Recap

Haar

- simple and fast wavelet transform

Limitations

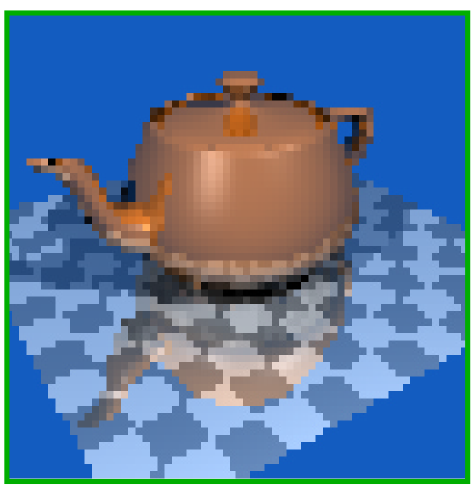
- not smooth enough: blocky

How to improve?

- classical approach: basis functions
- Lifting: transforms

1

Erasing Haar Coefficients



2

Classical Constructions

Fourier analysis

- regular samples, infinite setting
- analysis of polynomials

Conditions:

- smoothness
- perfect reconstruction

But...

- Fourier analysis not always applicable

3

Lifting Scheme

Custom design construction

- entirely in spatial domain

Second generation wavelets

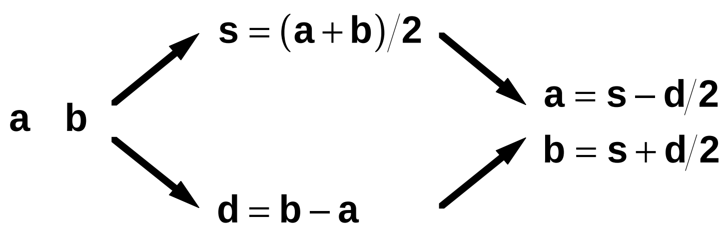
- boundaries
- irregular samples
- curves, surfaces, volumes

4

Haar Transform

Averages and differences

- two neighboring samples



5

Haar Transform

In-place Version

- want to overwrite old values with new values
- rewrite

$$d = b - a \qquad s = a + d/2$$

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

- inverse: run code backwards!

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

6

Haar Transform

Forward

```
for( s = 2; s <= n; s *= 2 )
  for( k = 0; k < n; k += s ){
    c[k+s/2] -= c[k];
    c[k] += c[k+s/2] / 2;
  }
```

7

Haar Transform

Inverse

```
for( s = n; s >= 2; s /= 2 )
  for( k = 0; k < n; k += s ){
    c[k] -= c[k+s/2] / 2;
    c[k+s/2] += c[k];
  }
```

8

Haar Transform

Lifting version

- split into even and odd

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

- predict and store difference: detail coefficient

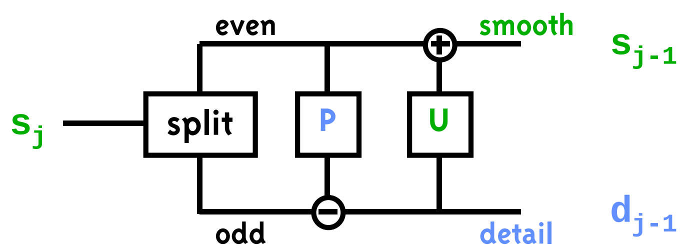
  $d_{j-1} = odd_{j-1} - even_{j-1}$

- update even with detail: smooth coefficient

  $s_{j-1} = even_{j-1} + d_{j-1}/2$

9

Haar Transform



$$d_{j-1} = odd_{j-1} - P(even_{j-1})$$

$$s_{j-1} = even_{j-1} + U(d_{j-1})$$

10

Haar Transform

Predict

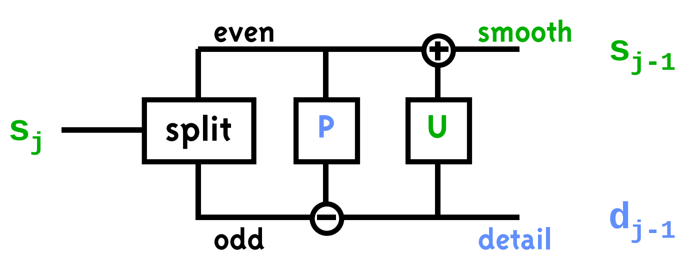
- perfect if function is constant
  - detail coefficients zero
- removes constant correlation

