Advanced Topics Multimedia Video (5LSH0), Module 01

Introduction to (01A) Wavelet Coding, and (01B) JPEG2000

Peter H.N. de With & Guido T.G. Volleberg

(p.h.n.de.with@tue.nl)

Overview

* **A1: Wavelet transform**
  – Introduction and Fourier analysis
  – Subband filters, Quadrature Mirror Filters
  – Wavelets
* **A2: Wavelet video coding**
  – Wavelet coefficient coding for compression
  – Intraframe and interframe coding
* **B: JPEG2000 standard**
  – Standard principles
  – Special coding modes, forms of scalability

Module 01A (part 1): Introduction to Wavelet Theory

Fourier analysis, STFT, Wavelet and Scaling functions, Filter banks

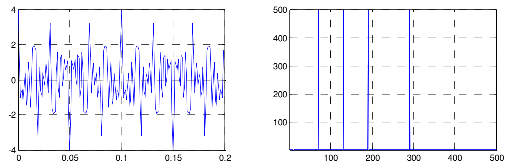
Introduction / Random signal

* **A given (random) signal:**
  – Consists of low and high frequencies
  – These frequencies vary in amplitude
  – These frequencies may exist for a defined duration, so may exist only for a short amount of time or say at a given moment/location
  – Generally assumed the signal can be decomposed into a set of ***basis functions***

Introduction / Signal example

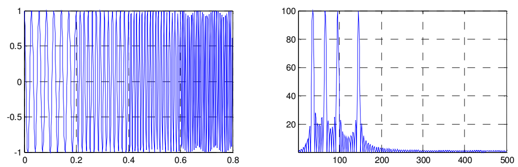
Stationary signal:

```
x =cos(2*pi*70*t)+cos(2*pi*130*t)+cos(2*pi*190*t)+cos(2*pi*290*t)
```



Non-stationary signal:

```
x() = cos(2*pi* 35*t(  1:200))+ cos(2*pi* 65*t(201:400))+
   cos(2*pi* 95*t(401:600))+ x(601:800)+cos(2*pi*145*t(601:800));
```



Introduction / random signal

- **For signal analysis one would like a tool, which provides these properties:**
  - Frequency
  - Amplitude
  - Location
- **Motivation: analysis components can be used as input for (a.o.) compression systems.**
  - Which "tools" suffice?

Fourier transform

Fourier transform based on a signal decomposition of sine/cosine waves:

$$\mathrm{F}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

- **It has:**
  - Good sine wave detection/recognition
  - No time-localization property
  - No transient detection
- **Good tool for frequency analysis where time-localization is not important.**

Short-Time Fourier Transform (STFT) –(1)

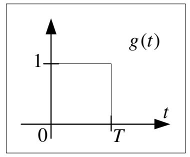
- **Because: if $f(t)$ is a non-periodic signal, the summation of periodic functions, sine & cosine (as with Fourier), does not accurately represent the signal.**
- **STFT: decompose the signal $f(t)$ into pieces to achieve time-localization by means of a "window"-function**

(a.k.a. Windowed Fourier Transform)

Short-Time Fourier Transform – (2)

9

* **Short-Time:**
  - Definition: $F(\omega,\tau)=\int_{-\infty}^{\infty} f(t)g(t-\tau)e^{-j\omega t}dt$
  - Where *g(t)* is a rectangular window, *g(t)*=1 for *t*=<0,*T*] and *g(t)*=0 otherwise



  - Sometimes other window functions are used e.g. if *g(t)* is Gaussian, the STFT is the Gabor transform

Short-Time Fourier Transform – (3)

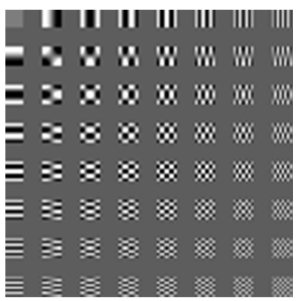
10

* **Properties:**
  - Able to analyze frequencies
  - Able to do time-localization but…
    - The localization window is fixed! Hereby not scaling along with the (low-high) frequencies.
      - ➢ High frequencies are (very) location precise
      - ➢ Low frequencies are location in-precise

Short-Time Fourier Transform – (4)

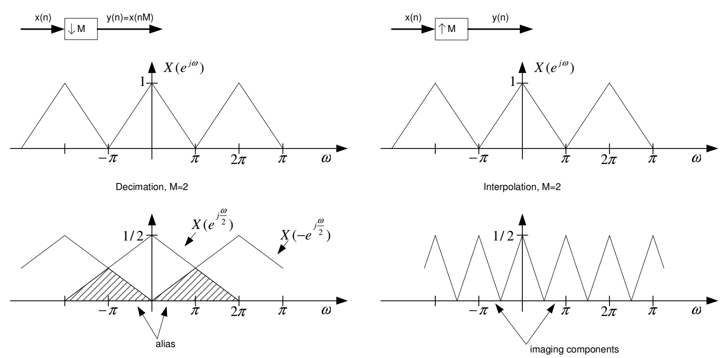
11

* **Image & video coding is based on DCT**
  - 8x8 video block sampling
* **2D DCT decomposition**
  - for 8x8 blocks
  - Figure shows basis functions
  - Patterns for projections
* **However ,…**
  - Block aliasing occurs



