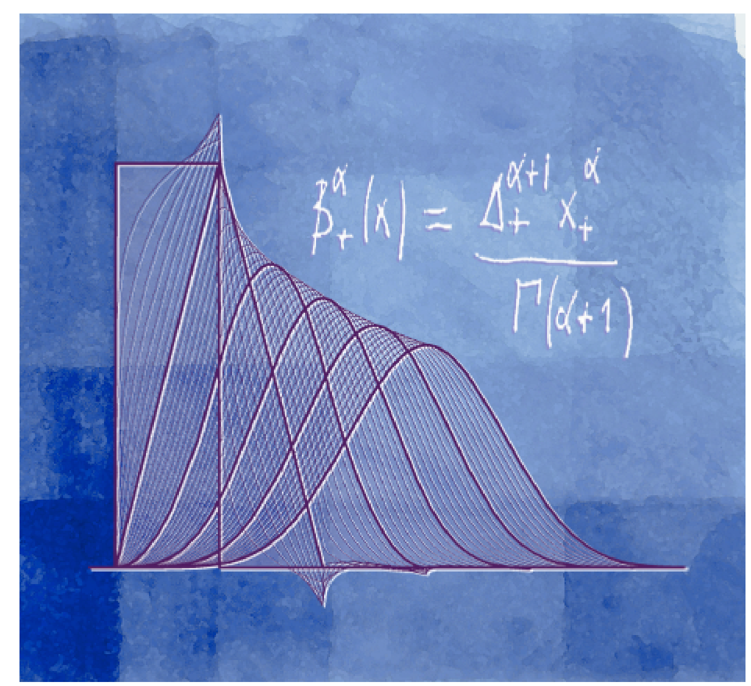
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Splines: A unifying framework for image processing

Michael Unser

Biomedical Imaging Group
EPFL, Lausanne
Switzerland

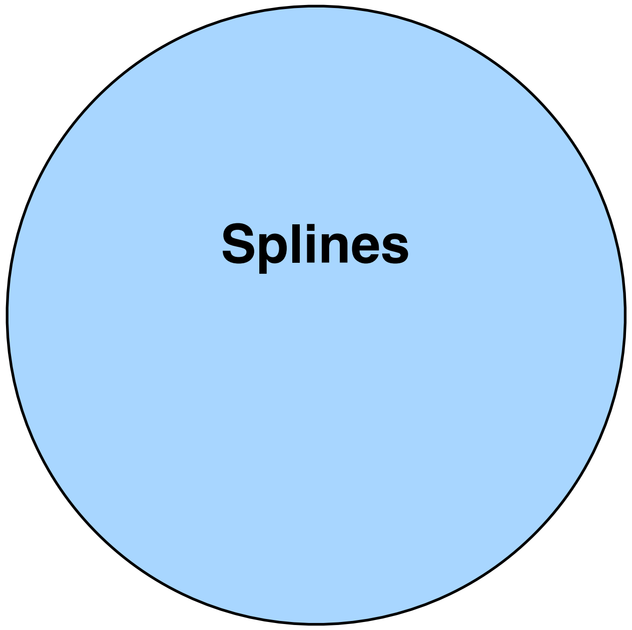


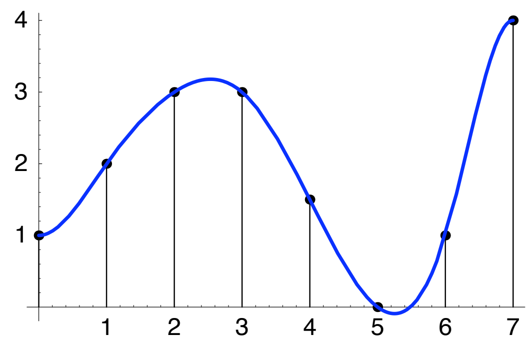
Plenary talk, ICIP 2005, Genoa, September 2005

Splines: A unifying framework

Linking the ***continuous*** *and the* ***discrete*** *…..*

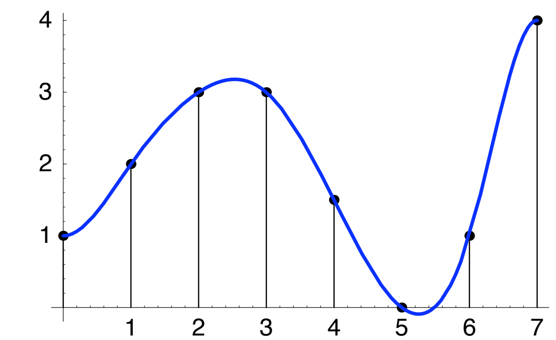
- Sampling and acquisition



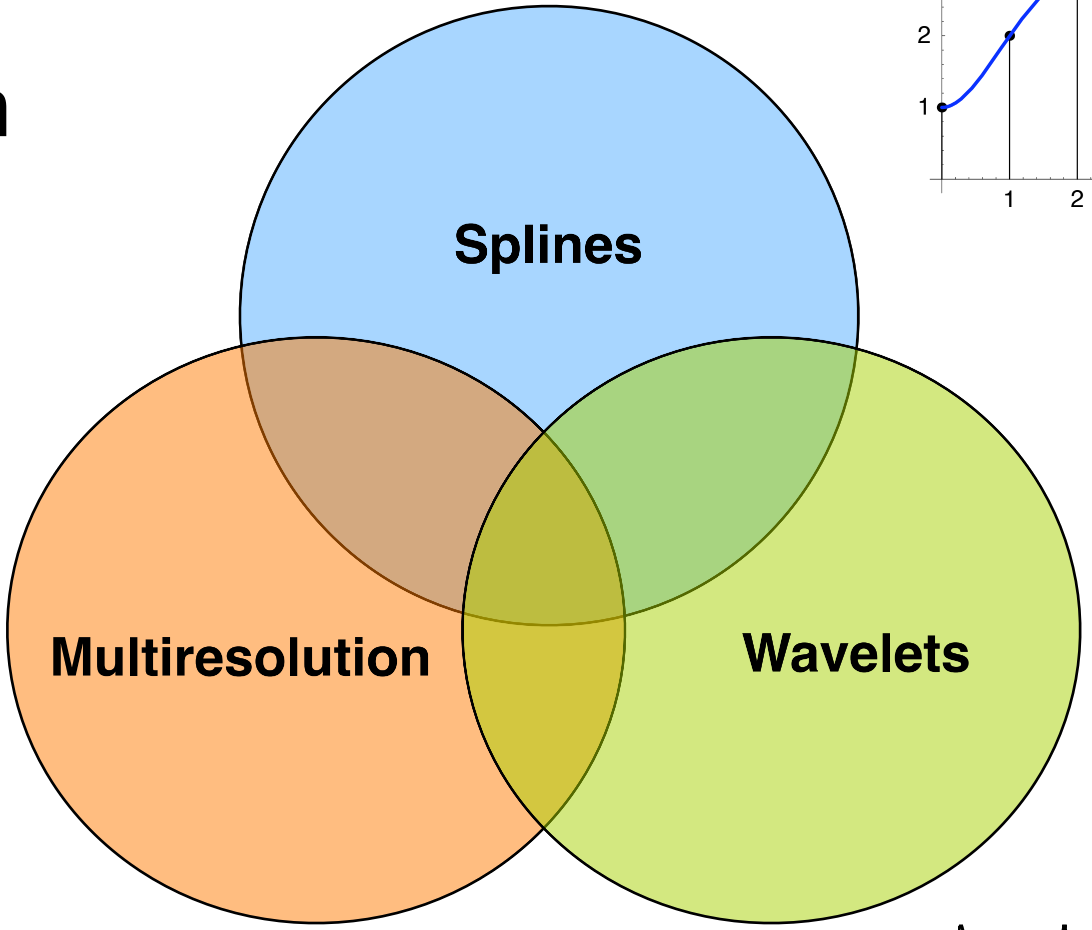


Splines: A unifying framework

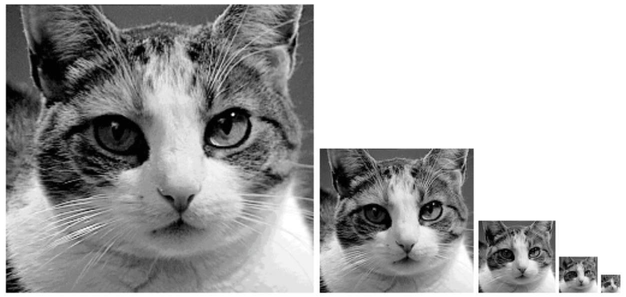
Linking the ***discrete*** *and the* ***continuous*** *…..*

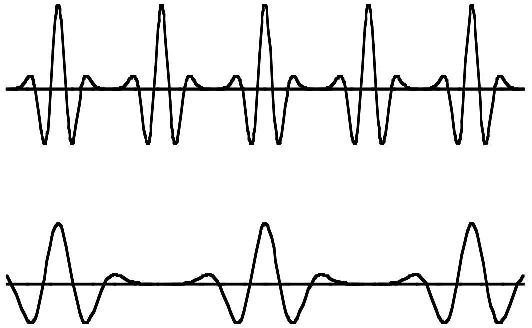


- Sampling and acquisition



- Algorithm design
  "Think analog, act digital"
  - Geometric processing
  - Feature extraction
  - PDEs ....
- Multi-scale approaches



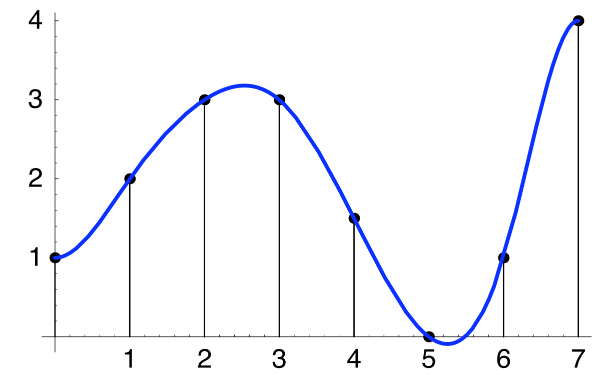


OUTLINE

- The basic atoms: B-splines
- Spline-based image processing
  - Interpolation vs. approximation
  - Fast algorithms
  - Applications
- Further perspectives
  - Splines and wavelet theory
  - Splines and fractals

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:

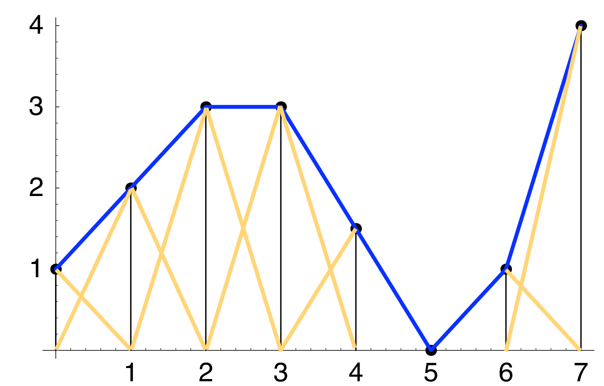


- Piecewise polynomial:

  $s(x)$ 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:

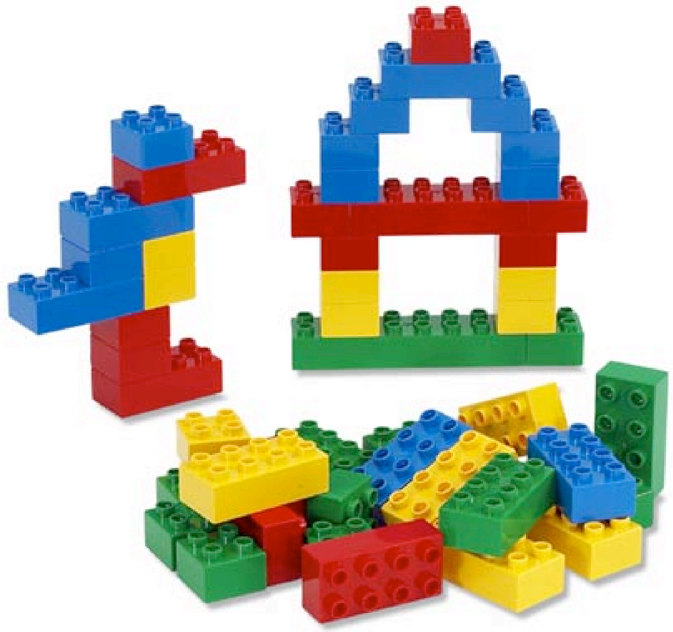
$n+1$	$-$	n	$=$	1
(polynomial coefficients)		(constraints)		

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

  ⇨ The right framework for signal processing

THE BASIC ATOMS: B-SPLINES

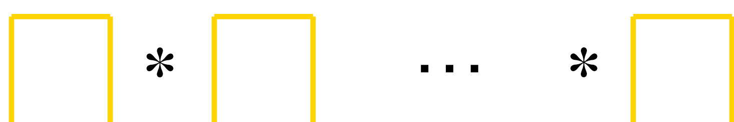
- Polynomial B-splines
- B-spline representation
- Differential properties
- Dilation properties
- Multidimensional B-splines

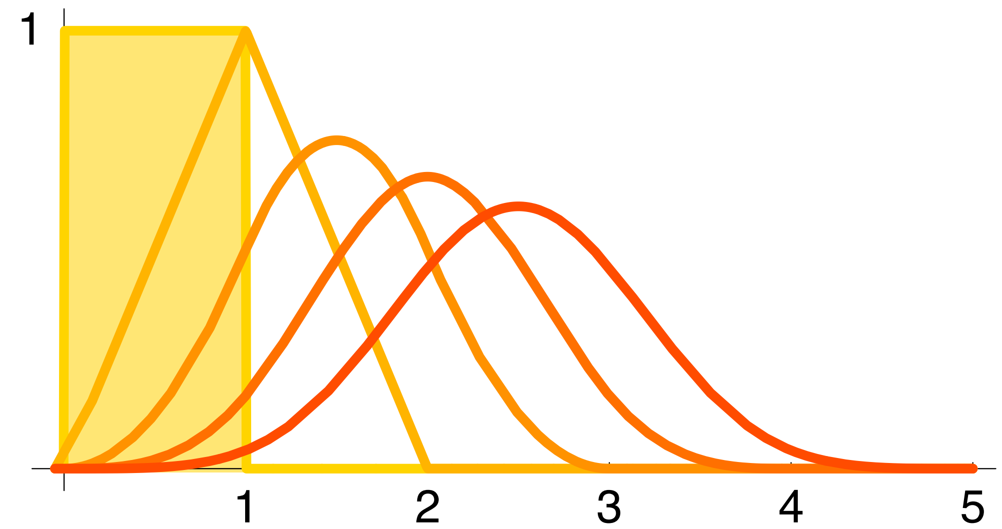


Polynomial B-splines

- B-spline of degree n

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



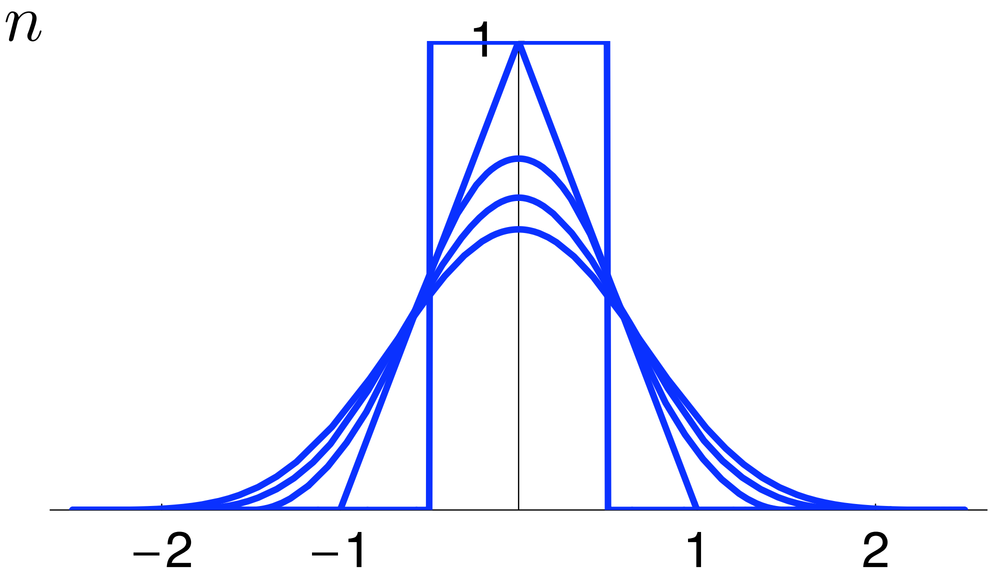


$$\beta_+^0(x) = \begin{cases} 1, & x \in [0,1) \\ 0, & \text{otherwise.} \end{cases}$$

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

- Symmetric B-splines

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



B-spline representation

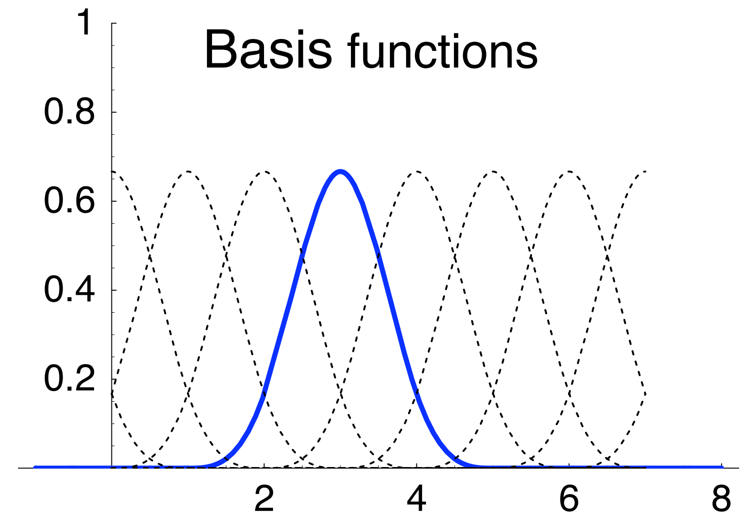
Theorem (Schoenberg, 1946)

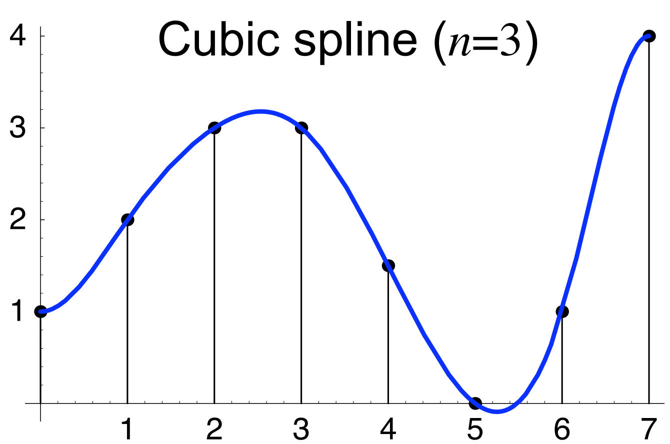
Every cardinal polynomial spline, $s(x)$, has a unique and stable representation in terms of its B-spline expansion

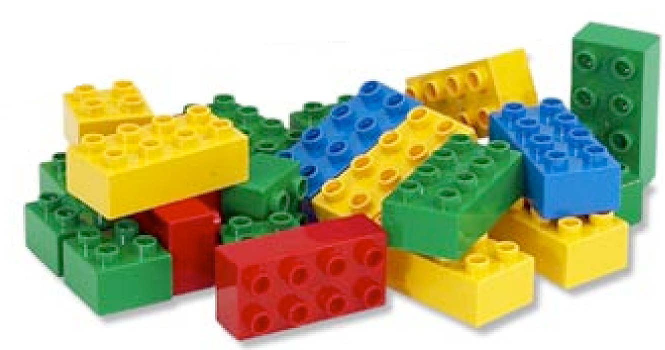
$$s(x) = \sum_{k \in \mathbb{Z}} c[k]\, \beta_+^n(x - k)$$

analog signal

discrete signal
(B-spline coefficients)