Update

- preserve averages of coarser versions
- avoid aliasing
- obtain frequency localization

11

Haar Transform

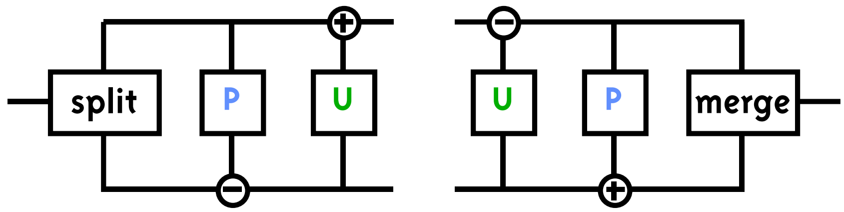


12

Lifting Scheme

Advantages

- in-place computation
- efficient, general
- parallelism exposed
- easy to invert



13

Lifting

Build more powerful versions

- higher order prediction
  - Haar has order 1
- higher order update
  - preserve more *moments* of coarser data

An example

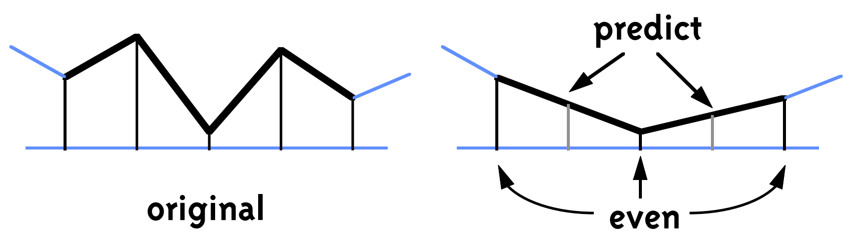
- linear wavelet transform

14

Linear Prediction

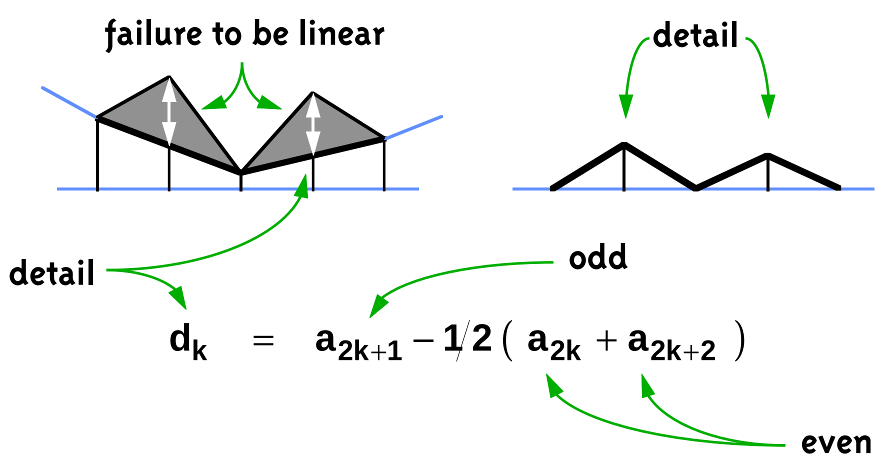
Use even on either side

- keep difference with prediction
- exploit more coherence/smoothness/ correlation



