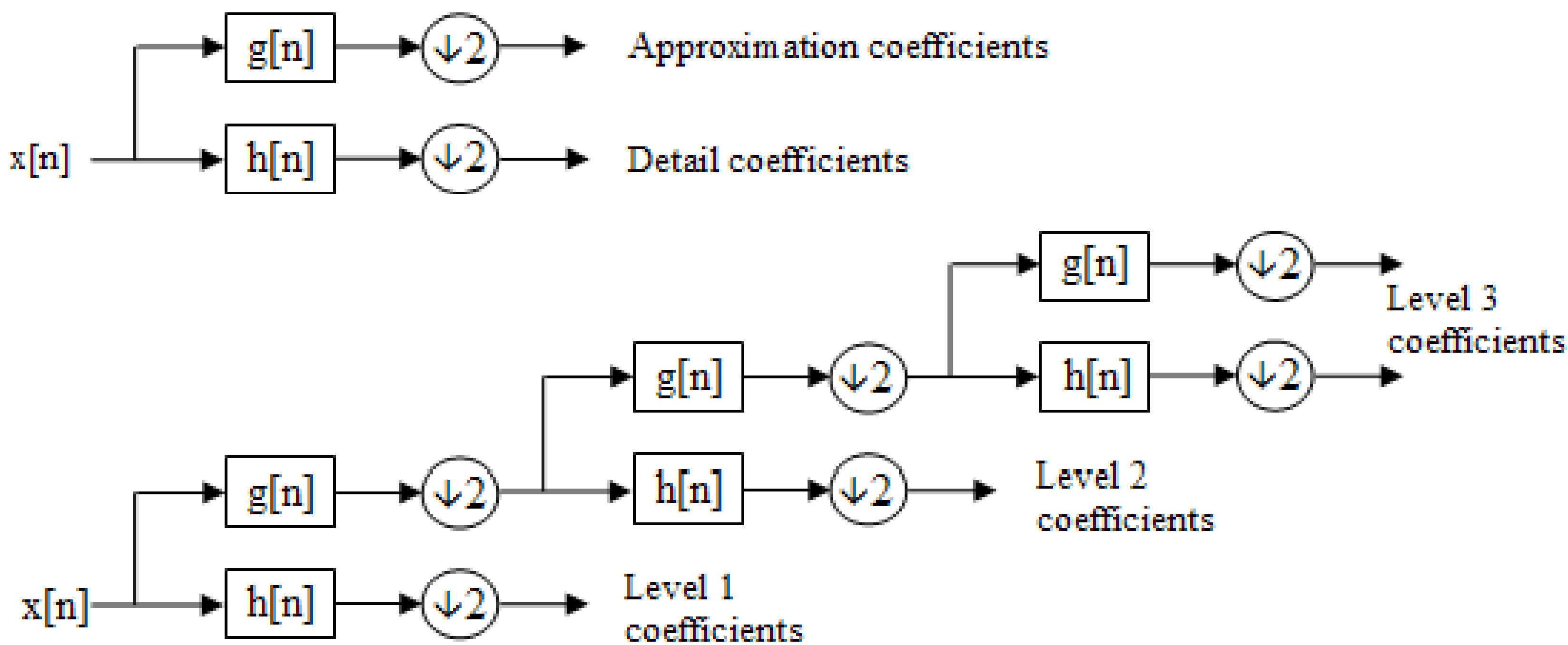
Usual implementation

Lowpass filter = g

Highpass filter = h



2-D Biorthogonal Wavelets

We are given two low-pass symmetric filters. For example, a low-pass filter

$LA = [LA_{-4} , LA_{-3} , LA_{-2} , LA_{-1} , LA_0 , LA_1 , LA_2 , LA_3 , LA_4]$
where $LA_{-4} = LA_4$, $LA_{-3} = LA_3$, $LA_{-2} = LA_2$, $LA_{-1} = LA_1$,

and a low-pass filter

$LS = [LS_{-4} , LS_{-3} , LS_{-2} , LS_{-1} , LS_0 , LS_1 , LS_2 , LS_3 , LS_4]$,
where $LS_{-4} = LS_4$, $LS_{-3} = LS_3$, $LS_{-2} = LS_2$, $LS_{-1} = LS_1$.