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

The lego revisited

- Continous operator

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

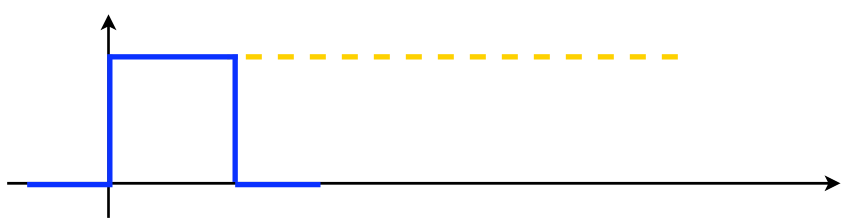
- Discrete operator

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

- Construction of the B-spline of degree 0

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

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



- Fourier domain formula

$$\hat{\beta}_+^0(\omega) = \frac{1 - e^{-j\omega}}{j\omega}$$

Discrete operator (finite difference)

Continuous operator (derivative)

B-splines: differential interpretation

- Continuous operators

  Derivatives

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

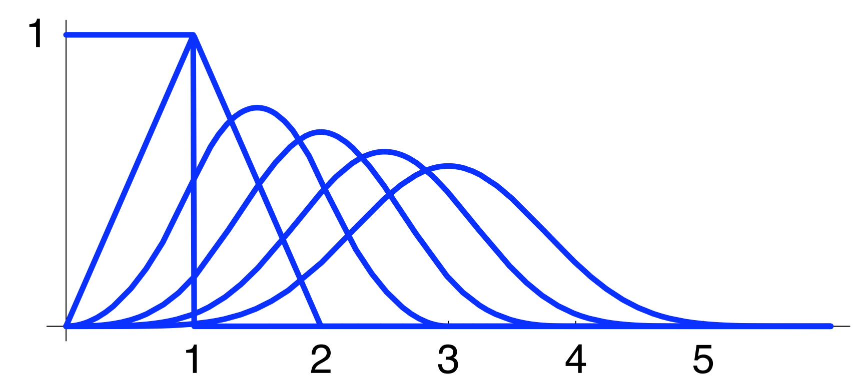
- Discrete operators

  Finite differences

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

- B-spline construction

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



- Fourier domain formula

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

One-sided power function:

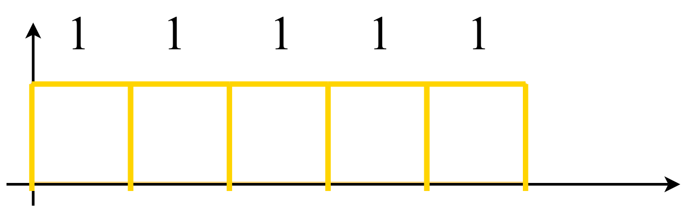
$$x_+^n = \begin{cases} x^n, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

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

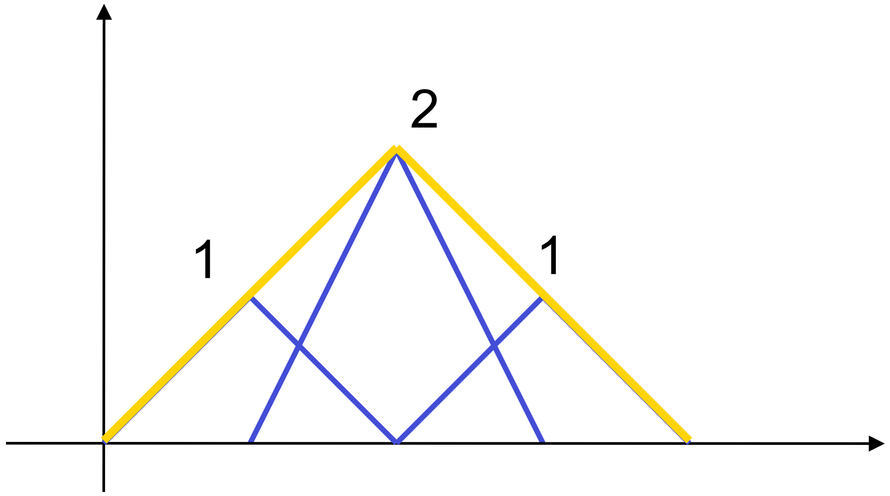
- Dilation by a factor of 2

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

- Binomial filter

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

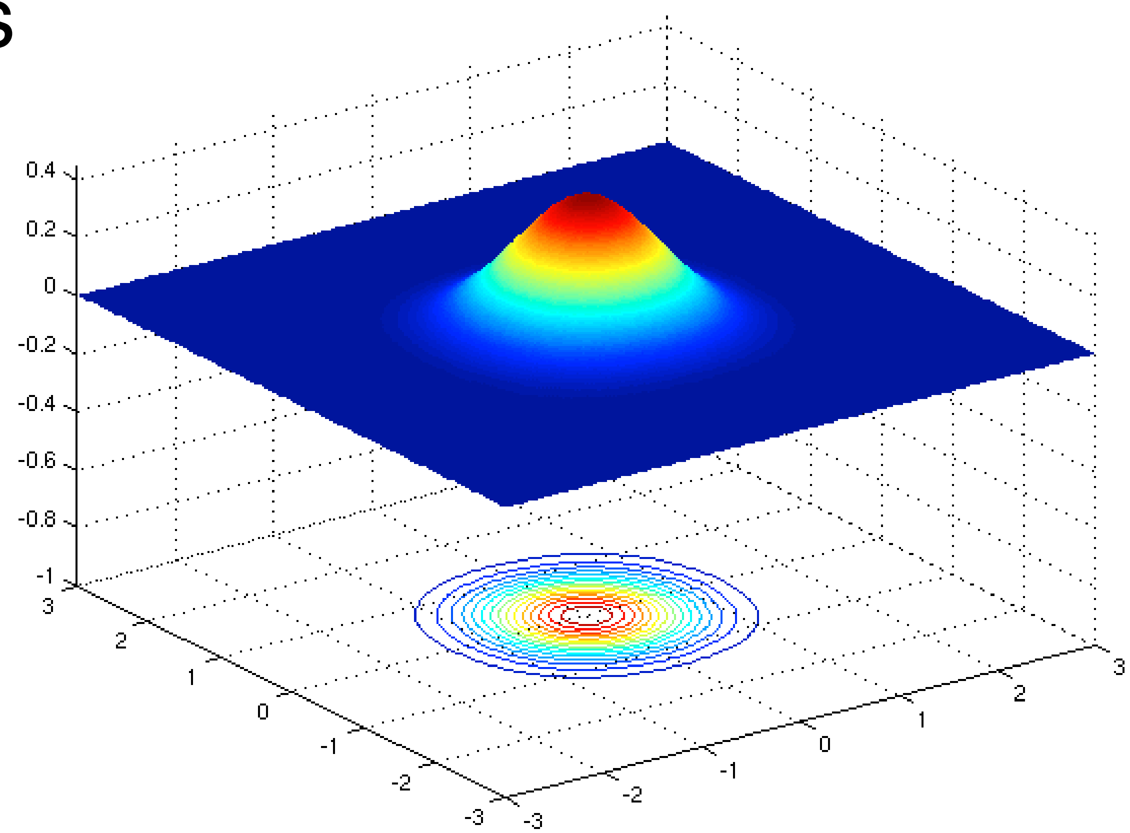
- Example: piecewise linear splines



B-spline representation of images

- Symmetric, tensor-product B-splines

$$\beta^n(x_1, \cdots, x_d) = \beta^n(x_1) \times \cdots \times \beta^n(x_d)$$



- Multidimensional spline function

$$s(x_1, \cdots, x_d) = \sum_{(k_1, \cdots k_d) \in \mathbb{Z}^d} c[k_1, \cdots, k_d] \, \beta^n(x_1 - k_1, \cdots, x_d - k_d)$$

continuous-space image

image array
(B-spline coefficients)

Compactly supported
basis functions

SPLINE-BASED IMAGE PROCESSING

- Spline fitting: overview
  - interpolation
  - approximation
- Designing simple, efficient algorithms
  - B-spline interpolation
  - Fast multi-scale processing
- 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

- Exact fit: **interpolation** (reversible)

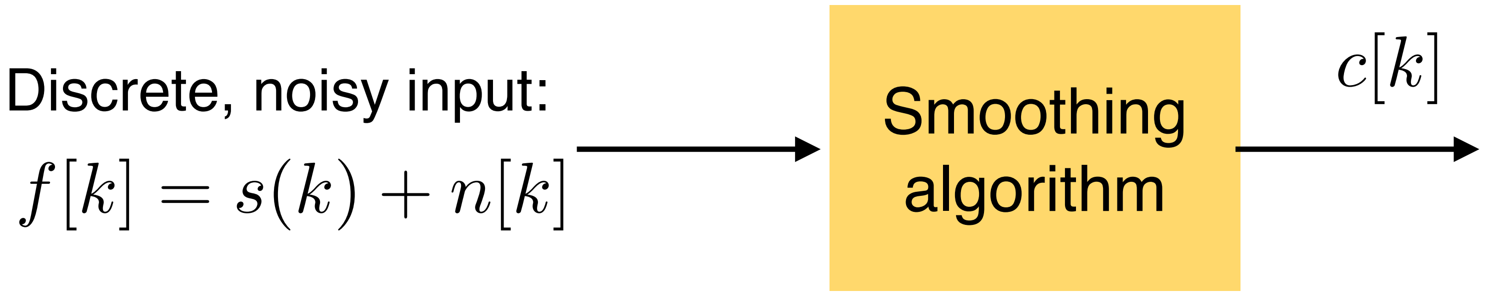
  Discrete input $f[k]$ → Interpolation algorithm → $c[k]$ such that $s(x)|_{x=k} = f[k]$

- Regularized fit: **smoothing splines**

- Least squares approximation: **spline projectors**
  - Generalized sampling theory
  - Multi-scale approximation (resizing, pyramids, wavelets)

Regularized fit: Smoothing splines

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



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

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

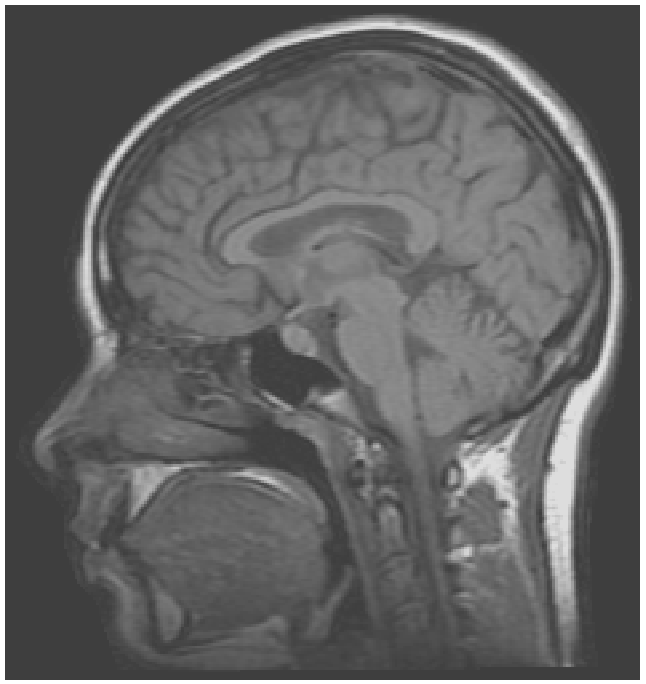
is a cardinal spline of degree $2m-1$. In addition, its coefficients $c[k] = h_\lambda * f[k]$ can be obtained by suitable digital filtering of the input samples $f[k]$.