15

Prediction



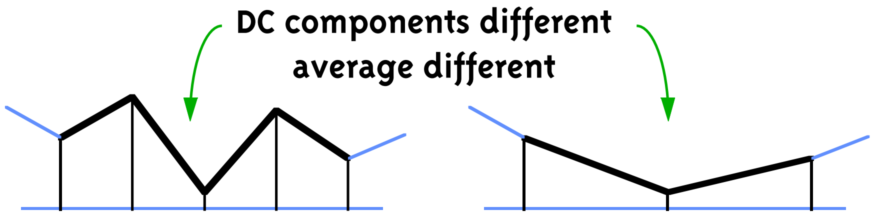
$$d_k = a_{2k+1} - 1/2\,(a_{2k} + a_{2k+2})$$

16

Update

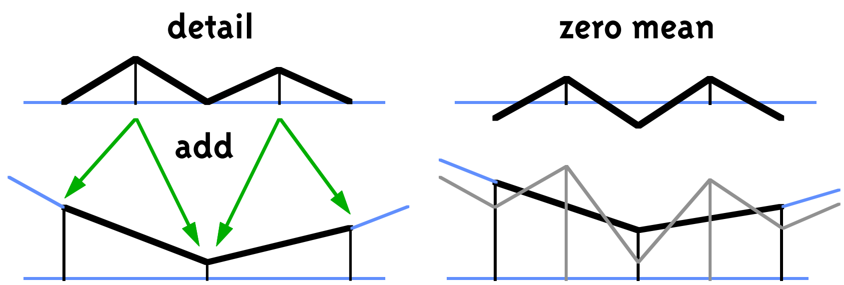
Even values are subsampled

- aliasing!



17

Update



even

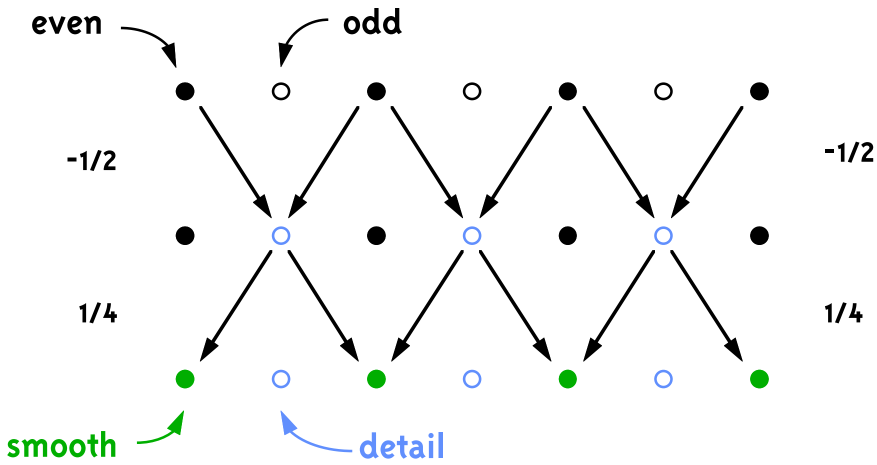
$$s_k = a_{2k} + 1/4\,(\,d_{k-1} + d_k\,)$$

smooth

detail

18

Inplace Wavelet Transform

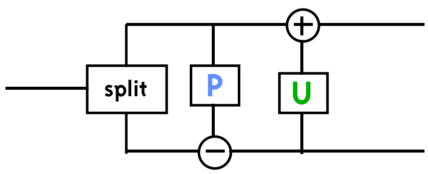


19

Linear Wavelet Transform

Order

- linear accuracy: 2nd order
- linear moments preserved: 2nd order
- (2,2) of Cohen-Daubechies-Feauveau

