Wavelets

Prof. V. M. Gadre
EE Department
IIT Bombay

What are Wavelet Transforms?

- Mathematical Functions
- Much like Fourier Transforms
- Approximating functions contained neatly in finite domain
- Well-suited for approximating data with sharp discontinuities.

Wavelets in Compression

- Original signal represented using coefficients in a linear combination of wavelet functions

- Data operations performed using just the corresponding wavelet coefficients

- Truncating the coefficients below a threshold, helps in representing the data sparsely, thus resulting in “Compression”

Advantages over Fourier Transform (FT)

- FT shows the frequency content of a function but loses the time information
- FT does a very poor job in approximating sharp spikes and discontinuities as compared to WT
- Reason : The basis function used by FT (sine and cosines) are infinitely long

Short Time Fourier Transform (STFT)

- STFT was first developed to have **time localization** and to analyze non-stationary signals

- In STFT analysis, window function is placed first at the beginning of signal and slided along with time to capture different segments of signal at different times

- Hann, Hamming, Kaiser are some examples of windows

- Ideal characteristic of a window is that the local spectral behaviour of the signal must be identified accurately.

STFT Cont'd.

- The STFT is given as

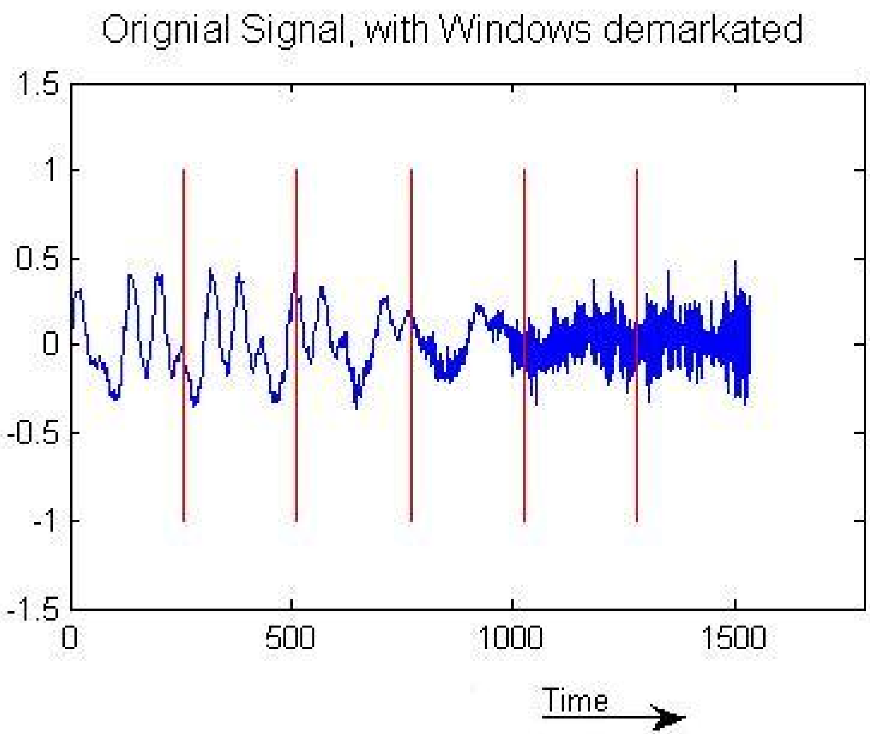
$$X_{STFT}(b,w) = \int_{-\infty}^{+\infty} x(t)\overline{v(t-b)}e^{-jwt}\,dt$$

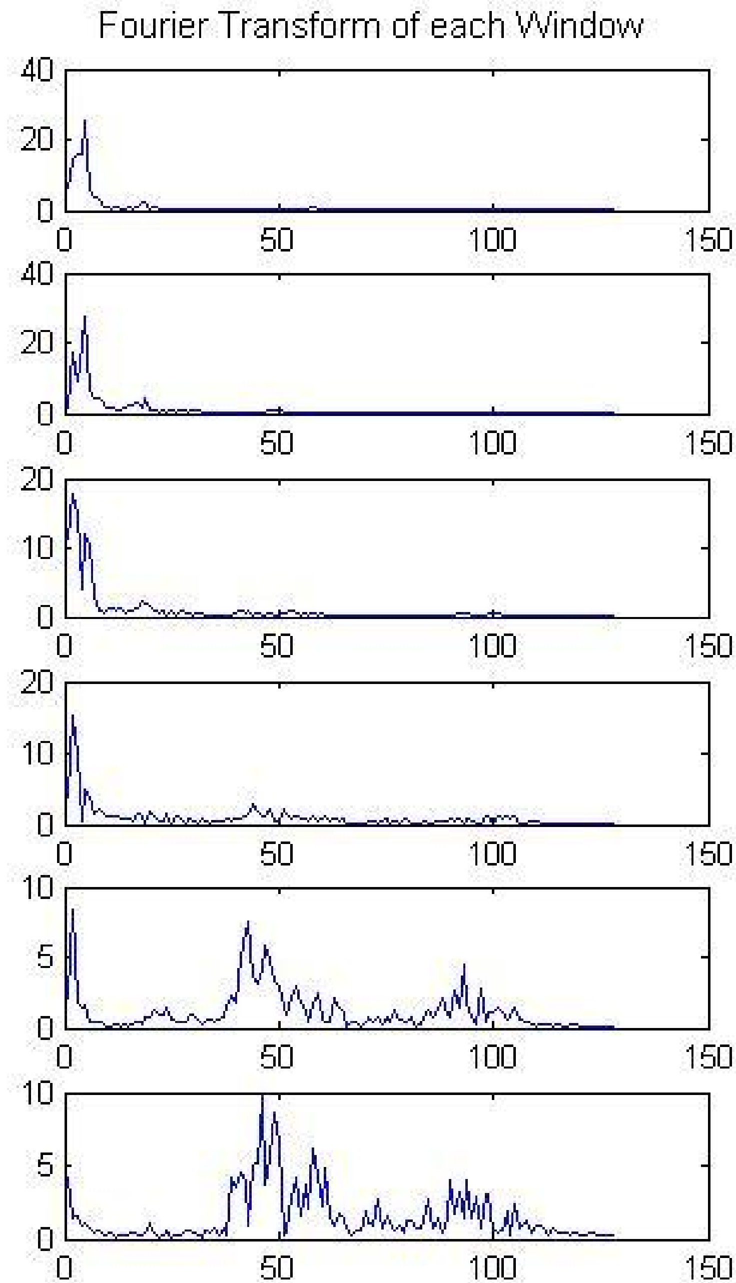
where, V is window function

- The inverse relationship is given as

$$x(t) = A\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} X_{STFT}(b,w)v(t-b)e^{jwt}\,dw\,db$$

Illustration of STFT





History of Wavelets

<u>PRE - 1930's</u>

- First mention : Appendix to thesis of A.Haar(1909)
- 1807- Joseph Fourier, Fourier Synthesis
- Gradually focus shifted from the previous notion of *frequency analysis* to *scale analysis.*
- *Scale analysis* – Less sensitive to noise as it measures average fluctuations of signal at different scales.

History of Wavelets (contd.)

<u>1930's</u>

- Representation of functions using *scale-varying basis functions*
- Paul Levy investigated Brownian Motion using Haar basis function. Found Haar basis function to be better than Fourier basis function

<u>1960 – 80's</u>

- Grossman and Morlet defined wavelets in context of quantum physics.

History of Wavelets (contd.)

Post – 1980’s

- Stephane Mallat(1985) - Gave wavelets an additional jump-start through his work in Digital Signal Processing
- Y.Meyer – Meyer wavelets, continuously differentiable
- Ingrid Daubechies – Perhaps the most elegant set of wavelet orthonormal basis functions

Piecewise Constant Approx.

Piecewise Constant Approximation

- Function takes constant value on small intervals
- Constant Value = Average of the signal over that interval

$$x_T(t) = \frac{1}{T}\int_{(T)} x(t)dt$$

- Information obtained
  - Specific to the resolution
  - Incremental Information

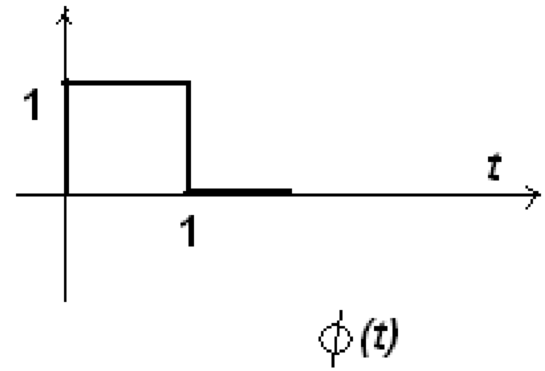
Piecewise Constant Approximation (contd.)

- Incremental Information expressed as multiple of function $\psi(t)$
- Translates and dilates - $\psi(2^m t - nt)$ capture information specific to other resolutions
- Over an interval, a smooth function $x(t)$ =

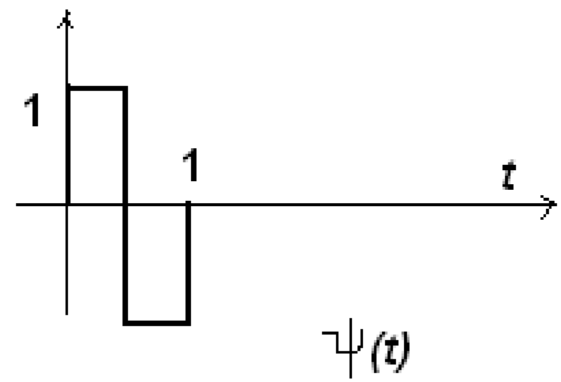
$$x_1(t) + \sum_{m=0}^{\infty} \sum_{n=0}^{2^m-1} c_{m,n} \psi(2^m t - n)$$

The Haar Transform

- Multiresolution analysis
  - Scaling function *ø(t)*



  - Wavelet function *ψ(t)*

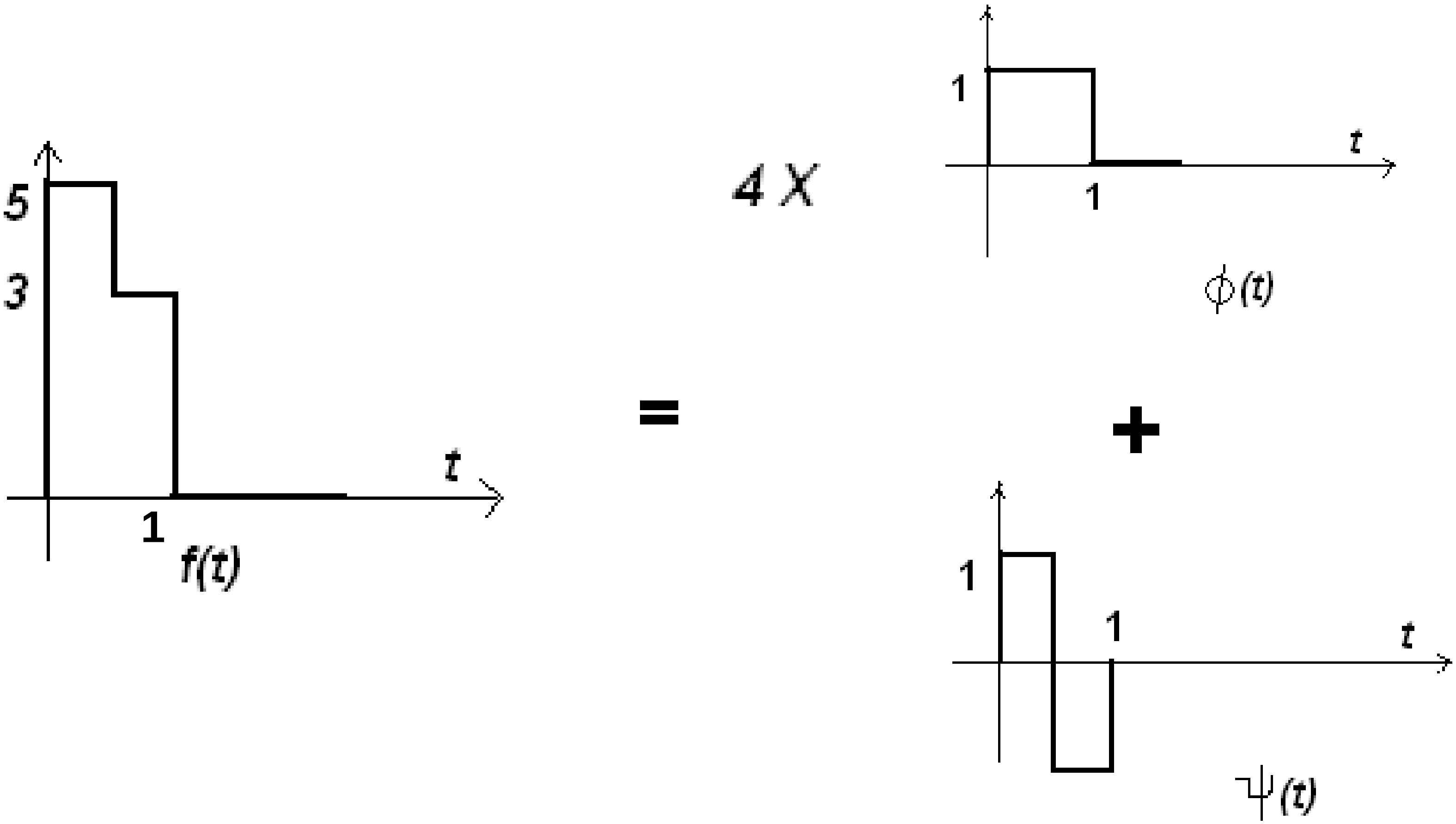


Haar Transform (contd.)

- Haar – simplest orthogonal wavelet system
- Compact Support
- Large Class of signals represented as

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \phi(t-k) + \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} d_{j,k} \psi(2^j t - k)$$

Illustration:



Original Steps (5,3) transformed to (low resolution) **average** 4 and the (high resolution) **detail** 1.

Illustration 2:

- **Input :** An array of 8 pixel values (1D) –

  (1, 2, 3, 4, 5, 6, 7, 8)

- **Level-1:** Averages + Difference (detail)

  (3/2, 7/2, 11/2, 15/2, -1/2, -1/2, -1/2, -1/2)

- **Level-2:**

  (5/2, 13/2, -1, -1, -1/2, -1/2, -1/2, -1/2)

- **Level-3:**

  (9/2, -2, -1, -1, -1/2, -1/2, -1/2, -1/2)

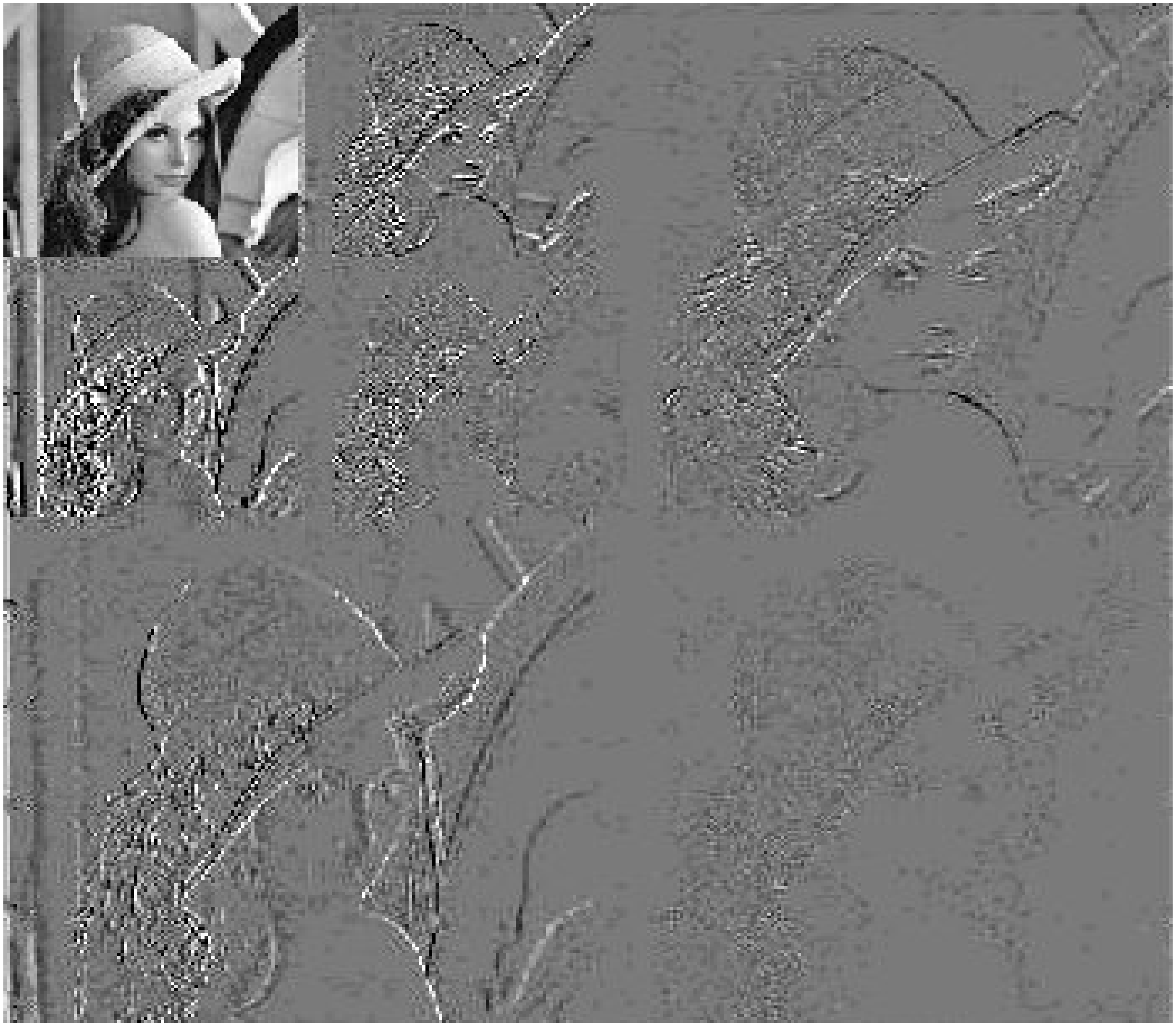
The final array is the **Haar Wavelet transform** of the input data

Haar Transform on Images

12	12	12	12	14	12	12	12
12	12	12	12	14	12	12	12
12	12	12	12	14	12	12	12
12	12	12	12	14	12	12	12
16	16	16	16	14	16	16	16
12	12	12	12	14	12	12	12
12	12	12	12	14	12	12	12
12	12	12	12	14	12	12	12

12	12	13	12	0	0	2	0
12	12	13	12	0	0	2	0
12	12	13	12	0	0	2	0
12	12	13	12	0	0	2	0
16	16	15	16	0	0	-2	0
12	12	13	12	0	0	2	0
12	12	13	12	0	0	2	0
12	12	13	12	0	0	2	0

12	12	13	12	0	0	2	0
12	12	13	12	0	0	2	0
14	14	14	14	0	0	0	0
12	12	13	12	0	0	2	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
4	4	2	4	0	0	-4	0
0	0	0	0	0	0	0	0



Compression applied to “Lena”

Original

70% coefficients

40% coefficients

10% coefficients

Properties – Scaling and Wavelet Function

- $\phi_k(t) = \phi(t-k) \qquad k \in \mathbb{Z} \qquad \phi(t) \in \mathbb{L}^2(\mathbb{R})$
- defined as

$$V_j = \overline{Span_k\{\phi_k(2^j t)\}} = \overline{Span_k\{\phi_{j,k}(t)\}}$$

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k)$$

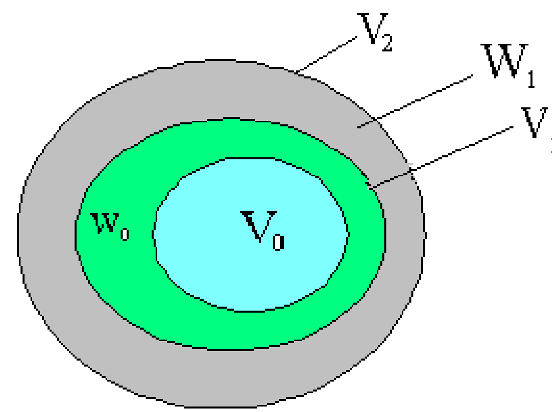
$$V_0 \subset V_1 \subset V_2 \subset \ldots. \subset \mathbb{L}^2(\mathbb{R})$$

- $\langle \phi_{j,k}(t), \psi_{j,l}(t) \rangle = \int \phi_{j,k}(t)\psi_{j,l}(t)dt = 0$
- All members of V_j are orthogonal to members of W_j

Properties – Scaling and Wavelet Function (contd.)

- Wavelet spa $V_1 = V_0 \bigoplus W_0$ o defined as

  whic $V_2 = V_0 \bigoplus W_0 \bigoplus W_1 \bigoplus$



- In general, this gives

$$\mathbb{L}^2(\mathbb{R}) = V_0 \bigoplus W_0 \bigoplus W_1 \bigoplus \ldots$$

Embedded
Zero-Tree Wavelet (EZW)

EZW – A new approach!

- Introduced by Jerome M. Shapiro in 1993
- A very effective image compression algorithm
- Yields a fully Embedded Code

Key Features

- DWT which provides a compact multiresolution representation of the image
- **Zerotree coding** provides compact multiresolution representation of *significance maps.*
- Entropy-coded successive-approximation quantization
- Universal lossless data compression achieved via adaptive arithmetic coding

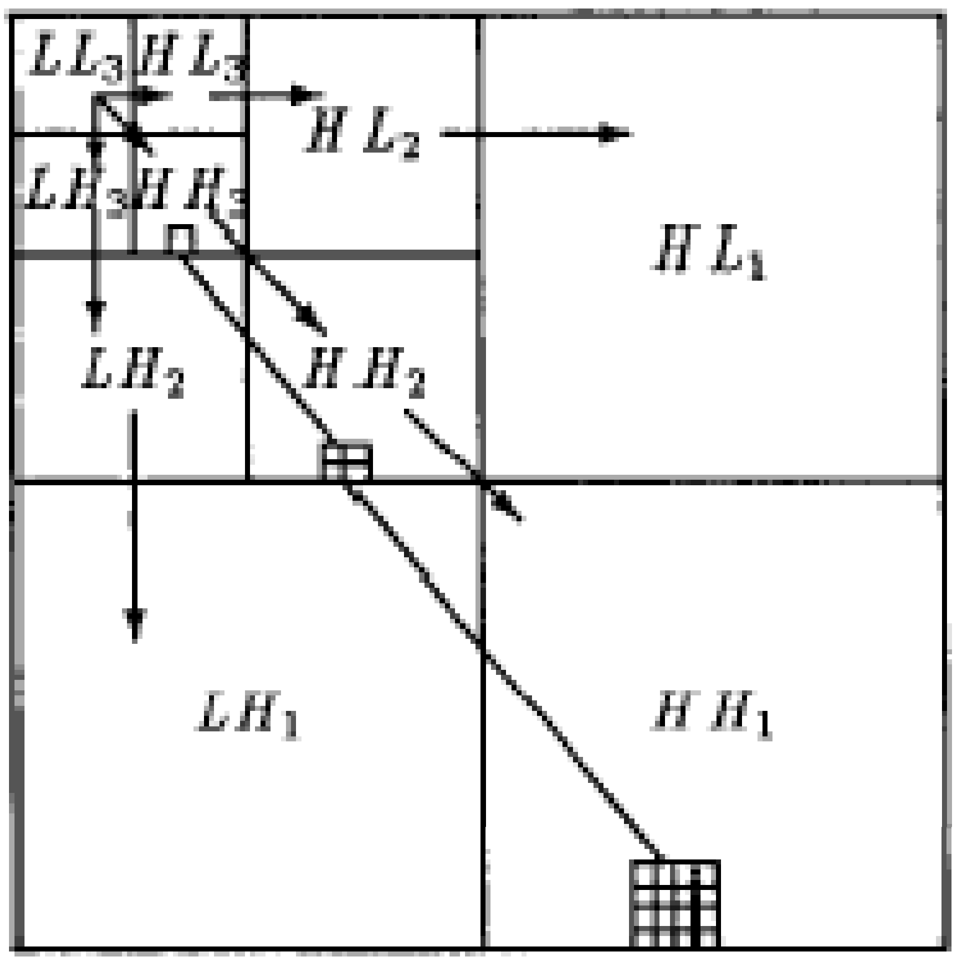
Zerotree Coding Hypothesis

- If a wavelet coefficient is insignificant at a coarse scale with respect to a given threshold T, then **all** wavelet coefficients of the same orientation in the same spatial location at finer scales are insignificant
- More specifically, with the exception of the highest frequency subbands, every coefficient at a given scale can be related to a set of coefficients at the next finer scale of similar orientation.
- Coefficient at coarse scale:***Parent*** .
- Coefficents corresponding to same spatial location at the next finer scale of similar orientation : ***Children***

Parent-Child dependencies of subbands

•With the exception of lowest frequency subband, all parents have 4 children

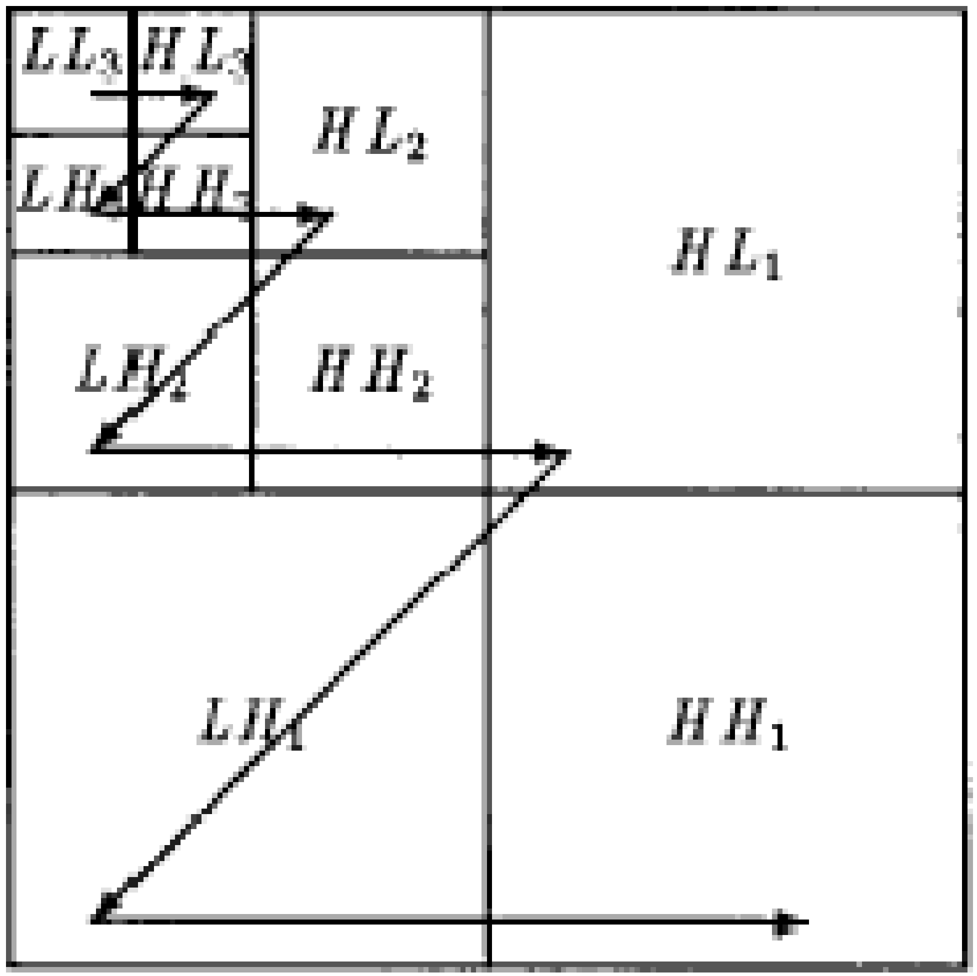
•For lowest frequency subband, Parent-child relationship is defined such that :
Each parent node has three children



Zerotree Coding

- No child is scanned before its parent
- Each coefficient within a given subband is scanned before any coefficient in the next subband

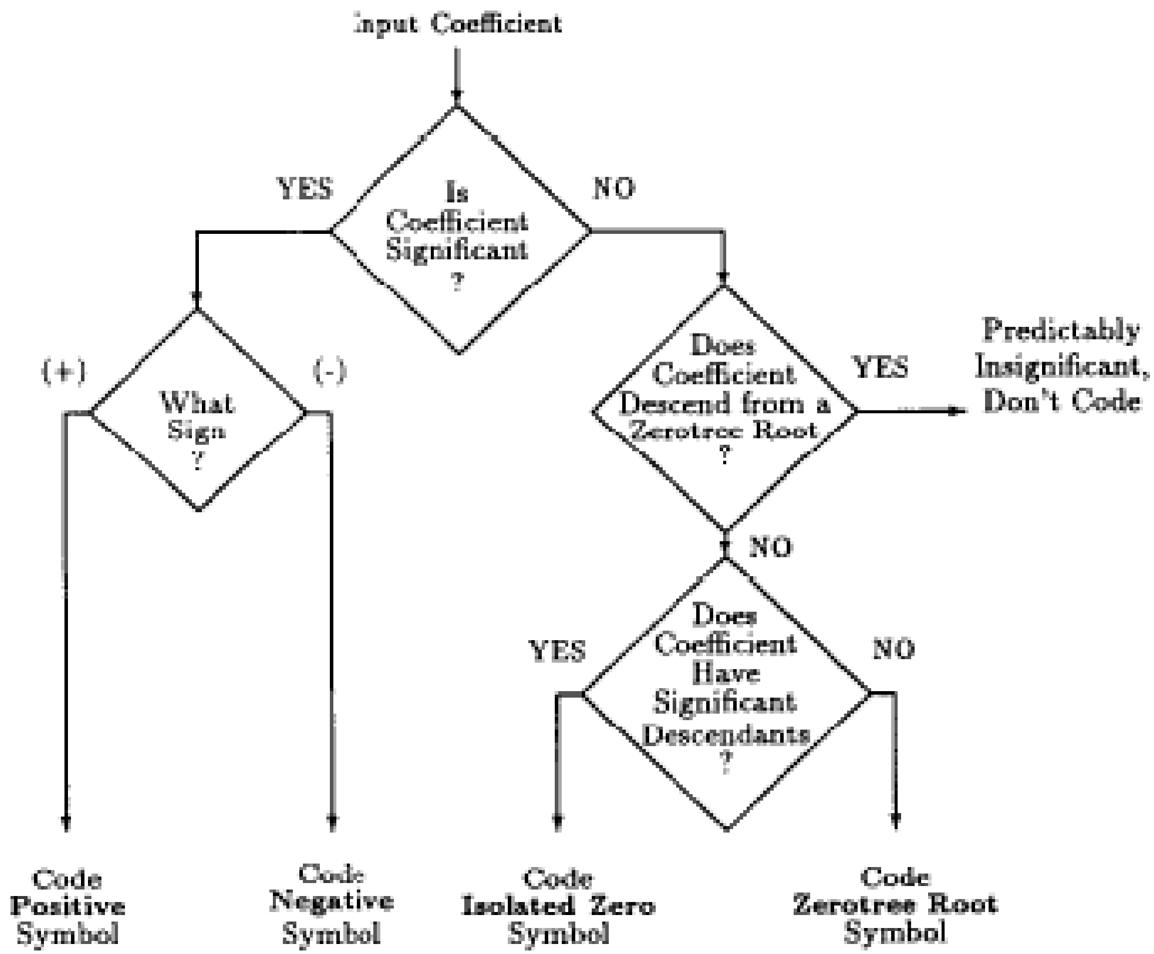
Scanning Order for Encoding



Zerotree Algorithm

- A coefficient x is said to be an element of a *zerotree* for threshold T, if itself and all of its descendants are insignificant with respect to T
- An element of a zerotree is a *zerotree root* if it is not the descendant of a previously found zerotree root

Zerotree Algorithm!



