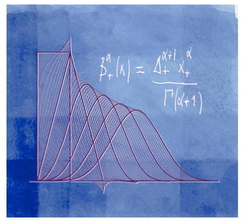
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Splines: on scale, differential operators and fast algorithms

Michael Unser

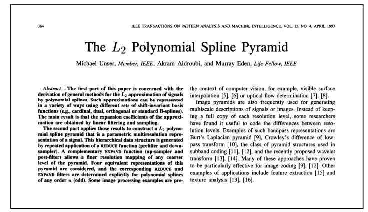
Biomedical Imaging Group
EPFL, Lausanne
Switzerland



Plenary talk, Scale Space 2005, Hofgeismar, April, 2005

Getting ideas across takes time ...

- 1990 submission: "The L_2 polynomial spline pyramid: a discrete representation of continuous signals in **scale space**"
- Journal: *IEEE Trans. Pattern Analysis and Mach. Intel.*


The L_2 Polynomial Spline Pyramid

Michael Unser, *Member, IEEE*, Akram Aldroubi, and Murray Eden, *Life Fellow, IEEE*

Give in when necessary .... but persevere

OUTLINE

- Introduction
- The basic atoms: B-splines
- Spline-based signal processing
  - Interpolation
  - Fast multi-scale algorithms
  - Applications
- Splines and wavelet theory

MOTIVATION

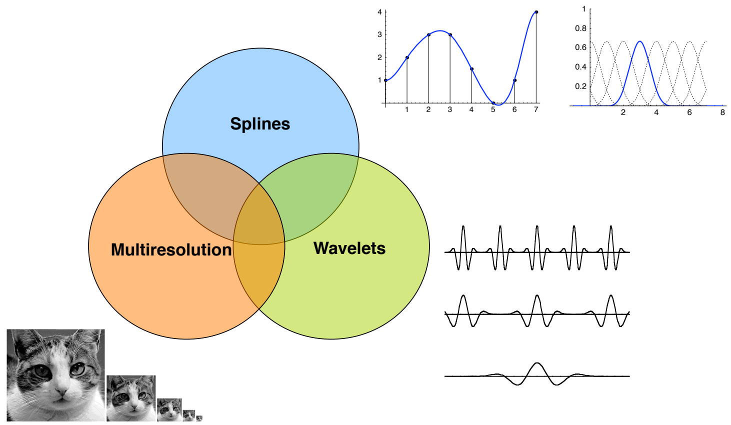
At the beginning there was a continuum.
Man made it discrete !

Continuous domain: $L_2(R^p)$ $f(x),\ x \in R^p$	⇔	Discrete domain: $\ell_2(Z^p)$ $f(k),\ k \in Z^p$
• real world objects • signals, images • sensor input		• measurements • algorithms • signal and image processing

- Sampling and signal acquisition
- Continuous/discrete algorithm design
  - Edge detection, PDEs, image registration, …
- Multi-scale approaches
  - Scale space
  - Image pyramids, wavelets
  - Coarse-to-fine and multigrid algorithms

Splines: a unifying framework

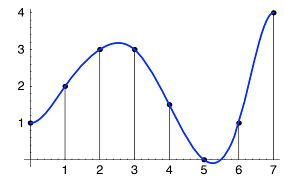
Linking the discrete and the continuous …..



Splines: definition

Definition: A function $s(x)$ is a polynomial spline of degree n with knots $\cdots < x_k < x_{k+1} < \cdots$ iff it satisfies the following two properties:

- Piecewise polynomial:
  $s(x)$ is a polynomial of degree n within each interval $[x_k, x_{k+1})$;
- Higher-order continuity:
  $s(x), s^{(1)}(x), \cdots, s^{(n-1)}(x)$ are continuous at the knots x_k.



