IEEE International Conference on Acoustics, Speech, and Signal Processing, June 5-9, ICASSP'2000, Istanbul, Turkey.

SPLINE KERNELS FOR CONTINUOUS-SPACE IMAGE PROCESSING

Stefan Horbelt, Arrate Muñoz, Thierry Blu and Michael Unser

Biomedical Imaging Group, DMT/IOA
Swiss Federal Institute of Technology Lausanne
CH-1015 Lausanne EPFL, Switzerland

ABSTRACT

We present an explicit formula for spline kernels; these are defined as the convolution of several B-splines of variable widths h_i and degrees n_i. The spline kernels are useful for continuous signal processing algorithms that involve B-spline inner-products or the convolution of several spline basis functions. We apply our results to the derivation of spline-based algorithms for two classes of problems. The first is the resizing of images with arbitrary scaling factors. The second is the computation of the Radon transform and of its inverse; in particular, we present a new spline-based version of the filtered backprojection algorithm for tomographic reconstruction. In both cases, our explicit kernel formula allows for the use of high-degree splines; these offer better approximation performance than the conventional lower-order formulations (e.g., piecewise constant or piecewise linear models).

keywords: spline kernel, anti-aliasing, resizing, Radon transform, filtered back-projection

1. INTRODUCTION

Continuous-space image processing deals with problems that are specified in the continuous domain. In the Hilbert-space formulation, images are represented as linear combinations of continuously-defined basis functions $\varphi_k(x)$. A convenient representation uses tensor-product basis functions obtained by uniformly translating a single prototype: $\varphi_k(x) = \varphi(x-k)$. If the continuous signal processing operator T to be implemented is linear, we can discretize it by determining its effect on the basis functions: $T\{\varphi_k\}$. The least-squares solution is obtained by projecting $T\{\varphi_k\}$ onto the same discrete-continuous function space $V = \text{span}\{\varphi_k\}_{k\in\mathbb{Z}}$ [9]. In practice, the projection involves the computation of inner products with a set of analysis functions $\tilde{\varphi}_k$ that are the duals of φ_k. B-splines basis functions are well suited for this kind of computation [7]. In particular, they have been applied to the problems of image resizing [2],[4],[8] and to tomographic reconstruction by filtered back-projection [1],[3]. The main difficulty of the above-mentioned algorithms is that they involve the computation of inner products (or continuously-defined convolutions) of B-splines of different widths. Until now, exact formulas had only been available for relatively low order models: $n \leq 1$ for least-squares image resizing, and $n = 0$ for tomography. Higher-order solutions were considered for image resizing but either they used a Gaussian approximation of B-spline convolutions [8], or they computed a

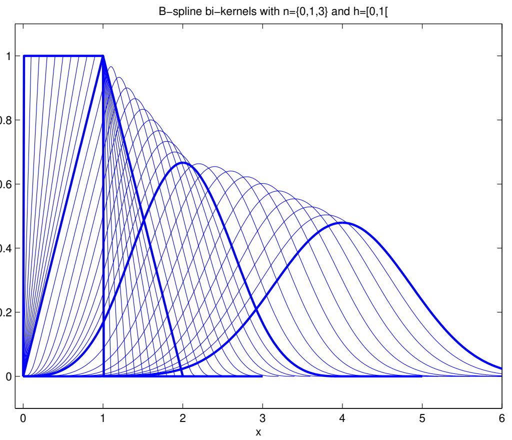

Figure 1: Spline bi-kernels by convolution of two causal (non-centered) B-splines $\beta_1^n(x) * \beta_h^n(x)$ with $n = \{0, 1, 3\}$ and $h \in [0, 1[$. The kernels generate a smooth transition between B-splines of degree n and degree $2n+1$. For $h = 0$ and $h = 1$ the kernels are B-splines, as $\beta_1^n(x) * \beta_{h=0}^n(x) = \beta_1^n(x)$ and $\beta_1^n(x) * \beta_{h=1}^n(x) = \beta_1^{2n+1}(x)$.

sub-optimal solution by oblique projection instead of an orthogonal one [4].