Subband coding – (1) / Fundamentals of multirate theory

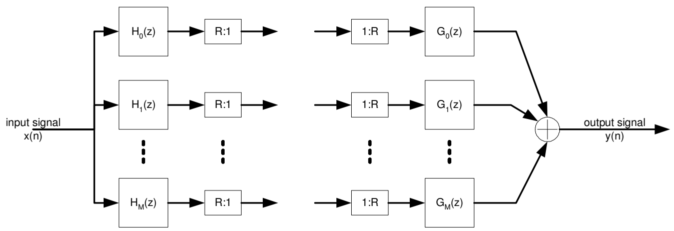
12



Subband coding – (2) / M-subband Analysis/Synth. System

13

- Analysis, low-/highpass filter pairs $H_0, H_1, \ldots, H_M$
- Synthesis, low-/highpass filter pairs $G_0, G_1, \ldots, G_M$



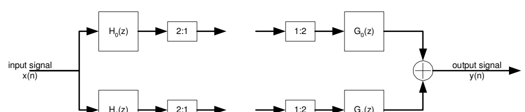
Subband coding theory – (3) / Subband Analysis/Synthesis Filters

14

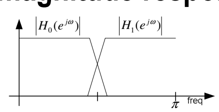
- Design of such band filters requires great care, they need to eliminate:
  - Aliasing, which adds to (visual) distortion (due to decimation)
  - Possible spectral magnitude & spectral phase distortion
  - Constraints exist on total full-chain response, bandwidth suppression in certain intervals, etc.

Subband coding theory – (4) / One-dimensional filter pair (2-band)

15



- Analysis filter magnitude responses



- Transfer function

$$Y(z)=\frac{1}{2}[H_0(z)\cdot G_0(z)+H_1(z)\cdot G_1(z)]\cdot X(z)+\frac{1}{2}[H_0(-z)\cdot G_0(z)+H_1(-z)\cdot G_1(z)]\cdot X(-z)$$

Subband coding theory – (5) / Design Quadrature Mirror Filters (QMF)

16

- Biorthogonality principle for perfect reconstruction

$$Y(z)=\frac{1}{2}[H_0(z)\cdot G_0(z)+H_1(z)\cdot G_1(z)]\cdot X(z)+\frac{1}{2}[H_0(-z)\cdot G_0(z)+H_1(-z)\cdot G_1(z)]\cdot X(-z)$$

- Required: $[H_0(z)\cdot G_0(z)+H_1(z)\cdot G_1(z)]=2\cdot z^{-k}$
- Alias: $[H_0(-z)\cdot G_0(z)+H_1(-z)\cdot G_1(z)]=0$
- When: $G_0(z)=z^k\cdot H_1(-z)$ and $G_1(z)=-z^k\cdot H_0(-z)$

Subband coding theory – (6) Quadrature Mirror Filters (QMF)

* H_0 with H_1 and G_0 with G_1 are **orthogonal** pairs

  since $H_1(z) = H_0(-z)$ therefore $h_1(n) = (-1)^k \cdot h_0(n)$

* For filter coefficients, it holds (HP mirror of LP):

  i.e $H_1(e^{j\omega}) = H_0(e^{j(\pi-\omega)})$

  substituting $\omega = \frac{\pi}{2} - \omega$

  hence $H_1(e^{j(\frac{\pi}{2}-\omega)}) = H_0(e^{j(\frac{\pi}{2}+\omega)})$

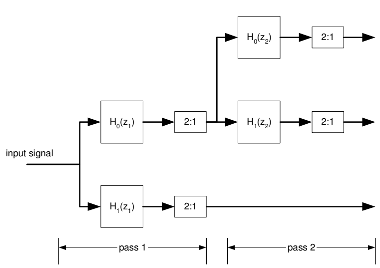
* Shows the **mirror image** property around $\pi/2$, hence H_0 and H_1 are called **QMFs**

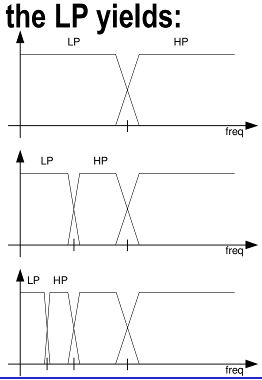
Subband coding theory – (7) / Quadrature Mirror Filters (QMF)

* Remarks:
  - QMFs can also be designed in a ***M*-band** signal split setup (could of course also be done recursively with the 2-band split).
  - QMFs provide **perfect reconstruction** (allowing for a given time-delay) in the filter-bank.
  - However they are **not naturally good frequency cut-off** filters, which has an impact on usability for compression or other signal analysis.

Subband coding theory – (8) / Repeated decomposition

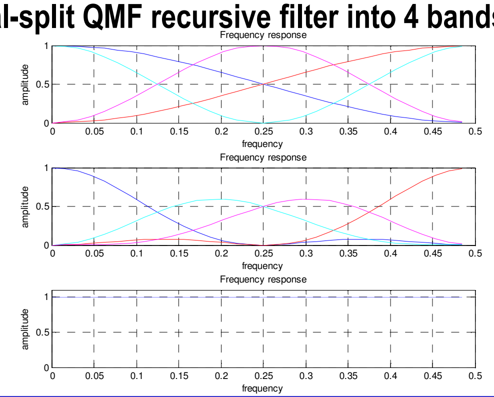
* Recursively applied QMF on the LP yields:



