

*To describe a geometrical curve which shall pass through any given points.... Although the problem may seem to be intractable at first sight, it is quite the contrary. Perhaps indeed it is one of the prettiest problems I can ever hope to solve.*

— Isaac Newton, in a letter to Henry Oldenburg  
(24 October 1676)

# Chapter 6

## Quantitative Evaluation of Convolution-Based Methods for Medical Image Interpolation

**Abstract** — Interpolation is required in a variety of medical image processing applications. Although many interpolation techniques are known from the literature, evaluations of these techniques for the specific task of applying geometrical transformations to medical images are still lacking. In this chapter we present such an evaluation. We consider convolution-based interpolation methods and rigid transformations. A large number of sinc-approximating kernels are evaluated, including piecewise polynomial kernels and windowed sinc kernels, with spatial supports ranging from 2 to 10 grid intervals. In the evaluation we use images from a wide variety of medical image modalities. The results of the evaluation show that for all modalities, spline interpolation constitutes the best trade-off between accuracy and computational cost, and therefore is to be preferred over all other methods.

### 6.1 Introduction

Interpolation of sampled data is required in many digital image processing operations, such as subpixel translation, rotation, elastic deformation or warping, magnification, or minification, which need to be carried out for the purpose of image registration or volume visualization. In most applications, it is of paramount importance to limit as much as possible the grey-value errors introduced by interpolation. For example, in multimodality registration of computed tomography (CT), magnetic resonance imaging (MRI), or positron emission tomography (PET) data, it has been observed that interpolation errors influence the value of the optimization cost function, which may lead to registration errors [299]. Similar problems had been reported earlier in monomodality registration of MRI images [134]. It has been pointed out that in digital subtraction angiography (DSA), improved registration and resampling methods result in improved image quality [286], which allows for reduction of contrast material or X-ray dose. It has also been pointed out that in functional

magnetic resonance imaging (fMRI), interpolation errors induced by registration operations may influence the interpretation of longitudinal studies [278].

The common denominator in all of these applications is geometrical transformation of medical image data. Although many interpolation techniques have been put forward over the years, evaluations of these techniques for this particular task are still lacking. One of the earlier studies in this area was reported by Parker *et al.* [286], who compared the performance of nearest-neighbor and linear interpolation, as well as cubic convolution, by analyzing the effects of these techniques on the rotation of images. No quantitative measures were computed, however, and the only medical image included in that study was a single coronary angiogram. A quantitative evaluation of the performance of convolution-based interpolation techniques in combination with specific fast image rotation algorithms was presented by Unser *et al.* [382]. Apart from the interpolation techniques analyzed by Parker *et al.* [286], their evaluation also included spline and sinc interpolation. However, no medical images were used. A more recent study was presented by Ostuni *et al.* [278], who compared the performance of linear, cubic spline, and truncated and Hann-windowed sinc interpolation for the geometrical transformation of fMRI images. However, that study did not include images from other modalities. More elaborate evaluation studies were recently published by Lehmann *et al.* [213] and Grevera & Udupa [128]. The former study concerned geometrical transformation of medical images. However, only MRI and dental X-ray images were considered. The latter study involved a number of both convolution- and shape-based interpolation methods for the purpose of slice doubling in MRI and CT images. However, the effects of these techniques on the geometrical transformation of images from these and other medical image modalities were not investigated. The same holds for the studies of Schreiner *et al.* [339] and Chuang [54], in which interpolation techniques were compared for the purpose of generating maximum intensity projections (MIPs) from MRA data, and surface rendering, respectively. Finally, we mention our recent study [249], in which we analyzed the effects of several piecewise polynomial interpolation kernels on the geometrical transformation of images. However, no medical images were included in that study.

The purpose of this chapter is to present the results of an elaborate evaluation, in which we quantitatively studied the performance of a large number of interpolation methods when using them to apply geometrical transformations to images from a wide variety of medical imaging modalities. The results of this evaluation are important for the tasks of *e.g.* mono- and multimodality medical image registration: the use of optimal interpolation methods minimizes the loss of information caused by the transformation of images. In order to limit the size of this work, we considered only rigid transformations, in particular rotations and translations. We also restricted ourselves to convolution-based interpolation techniques. Although recent developments have resulted in new, fundamentally different interpolation techniques, such as shape- or morphology-based methods [127, 130, 144–146, 308], or Fourier-based methods, such as voxel-shift interpolation or, equivalently, zero-filled interpolation [74, 77, 162, 187], the vast majority of interpolation techniques used in medical image registration are convolution-based techniques. The reason for this is probably that these techniques are less complex than shape-based techniques. That is to say, they are easier to implement and require no or considerably less preprocessing time. Furthermore, compared

to Fourier-based techniques, they are better suited for local interpolation problems, such as those occurring in registration based on control points.

This chapter is organized as follows. First, in Section 6.2, we provide the necessary theoretical background information and conclude that convolution-based interpolation requires the use of what we call *sinc-approximating kernels*. Next, the sinc-approximating kernels incorporated in this study are presented and discussed briefly in Section 6.3. The evaluation strategy and the results are described in Section 6.4. Both are discussed in detail in Section 6.5. Finally, concluding remarks are made in Section 6.6.

## 6.2 Convolution-Based Interpolation

In general, a digital  $N$ -dimensional ( $ND$ ) real-valued image  $I_s$  is the result of a number of local measurements (observations) of a physical source field, or a number of evaluations of a mathematical function describing some synthetic object or scene. Continuous measurements or evaluations would have resulted in an image  $I(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_N) \in \mathbb{R}^N$ . In digital image processing, the only available information about  $I$  is the set of samples  $I_s(\mathbf{p})$ ,  $\mathbf{p} = (p_1, \dots, p_N) \in P$ , where  $P$  is usually a Cartesian grid  $\mathbb{Z}(\Delta_1) \times \dots \times \mathbb{Z}(\Delta_N)$ , with  $\Delta_i$ ,  $i = 1, \dots, N$ , denoting the inter-sample distances in each dimension. However, it is frequently desired to know the image value  $I$  at a position  $\mathbf{x} \notin P$  in a certain region of interest  $X \subset \mathbb{R}^N$ , while resampling of the original field or function is not possible since it is no longer available. Under these circumstances, it is required to reconstruct the image  $I(\mathbf{x})$ ,  $\mathbf{x} \in X$ , from its samples  $I_s(\mathbf{p})$  in that region by means of interpolation.