Successive-Approximation Quantization (SAQ)

- SAQ sequentially applies a sequence of thresholds $T_0, T_1, \ldots T_{N-1}$ to determine significance, where thresholds are chosen so that $T_i = T_{i-1}/2$ T_0 is chosen such that $|X_j| < 2T_0$ for all transform coefficients X_j

- **2 separate lists are maintained – *dominant list* and *subordinate list***

- Dominant list: *Coordinates* of those coefficients that have not yet been found to be significant in the same relative order as the initial scan

- Subordinate List : *Magnitudes* of those coefficients that have been found to be significant

SAQ (Contd..)

- For each threshold, each list is scanned once
- During a dominant pass, coefficients are compared to the threshold to determine their significance. The significance map is then zerotree coded
- Each time a coefficient is encoded as significant, its magnitude is appended to the subordinate list and coefficient in wavelet transform array is set to zero
- Dominant pass followed by subordinate pass where magnitudes are refined to an additional bit of precision

Illustration

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

Illustration (Contd..)

- The largest coefficient magnitude is 63, therefore we chose our threshold $T_0 = 32$
- First Dominant pass:
  - First coefficient has magnitude 63 which is greater than threshold, and is positive so a positive symbol is generated
  - Even though the coefficient 31 is insignificant with respect to the threshold 32, it has a significant threshold two generations down in subband LH1with magnitude 47. Thus, symbol for an isolated zero is generated
  - The magnitude 23 is less than 32 and all descendants are insignificant. A zerotree symbol is generated, and no symbol will be generated for any descendant coefficient during current dominant pass. *AND so on…*
- No symbols were generated from subband HH2 which would ordinarily precede HL1 in the scan. Also, since HL1 has no descendants, entropy coding can resume using a 3-symbol alphabet where IZ and ZTR symbols are merged into the Z (zero) symbol

Illustration (Contd...)

Processing of First Dominant Pass

Comment	Subband	Coefficient Value	Symbol	Reconstruction Value
(1)	LL3	63	POS	48
	HL3	-34	NEG	-48
(2)	LH3	-31	IZ	0
(3)	HH3	23	ZTR	0
	HL2	49	POS	48
(4)	HL2	10	ZTR	0
	HL2	14	ZTR	0
	HL2	-13	ZTR	0
	LH2	15	ZTR	0
(5)	LH2	14	IZ	0
	LH2	-9	ZTR	0
	LH2	-7	ZTR	0
(6)	HL1	7	Z	0
	HL1	13	Z	0
	HL1	3	Z	0
	HL1	4	Z	0
	LH1	-1	Z	0
(7)	LH1	47	POS	48
	LH1	-3	Z	0
	LH1	-2	Z	0

Illustration (Contd...)

- During the first dominant pass, which used a threshold of 32, four significant coefficients were identified.
- First subordinate pass will **refine** these magnitudes and identify them as being in the interval [32,48) which will be encoded with symbol "0" or in the interval [48,64) which will be encoded with the symbol "1"
- After the completion of the subordinate pass, the magnitudes on the subordinate list are sorted in decreasing magnitude

Coefficient Magnitude	Symbol	Reconstruction Magnitude
63	1	56
34	0	40
49	1	56
47	0	40

Illustration (Contd...)

- The process continues to the second dominant pass with a new threshold of 16
- Only those coefficients not yet found to be significant are scanned. Previously significant are treated as zero
- The process continues alternating between dominant and subordinate passes and can stop at any time

Set Partitioning in Hierarchical Trees (SPIHT)

SPIHT?

- Introduced by Amir Said and William Pearlman in 1996

- An extension of EZW

- Provides better performance than EZW

- **Reference :**

  A Said and W Pearlman, "**A New, Fast, and Efficient Image Codec based on Set Partitioning in Hierarchical Trees**", *IEEE Trans. on Circuits and Systems for Video Technology*, vol.6, no.3, June 1996

Features of SPIHT

- Ordering data is not explicitly transmitted

- Execution path of any algo is defined by the results of the comparisons on its branching points. Therefore, if encoder and decoder have same **sorting algorithm**, then the decoder can duplicate encoder's execution path if it receives the results of magnitude comparisons

Set Partitioning Sorting Algorithm

- Sorting algo divides the set of pixels into partitioning subsets T_m and performs the magnitude test

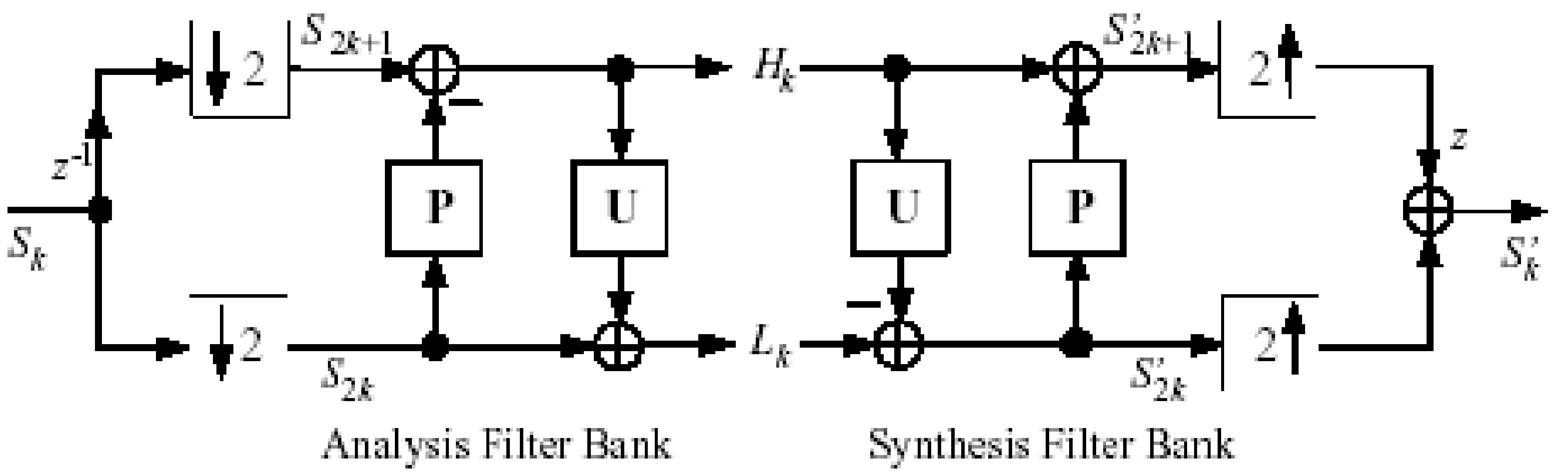
$$\max_{(i,j)\in T_m} \{|c_{i,j}|\} \geq 2^n$$

- “No”: All coefficients in T_m are insignificant
- “Yes”: A certain rule shared by the encoder and the decoder is used to partition T_m into new subsets $T_{m,l}$ Significance test is then applied to the new subsets

- This set division process continues until the magnitude test is done to all single coordinate significant subsets in order to identify each significant coefficient

Motion Compensated Temporal Filtering (MCTF)

MCTF – based Video Coding



Obtaining the high-pass (prediction residual) pictures

$$H_k = S_{2k+1} - P(S_{2k})$$

Obtaining the low-pass pictures

$$L_k = S_{2k} + U(S_{2k+1} - P(S_{2k})) = \tfrac{1}{2} S_{2k} + U(S_{2k+1})$$

where, $U(P(s)) = s/2$

Recent Advances in MCTF - based Video Coding

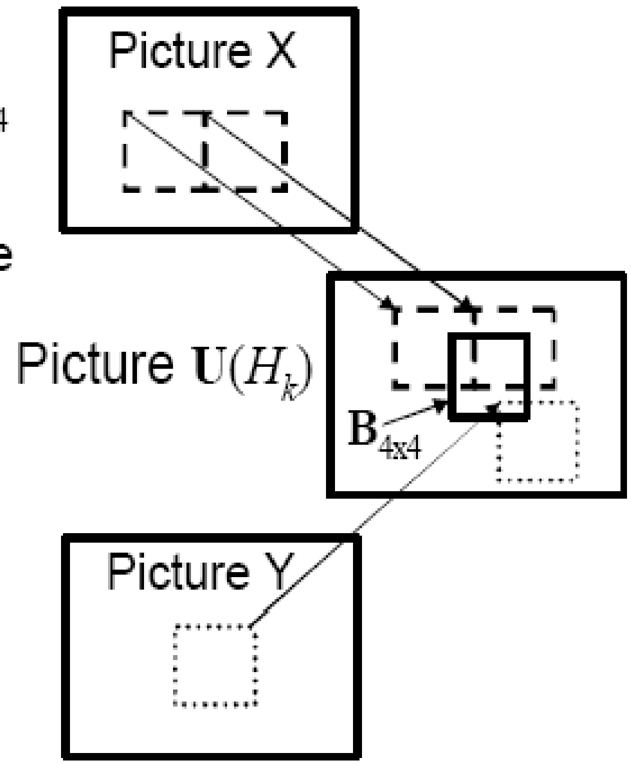
- Incorporation of motion compensation into the lifting steps of a temporal filter bank
- Lifting scheme is invertible
- Motion compensation with any motion model possible to incorporate
  - Variable block-size motion compensation
  - Sub-sample accurate motion vectors

MCTF Extension to H.264/AVC

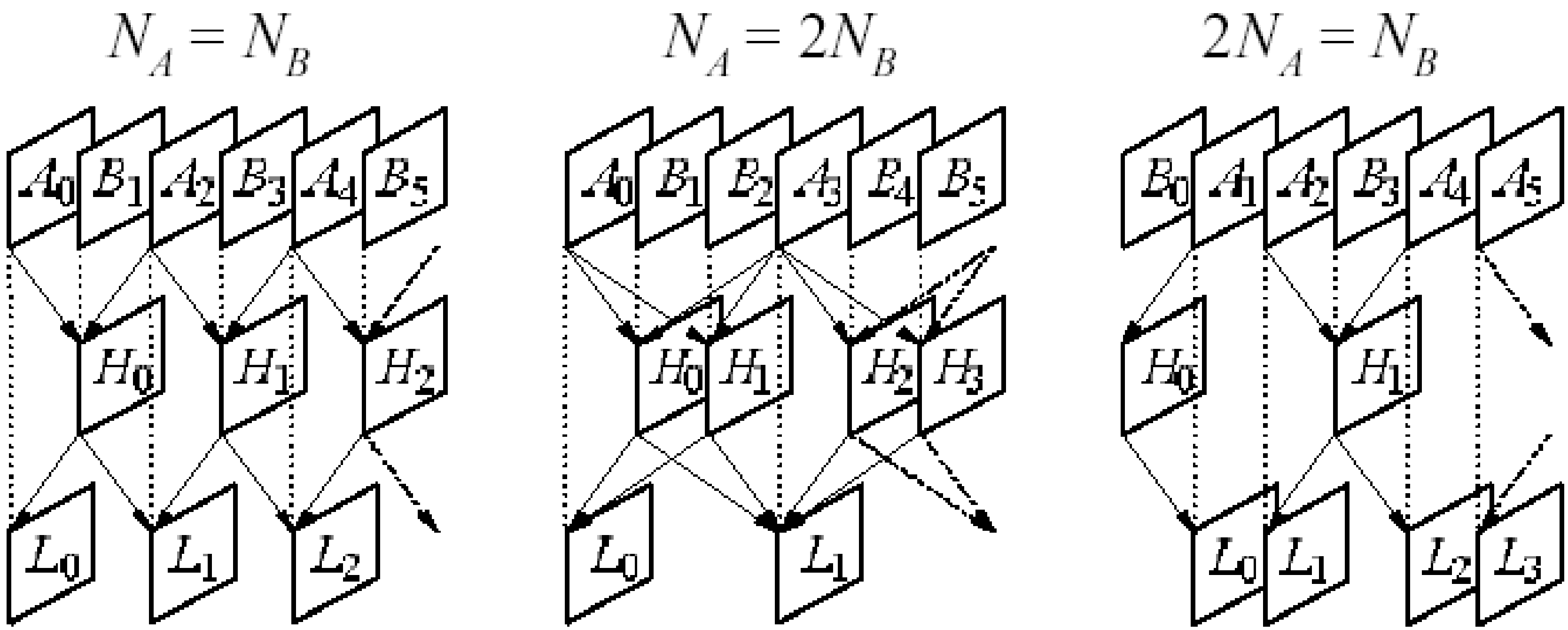
- Highly efficient motion model of H.264/AVC
- Lifting steps are similar to motion compensation in B slices
- Block-based residual coding
- Open-loop structure of the analysis filter bank offers the possibility to efficiently incorporate scalability
- Incorporation of multiple reference picture capabilities makes it similar to H.264/AVC
- Update Operators are derived at the decoder enabling B-frame like representation

Derivation of Update Operators

- For each 4x4 luma block $\mathbf{B}_{4x4}$ in the picture $\mathbf{U}(H_k)$, derive $\mathbf{m}_{U0}$, r_{U0}, $\mathbf{m}_{U1}$, and r_{U1} as follows
  1. Evaluate all $\mathbf{m}_{P0}$ and $\mathbf{m}_{P1}$ that point into $\mathbf{B}_{4x4}$
  2. Select those $\mathbf{m}_{P0}$ and $\mathbf{m}_{P1}$ that use maximum number of samples for reference out of $\mathbf{B}_{4x4}$
  3. Set $\mathbf{m}_{U0} = -\mathbf{m}_{P0}$ and $\mathbf{m}_{U1} = -\mathbf{m}_{P1}$
  4. Set r_{U0} and r_{U1} to point to those pictures into which MC is conducted using $\mathbf{m}_{P0}$ and $\mathbf{m}_{P1}$, respectively
  5. Harmonize derived $\mathbf{m}_{U0}$, r_{U0}, $\mathbf{m}_{U1}$, and r_{U1} with H.264/MPEG-4 AVC syntax



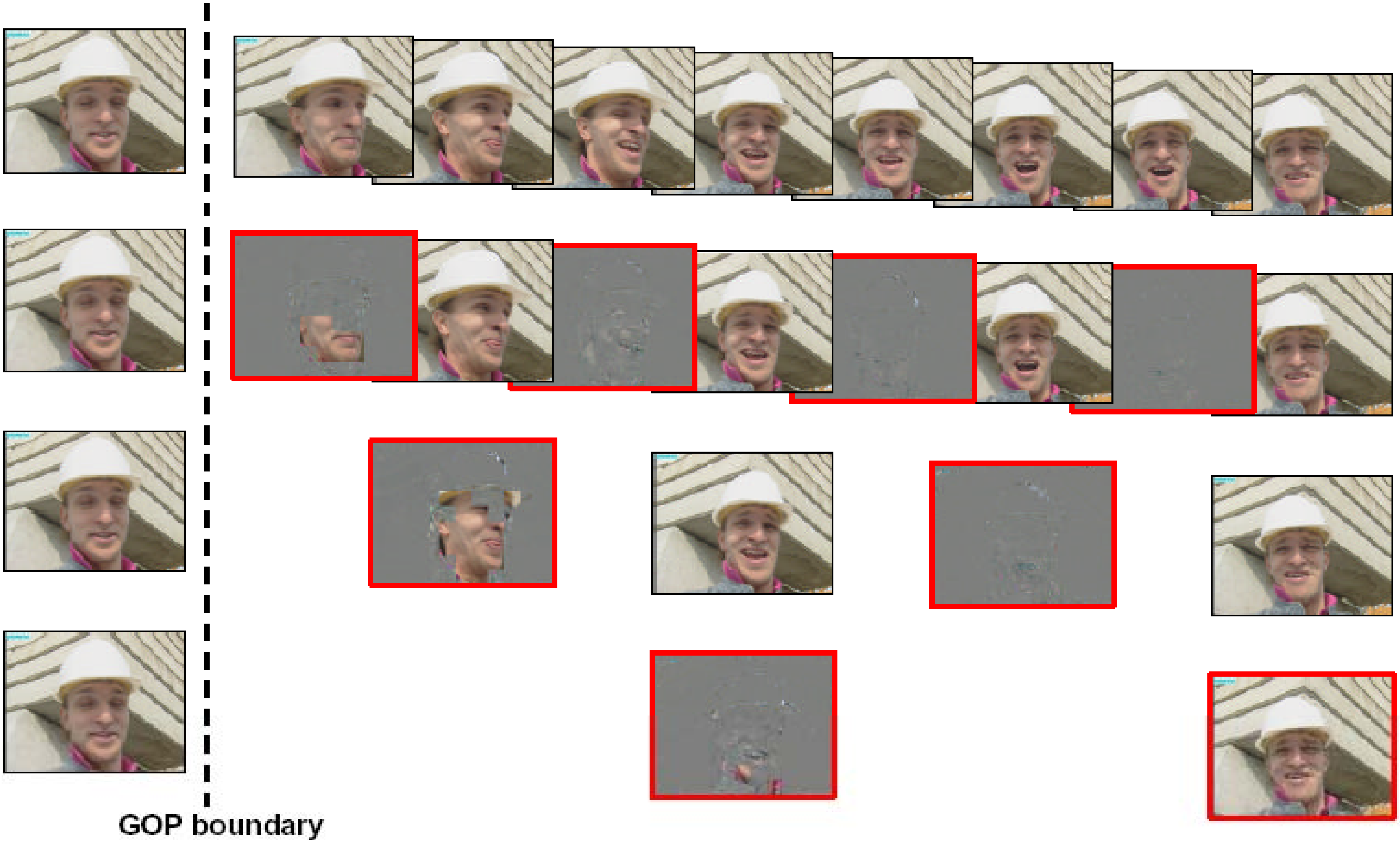
Temporal Coding Structure



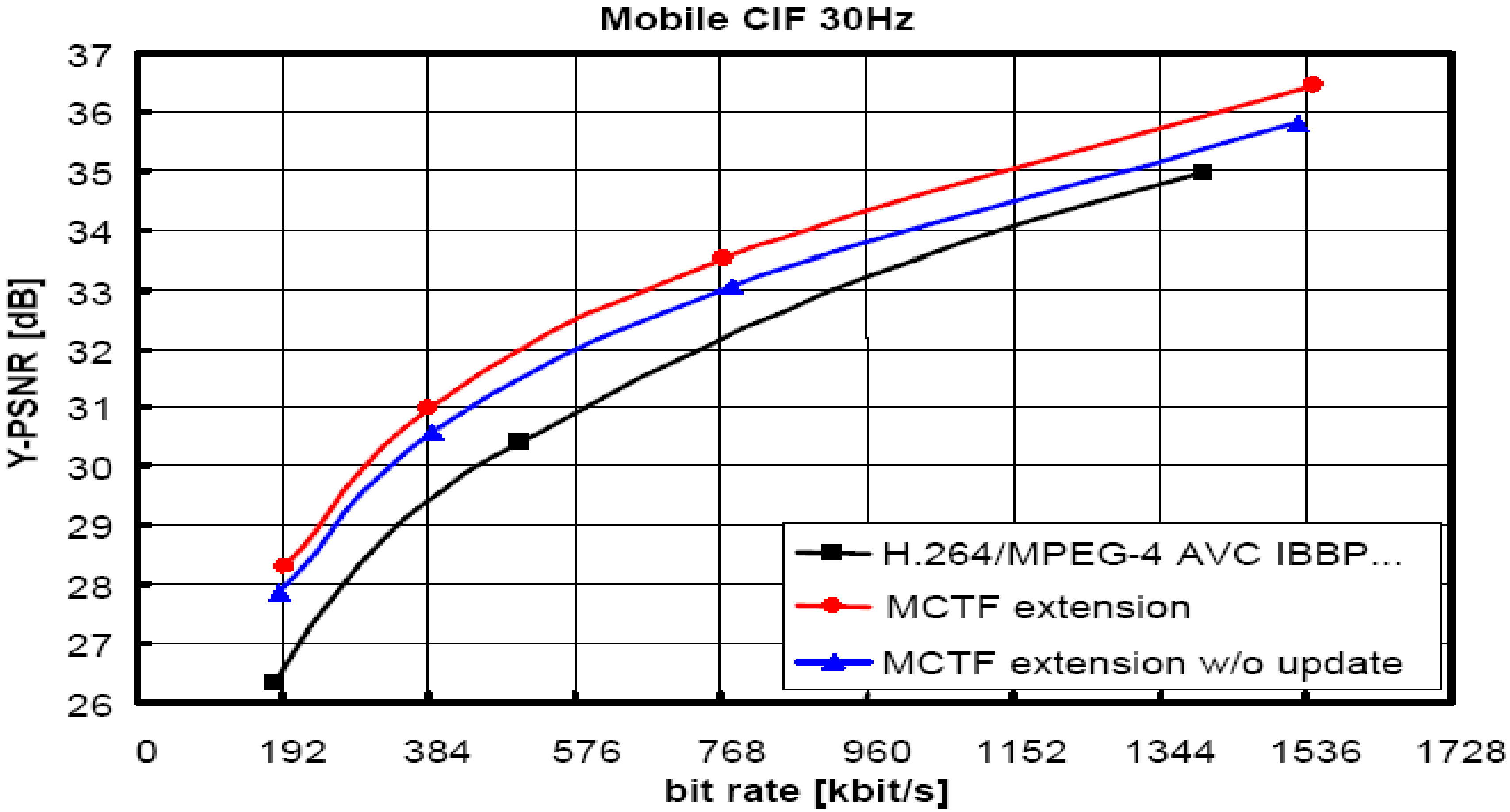
Temporal Coding Structure (Contd..)

- Group of N_0 input pictures partitioned into two sets:
  - Set 1: N_A ($0 < N_\mathrm{A} < N_0$), Set 2: $N_B = N_0 - N_A$
- Set 1: pictures A_K, Set 2: pictures B_K
- Pictures H_K are spatially shift-aligned with pictures B_K
- Pictures L_K are spatially shift-aligned with pictures A_K

Temporal Decomposition Structure



Results for the MCTF Extension



Courtesy : HHI

Lattice Structure for Perfect Reconstruction Filter Bank
Polyphase components and Noble identities

Noble Identities

Noble identity for down sampling

Noble identity for up sampling

Polyphase Components

- Signal x[n] can be separated into different phases
  - For general M, there are M-phase
    - k^{th} phase : x[Mn+k]
    - K = 0, 1, 2 … M-1
  - For M = 2, we can divide x[n] into:
    - Even phase: {…, x[-2], x[0], x[2], …}
    - Odd phase: {…, x[-1], x[1], x[3], …}
- $(\downarrow M)x[n]$: only the 0-phase, {x[Mn]} survives

Polyphase Components (cont'd.)

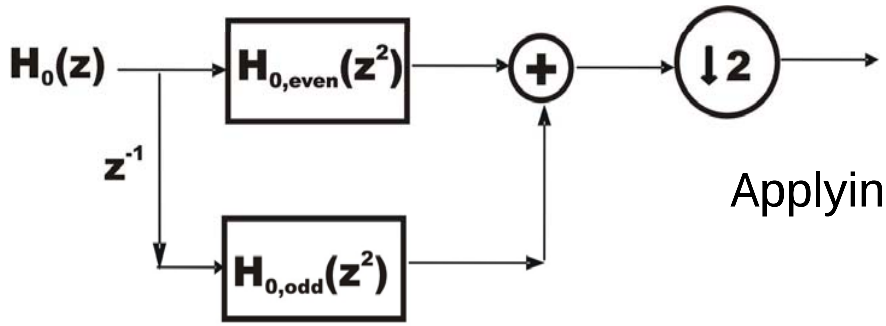
$$B(z) = \sum_{n} b[n] z^{-n}$$

$$B(z) = \sum_{k} b[2k] z^{-2k} + \sum_{k} b[2k+1] z^{-(2k+1)}$$

$$B(z) = B_0(z^2) + z^{-1}B_1(z^2)$$

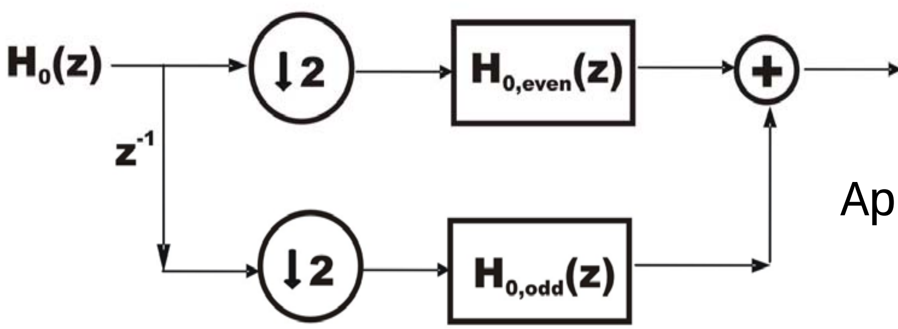
$B_0(z^2)$ and $z^{-1}B_1(z^2)$ are the polyphase components of b[n]

Application of Polyphase components and noble identities to PRFB



Applying polyphase components

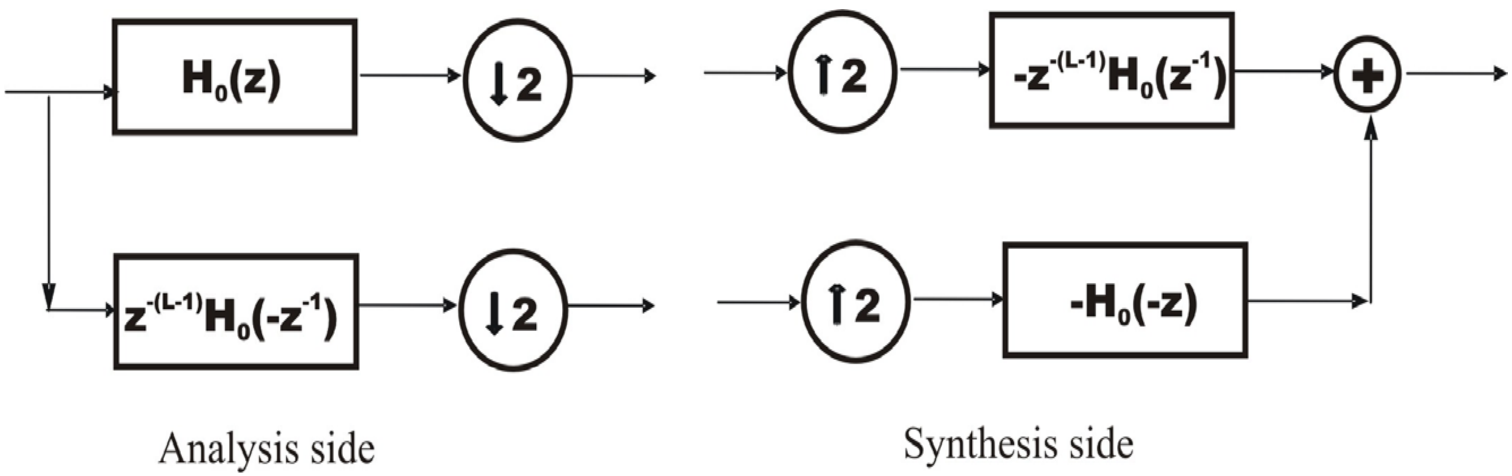
(a)



Applying noble identities

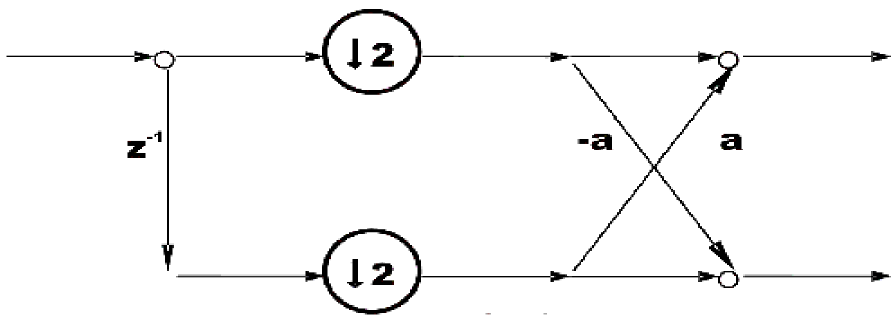
(b)

Conjugate Quadratic Perfect Reconstruction Filter Bank



$$H_0(z) = 1 + \alpha z^{-1} \text{ and } \mathrm{L} = 2$$
$$G_0(z) = z^{-(L-1)} H_0(-z^{-1}) = z^{-1} - \alpha$$

Lattice Structure



Analysis side

On the Synthesis side, we reverse arrows and replace the downsampler by an upsampler with the same factor. This is called **transposition**.