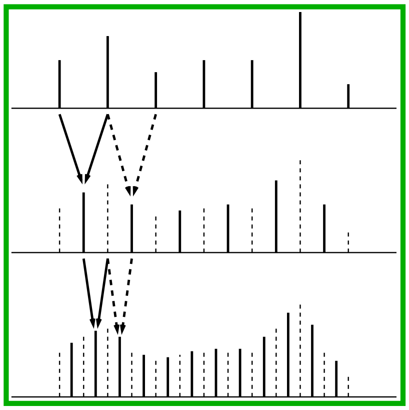


Extend

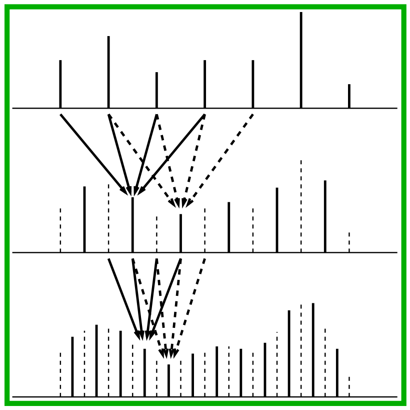
- build higher polynomial order predictors

20

Higher Order Prediction



linear



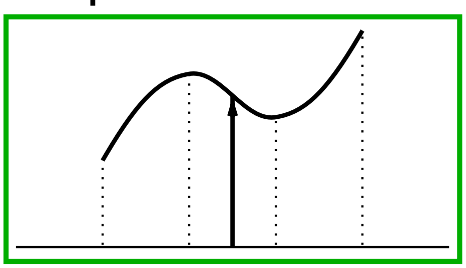
cubic

21

Higher Order Prediction

Use more (D) neighbors on left and right

- define interpolating polynomial of order N=2D
- sample at midpoint for prediction value
- example: D=2



effective weights:

-1/16 9/16 9/16 -1/16

22

Summary

Lifting Scheme

- construction of transforms
- spatial, Fourier

Haar example

- rewriting Haar in place

Two steps

- Predict
- Update

23

Summary

Predict

- detail coefficient is failure of prediction

Update

- smooth coefficient to preserve moments, e.g., average

Higher order extensions

- increase order of prediction and update

24

Building Blocks

Transform

- forward

$$W\{s_{n,k}\} = \{d_{j,l}\}$$

- inverse

$$\{s_{n,k}\} = W^{-1}\{d_{j,l}\}$$

- superposition

building blocks

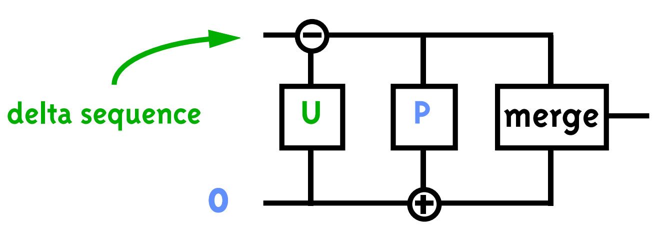
$$\{s_{n,k}\} = \sum d_{j,l} \, W^{-1}\{\delta_{j,l}\}$$