From the Whittaker-Shannon sampling theorem [171, 274, 348, 410, 412] it follows that exact reconstruction of a continuous  $ND$  image,  $I$ , is possible in those cases where the sampling frequencies  $F_{s_i}$  satisfied the Nyquist criterion:<sup>1</sup>  $F_{s_i} > 2F_{m_i}$ ,  $\forall i = 1, 2, \dots, N$ , where  $F_{m_i}$  is the highest frequency in the  $i$ th dimension of the original image  $I$ . To this end, the sampled image  $I_s$  must be convolved with a filter having the following Fourier spectrum:

$$\tilde{H}(\mathbf{f}) = \begin{cases} \kappa & \text{if } |f_i| \leq \frac{1}{2}F_{s_i}, \forall i = 1, \dots, N, \\ 0 & \text{otherwise,} \end{cases} \quad (6.1)$$

<sup>1</sup>It must be pointed out that in those cases where, apart from the original signal  $I(\mathbf{x})$ , also the derivatives  $\partial^k I(\mathbf{x})/(\partial x_i)^k$ ,  $\forall k = 1, 2, \dots, K$  and  $\forall i = 1, 2, \dots, N$  are sampled, it is sufficient for the sampling frequencies  $F_{s_i}$  to satisfy  $F_{s_i} > 2F_{m_i}/(K+1)$ . (In the limiting case  $K \rightarrow \infty$ , the requirement becomes  $F_{s_i} > 0$ , which implies that in order to be able to reconstruct the original signal it is sufficient to sample the function and its derivatives at a single position only, since in that case the samples provide a complete Taylor series representation.) This was first remarked by Shannon [348] and has later been stated and proved by Fogel [108], without reference to Shannon's remark. The form of the sampling theorem involving the original signal and its first derivative was subsequently presented by Jagerman & Fogel [164]. As an explicit response to Shannon's remark, the generalized sampling theorem was presented by Linden & Abramson [218, 219]. We note that it also follows directly from the generalized sampling expansion proposed by Papoulis [283]. Although this version of the theorem may be of interest in specific application areas, it is not of practical importance in medical imaging since, in most cases, samples of the derivatives of the original image are not available. Therefore, this issue will not be considered further here.

where  $\mathbf{f} = (f_1, \dots, f_N) \in \mathbb{R}^N$  denotes  $ND$  frequency, and  $\kappa = \prod_{i=1}^N F_{s_i}^{-1}$ . By using this  $ND$  box-filter, the continuous image  $I$  and its sampled version  $I_s$  are related by:  $\tilde{I}(\mathbf{f}) = \tilde{I}_s(\mathbf{f})\tilde{H}(\mathbf{f})$ , where  $\tilde{I}$  and  $\tilde{I}_s$  are the Fourier transforms of  $I$  and  $I_s$ , respectively. It can easily be verified that, since the  $ND$  box-filter can be written as a product of  $N$  one-dimensional box-filters,  $ND$  image reconstruction in the spatial domain can be carried out by  $N$  successive 1D convolutions:

$$I(\mathbf{x}) = (\cdots ((I_s(\mathbf{p}) * h(x_1)) * h(x_2)) * \cdots) * h(x_N), \quad (6.2)$$

where the convolution kernel  $h : \mathbb{R} \rightarrow \mathbb{R}$  is the inverse Fourier transform of the 1D box-filter. By assuming unit distance between the grid points,<sup>2</sup> this kernel can be derived to be the well-known sinc function:

$$h(x) = \text{sinc}(x) \triangleq \frac{\sin(\pi x)}{\pi x}. \quad (6.3)$$

Although the sinc function is the theoretically optimal kernel for convolution-based interpolation of originally band-limited images, it is not the ideal kernel in most practical situations. First of all, since the objects that are being imaged have finite spatial extent, the resulting images cannot be strictly band limited. This implies that, in practice, it is not possible for the sampling frequencies to satisfy the Nyquist criterion. Consequently, it is impossible to retrieve the original images exactly from the resulting samples by means of sinc interpolation. Another problem of sinc interpolation is the fact that, since the sinc function has infinite support, Eq. (6.2) cannot be computed in practice, except in the case of periodic images [202, 204, 331], which are not likely to occur in medical imaging. Furthermore, interpolation by means of a band-limiting convolution kernel may result in Gibbs phenomena, which are very disturbing in images.

For convolution-based interpolation, the only solution to these problems is to choose an alternative convolution kernel. However, in order for any convolution kernel to actually interpolate the given samples, it *must* satisfy the following requirements, which are ultimately satisfied by the sinc function:

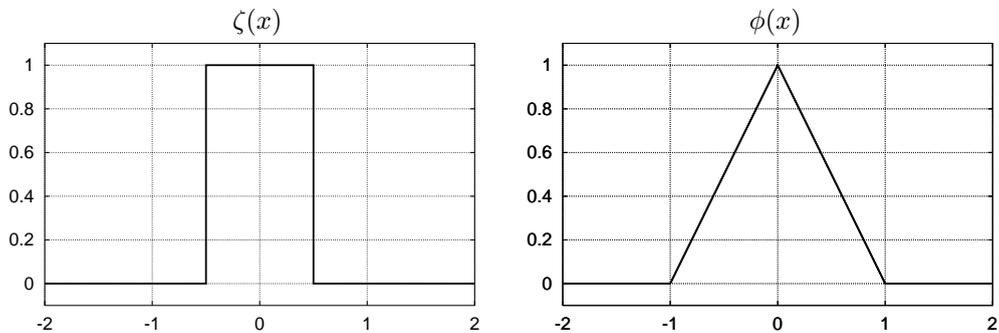
$$h(x) = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{if } x \in \mathbb{Z}, x \neq 0. \end{cases} \quad (6.4)$$

In this chapter, we will refer to kernels satisfying Eq. (6.4) as *sinc-approximating kernels*, even though there exist infinitely many kernels that satisfy these requirements but do not necessarily “resemble” the sinc function.

### 6.3 Sinc-Approximating Kernels

In this section, we introduce the sinc-approximating kernels incorporated in the evaluation. These include the nearest-neighbor and linear interpolation kernel, as well

<sup>2</sup>This is not a restriction; any function  $I(x_1, \dots, x_N)$ ,  $x_i \in \mathbb{Z}(\Delta_i)$ , can be reparameterized so as to end up with a function  $I(x_1, \dots, x_N)$ ,  $x_i \in \mathbb{Z}$ . For example, spatial or temporal quantities may be expressed in pixels instead of millimeters or seconds.



**Figure 6.1.** **Left:** the nearest-neighbor interpolation kernel. **Right:** the linear interpolation kernel. See Section 6.3.1 for the definitions of these kernels.

as the Lagrange, generalized convolution, cardinal spline, and windowed sinc kernels. Since the main purpose of this chapter is to present the results of an empirical evaluation of the performance of interpolation kernels for the specific task of geometrically transforming medical images, we do not discuss the application-independent spatial and spectral properties of the kernels in great detail in this section. For more in-depth discussions of these particular properties, we refer to numerous other sources [20, 227, 228, 235, 245, 249, 257, 285, 286, 332, 420].

### 6.3.1 Nearest-Neighbor and Linear Interpolation Kernel

The simplest and computationally cheapest approach to obtain a sinc-approximating kernel that complies with the definition of Eq. (6.4) is to use zeroth-degree or first-degree polynomials, resulting in the nearest-neighbor and linear interpolation kernel, respectively defined as

$$\zeta(x) = \begin{cases} 1 & \text{if } -\frac{1}{2} \leq x < \frac{1}{2}, \\ 0 & \text{otherwise,} \end{cases} \quad (6.5)$$

and

$$\phi(x) = \begin{cases} 1 - |x| & \text{if } 0 \leq |x| < 1, \\ 0 & \text{if } 1 \leq |x|. \end{cases} \quad (6.6)$$

Plots of these kernels are provided in Fig. 6.1. As can be concluded from recent literature, the linear interpolation kernel is still the most frequently used kernel in a wide variety of applications [128].