Advantages of Lattice Structure

- Modular Realization
  - Hardware implementation easier
- Amenable to optimization
  - the noble identities can be easily applied

Wave-packet transform

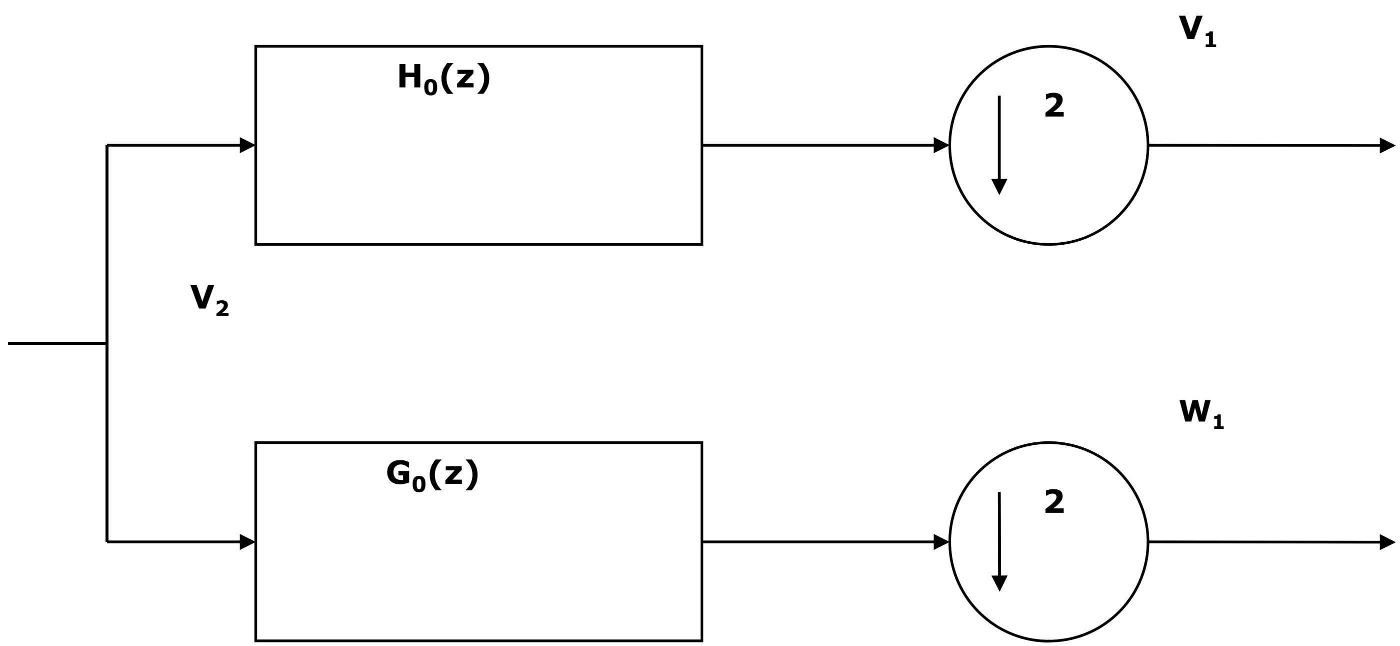
Multi-resolution Analysis

- $\mathbf{L^2(R)}$ denotes the set of square integrable functions.
- **MRA** of $\mathbf{L^2(R)}$ is ladder of subspaces,

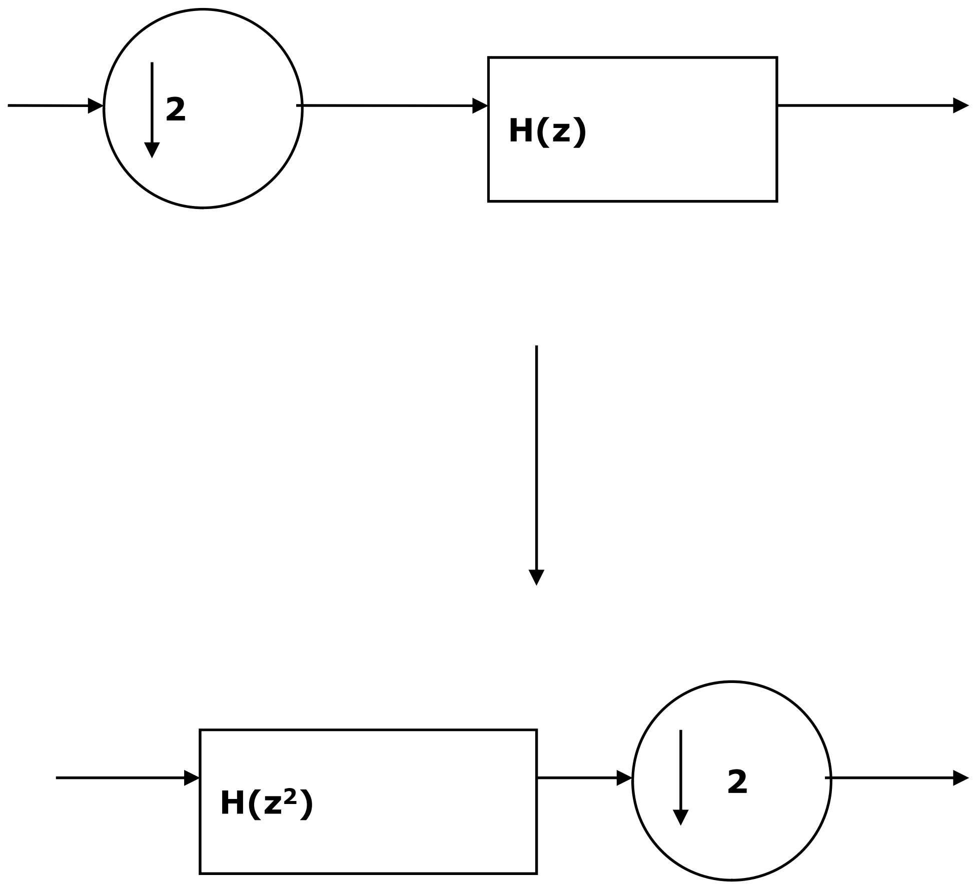
$$\ldots \subset \mathbf{V_{-1}} \subset \mathbf{V_0} \subset \mathbf{V_1} \subset \mathbf{V_2} \ldots\ldots\ldots\ldots\ldots$$

- In general, $\mathbf{V_{k+1} = V_k + W_k}$
- We are interested in splitting W_k

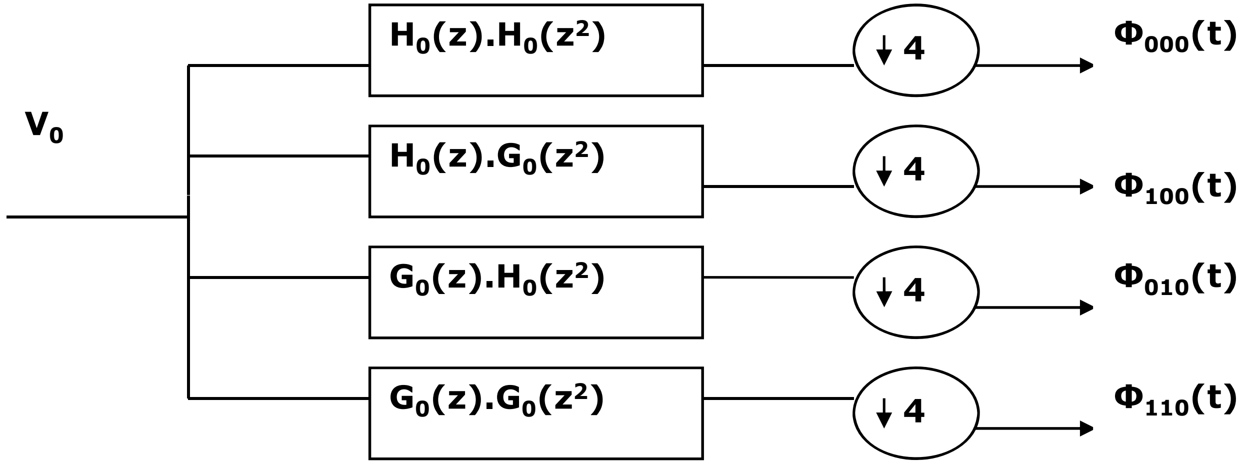
Filter Banks



Noble Identity



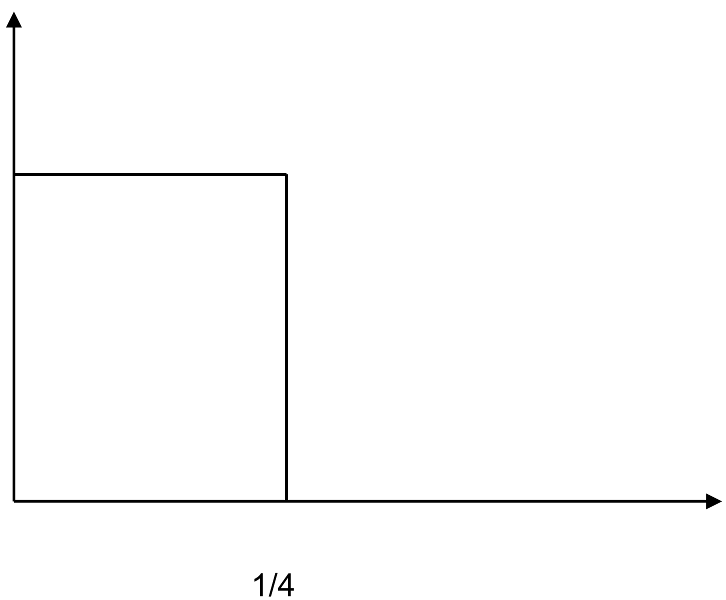
Wave-packet decomposition



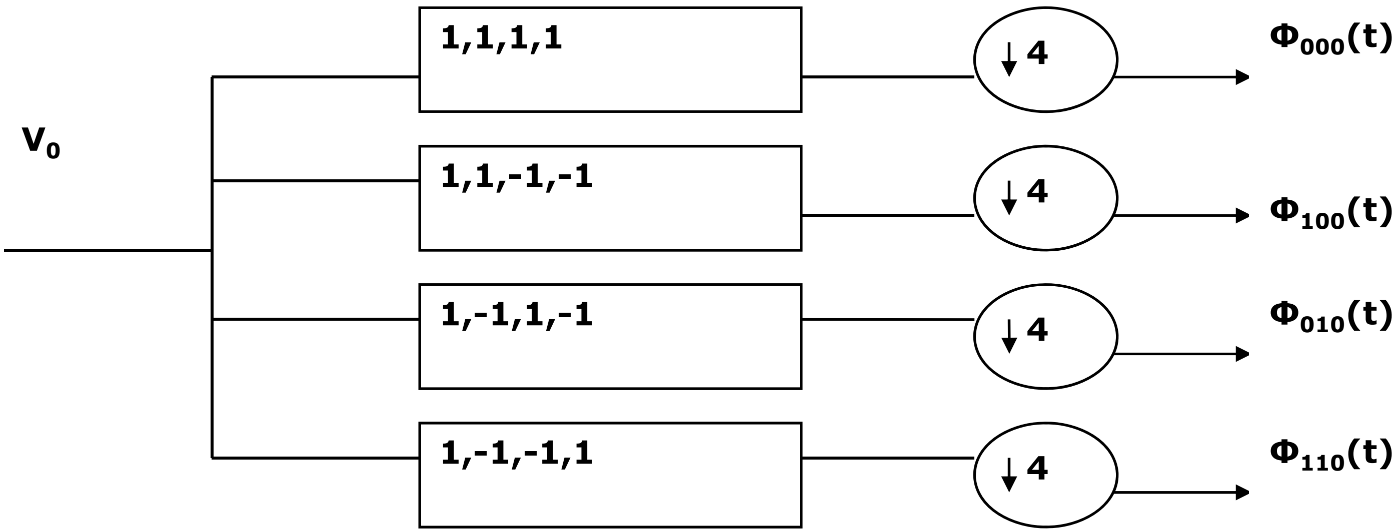
Example using Haar wavelet

- $H_0(z^2) = 1+z^2$ and $G_0(z^2) = 1-z^2$

$\Phi(4t)$



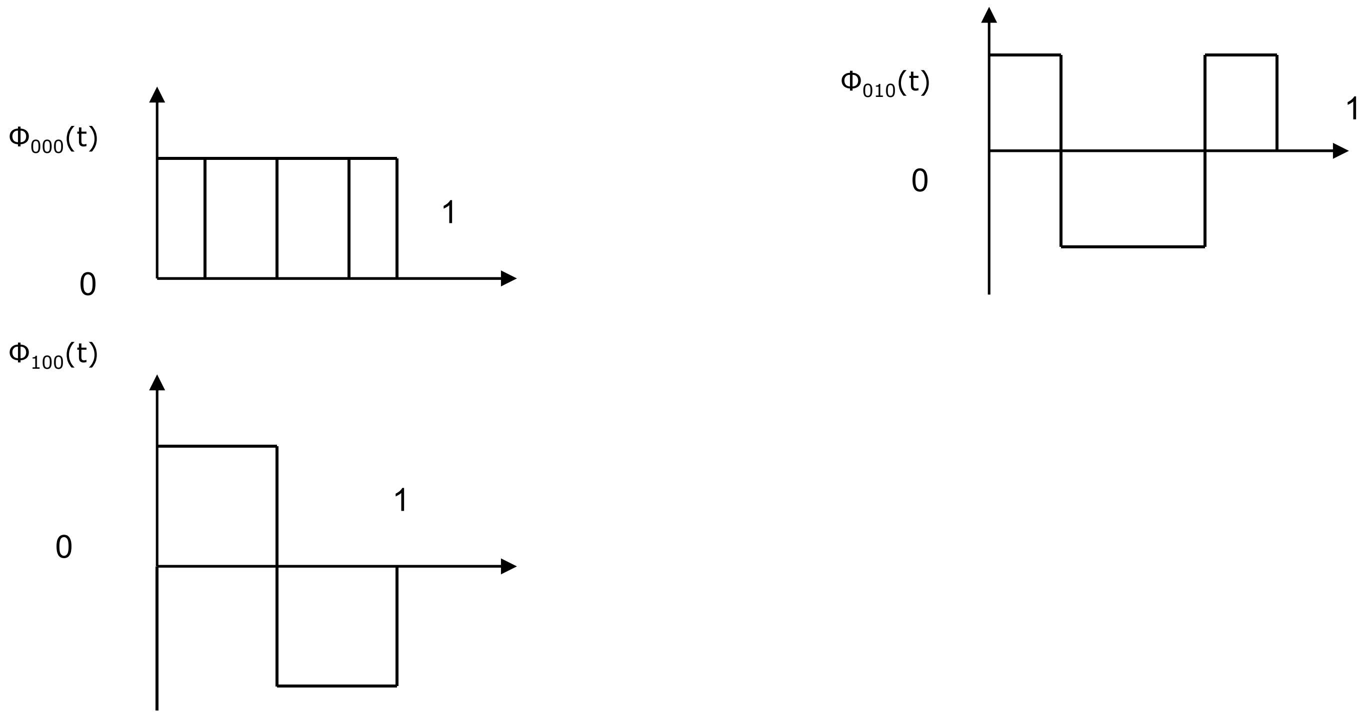
Example cont'd



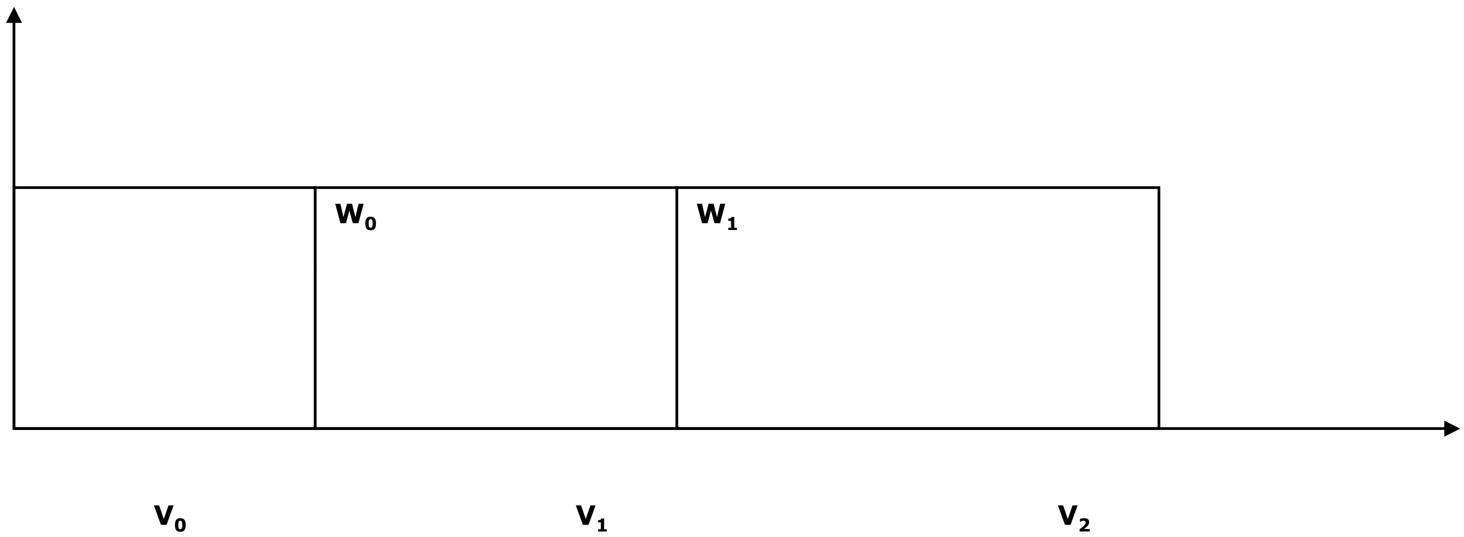
Example cont'd

- $\Phi_{000}(t) = \Phi(4t) + \Phi(4t-1) + \Phi(4t-2) + \Phi(4t-3)$
- $\Phi_{100}(t) = \Phi(4t) + \Phi(4t-1) - \Phi(4t-2) - \Phi(4t-3)$ and so on.
- All these are wave-packets generated out of filter banks.

Plots of wave-packets



Spectral domain representation



Lifting Scheme

The Lifting Scheme

- Lifting based scheme requires far fewer computations and reduced memory for the DWT.
- Hence, it is faster, consume less power and occupy smaller area.
- Lifting based scheme has an additional advantage of in-place computation

The Basic Idea Behind Lifting

A canonical case of lifting consists of three stages, which we refer to as: *split, predict, and update*.

We start with an abstract data set λ_0

• In the first stage we **<u>split</u>** the data into two smaller subsets λ_{-1} and γ_{-1}.

•In the second stage, we use λ_{-1} subset to **<u>predict</u>** the γ_{-1} subset based on the correlation present in the original data. If we can find a prediction operator *P*, independent of the data, $\gamma_{-1} = P(\lambda_{-1})$.

•$P(\lambda_{-1})$ is likely to be close to γ_{-1}. Therefore, we might want to replace γ_{-1} with the difference between itself and its predicted value $P(\lambda_{-1})$. This difference will contain much less information than original γ_{-1} set.

The Basic Idea Behind Lifting (contd..)

- We denote this abstract difference operator with a $-$ sign and thus get

$$\gamma_{-1} := \gamma_{-1} - P\,(\lambda_{-1}).$$

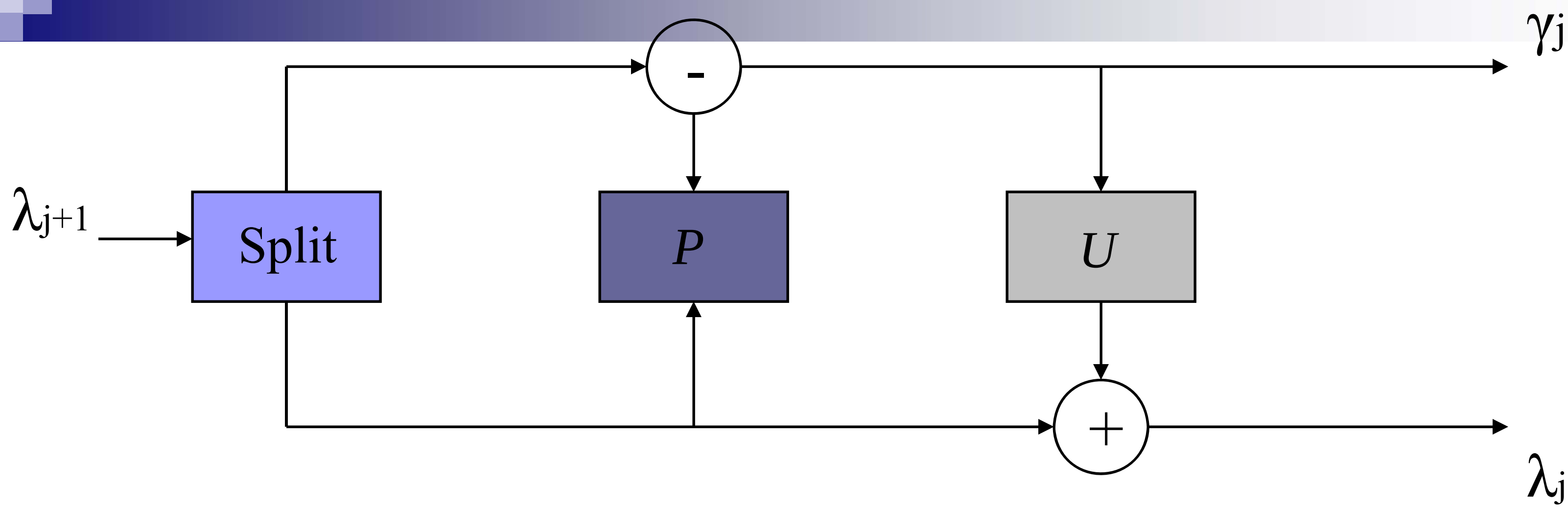
The wavelet subset now encodes how much the data deviates from the model on which *P* was built.

- To find a better λ_{-1} so that a certain scalar quantity Q ( ), for e.g. the mean, is preserved, or

$$Q\,(\lambda_{-1}) = Q\,(\lambda_0).$$

Therefore the already computed wavelet set γ_{-1} is used to update λ_{-1} so that the later preserves Q ( ). In other words, an operator U is used to update λ_{-1} as

$$\lambda_{-1} := \lambda_{-1} + U\,(\gamma_{-1}).$$



This leads to the following wavelet transform algorithm:

$$\{\lambda_j, \gamma_j\} := \text{Split}(\lambda_{j+1})$$

For j = -1 downto –n:

$$\gamma_j \mathrel{-}= P(\lambda_j)$$

$$\lambda_j \mathrel{+}= U(\gamma_j)$$

Once we have the forward transform, we can immediately derive the inverse. The only thing to do is to reverse the operations and toggle + and -. This leads to the following algorithm for the inverse wavelet transform:

$$\lambda_j \mathrel{-}= U(\gamma_j)$$

For j = -n to –1:

$$\gamma_j \mathrel{+}= P(\lambda_j)$$

$$\lambda_{j+1} := \text{Join}(\lambda_j, \gamma_j).$$

Lifting Algorithm

- Let h(z) and g(z) be the lowpass and highpass analysis filters

$$h(z)=h_e(z^2)+z^{-1}h_o(z^2) \qquad \tilde{h}(z)=\tilde{h}_e(z^2)+z^{-1}\tilde{h}_o(z^2)$$
$$g(z)=g_e(z^2)+z^{-1}g_o(z^2) \qquad \tilde{g}(z)=\tilde{g}_e(z^2)+z^{-1}\tilde{g}_o(z^2)$$

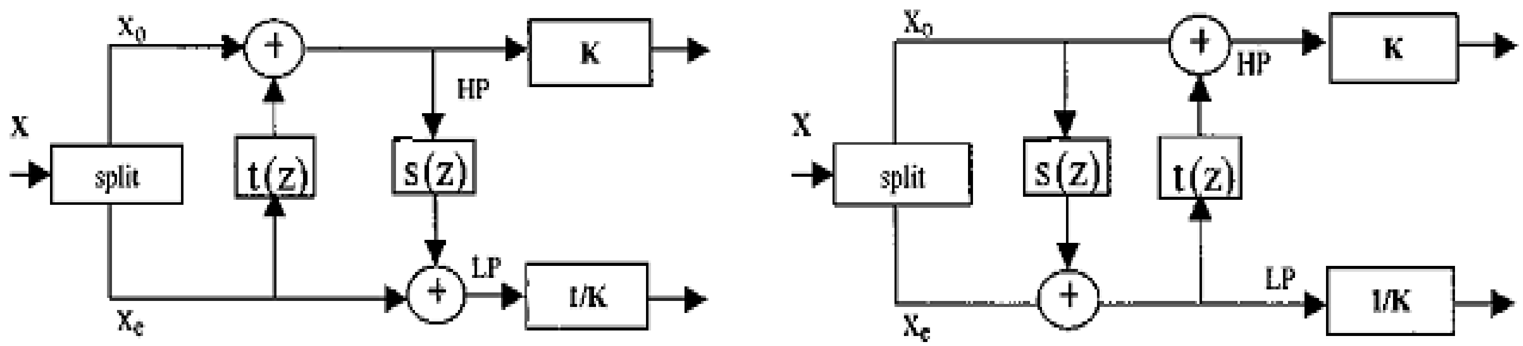
- The corresponding polyphase matrices are defined as

$$P(z)=\begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \qquad \tilde{P}(z)=\begin{bmatrix} \tilde{h}_e(z) & \tilde{h}_o(z) \\ \tilde{g}_e(z) & \tilde{g}_o(z) \end{bmatrix}$$

- If (h,g) is a complementary filter pair, then P(z) can be factored as

$$P_1(z)=\begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}\prod_{i=1}^{m}\begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \text{or } P_2(z)=\begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}\prod_{i=1}^{m}\begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix}\begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix}$$

- The two types of lifting scheme are shown in figure below

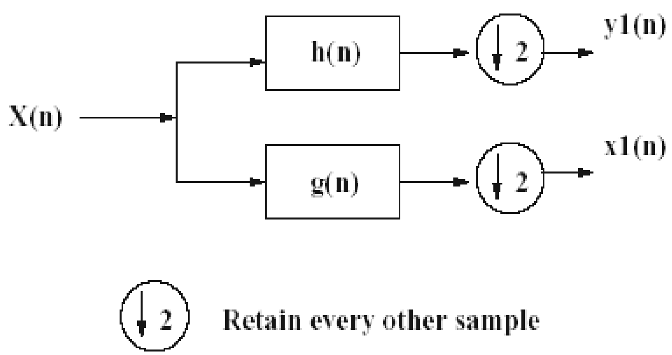


- Split: The entries are stored in even and odd entries.
- Predict Step: Even samples are multiplied by the time domain equivalent of t(z) and are added to odd samples.
- Update Step: Updated odd samples are multiplied by the time domain equivalent of s(z) and added to even samples.
- Scaling: Even samples multiplied by 1/K and odd by K.