25

Scaling Functions

Cascade/Subdivision

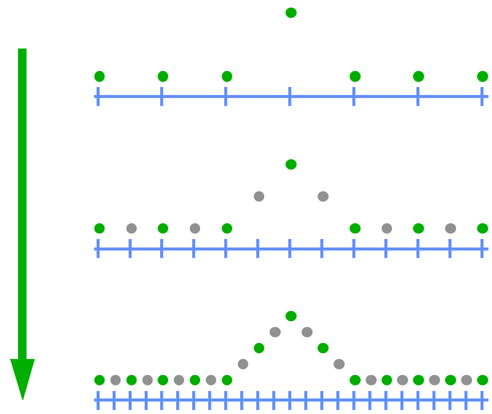
- single smooth coefficient



26

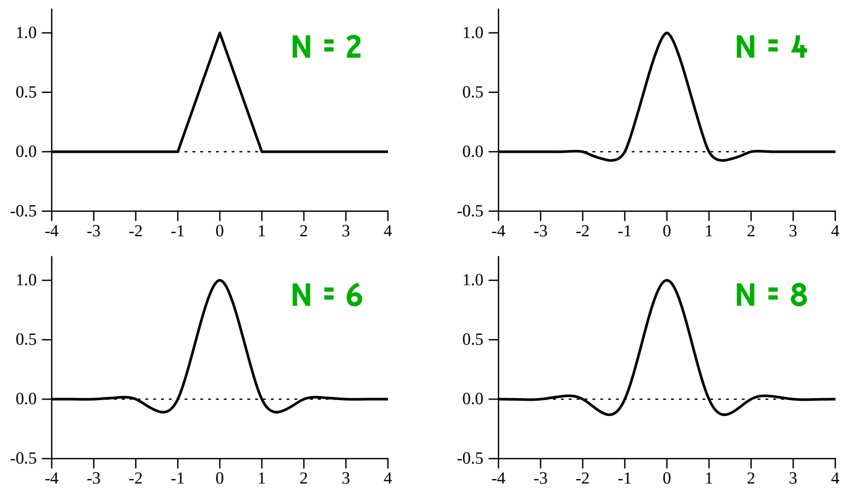
Scaling Functions

Cascade/Subdivision



27

Scaling Functions



28

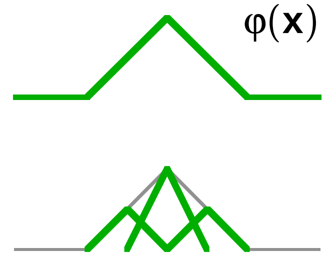
Twoscale Relation

Scaling function

0 0 1 0 0 →

{h_l} subdivide

0 0 0 ½ 1 ½ 0 0 0 →



$$\varphi(x) = \sum h_l \varphi(2x - l)$$

29

Duality

Function at 2 successive scales

$$\sum_k s_{j,k} \varphi_{j,k}(x) = f(x) = \sum_l s_{j+1,l} \varphi_{j+1,l}(x)$$

coarse fine

column vectors of coefficients

$$\begin{pmatrix} \vdots \\ s_{j+1,l} \\ \vdots \end{pmatrix} = H \begin{pmatrix} \vdots \\ s_{j,k} \\ \vdots \end{pmatrix} \qquad \begin{pmatrix} \cdots & \varphi_{j,k} & \cdots \end{pmatrix} = \begin{pmatrix} \cdots & \varphi_{j+1,l} & \cdots \end{pmatrix} H$$

row vectors of bases

30

Interpolating Scaling Functions

Properties for order N=2D

- compact support:

$$\varphi(\mathbf{x}) = \mathbf{0} \qquad \mathbf{x} \notin [-\mathbf{N}+\mathbf{1}, \mathbf{N}-\mathbf{1}]$$

- interpolation:

$$\varphi(\mathbf{k}) = \delta_{\mathbf{k}}$$

- polynomial reproduction:

$$\sum_{\mathbf{k}} \mathbf{k}^{\mathbf{p}} \varphi(\mathbf{x}-\mathbf{k}) = \mathbf{x}^{\mathbf{p}}$$

31

Interpolating Scaling Functions

Properties for order N=2D

- smoothness:

$$\varphi_{\mathbf{j,k}} \in \mathbf{C}^{\alpha(\mathbf{N})}$$

- twoscale relation:

$$\varphi(\mathbf{x}) = \sum_{\mathbf{l}=-\mathbf{N}}^{\mathbf{N}} \mathbf{h}_{\mathbf{l}} \varphi(\mathbf{2x}-\mathbf{l})$$

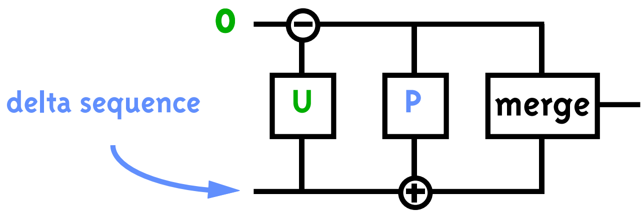
$$\mathbf{s}_{\mathbf{j+1,l}} = \sum_{\mathbf{k}} \mathbf{h}_{\mathbf{l-2k}} \mathbf{s}_{\mathbf{j,k}} \qquad \varphi_{\mathbf{j,k}}(\mathbf{x}) = \sum_{\mathbf{l}} \mathbf{h}_{\mathbf{l-2k}} \varphi_{\mathbf{j+1,l}}(\mathbf{x})$$