From LS we build an analysis high-pass filter

$HA = [-LS_4 , LS_3 , -LS_2 , LS_1 , -LS_0 , LS_1 , -LS_2 , LS_3 , -LS_4]$,

and from LA a synthesis high-pass filter

$HS = [-LA_4 , LA_3 , -LA_2 , LA_1 , -LA_0 , LA_1 , -LA_2 , LA_3 , -LA_4]$

2-D Biorthogonal Wavelets

Matrix interpretation

fold

Encoding matrix T_A

$$
\begin{array}{cccc|c|cccc|c|cccc}
L_4 & L_3 & L_2 & L_1 & L_0 & L_1 & L_2 & L_3 & L_4 & & & & & \\
 & H_4 & H_3 & H_2 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4 & & & & \\
 & & L_4 & L_3 & L_2 & L_1 & L_0 & L_1 & L_2 & L_3 & L_4 & & & \\
 & & & H_4 & H_3 & H_2 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4 & & \\
 & & & & L_4 & L_3 & L_2 & L_1 & L_0 & L_1 & L_2 & L_3 & L_4 & \\
 & & & & & H_4 & H_3 & H_2 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4
\end{array}
$$

fold

L: LA low-pass filter

H:High-pass filter built from LS

Fold over left/right pivot columns and add.

fold

$$
\begin{array}{cccccc}
L_4 & & & & & \\
H_3 & H_4 & & & & \\
L_2 & L_3 & L_4 & & & \\
H_1 & H_2 & H_3 & H_4 & & \\
\hline
L_0 & L_1 & L_2 & L_3 & L_4 & \\
\hline
H_1 & H_0 & H_1 & H_2 & H_3 & H_4 \\
L_2 & L_1 & L_0 & L_1 & L_2 & L_3 \\
H_3 & H_2 & H_1 & H_0 & H_1 & H_2 \\
L_4 & L_3 & L_2 & L_1 & L_0 & L_1 \\
\hline
 & H_4 & H_3 & H_2 & H_1 & H_0 \\
\hline
 & & L_4 & L_3 & L_2 & L_1 \\
 & & & H_4 & H_3 & H_2 \\
 & & & & L_4 & L_3 \\
 & & & & & H_4
\end{array}
$$

fold

Decoding matrix T_S

L: LS low-pass filter

H:High-pass filter built from LA

Fold over top/bottom pivot rows and add.

Symmetric low-pass filters LA and LS are given

Example: Encoding matrix of dimensions 12x12. The matrix is obtained by first lining up the offset filters, then folding/adding about pivot columns as shown below so all resulting values fit within 12x12 red box as shown on next page.

fold **Outline of Encoding Matrix T_A**

$$
\begin{array}{cccccccccccccccccccc}
L_4 & L_3 & L_2 & L_1 & L_0 & L_1 & L_2 & L_3 & L_4 & & & & & & & & & & & \\
 & H_4 & H_3 & H_2 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4 & & & & & & & & & & \\
 & & L_4 & L_3 & L_2 & L_1 & L_0 & L_1 & L_2 & L_3 & L_4 & & & & & & & & & \\
 & & & H_4 & H_3 & H_2 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4 & & & & & & & & \\
 & & & & L_4 & L_3 & L_2 & L_1 & L_0 & L_1 & L_2 & L_3 & L_4 & & & & & & & \\
 & & & & & H_4 & H_3 & H_2 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4 & & & & & & \\
 & & & & & & L_4 & L_3 & L_2 & L_1 & L_0 & L_1 & L_2 & L_3 & L_4 & & & & & \\
 & & & & & & & H_4 & H_3 & H_2 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4 & & & & \\
 & & & & & & & & L_4 & L_3 & L_2 & L_1 & L_0 & L_1 & L_2 & L_3 & L_4 & & & \\
 & & & & & & & & & H_4 & H_3 & H_2 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4 & & \\
 & & & & & & & & & & L_4 & L_3 & L_2 & L_1 & L_0 & L_1 & L_2 & L_3 & L_4 & \\
 & & & & & & & & & & & H_4 & H_3 & H_2 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4
\end{array}
$$

fold

For example,by adding the columns that are equally offset to the left and right of the pivot column (*i.e. folding from left to right) we get*

$$
\begin{array}{cccccc}
L_0 & L_1+L_1 & L_2+L_2 & L_3+L_3 & L_4+L_4 & \\
H_1 & H_0+H_2 & H_1+H_3 & H_2+H_4 & H_3 & H_4
\end{array}
$$

$$
\begin{array}{cccccccc}
L_2 & L_1+L_3 & L_0+L_4 & L_1 & L_2 & L_3 & L_4 & \\
H_3 & H_2+H_4 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4
\end{array}
$$

Rows 8-12 are unaffected by folding. For rows 9-12 folding is from right to left.