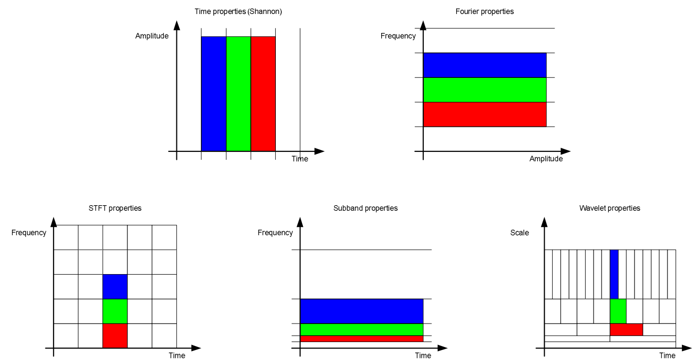


Subband coding theory – (9) / Case

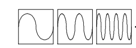
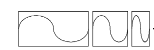
* Dual-split QMF recursive filter into 4 bands



Overview of time-frequency for different decompositions



Wavelets / Introduction

- With the STFT, the time window remains equal while other cycles (i.e. frequency) of the *basis function* (sine/cosine) existed in that window 
- With a *wavelet basis function*, the window size changes while the #cycles remains equal 
- Cycle and location detection are done by "scaling" and "translating" the *basis function* / *mother wavelet*

Wavelets / Mother wavelet

- 'Mother' maintains its properties, derive the remaining scaled and translated wavelet functions ("psi") by:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$$

where ***a*** represents **scaling** and ***b*** represents **translation**

- **Scaling** $\frac{1}{\sqrt{a}}$ is needed to maintain "norm"

$$\|f(t)\|^2 = \int_{-\infty}^{\infty} f^2(t)dt$$

Wavelets / Continuous functions

- Defining Continuous Wavelet Transform (CWT)

$$y(a,b) = \int_{-\infty}^{\infty} x(t)\cdot\psi_{a,b}(t)dt$$

- and its inverse

$$x(t) = \frac{1}{C_\psi}\int_{-\infty}^{\infty}\int_{0}^{\infty}\frac{1}{a^2}\, y(a,b)\,\psi_{a,b}(t)\,da\,db$$

where $C_\psi = \int_0^{\infty}\frac{|\Psi(\omega)|^2}{\omega}d\omega$ and $\Psi(\omega) = F[\psi(t)]$

as its Fourier transform

Wavelets / Energy constraint

- Needed $\Psi(0)=0$ for the integral to exist: $\int_{-\infty}^{\infty}\psi(t)dt=0$
  - Hence the wavelet function is a "specific" HP-filter, avg = 0
- And we would like finite energy (via Parseval)

  $\int_{-\infty}^{\infty}|\Psi(\omega)|^2 d\omega<\infty$ this happens if $|\Psi(\omega)|^2$ decays

  when ω goes to ∞
  - Energy now in narrow frequency band: => **frequency localization**!

Wavelets / Definition of Discrete WT

- Discrete Wavelet Transform (DWT)

  let $a=a_0^{-m},\quad b=n\cdot b_0\cdot a_0^{-m}$

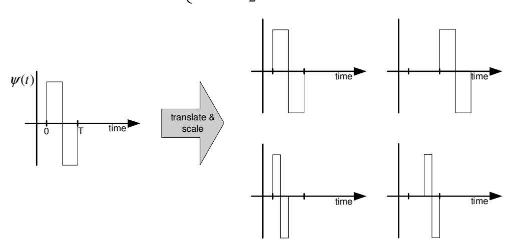
  then $\psi_{m,n}(t)=a_0^{m/2}\cdot\psi(a_0^m t-nb_0),\quad m,n\in Z$

  for $a_0=2,\quad b_0=1$, we have $\psi_{m,n}(t)=2^{m/2}\cdot\psi(2^m t-n)$
  - Dyadic wavelets: **integer time shift and power of two scaling**.
  - …these are however not the only possible DWT.

Wavelets / Example of shifting, scaling

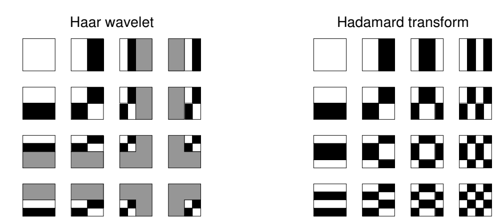
- Example: the Haar wavelet

$$\psi(t)=\begin{cases}1 & 0\le t<\frac{1}{2}\\ -1 & \frac{1}{2}\le t<1\end{cases}$$



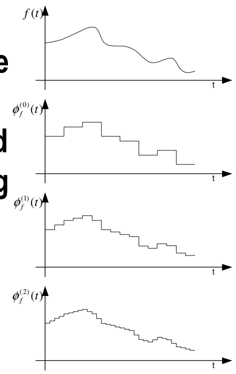
Wavelets / Compare Haar & Hadamard

- Haar wavelet versus the Hadamard transform
  - 2D decomposition (2x1D)



Wavelets / Scaling function

- **Scaling function** $\phi(t)$ has the property that function $f(t)$ can be represented by the scaling function, but also be represented by **dilated versions** of the scaling function ("phi")
- Scaling function acts as the **LP counterpart** of the wavelet
- Hence: composition is phi+psi!



