ECE6258 Lecture 27

Lifting implementations of wavelets

Motivation

- JPEG 2000 allows for a *lifting* or a *FWT* implementation of the wavelet transform.
- Lifting is an alternative implementation of a wavelet implementation.
  - Does not require explicit wavelet design.
  - Computationally and memory efficient
  - Allows generalizations to
    - Non-separable 2-D systems
    - Nonlinear filters
    - Incorporation of boundary conditions …

Motivating Example: Haar wavelet

- Let *a* and *b* be neighboring samples of a two point sequence.
- If they are correlated, it should be more efficient to encode their average and difference, than the values themselves.

transformation

$$s = \frac{a+b}{2}$$

$$d = b - a$$

inverse

$$a = s + \frac{d}{2}$$

$$b = s - \frac{d}{2}$$

Motivating Example (2)

- Now consider a signal $s_n[l]$ of 2^n values.

$$s_n = \{s_n[l], l = 0,1,\ldots 2^n - 1\}$$

- We apply this average and difference idea to each pair.

$$s_{n-1}[l] = \frac{s_n[2l] + s_n[2l+1]}{2}$$

$$d_{n-1}[l] = s_n[2l+1] - s_n[2l]$$

- Note that the average and difference sequences are each 2^{n-1} points long.

Motivating example (3)

- s_{n-1} is a coarser representation of the signal
- d_{n-1} is the information needed to go from the coarser representation to a finer one.
- We can repeat this decomposition on the signal s_{n-1}.
- In the limit
  - s_0 – 1 sample
  - d_j – 2^j samples, $j = 0,1,\ldots,n-1$
  - Total number of samples is

$$1+\sum_{j=0}^{n-1} 2^j = 2^n$$

In-place computation

- Suppose we want to compute the Haar transform as an in-place computation, i.e., store *s* on top of *a* and *d* on top of *b*.

$$a+b \to a$$
$$b-a \to b$$

Produces wrong result !

- Alternatively, we can reverse the order of the computations

$$b-a = d \to b$$
$$s = a+\frac{d}{2} \to a$$

Correct result !

$$a+\frac{d}{2} = a+\frac{b-a}{2} = \frac{a+b}{2}$$

Inversion

- Summarizing the algorithm (in pseudo-C)

$$b -= a;$$
$$a += b/2;$$

Forward

- To get the inverse, we run the code backwards (reverse the order and flip the signs.)

$$a -= b/2;$$
$$b += a;$$

Inverse

Lifting

- The lifting scheme involves three operations.
  - Split:

    $(\text{even}_{j-1}, \text{odd}_{j-1}) := \text{split}(s_j);$

  - Note: In the Haar example, *a* was an even sample and *b* was an odd sample.

Lifting (2)

- Predict:

$$d_{j-1} = \text{odd}_{j-1} - P(\text{even}_{j-1})$$

  - The operator *P*( ) predicts a sample in the odd sequence from the even sequence.
  - For the Haar sequence *P*(*a*)=*a* and the above equation reduces to

$$d_{j-1}[l] = s_{j-1}[2l+1] - s_{j-1}[2l]$$

correction — odd sample — prediction

Lifting (3)

- Update:
  - We require that the spaces defined by the scaling functions at different scales be nested.
  - One way to guarantee this is to require that the average values (DC value) of the approximation signals be the same, i.e.,

$$S = 2^{-j} \sum_{l=0}^{2^j - 1} s_j[l] \neq S(j)$$

  - This implies that $s_0[0]$ is the DC value.
  - This condition was assured by

$$s_{j-1}[l] = s_j[2l] + d_{j-1}[l]/2$$

Lifting (4)

- Proof:

$$\sum_{l=0}^{2^j-1} s_{j-1}[l] = \sum_{l=0}^{2^j-1} (s_j[2l] + d_{j-1}[l]/2)$$

$$= \frac{1}{2}\sum_{l=0}^{2^j-1} (s_j[2l] + s_j[2l+1]) = \frac{1}{2}\sum_{i=0}^{2^j-1} s_j[l]$$

- This defines an update operator

$$s_{j-1} = \text{even}_{j-1} + U(d_{j-1})$$

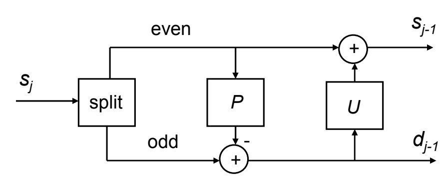
Lifting (5)

- Summary:
  - Analysis:

(odd$_{j-1}$,even$_{j-1}$) := split(s_j);

odd$_{j-1}$ -= *P*(even$_{j-1}$);

even$_{j-1}$ += *U*(odd$_{j-1}$);



