The lifting steps implementation

Second generation wavelets

Introduction

- The idea of lifting
- Mathematical context
- The filter-bank side
- Polyphase representation
- Features
- Conclusions
- Reference: “Factoring wavelet transforms into lifting steps”, I. Daubechies, W. Sweldens, 1996
  - Available on line

Sr. Vice President of Technology Commercialization at Lucent Tech. Bell Labs

Professor, Princeton University

The lifting scheme

- New philosophy in *Biorthogonal* wavelet construction
- Sweldens, ~95
- Both linear and non-linear wavelets
  - Integer implementation enabling lossless coding

Biorthogonal basis:why?

- FIR orthonormal filters: no symmetry
    - (except Haar filter)
- FIR biorthogonal filters: symmetry
    - linear phase
    - better boundary conditions

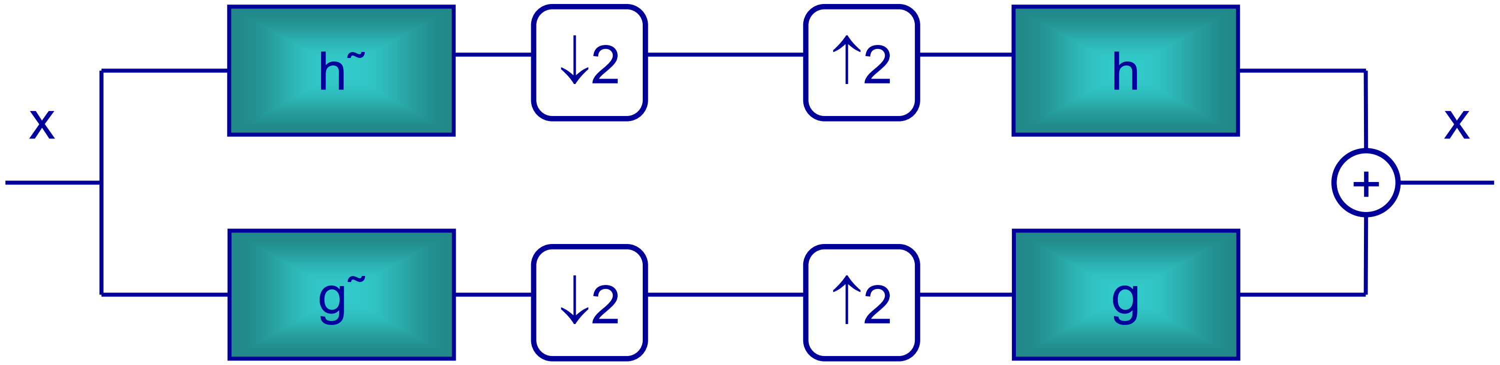
Basis oh the Hilbert space

- Orthonormal basis:
  - $\{e_n\}_{n \in N}$: family of the Hilbert space
  - $< e_n, e_p >= 0 \qquad \forall n \neq p$
  - $\forall x \in H, \quad \exists \lambda(n) = < x, e_n >$
  - $|e_n|^2 = 1$
  - $x = \sum_n \lambda(n)\, e_n$

Basis of the Hilbert space

- Riesz bases:
  - $\{e_n\}_{n \in N}$: linearly independent
  - $\forall y \in H$, $\exists A>0$ and $B>0$: $y=\sum_n \lambda(n)\, e_n$
  - $|y|^2/B \leq \sum_n |\lambda(n)|^2 \leq |y|^2/A$
  - $\lambda(n)=< y, e\tilde{}_n>$
  - $\{e\tilde{}_n\}_{n \in N}$: *dual family*
  - *Biorthogonality relationship:* $< e_n, e\tilde{}_p>=\delta(n-p)$
  - $y=\sum_n < y, e\tilde{}_n> e_n$
  - $A=B=1 \Rightarrow$ orthogonal basis

Biorthogonal filters



$$g[n] = (-1)^{1-n}\tilde{h}[1-n] \qquad g(z) = z^{-1}\tilde{h}(-z^{-1})$$
$$\tilde{g}[n] = (-1)^{1-n}h[1-n] \Leftrightarrow \tilde{g}(z) = z^{-1}h(-z^{-1})$$

$$h(z)\tilde{h}(z^{-1}) + g(z)\tilde{g}(z^{-1}) = 2$$ perfect reconstruction

$$h(z)\tilde{h}(-z^{-1}) + g(z)\tilde{g}(-z^{-1}) = 0$$ alias-free

Rationale

- Goal: Exploit the correlation structure present in most real life signals to build a sparse approximation
  - The correlation structure is typically local in space (time) and frequency
- Basic idea
  - Split the signal x in its *polyphase* components (even and odd samples)

$$x_e = \left(x_{2k}\right)_{k\in\mathbb{Z}}$$
$$x_o = \left(x_{2k+1}\right)_{k\in\mathbb{Z}}$$

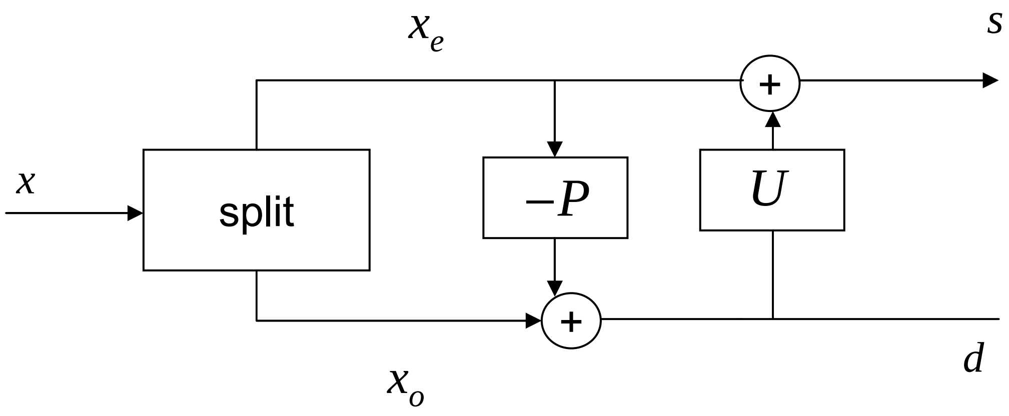
  - These two are highly correlated. It is thus natural to use one of them (e.g. the odds) to predict the other (e.g. the even)

$$x_o' = P\left(x_e\right) \qquad \text{predicted}$$
$$d = x_o - P\left(x_e\right) \qquad \textbf{difference} \text{ or } \textbf{detail}$$

  - The operation of computing a prediction and recording the detail we call *lifting step*

Lifting steps

- To get a good frequency splitting, the evens are also *updated* by replacing them with a smoothed version
- Built-in feature of lifting: no matter how P and U are chosen, the scheme is ***always invertible*** and thus leads to ***critically sampled perfect reconstruction filter banks***



$$x_o' = P(x_e)$$
$$d = x_o - P(x_e)$$

$$s = x_e + U(d)$$
$$x_e = s - U(d)$$

Biorthogonal FB

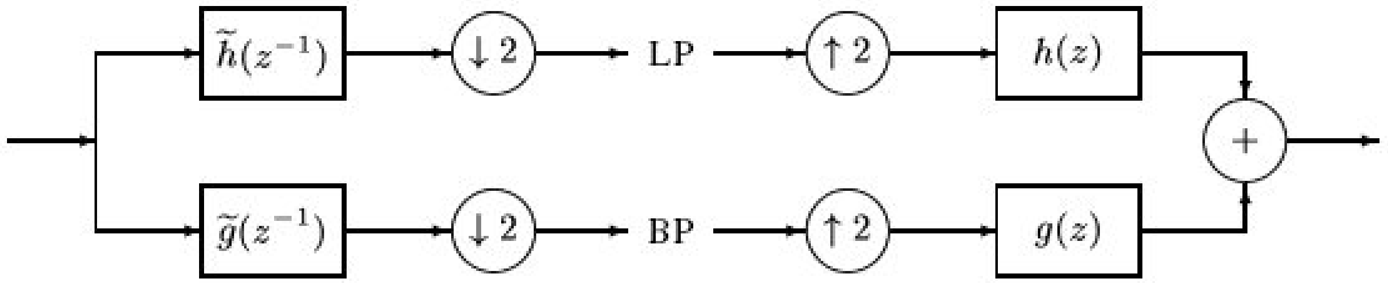


FIGURE 3. Discrete wavelet transform (or subband transform): The forward transform consists of two analysis filters $\tilde{h}$ (low-pass) and $\tilde{g}$ (high-pass) followed by subsampling, while the inverse transform first upsamples and then uses two synthesis filters h (low-pass) and g (high-pass).

$$\tilde{g}(z) = z^{-1}h(-z^{-1})$$
$$\tilde{h}(z) = -z^{-1}g(-z^{-1})$$

(h,g) in the analysis part

$$h\left(z^{-1}\right)\tilde{h}(z)+g\left(z^{-1}\right)\tilde{g}(z)=2$$
$$h\left(-z^{-1}\right)\tilde{h}(z)+g\left(-z^{-1}\right)\tilde{g}(z)=0$$