- Effective degrees of freedom per segment:

  $n{+}1$ (polynomial coefficients) $-$ n (constraints) $=$ 1

- **Cardinal splines** = unit spacing and infinite number of knots

  ⇨ The right framework for signal processing

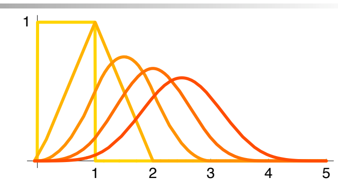
THE BASIC ATOMS: B-SPLINES

- Polynomial B-splines
- B-spline representation
- Differential properties
- Dilation properties
- Generalization: fractional B-splines
- Gaussian-like windows

Polynomial B-splines

- B-spline of degree n

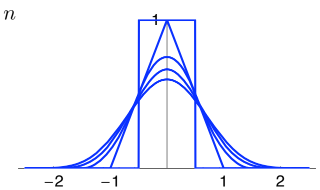
$$\beta_+^n(x) = \underbrace{\beta_+^0 * \beta_+^0 * \cdots * \beta_+^0(x)}_{(n+1)\text{ times}}$$



$$\beta_+^0(x) = \begin{cases} 1, & x \in [0,1) \\ 0, & \text{otherwise.} \end{cases}$$

- Key properties
  - Compact support: shortest polynomial spline of degree n
  - Positivity
  - Piecewise polynomial
  - Smoothness: Hölder continuous of order n
- Symmetric B-splines

$$\beta^n(x) = \beta_+^n\left(x + \tfrac{n+1}{2}\right)$$



B-spline representation

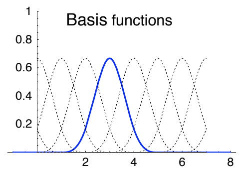
Theorem (Schoenberg, 1946)

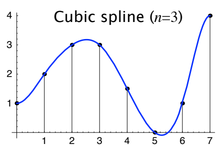
Every cardinal polynomial spline, $s(x)$, has a unique and stable representation in terms of its B-spline expansion

$$s(x) = \sum_{k\in\mathbb{Z}} c[k]\,\beta_+^n(x-k)$$

analog signal

discrete signal (B-spline coefficients)





In modern terminology: $\{\beta_+^n(x-k)\}_{k\in\mathbb{Z}}$ forms a Riesz basis.

Spline-related differential operators

- Continuous operators

Derivatives

$$\mathrm{D}\{\cdot\} = \frac{\mathrm{d}}{\mathrm{d}x} \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad j\omega$$

$$\mathrm{D}^m\{\cdot\} \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad (j\omega)^m$$

Integrators

$$\mathrm{D}^{-(n+1)}\{\delta(x)\} = \frac{x_+^n}{n!}$$

(impulse response of $(n+1)$-fold integrator)

- Discrete operators

Finite differences

$$\Delta_+\{\cdot\} \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad 1-e^{-j\omega}$$

$$\Delta_+^m\{\cdot\} \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad (1-e^{-j\omega})^m$$

One-sided power function:

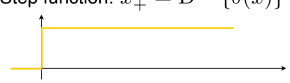
$$x_+^n = \begin{cases} x^n, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

B-splines: differential interpretation

- Construction of the B-spline of degree 0

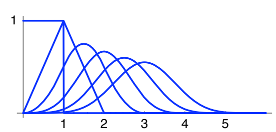
Step function: $x_+^0 = \mathrm{D}^{-1}\{\delta(x)\}$

$$\beta_+^0(x) = x_+^0 - (x-1)_+^0 = \Delta_+^1 x_+^0$$



- Generalization

$$\beta_+^n(x) = \Delta_+^{n+1}\mathrm{D}^{-(n+1)}\{\delta(x)\} = \frac{\Delta_+^{n+1}x_+^n}{n!}$$



- Fourier domain formula

$$\hat{\beta}_+^n(\omega) = \left(\frac{1-e^{-j\omega}}{j\omega}\right)^{n+1} = \frac{(1-e^{-j\omega})^{n+1}}{(j\omega)^{n+1}}$$

Discrete operator (finite difference)