In this paper, we propose to develop an explicit solution applicable for higher-order splines. It is based on what we call B-spline convolution kernels. To use these kernels, both the input and the output signals have to be polynomial splines. Moreover, the basis functions and analysis functions have to be B-splines. This means that the input signal has to be expressed in terms of B-splines, while the output must be represented in terms of their duals (D-spline) so that the optimal analysis functions $\tilde{\varphi}_k$ are B-splines as well. Unser et al. described efficient routines to change between the standard (cardinal) representation of a signal in terms of its samples, its B-spline expansion, and its representation in terms of dual splines [7].

2. B-SPLINE CONVOLUTION KERNELS

2.1. The B-spline function

Instead of the usual definition of a B-spline ($(n+1)$-fold convolution of a box function), we choose to express the centered B-spline of degree n and width h as a sum of shifted and weighted one-sided power functions

$$\beta_h^n(x) = \frac{1}{h}\beta^n\left(\frac{x}{h}\right) = \Delta_h^{n+1} * \frac{(x)_+^n}{n!}, \qquad (1)$$

where $(x)_+^n = \max(0,x)^n$ is the one-sided power function and where Δ_h^{n+1} is the centered finite-difference operator at width h:

$$\Delta_h^{n+1} = \sum_{k=0}^{n+1}(-1)^k \binom{n+1}{k} \frac{\delta(x + h\cdot(\frac{n+1}{2} - k))}{h^{n+1}}. \qquad (2)$$

Thus we can write, the explicit form of the B-spline:

$$\beta_h^n(x) = \sum_{k=0}^{n+1}(-1)^k \binom{n+1}{k} \frac{(x + h\cdot(\frac{n+1}{2} - k))_+^n}{n!\cdot h^{n+1}}.$$

Its support is $[-h(n+1)/2\,,\ h(n+1)/2[$, its degree is n, and its Fourier transform is $\widehat{\beta}_h^n(\omega) = \mathrm{sinc}^{n+1}(\frac{\omega\cdot h}{2\pi})$.

2.2. Convolution properties

We can show that the convolution of several one-sided power functions of degrees $n_1,\ldots,n_m$ is again a one-sided power function

$$x_+^{n_1} * \ldots * x_+^{n_m} = x_+^{m-1+\sum_{i=1}^m n_i} = x_+^{N_m}; \qquad (3)$$

where the degree is $N_m = m - 1 + \sum_{i=1}^m n_i$.

The convolution of several difference operators $\Delta_{h_1}^{n_1}, \ldots, \Delta_{h_m}^{n_m}$ is written as

$$\Delta_{h_1,\ldots,h_m}^{n_1,\ldots,n_m} = \Delta_{h_1}^{n_1} * \ldots * \Delta_{h_m}^{n_m}.$$

In the case where they all have the same width h, the difference operator simplifies to

$$\Delta_{h,\ldots,h}^{n_1+1,\ldots,n_m+1} = \Delta_h^{m+\sum_{i=1}^m n_i} = \Delta_h^{N_m+1}. \qquad (4)$$

By combining (3) and (4), we derive the well-known convolution property of the B-splines:

$$\beta_h^{n_1} * \beta_h^{n_2}(x) = \Delta_h^{n_1+n_2+2} * \frac{x_+^{n_1+n_2+1}}{(n_1+n_2+1)!} = \beta_h^{n_1+n_2+1}(x).$$

2.3. The B-spline convolution kernel

The convolution of two B-splines of degrees n_1, n_2 and widths h_1, h_2, called spline bi-kernel (Fig.1), is given by

$$\beta_{h_1,h_2}^{n_1,n_2}(x) = \beta_{h_1}^{n_1} * \beta_{h_2}^{n_2}(x) = \Delta_{h_1,h_2}^{n_1+1,n_2+1} * \frac{x_+^{n_1+n_2+1}}{(n_1+n_2+1)!}. \qquad (5)$$

In the same way, we define the spline m-kernel, which is the convolution of a set of m B-splines of degrees $n_1,\cdots,n_m$ and widths $h_1,\cdots,h_m$. It can be evaluated as follows:

$$\beta_{h_1,\ldots,h_m}^{n_1,\ldots,n_m}(x) = \beta_{h_1}^{n_1} * \ldots * \beta_{h_m}^{n_m}(x) = \Delta_{h_1,\cdots,h_m}^{n_1+1,\cdots,n_m+1} * \frac{x_+^{N_m}}{N_m!}. \qquad (6)$$

Its support is $\left[-\sum_{i=1}^m h_i\,(n_i+1)\,/2, \sum_{i=1}^m h_i\,(n_i+1)\,/2\right[$. The spline m-kernel is a non-uniform spline of degree $N_m = m - 1 + \sum_{i=1}^m n_i$. The spline m-kernel formula can be expanded using (2) to obtain a closed-form expression.

