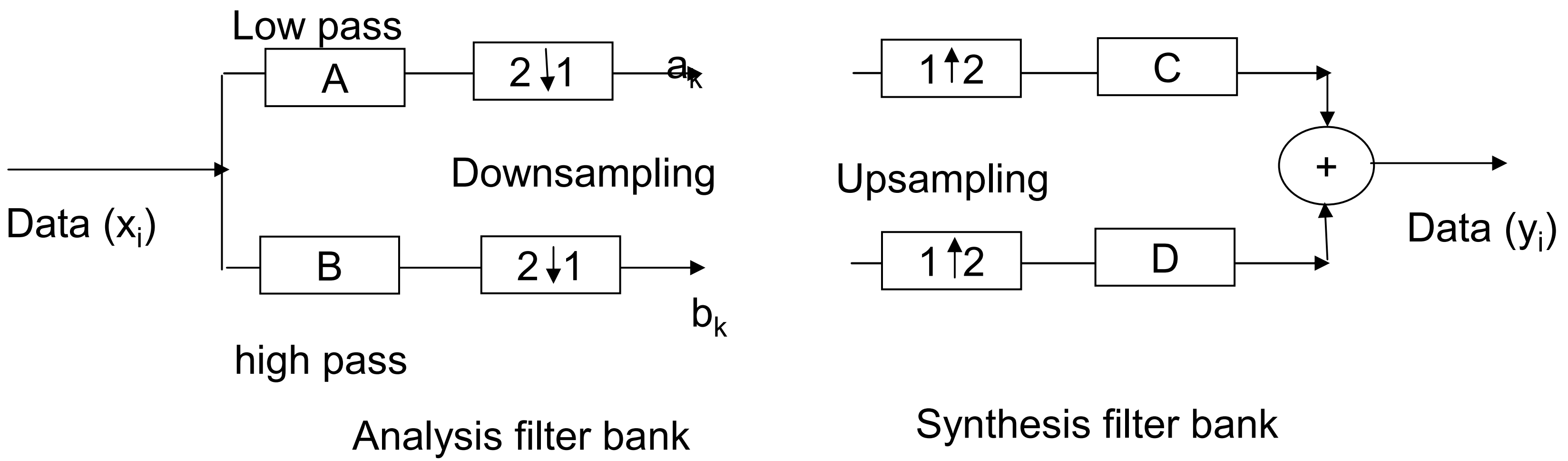
A = {1 0 -8 16 46 16 -8 0 1}

C = {0 -1 0 9 16 9 0 -1 0}

Is this perfect reconstructing?
What has to be done to make it regular?

Biorthogonal Filter Banks

Filter Banks - Implementation



Exercise: Write down pseudo-codes for the forward and inverse wavelet transforms using filter banks.

Filter Banks - Implementation

Estimate the number of multiplications and additions in the forward transform:

Assume data length is N
The length of the low pass filter (A) is L
The length of the high pass filter (B) is H

For low pass:

For high pass:

Total:

Any drawback using filter bank implementation?

Filter Banks - Implementation

To overcome that (and to speed up) we could swap the order of filtering and down sampling:

Example: It includes the following steps:

1. Obtain one channel with down sampled input signal
2. Obtain the other channel with the discarded data. (This is the same as down sampling a delayed input signal)
3. Now if we can decompose the filters A and B into $A_{0,}$ $A_{1,}$ B_0, and B_1 , we can design a fast implementation. [how to find $A_{0,}$ $A_{1,}$ B_0, and B_1?]

This is called a lattice structure signal flow based implementation. We can covert this to a ladder structure signal flow and make it even faster. Converting a filter bank to the ladder structure is the basis for lifting.

Lifting

The lifting concept:

1. First decompose the Filters A and B into their even indexed and odd indexed components (known as polyphase components) and obtain the polyphase matrix (in z-transform representation)

$$M(z) = \begin{bmatrix} A_{even}(z) & A_{odd}(z) \\ B_{even}(z) & B_{odd}(z) \end{bmatrix}$$

2. Now, considering the input signal is also represented using polyphase components ($X_{even}(z)$ and $X_{odd}(z)$), we can write:

$$\begin{bmatrix} L(z) \\ H(z) \end{bmatrix} = \begin{bmatrix} A_{even}(z) & A_{odd}(z) \\ B_{even}(z) & B_{odd}(z) \end{bmatrix} \begin{bmatrix} X_{even}(z) \\ X_{odd}(z) \end{bmatrix}$$