Wavelets / Multi Resolution Analysis (MRA) – (1)

- Scaling function can also be represented by its dilations at a higher resolution (special case with half-band filters)

$$\phi(t) = \sum_k h_k \sqrt{2}\phi(2t-k)$$

- A.k.a. **dilation equation**

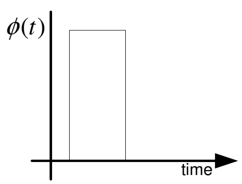
TU/e PdW-AP-EB-FS-FG-SZ / 2017 Fac. EE SPS-VCA Adv. Topics MMedia Vid. Cod. / 5LSH0 / Module 01A Wavelets VCA

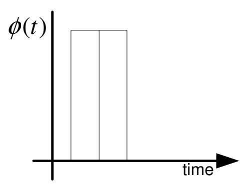
Wavelets / Multi Resolution Analysis (MRA) – (2)

$$\phi(t) = \sum_k h_k \sqrt{2}\phi(2t-k)$$

- Does it hold for the Haar scaling function?

$$\phi(t) = \begin{cases} \frac{1}{\sqrt{2}} & 0 \le t < 1 \\ 0 & otherwise \end{cases} \qquad h_0 = h_1 = \frac{1}{\sqrt{2}}, \quad h_k = 0 \quad for\ k > 1$$





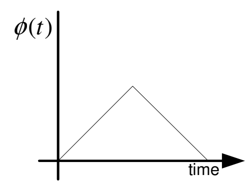
TU/e PdW-AP-EB-FS-FG-SZ / 2017 Fac. EE SPS-VCA Adv. Topics MMedia Vid. Cod. / 5LSH0 / Module 01A Wavelets VCA

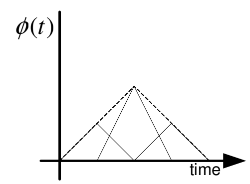
Wavelets / Multi Resolution Analysis (MRA) – (3)

$$\phi(t) = \sum_k h_k \sqrt{2}\phi(2t-k)$$

- Does it hold for the Triangular scaling function?

$$h_0 = \frac{1}{2\sqrt{2}}, h_1 = \frac{1}{\sqrt{2}}, h_2 = \frac{1}{2\sqrt{2}}$$





TU/e PdW-AP-EB-FS-FG-SZ / 2017 Fac. EE SPS-VCA Adv. Topics MMedia Vid. Cod. / 5LSH0 / Module 01A Wavelets VCA

Wavelets / Scaling function, examples – (1)

- **Given rules:** $\sum_k h_k = \sqrt{2}$ **and** $\sum_k h_k^2 = 1$

  (normalisation & orthogonality)

- **Suppose** k **=2,** $h_0 + h_1 = \sqrt{2}$

  $h_0^2 + h_1^2 = 1$

- **Results in the Haar scaling function**

$$h_0 = h_1 = \frac{1}{\sqrt{2}}$$

Wavelets / Scaling function, examples – (2)

- **Given rules:** $\sum_k h_k = \sqrt{2}$, $\sum_k h_k^2 = 1$ **and** $\sum_k h_k h_{k-2m} = \delta_m$

  (normalisation & orthogonality, like for perfect reconstruction)

- **Suppose** k **=3,** $h_0 + h_1 + h_2 = \sqrt{2}$

  $h_0^2 + h_1^2 + h_2^2 = 1$

  $h_0 h_2 = 0$

- **…the two-coefficient Haar scaling function!**
  - In case of an odd number of coefficients, 1 coefficient will be forced zero. Thus: always an **even number** of coefficients.

Wavelets / Scaling function, examples – (3)

- **Given rules:** $\sum_k h_k = \sqrt{2}$, $\sum_k h_k^2 = 1$ **and** $\sum_k h_k h_{k-2m} = \delta_m$

  (normalisation & orthogonality, like for perfect reconstruction)

- **Suppose** k **=4,** $h_0 + h_1 + h_2 + h_3 = \sqrt{2}$

  $h_0^2 + h_1^2 + h_2^2 + h_3^2 = 1$

  $h_0 h_2 + h_1 h_3 = 0$

- **Results: more possible solutions (Daubechies 4)…**

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

Wavelets / Scaling vs. Wavelet function – (1)

- **The wavelet function is orthogonal to the scaling function** $\int \phi(t-k)\psi(t-m)dt = 0$

then $w_k = \pm(-1)^k h_{N-k}$

and $\sum_k h_k w_{n-2k} = 0$

Furthermore $\sum_k w_k = 0$

Wavelets / Scaling vs. Wavelet function – (2)

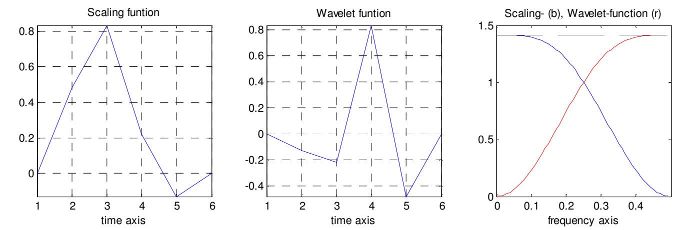
37

* The **scaling function** acts as a **low-pass** (LP) filter

* The **wavelet function** acts as a **high-pass** (HP) filter

Wavelets / Daubechies 4 – (1)