### 6.3.2 Lagrange Interpolation Kernels

In order to obtain higher order interpolants, one possibility is to use classical polynomial interpolation formulae. In that case, an interpolant is expressed either in terms

of (divided) differences, as proposed by Gregory [126] and Newton [266, 267] in the late 17th century, or directly in terms of the sample values, as in the interpolation formula originally due to Waring [399] and Euler [83], but nowadays usually attributed to Lagrange [201]. In principle, these schemes are all equivalent.<sup>3</sup>

When using Lagrange central interpolation, an  $n$ th-degree interpolant is obtained by evaluating the following sum:

$$\sum_{k=k_{\min}}^{k_{\max}} I_s(p_k) L_k^n(x), \tag{6.7}$$

where  $n \geq 1$ ,  $k_{\min} = -\lceil n/2 \rceil$ ,  $k_{\max} = \lceil n/2 \rceil$ ,  $p_k = (p_0 + k)$ ,  $p_0 = \lfloor x \rfloor$ ,  $x \in \mathbb{R}$ , and  $L_k^n$  are the so called Lagrange coefficients, defined by

$$L_k^n(x) \triangleq \prod_{\substack{i=k_{\min} \\ i \neq k}}^{k_{\max}} \frac{(x - p_i)}{(p_k - p_i)}. \tag{6.8}$$

As shown by Schafer & Rabiner [330], the Lagrange central interpolation formula (6.7) can be rewritten in the form of a convolution:

$$\sum_{k=-\infty}^{+\infty} I_s(p_k) \lambda^n(x - p_k), \tag{6.9}$$

where  $\lambda^n$  is the  $n$ th-degree Lagrange central interpolation kernel. As can be observed from (6.7) and (6.9), the explicit form of this kernel is obtained by evaluating the following set of equations:

$$\left\{ \begin{array}{l} \lambda^n(\xi - k_{\max}) = L_{k_{\max}}^n(\xi), \\ \vdots \\ \lambda^n(\xi - 1) = L_1^n(\xi), \\ \lambda^n(\xi) = L_0^n(\xi), \\ \lambda^n(\xi + 1) = L_{-1}^n(\xi), \\ \vdots \\ \lambda^n(\xi - k_{\min}) = L_{k_{\min}}^n(\xi), \end{array} \right. \tag{6.10}$$

where

$$\xi \in \begin{cases} [0, 1] & \text{if } n \text{ odd,} \\ [-\frac{1}{2}, \frac{1}{2}] & \text{if } n \text{ even,} \end{cases} \tag{6.11}$$

<sup>3</sup>This can easily be deduced from the fact that for any two polynomial interpolants of degree  $n$ , e.g.  $\hat{I}_1(x)$  and  $\hat{I}_2(x)$ ,  $x \in \mathbb{R}$ , the difference  $\hat{I}_1(x) - \hat{I}_2(x)$  is a polynomial of at most degree  $n$ , while it has  $n + 1$  zeroes, viz., the sample points. According to the fundamental theorem of algebra this can only be true if  $\hat{I}_1(x) - \hat{I}_2(x) = 0$ ,  $\forall x \in \mathbb{R}$ . For more details regarding classical polynomial interpolation we refer to Whittaker & Robinson [411], Hildebrand [148], or Jeffreys & Jeffreys [170].

and by explicitly defining  $\lambda^n(x) = 0$  for  $|x| > (n+1)/2$ . It is important to respect the requirement expressed in (6.11). If, for any  $n$  even, the set of equations given in (6.10) is evaluated in the interval  $[0, 1]$ , the resulting kernel will not be symmetric. This has led some authors to the (incorrect) conclusion that, in general, even-degree Lagrange kernels are not symmetric and lead to phase distortions [330, 420].

The kernels constituted by these equations were studied *e.g.* by Schaum [332]. Notice that the Lagrange kernel corresponding to  $n = 1$  is equal to the linear interpolation kernel given in Eq. (6.6). In the evaluation presented in this chapter, we also included the quadratic ( $n = 2$ ), cubic ( $n = 3$ ), quartic ( $n = 4$ ), quintic ( $n = 5$ ), sextic ( $n = 6$ ), septic ( $n = 7$ ), octic ( $n = 8$ ), and nonic ( $n = 9$ ) Lagrange central interpolation kernel. See Fig. 6.2 for plots of some of these kernels. As can be appreciated from these plots,  $\lambda^n$  more and more resembles the sinc function as  $n$  increases. In fact, it can be shown that for  $n \rightarrow \infty$ , the Lagrange central interpolation kernel converges to the sinc function [166, 245].

### 6.3.3 Generalized Convolution Kernels

The symmetrical piecewise polynomial kernels described in the previous subsections all result in interpolants which are not continuously differentiable. In particular applications, it may be desirable to use smoother interpolation kernels, which allow for the computation of higher order derivatives of the interpolant. In this subsection we describe a class of smooth piecewise polynomial kernels, which contains important special cases that are well known in the literature.

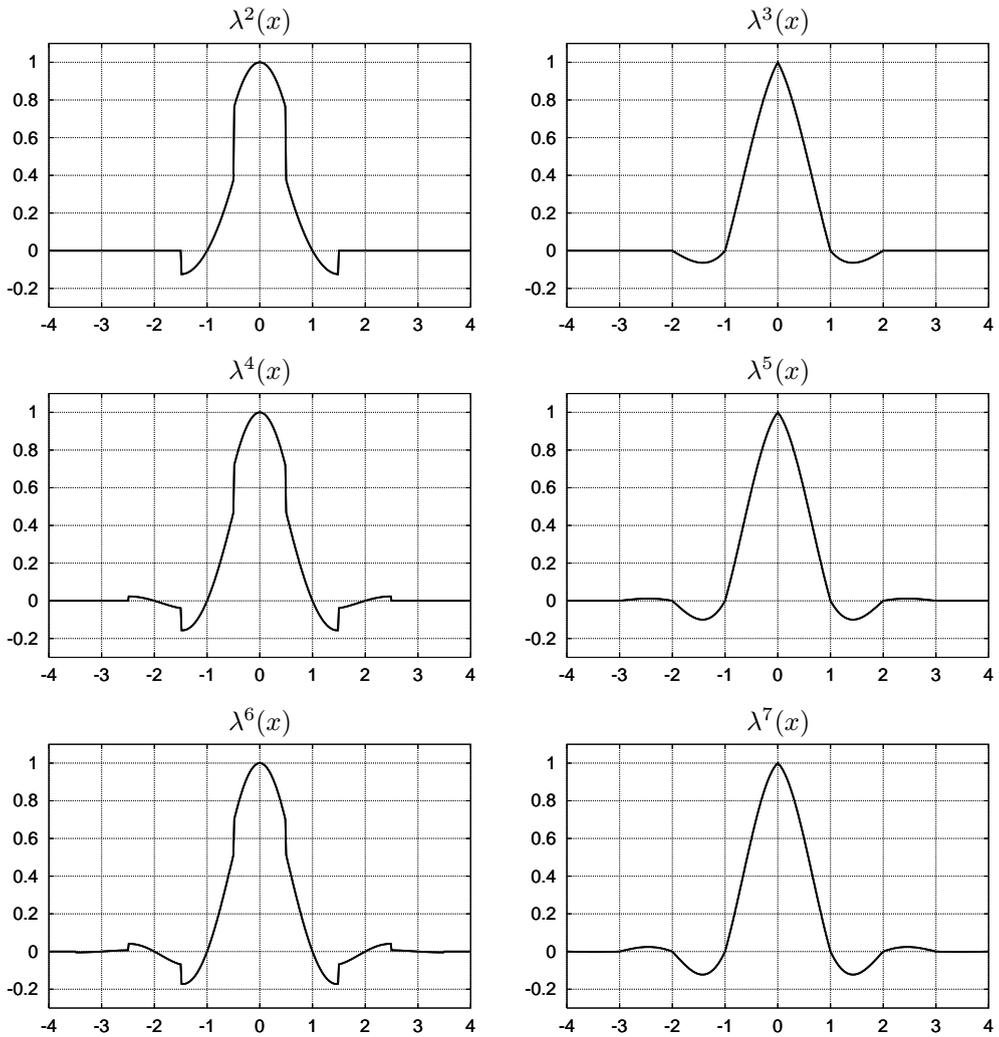
In general, piecewise  $n$ th-degree polynomial kernels can be written in the form:

$$\psi^n(x) = \begin{cases} \sum_{i=0}^n a_{ij}|x|^i & \text{if } j - \xi \leq |x| < j + 1 - \xi, \\ 0 & \text{if } m \leq |x|, \end{cases} \quad (6.12)$$

where  $n \geq 1$ ,  $\xi = 1/2$  for  $n$  even and  $\xi = 0$  for  $n$  odd,  $j = 0, 1, \dots, m + \xi - 1$ , and the parameter  $m$  determines the spatial support of the kernel. In the evaluation presented in this chapter, we restricted ourselves to the class of kernels for which  $n$  and  $m$  are related by  $n = 2m - 1$ . This is a rather broad class, which includes the linear interpolation kernel of Eq. (6.6) and all of the Lagrange central interpolation kernels described in the previous subsection. It also includes the quadratic piecewise polynomial kernel due to Dodgson [70]:

$$\psi^2(x) = \begin{cases} 1 - 2|x|^2 & \text{if } 0 \leq |x| < \frac{1}{2}, \\ \frac{3}{2} - \frac{5}{2}|x| + |x|^2 & \text{if } \frac{1}{2} \leq |x| < \frac{3}{2}, \\ 0 & \text{if } \frac{3}{2} \leq |x|. \end{cases} \quad (6.13)$$

In the remainder of this subsection we concentrate on a family of odd-degree convolution kernels which are at least  $C^1$ . The  $(n+1)m$  coefficients  $a_{ij}$  of the odd-degree polynomial pieces can be solved by imposing constraints on the shape of the kernel. The first and most important constraint was already given in Eq. (6.4). In



**Figure 6.2.** From top left to bottom right: the quadratic, cubic, quartic, quintic, sextic, and septic Lagrange central interpolation kernel. See Section 6.3.2 for the precise definitions of these kernels.

order for the resulting interpolant to have continuous derivatives, it is also required that  $\psi^{(l)}(x)$  is continuous at  $|x| = 0, 1, 2, \dots, m$ , where the superscript ( $l$ ) denotes the  $l$ th-order derivative. It can be shown [249] that, given any odd degree  $n \geq 3$ , the maximum allowable value for  $l$  that will not result in an over-constrained problem is  $n - 2$ , in which case the total number of equations to be solved is  $(n + 1)m - 1$ . This implies that the kernels can be expressed in terms of a free parameter, which we denote by  $\alpha$ . In order to obtain a unique value for  $\alpha$ , one additional constraint needs to be imposed. In this chapter, we used the following constraints: (i) the

Kernel	$\alpha_{\zeta}$	$\alpha_{\sim}$	$\alpha_{\flat}$
$\psi^3$	-1	$-\frac{3}{4}$	$-\frac{1}{2}$
$\psi^5$	$\frac{11}{96}$	$\frac{1}{13}$	$\frac{3}{64}$
$\psi^7$	$-\frac{1027}{452574}$	$-\frac{3133}{2275008}$	$-\frac{71}{83232}$
$\psi^9$	$\frac{34814699}{2509872453120}$	$\frac{17671607}{2324998440576}$	$\frac{3829}{788235264}$

**Table 6.1.** The values of the free parameter  $\alpha$  for the cubic, quintic, septic, and nonic convolution kernel described in Section 6.3.3, resulting from the slope constraint ( $\alpha_{\zeta}$ ), continuity constraint ( $\alpha_{\sim}$ ), and flatness constraint ( $\alpha_{\flat}$ ), respectively.

slope constraint [315], which implies that the slope of the kernel is constrained in such a way that it equals the slope of the sinc function at  $x = 1$ ; (ii) the continuity constraint [352], which implies that the kernel is constrained in such a way that its  $(n-1)$ th-order derivative is continuous at  $x = 1$ ; (iii) the flatness constraint [249, 285], which implies that the frequency spectrum  $\tilde{\Psi}(f)$  of the kernel is required to be flat at  $f = 0$ . It can be shown that the latter constraint yields the mathematically most precise interpolant, in the sense that the Taylor series expansion agrees in as many terms as possible with the original signal [180, 243].

A well-known member of this family of kernels is the cubic convolution kernel [180, 285, 315, 352]. In fact, the aforementioned constraints were adopted from the literature on cubic convolution. The cubic convolution kernel as a function of the free parameter  $\alpha$  is given by