3. The polyphase matrix can be factorised into a series of alternating unit diagonal upper and lower triangular matrices followed by a normalisation constant.

$$\begin{bmatrix} L(z) \\ H(z) \end{bmatrix} = \begin{bmatrix} k & 0 \\ 0 & 1/k \end{bmatrix} \begin{bmatrix} 1 & U_n(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -P_n(z) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & U_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -P_1(z) & 1 \end{bmatrix} \begin{bmatrix} X_{even}(z) \\ X_{odd}(z) \end{bmatrix}$$

4. Each of these triangular matrices can be represented as a lifting step.

Lifting

The lifting concept (cont..):

$$\begin{bmatrix} L(z) \\ H(z) \end{bmatrix} = \begin{bmatrix} k & 0 \\ 0 & 1/k \end{bmatrix} \begin{bmatrix} 1 & U_n(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -P_n(z) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & U_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -P_1(z) & 1 \end{bmatrix} \begin{bmatrix} X_{even}(z) \\ X_{odd}(z) \end{bmatrix}$$

4. Each of these triangular matrices can be represented as a lifting step.

$$\text{From Matrix 1}: \quad X^1_{odd}(z) = X_{odd}(z) - P_1(z) X_{even}(z)$$

Usually known as a prediction lifting step

$$\text{From Matrix 2}: \quad X^1_{even}(z) = X_{even}(z) + U_1(z) X_{odd}(z)$$

Usually known as an updating lifting step

$\vdots$

$$\text{Finally} \quad : \quad H(z) = \frac{1}{k} X^n_{odd}(z)$$

$$: \quad L(z) = k X^n_{even}(z)$$

Lifting

The lifting concept (cont..):

5. The lifting steps can be represented as a ladder structure based implementation.
{Draw a block diagram for the lifting transform}

5. Any potential advantage in lifting based implementation of wavelet transforms?

Lifting

The Haar transform in lifting:

Polyphase matrix for the Haar transform is as follows:

$$\begin{bmatrix} L(z) \\ H(z) \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} X_{even}(z) \\ X_{odd}(z) \end{bmatrix}$$

Can factorise into two lifting matrices and the normalisation:

$$\begin{bmatrix} L(z) \\ H(z) \end{bmatrix} = \begin{bmatrix} k & 0 \\ 0 & 1/k \end{bmatrix} \begin{bmatrix} 1 & U \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -P & 1 \end{bmatrix} \begin{bmatrix} X_{even}(z) \\ X_{odd}(z) \end{bmatrix}$$

The values of U, P and k: P=1 U=1/2 k=?

What are the lifting steps for the forward transform?

Lifting

The Haar transform in lifting (Cont…):

What are the lifting steps for the inverse transform?