38

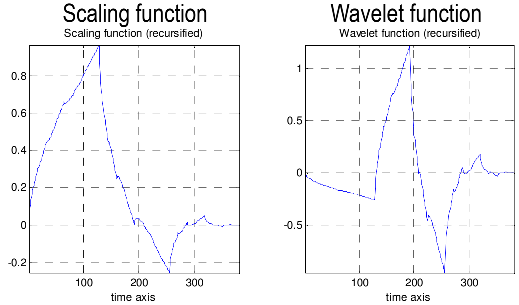


Scaling function · Wavelet function · Joint freq. response

Wavelets / Daubechies 4 – (2)

39

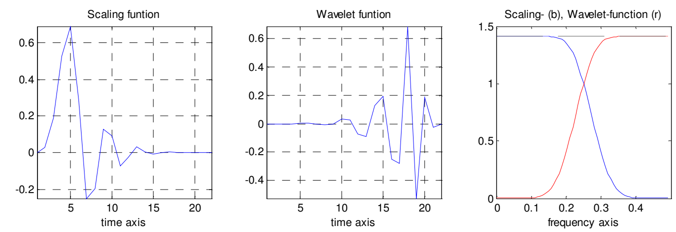
Scaling function · Wavelet function



* ...notice the recursive (fractal-like) nature

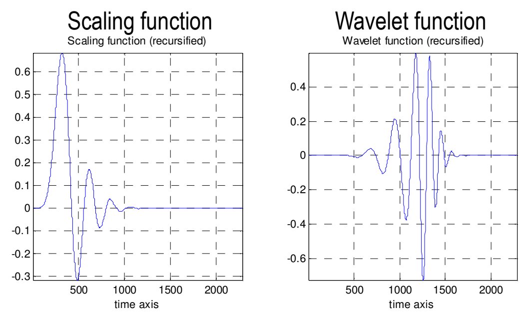
Wavelets / Daubechies 20 – (1)

40

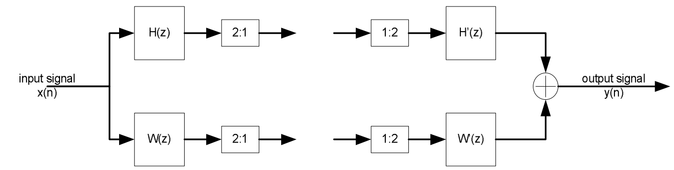


Scaling function · Wavelet function · Joint freq. response

Wavelets / Daubechies 20 – (2)



Wavelets and filterbanks



Analysis:

- H, the scaling function
- W, the wavelet function

$$w_k = (-1)^k h_{n-k-1}$$

Synthesis:

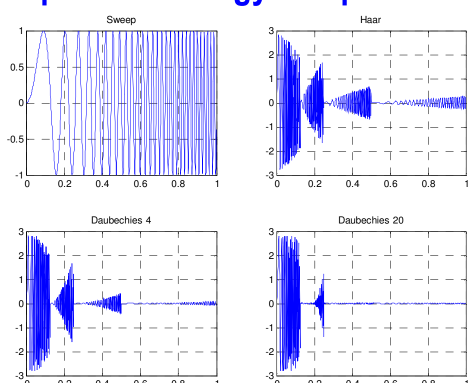
- H' for LP reconstruction
- W' for HP reconstruction

$$h_k^{-1} = \begin{cases} h_k & k = odd \\ h_{n-k-1} & k = even \end{cases}$$

$$w_k^{-1} = (-1)^k h_{n-k-1}^{-1}$$

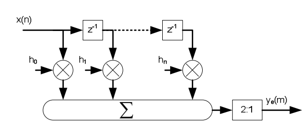
Wavelets/ towards compression

compare the energy compaction…!



Wavelets / Lifting introduction – (1)

- **The Scaling & Wavelet function are FIR filters, hence they can be implemented in a normal (direct-form) FIR structure**



where the result is sub-sampled after FIR filtering

Wavelets / Lifting introduction – (2)

45

- Write out the actual multiplications:

$$y_0 = h_0 \cdot x_0 + h_1 \cdot x_1 + h_2 \cdot x_2 + h_3 \cdot x_3 + \ldots$$

$$\cancel{y_1 = h_0 \cdot x_1 + h_1 \cdot x_2 + h_2 \cdot x_3 + h_3 \cdot x_4 + \ldots}$$

$$y_2 = h_0 \cdot x_2 + h_1 \cdot x_3 + h_2 \cdot x_4 + h_3 \cdot x_5 + \ldots$$

$$\cancel{y_3 = h_0 \cdot x_3 + h_1 \cdot x_4 + h_2 \cdot x_5 + h_3 \cdot x_6 + \ldots}$$

$$y_4 = h_0 \cdot x_4 + h_1 \cdot x_5 + h_2 \cdot x_6 + h_3 \cdot x_7 + \ldots$$

$$\cancel{y_5 = h_0 \cdot x_5 + h_1 \cdot x_6 + h_2 \cdot x_7 + h_3 \cdot x_8 + \ldots}$$

- Notice the odd/even coefficient/data dependency:
  - the same parity has the same MUAC operations!