Illustration using DWT Implementation

- Two schemes of Implementation

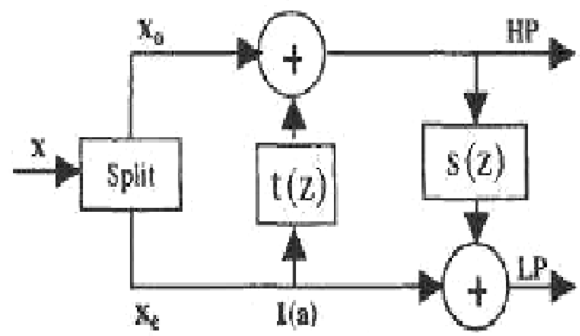
-- Convolution



$$
\begin{aligned}
y[2n] &= a_{aL}\cdot x[2n]+b_{aL}\cdot\{x[2n+1]+x[2n-1]\}+\\
&\quad c_{aL}\cdot\{x[2n+2]+x[2n-2]\}+d_{aL}\cdot\{x[2n-3]+x[2n+3]\}+\\
&\quad e_{aL}\cdot\{x[2n+2]+x[2n-2]\}\\
y[2n+1] &= a_{aH}\cdot x[2n+1]+b_{aH}\cdot\{x[2n]+x[2n+2]\}+\\
&\quad c_{aH}\cdot\{x[2n-1]+x[2n+3]\}d_{aH}\cdot\{x[2n-2]+x[2n+4]\}
\end{aligned}
$$

Convolution Equation

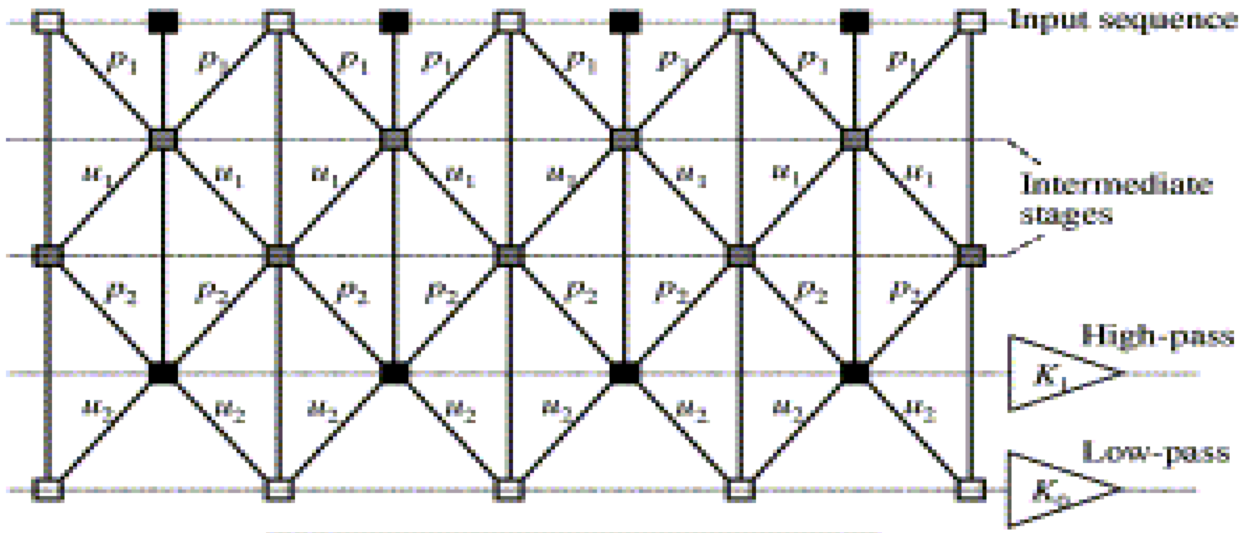
--Lifting



$$
\begin{aligned}
u[2n+1] &= x[2n+1]+P_1\cdot\{x[2n]+x[2n+2]\}\\
u[2n] &= x[2n]+U_1\cdot\{u[2n+1]+u[2n+3]\}\\
y[2n+1] &= u[2n+1]+P_2\cdot\{u[2n]+u[2n+2]\}\\
y[2n] &= u[2n]+U_2\cdot\{y[2n+1]+y[2n+3]\}\\
z[2n+1] &= -K\cdot y[2n+1]\\
z[2n] &= K^{-1}\cdot y[2n+1]
\end{aligned}
$$

Lifting Equations For Daubechies (9, 7) Filter

Lifting Scheme for DWT using Daubechies (9, 7) Filters



p_1	- 1.586134342059924
u_1	- 0.052980118572961
p_2	+0.882911075530934
u_2	+0.443506852043971
$K_1 = 1/K_0$	+1.230174104914001

An overview of the JPEG 2000 still image compression standard
Majid Rabbani, and Rajan Joshi

JPEG2000 filters

Analysis Filter Coefficients		
n	Low-Pass Filter h(n)	High-Pass Filter g(n)
0	6/8	1
±1	2/8	-1/2
±2	-1/8	
Synthesis Filter Coefficients		
n	Low-Pass Filter h~(n)	High-Pass Filter g~ (n)
0	1	6/8
±1	1/2	-2/8
±2		-1/8

Integer Coefficients of Lossless (5,3) Le Gall filter

Analysis filter coefficients		
n	Low-Pass Filter h (n)	High-Pass Filter g (n)
0	0.6029490182363579	1.115087052456994
±1	0.2668641184428723	-0.5912717631142470
±2	-0.07822326652898785	-0.05754352622849957
±3	-0.01686411844287495	0.09127176311424948
±4	0.02674875741080976	
Synthesis Filter Coefficients		
n	Low-Pass Filter h~ (n)	High-Pass Filter g~ (n)
0	1.115087052456994	0.6029490182363579
±1	0.5912717631142470	-0.2668641184428723
±2	-0.05754352622849957	-0.07822326652898785
±3	-0.09127176311424948	0.01686411844287495
±4	0.02674875741080976	

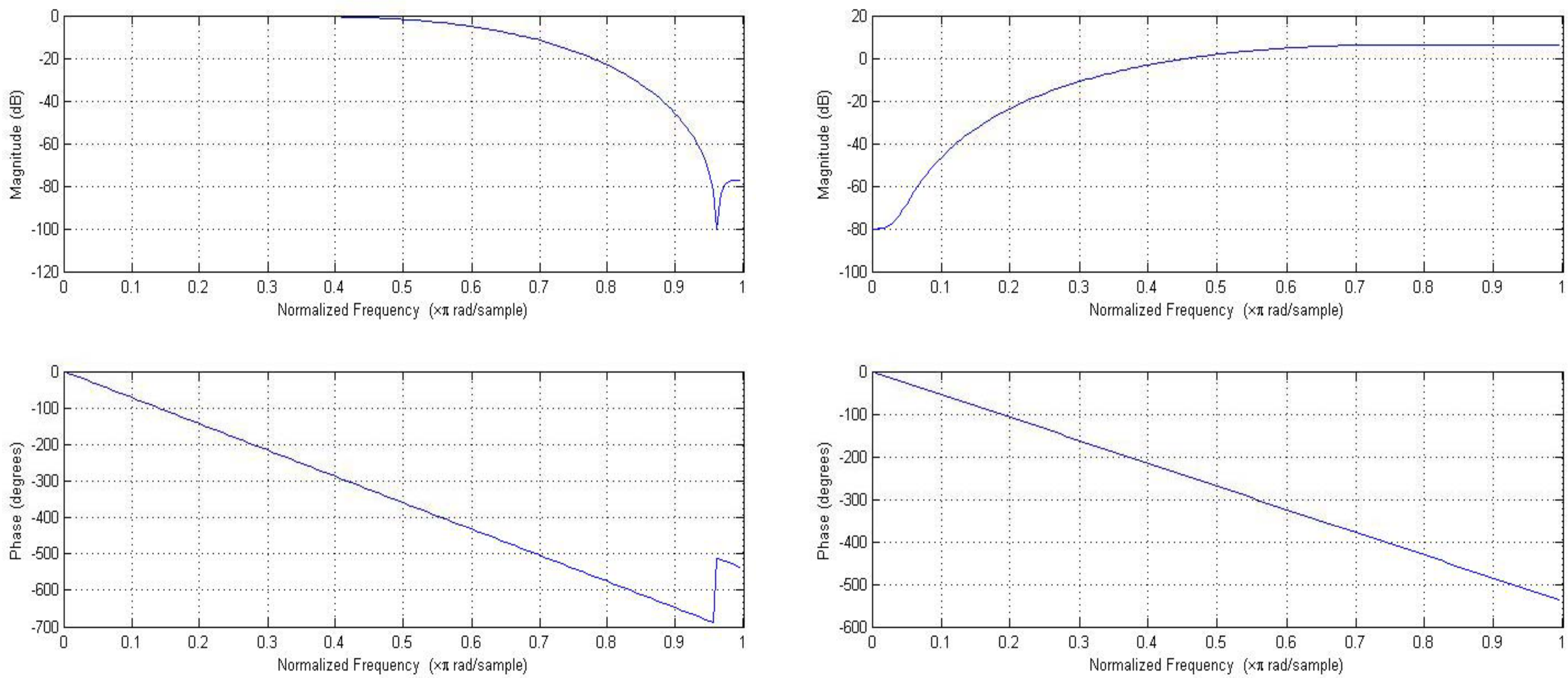
Floating pt Coefficients of Lossy (9,7) Daubechies filter

Le Gall (5,3) Lifting Coefficients	
p_1	-0.5
u_1	0.25
Daubechies (9,7) Lifting Coefficients	
p_1	-1.586134342059924
u_1	-0.052980118572961
p_2	-0.05754352622849957
u_2	0.443506852043971
K	1.230174104914001

Lifting Filter Coefficients of (5,3) and (9,7)

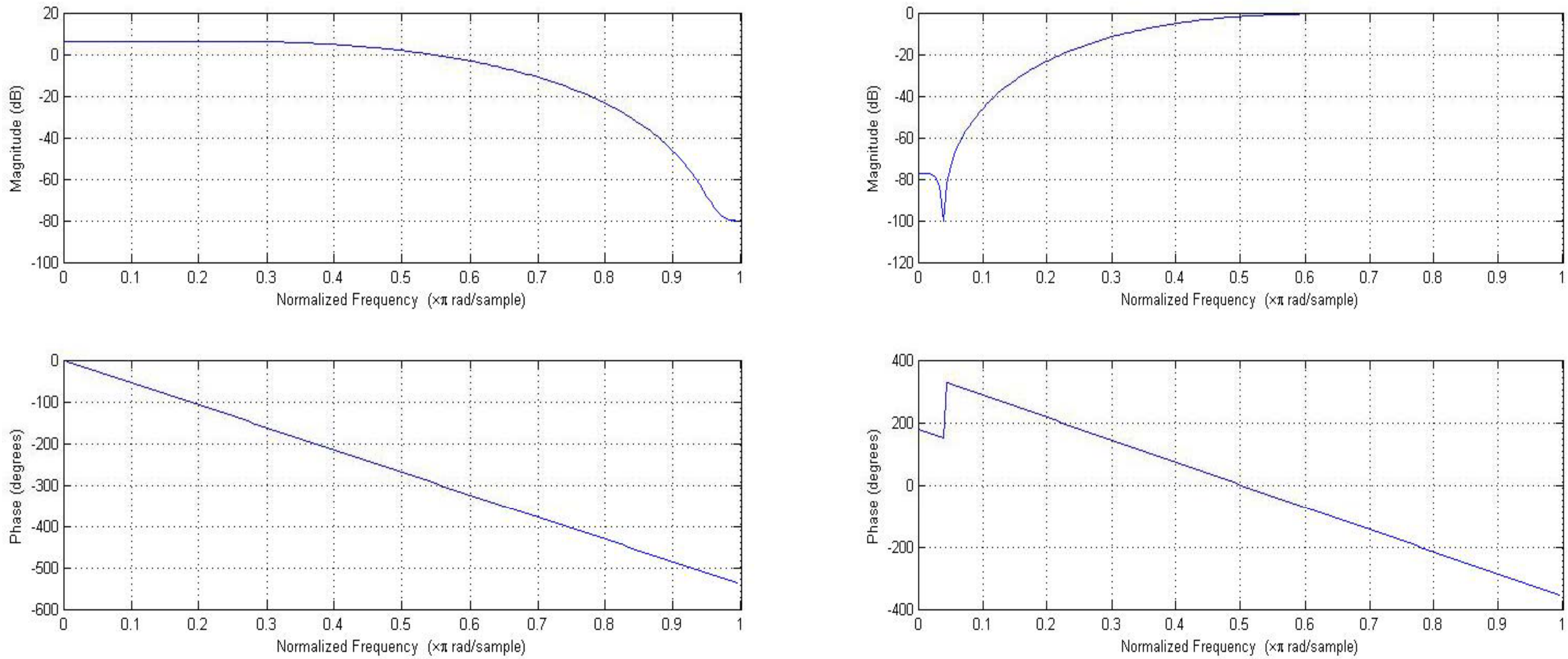
Daubechies 9/7 Analysis Filter

i	Lowpass Filter $h_L(i)$	Highpass Filter $h_H(i)$
0	0.6029490182363579	1.115087052456994
±1	0.2668641184428723	-0.5912717631142470
±2	-0.07822326652898785	-0.05754352622849957
±3	-0.01686411844287495	0.09127176311424948
±4	0.02674875741080976	



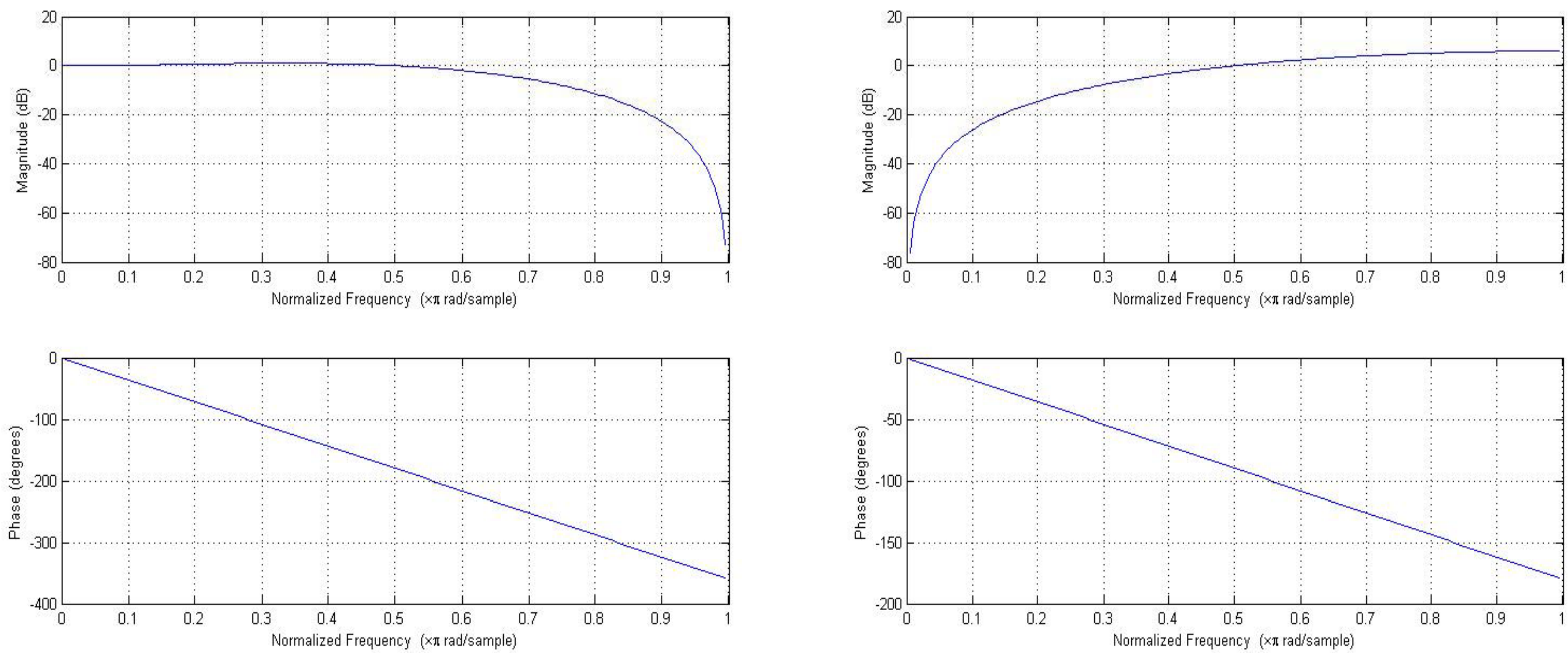
Daubechies 9/7 Synthesis Filter

i	Lowpass Filter $h_L(i)$	Highpass Filter $h_H(i)$
0	1.115087052456994	0.6029490182363579
±1	0.5912717631142470	-0.2668641184428723
±2	-0.05754352622849957	-0.07822326652898785
±3	-0.09127176311424948	0.01686411844287495
±4		0.02674875741080976



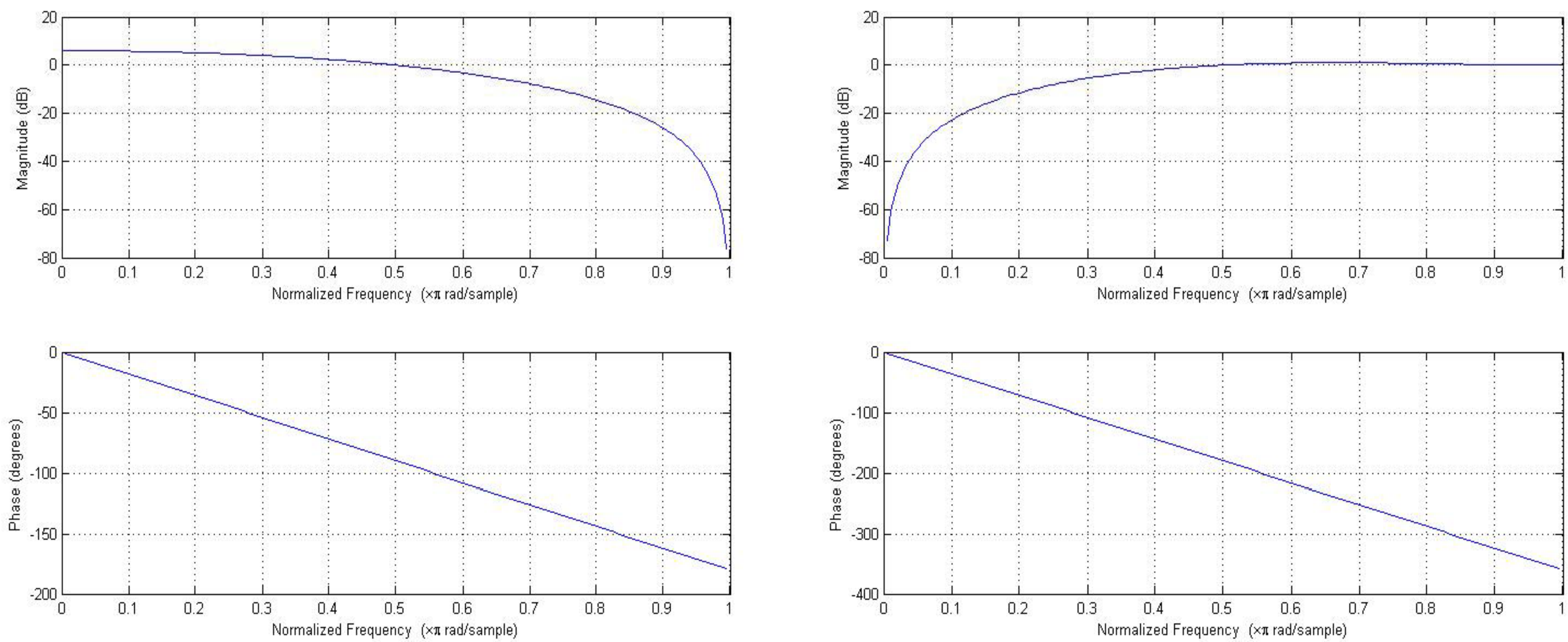
5/3 Analysis Filter

i	Lowpass Filter $h_L(i)$	Highpass Filter $h_H(i)$
0	6/8	1
± 1	2/8	-1/2
± 2	-1/8	



5/3 Synthesis Filter

i	Lowpass Filter $h_L(i)$	Highpass Filter $h_H(i)$
0	1	6/8
±1	1/2	-2/8
±2		-1/8



VLSI Implementation of JPEG2000 Encoder

Presented By

Prof. V. M. Gadre
Shantanu Bhaduri
EE Department
IIT Bombay

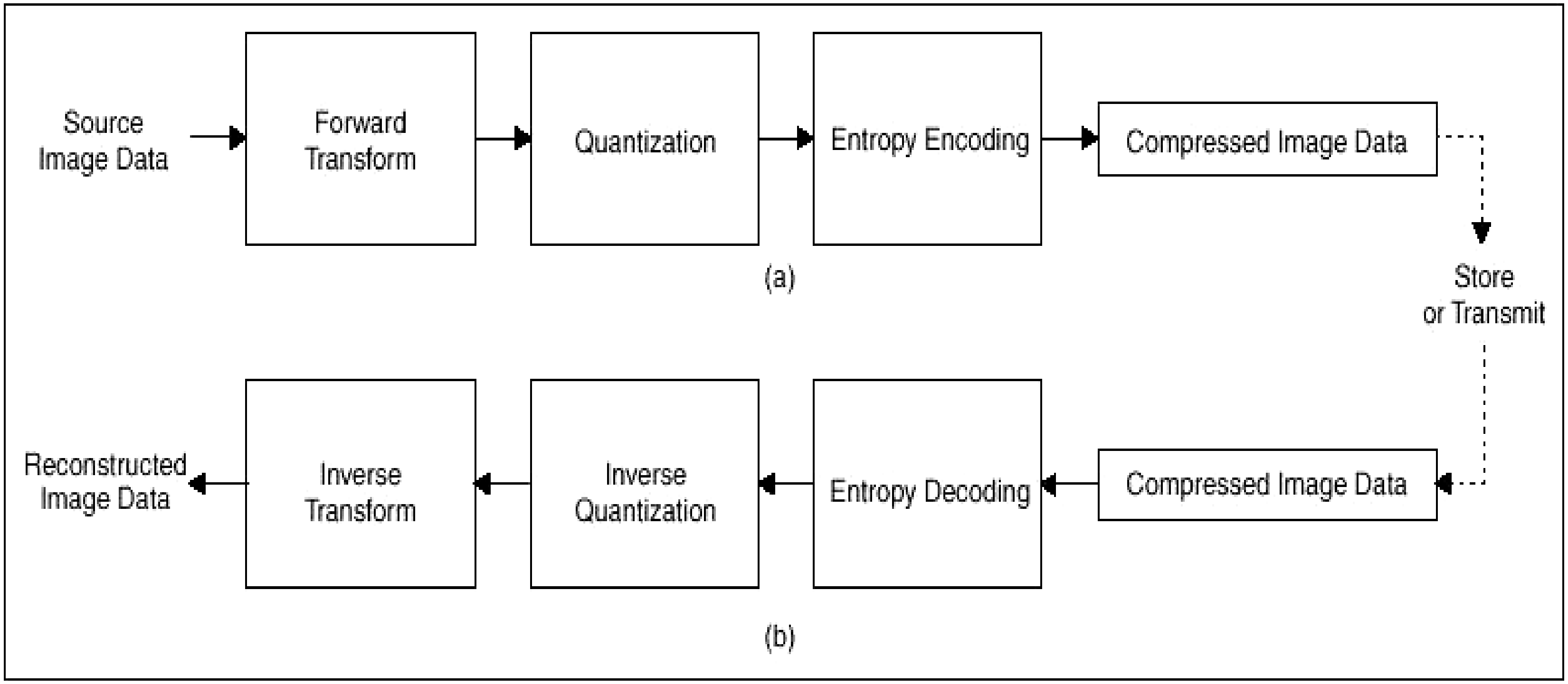
Introduction of JPEG2000

- JPEG (Joint Photographic Experts Group) committee was formed in 1986.
- The committee's first published standard was named as Baseline JPEG.
- It enjoyed its wide spread use in many digital imaging applications.
- In 1996, JPEG committee began to investigate possibilities for new image compression standard that can serve current and future applications.
- This initiative was named as JPEG2000.

JPEG2000 STANDARD

1. Superior Low Bit Rate performance
2. Lossless and Lossy compression
3. Progressive transmission by pixel accuracy and resolution
4. Tiling of digital image
5. Region Of Interest Coding
6. Protective image security

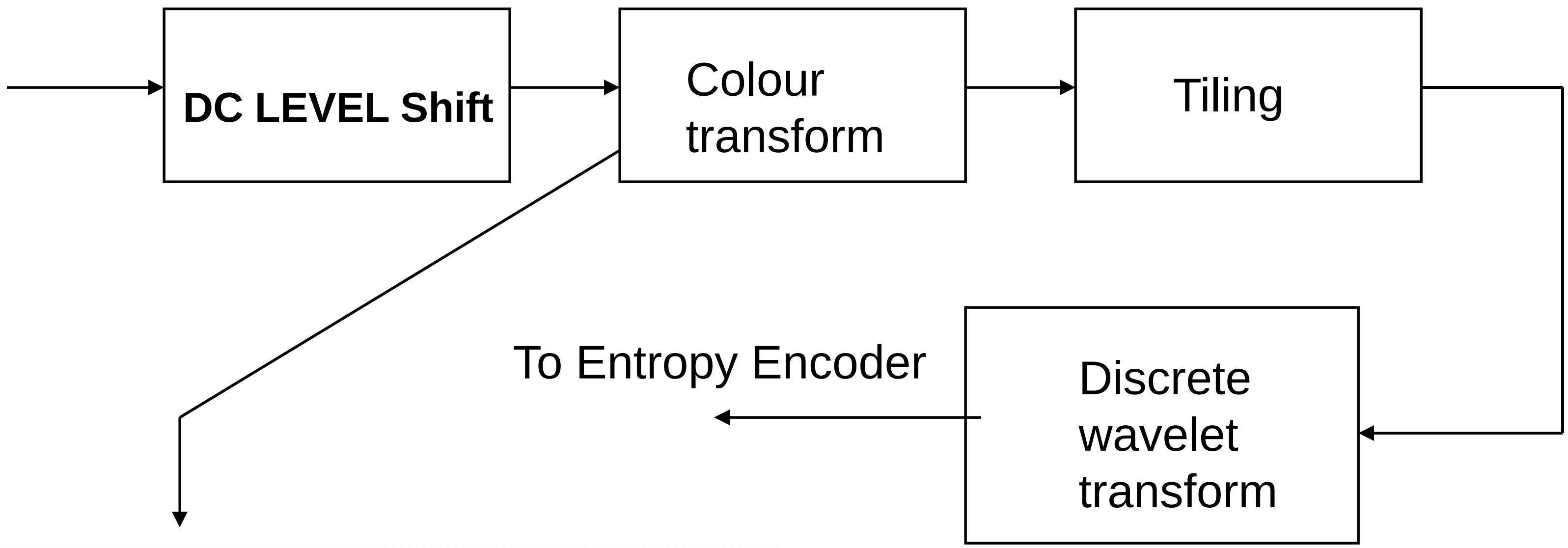
JPEG2000 Engine



General block diagram of the JPEG 2000 (a) encoder and (b) decoder.

A. Skodras, C. Christopoulos and T. Ebrahimi, "The JPEG2000 Still Image Compression Standard," IEEE Signal Process. Magazine, pp. 36-58, Sept. 2001.

Forward Transform



$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

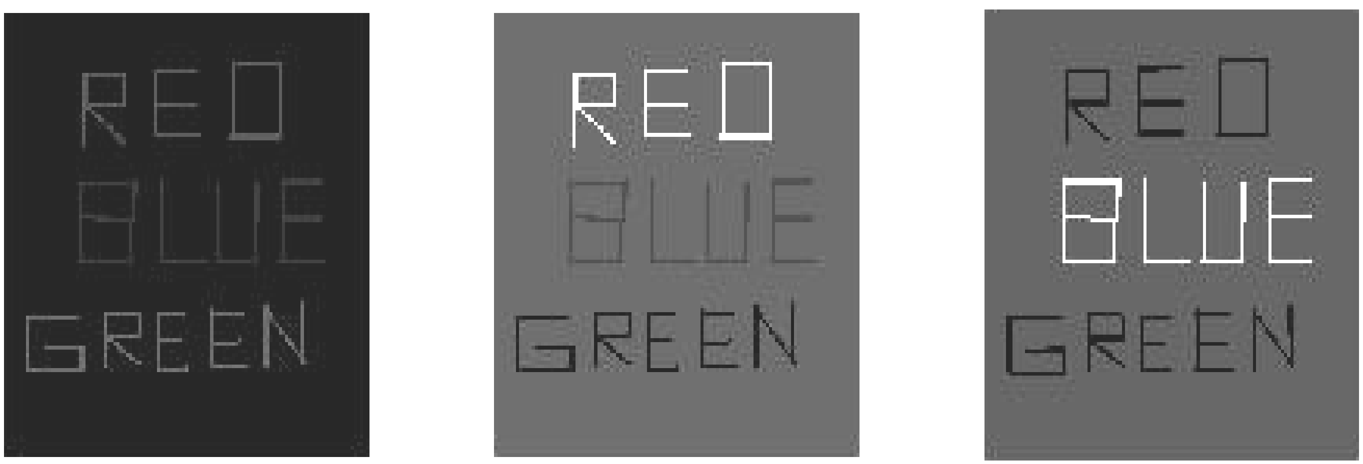
Irreversible Colour transform

Block Diagram Of Forward Transform Block

Matlab Simulation

ORIGINAL IMAGE

Red Green and Blue Component of image



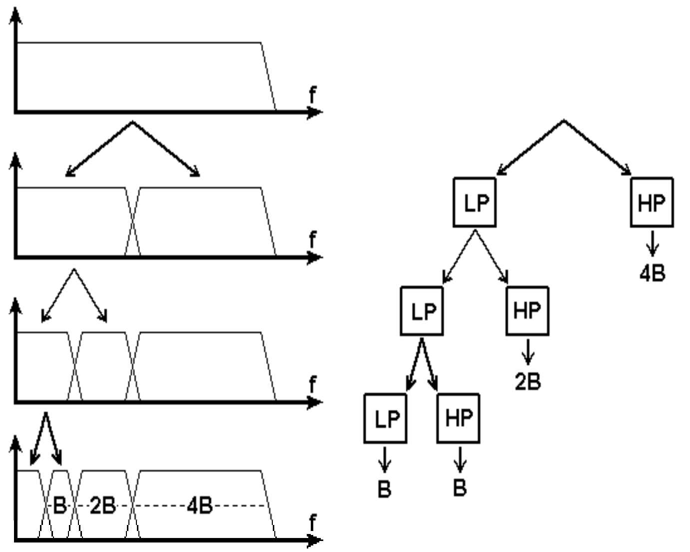
Y Cr Cb component obtained after colour transform

Discrete Wavelet Transform

➢ The 1D DWT →

- The forward 1D DWT is best understood as successive applications of a pair of **low-pass** and **high-pass filters**.

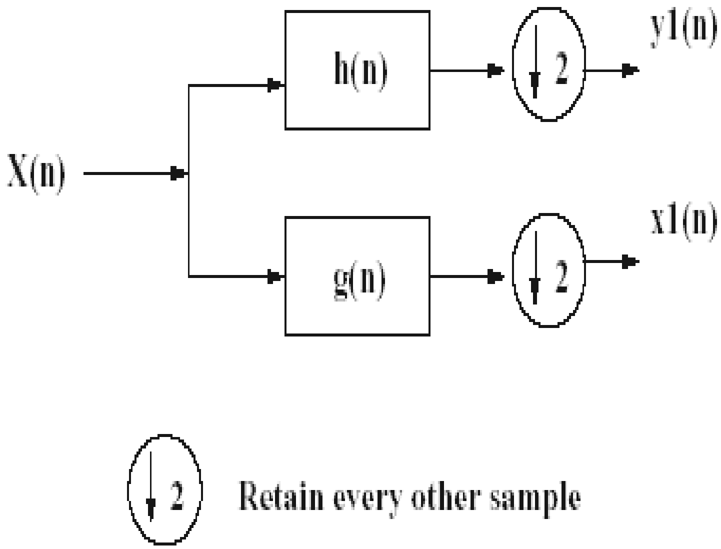
- Which is followed by **down sampling** by a factor of two.