Differentiation operator

B-splines: differential properties

- Fourier domain formula

$$\hat{\beta}_+^n(\omega) = \frac{(1-e^{-j\omega})^{n+1}}{(j\omega)^{n+1}}$$

Discrete derivatives

Exact derivatives

- Link between “discrete” and exact derivatives

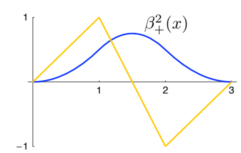
$$\forall f \in S', \quad \Delta_+^m f(x) = \beta_+^{m-1} * \mathrm{D}^m f(x)$$

- B-spline differentiation formula

$$\mathrm{D}^m\beta_+^n(x) = \Delta_+^m\beta_+^{n-m}(x)$$

finite difference operator

Spline degree reduction



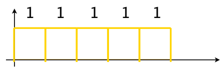
Sketch of proof: $(j\omega)^m\hat{\beta}_+^n(\omega) = (1-e^{-j\omega})^m \cdot \left(\frac{1-e^{-j\omega}}{j\omega}\right)^{n+1-m}$

B-splines: dilation properties

- Dilation by a factor m

$$\beta_+^n(x/m) = \sum_{k\in\mathbb{Z}} h_m^n[k]\beta^n(x-k) \quad \text{with} \quad H_m^n(z) = \frac{1}{m^n}\left(\sum_{k=0}^{m-1} z^{-k}\right)^{n+1}$$

- Piecewise constant case $(n = 0)$



$$H_m^0(z) = 1 + z^{-1} + \cdots z^{-(m-1)}$$ (Moving sum filter)

- Applications: fast spline-based algorithms
  - Zooming
  - Smoothing
  - Multi-scale processing
  - Wavelet transform

Dyadic case: wavelets

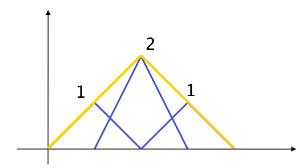
- Dilation by a factor of 2

$$\beta_+^n(x/2) = \sum_{k\in\mathbb{Z}} h_2^n[k]\beta_+^n(x-k)$$

- Binomial filter

$$H_2^n(z) = 2\left(\frac{1+z^{-1}}{2}\right)^{n+1} = \frac{1}{2^n}\sum_{k=0}^{n+1}\binom{n+1}{k} z^{-k}$$

- Example: piecewise linear splines



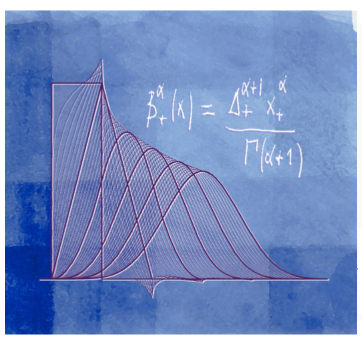
Generalization: fractional B-splines

$$\beta_+^0(x) = \Delta_+ x_+^0 \quad \overset{\mathcal{F}}{\longleftrightarrow} \quad \frac{1-e^{-j\omega}}{j\omega}$$

$$\vdots \qquad\qquad \vdots$$

$$\beta_+^\alpha(x) = \frac{\Delta_+^{\alpha+1} x_+^\alpha}{\Gamma(\alpha+1)} \quad \overset{\mathcal{F}}{\longleftrightarrow} \quad \left(\frac{1-e^{-j\omega}}{j\omega}\right)^{\alpha+1}$$

One-sided power function: $x_+^\alpha = \begin{cases} x^\alpha, & x \geq 0 \\ 0, & x < 0 \end{cases}$



- Properties
  - $\{\beta_+^\alpha(x-k)\}_{k\in\mathbb{Z}}$ is a valid Riesz basis for $\alpha < -\frac{1}{2}$
  - Convolution property: $\beta_+^{\alpha_1} * \beta_+^{\alpha_2} = \beta_+^{\alpha_1+\alpha_2+1}$

(Unser & Blu, *SIAM Rev*, 2000)