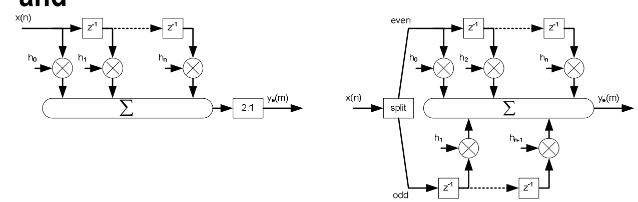
Wavelets / Lifting introduction – (3)

46

- Hence symbolically:

$$y_{even} = h_{even} \cdot x_{even} + h_{odd} \cdot x_{odd}$$

and



Wavelets / Special fast implementation: Lifting

47

- For an efficient implementation, compare with the fast DCT (Lifting developed by W. Sweldens and others)

  1) The signal is split in *odd* and *even* samples

  2) Hereafter:

```
odd [n] = odd [n-1] – Predict( even [n])
even[n] = even [n-1] + Update ( odd [n])
```

➢ Here illustrated with Daubechies 4-tap

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

Wavelets / Lifting (forward transform)

48

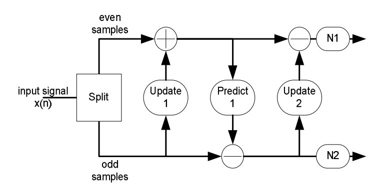
```
after Split x(0)..x(half-1) => even samples, x(half)..x(N-1) => odd samples

* Update 1 (even):
for n=0:half-1
  x(n) = x(n)+sqrt(3)*x(half+n)

* Predict 1 (odd):
x(half) = x(half)-sqrt(3)/4*x(0)-(sqrt(3)-2)/4*x(half-1)
for n=1:half-1
  x(half+n) = x(half+n)-sqrt(3)/4*x(n)-(sqrt(3)-2)/4*x(n-1)

* Update 2 (even):
for n=0:half-2
  x(n) = x(n)-x(half+n+1)
x(half-1) = x(half-1)-x(half)

* Normalize (both):
for n=0:half-1
  x(n) = (sqrt(3)-1)/sqrt(2)*x(n)
  x(n+half) = (sqrt(3)+1)/sqrt(2)*x(n+half)
```



Wavelets / Lifting (inv., mirror forward)

49

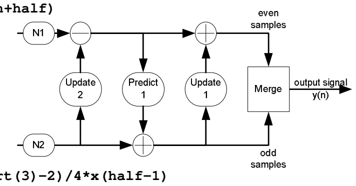
```
* Normalize' (both):
for n=0:half-1
  x(n) = (sqrt(3)+1)/sqrt(2)*x(n)
  x(n+half) = (sqrt(3)-1)/sqrt(2)*x(n+half)

* Update 2' (even):
for n=0:half-2
  x(n) = x(n)+x(half+n+1)
x(half-1) = x(half-1)+x(half)

* Predict 1' (odd):
x(half) = x(half)+sqrt(3)/4*x(0)+(sqrt(3)-2)/4*x(half-1)
for n=1:half-1
  x(half+n) = x(half+n)+sqrt(3)/4*x(n)+(sqrt(3)-2)/4*x(n-1)

* Update 1' (even):
for n=0:half-1
  x(n) = x(n)-sqrt(3)*x(half+n)

after Merge the output signal y(n) is available
```



50

Module 01A (part 2): Introduction to wavelet theory

Coefficient coding EZW and SPIHT, Intraframe and interframe schemes

Wavelet Video Coding

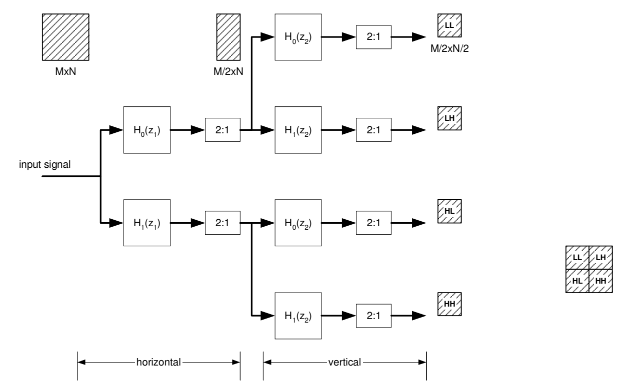
51

- **Intraframe** (within a picture)
  - Still pictures
  - Only 2D, within on field/frame
- **Interframe** (between pictures)
  - Moving video
  - 2D+time, sometimes called 3D

(true 3D coding is a completely different topic)

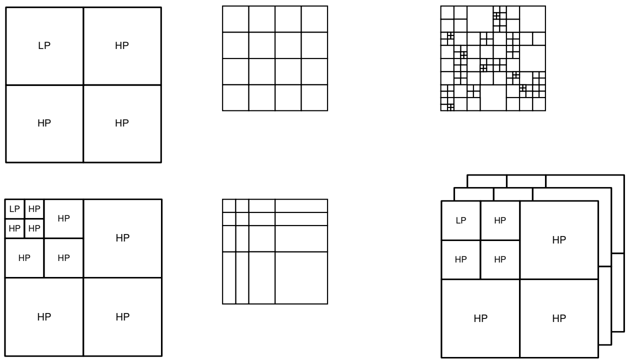
Wavelet intraframe video coding – (1) Two-dimensional filterbank