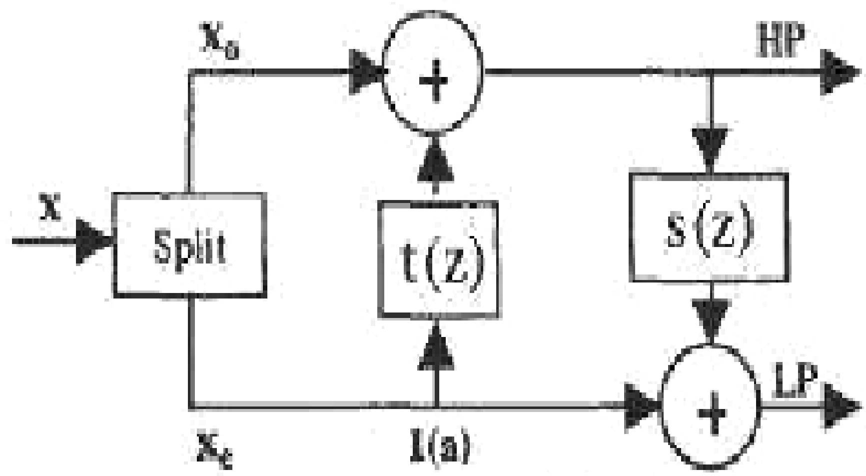


Frequency Domain analysis of filtering

Discrete Wavelet Transform Cont.

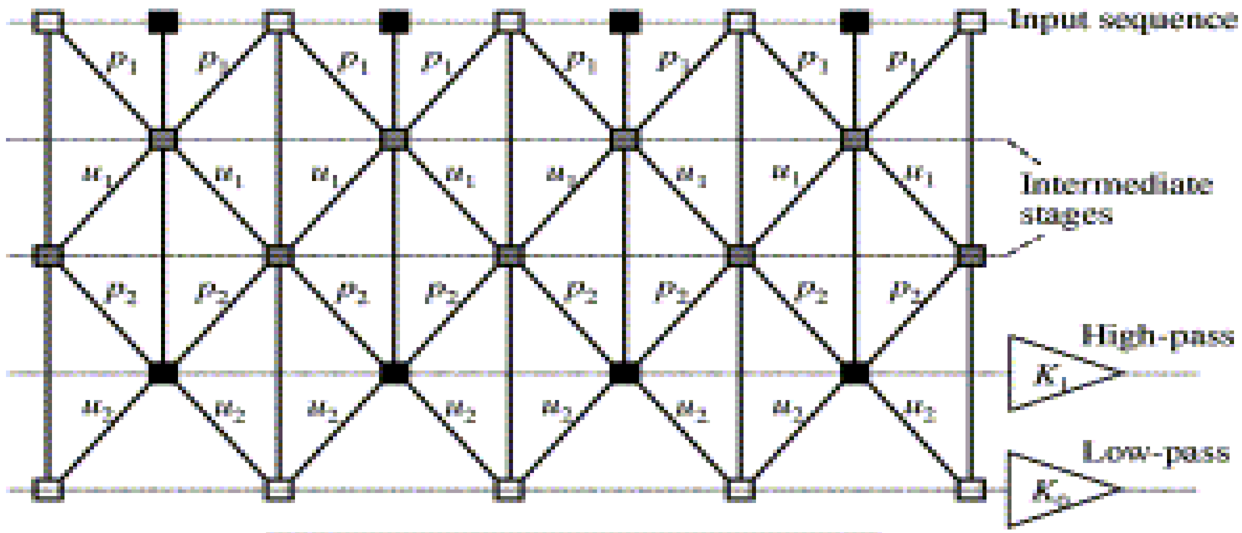
- Two schemes of Implementation





- Convolution Scheme
- Lifting Scheme

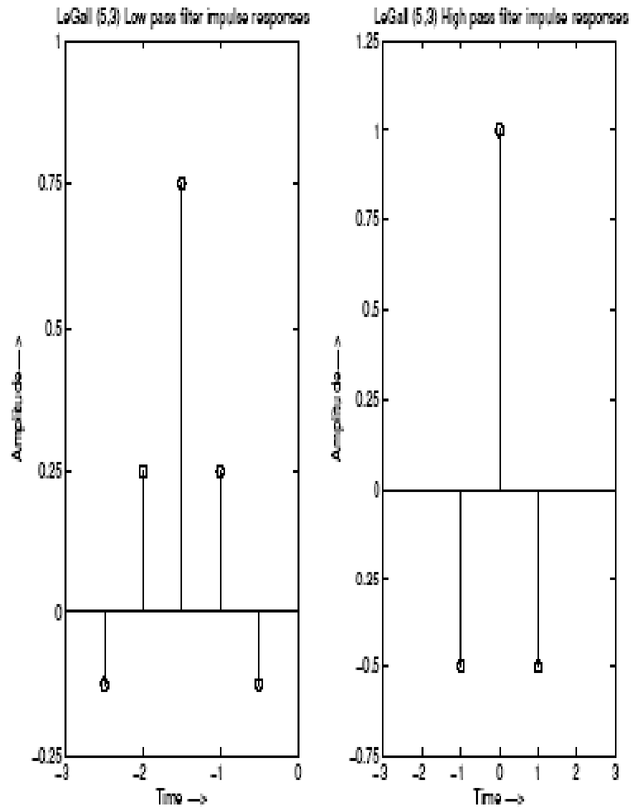
Lifting Scheme for DWT using Daubechies (9, 7) Filters



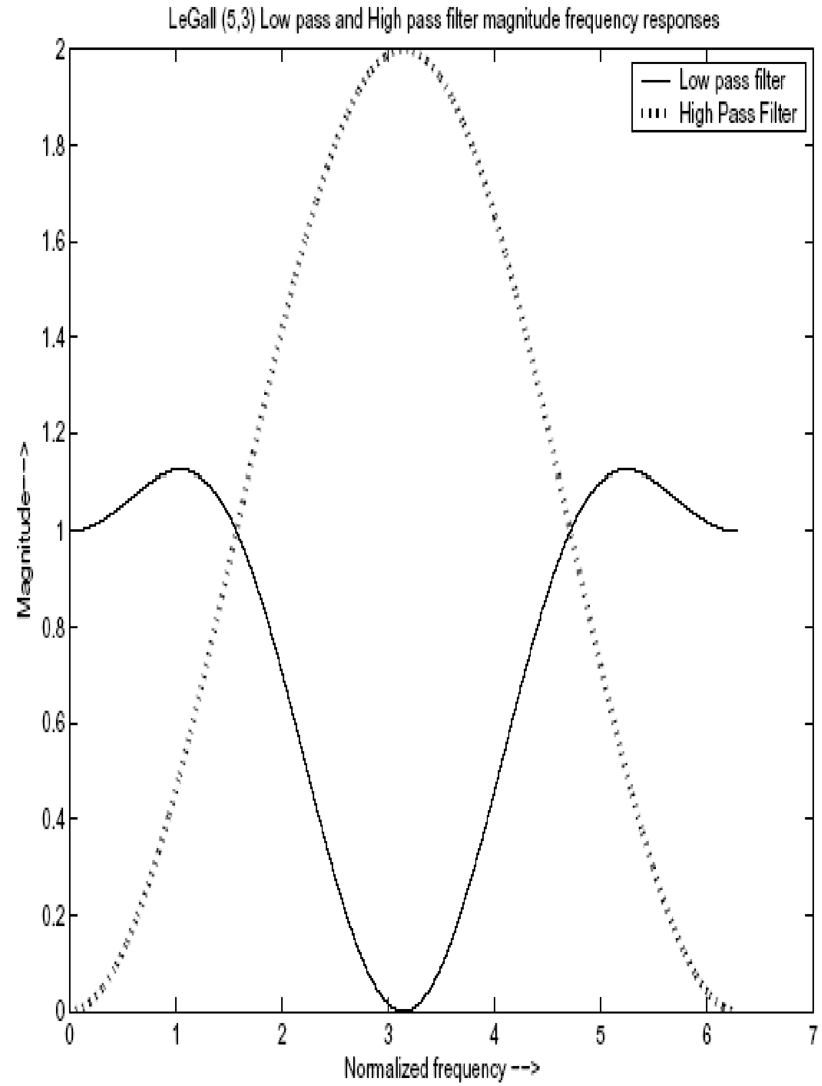
p_1	- 1.586134342059924
u_1	- 0.052980118572961
p_2	+0.882911075530934
u_2	+0.443506852043971
$K_1 = 1/K_0$	+1.230174104914001

An overview of the JPEG 2000 still image compression standard
Majid Rabbani, and Rajan Joshi

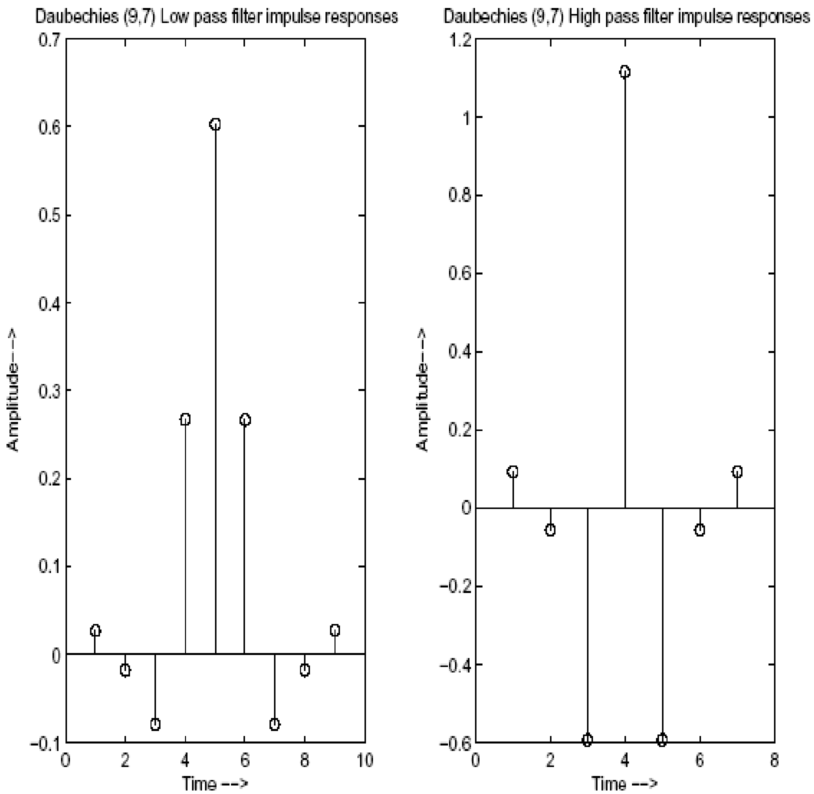
Response Of Le-Gall (5,3) filter



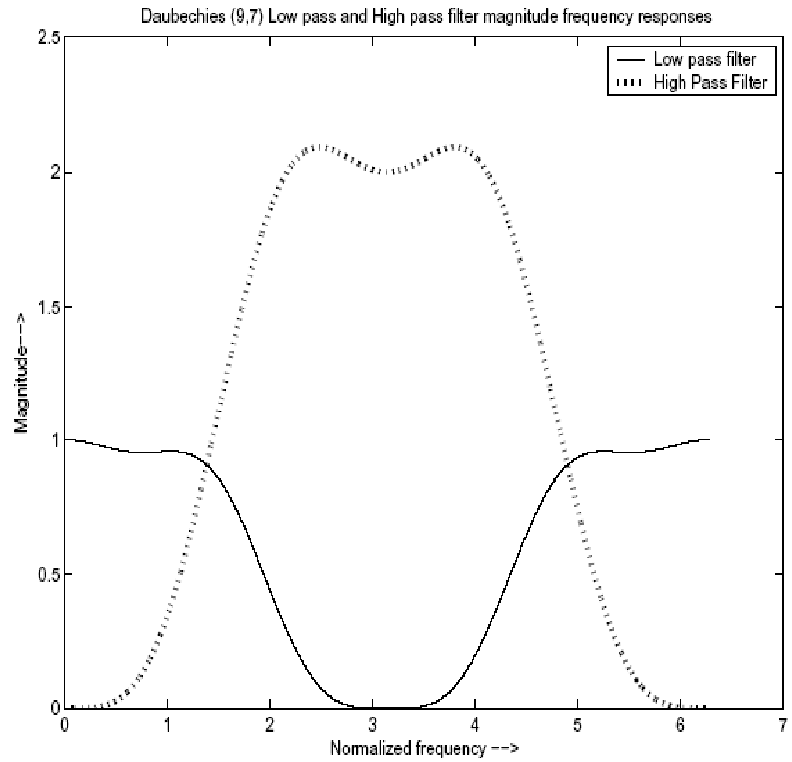
LeGall (5,3) filter impulse response (a)LPF Time domain, (b) HPF time domain



Response Of Daubechies (9,7) Filter



Time Domain response of Low pass And High Pass Filter

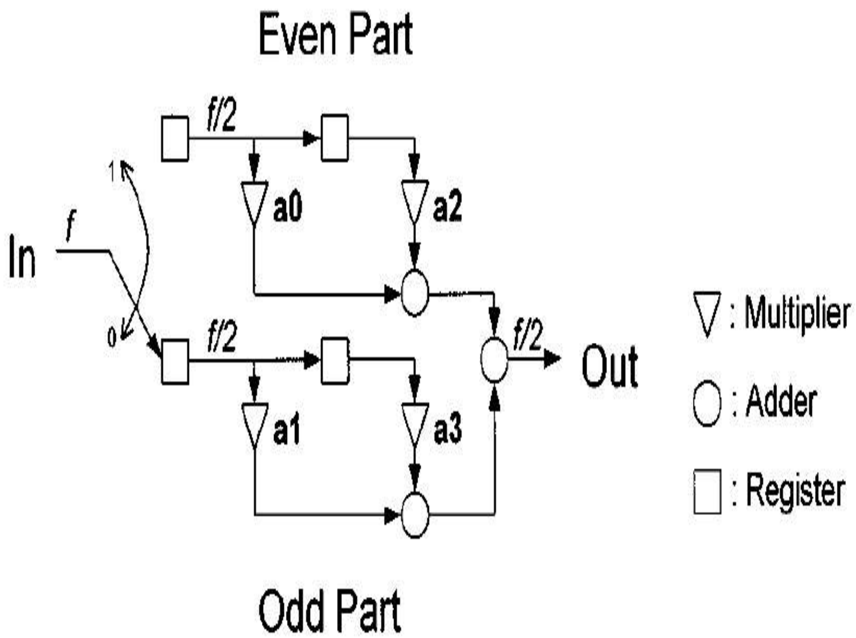


Frequency Domain Response of Low Pass And High Pass Filter

Architectures Implemented for 1D DWT

- Polyphase Decomposition Based Convolution Scheme
- Coefficient Folding Convolution Based Convolution Scheme
- Lifting Scheme
- Coefficient Folding Convolution Based Lifting Scheme

Polyphase Decomposition Scheme

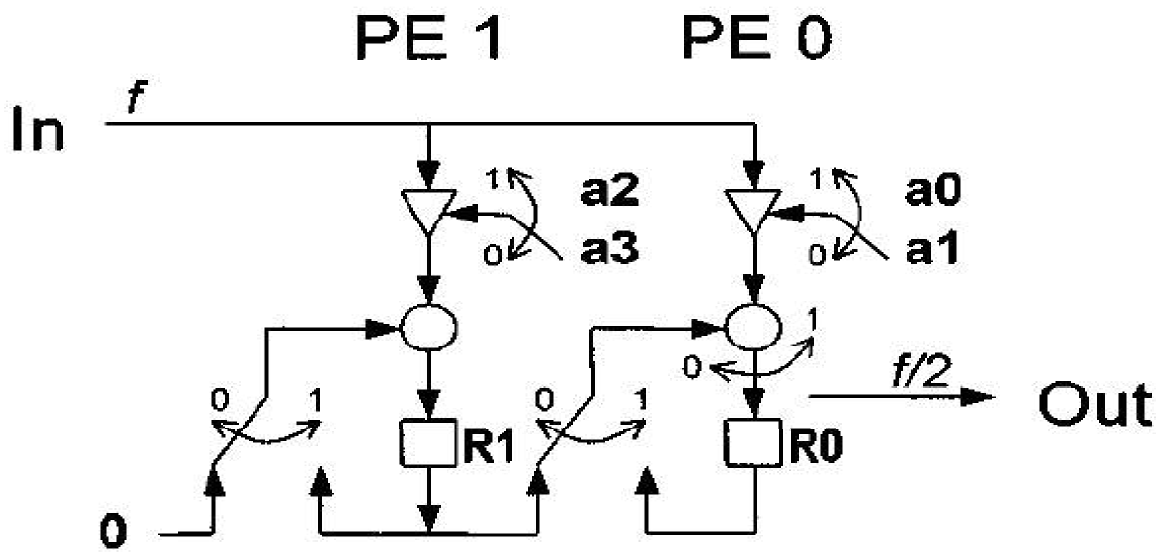


Decimation filter employing the polyphase decomposition technique.

- Input is switched between Even and Odd samples
- Area cost is same *
  No. of multipliers same
- Time cost is half

* Compared to convolution scheme

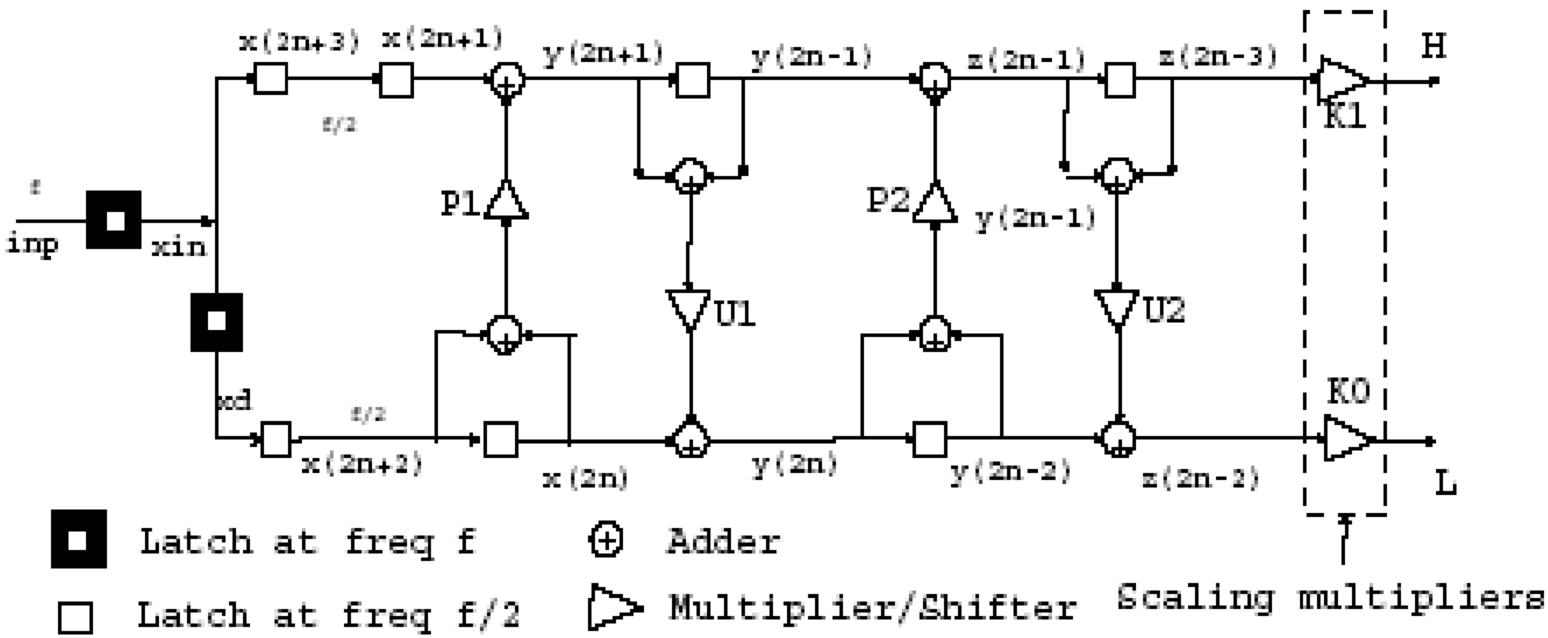
Coefficient Folding Based Lifting Scheme



- Coefficient are switched between one multiplier.
- Input is not switched between even and odd samples
  - Area cost is half *
  - Time cost remains same

* Compared to convolution scheme

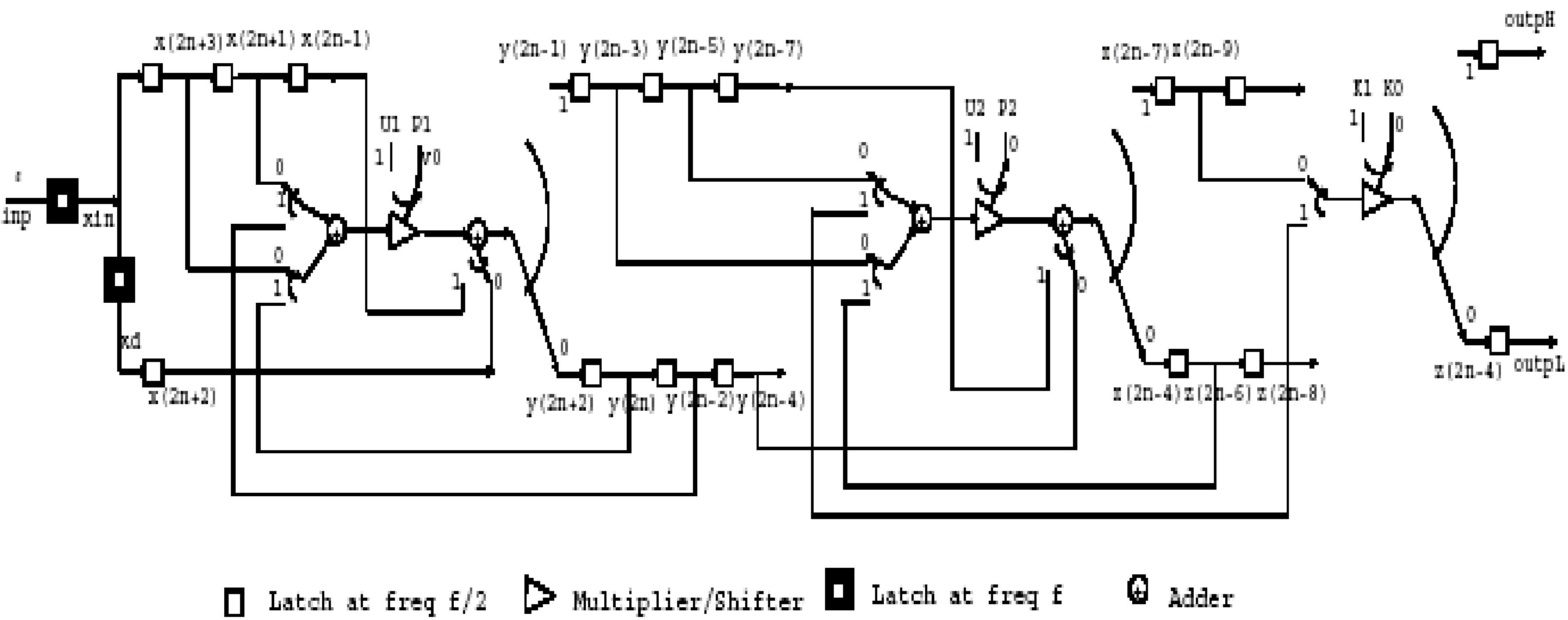
Lifting Scheme



Area Cost is Less than half *
Time cost is half as input is split between even and odd samples

* Compared to convolution scheme

Proposed Coefficient Based lifting Scheme



A combination of Coefficient Folding and lifting Scheme

Number of multipliers are even half as compared to normal lifting scheme

Comparison of Architectures for Daubechies (9, 7) filters

Design	Number of Multiplier	Number of Adder	Critical path
Convolution Scheme	16	14	Tm+8Ta
Polyphase Decomposition	16	14	Tm+5Ta
Coefficient Folding Scheme	9	9	Tm +Ta
Lifting Scheme	6	8	4Tm+8Ta
Coefficient Folding Based Lifting Scheme	3	4	Tm+2Ta

*** Tm is multiplier delay and Ta is adder delay**

Comparison of Architectures

Arch1 = Simple lifting scheme, adopted from [4]
Arch2 = Proposed lifting-based coefficient folding technique
Arch2 = Pipelined lifting-based coefficient folding technique

Design Summary

Parameters	ARCH1	ARCH2	ARCH3	UNITS
No of Slices	1144 (9 %)	805(6%)	827(6%)	Out of 12288(%)
No of Slice Flip Flop	260(1%)	255 (1%)	548(2%)	Out of 24576(%)
No of LUT4	1982(8%)	1290(5%)	1195(4%)	Out of 24576(%)
Equivalent gate count	28172	18100	19585	Gates

Timing Summary

Parameter	ARCH1	ARCH2	ARCH3	UNITS
Minimum Period	81.702	27.325	22.141	nsec
Maximum Frequency	12.204	36.597	45.165	MHz

Modified Booth's Algorithm

$y_{i-1}y_iy_{i+1}$	z_i	z_iX	Comments
000	00	+0	String of 0's
001	01	+X	End of String of 1's
010	01	+X	A Single 1
011	10	+2X	End of String of 1's
100	(-1)0	-2X	Beginning of String of 1's
101	0(-1)	-X	A Single 0
110	0(-1)	-X	Beginning of String of 1's
111	00	+0	String of 1's

- **Reduce the number of partial products by re-coding the multiplier operand**
- **Works for signed numbers**
- **Because we need constant multipliers , so instead of 8:1 mux we require 2:1 mux, so reduced area.**

Example

Example: Multiply -118d by -99d

B = -118d = 1000 1010b
-B = 118d = 0111 0110b
2B = -236d = 1 0001 0100b
-2B = 236d = 0 1110 1100b

A = -99d = 1001 1101b
-99d = $\bar{2}2\bar{1}1$

Sign Extension

Radix-4 Booth

Step1) Initialize

-118d = 0111 0110b
-99d = $\bar{2}\ 2\ \bar{1}\ 1$

Step2) Find partial products

1111111110001010b	B
01110110 b	-B
11100010100 b	2B
011101100 b	-2B
0010110110100010 b	

Step3) Sum up the shifted partial products

Convert 2's-Comp back to decimal:
0010 1101 1010 0010 = 11682d

Comparison of architectures

Arch1 = Proposed Lifting –based coefficient folding technique
Arch2 = Proposed lifting-based coefficient folding technique using Booth's Multiplier
Arch2 = Pipelined lifting-based coefficient folding technique using Booth's Multiplier

Design Summary

Parameters	ARCH1	ARCH2	ARCH3	UNITS
No of Slices	805(6%)	644(5%)	910(7%)	Out of 12288(%)
No of Slice Flip Flop	255 (1%)	260 (1%)	1,149(4.5%)	Out of 24576(%)
No of LUT4	1290(5%)	901(3.6%)	1,340(5.5%)	Out of 24576(%)
Equivalent gate count	18100	12,291	23,261	Gates

Timing Summary

Parameter	ARCH1	ARCH2	ARCH3	UNITS
Minimum Period	27.325	26.358	11.985	nsec
Maximum Frequency	36.597	37.939	84.25	MHz

Advantages Of Proposed Scheme

- Reduced area as the number of multipliers are reduced. Nearly 30% reduction in area
- Reduced critical path
- Speed is increased 3 times
- Reduced glitches in outputs
- Reduction in total power dissipation

Line based-Lifting Scheme for 2D DWT Architecture

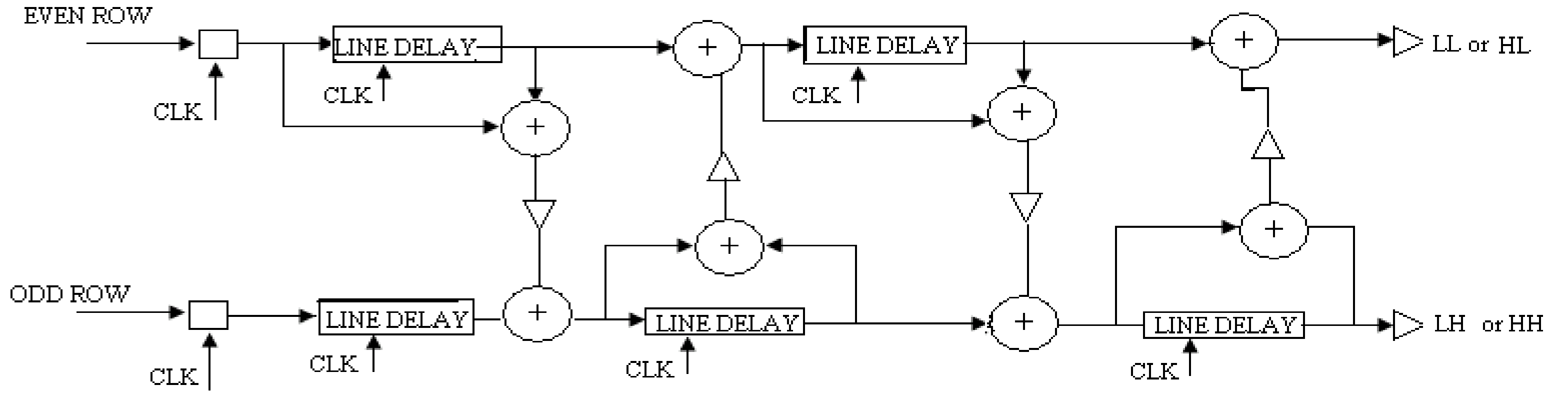
Direct Implementation

N x N storage space is required
For an N x N image

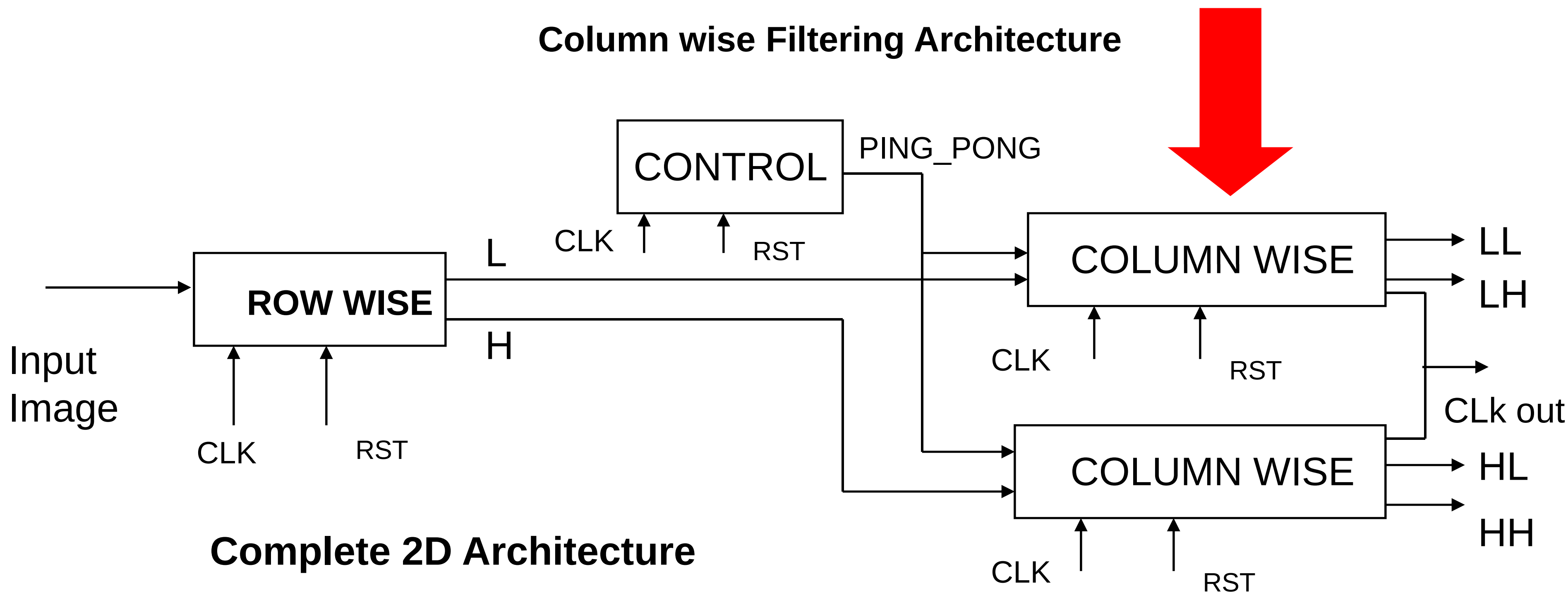
Line Based Implementation

10N storage space is required
For N x N image

2D Architecture Cont..

