Filters in Image Processing

2^{nd} Generation of Wavelets – Lifting Scheme

David Svoboda

email: svoboda@fi.muni.cz
Centre for Biomedical Image Analysis
Faculty of Informatics, Masaryk University, Brno, CZ

CBIA

October 21, 2019

Outline

Motivation

Practical use of DWT or FWT?

- Complexity of:
  - DWT: $O(n^2)$
  - FWT: $O(cn)$

  Can we do it faster?
- DWT/FWT are computed in floating point arithmetic.
  Can we restrict ourselves to integer number?
- DWT/FWT are computed *out-of-place*.
  Can we reduce the memory usage?
- Some basis functions in DWT are not orthogonal $\rightarrow$ dual basis must be found.
  Can we avoid this issue?

Z-Transform

Definition

(bilateral) Z-Transform

$$\mathcal{F}(z) = \sum_{n=-\infty}^{\infty} f(n)z^{-n}$$

(unilateral) Z-Transform

$$\mathcal{F}(z) = \sum_{n=0}^{\infty} f(n)z^{-n}$$

Notice: Here $z \in \mathbb{C}$, i.e. for $z = e^{i\omega}$ we get a special case of Z-transform, which is DFT.

Z-Transform
Definition

Forward transform

- converts discrete series into continuous signal (Z-plane)

$$\mathcal{F}(z) = \sum_{n=0}^{\infty} f(n) z^{-n}$$

Inverse transform

- maps continuous signal into discrete series

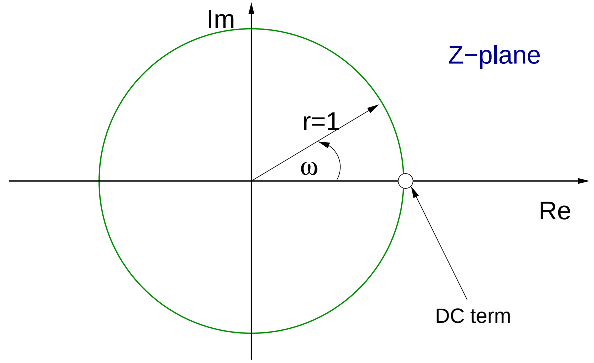
$$f(n) = Z^{-1}(\mathcal{F}(z)) = \frac{1}{2\pi i} \oint_C \mathcal{F}(z) z^{n-1} dz$$

where C is a counterclockwise closed path encircling (with radius 1) the origin

Z-Transform
Properties

Important notes:

- green circle ($z = e^{i\varphi} \Rightarrow |z| = 1$) reduces Z-transform simply to discrete Fourier transform
- DC (direct current) term is DC term from DFT



Z-Transform
Properties

List of the most important properties

- linearity:
$$Z\{af(n) + bg(n)\} = a\mathcal{F}(z) + b\mathcal{G}(z)$$
- delay (shift):
$$Z\{f(n-k)\} = \mathcal{F}(z) z^{-k}$$
- convolution theorem:
$$Z\{f(n) * g(n)\} = \mathcal{F}(z) \cdot \mathcal{G}(z)$$
- symmetry:
$$Z\{f(-n)\} = \mathcal{F}(z^{-1})$$

Z-Transform
Some examples

- Z-transform of a constant signal:

$$\begin{aligned} f(n) &= [1, 1, 1, 1, \ldots] \\ \mathcal{F}(z) &= 1 + z^{-1} + z^{-2} + z^{-3} + \cdots = \frac{1}{1 - z^{-1}} = \frac{z}{z-1} \end{aligned}$$

- Z-transform of a linear filter:

$$\begin{aligned} \mathit{filter}(f(k)) &= (-2)f(k-1) + 3f(k) + 5f(k+2) \\ &= [-2, \boxed{3}, 0, 5] \quad \text{/written as a kernel/} \\ \mathit{Filter}(z) &= (-2)z^{-1} + 3z^{0} + 5z^{2} \end{aligned}$$

Notice: Inverse for linear filters is straightforward.