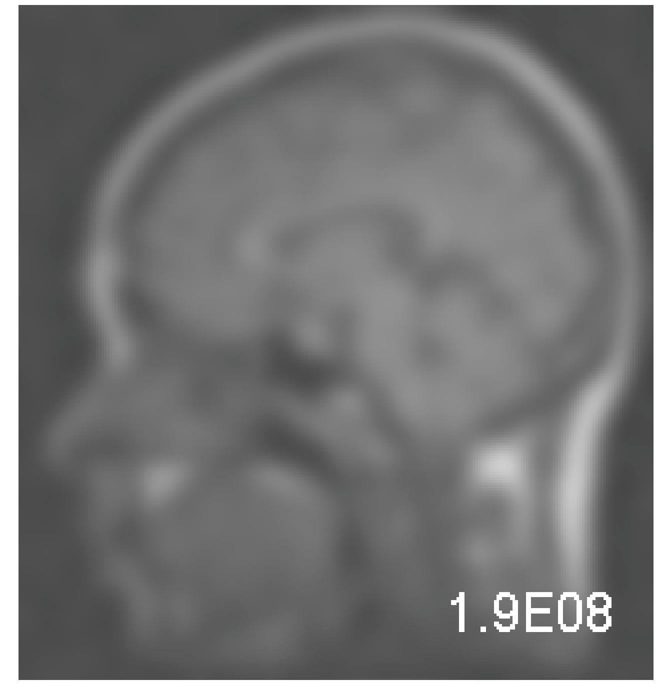
- Special case: the draftman's spline

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

It is a cubic spline !

Smoothing splines: Example





Smoothing with increasing values of λ

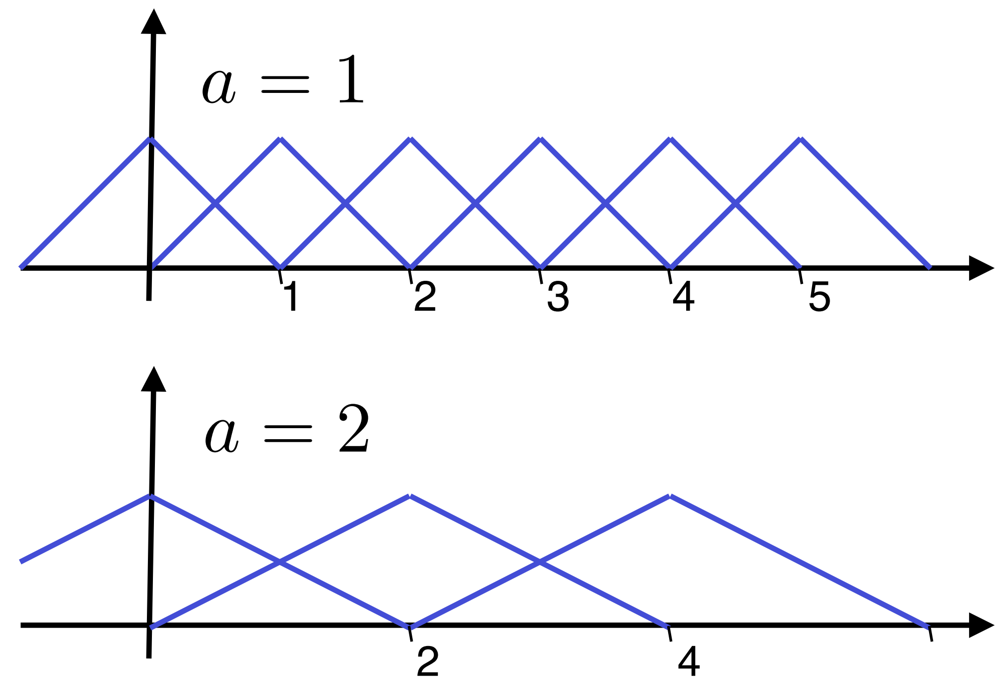
- Efficient implementation: separable, recursive filtering

Least squares fit: Multi-scale approximation

- Spline space at scale a

$$V_a = \left\{ s(x) = \sum_{k\in\mathbb{Z}} c[k]\beta_a^n(x - ak) : c[k] \in \ell_2 \right\}$$

  - Rescaled basis function: $\beta_a^n(x) := \beta^n\left(\frac{x}{a}\right)$



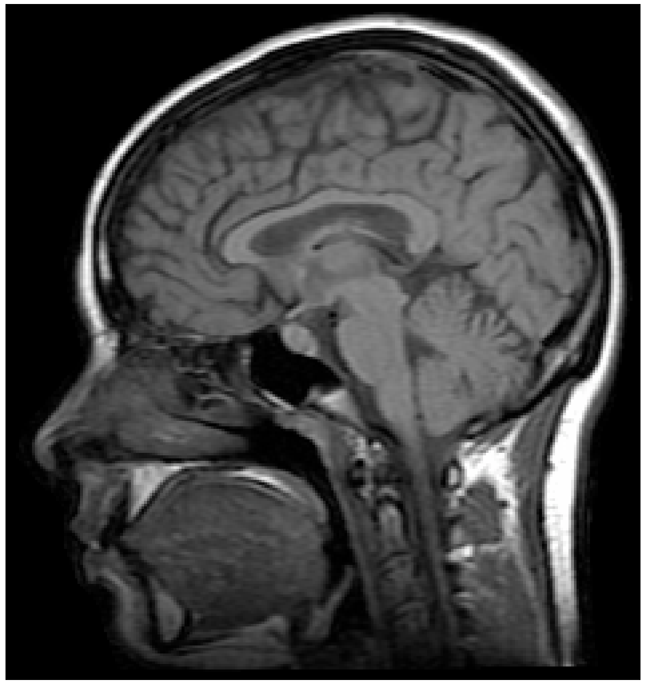
- Minimum error approximation at scale a

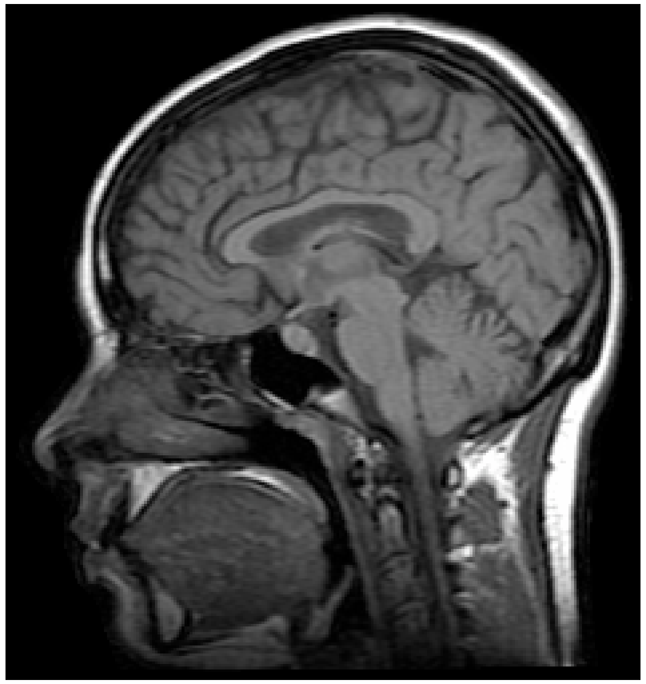
Continuous-space input $f(x)$

$c[k] = \langle f, \tilde{\beta}_a^n(\cdot - ak)\rangle$

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

Spline approximation: LS resizing





Orthogonal projection onto V_a (cubic spline)

$$a = 1 \to 10$$

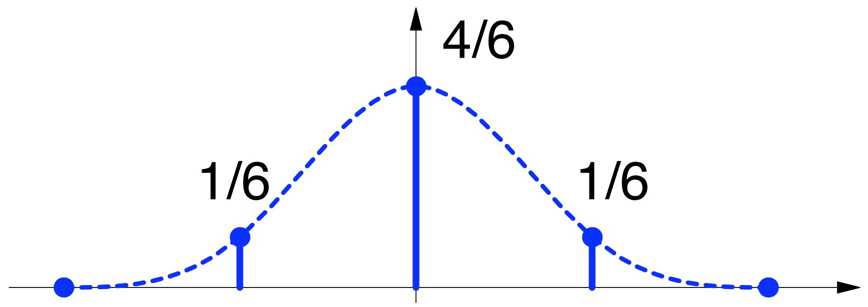
Even though splines are quite sophisticated mathematically,

... they can be implemented simply and efficiently !