3. RESIZING

We now use these kernels to obtain an exact implementation of the least-squares resizing method [8] for splines of degree $n > 1$. The proposed algorithm is an alternative to the standard interpolation which performs resizing by interpolation and resampling. In the least-squares formulation, the resampling step is replaced by projection using an evaluation of inner products with the analysis functions $\tilde{\varphi}_k$. In essence, this is equivalent to applying an analog prefilter prior to sampling. The advantage is a suppression of aliasing and a reduction of blocking artifacts. The algorithm, which works for arbitrary scaling factors (enlargement or reduction), is described next.

Since resizing is a separable procedure, we can describe the method in 1D. The input is a discrete signal f_k. The first step is to construct a spline interpolant

$$f(x) = \sum_k c_k \beta^{n_1}(x-k) \qquad (7)$$

with coefficients $c_k = g * f_k$. The change from cardinal to B-spline representation is achieved by digital filtering; the transfer function of the direct B-spline filter is $G(z) = 1/\sum_k \beta_k^{n_1} z^{-k}$ (see [7]). The coefficients c_k can be retrieved from the continuous function $f(x)$ by the inner product

$$c_k = \left\langle f(x), \widetilde{\beta}^{n_1}(x-k)\right\rangle.$$

We resize the image by the factor a and shift it by b. The image is enlarged if $a > 1$ and shrunk if $a < 1$. The least-squares approximation of the resized continuous image in the space $V \in \mathrm{span}\{\widetilde{\beta}^{n_2}(x-l)\}$ is

$$f_a(x) = \sum_{l\in\mathbb{Z}} \underbrace{\left\langle f(\frac{x-b}{a}), \beta^{n_2}\,(x-l)\right\rangle}_{d_l} \cdot \widetilde{\beta}^{n_2}(x-l). \qquad (8)$$

As the scalar product is equivalent to a sampled convolution, we have $d_l = \sum_k c_k \cdot \left\langle \beta^{n_1}(\frac{x-b}{a} - k), \beta^{n_2}\,(x-l)\right\rangle = \sum_k c_k \cdot a \cdot \beta_{a,1}^{n_1,n_2}(l - ak - b)$.

Finally, we resample the rescaled approximation $f_a(x)$; this is achieved by simple digital filtering (conversion from dual to cardinal representation).

We propose a fast implementation that takes advantage of the short support of the bi-kernel and of the fact that the sampled values $\beta_{a,1}^{n_1,n_2}(l - ak - b)$ can be precomputed for all relevant values of k and l. These values are then stored in a lookup table that can be used repeatedly to resize the rows (and columns) of the image.

4. RADON TRANSFORM AND FILTERED BACK-PROJECTION

In this section, we recall some standard results on the Radon transform and on its inverse [5]; these will be used in the next section to obtain a discretization using splines.

4.1. Radon transform

The Radon transform R_θ of an image $f(\vec{x})$, $\vec{x} \in R^2$, is the set of line integrals along the direction $\vec{\theta}$ at the distance t from the origin

$$R_\theta\{f(\vec{x})\} = R_\theta f(t) = \int_{\vec{x}\in\mathbb{R}^2} f(\vec{x})\delta(t - \vec{x}^\top \cdot \vec{\theta})d\vec{x}, \qquad (9)$$

where $\delta(t)$ is the Dirac impulse and $\vec{\theta} = (\cos\theta, \sin\theta)^\top \in \mathbb{S}^2$ is a unit vector specifying the direction of the integration.

4.2. Filtered back-projection

The basis for the inverse Radon transform [3] is the well-known identity

$$f(\vec{x}) = R^* K R_\theta\{f(\vec{x})\} = R^*(q * R_\theta\{f(\vec{x})\}), \qquad (10)$$

where K is a filter, in which each projection is convolved with the 1D ramp filter q defined in Fourier by $\hat{q}(\omega) = |\omega|$.