(h,g) in the synthesis part

$$h(z)\,\tilde{h}(z^{-1})+g(z)\,\tilde{g}(z^{-1})=2$$
$$h(z)\,\tilde{h}(-z^{-1})+g(z)\,\tilde{g}(-z^{-1})=0.$$

Polyphase representation

- Given the biorthoganl filters h and g

$$h(z) = h_e(z^2) + z^{-1} h_o(z^2)$$

$$h_e(z) = \sum_k h_{2k} z^{-k}$$ even coefficients

$$h_o(z) = \sum_k h_{2k+1} z^{-k}$$ odd coefficients

$$h_e(z^2) = \frac{h(z) + h(-z)}{2}$$

$$h_o(z^2) = \frac{h(z) - h(-z)}{2z^{-1}}$$

Polyphase matrix

$$P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix}$$

PR condition $P(z)\tilde{P}(z^{-1})^t = I$

Polyphase representation qui

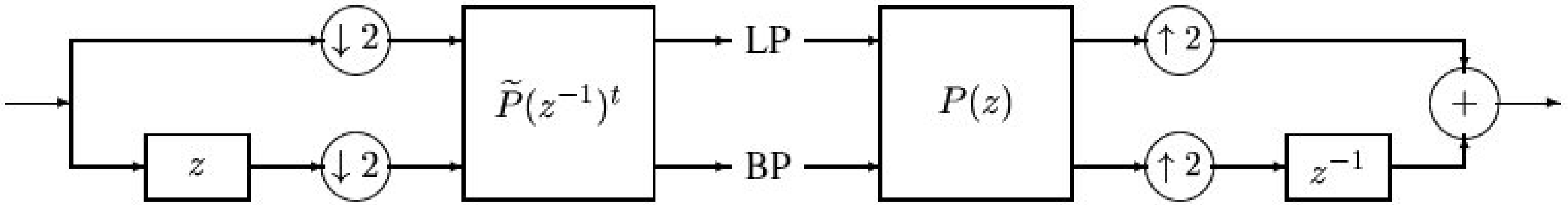
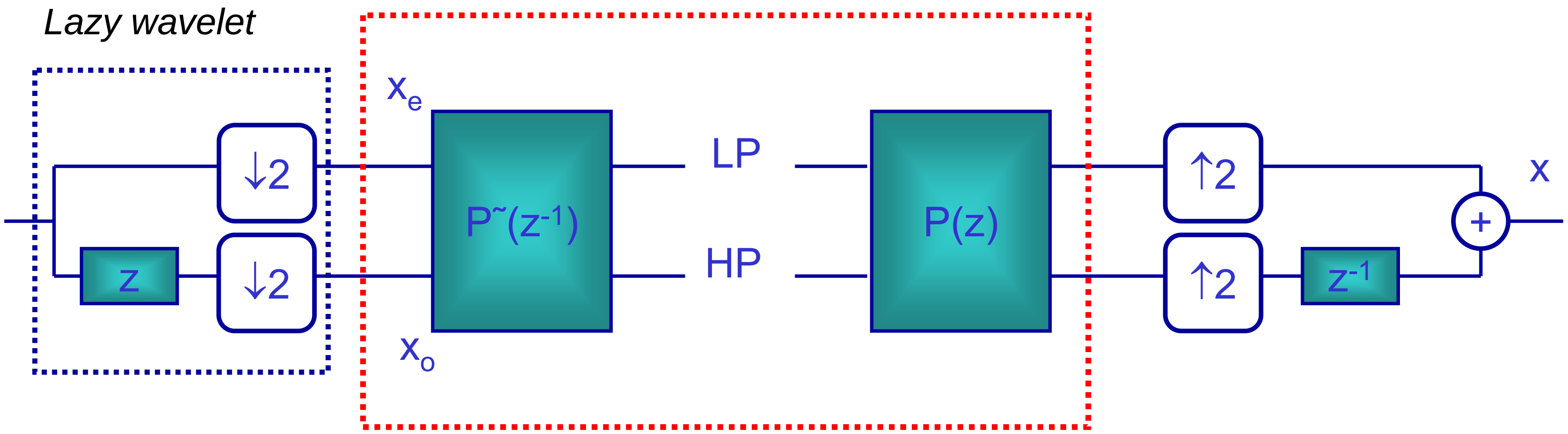


FIGURE 4. *Polyphase representation of wavelet transform: first subsample into even and odd, then apply the dual polyphase matrix. For the inverse transform: first apply the polyphase matrix and then join even and odd.*

→ the other way around

- The problem of finding a FIR wavelet transform then amounts to finding a matrix P(z) with determinant =1
- Once the matrix is given, the filters follow
  - One can show that this corresponds to the biorthogonality relations

$$\tilde{g}(z) = z^{-1} h(-z^{-1})$$
$$\tilde{h}(z) = -z^{-1} g(-z^{-1})$$



The lifting scheme

- Definition 1. A filter pair (h,g) is *complementary* if the corresponding polyphase matrix P(z) has determinant 1
  - If (h,g) is complementary, so is $\left(\tilde{h}, \tilde{g}\right)$

- Theorem 3 (Lifting). Let (h,g) be complementary. Then, any other finite filter $g^{new}(z)$ complementary to h is of the form

$$g^{new}\left(z\right) = g\left(z\right) + h\left(z\right)s\left(z^2\right)$$

  where s(z) is a Laurent polynomial. Conversely, any filter of this form is complementary to h

- Proof

$$P^{new}\left(z\right) = P\left(z\right)\begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix} \rightarrow \det P^{new}\left(z\right) = \det P\left(z\right) = 1$$

proof and consequences

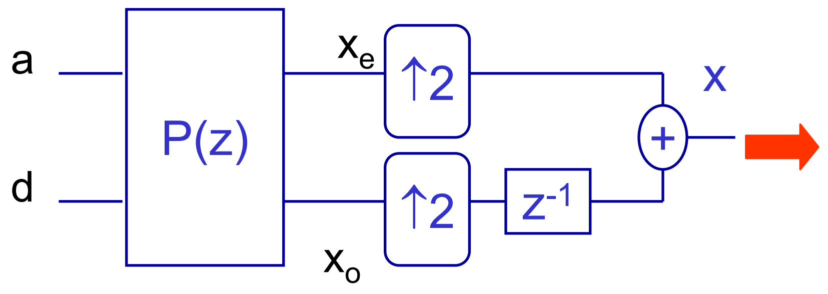
$$g'(z) = g(z) + h(z)s(z^2)$$

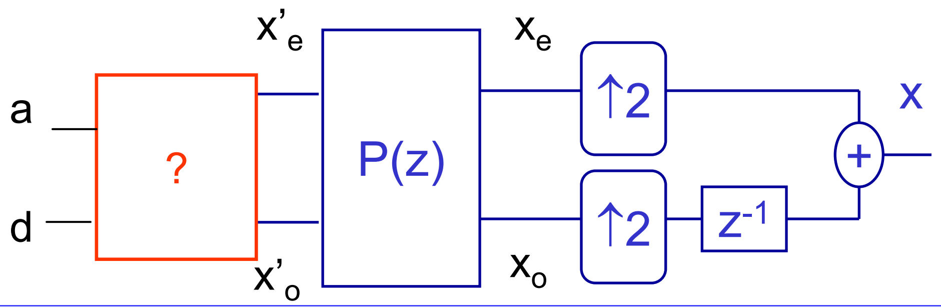
$$\begin{cases} g_e'(z) = g_e(z^2) + h_e(z^2)s(z^2) \\ g_o'(z) = g_o(z^2) + h_o(z^2)s(z^2) \end{cases}$$

$h(z)$ is unchanged

$$P'(z) = \begin{bmatrix} h'_e(z) & g'_e(z) \\ h'_o(z) & g'_o(z) \end{bmatrix} = \begin{bmatrix} h_e(z) & g_e(z) + s(z^2)h_e(z) \\ h_o(z) & g_o(z) + s(z^2)h_o(z) \end{bmatrix} =$$

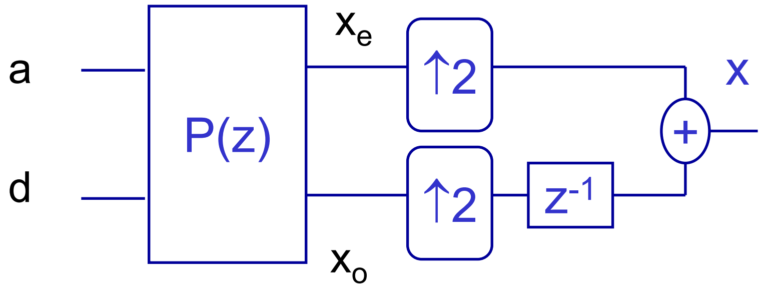
$$= \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \times \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix} = P(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}$$