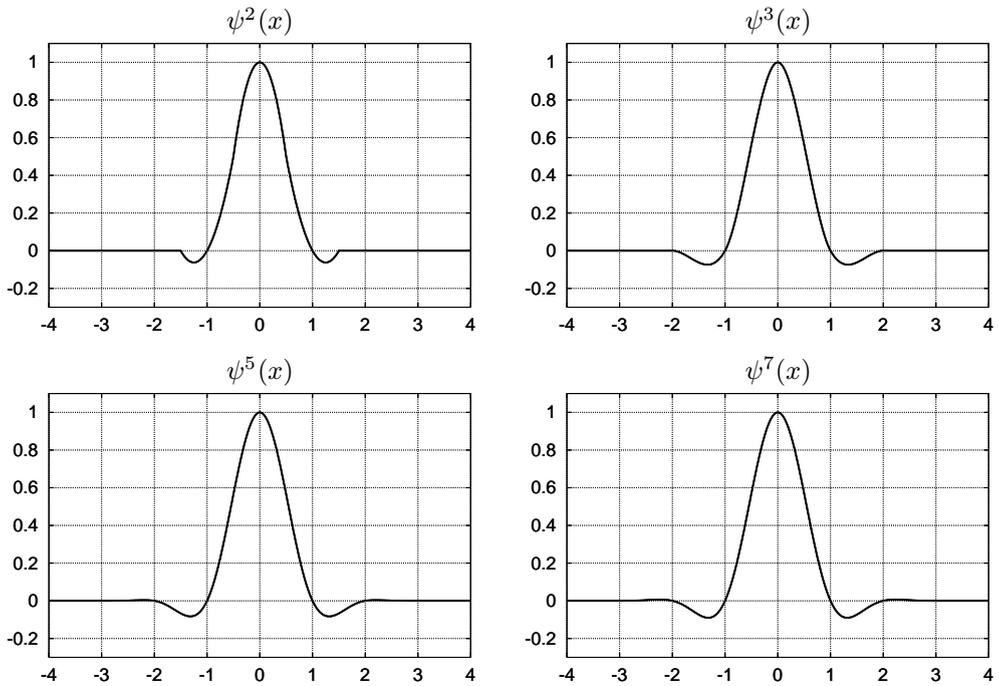
$$\psi^3(x) = \begin{cases} 1 - (\alpha + 3)|x|^2 + (\alpha + 2)|x|^3 & \text{if } 0 \leq |x| < 1, \\ -4\alpha + 8\alpha|x| - 5\alpha|x|^2 + \alpha|x|^3 & \text{if } 1 \leq |x| < 2, \\ 0 & \text{if } 2 \leq |x|. \end{cases} \quad (6.14)$$

In the literature on visualization and computer graphics, the cubic convolution kernel resulting from the flatness constraint is also known as the Catmull-Rom spline [48] or the modified cubic spline [143], and is sometimes erroneously referred to as the cardinal cubic spline [255, 257].

By analogy with cubic convolution, the kernels from this family are referred to as generalized convolution kernels in this chapter. Apart from the quadratic ( $n = 2$ ) and cubic ( $n = 3$ ) convolution kernel, we also included the quintic ( $n = 5$ ), septic ( $n = 7$ ), and nonic ( $n = 9$ ) convolution kernel in the evaluation. See Fig. 6.3 for plots of some of these kernels. For precise definitions of the higher order kernels we refer to an earlier paper [249]. The corresponding values of the free parameter  $\alpha$  resulting from the aforementioned constraints, are presented in Table 6.1.

### 6.3.4 Cardinal Spline Kernels

An alternative approach to piecewise polynomial interpolation is spline interpolation, originally proposed by Schoenberg [335, 336], which involves the use of so called B-



**Figure 6.3.** From top left to bottom right: the quadratic convolution kernel, and the cubic, quintic, and septic convolution kernel resulting from the flatness constraint. See Section 6.3.3 for details.

splines, recursively obtained by auto-convolution of a rectangular pulse (equal to the nearest-neighbor kernel given in Eq. (6.5)), that is,

$$\beta^n(x) = \beta^{n-1}(x) * \beta^0(x), \quad \text{with} \quad \beta^0(x) = \zeta(x). \quad (6.15)$$

The explicit form of a B-spline of degree  $n$  can be obtained by analyzing the Fourier transform of Eq. (6.15), see *e.g.* Unser [378], and is given by

$$\beta^n(x) = \frac{1}{n!} \sum_{i=0}^{n+1} \binom{n+1}{i} (-1)^i \left(x - i + \frac{n+1}{2}\right)_+^n, \quad (6.16)$$

with

$$(x)_+^n \triangleq \begin{cases} x^n & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases} \quad (6.17)$$

Since  $\beta^n$  does not satisfy the requirements expressed in Eq. (6.4) for all  $n \geq 2$ , interpolation by means of B-splines requires preprocessing of the raw image data in those cases. This can be done either by matrix manipulations [159, 207], or by means

of recursive filtering techniques [379–381], the latter of which are easier to implement and are computationally much more efficient. When using the latter approach,  $n$ th-degree spline interpolation is carried out by evaluating the following expression:

$$\sum_{k=-\infty}^{+\infty} ((b^n)^{-1} * I_s)(k) \beta^n(x - k), \quad (6.18)$$

where  $(b^n)^{-1}$  denotes the recursive prefilter, also known as the direct B-spline filter (see Appendix 6.B for more details). Although it is never explicitly implemented this way, the double convolution in (6.18) can be rewritten so as to obtain the implicit interpolation kernel:

$$\eta^n(x) = \sum_{k=-\infty}^{+\infty} (b^n)^{-1}(k) \beta^n(x - k), \quad (6.19)$$

which is known as the cardinal spline of degree  $n$ . We note that the cardinal spline of degree one is equal to the linear interpolation kernel of Eq. (6.6). Examples of higher order B-splines and their corresponding cardinal splines are shown in Fig. 6.4. Notice that the cardinal splines satisfy the requirements expressed in Eq. (6.4). We also note that for  $n \rightarrow \infty$ , the cardinal spline converges to the sinc function [4, 337].

In the evaluation presented in this chapter, we included quadratic ( $n = 2$ ), cubic ( $n = 3$ ), quartic ( $n = 4$ ), quintic ( $n = 5$ ), sextic ( $n = 6$ ), septic ( $n = 7$ ), octic ( $n = 8$ ), and nonic ( $n = 9$ ) spline interpolation, implemented by using Eq. (6.18). Notice that the corresponding B-spline kernels are piecewise  $n$ th-degree polynomial kernels which are non-zero only in the interval  $(-m, m)$ , where  $n$  and  $m$  are related by  $n = 2m - 1$ , similar to the Lagrange and generalized convolution kernels.