Analysis of FWT

Let $|f| = N = 8 = 2^3 = 2^J$ and $j_0 = 0$

$$\begin{array}{cccccccc} f(0) & f(1) & f(2) & f(3) & f(4) & f(5) & f(6) & f(7) \\ \| & \| & \| & \| & \| & \| & \| & \| \\ A_3(0) & A_3(1) & A_3(2) & A_3(3) & A_3(4) & A_3(5) & A_3(6) & A_3(7) \end{array}$$

$$\Downarrow$$

$$A_2(0) \quad A_2(1) \quad A_2(2) \quad A_2(3) \quad D_2(0) \quad D_2(1) \quad D_2(2) \quad D_2(3)$$

$$\Downarrow$$

$$A_1(0) \quad A_1(1) \quad D_1(0) \quad D_1(1)$$

$$\Downarrow$$

$$A_0(0) \quad D_0(0)$$

Analysis of FWT

An example (Haar wavelets)

Let $f = A_2 = [1\ \ 4\ \ \text{-}3\ \ 0]$ be an input signal:

$$A_1 = \left(f * \left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right]\right)(\downarrow 2\times)$$

$$D_1 = \left(f * \left[\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right]\right)(\downarrow 2\times)$$

During one transition, we perform only one averaging and difference:

$$A_1(k) = \left((f(k) + f(k+1))\frac{\sqrt{2}}{2}\right)(\downarrow 2\times)$$

$$D_1(k) = \left((f(k+1) - f(k))\frac{\sqrt{2}}{2}\right)(\downarrow 2\times)$$

Analysis of FWT

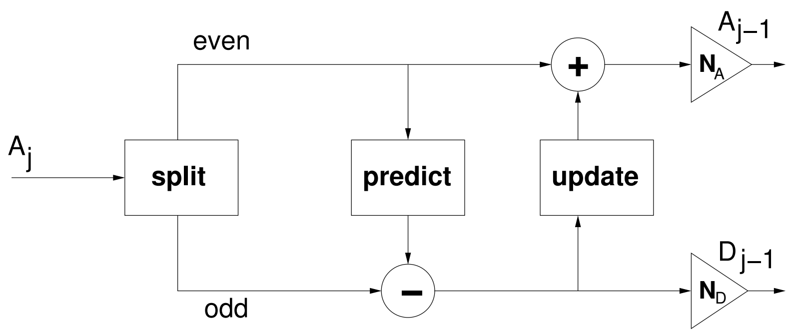
An example (Haar wavelets)

```
>> [LoD, HiD, LoR, HiR] = wfilters('haar')
LoD = 0.7071    0.7071
HiD =-0.7071    0.7071
LoR = 0.7071    0.7071
HiR = 0.7071   -0.7071

>> [A1,D1] = dwt([1 4 -3 0], LoD, HiD)
A1 = 3.5355   -2.1213
D1 =-2.1213   -2.1213

>> [A0,D0] = dwt(A1, LoD, HiD)
A0 = 1.0000
D0 = 4.0000
```

Here $c = |LoD| = |HiD| = 2$.

Lifting scheme (LS)

- Developed in 1996 by Wim Sweldens
- Adopted idea of FastDWT
- The transition from level j to $j-1$ is computed efficiently
- Basic idea: split $\rightarrow$ predict $\rightarrow$ update $\rightarrow$ normalize

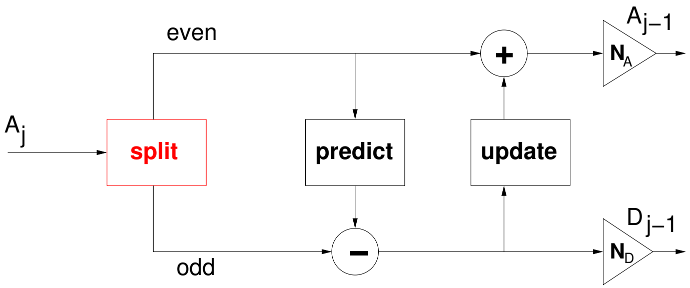


Lifting scheme

Split step

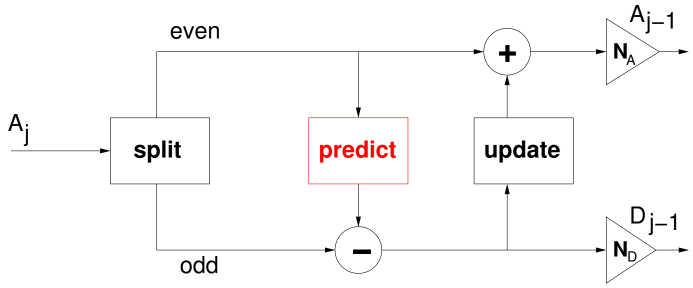
- Also known as *lazy wavelet*.
- The signal A_j is split into odd and even samples:

$$(A_{j-1}, D_{j-1}) = split(A_j)$$



Lifting scheme

Prediction step

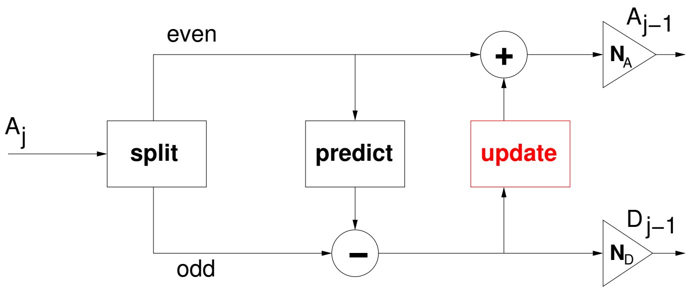


$$D_{j-1}(k) = D_{j-1}(k) - predict(A_{j-1}(k))$$

- Also known as *dual lifting step*
- When using Haar wavelets the neighbouring samples are supposed to be equal, i.e. the predictor is simple: $predict_{Haar}(f(k)) = f(k)$
  $D_{j-1}(k) = D_{j-1}(k) - A_{j-1}(k)$

Lifting scheme

Update step

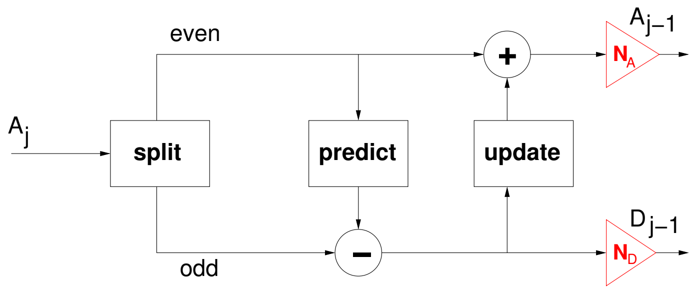


$$A_{j-1}(k) = A_{j-1}(k) + update(D_{j-1}(k))$$

- Also known as *primal lifting step*
- Update step repairs the wrong estimate of the prediction step.
- When using Haar wavelet, we use: $update_{Haar}(f(k)) = \frac{1}{2} f(k)$
  $A_{j-1}(k) = A_{j-1}(k) + \frac{1}{2} D_{j-1}(k)$

Lifting scheme

Normalization step



$$\begin{aligned} A_{j-1}(k) &= A_{j-1}(k) \cdot N_A \\ D_{j-1}(k) &= D_{j-1}(k) \cdot N_D \\ N_A \cdot N_D &= 1 \end{aligned}$$

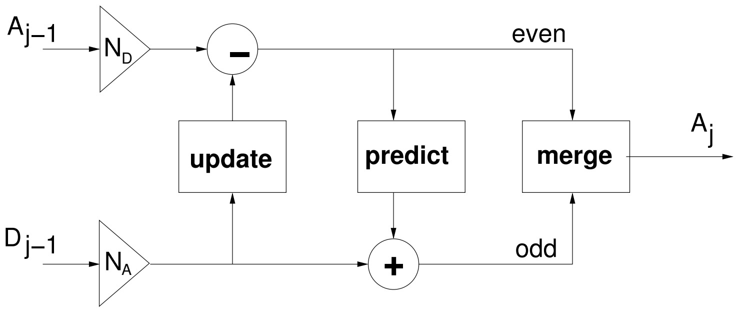
- The output signals are normalized to avoid boosting the signal.