B-spline interpolation made simple

- 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

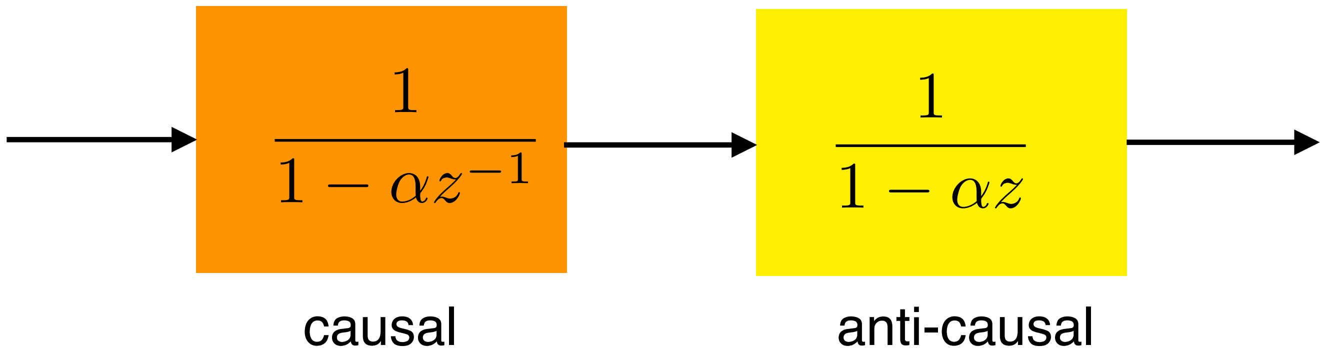
$$f[k] = \sum_{k \in \mathbb{Z}} c[l]\, \beta^n(x-l)|_{x=k} = (b_1^n * c)[k] \quad \Rightarrow \quad c[k] = (b_1^n)^{-1} * f[k]$$

- Efficient recursive implementation

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

(symmetric exponential)

Cascade of first order recursive filters



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

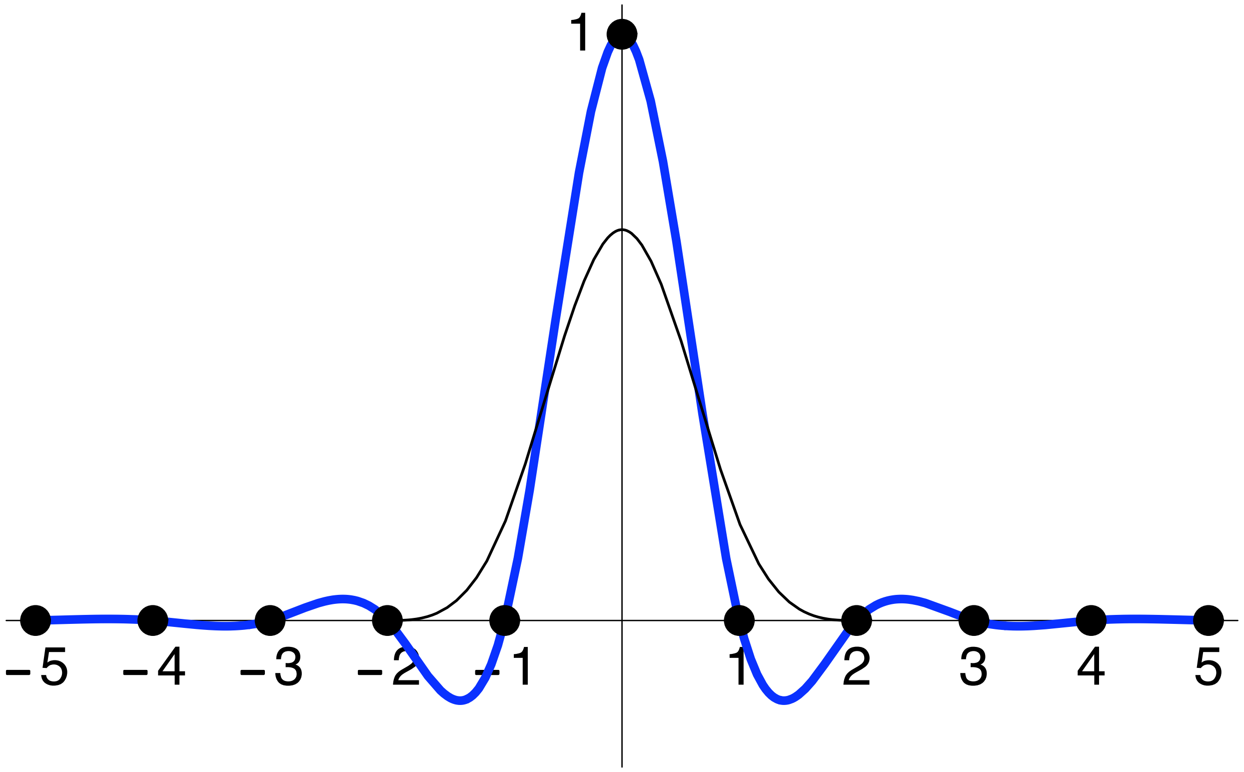
- Equivalent forms of spline representation

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

$$= \quad \sum_{k\in\mathbb{Z}} s(k)\varphi_{\text{int}}^n(x-k)$$

- Cardinal (or fundamental) spline

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



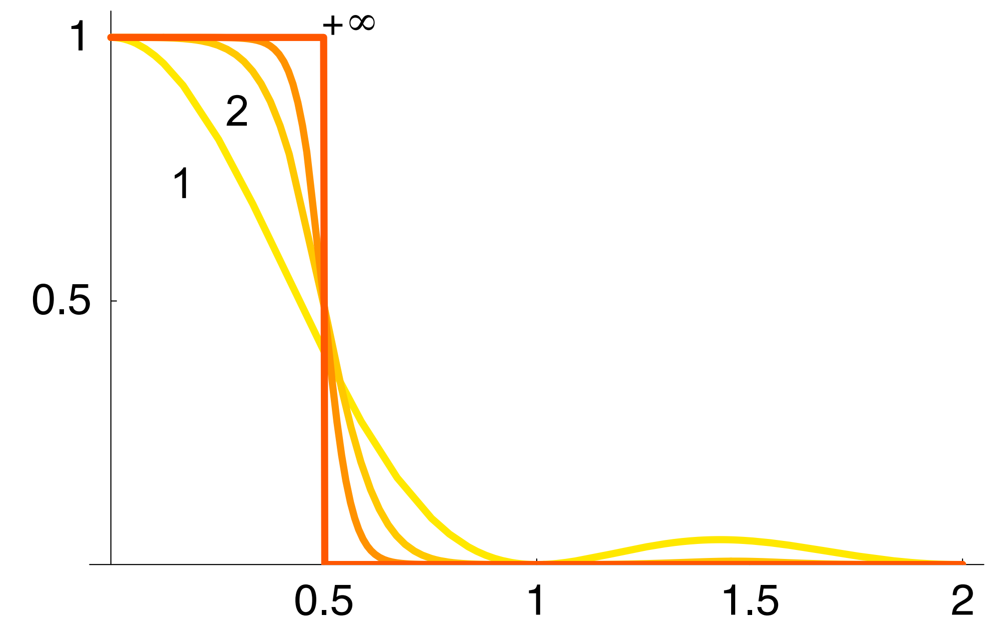
⇨ Finite-cost implementation of an infinite impulse response interpolator !

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)

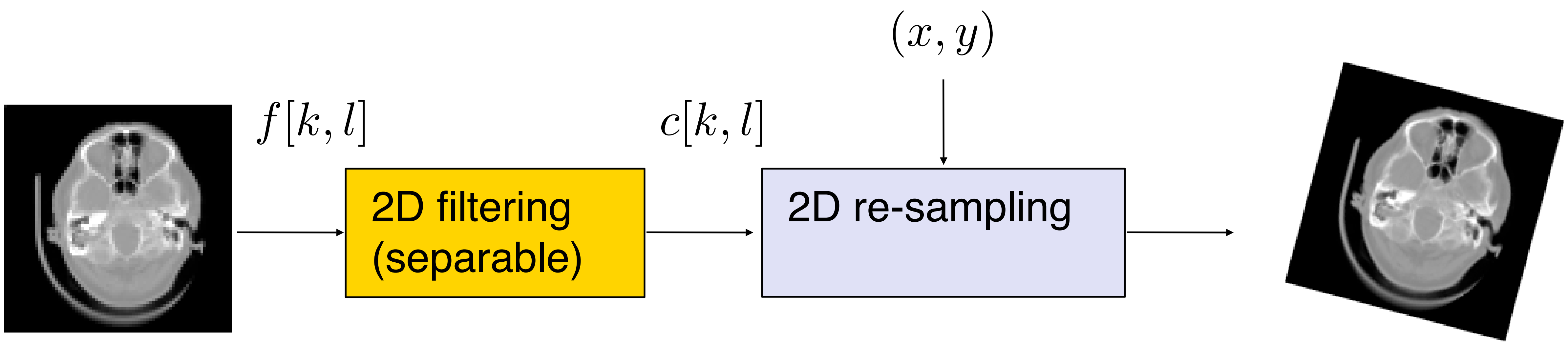
(Aldroubi et al., *Sig. Proc.*, 1992)

➡ Includes Shannon's theory as a particular case !

Geometric transformation of images

- 2D separable model

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



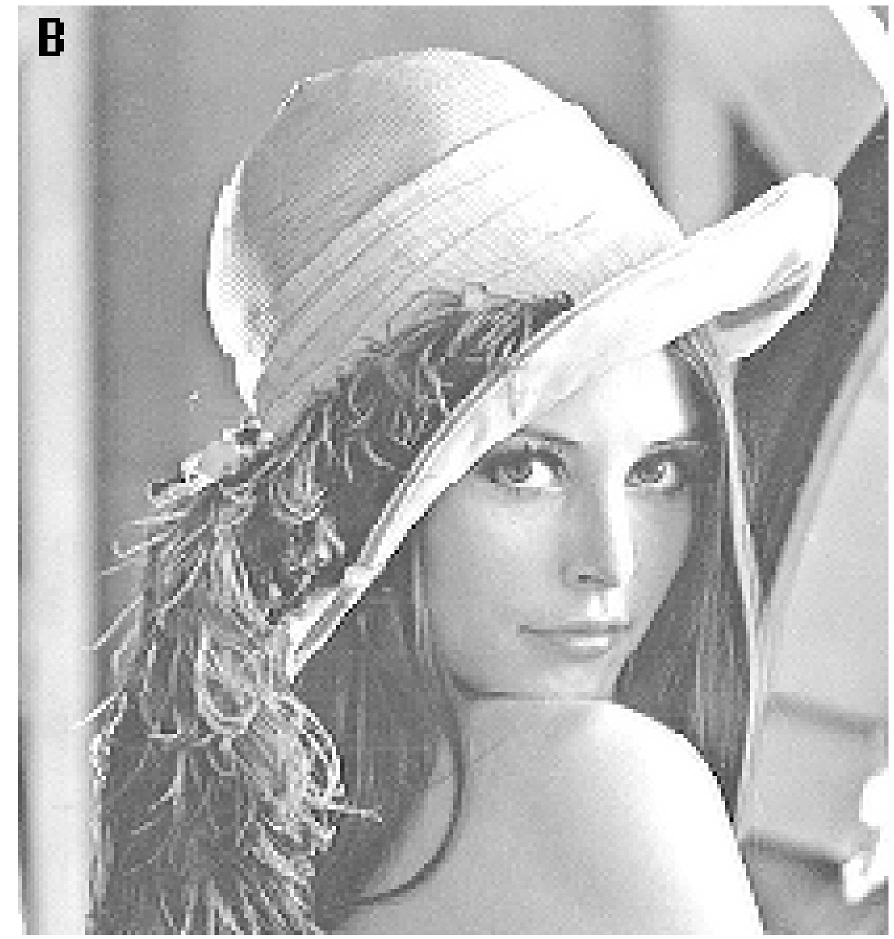
- Applications

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

Cubic spline coefficients in 2D

Pixel values $f[k, l]$

Digital filter
(recursive,
separable)



B-spline coefficients $c[k, l]$

Interpolation benchmark

Cumulative rotation experiment: the best algorithm wins !