The backprojection operator R^* is the adjoint of R:

$$(R^* p)(\vec{x}) = \int_0^\pi p(t, \theta)d\theta,$$

where $t = \vec{x}^\top \cdot \vec{\theta}$. The widely used filtered back-projection (FBP) algorithm corresponds to the direct discretization of the right-hand side of the inversion formula (R^*K).

5. SPLINE-BASED RADON TRANSFORM AND FILTERED BACK-PROJECTION

Guédon et al. [1] have shown that the standard FBP could be improved by using an explicit piecewise constant model of the image with basis functions that are B-splines of degree $n = 0$. Here, we will use our B-spline convolution kernels to extend the approach to higher-order splines. We also consider the direct problem which is the discretization of the Radon transform. We use splines consistently to represent the image as well as the sinogram (i.e., the Radon transform).

5.1. Radon transform of a B-spline

The image is represented using basis functions generated by the 2D separable B-spline of degree n: $\beta_h^n(\vec{x}) = \beta_h^n(x)\beta_h^n(y)$. Using the Fourier slice theorem, we can determine the Radon transform of the 2D B-spline by

$$R_\theta\{\beta_h^n(\vec{x})\} = \beta^n_{h\cos\theta}(t) * \beta^n_{h\sin\theta}(t) = \beta^{n,n}_{h\cos\theta, h\sin\theta}(t), \quad (11)$$

where $t = \vec{x}^\top\vec{\theta}$; it is precisely a B-spline bi-kernel whose explicit form is given by (5). Also, note that, when the basis function is shifted by $h\vec{k}$, $\vec{k} = (k, l)$, its Radon transform $R_\theta\{\beta_h^n(\vec{x} - h \cdot \vec{k})\} = R_\theta\beta_h^n(t - h\vec{k}^\top\vec{\theta})$ is shifted by $h\vec{k}^\top\vec{\theta} = hk\cos\theta + hl\sin\theta$.

5.2. Spline-based Radon transform

The basis functions in each of the following steps are chosen to benefit from the spline kernel. The input image $f(\vec{x})$ is approximated using B-splines $f_h(\vec{x}) = \sum_{k,l} c_{k,l}\beta_h^n(\vec{x} - h\vec{k})$, where $c_{k,l} = h^2\left\langle f(\vec{x}), \tilde{\beta}_h^n(\vec{x} - h\vec{k})\right\rangle$. Note the factor h^2 from the biorthogonality condition $h^2\left\langle \beta_h^n(\vec{x}), \tilde{\beta}_h^n(\vec{x} - h\vec{k})\right\rangle = \delta(\vec{k})$. The Radon transform of $f_h(t)$ is

$$g_\theta(t) = R_\theta f_h(t) = \sum_{k,l} c_{k,l} R_\theta \beta_h^n(t - h\vec{k}^\top\vec{\theta}). \qquad (12)$$

The key quantity that needs to be determined is the least-squares spline approximation of the Radon transform of the 2D basis function: $R_\theta\beta_h^n(t - h\vec{k}^\top\vec{\theta})$. It is given by

$$P_s\{R_\theta\beta_h^n(t - h\vec{k}^\top \cdot \vec{\theta})\} =$$
$$= \sum_i s \cdot \left\langle R_\theta\beta_h^n(t - h\vec{k}^\top\vec{\theta}), \beta_s^n(t - is)\right\rangle \tilde{\beta}_s^n(t - is) =$$
$$= \sum_i s \cdot \underbrace{\beta^{n,n,n}_{h\cos\theta, h\sin\theta, s}(h\vec{k}^\top\vec{\theta} - is)}_{d_{i,\theta,k,l}} \tilde{\beta}_s^n(t - is) \qquad (13)$$

and can be calculated using our explicit formula (6) for the B-spline tri-kernel. The width of β_s^n is s. Finally we get

$$g_{\theta,s}(t) = P_s\{g_\theta(t)\} = \sum_i\sum_{k,l} c_{k,l} \cdot s \cdot d_{i,\theta,k,l} \cdot \tilde{\beta}_s^n(t - is). \quad (14)$$

5.3. Spline-based filtered back-projection

The first step is to filter the sinogram $\hat{g}_{\theta,s}(\omega)$ in Fourier with an ideal ramp filter (see section 4.2):

$$\hat{g}_{q,\theta}(\omega) = \hat{g}_{\theta,s}(\omega) \cdot \hat{q}(\omega).$$

Next, in order to invert the Radon transform with the spline kernel, the filtered signal is projected onto some spline space and represented using B-spline basis functions:

$$g_{s,q,\theta}(t) = P_s g_{q,\theta}(t) = \sum_i c_{\theta,i} \cdot \beta_s^n(t - is),$$

where $c_{\theta,i} = s \cdot \langle\tilde{\beta}_s^n(t - is), g_{q,\theta}(t)\rangle$.

