# Wavelet transforms, lifting

© 2011-2016 Josef Pelikán
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz
http://cgg.mff.cuni.cz/~pepca/

# Wavelet transforms

- traditional approaches to wavelet transform:
  - **scaling functions and wavelets** – dilation and translation of a wavelet function
  - **filter banks** – low-pass and high-pass filtering
- the approaches produce the same results (proof by Ingrid Daubechies)
- filter bank approach is more popular in signal processing and compression

# Lifting basics

- building sparse data representation in spatial domain
- original discrete signal: $(x_k)_{k \in Z},\ x_k \in R$
- splitting into polyphase components, **„evens“** and **„odds“**:

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

- $x_e$ and $x_o$ use to be closely **correlated**
  - chance of a good **prediction P** (d .. detailed/difference)

$$d = x_o - P(x_e), \quad \text{recovering: } x_o = P(x_e) + d$$

# Prediction example

- ideal for nearly linear data

  $d_k = x_{2k+1} - (x_{2k} + x_{2k+2}) / 2$

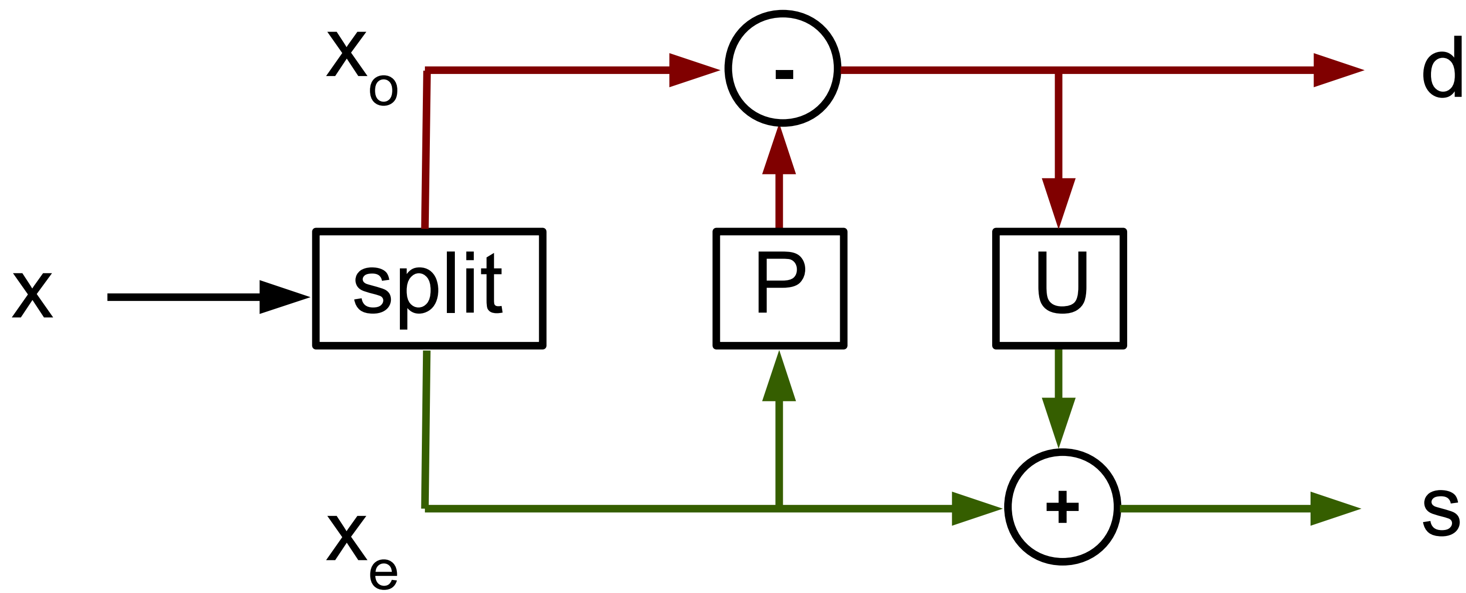
- **zero** detail coefficients for locally linear data
- computing **d** ... „lifting step“
- idea of DPCM .. $(x_e, x_o) \rightarrow (x_e, d)$
- subsampling $(x \rightarrow x_e)$ .. poor frequency separation (aliasing)
  - e.g. different **running average** and other **moments**

# Update operator U

- $2^{nd}$ phase of lifting..