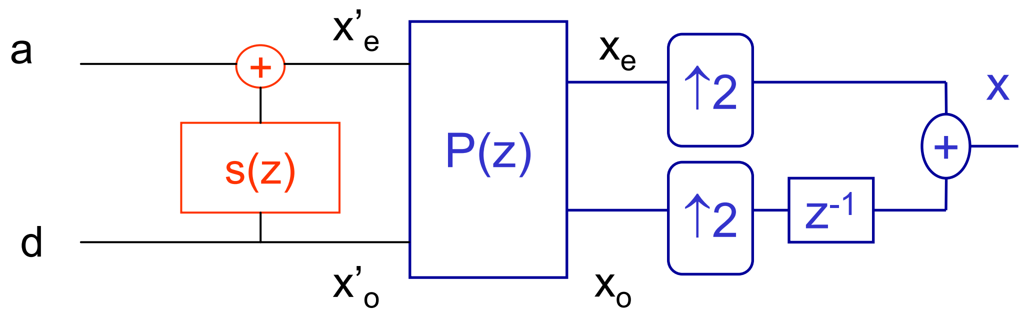
“Lifted” FB

$$\begin{bmatrix} x_e \\ x_o \end{bmatrix} = P(z)\begin{bmatrix} a \\ d \end{bmatrix}$$

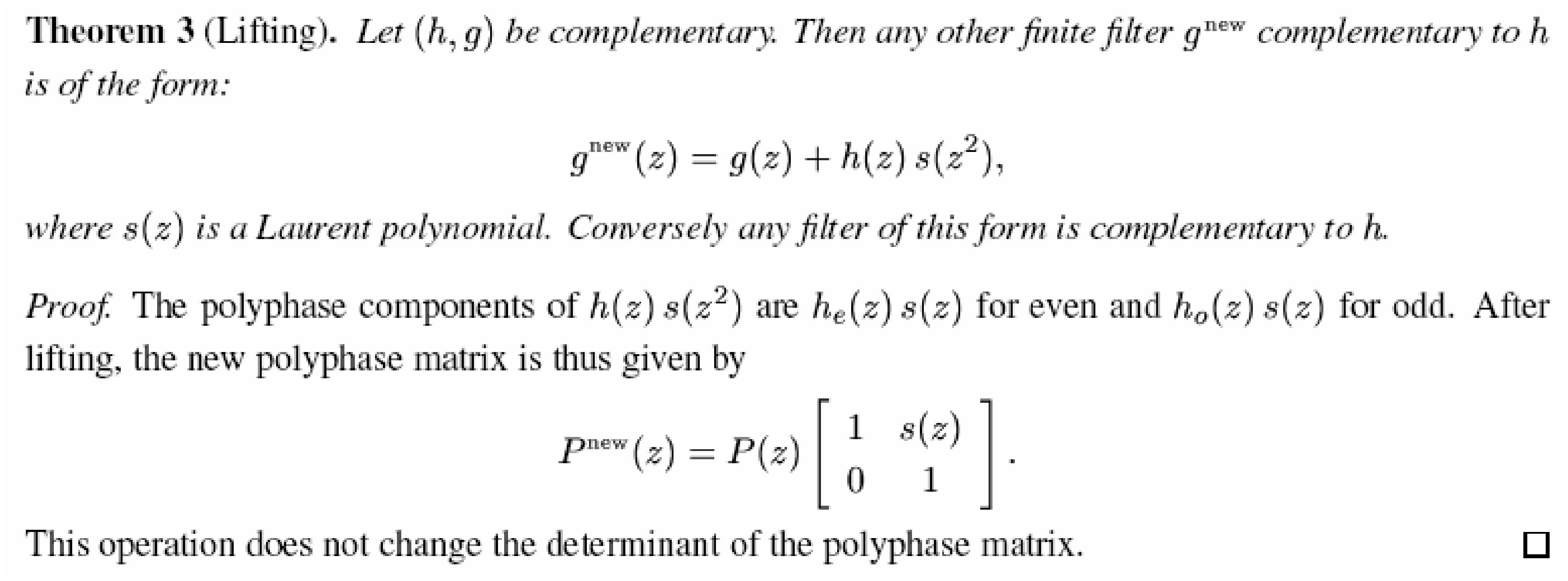


$$\begin{bmatrix} x'_e \\ x'_o \end{bmatrix} = \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} a \\ d \end{bmatrix}$$

$$\begin{bmatrix} x_e \\ x_o \end{bmatrix} = P(z)\begin{bmatrix} x'_e \\ x'_o \end{bmatrix} = \underbrace{P(z)\begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}}_{P'(z)}\begin{bmatrix} a \\ d \end{bmatrix}$$



From Swelden’s paper



Theorem 3 (Lifting). *Let (h, g) be complementary. Then any other finite filter g^{new} complementary to h is of the form:*

$$g^{\text{new}}(z) = g(z) + h(z)\, s(z^2),$$

where $s(z)$ is a Laurent polynomial. Conversely any filter of this form is complementary to h.

Proof. The polyphase components of $h(z)\, s(z^2)$ are $h_e(z)\, s(z)$ for even and $h_o(z)\, s(z)$ for odd. After lifting, the new polyphase matrix is thus given by

$$P^{\text{new}}(z) = P(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}.$$

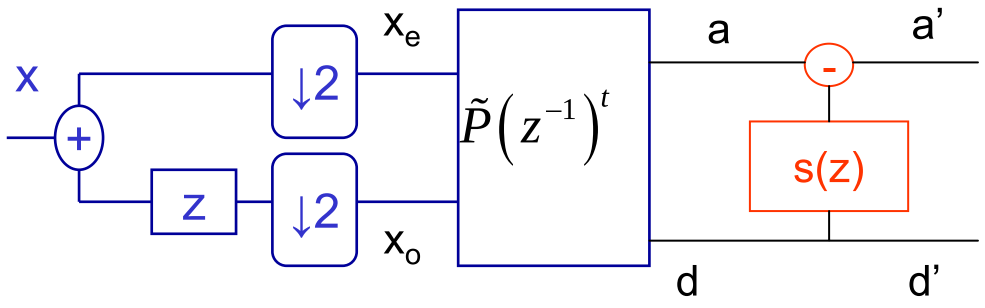
This operation does not change the determinant of the polyphase matrix. □

Analysis FB

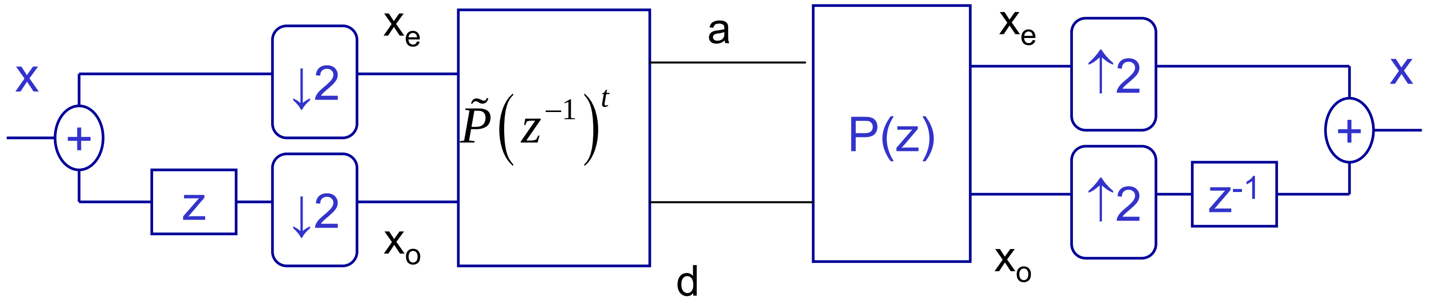
$$\tilde{g}(z) = z^{-1}h(-z^{-1})$$
$$\tilde{h}(z) = -z^{-1}g(-z^{-1})$$

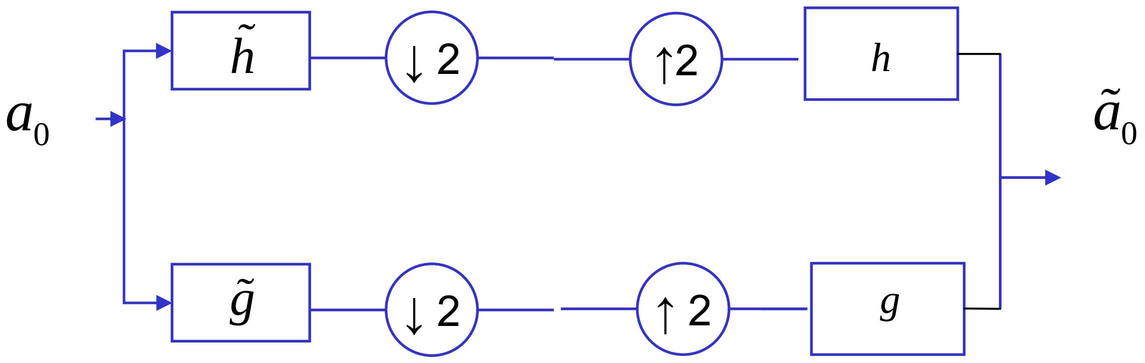
$$\tilde{P}^{new}(z) = \tilde{P}(z)\begin{bmatrix} 1 & 0 \\ -s(z^{-1}) & 1 \end{bmatrix} \rightarrow \tilde{h}^{new}(z) = \tilde{h}(z) - \tilde{g}(z)s(z^{-2})$$