Example: Encoding matrix of dimensions 12x12. The matrix is obtained by first lining up the offset filters, then folding/adding about pivot columns so all resulting values fit within 12x12 red box.

$$
\begin{array}{cccccccccc}
L_0 & L_1+L_1 & L_2+L_2 & L_3+L_3 & L_4+L_4 & & \cdot\;\cdot & \cdot\;\cdot & \cdot\;\cdot & \\
H_1 & H_0+H_2 & H_1+H_3 & H_2+H_4 & H_3 & H_4 & & & & \\
L_2 & L_1+L_3 & L_0+L_4 & L_1 & L_2 & L_3 & L_4 & & & \\
H_3 & H_2+H_4 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4 & & \\
L_4 & L_3 & L_2 & L_1 & L_0 & L_1 & L_2 & L_3 & L_4 & \\
 & H_4 & H_3 & H_2 & H_1 & H_0 & H_1 & H_2 & H_3 & H_4 \\
 & & & & & & & & & \\
 & & \cdot\;\cdot & \cdot\;\cdot & \cdot\;\cdot & & & & &
\end{array}
$$

Example: CDF2.4 filters

LA = [0.03314563036812 -0.06629126073624 -0.17677669529664 ...
0.41984465132951 0.99436891104358 0.41984465132951 ...
-0.17677669529664 -0.06629126073624 0.03314563036812];
LS = [0 0 0 ...
0.35355339059327 0.70710678118655 0.35355339059327 ...
0 0 0];

Encoding low and high Pass filters

L0 = 0.99436891104358
L1 = L-1 = 0.41984465132951
L2 = L-2 = -0.17677669529664
L3 = L-3 = -0.06629126073624
L4 = L-4 = 0.03314563036812
H0 = -0.70710678118655
H1 = H-1 = 0.35355339059327
H2 = H-2 = H3 = H-3 = H4 = H-4 = 0

First and last few rows and columns of encoding matrix T_A:

First 10 rows. First 5 columns.

0.9944	0.8397	-0.3536	-0.1326	0.0663
0.3536	-0.7071	0.3536	0.0000	0.0000
-0.1768	0.3536	1.0275	0.4198	-0.1768
0.0000	0.0000	0.3536	-0.7071	0.3536
0.0331	-0.0663	-0.1768	0.4198	0.9944
0.0000	0.0000	0.0000	0.0000	0.3536
0.0000	0.0000	0.0331	-0.0663	-0.1768
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0331
0.0000	0.0000	0.0000	0.0000	0.0000

Last 10 rows. Last 5 columns.

0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
-0.0663	0.0331	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.4198	-0.1768	-0.0663	0.0331	0.0000
-0.7071	0.3536	0.0000	0.0000	0.0000
0.4198	0.9944	0.4198	-0.1436	-0.0663
0.0000	0.3536	-0.7071	0.3536	0.0000
-0.0663	-0.1436	0.3536	0.8176	0.4198
0.0000	0.0000	0.0000	0.7071	-0.7071

Encoding: $B = P * T_A * A * T_A' * P'$

Example: CDF2.4 filters

```
LA  = [ 0.03314563036812  -0.06629126073624  -0.17677669529664  ...
         0.41984465132951   0.99436891104358   0.41984465132951  ...
        -0.17677669529664  -0.06629126073624   0.03314563036812  ];
LS  = [ 0              0              0              ...
         0.35355339059327   0.70710678118655   0.35355339059327  ...
         0              0              0              ];
```

Decoding low and high Pass filters

```
L0  =  0.70710678118655
L1  =  L-1  =  0.35355339059327
L2  =  L-2  =  L3  =  L3  =  L4  =  L-4  =  0
H0  = -0.99436891104358
H1  =  H-1  =  0.41984465132951
H2  =  H-2  =  0.17677669529664
H3  =  H-3  = -0.06629126073624
H4  =  H-4  = -0.03314563036812
```

First and last few rows and columns of matrix $T_S' = \mathrm{inv}(T_A)$

First 10 rows. First 5 columns.

0.7071	0.8397	0.0000	-0.1326	-0.0000
0.3536	-0.8176	0.3536	0.1436	0.0000
0.0000	0.3536	0.7071	0.4198	0.0000
0.0000	0.1436	0.3536	-0.9944	0.3536
-0.0000	-0.0663	0.0000	0.4198	0.7071
-0.0000	-0.0331	0.0000	0.1768	0.3536
-0.0000	-0.0000	-0.0000	-0.0663	0.0000
-0.0000	-0.0000	-0.0000	-0.0331	0.0000
0.0000	0.0000	-0.0000	-0.0000	-0.0000
0.0000	0.0000	-0.0000	-0.0000	-0.0000

Last 10 rows. Last 5 columns.

-0.0000	-0.0000	0.0000	0.0000	0.0000
-0.0331	-0.0000	-0.0000	-0.0000	0.0000
-0.0663	-0.0000	-0.0000	-0.0000	0.0000
0.1768	0.0000	-0.0331	-0.0000	-0.0000
0.4198	0.0000	-0.0663	-0.0000	-0.0000
-0.9944	0.3536	0.1768	0.0000	-0.0331
0.4198	0.7071	0.4198	0.0000	-0.0663
0.1768	0.3536	-1.0275	0.3536	0.1768
-0.0663	0.0000	0..3536	0.7071	0.4198
-0.0663	0 .0000	0..3536	0.7071	-0.9944

Decoding: $A = T_S' * P' * B * P * T_S$