  $s \;=\; x_e + U(d)$

- recovering of $x_e$: $x_e \;=\; s - U(d)$



# Update example

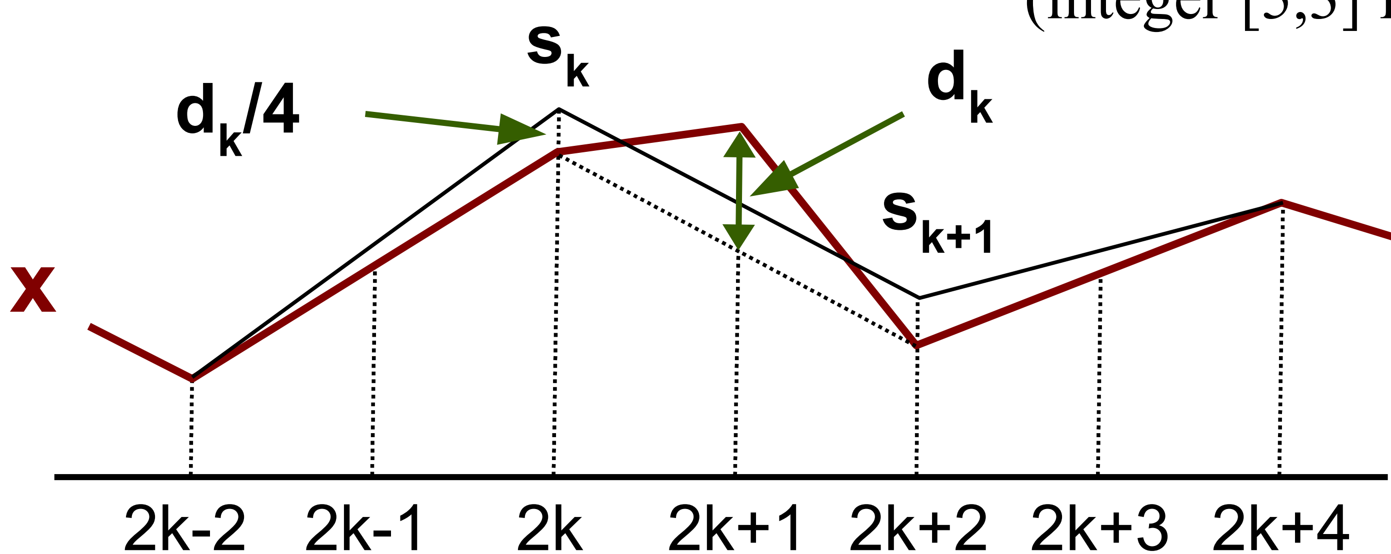
- linear prediction

$$d_k = x_{2k+1} - (x_{2k} + x_{2k+2}) / 2$$

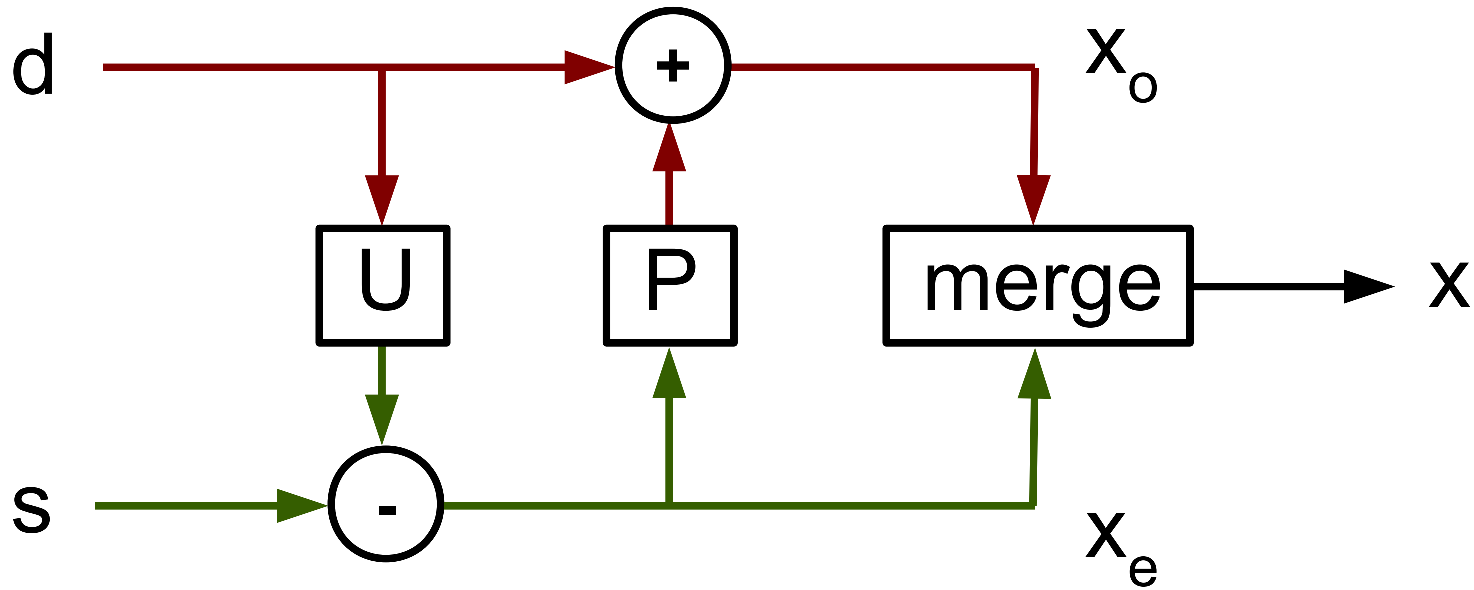
- update operator should restore the running average:

$$s_k = x_{2k} + (d_{k-1} + d_k) / 4$$

.. biorthogonal (2,2) wavelet (integer [5,3] filter)



# Inverse transform




- **exact inverse!**
  - as long as „+“ and „-“ are inverse operations
- P() and U() could be **integer** … losless transform

# Haar wavelet

- prediction

  $$d_k = x_{2k+1} - x_{2k}$$

- update:

  $$s_k = x_{2k} + d_k / 2$$

# Daubechies (9,7) filter

- transformed to integer arithmetic:

$$d_i^0 = x_{2i+1}\ ,\ s_i^0 = x_{2i}$$