Update

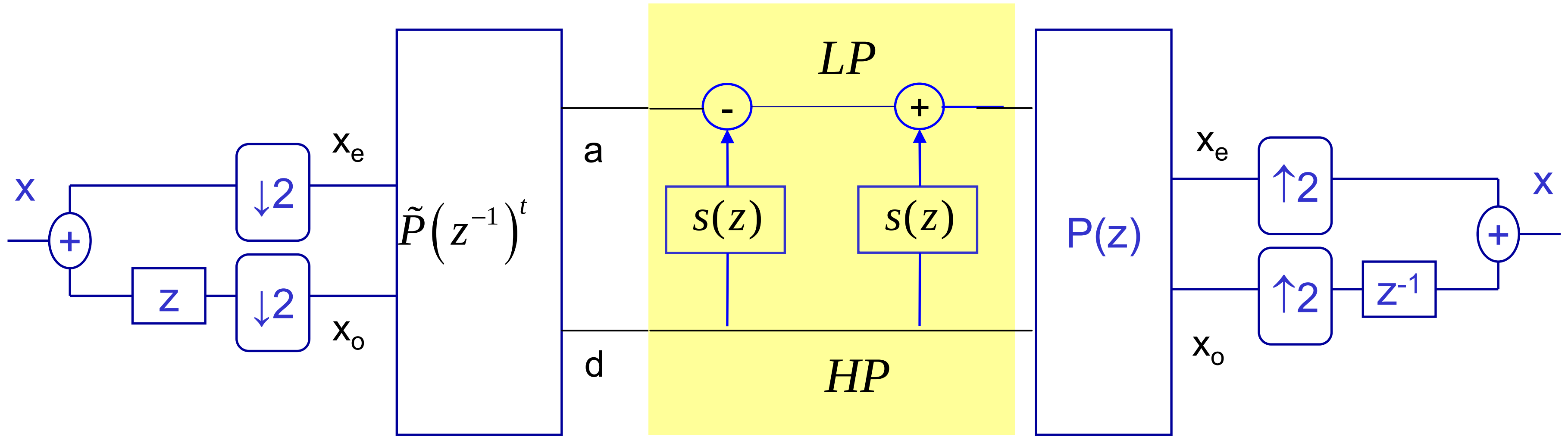


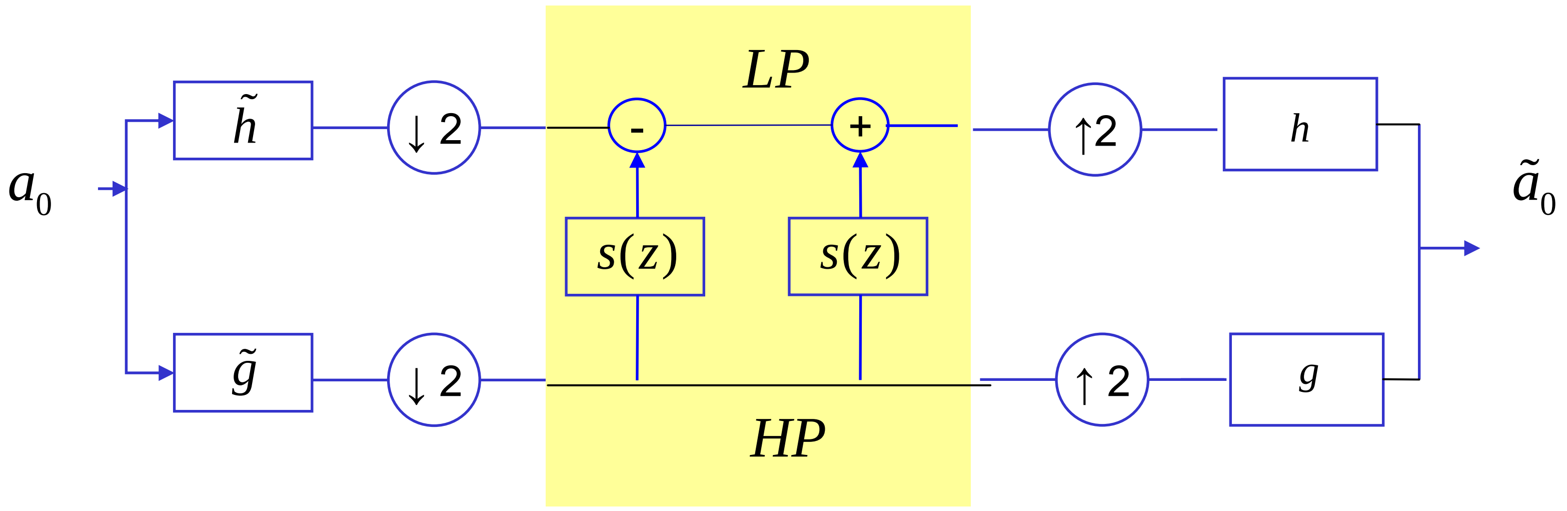
Equivalent representations



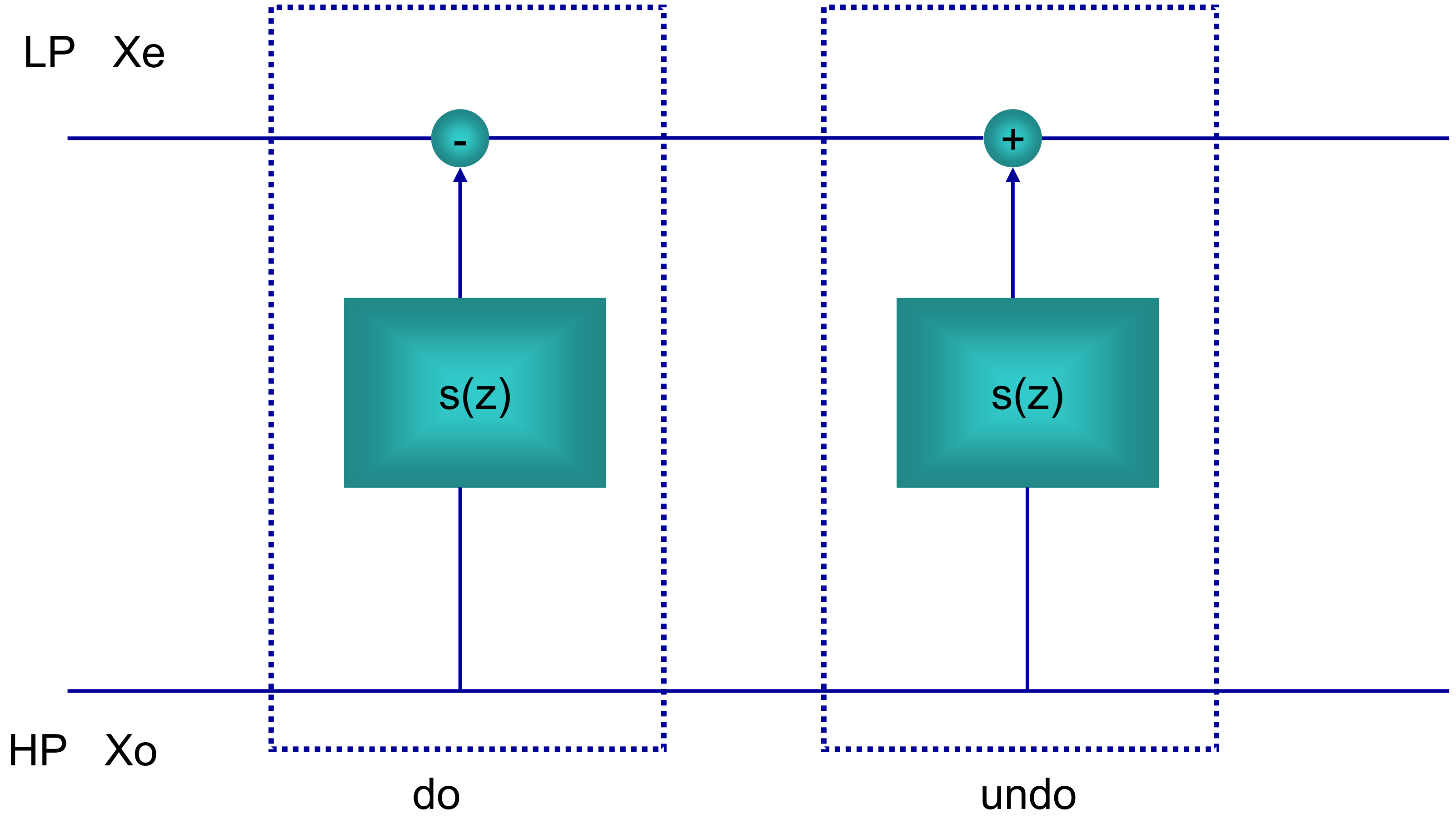


Equivalent representations





Towards lossless



Dual lifting

- Teorem 4. Let (h,g) be complementary. Then any other filter $h_{new}(z)$ complementary to g is of the form

$$h^{new}(z) = h(z) + g(z)\,t(z^2)$$

where t(z) is a Laurent polynomial. Conversely, any filter of this form is complementary to g