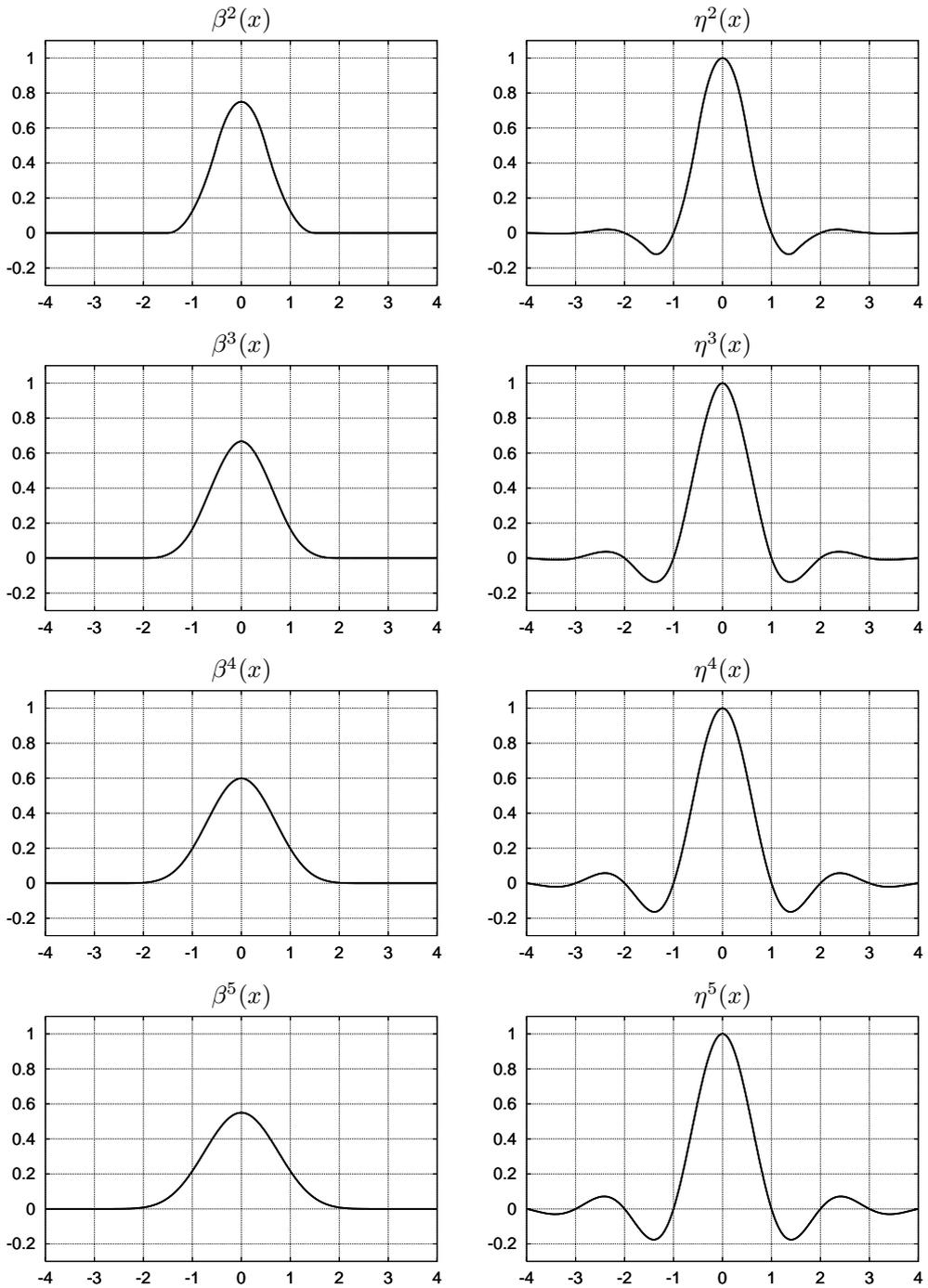
### 6.3.5 Windowed Sinc Kernels

A fundamentally different approach to obtain a sinc-approximating kernel is to multiply the sinc function, defined in Eq. (6.3), with a window function of limited spatial support:

$$h(x) = \omega(x) \text{sinc}(x), \quad \text{with} \quad \omega(x) = \begin{cases} w(x) & \text{if } 0 \leq |x| < m, \\ 0 & \text{if } m \leq |x|, \end{cases} \quad (6.20)$$

where  $\omega : \mathbb{R} \rightarrow \mathbb{R}$  is the window function, and  $w : \mathbb{R} \rightarrow \mathbb{R}$  determines the shape of the window in the interval  $(-m, m)$ , with  $m \in \mathbb{N}$ ,  $m \neq 0$ .

Window functions are well known in digital filter design [9, 135, 200, 276], where they are applied in order to reduce the Gibbs phenomena which appear when reducing the band-width of a signal, and also in spectroscopy and spectrometry [138], where they are known as apodization functions and are used to window spatial or temporal data prior to spectral analysis in order to reduce spectral leakage and loss of resolution. In the quantitative evaluation described in this chapter we used the following windows: Bartlett [14], Blackman [19], Blackman-Harris [138] (both three- and four-term), Bohman [23], Cosine [138], Gaussian [11, 175], Hamming [135], Hann [19, 135],



**Figure 6.4.** Left column: the quadratic, cubic, quartic, and quintic B-spline. Right column: the corresponding cardinal splines. See Section 6.3.4 for details.

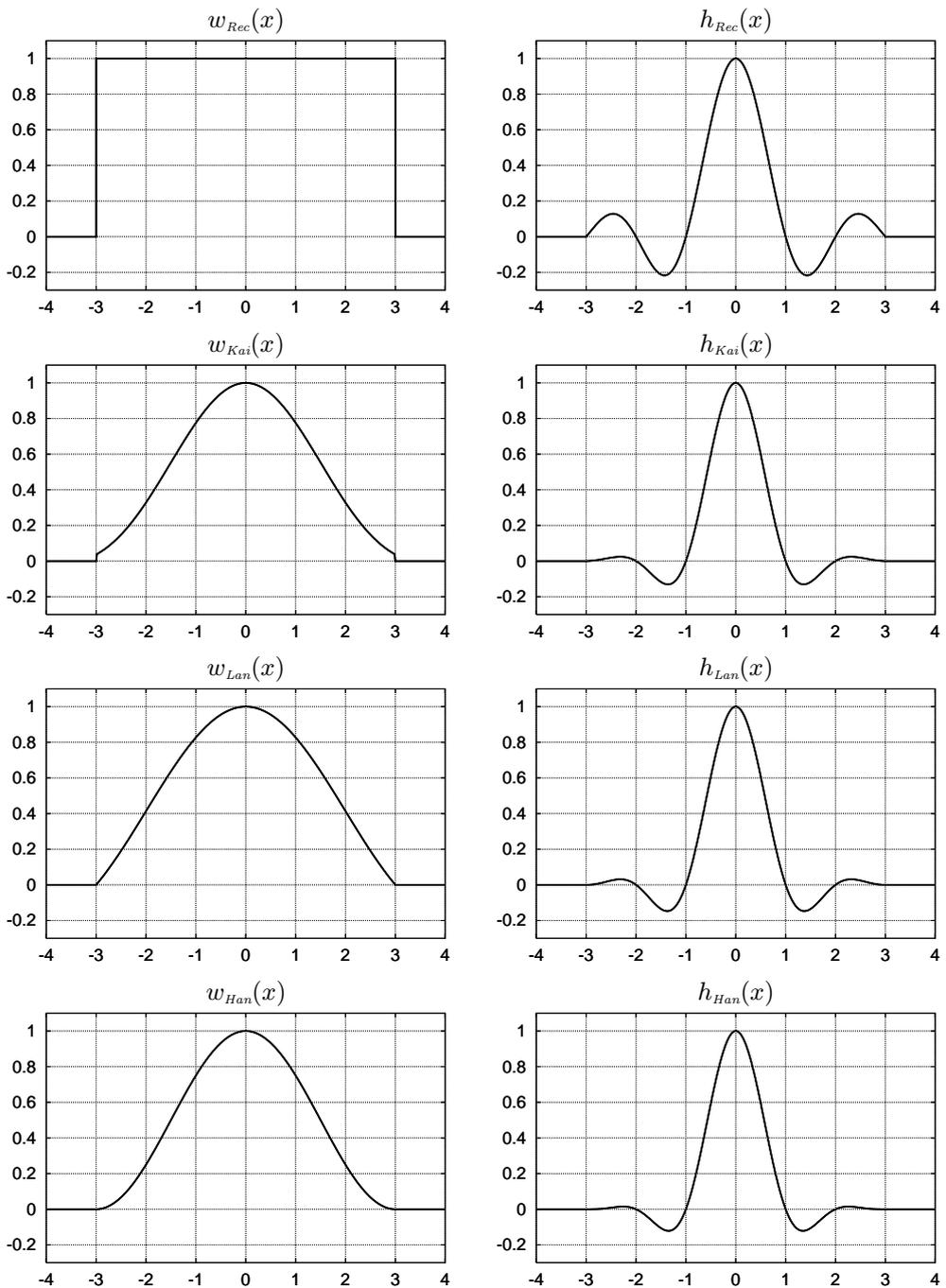
Window	Definition
Bartlett	$w_{Bar} \triangleq 1 - \frac{ x }{m}$
Blackman	$w_{Bla} \triangleq 0.42 + 0.50 \cos\left(\frac{\pi x}{m}\right) + 0.08 \cos\left(\frac{2\pi x}{m}\right)$
Blackman-Harris (3-term)	$w_{BH3} \triangleq 0.42323 + 0.49755 \cos\left(\frac{\pi x}{m}\right) + 0.07922 \cos\left(\frac{2\pi x}{m}\right)$
Blackman-Harris (4-term)	$w_{BH4} \triangleq 0.35875 + 0.48829 \cos\left(\frac{\pi x}{m}\right) +$ $0.14128 \cos\left(\frac{2\pi x}{m}\right) + 0.01168 \cos\left(\frac{3\pi x}{m}\right)$
Bohman	$w_{Boh} \triangleq \left(1 - \frac{ x }{m}\right) \cos\left(\frac{\pi x}{m}\right) + \frac{1}{\pi} \sin\left(\frac{\pi x }{m}\right)$
Cosine	$w_{Cos} \triangleq \cos\left(\frac{\pi x}{2m}\right)$
Gaussian	$w_{Gau} \triangleq \exp\left(-\frac{1}{2} \left(\alpha \frac{x}{m}\right)^2\right)$
Hamming	$w_{Ham} \triangleq 0.54 + 0.46 \cos\left(\frac{\pi x}{m}\right)$
Hann	$w_{Han} \triangleq 0.5 + 0.5 \cos\left(\frac{\pi x}{m}\right)$
Kaiser	$w_{Kai} \triangleq \frac{I_0(\beta)}{I_0(\alpha)}, \beta = \alpha \sqrt{1 - \left(\frac{x}{m}\right)^2}$
Lanczos	$w_{Lan} \triangleq \text{sinc}\left(\frac{\pi x}{m}\right)$
Rectangular	$w_{Rec} \triangleq 1$
Welch	$w_{Wcl} \triangleq 1 - \frac{x^2}{m^2}$