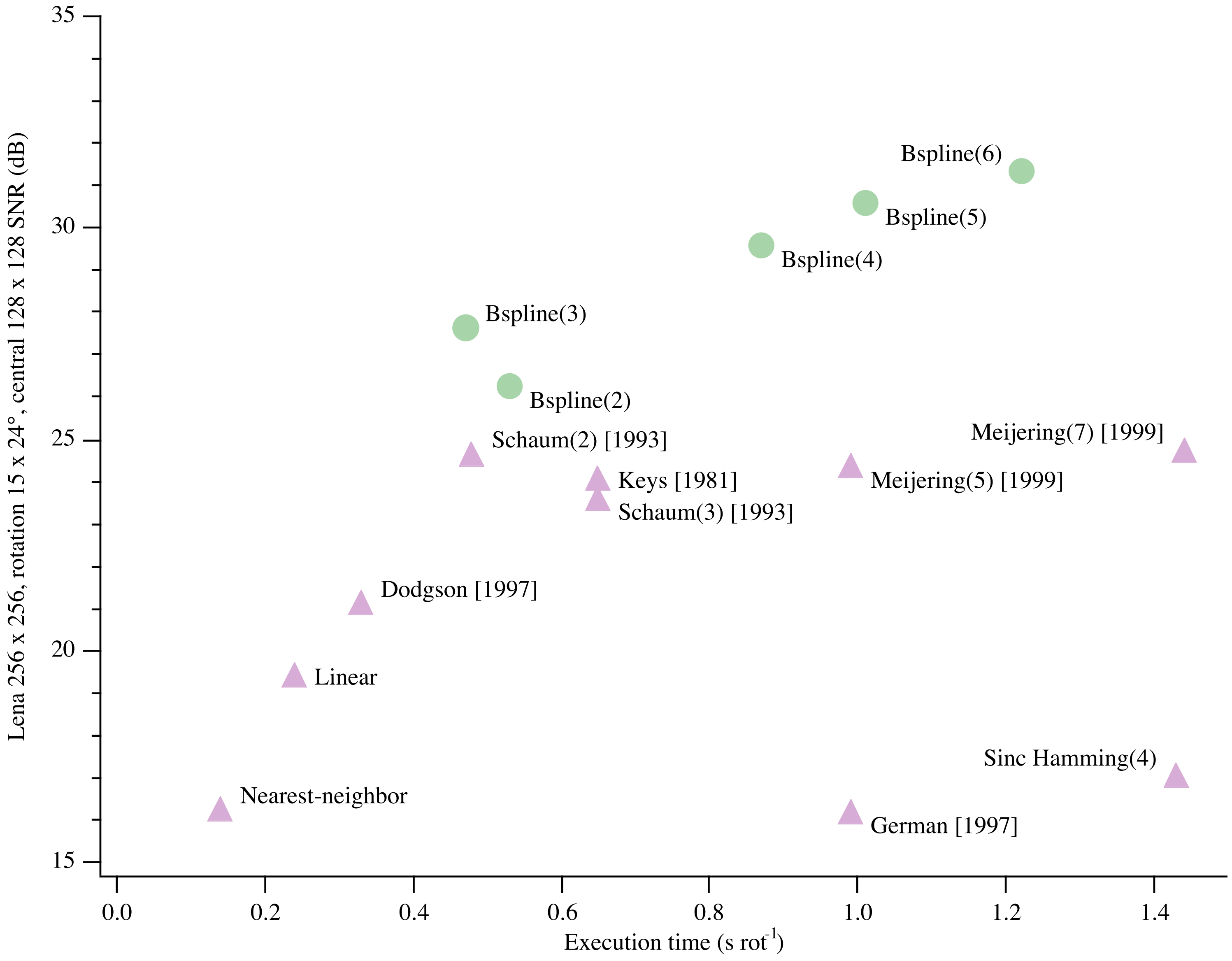

Bilinear


Windowed-sinc


Cubic spline

High-quality image interpolation

- Splines: best cost-performance tradeoff



Demo

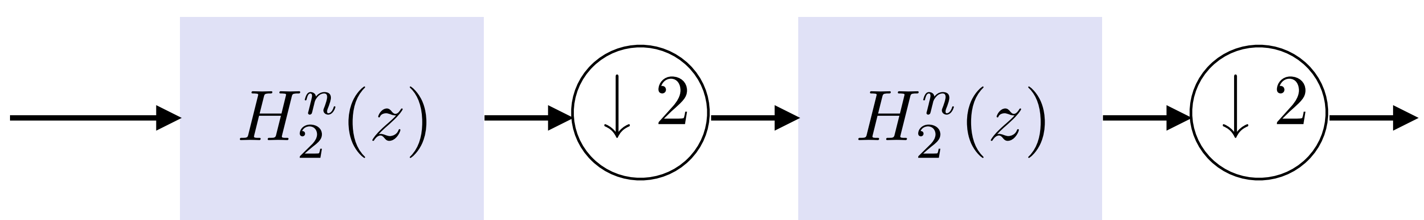
Thévenaz et al., *Handbook of Medical Image Processing*, 2000

Fast multi-scale filtering

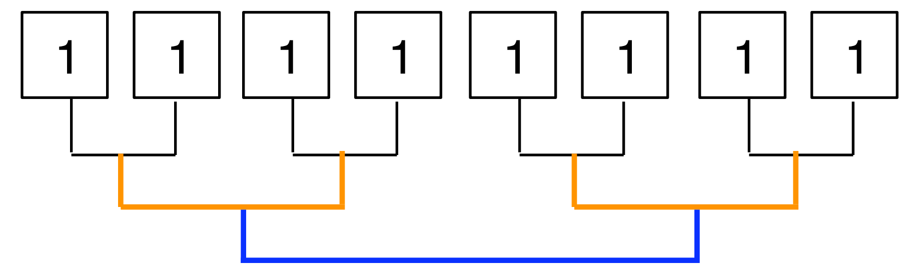
Three alternative methods for the fast evaluation of $f(x) * \beta^n(x/a)$

1) Pyramid or tree algorithms

$$a = 2^i$$



Binomial filter

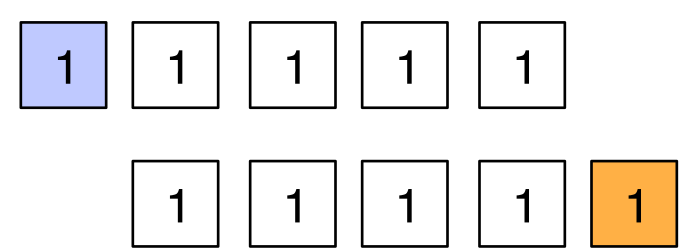


(Unser et al., *IEEE Trans. PAMI*, 1993)

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 $f(x) * \beta^n(x/a)$

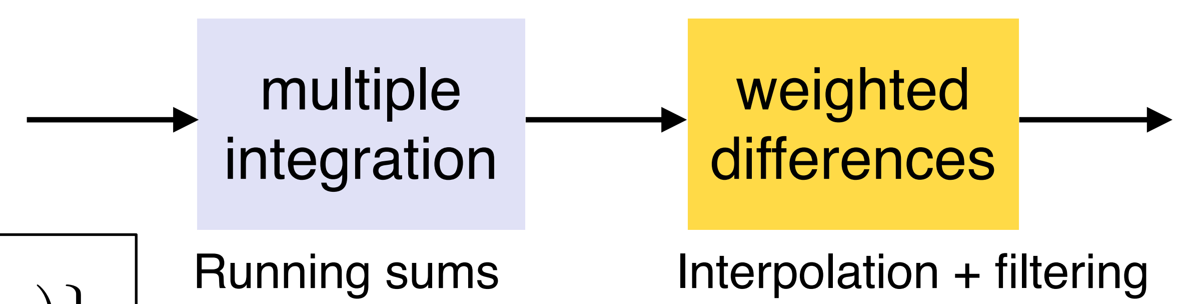
3) Differential approach $a \in \mathbb{R}^+$