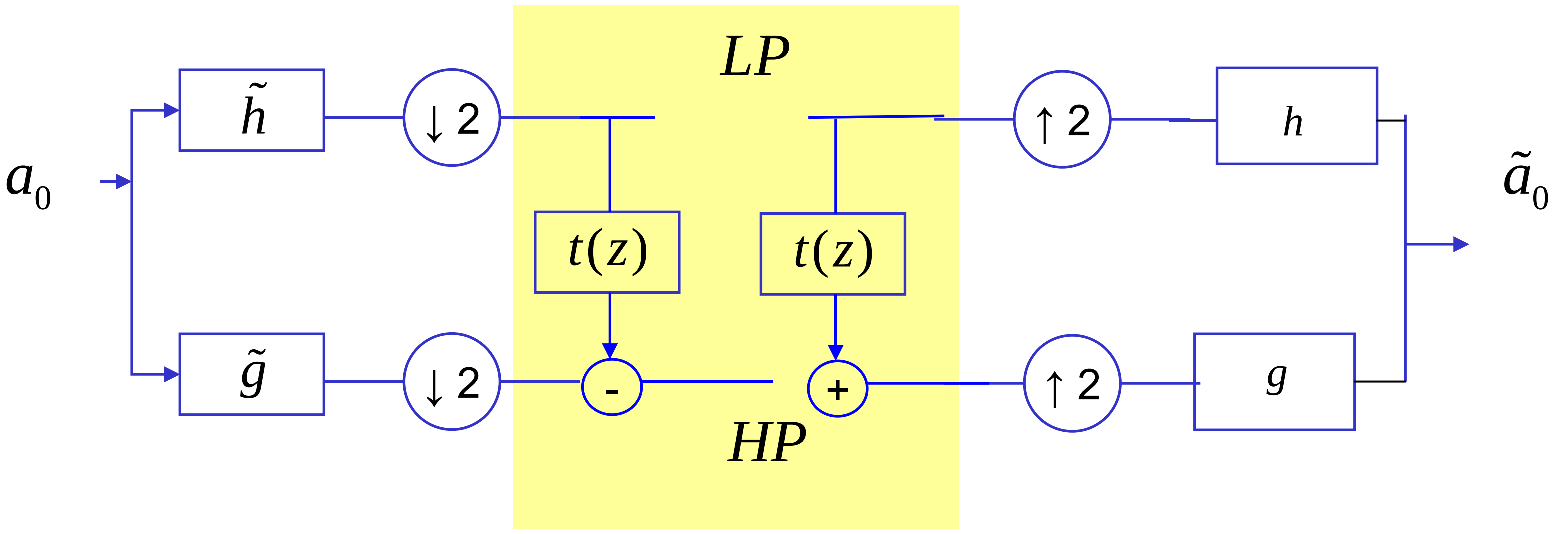
  - New polyphase matrix

$$P^{new}(z) = P(z)\begin{bmatrix} 1 & 0 \\ t(z) & 1 \end{bmatrix}$$

  - Dual lifting creates a new $\tilde{g}$ given by

$$\tilde{g}^{new}(z) = \tilde{g}(z) - \tilde{h}(z)\,t(z^{-2})$$

Dual lifting



Prediction

- *Prediction* steps: the HP coefficients are shaped (lifted) by filtering the LP ones by the filter t(z)
- *Update* steps: the LP coefficients are shaped by filtering the HP ones by s(z)
- One can start from the lazy wavelet and use lifting to gradually build one's way up to a multiresolution analysis with particular properties

lifting and dual lifting

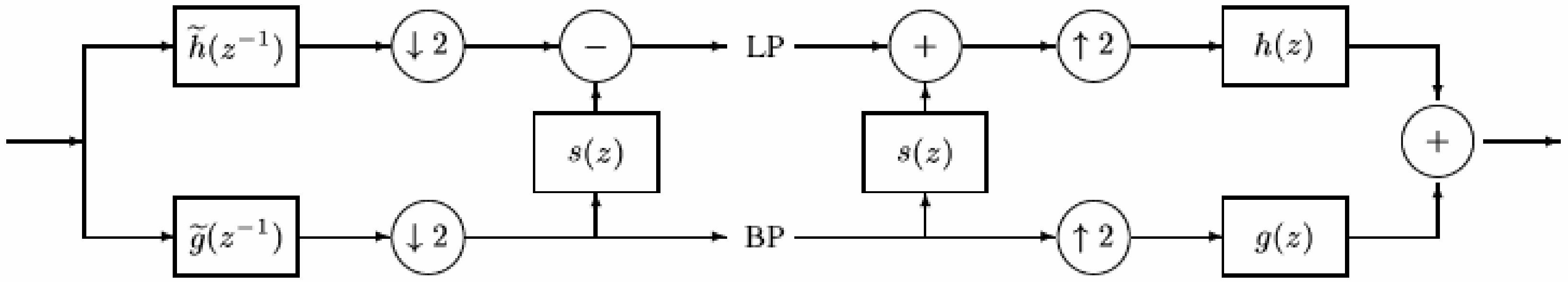


FIGURE 5. *The lifting scheme: First a classical subband filter scheme and then lifting the low-pass subband with the help of the high-pass subband.*

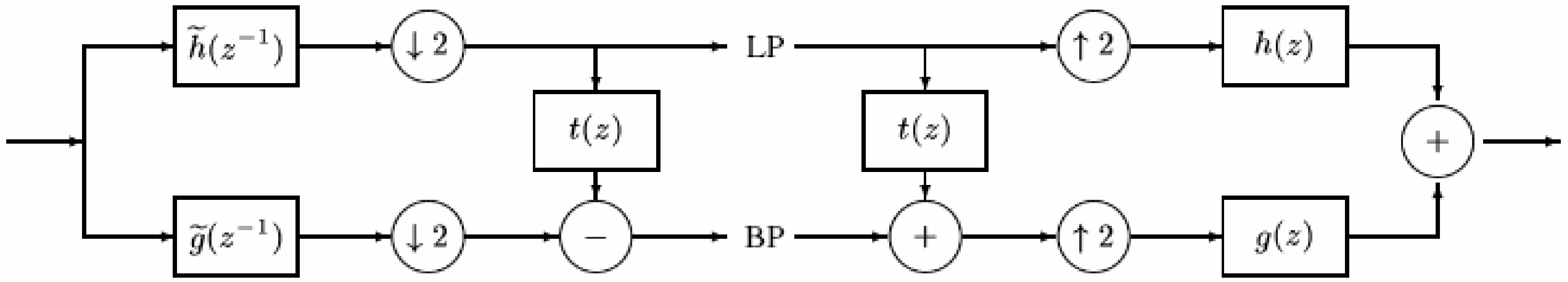


FIGURE 6. *The dual lifting scheme: First a classical subband filter scheme and later lifting the high-pass subband with the help of the low-pass subband.*

lazy wavalet

The problem of finding an FIR wavelet transform thus amounts to finding a matrix $P(z)$ with determinant one. Once we have such a matrix, $\widetilde{P}(z)$ and the four filters for the wavelet transform follow immediately. From (2) and Cramer's rule it follows that

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

This implies

$$\widetilde{g}(z) = z^{-1}\, h(-z^{-1}) \quad \text{and} \quad \widetilde{h}(z) = -z^{-1}\, g(-z^{-1}).$$

The most trivial example of a polyphase matrix is $P(z) = \mathbf{I}$. This results in $h(z) = \widetilde{h}(z) = 1$ and $g(z) = \widetilde{g}(z) = z^{-1}$. The wavelet transform then does nothing else but subsampling even and odd samples. This transform is called the polyphase transform, but in the context of lifting it is often referred to as the Lazy wavelet transform [44]. (The reason is that the notion of the Lazy wavelet can also be used in the second generation setting.)