Gaussian-like windows

Theorem: The (fractional) B-splines converge (in L_p-norm) to a Gaussian as the degree goes to infinity:

$$\lim_{\alpha\to\infty}\{\beta_+^\alpha(x)\} = \frac{1}{\sqrt{2\pi}\cdot\sigma_\alpha}\exp\left(\frac{-(x-x_\alpha)^2}{2\sigma_\alpha^2}\right)$$

with $\sigma_\alpha = \sqrt{\frac{\alpha+1}{12}}$

- Polynomial B-splines: $\alpha = n$ (integer)
  - Compact support: $[0, n+1]$
  - Fast convolution algorithms: recursive or multi-scale

SPLINE-BASED SIGNAL PROCESSING

- Spline fitting: overview
- B-spline interpolation
- Fast multi-scale algorithms
- Applications

Spline fitting: overview

- B-spline representation: $s(x) = \sum_{k\in\mathbb{Z}} c[k]\beta^n(x-k)$

  Goal: Determine $c[k]$ such that $s(x)$ is a "good" representation of our signal

- Interpolation (exact, reversible)

  Discrete input $f[k]$ → Interpolation algorithm → $c[k]$ such that $s(x)|_{x=k} = f[k]$

- Spline approximation (at scale a)

  Analog input $f(x)$ → Sampling algorithm → $c[k] = \langle f, \tilde{\varphi}(\cdot/a - k)\rangle$

  such that $\min_{s\in V_a} \|f-s\|_{L_2}^2$

Spline fitting (Cont'd)

- B-spline representation: $s(x) = \sum_{k\in\mathbb{Z}} c[k]\beta^n(x-k)$
- Smoothing splines

  Discrete, noisy input: $f[k] = s(k) + n[k]$ → Smoothing algorithm → $c[k]$

Theorem: The solution (among all functions) of the smoothing spline problem

$$\min_{s\in W_2^m}\left\{\sum_{k\in\mathbb{Z}} |f[k]-s(k)|^2 + \lambda\int_{-\infty}^{+\infty} |\mathrm{D}^m s(x)|^2 dx\right\}$$

is a cardinal spline of degree $2m-1$. Its coefficients $c[k] = h_\lambda * f[k]$ can be obtained by suitable digital filtering of the input samples $f[k]$.

- Special case: the draftman's spline

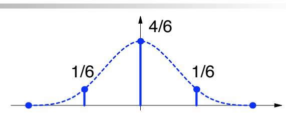
  The minimum curvature interpolant is obtained by setting $m=2$ and $\lambda \to 0$.

  It is a cubic spline !

B-spline interpolation

- Discrete B-spline kernels

  $b_1^n[k] = \beta^n(x)|_{x=k} \quad \overset{z}{\longleftrightarrow} \quad B_1^n(z) = \sum_{k=-\lfloor n/2\rfloor}^{\lfloor n/2\rfloor} \beta^n(k) z^{-k}$



- B-spline interpolation: inverse filter solution

  $f[k] = \sum_{k\in\mathbb{Z}} c[l]\ \beta^n(x-l)|_{x=k} = (b_1^n * c)[k] \quad \Rightarrow \quad c[k] = (b_1^n)^{-1} * f[k]$

- Efficient recursive implementation

  $(b_1^3)^{-1}[k] \quad \overset{z}{\longleftrightarrow} \quad \frac{6}{z+4+z^{-1}} = \frac{(1-\alpha)^2}{(1-\alpha z)(1-\alpha z^{-1})}$ (symmetric exponential)

  ⇒ Cascade of first order recursive filters

  → $\frac{1}{1-\alpha z^{-1}}$ (causal) → $\frac{1}{1-\alpha z}$ (anti-causal) →

Generic C-code (splines of any degree n)

- Main recursion

```
void   ConvertToInterpolationCoefficients (
           double c[ ],   long DataLength,    double  z[ ],   long NbPoles,   double  Tolerance)
 {double Lambda = 1.0; long n, k;
    if (DataLength == 1L) return;
    for (k = 0L; k < NbPoles; k++) Lambda = Lambda * (1.0 - z[k]) * (1.0 - 1.0 / z[k]);
    for (n = 0L; n < DataLength; n++) c[n] *= Lambda;
    for (k = 0L; k < NbPoles; k++) {
        c[0] = InitialCausalCoefficient(c, DataLength, z[k], Tolerance);
        for (n = 1L; n < DataLength; n++) c[n] += z[k] * c[n - 1L];
        c[DataLength - 1L] = (z[k] / (z[k] * z[k] - 1.0))
                    * (z[k] * c[DataLength - 2L] + c[DataLength - 1L]);
        for (n = DataLength - 2L; 0 <= n; n--) c[n] = z[k] * (c[n + 1L]- c[n]); }
}
```