The next step is to calculate the back-projection $R_\theta^* g_{s,q,\theta}(t)$ and to approximate it in the image space, using dual B-spline basis functions

$$\tilde{f}_h(\vec{x}) = \sum_{k,l} \tilde{c}_{k,l} \cdot \tilde{\beta}_h^n(\vec{x} - h\vec{k}), \qquad (15)$$

where $\tilde{c}_{k,l} = h^2 \cdot \sum_\theta \langle R_\theta^* g_{s,q,\theta}(t), \beta_h^n(t - h\vec{k}^\top\vec{\theta})\rangle$.

Proposition 1: For any given angle θ, the following adjoint relationship holds: $\langle f, R_\theta^* g\rangle = \langle R_\theta f, g\rangle$.

Using Proposition 1, the coefficients $\tilde{c}_{k,l}$ are written as

$$\tilde{c}_{k,l} = \sum_{i,\theta} c_{\theta,i} \cdot h^2 \cdot \underbrace{\langle\beta_s^n(t - is), R_\theta\beta_h^n(t - h\vec{k}^\top\vec{\theta})\rangle}_{d_{i,\theta,k,l}}, \qquad (16)$$

where also $d_{i,\theta,k,l} = \beta^{n,n,n}_{h\cos\theta, h\sin\theta, s}(h\vec{k}^\top\vec{\theta} - is)$ is the same spline kernel as in the spline-based Radon transform 13. Again, the choice of spaces was guided by the spline kernel usage. A fast algorithm to obtain $d_{i,\theta,k,l}$ consists of precalculating the B-spline tri-kernel at fine resolution so that we can use a lookup table to calculate $d_{i,\theta,k,l}$ efficiently.

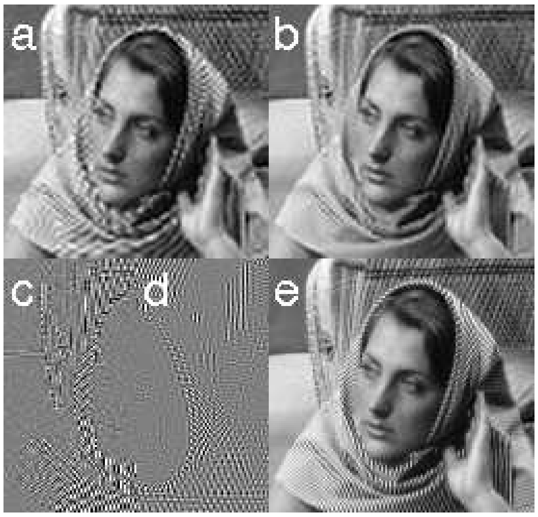

Figure 2: **Resizing:** The images were shrunk by $\sqrt{3}$ and then enlarged by the same factor. (e) original "Barbara", (a) standard resizing by cubic interpolation and sampling, (b) resizing with spline kernel of degrees $n_{1,2} = 3$, (c) error image of left side of (a), (d) error image of right side of (b).

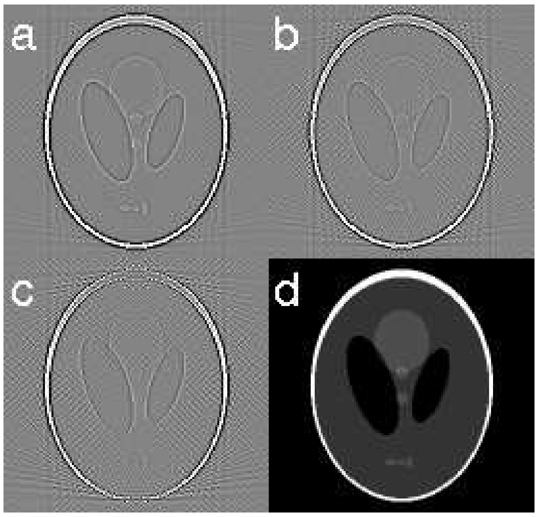

Figure 3: **Filtered back-projection:** (d) original image, (a) error of standard reconstruction with linear interpolation ($n = 1$) and Shepp-Logan filter, (b) error of reconstruction with spline tri-kernel of degrees $n_i = 0$, (c) error of reconstruction with spline tri-kernel of degrees $n_i = 1$.