$$d_i^1 = d_i^0 + \left\lfloor \alpha \times (s_i^0 + s_{i+1}^0) + \frac{1}{2} \right\rfloor,\ s_i^1 = s_i^0 + \left\lfloor \beta \times (d_{i-1}^1 + d_i^1) + \frac{1}{2} \right\rfloor$$

$$d_i^2 = d_i^1 + \left\lfloor \gamma \times (s_i^1 + s_{i+1}^1) + \frac{1}{2} \right\rfloor,\ s_i^2 = s_i^1 + \left\lfloor \delta \times (d_{i-1}^2 + d_i^2) + \frac{1}{2} \right\rfloor$$

$$d_i = K_1 \times d_i^2\ ,\ s_i = K_0 \times s_i^2$$

$\alpha = -1.586134342$ , $\beta = -0.05298011854$,
$\gamma = 0.8829110762$ , $\delta = 0.4435068522$,
$K_0 = K = 1.149604398$, $K_1 = K^{-1}$

# References


- Daubechies I., Sweldens W.: ***Factoring Wavelet Transforms into Lifting Steps***, Technical report, Bell Laboratories, revised Nov 1997
- Calderbank A.R., Daubechies I., Sweldens W., Yeo B.-L.: ***Wavelet Transforms That Map Integers to Integers***, Applied and Computational Harmonic Analysis, 5(3), pp 332–369, 1998