Lifting scheme

Inverse lifting

- The forward algorithm can be simply inverted:

$$
\begin{aligned}
A_{j-1}(k) &= A_{j-1}(k) - update(D_{j-1}(k)) \\
D_{j-1}(k) &= D_{j-1}(k) + predict(A_{j-1}(k)) \\
A_j &= merge(A_{j-1}, D_{j-1})
\end{aligned}
$$

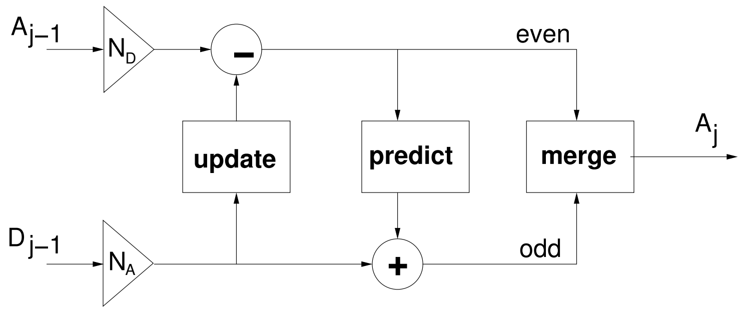


Lifting scheme

Inverse lifting (example)

- Namely for Haar wavelets we get:

$$
\begin{aligned}
A_{j-1}(k) &= A_{j-1}(k) \cdot N_D \\
D_{j-1}(k) &= D_{j-1}(k) \cdot N_A \\
A_{j-1}(k) &= A_{j-1}(k) - (D_{j-1}(k)/2) \\
D_{j-1}(k) &= D_{j-1}(k) + A_{j-1}(k) \\
A_j &= merge(A_{j-1}, D_{j-1})
\end{aligned}
$$



Lifting scheme

From Filters to Lifting

There exists algorithm invented by Wim Sweldens (1996):

1. Input: either `'LoD'` and `'HiD'` filters or ϕ and φ functions
2. Convert both filters `'LoD'` and `'HiD'` to Z-domain
3. Create *polyphase matrix* (2×2)
4. Factorize matrix into simple (lower and upper diagonal) matrices
5. Each simple matrix correspond either to one update or prediction step
6. Convert each matrix from Z-domain to time domain

The algorithm is quite tricky → we utilize Matlab function `'liftwave'` that performs steps 1-5. All we have to do is to enjoy step 6 ☺

Lifting scheme

From Filters to Lifting (Haar)

```
>> h = liftwave('haar')
h = 'd'          [     -1]    [0]
    'p'          [0.5000]    [0]
    [1.4142]     [0.7071]     []
```

- `'d'` ...**d**ual lifting step (predict)
  $Predict(z) = (-1) \cdot z^0 \quad \rightarrow \quad predict(f(k)) = (-1) \cdot f(k)$
- `'p'` ...**p**rimal lifting step (update)
  $Update(z) = 0.5 \cdot z^0 \quad \rightarrow \quad update(f(k)) = 0.5 \cdot f(k)$
- `[1.4142]` ... N_A
- `[0.7071]` ... N_D

Lifting scheme

From Filters to Lifting (Daubechies-2)

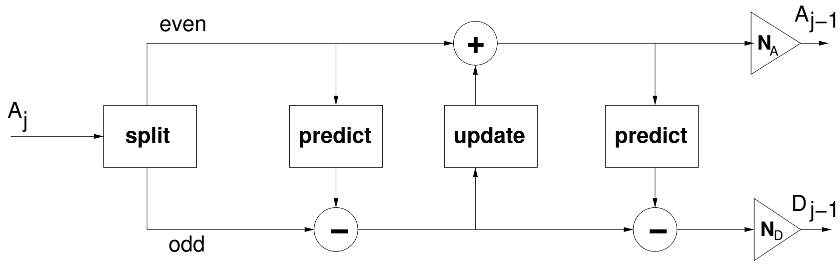
In general, there may appear two or more *predict*(.) and *update*(.) steps:

```
>> h = liftwave('db2')
h = {...
     'd'              [ -1.732]              [0]
     'p'              [ -0.067   0.433]      [1]
     'd'              [  1.000]              [-1]
     [  1.932]        [  0.518]              []
```