$$f(x) * \beta^0_+(x/a) = F(x) - F(x-a) = \Delta^1_a \mathrm{D}^{-1}\{f(x)\}$$

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

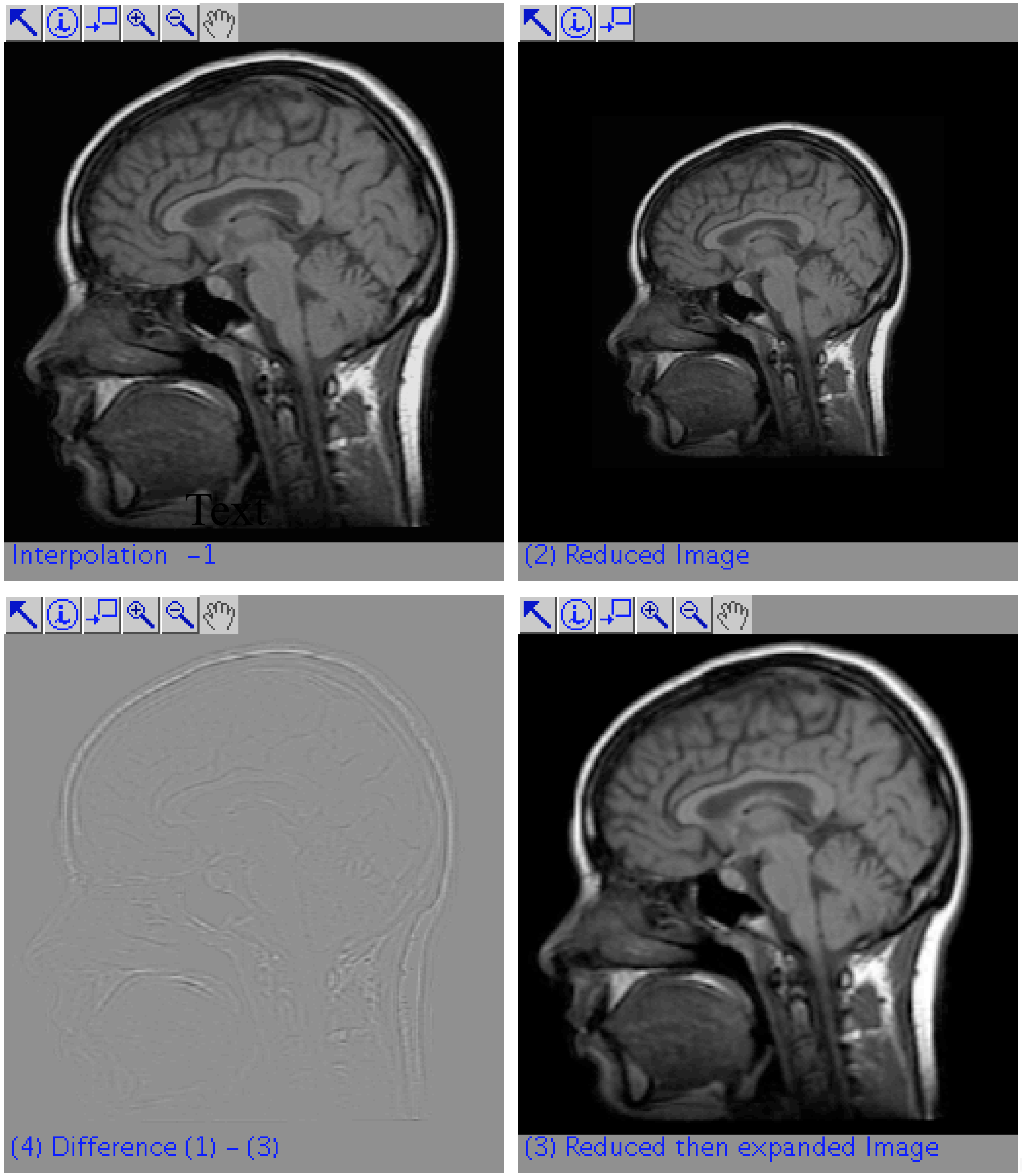


$$f(x) * \beta^n_+(x/a) = \frac{1}{a^n} \Delta^{n+1}_a \mathrm{D}^{-(n+1)}\{f(x)\}$$

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

Application: Image resizing

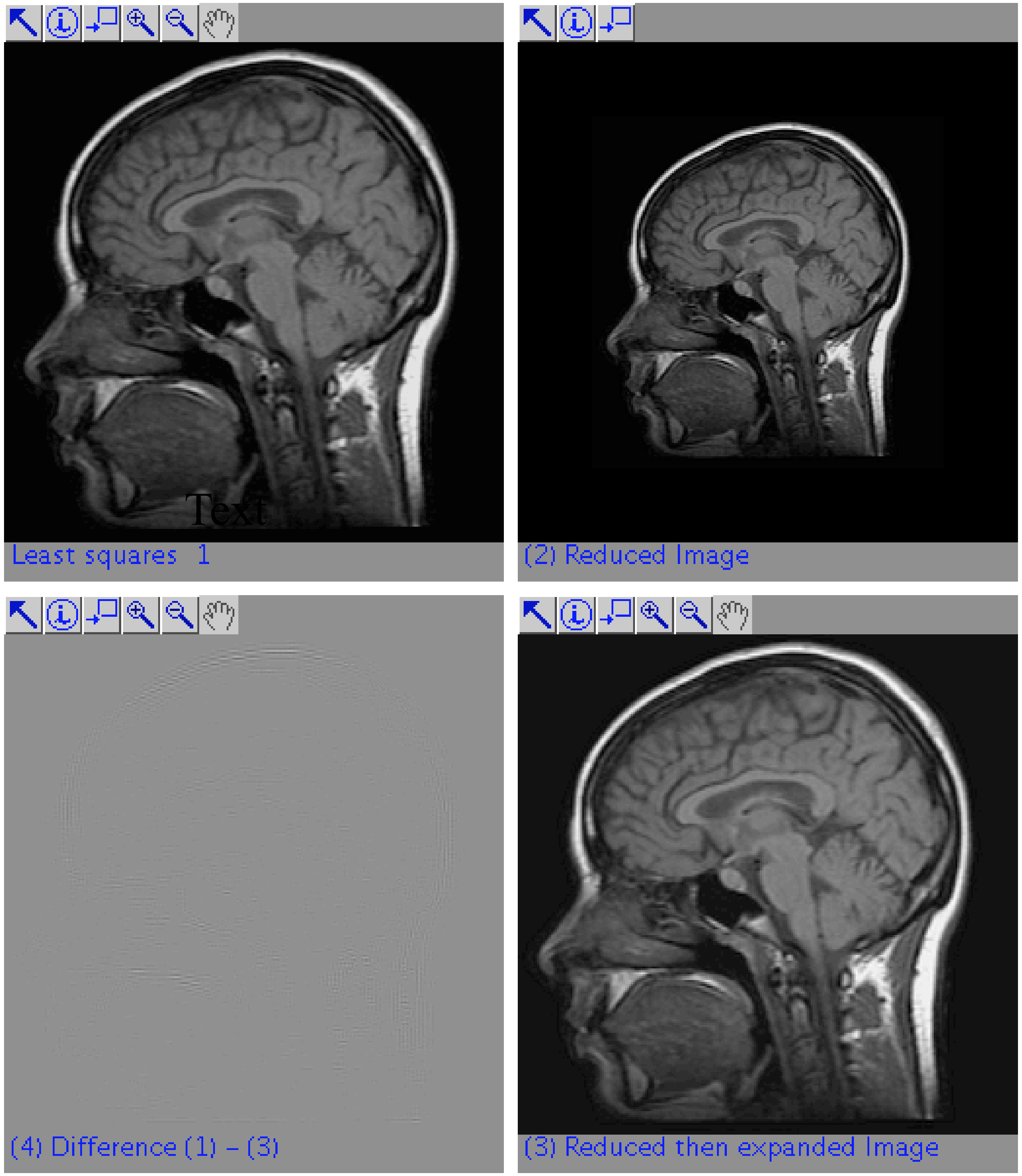
- Resizing algorithm
  - Interpolation
  - n=1
  - scaling= 70%



SNR=22.94 dB

Application: Image resizing (LS)

- Resizing algorithm
  - Orthogonal projector
  - n=1
  - scaling= 70%

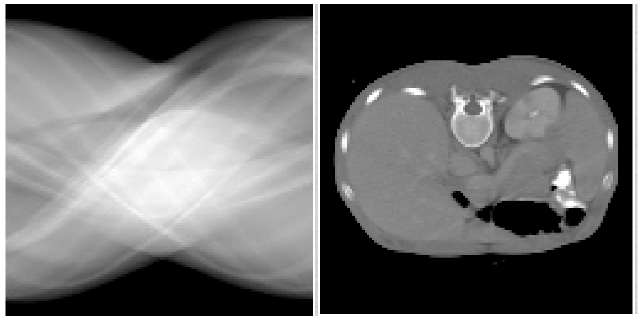


SNR=28.359 dB

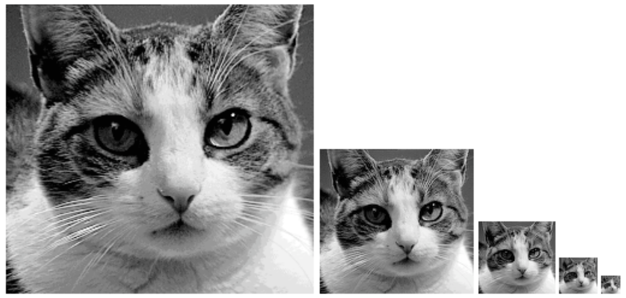
+ 5.419 dB

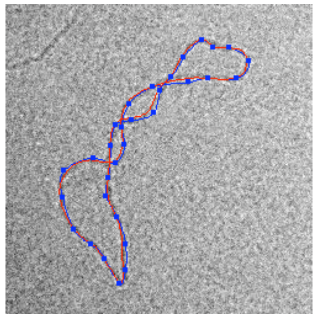
(Munoz et al., *IEEE Trans. Imag. Proc*, 2001)

Splines: More applications

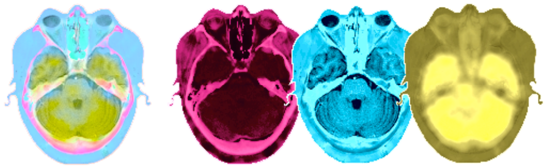


- Sampling and interpolation
  - Interpolation, re-sampling, grid conversion
  - Image reconstruction
  - Geometric correction

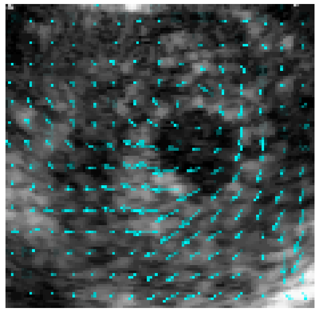




- Feature extraction
  - Contours, ridges
  - Differential geometry
  - Image pyramids
  - Shape and active contour models



- Image matching
  - Stereo
  - Image registration (multi-modal, rigid body or elastic)