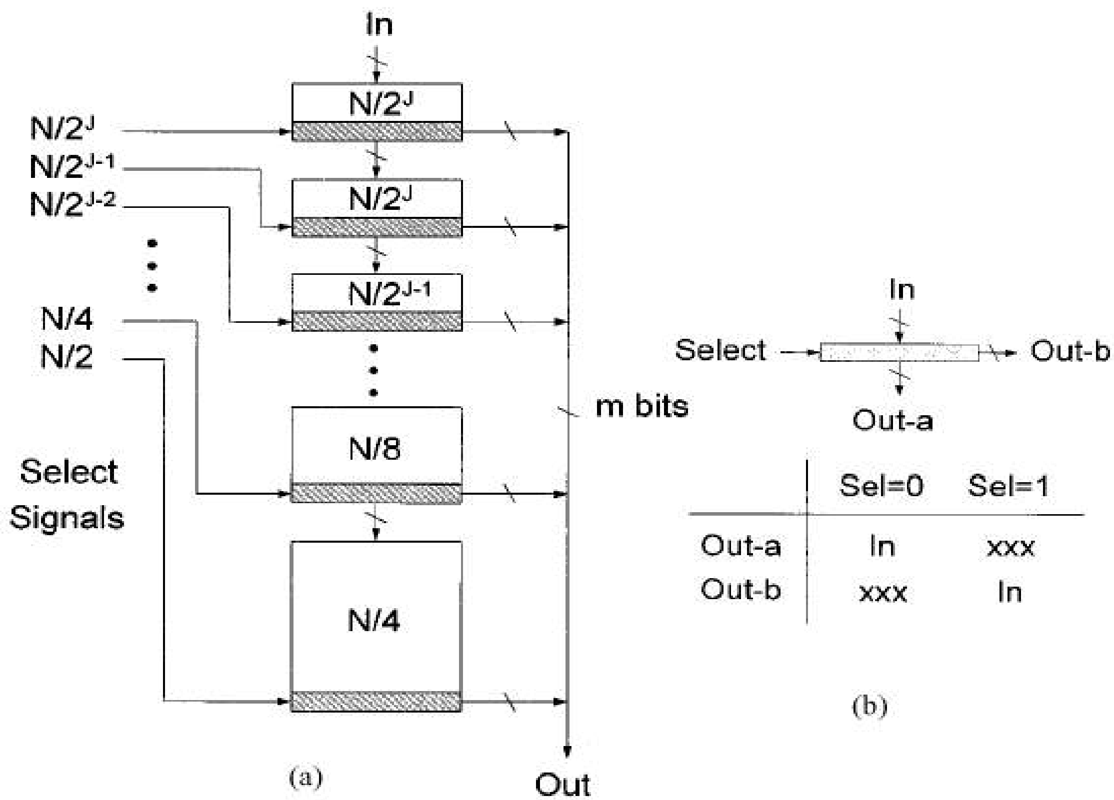


Column wise Filtering Architecture



Complete 2D Architecture

Variable length line Delays



(a) **Variable length** Line delay with select signals to change its size to $N/2, N/4, N/8, \ldots, N/2^J$ in the different decomposition levels.
(b) The 1 to 2 demultiplexer used in the line delay.

Synthesis Report

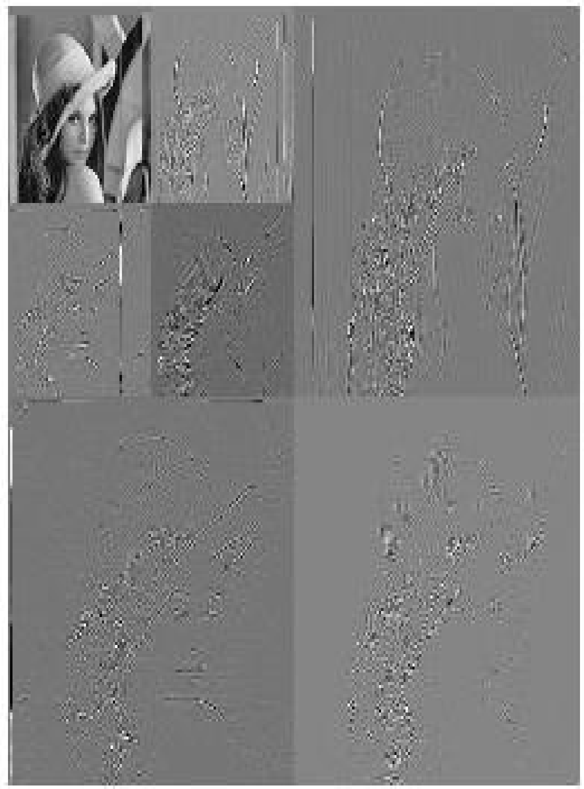
Following table shows final synthesis report for 2D DWT synthesized on Virtex2p FPGA

Design Summary

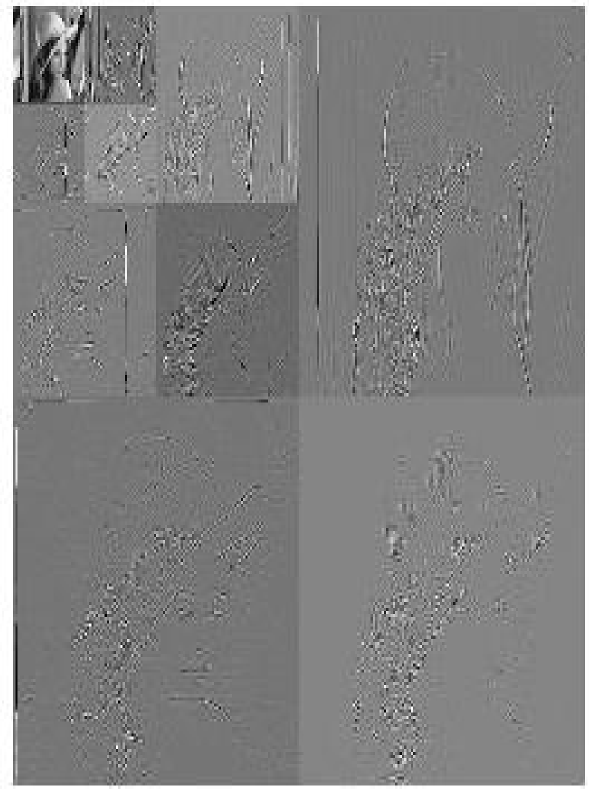
Parameters	2D Architecture	Out of
No of Slices	11,349	13,696
No of Slice Flip Flop	20,283	27,392
No of LUT4	1,955	27,392
Maximum Frequency	148.39 MHz	----
Minimum delay	6.739ns	------
Equivalent gate count	2,677,355	4,000,000

Simulation Result

(A)



(B)

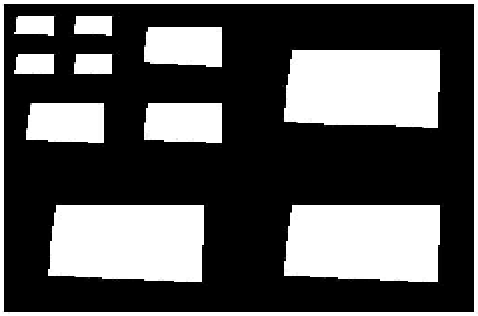


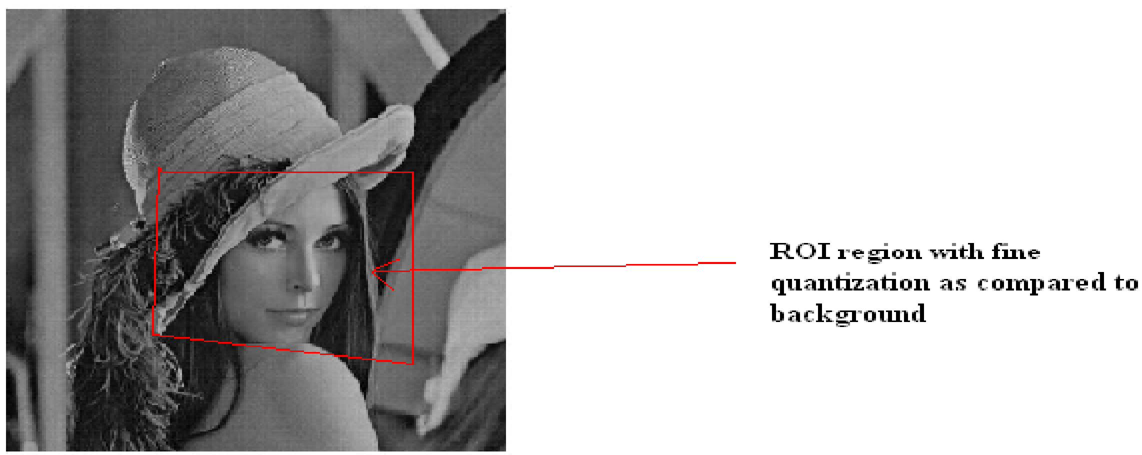
(C)

(A) Lena image after 1 level of filtering (B) after 2 level of filtering (C) after 3 level of filtering

Region of Interest Coding

1. Mask generation for arbitrarily shaped ROI

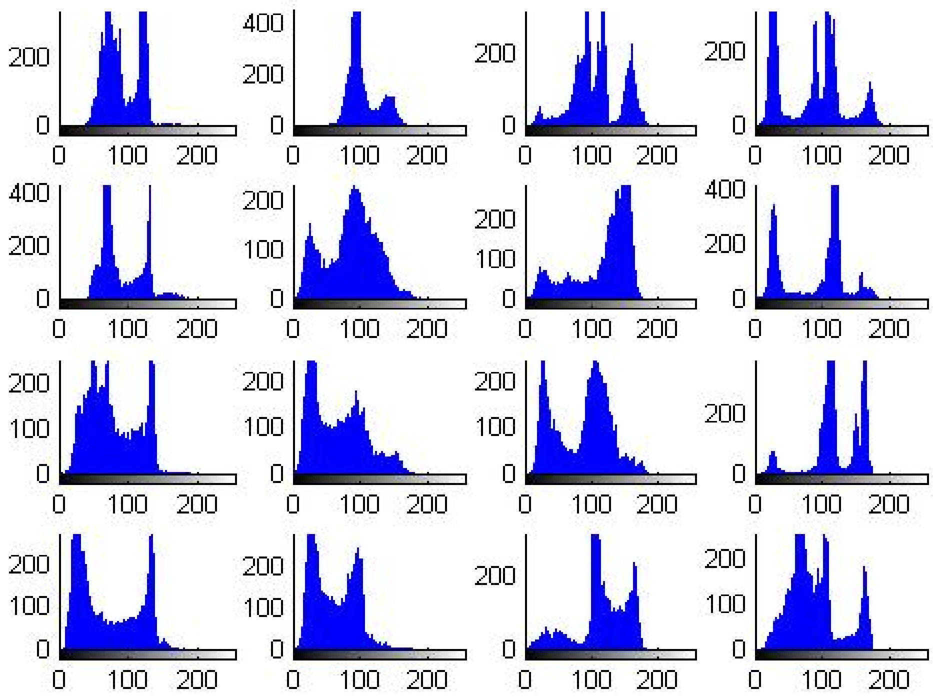


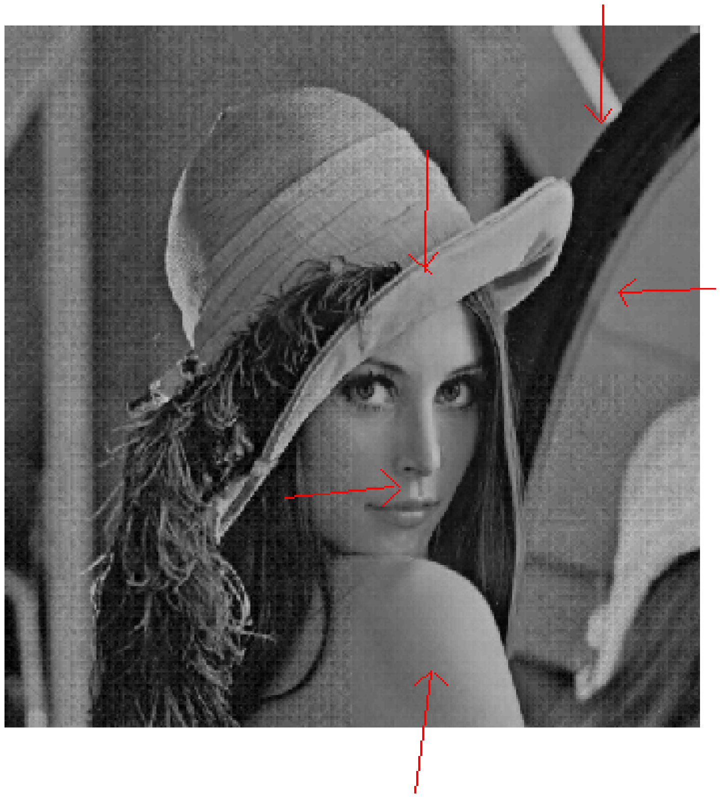


**(A) Three level decomposition of mask, (B) The ROI of the image,
(C) The image after quantization**

Region of interest (cont.)

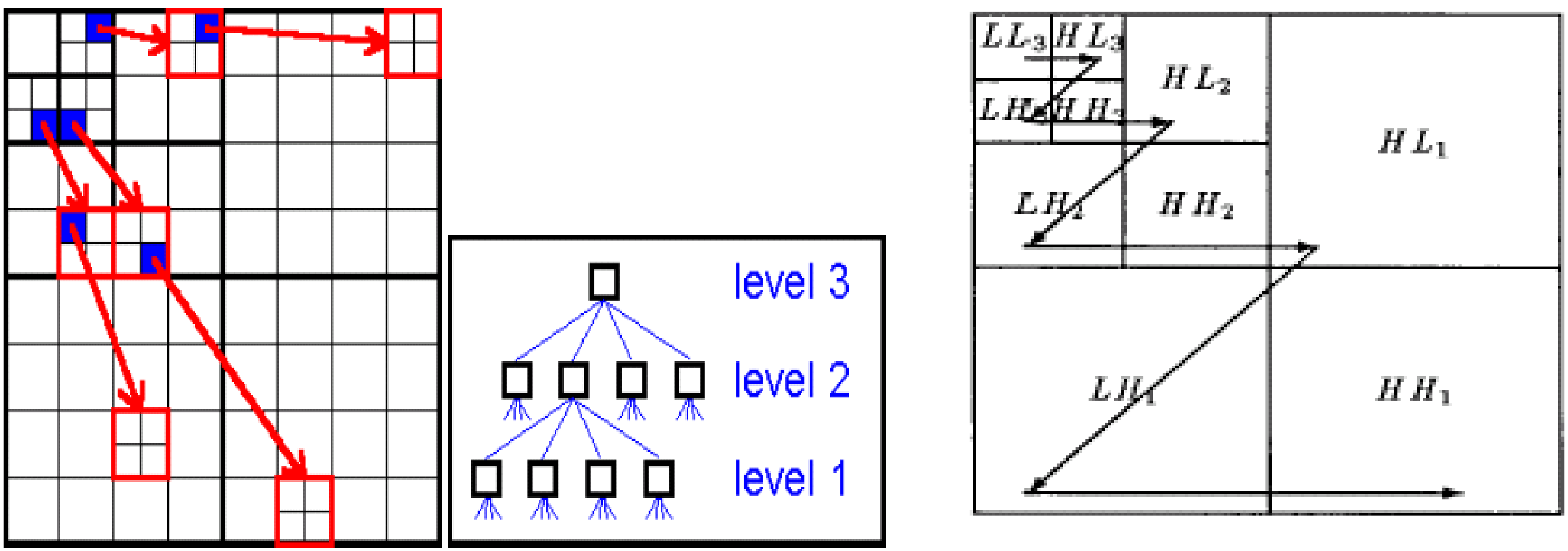
2. Adaptive determination of ROI





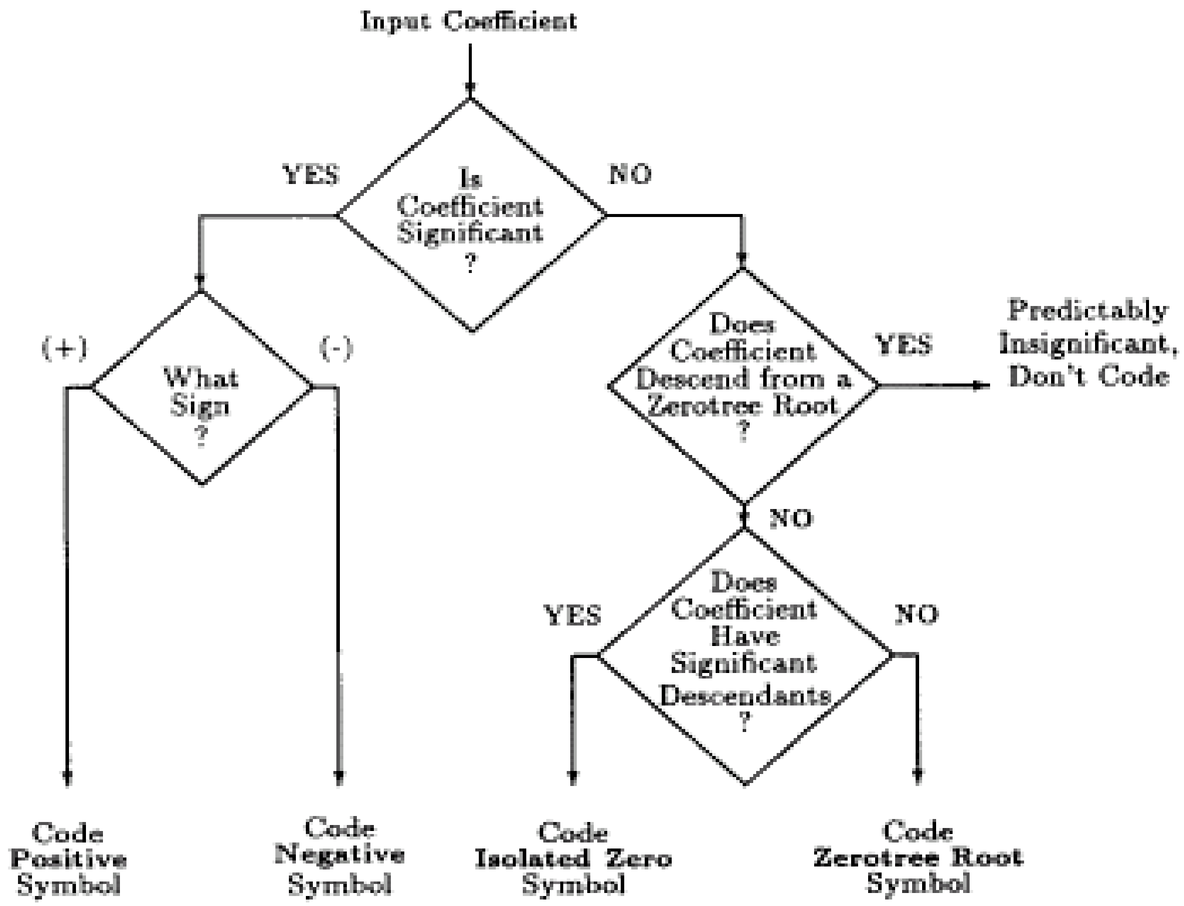
(A) The histogram of Lena image (B) The quantized Lena image

Embedded Zero Wavelet Tree Algorithm (EZW)



The relations between wavelet coefficients in different sub bands as quad-trees.

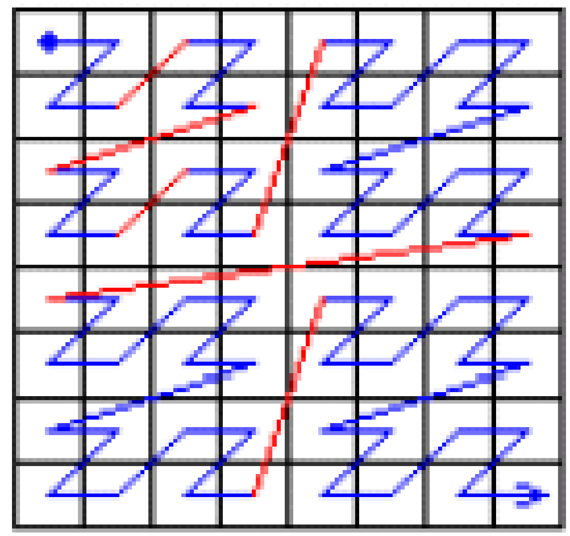
Flow Chart for EZW Algorithm



Flow chart for encoding a coefficient of the significance map

Example

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4



```
D1:  pnztpttttztttttttptt
S1:  1010
D2:  ztnptttttttt
S2:  100110
D3:  zzzzzppnppnttnnptpttnttttttttptttpttttttttttptttttttttttt
S3:  10011101111011011000
D4:  zzzzzzztztznzzzzpttptpptpnptntttttptpnpppptttttptptttpnp
S4:  11011111011001000001110110100010010101100
D5:  zzzzztzzzzztpzzzttpttttnptppttptttnppnttttpnnpttpttppttt
S5:  101111001101000101111101011011001000000000110110110011000111
D6:  zzzttztttztttttnnttt
```

Result

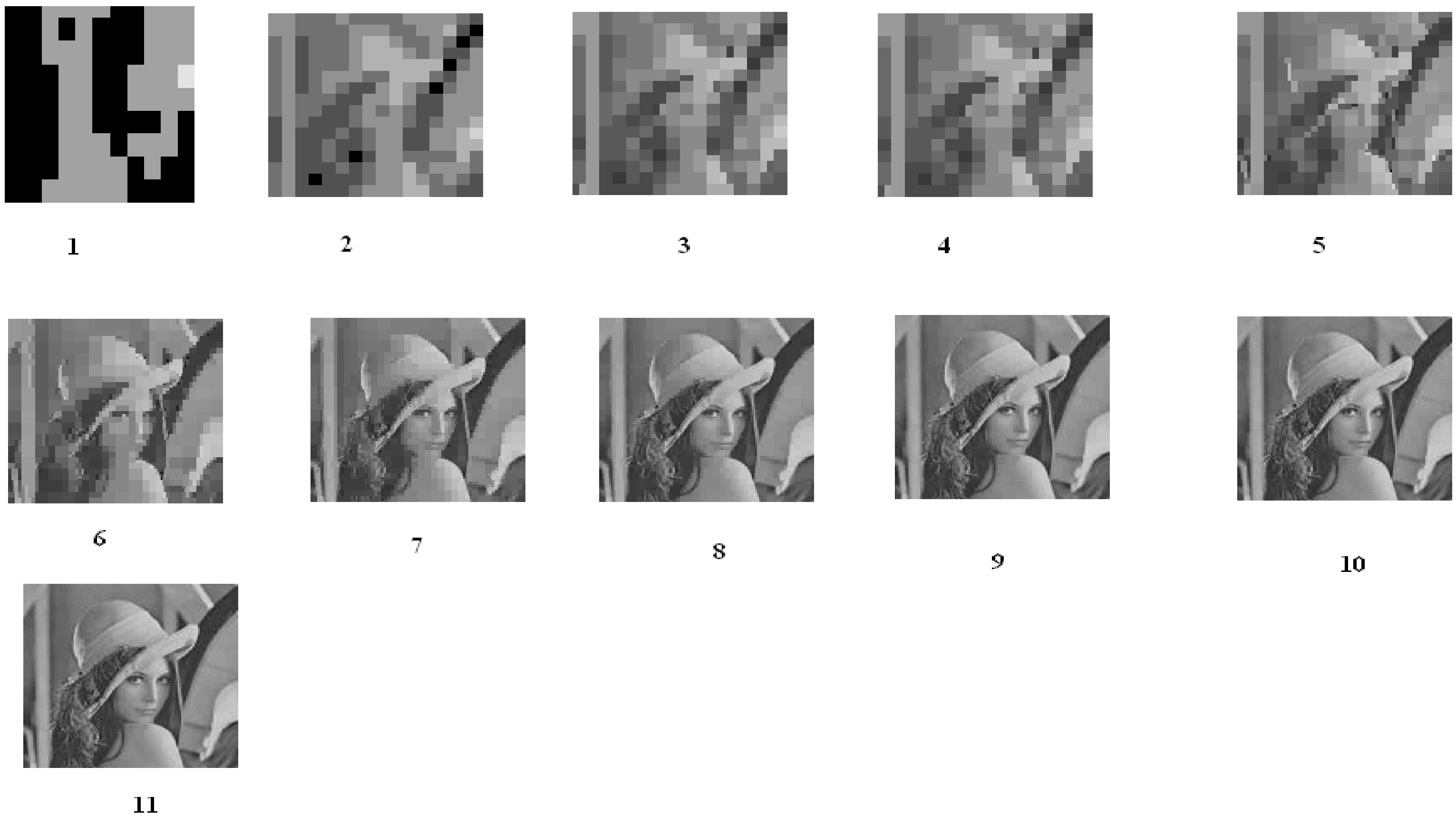


Image generated after decoding each level of encoding

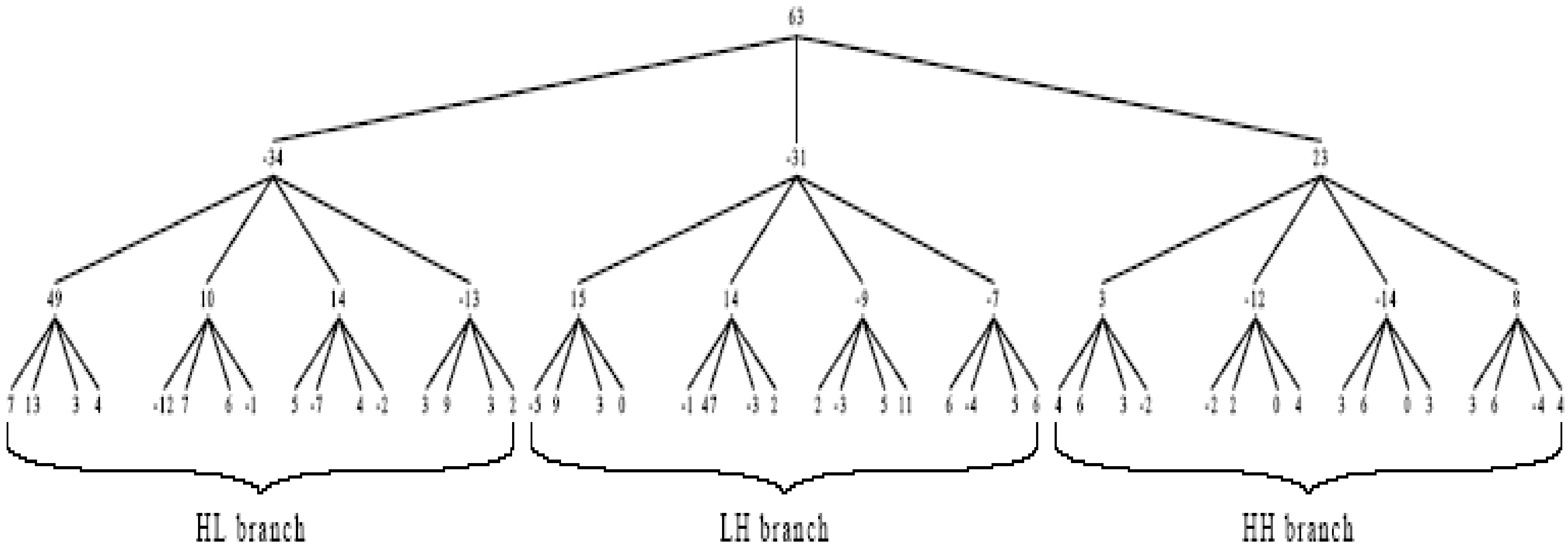
Results (Cont.)