- 'd' ...**dual** lifting step (predict)
  $Predict_1(z) = (-1.732) \cdot z^0 \quad \rightarrow \quad predict_1(f(k)) = (-1.732) \cdot f(k)$
- 'p' ...**primal** lifting step (update)
  $Update_1(z) = (-0.067) \cdot z^1 + 0.433 \cdot z^0 \quad \rightarrow$
  $update_1(f(k)) = (-0.067) \cdot f(k+1) + 0.433 \cdot f(k)$
- 'd' ...**dual** lifting step (predict)
  $Predict_2(z) = 1.000 \cdot z^{-1} \quad \rightarrow \quad predict_2(f(k)) = 1.000 \cdot f(k-1)$
- [1.932] ... N_A
- [0.518] ... N_D

Lifting scheme

From Filters to Lifting (Daubechies-2)



$$
\begin{aligned}
predict_1(f(k)) &= (-1.732)\, f(k) \\
update_1(f(k)) &= 0.433\, f(k) - 0.067\, f(k+1) \\
predict_2(f(k)) &= f(k-1)
\end{aligned}
$$

Lifting scheme

Technical/Implementation notes

Lifting ordering (for $N = 8$)

f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	f(6)	f(7)
$A_2(0)$	$D_2(0)$	$A_2(1)$	$D_2(1)$	$A_2(2)$	$D_2(2)$	$A_2(3)$	$D_2(3)$
$A_1(0)$		$D_1(0)$		$A_1(1)$		$D_1(1)$	
$A_0(0)$				$D_0(0)$			

Notice: The computation is performed completedly *in-place*.

From filters to lifting scheme

- conversion 'LoD','HiD' $\rightarrow$ *update*$(\cdot)$, *predict*$(\cdot)$ always exists but is not unique
- conversion is performed in frequency domain via Z-transform (decomposition of polyphase matrices)

Lifting scheme

FWT and DWT comparison (examples)

Price of one decomposition level using DWT ($|f| = N$)

family of wavelets	multiplications	additions
Haar	$4N$	$2N$
Daubechies-2	$8N$	$6N$

Extra memory usage: one memory buffer of size $O(N)$ needed for convolution.

Price of one decomposition level using LS ($|f| = N$)

family of wavelets	multiplications	additions
Haar	$2N$	N
Daubechies-2	$3N$	$2N$

Extra memory usage: $\emptyset$

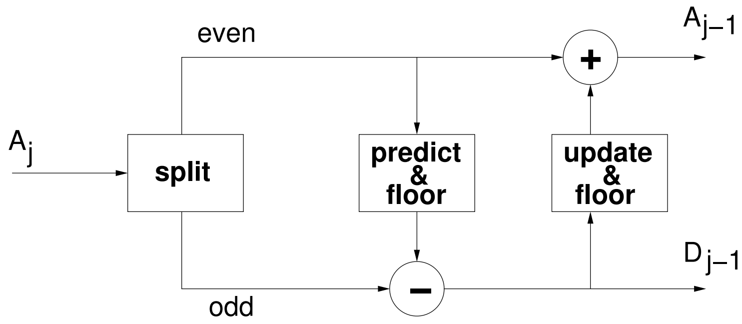
Integer Wavelet Transform

Basic idea & Properties

- IWT originates from lifting scheme (chain of predictions and updates).
- The fixed point arithmetic is guaranteed via `'floor'` function.
- The rounding error produced in forward transform is compensated by mirror `'floor'` in inverse lifting.
- The lifting is the same as the standard one but
  - each multiplication is followed but the truncation
  - no normalization is present

Integer Wavelet Transform

An example (Haar)



$$
\begin{aligned}
(A_{j-1}, D_{j-1}) &= split(A_j) \\
D_{j-1}(k) &= D_{j-1}(k) - floor\,(A_{j-1}(k)) \\
A_{j-1}(k) &= A_{j-1}(k) + floor\,(D_{j-1}(k)/2)
\end{aligned}
$$

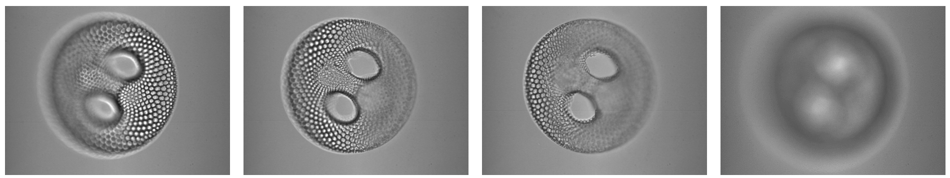
Integer Wavelet Transform

An example (Haar)