- Motion analysis
  - Optical flow

SPLINES: FURTHER PERSPECTIVES

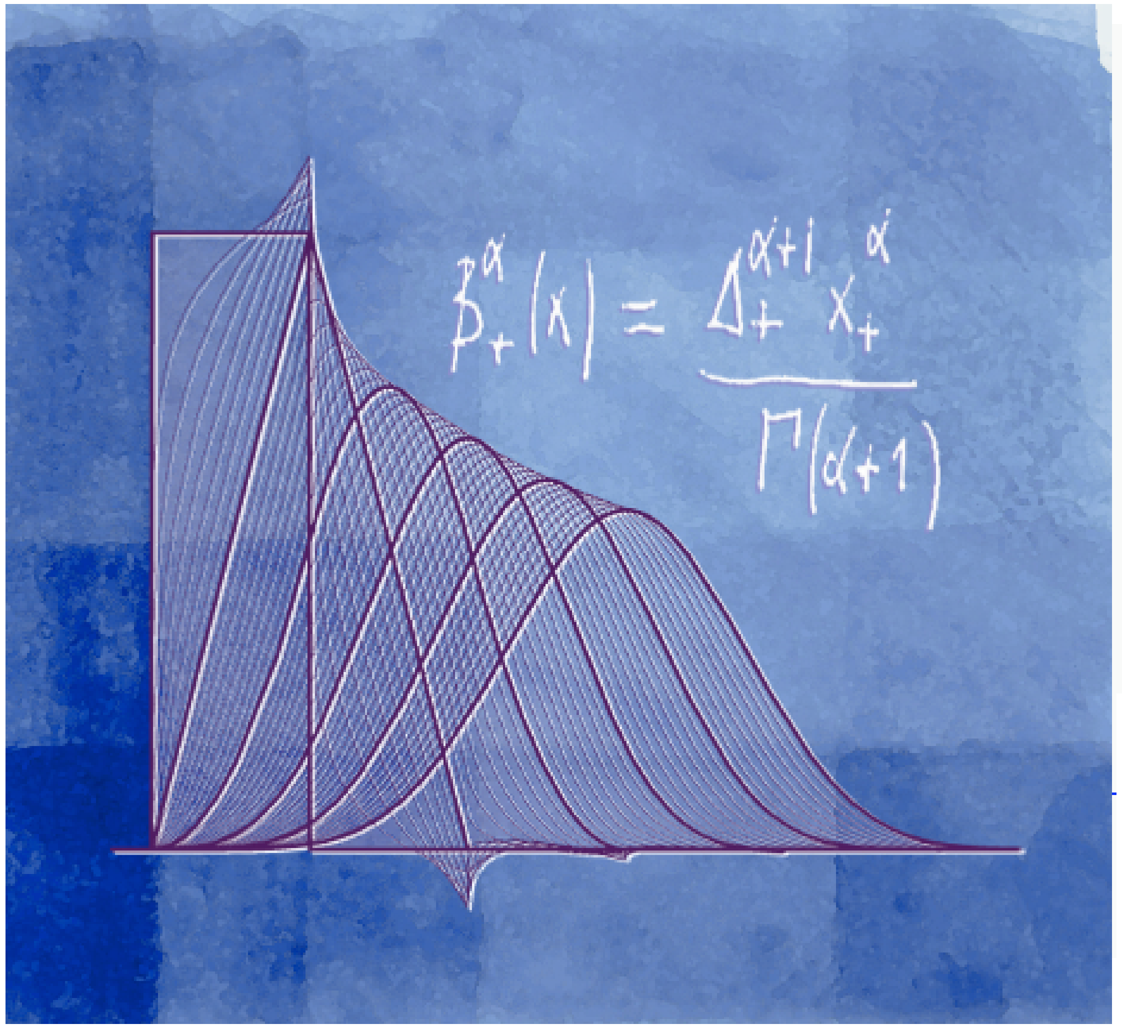
- Fractional B-splines

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

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

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

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



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

FURTHER PERSPECTIVES

- ## Splines and wavelet theory

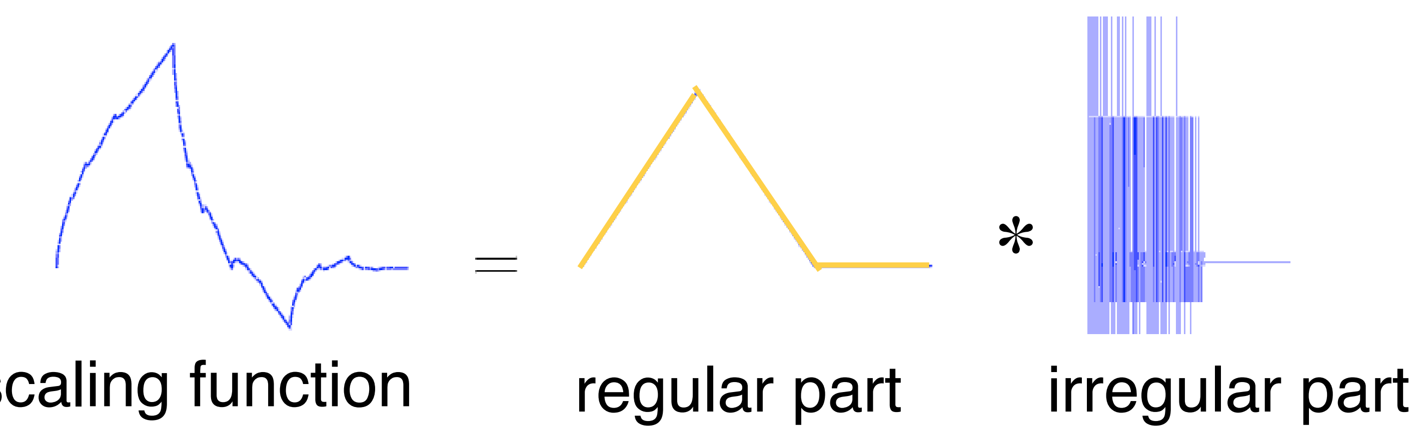
(Unser and Blu, IEEE-SP, 2003)

Factorization of any scaling function (or wavelet) of order γ:

$$\varphi(x) = \left(\beta_+^{\gamma-1} * \varphi_0\right)(x)$$

distribution

B-spline: explains all fundamental properties



- ## Splines and fractals

Splines are the optimal functions for the estimation of fractal processes with $1/\omega^{2H+1}$ spectral decay (fractional Brownian motion)

Splines: The key to wavelet theory

Sobolev smoothness

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

$\Downarrow \quad \gamma \geq s$

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

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

$\Downarrow \quad \Uparrow$

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:

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

(Unser and Blu, IEEE-SP, 2003)

CONCLUSION

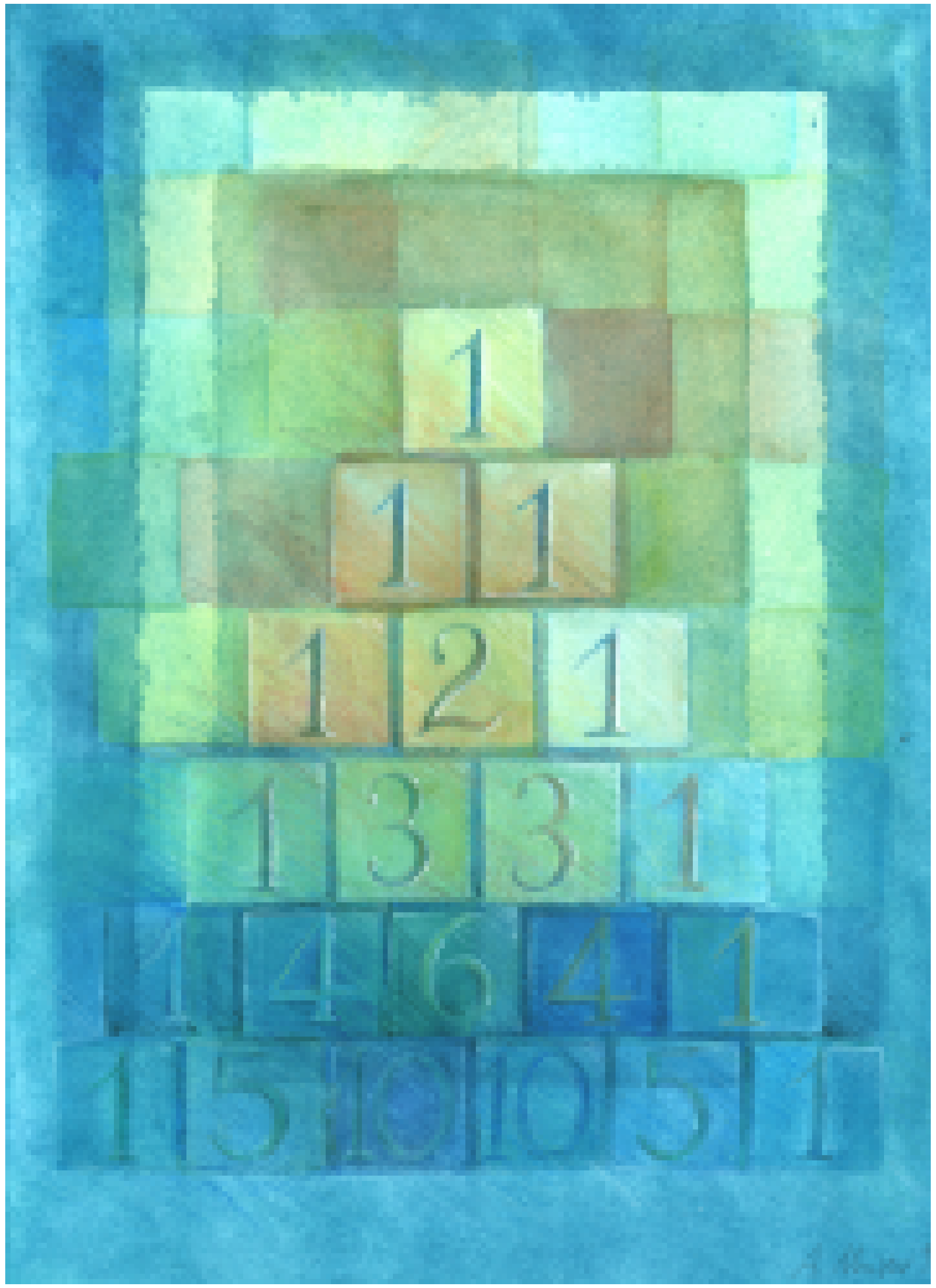
- Distinctive features of splines
  - Simple to manipulate
  - Smooth and well-behaved
  - Excellent approximation properties
  - Multiresolution properties
  - Fundamental nature (Green functions of derivative operators)
- Splines and image processing
  - A story of avoidance and, more recently, love…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

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*, vol. 53, no. 6, pp. 2146-2159, June 2005.
- Preprints and demos: **http://bigwww.epfl.ch/**