6. EXPERIMENTS

In both experiments, we compare the standard approaches, based on interpolation and sampling, with the least-squares optimal spline procedure that uses our spline kernels.

Resizing: The test image "Barbara" with size 128×128 has been scaled down by a factor of $\sqrt{3}$ and then enlarged by the same factor. Fig.2a shows the result for the standard algorithm with cubic interpolation and sampling. The PSNR is 22.48 dB. The result for resizing with the cubic spline tri-kernel $n_{1,2} = 3$ is displayed in Fig.2b (PSNR=23.84 dB). Improvements are due to the fact that the spline kernel acts like an optimal anti-alias filter. The error images show that the later method preserves more details (Fig.2d) than the standard method (Fig.2c).

Filtered back-projection: The Shepp-Logan phantom ([6], Fig.3d) is of size 128×128. Its Radon transform was computed over 256 equidistant angles with linear interpolation. Fig.3a displays the reconstruction error for the standard algorithm (Shepp-Logan filter) with linear interpolation for the back-projection. The PSNR is 26.89 dB. The reconstruction error for the proposed FBP algorithm with a spline tri-kernel of degrees $n_i = 0$ is shown in Fig.3b (PSNR=29.16). The best results (PSNR=29.34) were obtained with linear spline tri-kernels ($n_i = 1$) (Fig.3c);

7. CONCLUSION

Spline convolution kernels were first introduced in the context of image resizing, but up to now no closed-form was available beyond the degree $n = 1$. In this paper, we have derived explicit formulas valid for all B-spline degrees n and widths h. We have used these results to provide analytical solutions for least-squares image resizing, computation of image projections (Radon transform), and tomographic reconstruction by filtered back-projection using splines of any degree n. These are all examples of what we call continuous-space image processing.

The presented least-squares algorithms outperform the standard ones which use simple interpolation and are thus sub-optimal. Moreover, we have shown that the results can be improved even further by using higher-order splines. Of course, the price to pay is an increase in the amount of computation.

8. REFERENCES

[1] J.-P. Guédon and Y. Bizais, "Bandlimited and Haar filtered back-projection reconstruction", *IEEE Trans. Medical Imaging*, vol.12, no.3, pp.430-440, 1994.

[2] H.S. Hou and H.C. Andrews, "Cubic splines for image interpolation and digital filtering", *IEEE Trans. Acoust., Speech, Signal Proc.*, vol.26, pp.508-517, 1978.

[3] M. Lautsch, "A spline inversion formula for the Radon transform", *SIAM J. Num. Ana.,* vol.26, no.2, pp.456-467, 1989.

[4] C. Lee, M. Eden and M. Unser, "High-quality image resizing using oblique projection operators", *IEEE Trans. Image Proc.*, vol.7, no.5, pp.679-692, 1998.

[5] D. Ludwig, "The Radon transform on Euclidean space", *Comm. Pure and Applied Math.,* vol.19, pp.49-81, 1966.

[6] L.A. Shepp and B. F. Logan, "The Fourier reconstruction of a head section", *IEEE Trans. Nucl. Sci.,* vol.21, pp.21-43, 1974.

[7] M. Unser, A. Aldroubi and M. Eden, "B-spline signal processing: Part I—Theory and Part II—Efficient design", *IEEE Trans. Signal Proc.*, vol.41, no.2, pp.821-848, 1993.

[8] M. Unser, A. Aldroubi and M. Eden, "Enlargement or reduction of digital images with minimum loss of information", *IEEE Trans. Image Proc.*, vol.4, no.3, pp.247-258, 1995.

[9] M. Unser, "A general Hilbert space framework for the discretization of continuous signal processing operators", *Proc. SPIE*, vol.2569, part III, pp.51-61, 1998.