Level Number	Mean Square Error (MSE)	PSNR=$10\log_{10}(255^2/MSE)$ dB
1	6314.9	10.1279
2	702.5	19.6644
3	482.5	21.2713
4	329.4	22.9539
5	182	25.5301
6	71.1	29.6147
7	23.9	34.4090
8	7.1	39.8220
9	2.1	45.5381
10	0.9	48.6664
11	0.8	49.3142

VLSI Architecture for EZW Algorithm

- Depth First Search Algorithm (DPS)
- Proposed VLSI Architecture for EZW Algorithm

DPS Algorithm



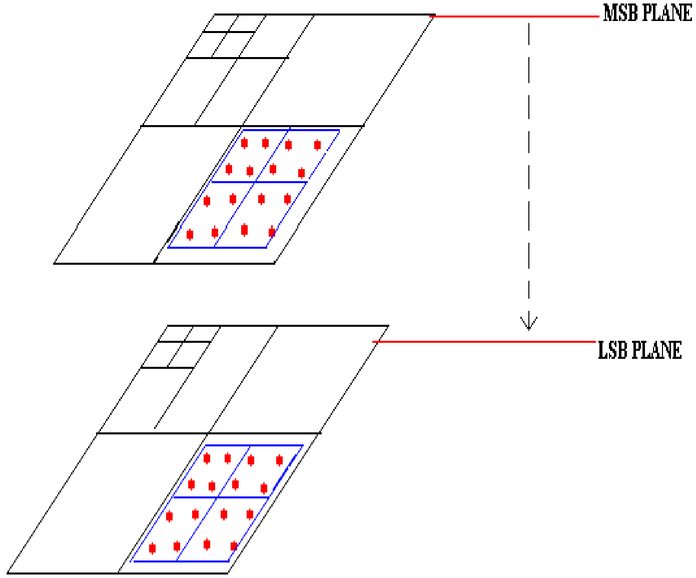
Tree representation of wavelet coefficients

Proposed Architecture for EZW Algorithm

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

Bit wise OR for every child group

A



B

A Grouping method used to reduce looping of tree
B BIT Plane Representation of Proposed Architecture

Design flow of Algorithm

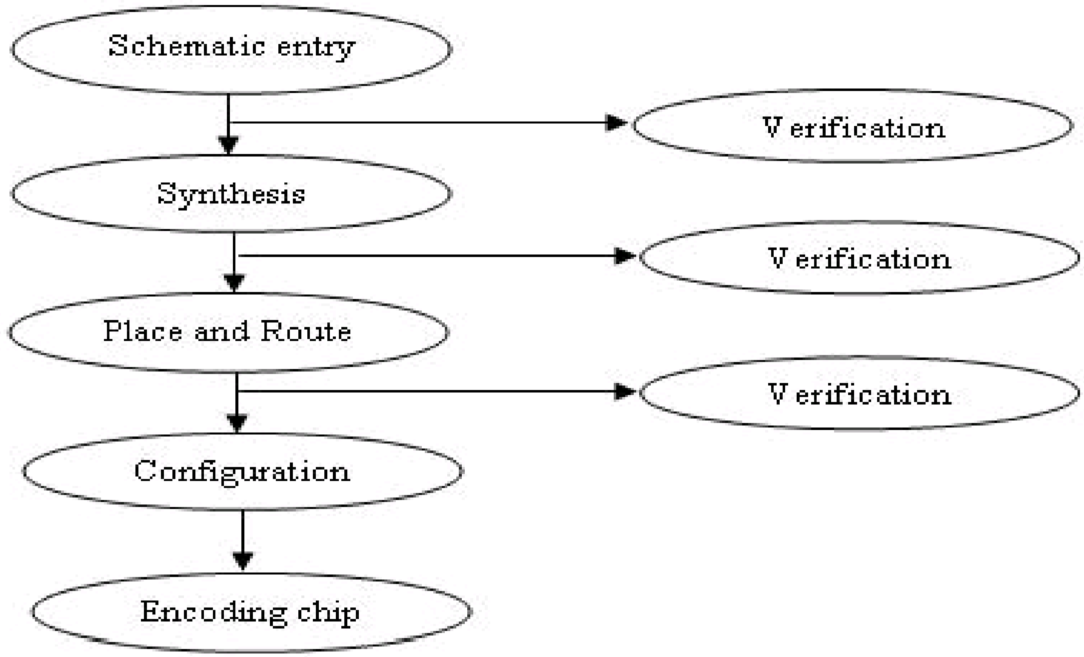
- Generate child group after each level of filtering
- Grouping goes upto second level
- Start Morton Scan
- If dominance is found then replace the coefficient with zero value
- If parent is found dominant then check flag used to indicate it to child
- Check flags used to indicate coding of coefficients used for subordinate pass.

Synthesis Result of Algorithm for 8x8 image

Design Summary

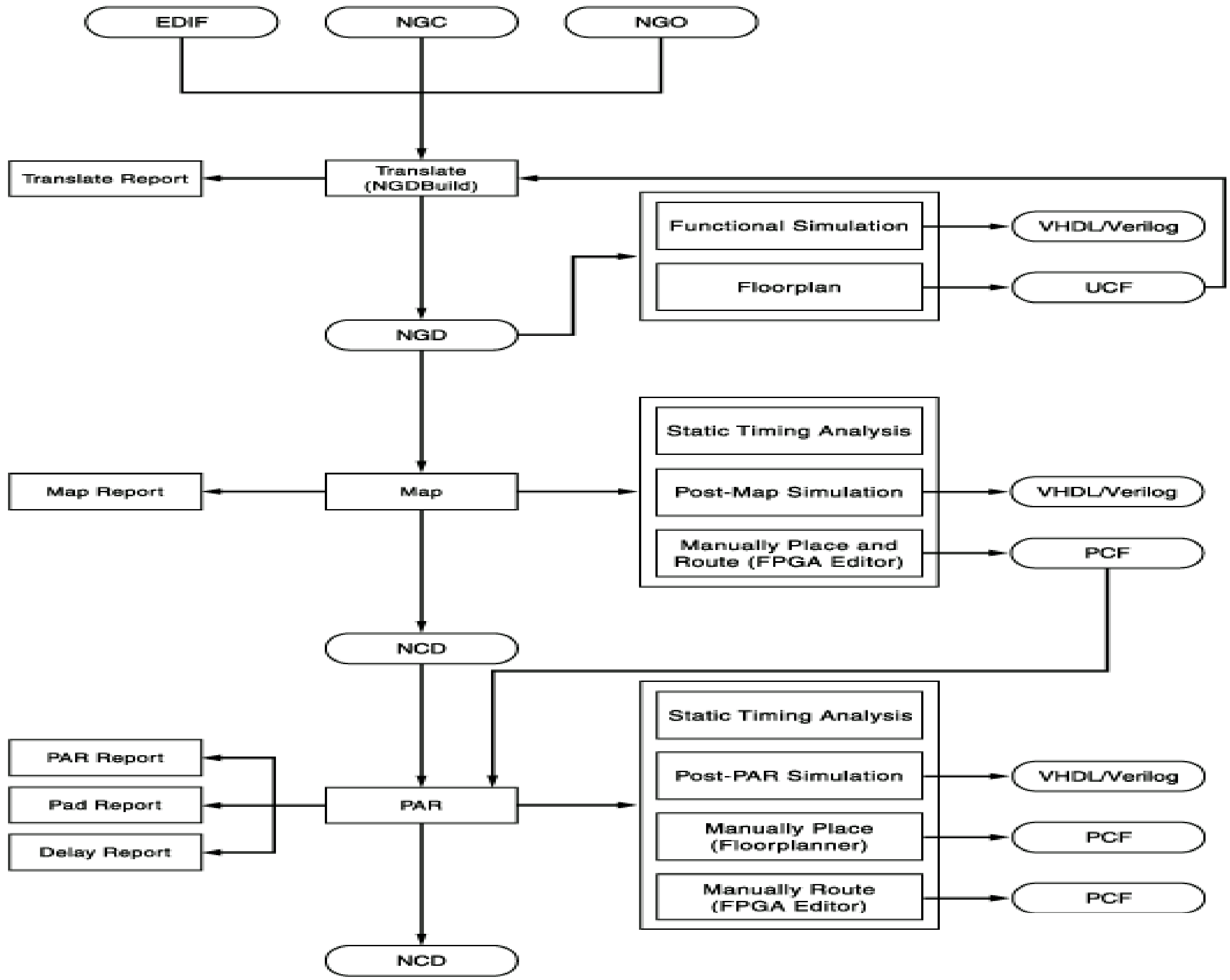
Parameters	2D Architecture	Out of
No of Slices	528	13,696
No of Slice Flip Flop	341	27,392
No of LUT4	671	27,392
Maximum Frequency	124.023 MHz	------
Minimum delay	8.063ns	------
Equivalent gate count	76,611	4,000,000

Testing of Hardware



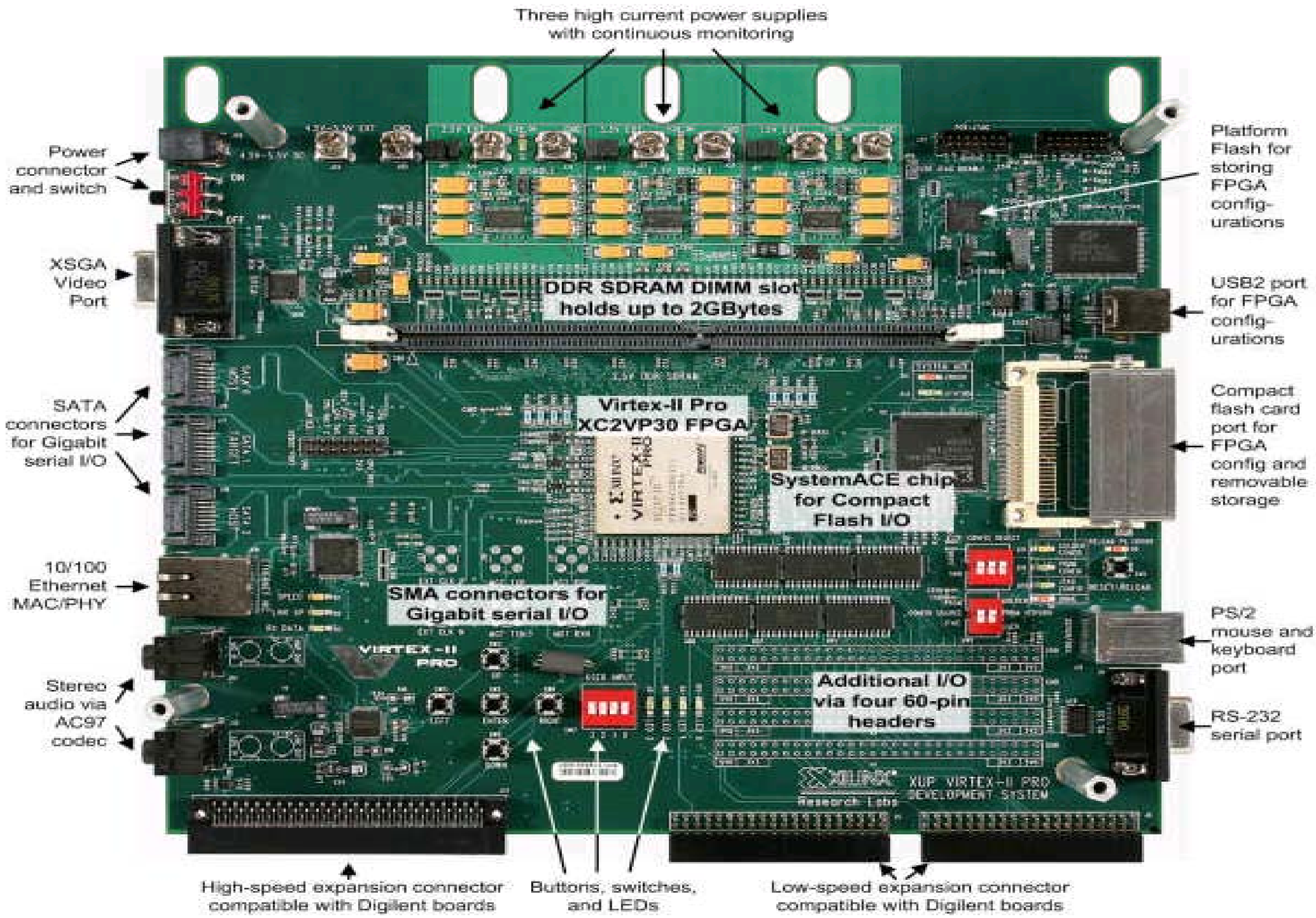
FPGA Design Flow

FPGA Flow cont…

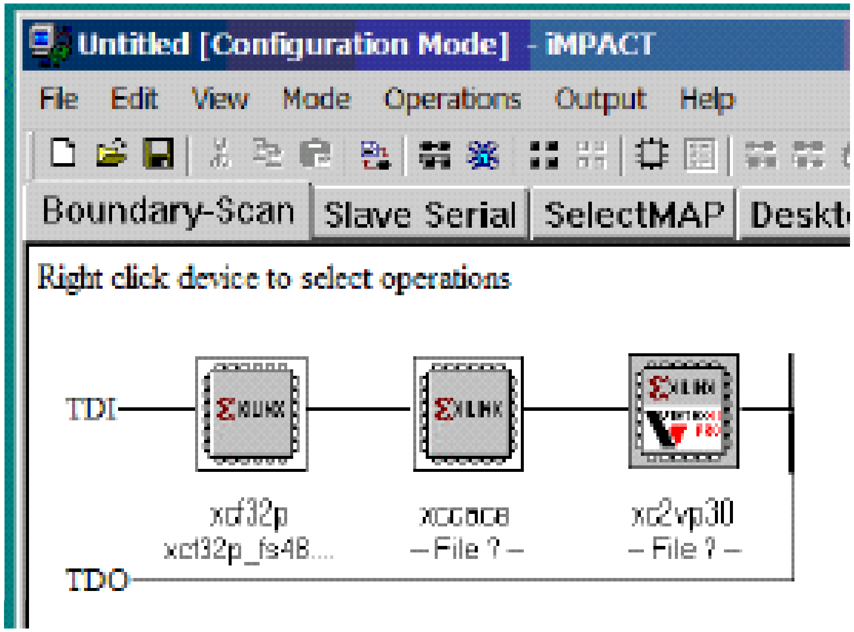


XUPV2P Development System

IIT Bombay

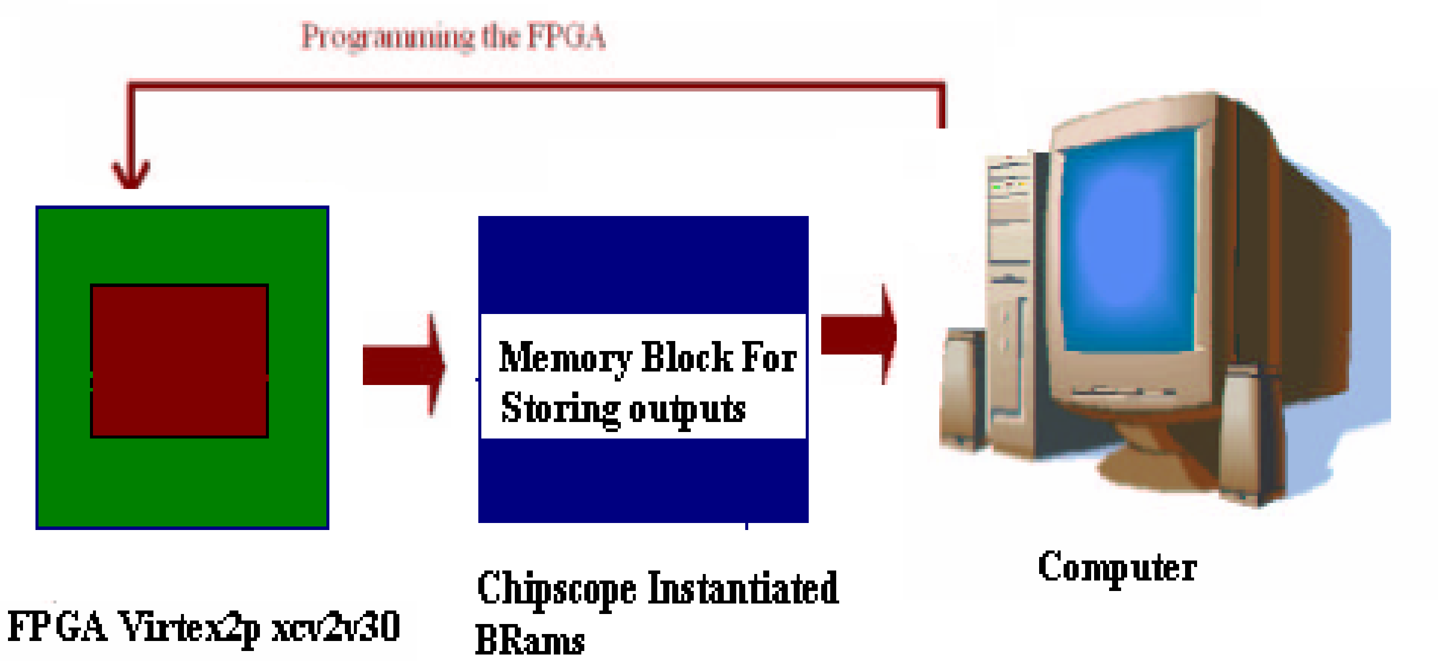


Boundary Scan



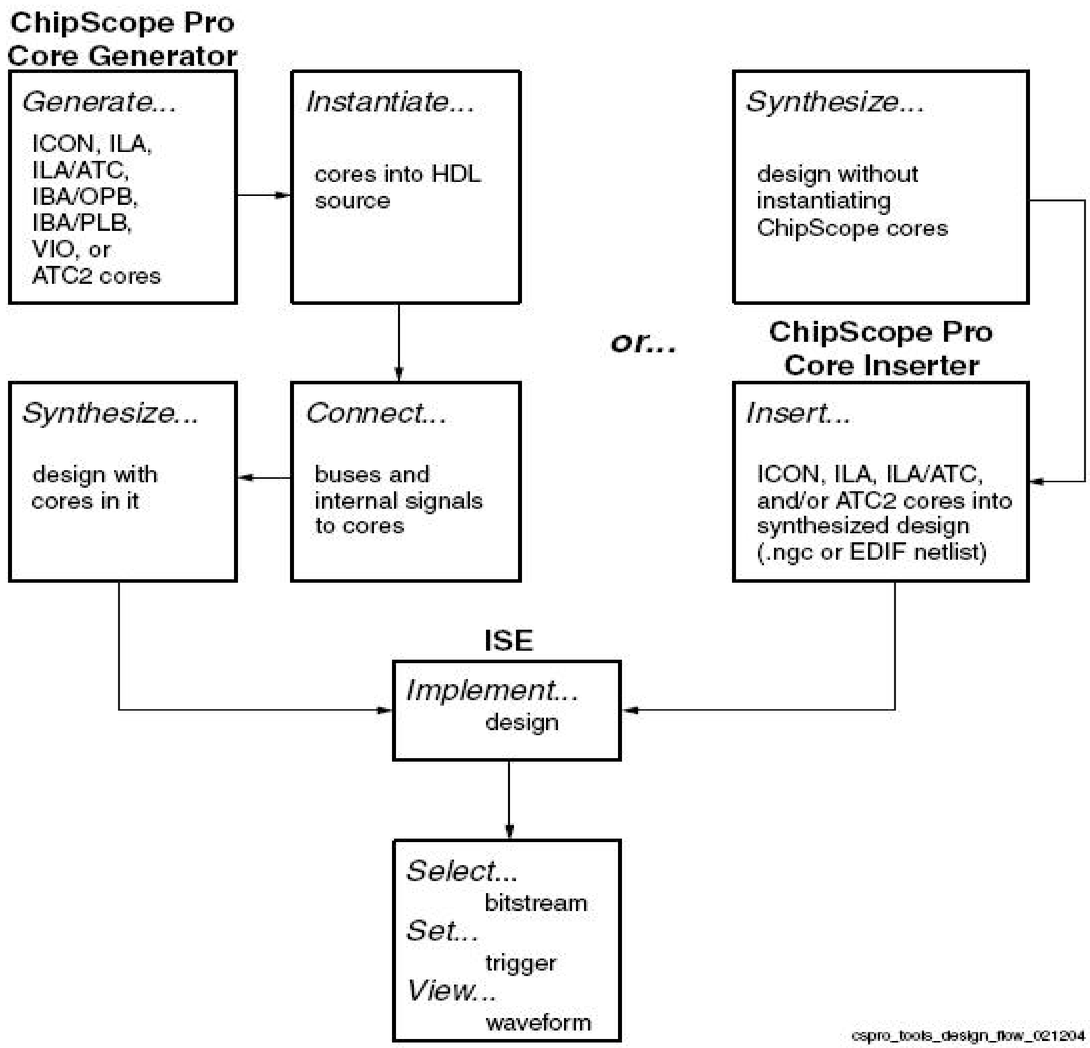
Properly Identified JTAG Configuration Chain

Hardware Setup Used



Chipscope Design Flow

IIT Bombay



Thank You

Wavelet Based Scalable Video Coding

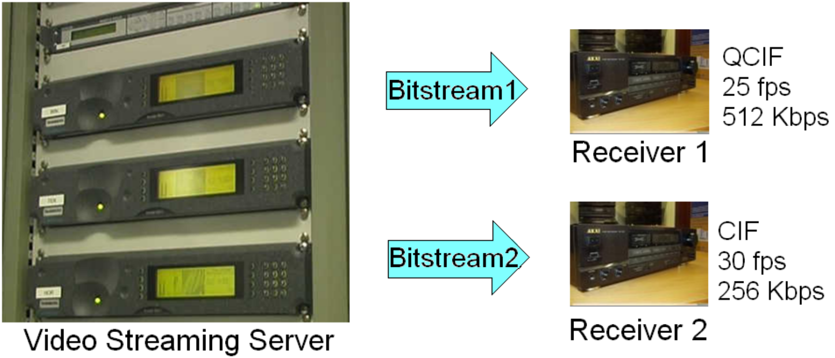
Ankur Gupta

Department of Electrical Engineering
Indian Institute of Technology, Bombay

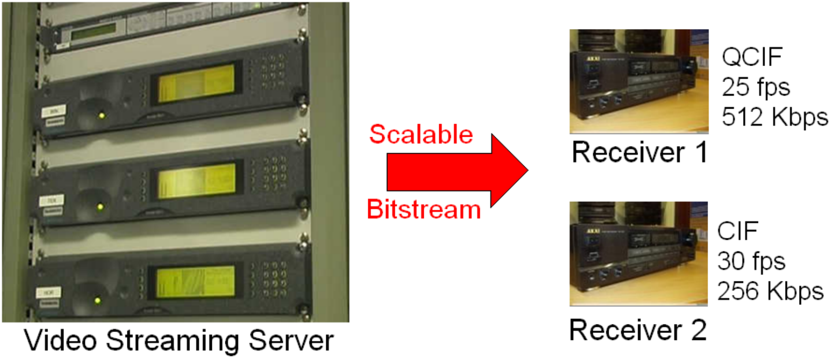
Dual Degree Presentation
3rd July, 2006

What is SVC?

SCALABLE VIDEO CODING



What is SVC?
SCALABLE VIDEO CODING



An Outline

1. 3-D SPIHT
   - SPIHT
   - Extension to 3-D SPIHT
   - Scalability of SPIHT Video Coder
2. 3D Scan based Wavelet Transform
   - Principle
   - Temporal scan based video wavelet transform
   - Experimental Results
3. Motion-Compensated Temporal Filtering
4. Resolution Scalable Coding
   - Error Feedback Hierarchical Coding
   - Experimental Results

An Outline

1. 3-D SPIHT
   - SPIHT
   - Extension to 3-D SPIHT
   - Scalability of SPIHT Video Coder
2. 3D Scan based Wavelet Transform
   - Principle
   - Temporal scan based video wavelet transform
   - Experimental Results
3. Motion-Compensated Temporal Filtering
4. Resolution Scalable Coding
   - Error Feedback Hierarchical Coding
   - Experimental Results

An Outline

1. 3-D SPIHT
   - SPIHT
   - Extension to 3-D SPIHT
   - Scalability of SPIHT Video Coder
2. 3D Scan based Wavelet Transform
   - Principle
   - Temporal scan based video wavelet transform
   - Experimental Results
3. Motion-Compensated Temporal Filtering
4. Resolution Scalable Coding
   - Error Feedback Hierarchical Coding
   - Experimental Results

An Outline

1. 3-D SPIHT
   - SPIHT
   - Extension to 3-D SPIHT
   - Scalability of SPIHT Video Coder
2. 3D Scan based Wavelet Transform
   - Principle
   - Temporal scan based video wavelet transform
   - Experimental Results
3. Motion-Compensated Temporal Filtering
4. Resolution Scalable Coding
   - Error Feedback Hierarchical Coding
   - Experimental Results

An Outline (contd..)

5. Improved Bidirectional MCTF
   - Subpixel Accuracy
   - Bidirectional MCTF
   - Experimental Results
6. Scalable Motion Coding
7. Overcomplete Motion Compensated Wavelet Coding
   - Principle
   - Wavelet-domain block matching algorithms
   - Experimental Results
8. Summary

An Outline (contd..)

5. Improved Bidirectional MCTF
   - Subpixel Accuracy
   - Bidirectional MCTF
   - Experimental Results
6. Scalable Motion Coding
7. Overcomplete Motion Compensated Wavelet Coding
   - Principle
   - Wavelet-domain block matching algorithms
   - Experimental Results
8. Summary

An Outline (contd..)

5. Improved Bidirectional MCTF
   - Subpixel Accuracy
   - Bidirectional MCTF
   - Experimental Results
6. Scalable Motion Coding
7. Overcomplete Motion Compensated Wavelet Coding
   - Principle
   - Wavelet-domain block matching algorithms
   - Experimental Results
8. Summary

An Outline (contd..)

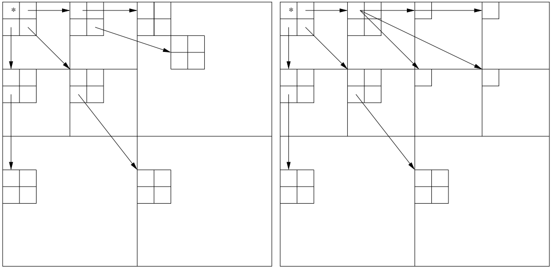
5. Improved Bidirectional MCTF
   - Subpixel Accuracy
   - Bidirectional MCTF
   - Experimental Results
6. Scalable Motion Coding
7. Overcomplete Motion Compensated Wavelet Coding
   - Principle
   - Wavelet-domain block matching algorithms
   - Experimental Results
8. Summary

Outline

SPIHT

SET PARTITIONING IN HIERARCHICAL TREES

- Introduced by Amir Said and William Pearlman in 1996

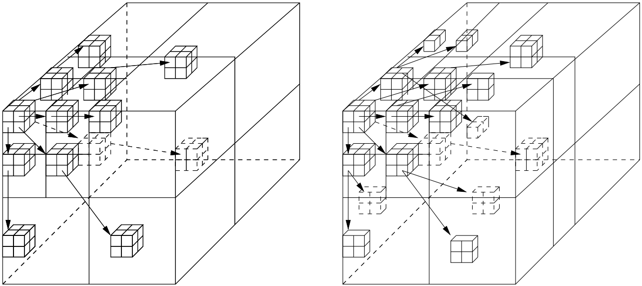


(a) (b)

Features

- Progressive Transmission
- Embedded Coding
- Partitioning coefficients for sets in spatial-orientation trees

3-D SPIHT

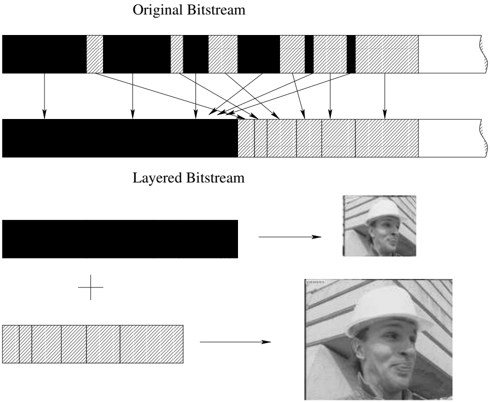


(a) (b)

- Similarly, on the 3D subband structure, 3D spatiotemporal orientation tree and its parent-child relationships are defined.