**Table 6.2.** Definitions of window functions. In the definition of the Kaiser window,  $\alpha \in \mathbb{R}^+$  is a free parameter, for which values of 5.0, 6.0, 7.0, and 8.0 were used in the evaluation.  $I_0$  is the zeroth-order modified Bessel function of the first kind, which can accurately be approximated by using its series expansion [138, 420]. For the free parameter  $\alpha \in \mathbb{R}^+$  in the definition of the Gaussian window, values of 2.5, 3.0, 3.5, and 4.0 were used in the evaluation.

Kaiser [174], Lanczos [202], Rectangular [420], and Welch [407]. Definitions of these windows are given in Table 6.2. Plots of some windows and their corresponding sinc-approximating kernel are shown in Fig. 6.5. For more elaborate discussions of the spectral properties of window functions we refer to Harris [138] or Wolberg [420].

## 6.4 Quantitative Evaluation

The sinc-approximating kernels described in the previous section were quantitatively evaluated by using them to apply several geometrical transformations to a variety of medical test images, and by computing figures of merit (FOMs) based on the grey-value differences between the transformed images and their corresponding reference images. The computational cost of these kernels was also determined. In this section we present the evaluation strategy and the results.



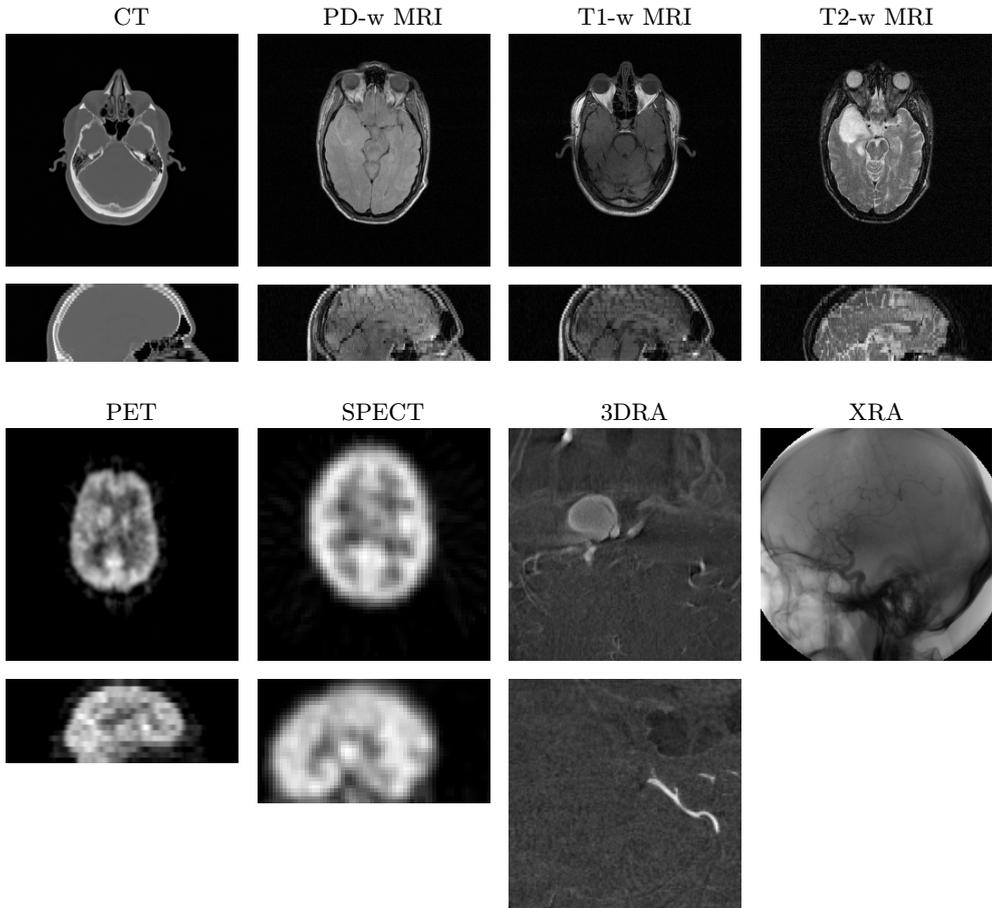
**Figure 6.5.** Examples of windows and windowed sinc kernels. **Left column:** the Rectangular, Kaiser ( $\alpha = 5.0$ ), Lanczos, and Hann window for  $m = 3$ . **Right column:** the corresponding windowed sinc kernels. See Section 6.3.5 for details.