32

Wavelets

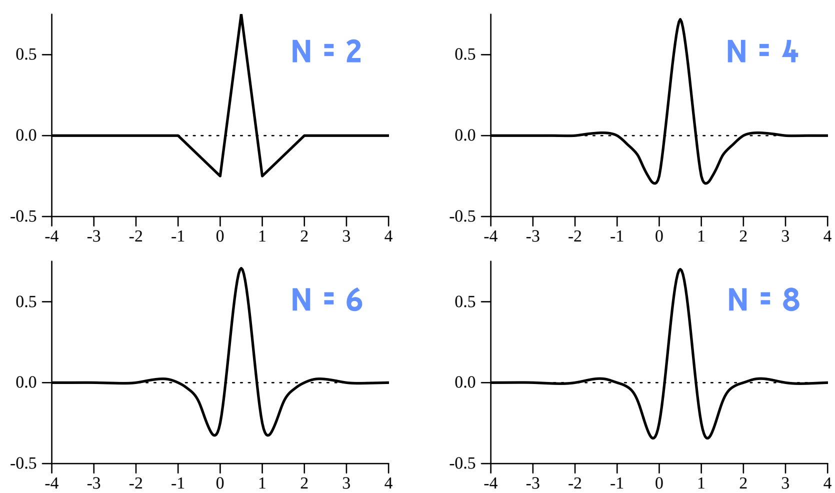
Cascade/Subdivision

- single detail coefficient



33

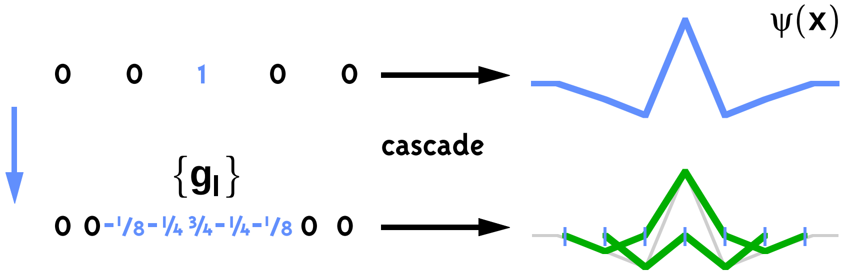
Wavelets



34

Twoscale Relation

Wavelet

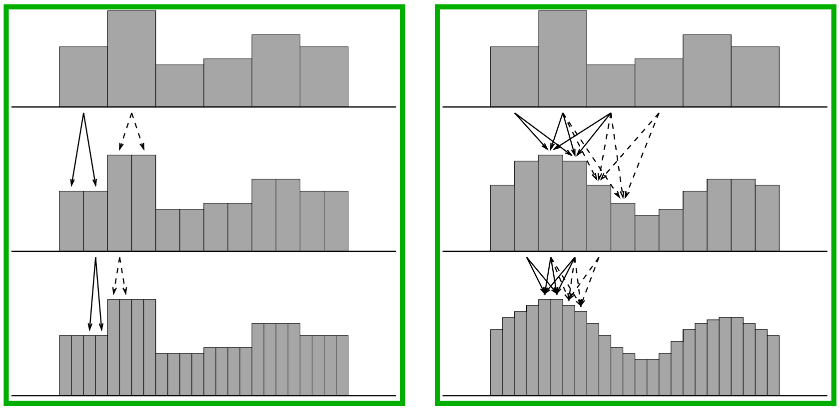


$$\psi(\mathbf{x}) = \sum \mathbf{g_l}\varphi(\mathbf{2x} - \mathbf{l})$$

35

36

Average Interpolation



constant

quadratic

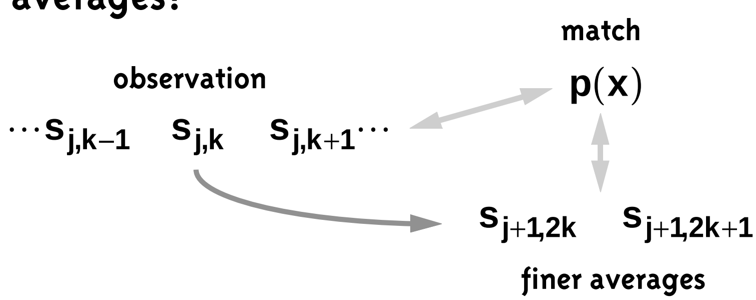
37

38

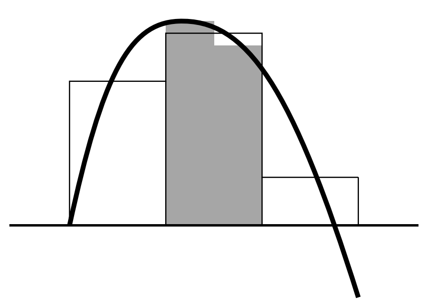
Average Interpolation

Idea

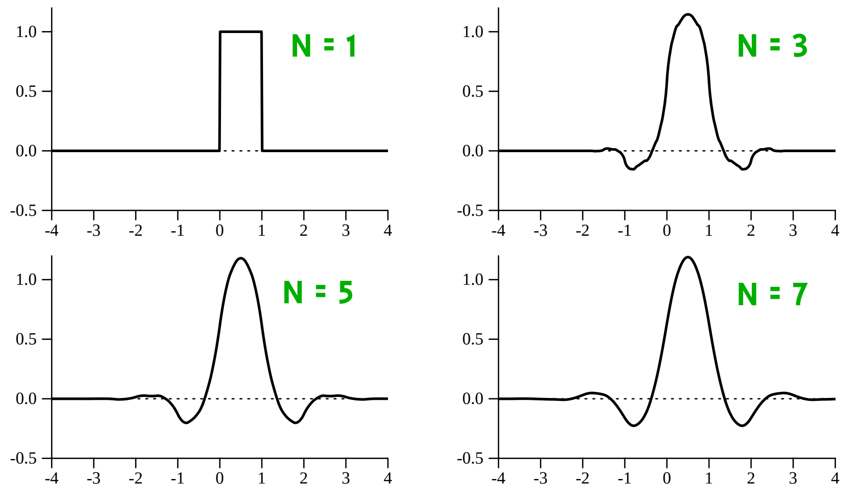
- assume observed samples are averages
- which polynomial would have produced those averages?



Average Interpolation



Scaling Functions



41

Average Interpolating Scaling Functions

Properties for order N=2D+1

- compact support:

$$\varphi(\mathbf{x}) = \mathbf{0} \qquad \mathbf{x} \notin [-\mathbf{N}+\mathbf{1},\mathbf{N}]$$

- average interpolation:

$$\int_{\mathbf{k}}^{\mathbf{k}+1} \varphi(\mathbf{x})\,\mathbf{dx} = \delta_{\mathbf{k}}$$

- polynomial reproduction:

$$\sum_{\mathbf{k}} \mathbf{Ave}\left(\mathbf{x}^{\mathbf{p}},\mathbf{k}\right)\varphi(\mathbf{x}-\mathbf{k}) = \mathbf{x}^{\mathbf{p}}$$