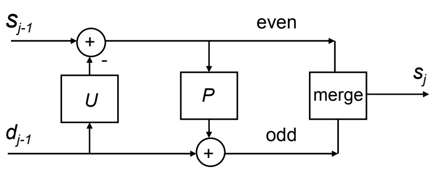
Lifting (6)

- Synthesis

  even_{j-1} -= $U(\text{odd}_{j-1})$;

  odd_{j-1} += $P(\text{even}_{j-1})$;

  s_j := merge(odd_{j-1},even_{j-1})



Moving onward

- One way to build other wavelet transforms is by using different prediction and/or update steps.
  - Haar predictor correct for inputs that are constant.
    - Eliminates zeroth-order correlation.
    - Update preserves zeroth-order moment (mean).
  - One way to get better wavelets is to find predictors that remove higher order correlations and updates that preserve more moments.

Linear Wavelets (1)

- Consider a predictor and update of order two.
  - Predictor will be exact if input is a straight line.
  - Update will preserve the mean and the first-moment.
- Let the predictor of an odd sample be the average of the two adjacent even samples

$$P(s_j) = \frac{1}{2}\left(s_j[2l] - s_j[2l+2]\right)$$

Linear wavelets (2)

- In the update stage to insure that the average of the signal is preserved, we require

$$\sum_l s_{j-1}[l] = \frac{1}{2}\sum_l s_j[l]$$

- Since we must update the even samples using the previously computed detail samples, we hypothesize an update of the form:

$$s_{j-1}[l] = s_j[2l] + A\left(d_{j-1}[l-1] + d_{j-1}[l]\right)$$

- To find A, we compute the average

$$\sum_l s_{j-1}[l] = \sum_l s_j[2l] + 2A\sum_l d_{j-1}[l] = (1-2A)\sum_l s_j[2l] + 2A\sum_l s_j[2l+1]$$

Linear wavelets (3)

- For this to be true, we must have

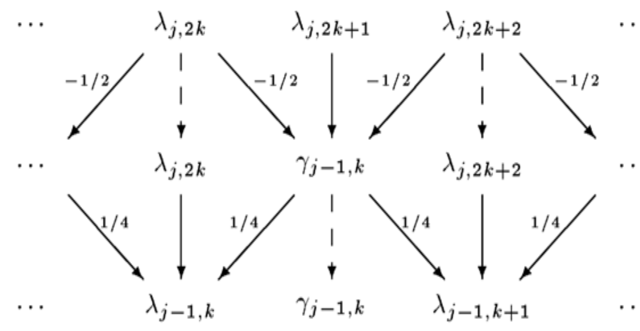
$$A = \frac{1}{4}$$

- It is straightforward to verify that for this update, the first moment is also preserved

$$\sum_l l s_{j-1}[l] = \frac{1}{2} \sum_l l s_j[l]$$

- This is equivalent to the (2,2) biorthogonal Cohen-Daubechies-Feauveau wavelet.

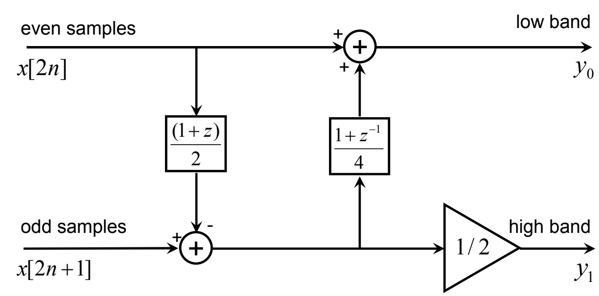
In-place computation



Advantages of lifting approach

- Perfect reconstruction guaranteed
- Wavelet design reduced to design of predictors and updates (no Fourier methods).
- Lifting approach more general
  - Predictors can be nonlinear.
  - 2-D nonseparable predictors straightforward.
  - Predictors can be adaptive, e.g., motion-compensated prediction.
  - Can be extended to new settings
    - Hexagonally sampled arrays
    - Arbitrarily-shaped objects
    - Curves, surfaces, volumes

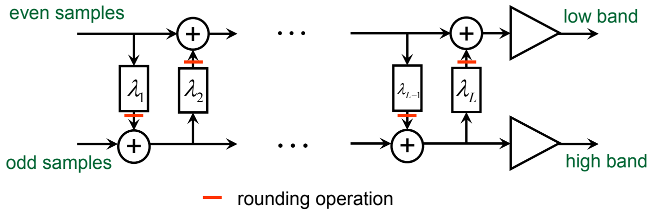
Example: lifting implementation of 5/3 filters



Verify by computing impulse response in even and odd input channel.

Reversible subband transform

- Observation: lifting operators can be nonlinear.
- Incorporate the necessary rounding into the lifting operator.



- Used in JPEG2000 as part of the 5/3 biorthogonal wavelet transform.