Lifted basis functions

- Lifting

$$\tilde{h}^{new}(z) = \tilde{h}(z) - \tilde{g}(z)\,s\left(z^{-2}\right)$$

$$g^{new}(z) = g(z) + \underbrace{h(z)}_{\downarrow}\,s\left(z^{2}\right)$$

does not change → lifting the wavelet through s(z)

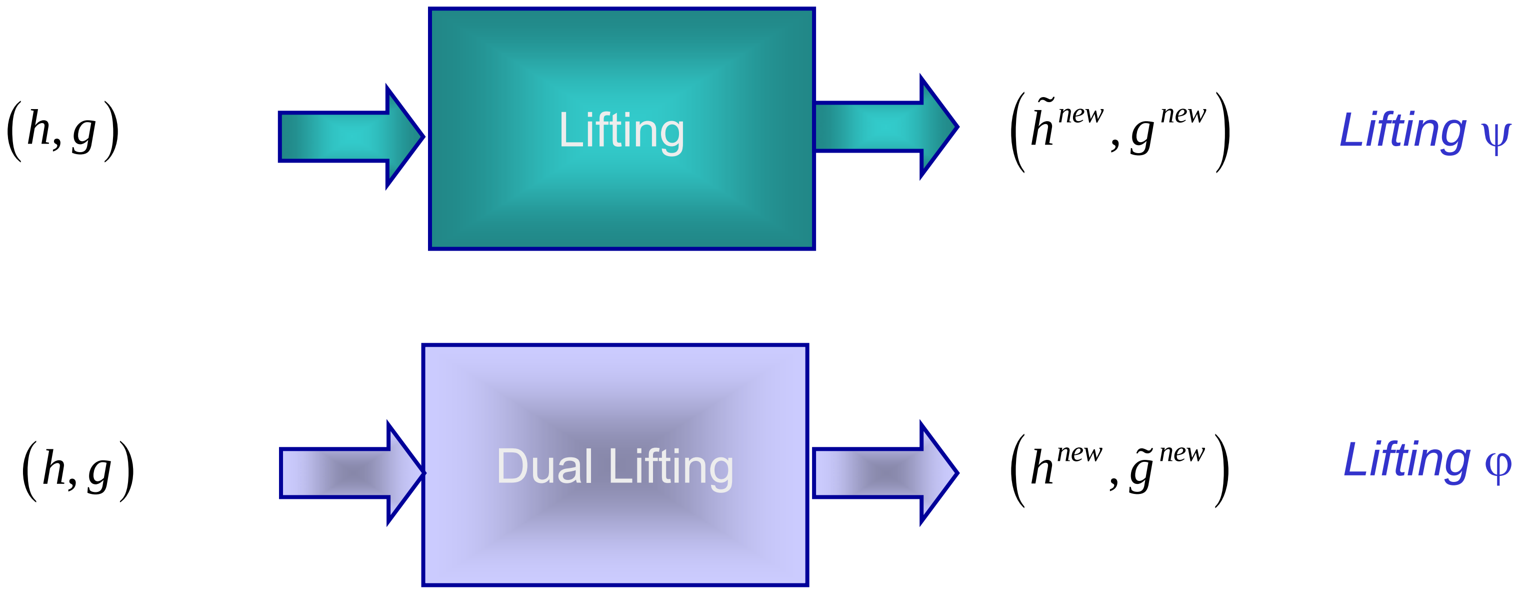
- Dual lifting

does not change → lifting the basis function through t(z)

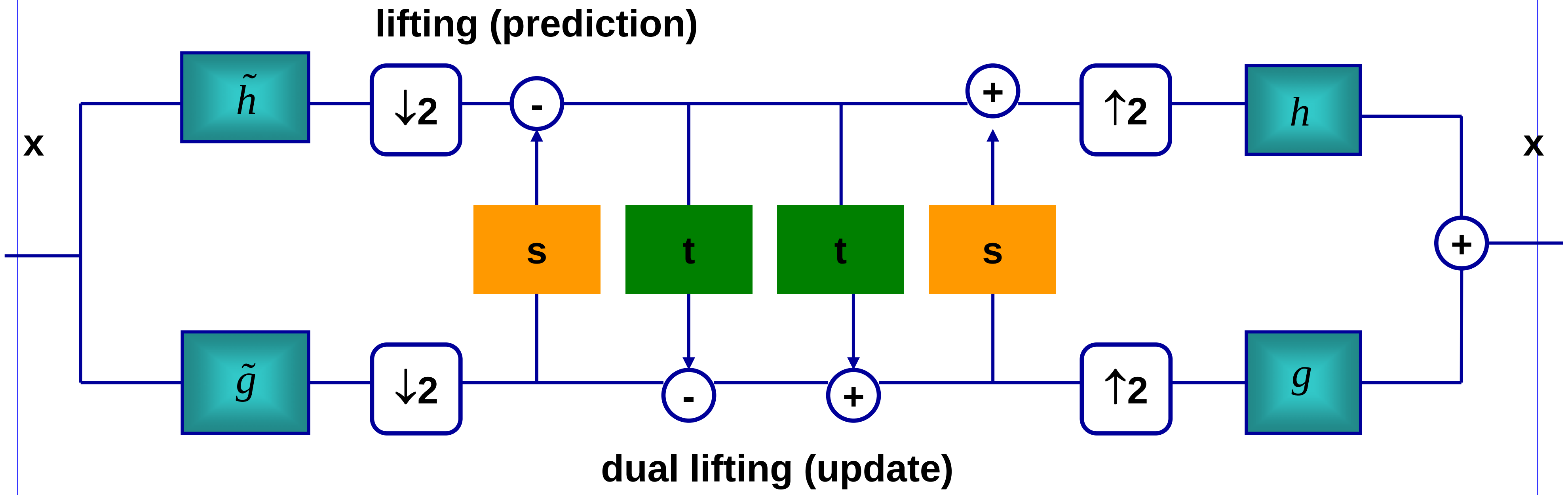
$$h^{new}(z) = h(z) + \overbrace{g(z)}^{\uparrow}\,t\left(z^{2}\right)$$

$$\tilde{g}^{new}(z) = \tilde{g}(z) - \tilde{h}(z)\,t\left(z^{-2}\right)$$

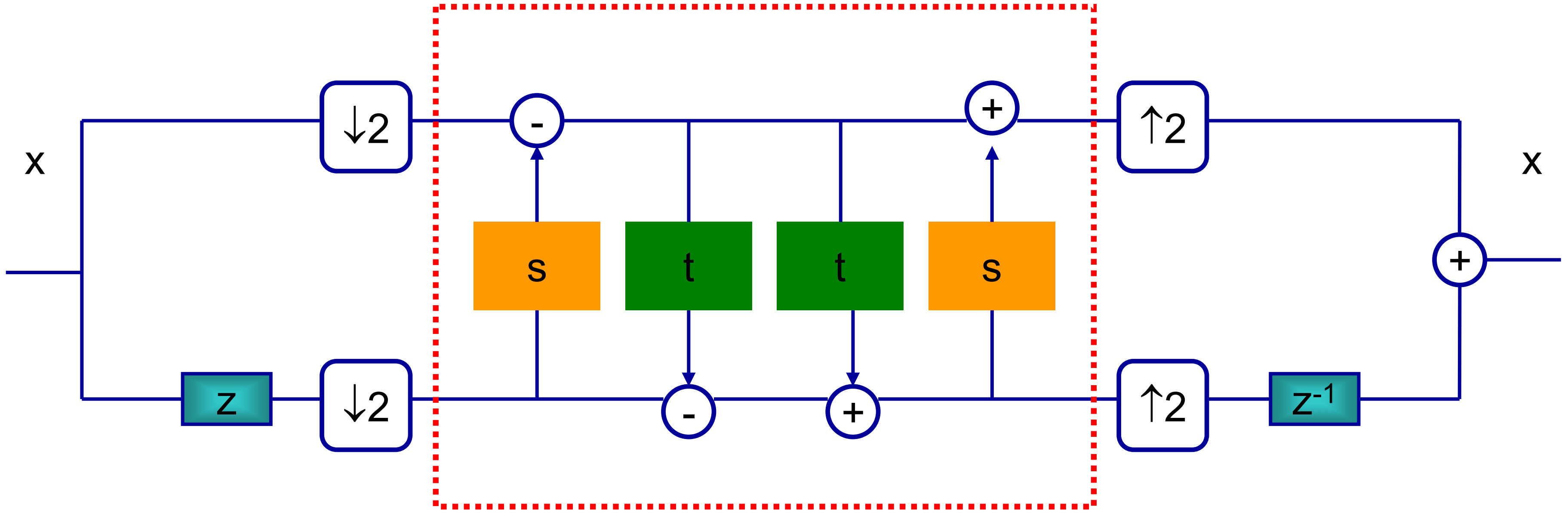
Global Lifting



Cakewalk construction



Lifting the Lazy wavelet



Every finite wvt can be obtained with a cakewalk starting from the Lazy wavelet

Lifting theorem

- Theorem 7. Given a complementary filter pair (h,g), then there always exist Laurent polynomials $s_i(z)$ and $t_i(z)$ for i=1,...,m and a non-zero constant K so that

$$P(z)=\prod_{i=1}^{m}\begin{bmatrix}1 & s_i(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ t_i(z) & 1\end{bmatrix}\begin{bmatrix}K & 0\\ 0 & 1/K\end{bmatrix}$$