52



Wavelet intraframe video coding – (2) Example subband decompositions

53



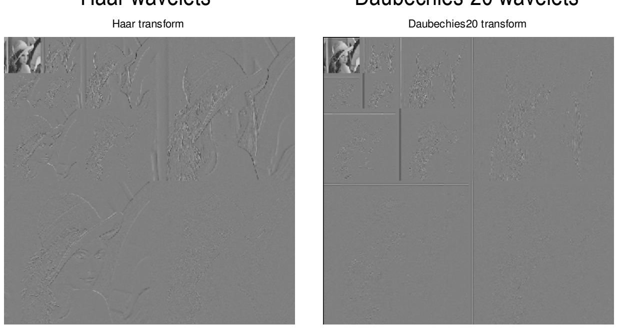
Wavelet intraframe video coding - (3) LENA example

54

Wavelet intraframe video coding – (4) LENA, 10-band decomposition

55

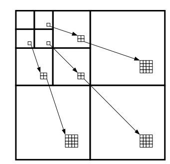
Haar wavelets　　Daubechies-20 wavelets

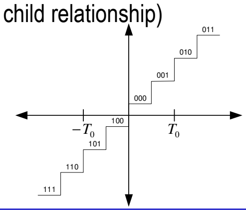


Wavelet intraframe video coding – (5)

56

* **Observation: the Wavelet (HP) sections are**
  - "relative" empty
  - related in the kind of information they carry (parent-child relationship)





Coding example, with a *N*-bit midrise quantizer: code a (complete or sub-) tree with less bits if all elements are below T_0.

Wavelet coefficient coding – (1) Embedded Zero-tree Wavelet (EZW)

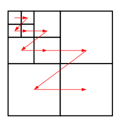
- **Invented by J. Shapiro, 1993**
- **Multi-pass (recursive) "tree" algorithm, each pass consisting of a** Dominant pass **(or significance map encoding) and a** Subordinate pass **(or refinement)**
- **Encoding stops when the bit-budget is exhausted, or another criterion is satisfied (e.g. a quality metric)**

Wavelet coefficient coding – (2) Embedded Zero-tree Wavelet (EZW)

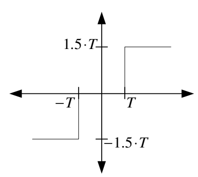
- **Dominant pass**
  - Given a threshold T
    - initially $T_0 = 2^{\lfloor \log_2 c_{\max} \rfloor}$ where c_{max} is the largest (abs) coefficient
    - And $T_i = \frac{1}{2} \cdot T_{i-1}$ recursively
  - Coefficient has a magnitude larger than T: **significant positive (sp) or significant negative (sn)**
  - Coefficient has a magnitude smaller AND all its descendants have magnitudes less than T: **zerotree root (zr)**
  - Coefficient has a magnitude smaller but one or more of its descendants have magnitudes larger than T: **isolated zero (iz)**
  - Each indicator costing 2 bits

Wavelet coefficient coding – (3) Embedded Zero-tree Wavelet (EZW)

- **Scanning order for encoding**



- **3-level Midtread quantizer (for reconstruction)**



Wavelet coefficient coding – (4) Embedded Zero-tree Wavelet (EZW)

- **Subordinate / refinement pass**
  - Reconstruct: $\text{reconstruction value} = 1.5 \cdot T_i$
  - Compute difference (pos/neg) with original and quantize with a two-level quantizer (costing 1 bit)

$$\text{reconstructed correction} \pm \frac{T_i}{4}$$

Wavelet coefficient coding - (5)
Embedded Zero-tree Wavelet (EZW)

- Example

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

- Initial threshold (T_0) is 16
- First scan (upper left): sp zr zr zr L_s={26}
  - Encoding costs: 8 bits

Wavelet coefficient coding - (6)
Embedded Zero-tree Wavelet (EZW)

- Refinement pass, 1.5x16=24
  - 26-24,=2 hence positive
  - Encoding cost 9 bits

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

- Reconstruction incl. refinement,
  - Correction is T_0 / 4 = 4

28	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Wavelet coefficient coding – (7)
Embedded Zero-tree Wavelet (EZW)

- 3-pass result from example

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

Pass 1 (T_0 = 16): `sp zr zr zr` `8 bits`

L_s={ 26}

Pass 2: `iz zr zr sp sp iz iz` `14 bits`

L_s={ 26, 13, 10}

Pass 3: `sp sn sp sp sp sp sn iz iz sp iz iz iz` `26 bits`

L_s={ 26, 13, 10, 6, -7, 7, 6, 4, 4, -4, 4}

Wavelet coefficient coding – (8)
Set Partitioning In Hierarchical Trees (SPIHT)

- **Invented by A. Said & W. Pearlman, 1996**
  - offering improved coding performance over EZW
- **Main idea: a multi-pass (recursive) "tree" algorithm is equal to EZW.**
  - It stops when a constraint is met (e.g. exhausted bit-budget)
- **In each step the following 3 lists are managed (in order of execution):**
  1. List of Insignificant Pixels (LIP)
  2. List of Insignificant Sets (LIS)
  3. List of Significant Pixels (LSP)

Wavelet coefficient coding – (9)

65

Set Partitioning In Hierarchical Trees (SPIHT)

- **Possible Sets (or Trees) are:**
  - *D* :set of descendants with insignificant children
  - *L* :set of descendants with significance (not in LIS since recognition of *D* suffices
- **Significance depending on threshold:** $T_i = 2^{n_i}$ **where** $n_0 = \lfloor \log_2 c_{\max} \rfloor$ **and** $n_i = n_{i-1} - 1$ **(eff. bitplane coding)**
- **Initialization:**
  - LIP: root nodes
  - LIS: root sets, except top
  - LSP: empty