- Initialization

```
double  InitialCausalCoefficient (
           double  c[ ],  long    DataLength,  double  z,  double  Tolerance)
{  double Sum, zn, z2n, iz; long n, Horizon;
   Horizon = (long)ceil(log(Tolerance) / log(fabs(z)));
   if (DataLength < Horizon) Horizon = DataLength;
   zn = z; Sum = c[0];
   for (n = 1L; n < Horizon; n++) {Sum += zn * c[n]; zn *= z;}
   return(Sum);
}
```


Spline interpolation

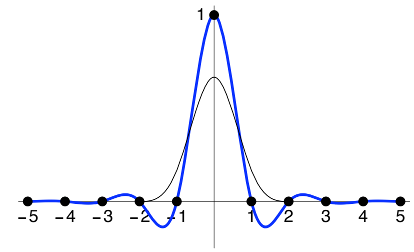
- Equivalent forms of spline representation

$$s(x) = \sum_{k\in\mathbb{Z}} c[k]\beta^n(x-k) \quad = \quad \sum_{k\in\mathbb{Z}} \left(s(k) * (b_1^n)^{-1}[k]\right)\beta^n(x-k)$$

$$= \sum_{k\in\mathbb{Z}} s(k)\varphi^n_{\text{int}}(x-k)$$

- Cardinal (or fundamental) spline

$$\varphi^n_{\text{int}}(x) = \sum_{k\in\mathbb{Z}} (b_1^n)^{-1}[k]\ \beta^n(x-k)$$



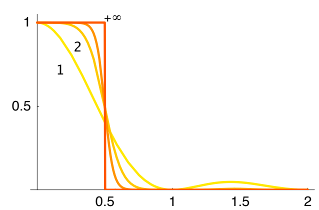
⇨ Finite cost implementation of an infinite impulse response interpolator !

Limiting behavior

- Spline interpolator

Impulse response | Frequency response

$$\varphi^n_{\text{int}}(x) \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad H^n(\omega) = \left(\frac{\sin(\omega/2)}{\omega/2}\right)^{n+1} \frac{1}{B_1^n(e^{j\omega})}$$



- Asymptotic property

The cardinal spline interpolators converge to the sinc-interpolator (ideal filter) as the degree goes to infinity:

$$\lim_{n\to\infty} \varphi^n_{\text{int}}(x) = \text{sinc}(x), \qquad \lim_{n\to\infty} H^n(\omega) = \text{rect}\left(\frac{\omega}{2\pi}\right)$$ (in all L_p-norms)

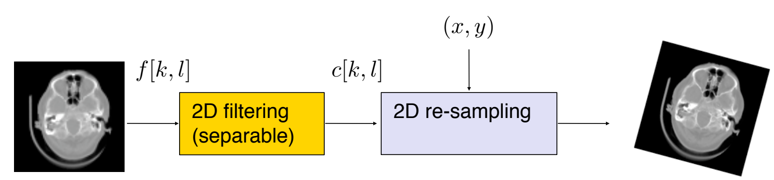
(Aldroubi et al., *Sig. Proc.*, 1992)

➡ Includes Shannon's theory as a particular case !