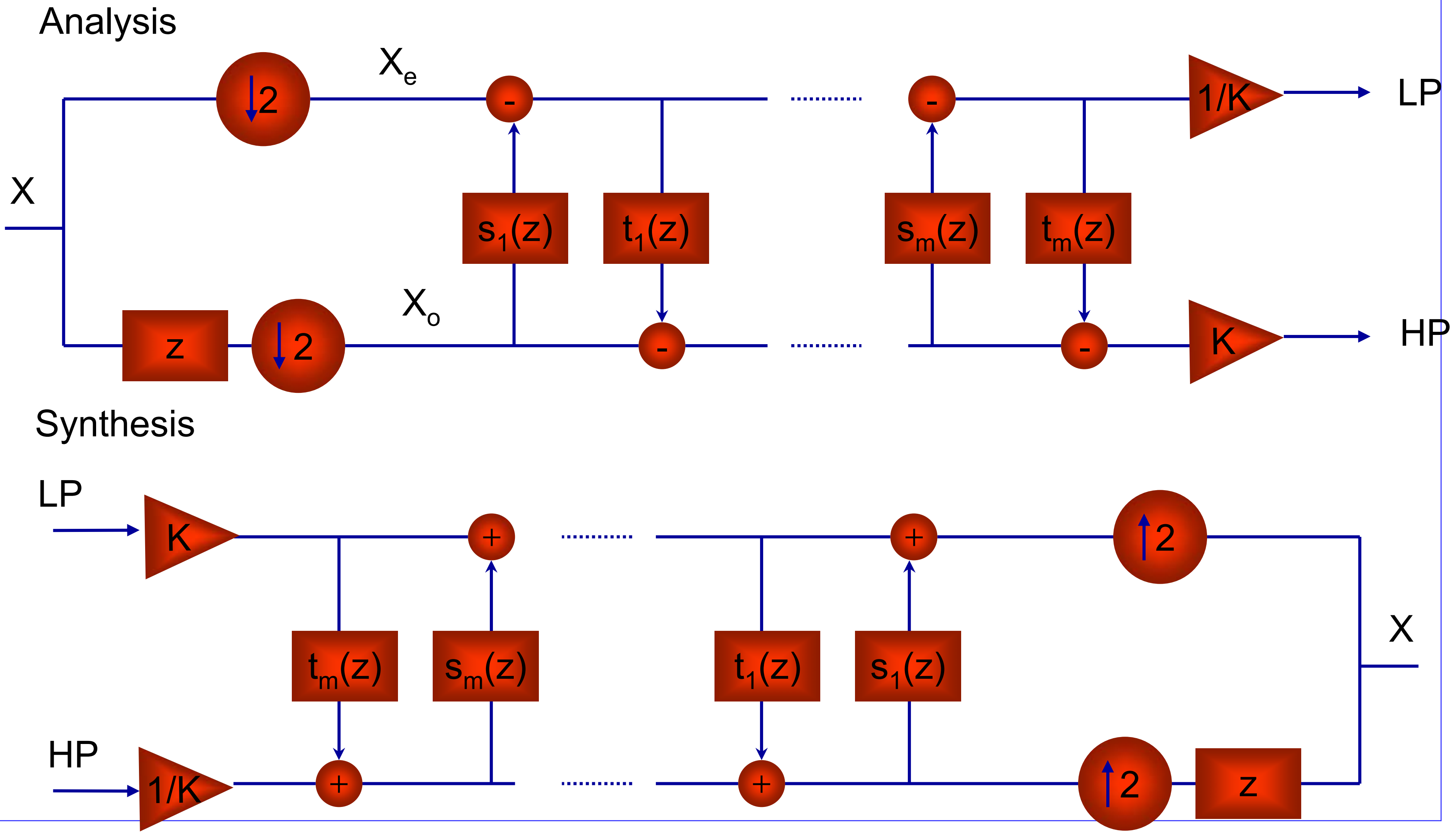
  - The dual polyphase matrix is given by

$$\tilde{P}(z)=\prod_{i=1}^{m}\begin{bmatrix}1 & 0\\ -s_i(z^{-1}) & 1\end{bmatrix}\begin{bmatrix}1 & -t_i(z^{-1})\\ 0 & 1\end{bmatrix}\begin{bmatrix}1/K & 0\\ 0 & K\end{bmatrix}$$

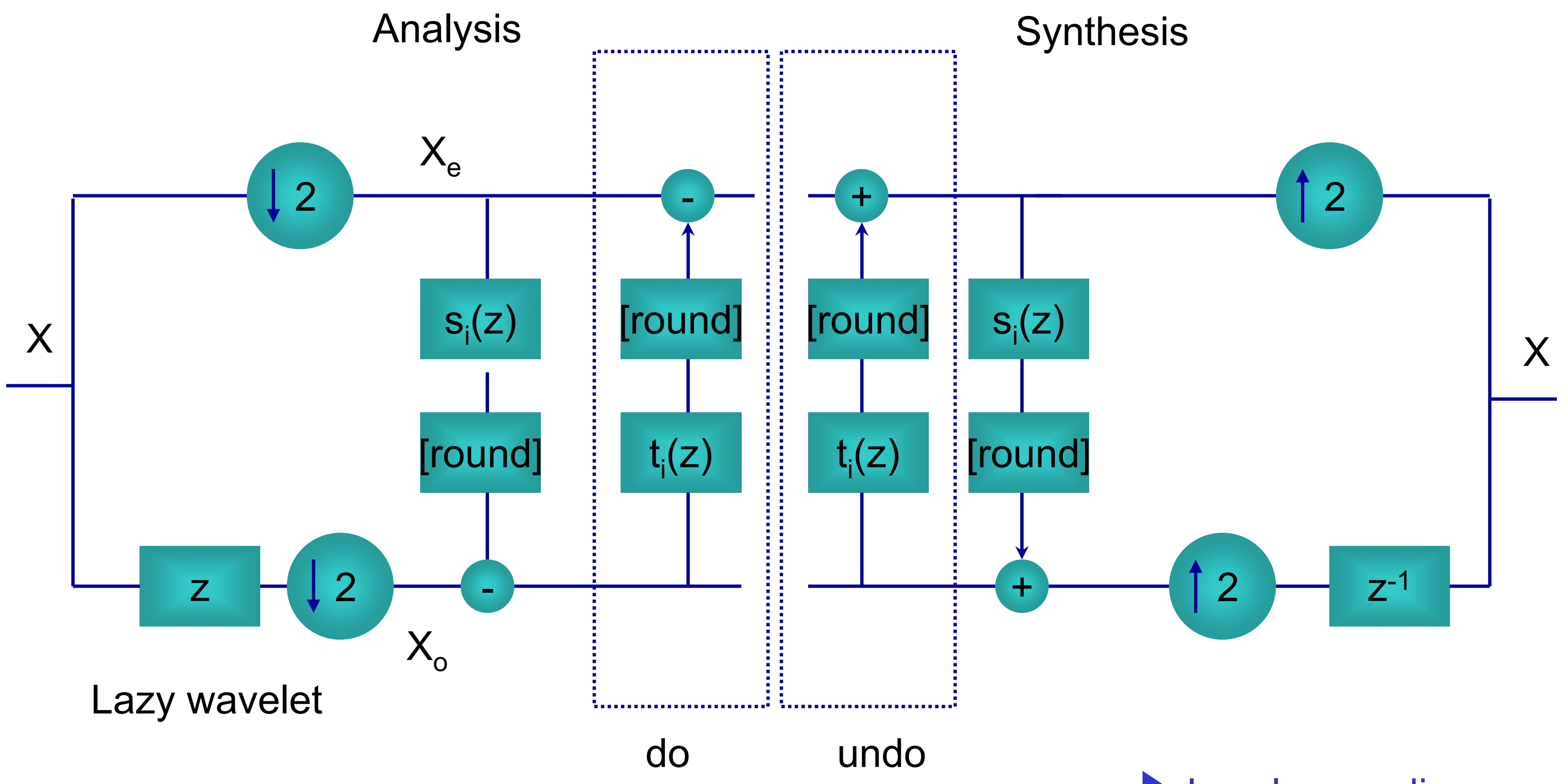
- Every finite filter wavelet transform can be obtained by starting with the lazy wavelet followed by m lifting and dual lifting steps, followed by a scaling

- The prediction and update steps are found by factorization of the polyphase matrix