42

Average Interpolating Scaling Functions

Properties for order N=2D+1

- smoothness:

$$\varphi_{j,k} \in C^{\alpha(N)}$$

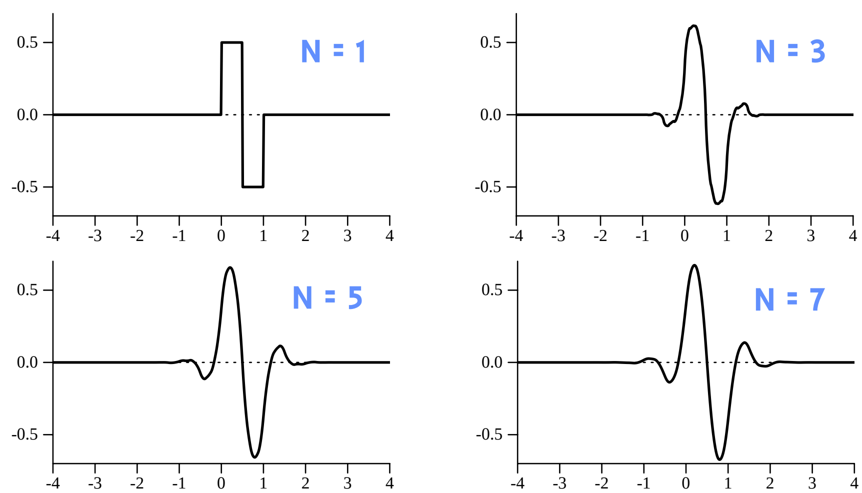
- twoscale relation:

$$\varphi(x) = \sum_{l=-N+1}^{N} h_l \varphi(2x - l)$$

$$\varphi_{j,k}(x) = \sum_{l} h_{l-2k} \varphi_{j+1,l}(x) \qquad s_{j+1,l} = \sum_{k} h_{l-2k} s_{j,k}$$

43

Wavelets



44

Differentiation

Interpolation and average interpolation

- given interpolation sequence compute exact derivative

$$\{s_{0,k}\} \qquad N = 2D$$

$$\{s'_{0,k} = s_{0,k+1} - s_{0,k}\} \qquad N' = 2D - 1$$

$$\frac{d}{dx}\varphi^{I}(x) = \varphi^{AI}(x+1) - \varphi^{AI}(x)$$

45

46

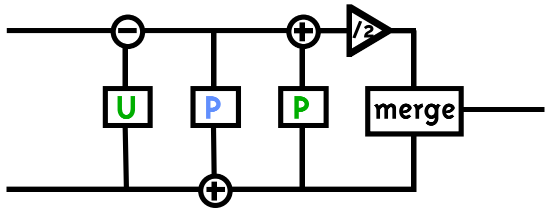
Cubic B-splines

Subdivision

- generate {1,4,6,4,1}

$$s_{j+1,2k+1} = \left(s_{j,k} + s_{j,k+1}\right)/2$$

$$s_{j+1,2k} = s_{j,k} + \left(s_{j+1,2k-1} + s_{j,2k+1}\right)/2$$

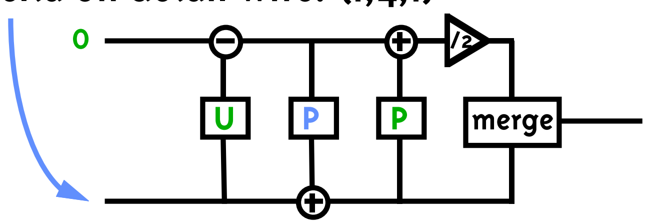


47

Cubic B-spline Wavelet

Completing the space

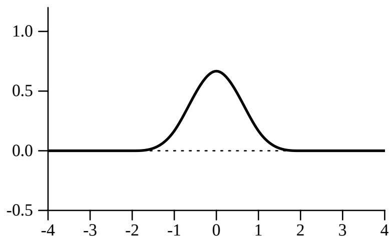
- put delta on detail wire: {1,4,1}



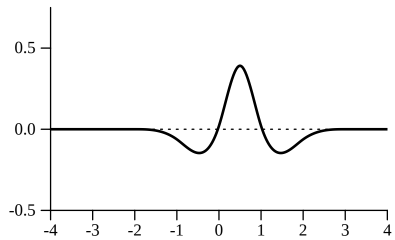
- get vanishing moment with update stage: {3/8,3/8}

48

Cubic B-spline



Scaling function



Wavelet