Geometric transformation of images

- 2D separable model

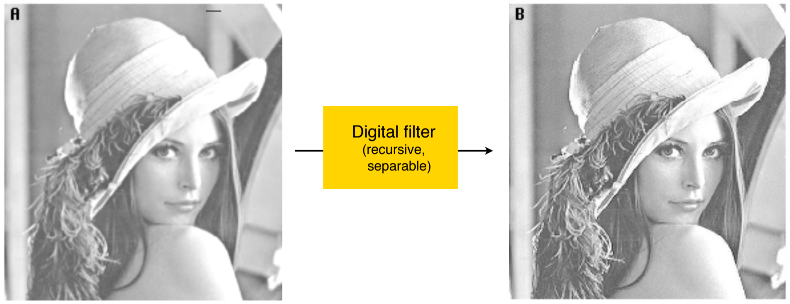
$$f(x,y) = \sum_{k=k_1}^{k_1+n+1} \sum_{l=l_1}^{l_1+n+1} c[k,l]\ \beta^n(x-l)\ \beta^n(y-l)$$



- Applications

zooming, rotation, re-sizing, re-formatting, warping

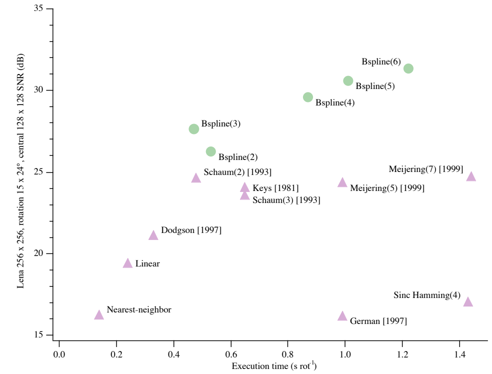
Cubic spline coefficients in 2D



Pixel values $f[k,l]$ B-spline coefficients $c[k,l]$

High-quality image interpolation

- Splines: best cost-performance tradeoff



Demo

Thévenaz et al., *Handbook of Medical Image Processing*, 2000

Interpolation benchmark

Cumulative rotation experiment: the best algorithm wins !

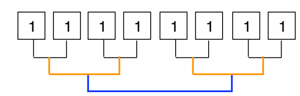
Bilinear Windowed-sinc Cubic spline

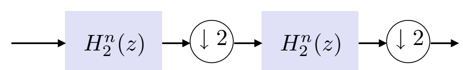
Fast multi-scale filtering

Three alternative methods for the fast evaluation of $f(x) * \beta^n(x/a)$

1) Pyramid or tree algorithms

$a = 2^i$



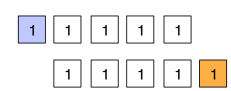


Binomial filter

(Unser et al., *IEEE Trans. PAMI*, 1993)

2) Recursive filtering (iterated moving average)

$a = m \in \mathbb{Z}^+$



$s_m[k] = s_m[k-1] + f[k] - f[k-m]$

(Unser et al., *IEEE Trans. Sig. Proc*, 1994)

Fast multi-scale filtering (Cont'd)

Challenge: $O(N)$ evaluation of $f(x) * \beta^n(x/a)$

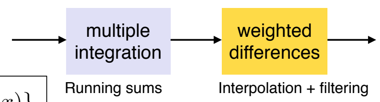
3) Differential approach $a \in \mathbb{R}^+$

$$f(x) * \beta_+^0(x/a) = F(x) - F(x-a) = \Delta_a^1 \mathrm{D}^{-1}\{f(x)\}$$

Integral (or primitive): $F(x) = \int_{-\infty}^{x} f(t)dt = \mathrm{D}^{-1}\{f(x)\}$

Finite difference with step a: $\Delta_a\{f(x)\} = f(x) - f(x-a)$

- Generalization



$$f(x) * \beta_+^n(x/a) = \frac{1}{a^n} \Delta_a^{n+1} \mathrm{D}^{-(n+1)}\{f(x)\}$$

(Munoz et al., *IEEE Trans. Imag. Proc*, 2001)

Principle: The integral of a spline of degree n is a spline of degree $n+1$.