Implementation



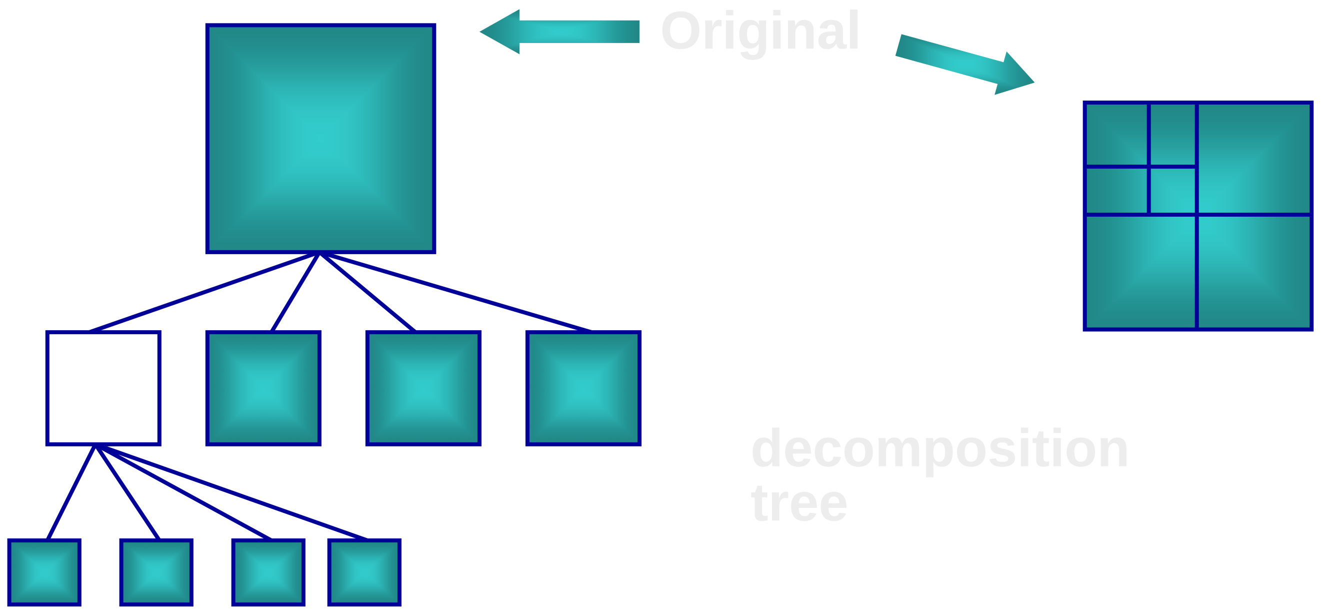
Integer wavelet transform



⇒ Lossless coding

Fully in-place implementation

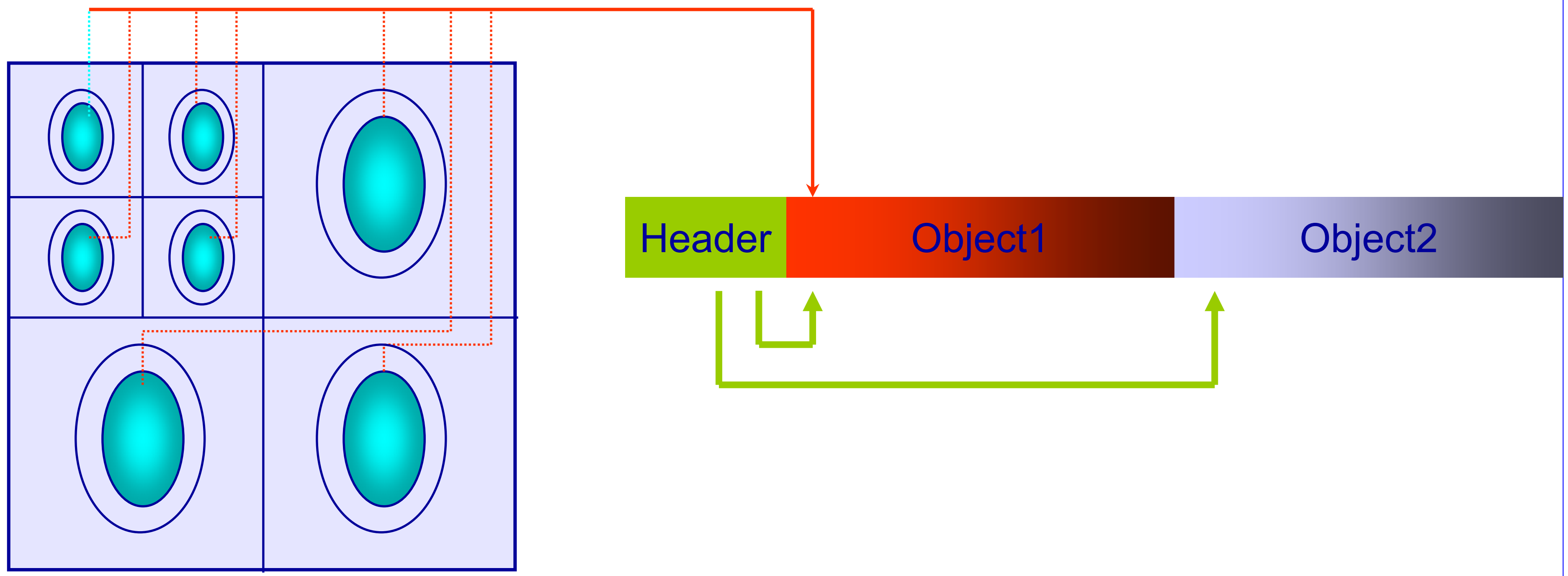
- Odd samples are used to *predict* even samples and viceversa
  - The original memory locations can be overwritten



Summary

- Biorthogonal (FIR) wavelets
- Perfect reconstruction ensured for any signal extension at borders
- Faster, fully *in-place* implementation
- Reduced computational complexity
- *Non-linear* lifting
- All operations within one lifting step can be done entirely parallel while the only sequential part is the order of the lifting operations
- Allows wavelets mapping integers to integers, important for hardware implementation and lossless coding
- Allows for adaptive wavelet transforms (i.e. wavelets on the sphere)

Application: Object-based coding



Border dimension

Appendix

Laurent polynomials

[sweldens paper]

Filters and Laurent polynomials

A *filter* h is a linear time invariant operator and is completely determined by its *impulse response*: $\{h_k \in \mathbf{R} \mid k \in \mathbf{Z}\}$. The filter h is a Finite Impulse Response (FIR) filter in case only a finite number of filter coefficients h_k are non-zero. We then let k_b (respectively k_e) be the smallest (respectively largest) integer number k for which h_k is non-zero. The z-transform of a FIR filter h is a Laurent polynomial $h(z)$ given by

$$h(z) = \sum_{k=k_b}^{k_e} h_k\, z^{-k}.$$

In this paper, we consider only FIR filters. We often use the symbol h to denote both the filter and the associated Laurent polynomial $h(z)$. The *degree* of a Laurent polynomial h is defined as

$$|h| = k_e - k_b.$$

So the length of the filter is the degree of the associated polynomial plus one. Note that the polynomial z^p seen as a Laurent polynomial has degree zero, while as a regular polynomial it would have degree p. In order to make consistent statements, we set the degree of the zero polynomial to $-\infty$.

The set of all Laurent polynomials with real coefficients has a commutative ring structure. The sum or difference of two Laurent polynomials is again a Laurent polynomial. The product of a Laurent polynomial of degree l and a Laurent polynomial of degree l' is a Laurent polynomial of degree $l + l'$. This ring is usually denoted as $\mathbf{R}[z, z^{-1}]$.

Laurent polynomials

Within a ring, exact division is not possible in general. However, for Laurent polynomials, division with remainder is possible. Take two Laurent polynomials $a(z)$ and $b(z) \neq 0$ with $|a(z)| \geqslant |b(z)|$, then there always exists a Laurent polynomial $q(z)$ (the quotient) with $|q(z)| = |a(z)| - |b(z)|$, and a Laurent polynomial $r(z)$ (the remainder) with $|r(z)| < |b(z)|$ so that

$$a(z) = b(z)\,q(z) + r(z).$$

We denote this as (C-language notation):

$$q(z) = a(z)\,/\,b(z) \quad \text{and} \quad r(z) = a(z)\,\%\,b(z).$$

If $|b(z)| = 0$ which means $b(z)$ is a monomial, then $r(z) = 0$ and the division is exact. A Laurent polynomial is invertible if and only if it is a monomial. This is the main difference with the ring of (regular) polynomials where constants are the only polynomials that can be inverted. Another difference is that the long division of Laurent polynomials is not necessarily unique. The following example illustrates this.

Laurent polynomials

Example 1. Suppose we want to divide $a(z) = z^{-1} + 6 + z$ by $b(z) = 4 + 4\,z$. This means we have to find a Laurent polynomial $q(z)$ of degree 1 so that $r(z)$ given by

$$r(z) = a(z) - b(z)\,q(z)$$

is of degree zero. This implies that $b(z)\,q(z)$ has to match $a(z)$ in two terms. If we let those terms be the term in z^{-1} and the constant then the answer is $q(z) = 1/4\,(z^{-1} + 5)$. Indeed,

$$r(z) = (z^{-1} + 6 + z) - (4 + 4\,z)(1/4\,z^{-1} + 5/4) = -4\,z.$$

The remainder thus is of degree zero and we have completed the division. However if we choose the two matching terms to be the ones in z and z^{-1}, the answer is $q(z) = 1/4\,(z^{-1} + 1)$. Indeed,

$$r(z) = (z^{-1} + 6 + z) - (4 + 4\,z)(1/4\,z^{-1} + 1/4) = 4.$$

Finally, if we choose to match the constant and the term in z, the solution is $q(z) = 1/4\,(5\,z^{-1} + 1)$ and the remainder is $r(z) = -4\,z^{-1}$.