Scalable Encoding

A LAYERED STRUCTURE



Outline

Need for a scan based system

- 3D subband coding suffers from significant memory requirements
- Hence, 3D blocks are used after temporal splitting. Results in temporal blocking artifacts(flickering)
- Solution: 3D Scan Based Wavelet Transform is introduced

Principle

Frames of the sequence are acquired and processed **on the fly** to generate 3D wavelet coefficients. Data is stored in memory only until these coeffecients have been encoded.

Need for a scan based system

- 3D subband coding suffers from significant memory requirements
- Hence, 3D blocks are used after temporal splitting. Results in temporal blocking artifacts(flickering)
- Solution: 3D Scan Based Wavelet Transform is introduced

Principle

Frames of the sequence are acquired and processed **on the fly** to generate 3D wavelet coefficients. Data is stored in memory only until these coeffecients have been encoded.

Temporal scan based video WT

3D DWT = 2D DWT (spatial) + 1D DWT (temporal)

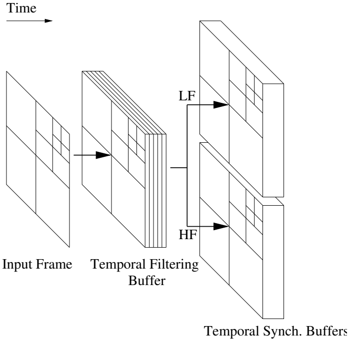


Figure: One level temporal system

The Temporal System

- $L = 2S + 1$;
  L : length of LP & HP Filters
- Receiving $S + 1^{st}$ transformed frame allows computation of first LP frame
- **HP frames** & **LP frames are obtained alternately**
- Waiting $S + 2$ frames = 1 LP & 1 HP temporal frames
- Can be extended to N-level temporal wavelet decomposition

Delay and Memory Requirements

- For a 2-level wavelet decomposition,
  Delay =$[(S+2)+(S+1)]+S$ **frames**

Delay for a N-level temporal wavelet decomposition

$D = 2^{N-1}(2S+1) - S + 1$

- Memory Requirement = Sum of frames in N filtering buffers + number of frames in the synchronization buffers
  - Sum of frames in N filtering buffers = $L \times N = (2S+1)N$
  - Number of frames in synchronization buffers = $2 + \sum_{j=2}^{N}(d_j - 1)$

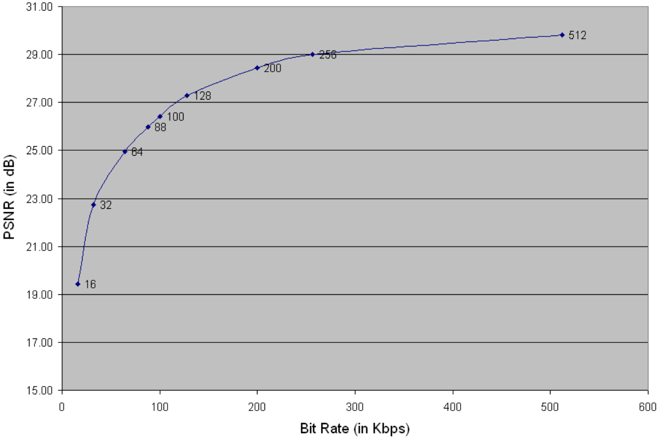
Memory requirement for a N-level temporal wavelet decomposition

$M = framesize \times [2^{N-1}(2S+1) + (N-1)S + N + 1]$

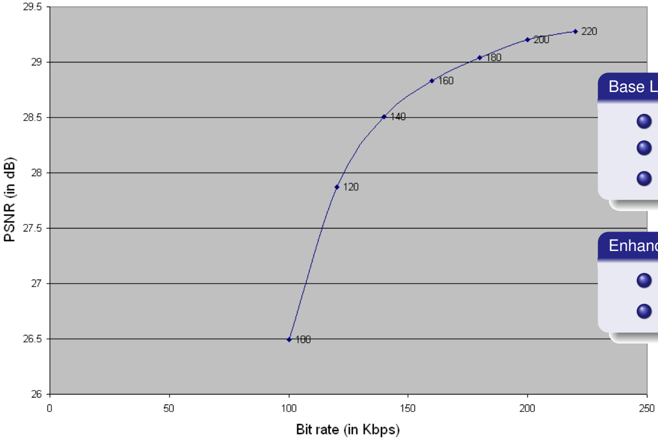
Experimental Setup

- Spatial Transform : Haar Filter
- Temporal Transform : 5/3 Filter
- Sequence : Foreman
- Frame Size : QCIF (176 x 144)
- Frame Rate : 10 fps
- Full Rate : 1980 Kbps
- Number of Frames : 100 Luminance Frames

PSNR v/s Bit Rate



Demonstrating Scalability



Base Layer

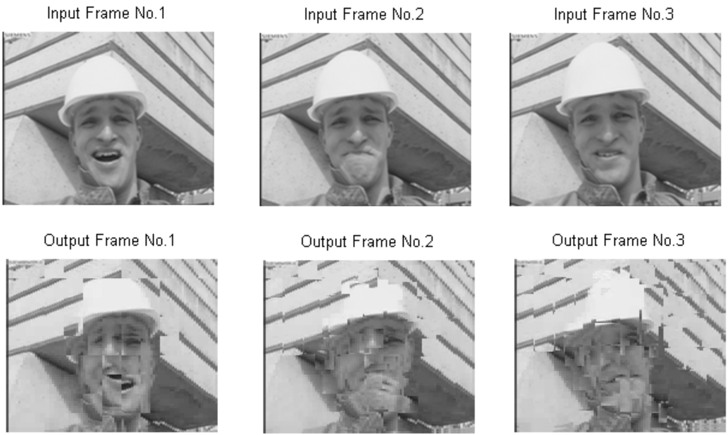
- Low Pass Frames
- 100 Kbps **(FIXED RATE)**
- 5 fps

Enhancement Layer 1

- High Pass Frames
- 0 - 120 Kbps **(VARIABLE)**

Pre-Processed Input Video

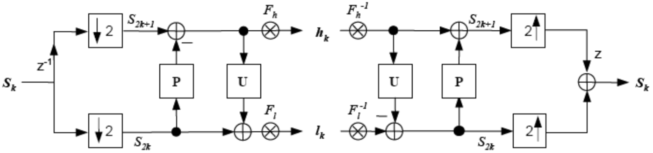
Motion Compensated Frames Result in Block Overlaps and Holes



Outline

Generic Lifting Scheme

POLYPHASE DECOMPOSITION, PREDICTION AND UPDATE



(a)
Lifting Scheme
(Analysis Filterbank)

(b)
Inverse Lifting Scheme
(Synthesis Filterbank)

- The high-pass (prediction residual) pictures are obtained by:

$$h[k] = s[2k+1] - P(s[2k]) \tag{1}$$

- The low-pass pictures are obtained by:

$$l[k] = s[2k] + U(h[k]) \tag{2}$$

MCTF for Video Coding

Notation

$s[l, k]$ is a video sample at spatial location $l = (x, y)$ at time instant k

The prediction and update operators for temporal decomposition using:

- Haar Wavelet

$$P_{Haar}(s[l, 2k]) = s[l + m_{P0}, 2k - 2r_{P0}] \tag{3}$$

$$U_{Haar}(h[l, k]) = \frac{1}{2} h[l + m_{U0}, k + r_{U0}] \tag{4}$$

- 5/3 Wavelet

$$P_{5/3}(s[l, 2k]) = \frac{1}{2}(s[l + m_{P0}, 2k - 2r_{P0}] + s[l + m_{P1}, 2k + 2 + 2r_{P1}] \tag{5}$$

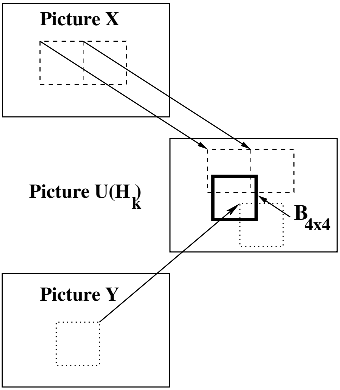
$$U_{5/3}(h[l, k]) = \frac{1}{4}(h[l + m_{U0}, k + r_{U0}] + h[l + m_{U1}, k - 1 - r_{U1}] \tag{6}$$

The prediction steps in both the cases exactly correspond to the predictive coding

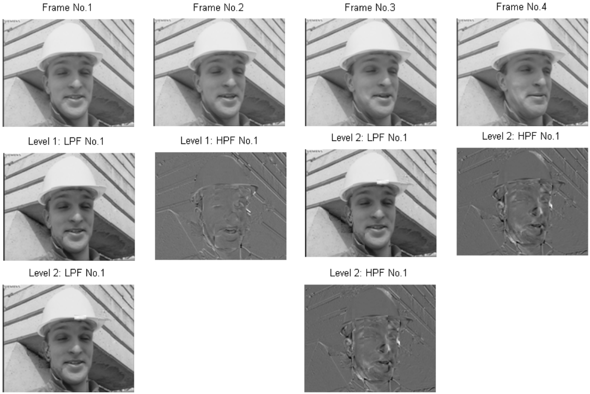
Derivation of Update Operator

For each 4x4 luma block B_{4x4} in the picture $U(H_k)$, m_{U0}, r_{U0}, m_{U1} and r_{U1} are derived as follows

- Evaluate all m_{P0} and m_{P1} that point into B_{4x4}
- Select those m_{P0} and m_{P1} that use maximum number of samples for reference out of B_{4x4}
- Set $m_{U0} = -m_{P0}$ and $m_{U1} = -m_{P1}$
- Set r_{U0} and r_{U1} to point to those picturesinto which MC is conducted using m_{P0} and m_{P1}, respectively

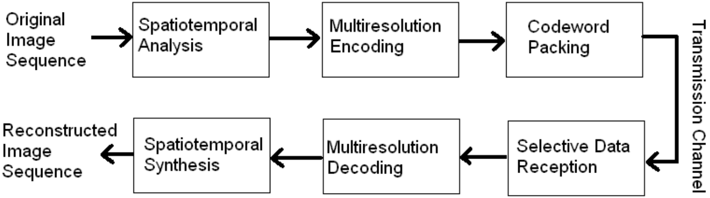


Temporal Decomposition Structure



Outline

RSC Block Diagram



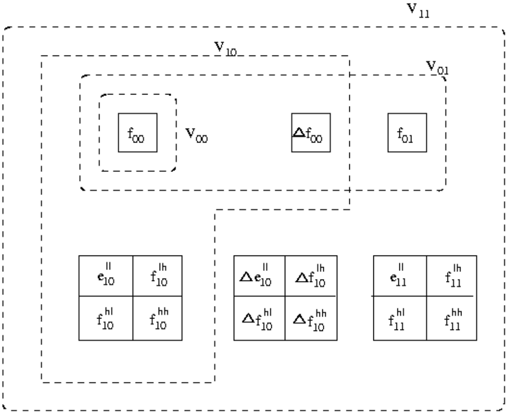
- Lower resolution and frame-rate videos are generated by a 3D subband filter bank
- Spatial Analysis followed by Temporal Analysis
- Temporal Subband Analysis: two-tap MCTF
- Spatial Subband Analysis : Haar, 5/3 filters, etc

Spatiotemporal Multiresolution Analysis/Synthesis

Notation

- Implemented for specific case of 6 different spatiotemporal resolutions:
  - 3 levels of spatial resolution
  - 2 different frame rates
- V_{ij}: Output video, where
  $i \in 0, 1, 2$ represents 3 spatial resolutions,
  $j \in 0, 1$ represents two temporal resolutions
- V_{i0}: Low frame rate video at i^{th} spatial resolution

Error Feedback Hierarchical Coding

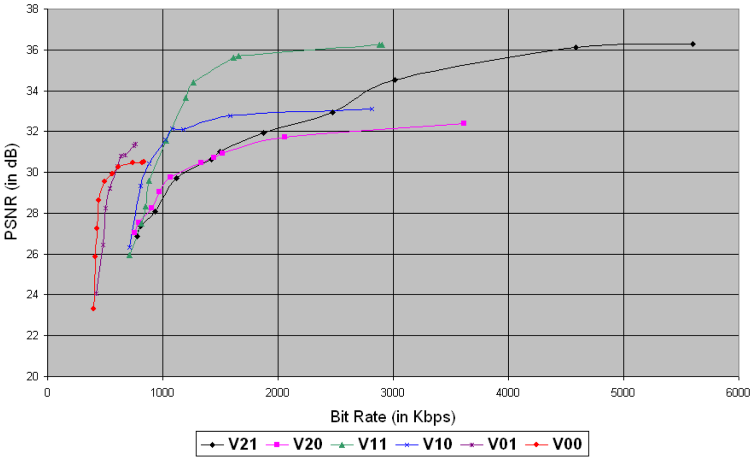


KEY

- *e* indicates residual after prediction
- Δf indicates a refinement due to cascaded quantization of corresponding low band

Experimental Setup

- Spatial Transform : Daubechies 4
- Temporal Transform : MCTF
- Sequence : Foreman
- Input Frame Size : CIF (352 x 288)
- Input Frame Rate : 30 fps
- Full Rate : 23760 Kbps
- Number of Frames : 80 Luminance Frames
- Output Frame Rates : 30 fps, 15 fps
- Output Frame Resolutions : CIF (352 x 288), QCIF (176 x 144) and Q-QCIF (88 x 72)


38
36
34
32
30
28
26
24
22
20
PSNR (in dB)
0
1000
2000
3000
4000
5000
6000
Bit Rate (in Kbps)
V21
V20
V11
V10
V01
V00

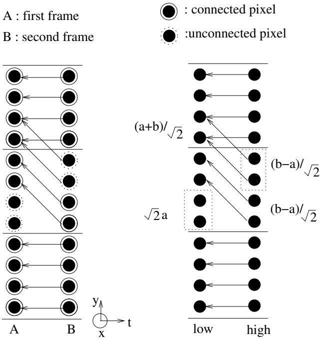
Outline

Problems with existing MCTF

- Filtering across poorly matched pixels decreases coding efficiency in MCTF
- Absence of subpixel accuracy while performing MCTF
- Bidirectional MCTF should perform better as compared to unidirectional MCTF

Pixels are Connected Pixels if

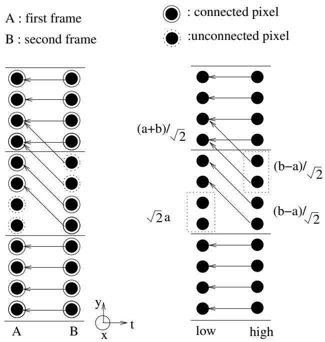
- There is a one-to-one connection between the pixels
- If several pixels in frame2 connect to the same pixel in frame1, only one of them is a connected pixel
- Unconnected pixels in frame1 are not used as reference for frame2



- MCTF is performed only on connected pixels. Unconnected pixels are not filtered
- For unconnected pixels in frame1, their scaled original values are inserted into temporal low subband
- For unconnected pixels in frame2, the scaled displaced frame differences are inserted into temporal high subband

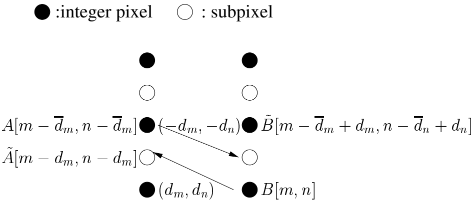
Pixels are Connected Pixels if

- There is a one-to-one connection between the pixels
- If several pixels in frame2 connect to the same pixel in frame1, only one of them is a connected pixel
- Unconnected pixels in frame1 are not used as reference for frame2



- MCTF is performed only on connected pixels. Unconnected pixels are not filtered
- For unconnected pixels in frame1, their scaled original values are inserted into temporal low subband
- For unconnected pixels in frame2, the scaled displaced frame differences are inserted into temporal high subband

Subpixel Accurate MCTF using Lifting Implementation



If MV's have subpixel accuracy, the lifting scheme calculates:

- Temoral High-Frame as

$$H[m,n] = \frac{1}{\sqrt{2}} I_{2t+1}[m,n] - \frac{1}{\sqrt{2}} \tilde{I}_{2t}[m - d_m, n - d_n] \tag{7}$$

- Temporal Low-Frame as

$$L[m-\bar{d}_m, n-\bar{d}_n] = \tilde{H}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n] + \sqrt{2} I_{2t}[m-\bar{d}_m, n-\bar{d}_n] \tag{8}$$

Bidirectional MCTF

THE ALGORITHM

- Two kinds of blocks - *Connected* and *Unconnected* Blocks
- For connected blocks, high-pass and low-pass coefficients are computed using Equation (7) and (8)
- For unconnected blocks, based on the SAD of spatial interpolation and the sum of absolute DFDs of forward and backward MCP, we get :
  - P block using frame one $(2t)$

$$H[m,n] = \frac{1}{\sqrt{2}} I_{2t+1}[m,n] - \frac{1}{\sqrt{2}} \tilde{I}_{2t}[m-d_m, n-d_n] \qquad (9)$$

  - P block using frame three $(2t+2)$

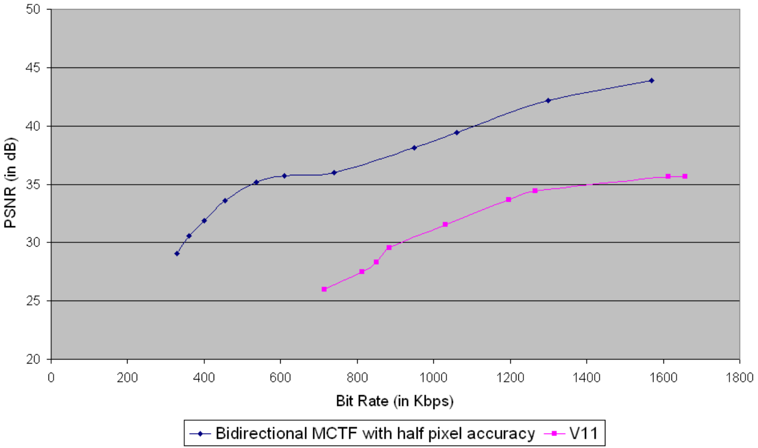
$$H[m,n] = \frac{1}{\sqrt{2}} I_{2t+1}[m,n] - \frac{1}{\sqrt{2}} \tilde{I}_{2t+2}[m-d_m, n-d_n] \qquad (10)$$

  - I block

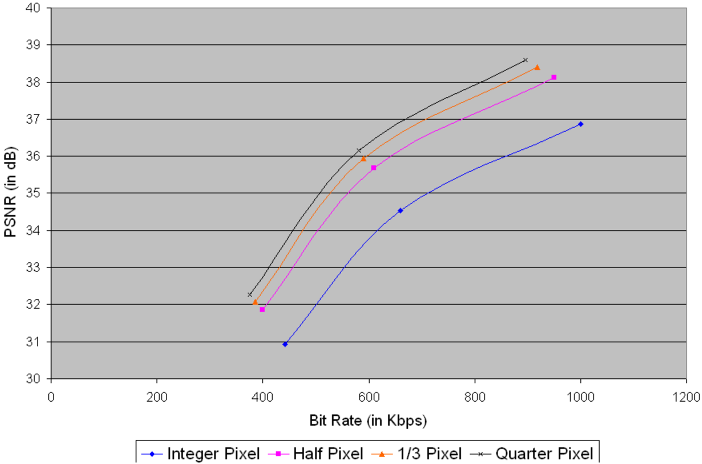
Experimental Setup

- Sequence : Foreman
- Input Frame Size : QCIF (176 x 144)
- Input Frame Rate : 30 fps
- Number of Frames : 50 Luminance Frames
- GOP Size : 16 frames

Bidirectional MCTF Comparison



Subpixel Accuracy Comparison



Outline

Introduction

- Proposed by Taubman and Secker in 2004
- Traditionally, motion parameters were coded losslessly due to non-linear interaction between motion and sample data
- But, this relationship turns out to be linear for all "optimal" combinations of motion and sample bit rates
- Therefore, motion and sample data can be coded independently of each other

Experimental Setup

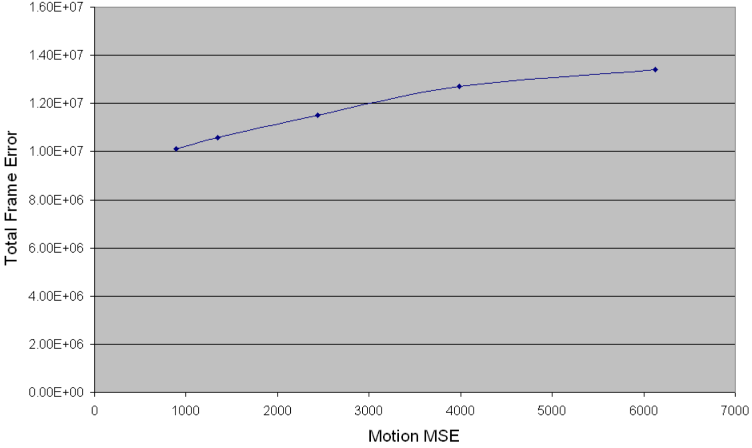
Aim:

1. To verify the linear relationship between Motion MSE and Total Frame Error
2. To see how the PSNR varies as we increase the Motion bits

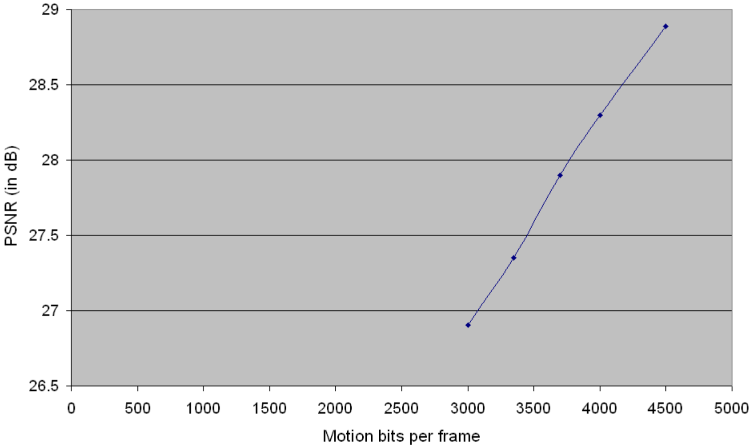
The *Scalable Motion Coding* model was applied to the highest spatial resolution video of the *RSC model*

- Sequence : Foreman
- Input Frame Size : CIF (352 x 288)
- Input Frame Rate : 30 fps
- Number of Frames : 80 Luminance Frames

Total Frame Error v/s Motion MSE



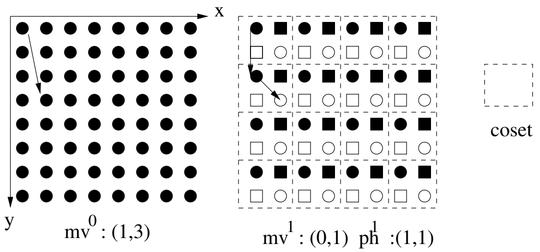
PSNR v/s Motion bits per frame



Outline

Principle

- For a 1-D sequence $x(n)$, either even-indexed or odd-indexed coefficients are sufficient for perfect reconstruction
- What about $y(n)$, where $y(n) = x(n + \tau)$?
- Reason: Frequency aliasing brought by decimation *OR* motion accuracy sacrifice



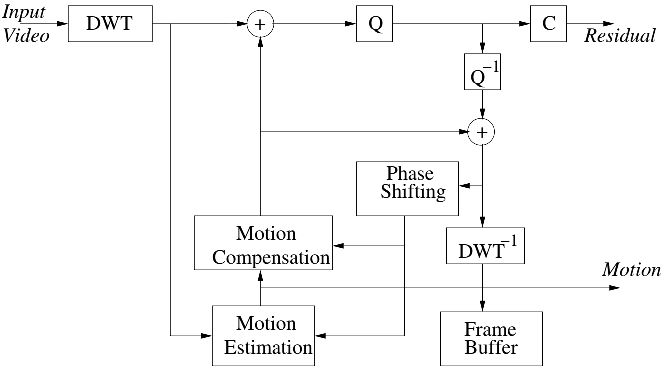
(a) (b)

OMCP Coder

OVERCOMPLETE MOTION COMPENSATED PREDICTION

Key Difference

Both ME and MC are performed between maximally-decimated wavelet decomposition of current frame and the overcomplete expansion of previous frame



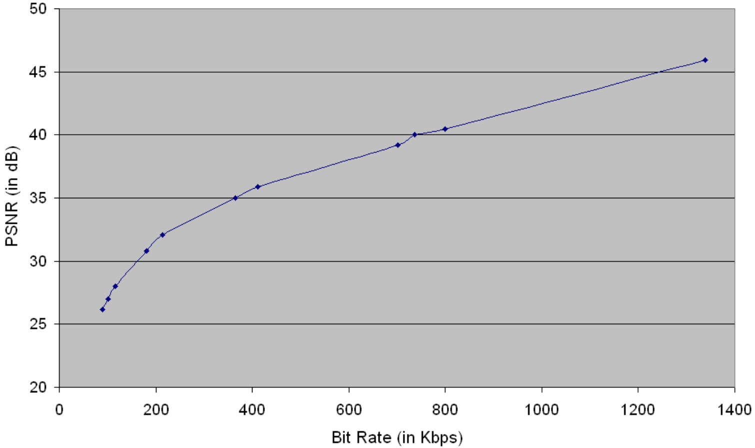
Wavelet-domain BMA v/s Spatial-domain BMA

- BMA is based on nonlinear criterion such as SAD
- W-BMA is less affected by occlusion because pixels in covered and uncovered areas are decorelated by spatial filtering first.
- S-BMA is highly sensitive to photmetric distortion.
  W-BMA alleviates this problem due to its frequency selectivity (high-band coefficients used for block matching would be less affected than low-band ones)

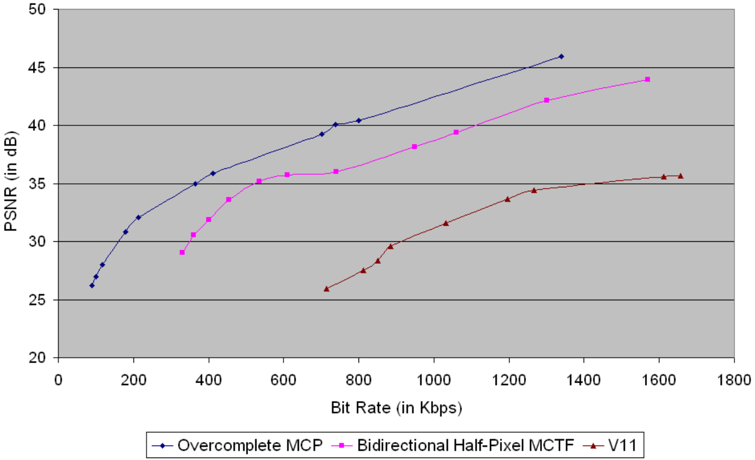
Experimental Setup

- Sequence : Foreman
- Input Frame Size : QCIF (176 x 144)
- Input Frame Rate : 30 fps
- Number of Frames : 80 Luminance Frames
- GOP Size : 16 frames

PSNR Variations



Comparison


50
45
40
35
30
25
20
PSNR (in dB)
0
200
400
600
800
1000
1200
1400
1600
1800
Bit Rate (in Kbps)
Overcomplete MCP
Bidirectional Half-Pixel MCTF
V11

Outline

Summary

We conclude that

- 3D Scan based WT helps in solving the problem of temporal artifacts and also greater complexity of 3D SPIHT
- 3D scan based WT face problems when grouped with motion compensation, making MCTF a necessity
- Bidirectional MCTF with subpixel accuracy offers a gain of nearly 7dB over unidirectional MCTF
- Overcomplete MCP increases the PSNR further by approximately 3dB
- Scalable Motion Coding also appears promising, specially at lower bitrates.

Summary

We conclude that

- 3D Scan based WT helps in solving the problem of temporal artifacts and also greater complexity of 3D SPIHT
- 3D scan based WT face problems when grouped with motion compensation, making MCTF a necessity
- Bidirectional MCTF with subpixel accuracy offers a gain of nearly 7dB over unidirectional MCTF
- Overcomplete MCP increases the PSNR further by approximately 3dB
- Scalable Motion Coding also appears promising, specially at lower bitrates.

Summary

We conclude that

- 3D Scan based WT helps in solving the problem of temporal artifacts and also greater complexity of 3D SPIHT
- 3D scan based WT face problems when grouped with motion compensation, making MCTF a necessity
- Bidirectional MCTF with subpixel accuracy offers a gain of nearly 7dB over unidirectional MCTF
- Overcomplete MCP increases the PSNR further by approximately 3dB
- Scalable Motion Coding also appears promising, specially at lower bitrates.

Summary

We conclude that

- 3D Scan based WT helps in solving the problem of temporal artifacts and also greater complexity of 3D SPIHT
- 3D scan based WT face problems when grouped with motion compensation, making MCTF a necessity
- Bidirectional MCTF with subpixel accuracy offers a gain of nearly 7dB over unidirectional MCTF
- Overcomplete MCP increases the PSNR further by approximately 3dB
- Scalable Motion Coding also appears promising, specially at lower bitrates.

Summary

We conclude that

- 3D Scan based WT helps in solving the problem of temporal artifacts and also greater complexity of 3D SPIHT
- 3D scan based WT face problems when grouped with motion compensation, making MCTF a necessity
- Bidirectional MCTF with subpixel accuracy offers a gain of nearly 7dB over unidirectional MCTF
- Overcomplete MCP increases the PSNR further by approximately 3dB
- Scalable Motion Coding also appears promising, specially at lower bitrates.

Future Work

- The unified scalable coder should be compared for performance to the Scalable Video Codec (SVC) being developed by JVT
- Another area of work is the rate control block for the scalable codecs. One of the possible algorithms that could be used is the Multidimensional Bit-Rate Control