Splines: more applications

- Sampling and interpolation
  - Interpolation, re-sampling, grid conversion
  - Image reconstruction
  - Geometric correction
- Feature extraction
  - Contours, ridges
  - Differential geometry
  - Image pyramids
  - Shape and active contour models
- Image matching
  - Stereo
  - Image registration (multi-modal, rigid body or elastic)
- Motion analysis
  - Optical flow

SPLINES AND WAVELET THEORY

- Scaling functions
- Order of approximation
- B-spline factorization theorem
- Splines: the key to wavelet theory
- Fractional B-spline wavelets

Scaling function

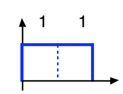
Definition: $\varphi(x)$ is an admissible scaling function of L_2 iff:

- Riesz basis condition

$$\forall c \in \ell_2, \quad A \cdot \|c\|_{\ell_2} \leq \left\| \sum_{k \in \mathbb{Z}} c[k]\varphi(x-k) \right\|_{L_2} \leq B \cdot \|c\|_{\ell_2}$$

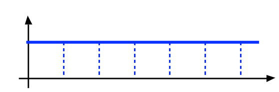
- Two-scale relation

$$\varphi(x/2) = \sum_{k \in \mathbb{Z}} h[k]\varphi(x-k)$$



- Partition of unity

$$\sum_{k \in \mathbb{Z}} \varphi(x-k) = 1$$



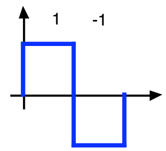
From scaling functions to wavelets

- Wavelet bases of L_2 [Mallat-Meyer, 1989]

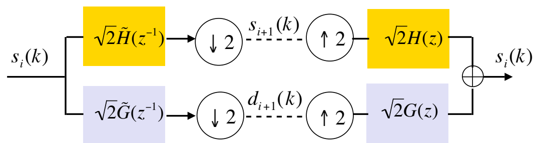
Theorem: For any given admissible scaling function of L_2, $\varphi(x)$, there exists a wavelet $\psi(x/2) = \sum_{k\in\mathbb{Z}} g[k]\varphi(x-k)$ such that the family of functions

$$\left\{2^{-i/2}\psi\left(\frac{x-2^ik}{2^i}\right)\right\}_{i\in\mathbb{Z},k\in\mathbb{Z}}$$

forms a Riesz basis of L_2.



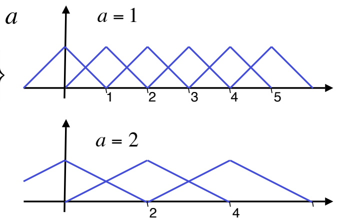
- Constructive approach: perfect reconstruction filterbank



Order of approximation

- General "shift-invariant" space at scale a

$$V_a(\varphi) = \left\{s_a(x) = \sum_{k\in\mathbb{Z}} c[k]\varphi\left(\frac{x}{a}-k\right) : c \in \ell_2\right\}$$



- Projection operator

$$\forall f \in L_2, \quad P_a f = \arg\min_{s_a \in V_a} \|f - s_a\|_{L_2} \quad \in V_2$$

- Order of approximation

Definition

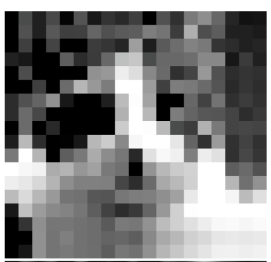
A scaling/generating function φ has order of approximation γ iff

$$\forall f \in W_2^\gamma, \quad \|f - P_a f\|_{L_2} \le C \cdot a^\gamma \cdot \|f^{(\gamma)}\|_{L_2}$$

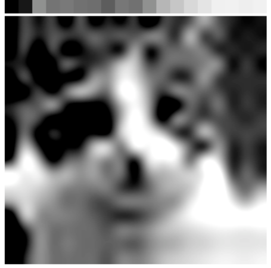
B-splines of degree α have order of approximation $\gamma=\alpha+1$