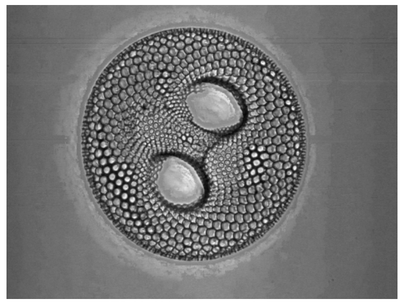
```
>> h2 = liftwave('haar', 'Int2Int')
h2 = 'd'          [    -1]     [0]
     'p'          [0.5000]     [0]
     [1.4142]     [0.7071]     'I'

>> [A1,D1] = lwt([1 4 -3 0], h2)
A1 = 2    -2
D1 = 3     3

>> ilwt(A1,D1,h2)
ans = 1      4    -3      0
```


Applications

Image fusion – in confocal microscopy



$\Downarrow$



source: B. Zítová, UTIA, CAS

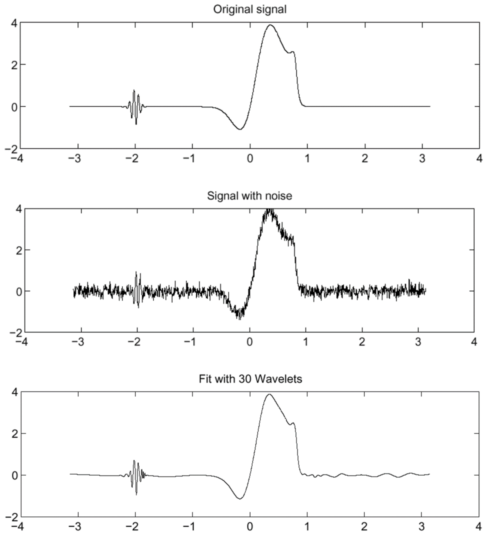
Applications

Image fusion – in photography

⇓

Applications

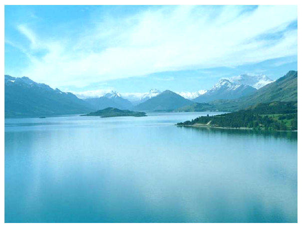
Noise removal



source: B. Zítová, UTIA, CAS

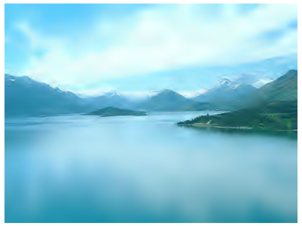
Applications

Image compression



original (979 kB)

JPEG (6.21 kB)



JPEG2000 (1.83 kB)

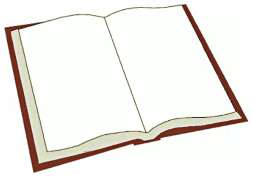
Applications

. . . and the others

- fusion of images with different resolution
- image registration
- edge detection
- . . .

Bibliography

- Li H., Manjunath B.S., Mitra S.K. Multisensor Image Fusion Using the Wavelet Transform, Graphical Models and Image Processing, Volume 57, Issue 3, May 1995, Pages 235-245, ISSN 1077-3169
- Jensen A., La Cour-Harbo A. Ripples in mathematics: the discrete wavelet transform, Springer, Berlin, 2001, ISBN 3-540-41662-5
- Sweldens W. The lifting scheme: A custom-design construction of biorthogonal wavelets. Applied and Computational Harmonic Analysis. 1996, Vol 3, Issue 2, pp 186200



You should know the answers . . .

- Describe the relationship between Fourier transform and Z-transform.
- Can you apply Z-transform to a given signal or a linear filter? Give an example.
- Explain three principal phases of *lifting scheme*.
- Compare the time and spatial complexity of FWT and lifting scheme.
- Compute one level of wavelet transform (using Haar's basis) via lifting scheme for signal `f=[3 5 0 -1 4 2]`.
- What does *integer* wavelet transform mean?
- How does image fusion via DWT work?
- How does image denoising via DWT work?