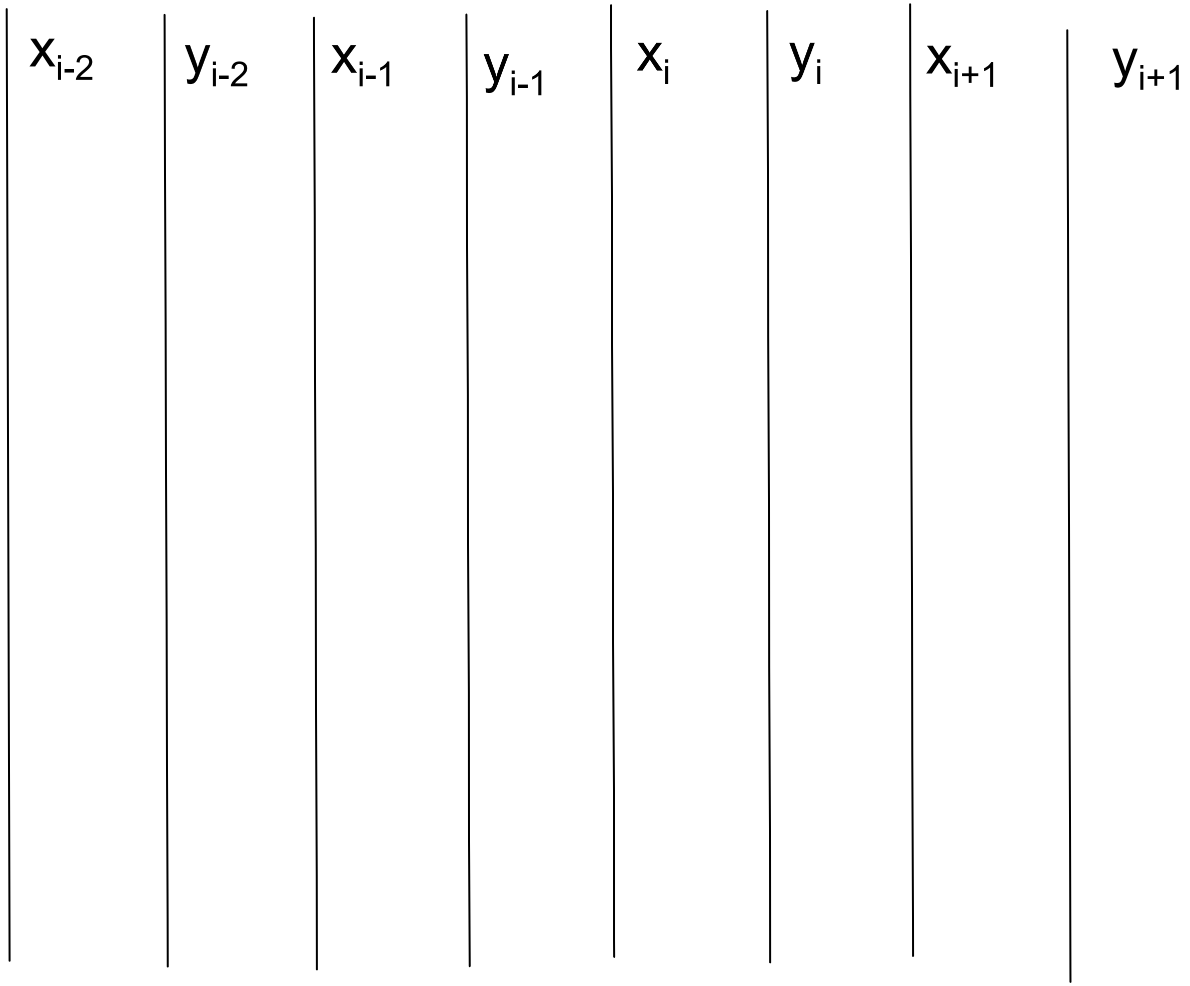
The inverse lifting steps can be found by changing the ------------------- and the ---------------------- of the forward transform.

Write the pseudo code for the forward and inverse Haar transform for the lifting based realisation:

Lifting

The Haar transform in lifting (Cont…):

Draw the signal flow diagram for lifting based Haar transform



Lifting

The 5/3 transform in lifting:

What happens if we use 2 neighbouring samples to predict and update:

x_{i-2} y_{i-2} x_{i-1} y_{i-1} x_i y_i x_{i+1} y_{i+1}

Lifting

The 5/3 transform in lifting:

Using the signal flow diagram in slide #18, write down the lifting steps for the 5/3 forward and inverse transforms:

Verify that using these lifting steps you can get the low pass and high pass filters with the following coefficients:

A= {-1, 2, 6, 2, -1} sqrt(2)/8
C= { 0, 1, 2, 1, 0} sqrt(2)/ 4

Lifting

Advantages of lifting:

1. Fast computation
2. In-place computation – efficient use of memory
3. Integer-to-integer transforms (with perfect reconstruction)
4. Easy design of new wavelet transforms (by using different prediction and update methods)
5. Easy design of content-adaptive and non-linear transforms

Lifting

Integer – to - integer transforms

The Lifting steps:

$y' = \; y - P(x)$
$x' = \; x + U(y')$

Let's say we started with integers (i.e., x and y are integers)

But due to filter operations x' and y' are not integers.

If we round them to the nearest integer to convert to integers we cannot get the perfect reconstruction.

Instead, we can round the terms at the right side of the lifting operation sign.

$$y' = \; y - \mathrm{round}\{P(x)\}$$
$$x' = \; x + \mathrm{round}\{U(y')\}$$

Lifting

Integer – to – integer transforms:

Example: Show the lifting steps for integer-to-integer Haar transform

THE END of WAVELET TRANSFORMS PART of EEE6081(421)