Spline reconstruction of a CAT-scan

Piecewise constant
$\gamma = 1$



Cubic spline
$\gamma = 4$



B-spline factorization

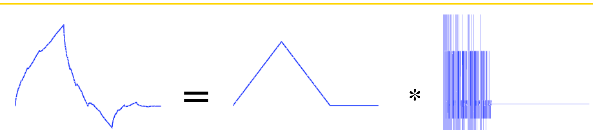
- Factorization theorem

A valid scaling function $\varphi(x)$ has order of approximation γ iff

$$\varphi(x) = \left(\beta_+^\alpha * \varphi_0\right)(x)$$

where β_+^α with $\alpha = \gamma - 1$: regular, B-spline part

$\varphi_0 \in S'$: irregular. distributional part



- Refinement filter: general case

$$H(z) = \underbrace{\left(\frac{1+z^{-1}}{2}\right)^\gamma}_{\text{spline part}} \cdot \underbrace{Q(z)}_{\text{distributional part}}$$

Splines: the key to wavelet theory

Sobolev smoothness
$\partial^s \varphi \in L_2$

$\Downarrow$ $\gamma \geq s$

B-spline factorization:
$\varphi = \beta_+^{\gamma-1} * \varphi_0$

$\Leftrightarrow$

Approximation order:
$\|f - P_a f\|_{L_2} = O(a^\gamma)$

$\Leftrightarrow$

Multi-scale differentiator
$\hat{\tilde{\psi}}(\omega) \propto (-j\omega)^\gamma, \ \omega \to 0$

$\Downarrow$ $\Uparrow$ general case: $n < \gamma \leq n+1$
compact support: $\gamma = n+1$ (Strang-Fix, 1971)

Polynomial reproduction
degree: $n = \lceil \gamma - 1 \rceil$

$\Updownarrow$

Vanishing moments:
$\int x^p \tilde{\psi}(x) dx = 0, \ \ p = 0, \ldots, n$

(Unser and Blu, IEEE-SP, 2003)

CONCLUSION

- Distinctive features of splines
  - Simple to manipulate
  - Smooth and well-behaved
  - Excellent approximation properties
  - Multiresolution properties
  - Fundamental nature (Green functions of derivative operator)
- Splines and image processing
  - A story of avoidance and, more recently, love….
  - Best cost/performance tradeoff
  - Many applications …..
- Unifying signal processing formulation
  - Tools: digital filters, convolution operators
  - Efficient recursive filtering solutions
  - Flexibility: piecewise constant to bandlimited

Scale space vs. splines

- Linear scale space (redundant)
  - Smoothing splines
  - Multi-resolution analysis (non-redundant)
- Finite difference methods
  - Hilbert space methods
- Non-linear diffusion
  - Non-linear smoothing splines
  - Wavelet denoising

No need to be dogmatic: you can also use splines to improve "scale-space" algorithms

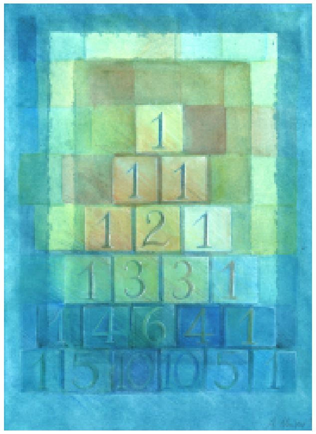
- Hilbert space framework: Think analog, act discrete !
- "Optimal" discretization of differential operators
- Fast multi-scale, multi-grid algorithms
- …

Acknowledgments

Many thanks to

- Dr. Thierry Blu
- Prof. Akram Aldroubi
- Prof. Murray Eden
- Dr. Philippe Thévenaz
- Annette Unser, Artist

\+ many other researchers, and graduate students



The end: Thank you!

- Spline tutorial
  - M. Unser, "Splines: A Perfect Fit for Signal and Image Processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22-38, 1999.
- Spline and wavelets
  - M. Unser, T. Blu, "Wavelet Theory Demystified," *IEEE Trans. on Signal Processing*, vol. 51, no. 2, pp. 470-483, 2003.
- Smoothing splines and stochastic formulation
  - M. Unser, T. Blu, "Generalized Smoothing Splines and the Optimal Discretization of the Wiener Filter," *IEEE Trans. Signal Processing*, in press.
- Preprints and demos: **http://bigwww.epfl.ch/**