Wavelet coefficient coding – (10)

66

Set Partitioning In Hierarchical Trees (SPIHT)

- **Example**

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

- **Initialization:**
  - LIP = {(0,0)26, (0,1)6, (1,0)-7, (1,1)7}
  - LIS = {(0,1)D, (1,0)D, (1,1)D}
  - LSP= {}

Wavelet coefficient coding – (11)

67

Set Partitioning In Hierarchical Trees (SPIHT)

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

- **Initialization**
  - LIP = {(0,0)26, (0,1)6, (1,0)-7, (1,1)7}
  - LIS = {(0,1)D, (1,0)D, (1,1)D}
  - LSP= {}
- **First pass, n=4, T=16:**
  - LIP = {(0,1)6, (1,0)-7, (1,1)7}
  - LIS = {(0,1)D, (1,0)D, (1,1)D}
  - LSP= {(0,0)26}
    - Encoding 10000 000

Wavelet coefficient coding – (12)

68

Set Partitioning In Hierarchical Trees (SPIHT)

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

- **First pass, n=4, T=16:**
  - LIP = {(0,1)6, (1,0)-7, (1,1)7}
  - LIS = {(0,1)D, (1,0)D, (1,1)D}
  - LSP= {(0,0)26}
- **Second pass, n=3, T=8:**
  - LIP = {(0,1)6, (1,0)-7, (1,1)7, (1,2)6, (1,3)4}
  - LIS = {(1,0)D,(1,1)D}
  - LSP= {(0,0)26, (0,2)13, (0,3)10}
    - Encoding 000 110100000 0

Wavelet coefficient coding – (13)

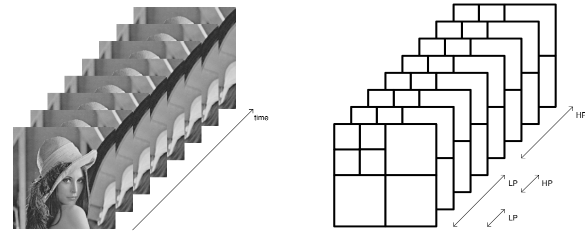
Set Partitioning In Hierarchical Trees (SPIHT)

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

- **Second pass, n=3, T=8:**
  - LIP = {(0,1)6, (1,0)-7, (1,1)7, (1,2)6, (1,3)4}
  - LIS = {(1,0)D,(1,1)D}
  - LSP= {(0,0)26, (0,2)13, (0,3)10}
- **Third pass, n=2, T=4:**
  - LIP = {(3,0)2, (3,1)-2, (2,3)-3, (3,2)-2, (3,3)0}
  - LIS = {}
  - LSP= {(0,0)26, (0,2)13, (0,3)10, 6, -7, 7, 6, 4, 4, -4, 4}
    - Encoding 1011101010 1101100110000 010 etc.

Wavelet interframe video coding – (1)

- **3D Wavelet Transform (no motion information)**



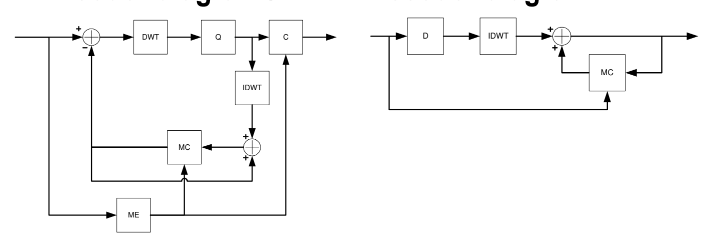
- **Grouping pictures (GOP/GOF): boundary effects!**
- **Wavelet encoding by: 3D-EZW, 3D-SPIHT, …**

Wavelet interframe video coding – (2)

- **Video coding & motion compensation (MC)**
  - For good video compression performance one needs motion compensation to get the highest correlation between frames (e.g. MPEG-2)
  - A good motion estimator (ME) is required
    - issues in: "Motion Estimation Techn. Overview"
  - Wavelets combined with MC prediction is still in its early stages

Wavelet interframe video coding – (3)

- **Possible replacement of DCT by Wavelet for a Motion Compensation loop (block issues)**
- **Encoder diagram / Decoder diagram**



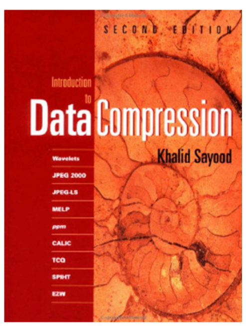
Wavelets conclusions / Complexity & coding aspects

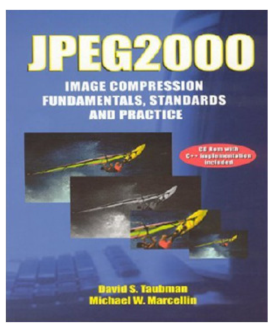
	DCT	Wavelets
Computational	+ (low)	+ (low)
Memory	+ (8x8 blocks)	-- (complete frame)
Perceptual quality	-- (blocking)	+ (smoothness)

Wavelets / Concluding remarks

- **Only an introduction**
  - to the mathematical background & reasons of wavelets and its processing techniques
  - on intraframe coding, more in JPEG2000
  - on interframe wavelet compression issues
    - ➢much recent work in this field
  - on signal analysis, e.g. for noise reduction
    - ➢often its quality rivals traditional methods

References & Recommended reading





...and the world wide web