### 6.4.1 Evaluation Strategy

The medical images used in the evaluation were obtained from 3D brain datasets of different modalities, *viz.*, computed tomography (CT), proton-density weighted (PD-w) magnetic resonance imaging (MRI), T1-weighted (T1-w) MRI, T2-weighted (T2-w) MRI, positron emission tomography (PET), single photon emission computed tomography (SPECT), and 3D rotational angiography (3DRA). Images from 2D cerebral X-ray angiography (XRA) sequences were also included.<sup>4</sup> From every subset (eight in total), we selected five datasets. The five CT datasets were of size  $512 \times 512$  times 28, 28, 29, 30, and 33 voxels, respectively, all with a voxel size of  $0.65 \times 0.65 \times 4.0$  mm<sup>3</sup>. The PD-w, T1-w, and T2-w MRI datasets (15 in total) were all of size  $256 \times 256 \times 26$  voxels, with a voxel size of  $1.25 \times 1.25 \times 4.0$  mm<sup>3</sup>. The five PET datasets were of size  $128 \times 128 \times 15$  voxels, one with a voxel size of  $1.94 \times 1.94 \times 8.0$  mm<sup>3</sup>, and the others with a voxel size of  $2.59 \times 2.59 \times 8.0$  mm<sup>3</sup>. The five SPECT datasets were of size  $64 \times 64$  times 30, 34, 36, 38, and 40 voxels, respectively, all with a voxel size of  $3.91 \times 3.91 \times 3.91$  mm<sup>3</sup>. The five 3DRA datasets were all of size  $128 \times 128 \times 128$  voxels, with a voxel size of  $0.6 \times 0.6 \times 0.6$  mm<sup>3</sup>. Finally, the five 2D XRA images were all of size  $512 \times 512$  pixels and were arbitrarily selected from their corresponding image sequences. In order to be able to study the performance of the interpolation kernels in different slice directions, one transversal (axial) and one sagittal slice was selected from each of the 3D datasets. This resulted in a total of 75 different 2D test images. Some examples of test images are shown in Fig. 6.6.

The test images were subjected to several geometrical transformations. As explained in the introduction (Section 6.1), we considered only rotations and subpixel translations, as these are the most frequently required transformations in mono- or multimodality registration problems. In the rotation experiments, the 2D test images were successively rotated over  $0.7^\circ$ ,  $3.2^\circ$ ,  $6.5^\circ$ ,  $9.3^\circ$ ,  $12.1^\circ$ ,  $15.2^\circ$ ,  $18.4^\circ$ ,  $21.3^\circ$ ,  $23.7^\circ$ ,  $26.6^\circ$ ,  $29.8^\circ$ ,  $32.9^\circ$ ,  $35.7^\circ$ ,  $38.5^\circ$ ,  $41.8^\circ$ , and  $44.3^\circ$ , which adds up to a total of  $360^\circ$ . We note that for every test image, these 2D transformations were carried out in the plane of the image. The interpolation errors made in the transversal and sagittal slices are representative for those resulting from a rotation of the entire 3D dataset around its  $z$ - and  $x$ -axis, respectively. In the subpixel translation experiments, the test images were successively shifted over 0.01, 0.04, 0.07, 0.11, 0.15, 0.18, 0.21, 0.24, 0.26, 0.29, 0.32, 0.35, 0.39, 0.43, 0.46, and 0.49 pixels, which adds up to a total of 4.00 pixels. Similar to the rotations, the subpixel translations were carried out in the plane of the test image. Notice, however, that these are 1D transformations. For the transversal slices, the translations were carried out in the  $x$ -direction, while for the sagittal slices they were carried out in the direction corresponding to the through-plane direction

<sup>4</sup>The CT, MR, and PET datasets were obtained from patients undergoing neurosurgery at Vanderbilt University Medical Center and were originally used in the project “Evaluation of Retrospective Image Registration”, National Institutes of Health, Project Number: 1 R01 NS33926-01, Principal Investigator: Prof. Dr. J. M. Fitzpatrick, Vanderbilt University, Nashville, TN, USA (see West *et al.* [409] for more details concerning the acquisition of these datasets). The SPECT datasets were obtained from patients with suspected functional abnormalities and were acquired at the University Medical Center Utrecht, the Netherlands, under the authority of the Department of Child Psychiatry (see Stokking [357] for more details). The 3DRA and XRA datasets were obtained from patients with suspected cerebral aneurysms and were also acquired at the University Medical Center Utrecht, the Netherlands.



**Figure 6.6.** Examples of the medical test images used in the experiments described in Section 6.4. For every modality (except XRA, of course), one transversal slice (top image) and one sagittal slice (bottom image) is shown. Note that for display purposes, the images of the sagittal slices of the 3D datasets shown in this figure were scaled so as to correct for the voxel anisotropy.

in the original 3D dataset. The resulting interpolation errors are representative for those resulting from the application of subpixel translations to the entire 3D dataset in these same directions.

For every test image, the experiments were repeated for all interpolation kernels. Of the types described in Section 6.3, we used all kernels with a spatial support equal to or less than 10 grid intervals ( $m \leq 5$ ), which amounts to a total of 126 kernels (*viz.*, the nearest-neighbor and linear interpolation kernel, the quadratic convolution kernel, the cubic, quintic, septic, and nonic convolution kernel using three different values for the free parameter  $\alpha$ , the quadratic, cubic, quartic, quintic, sextic, septic,

otic, and nonic Lagrange and spline interpolation kernels, and finally 13 families of windowed sinc kernels (two of which have a free parameter for which we used the four different values indicated in Table 6.2), obtained by using five different settings for  $m$ ). We note that in order to avoid border problems, all test images were mirrored around the borders in each dimension.

For every combination of test image, experiment (rotation or translation), and interpolation kernel, the cumulative interpolation errors in the resulting processed image were determined. Since in these experiments the grid points of the processed images coincided with those of the corresponding original images, a gold standard was available: for the rotation experiments, the reference images were simply the original images, and for the translation experiments the reference images were obtained by translating the original image by four pixels (which requires no interpolation). The errors were summarized in two FOMs: the root-mean-square error (RMSE) and the largest absolute error (LAE). In order to avoid quantization errors to interfere with the results, all computations were carried out with double precision floating-point numbers (12 significant decimals).

Finally, the relative computational cost of all interpolation kernels was assessed by carrying out a timing experiment, in which a synthetic 3D test image of size  $128 \times 128 \times 128$  voxels was translated over  $(\gamma, \gamma, \gamma)$  voxels (where  $0 < \gamma < 1$  was an arbitrary, but fixed offset) by using non-separated 3D interpolation operations. For this timing experiment, special attention was paid to computationally optimal implementation of each individual interpolation approach.

### 6.4.2 Results

The computation of two FOMs for all processed images, resulting from the application of 126 different kernels to perform two types of transformations on 75 different test images, resulted in a total of 37800 error figures. In order to be able to present these results in a compact form, we first make some general observations.

In many applications, the important issue is not just accuracy, but the trade-off between accuracy and computational cost. In order to get an impression of the performance of all interpolation kernels in these terms, scatter plots were generated. To this end, the five error figures (either RMSEs or LAEs) resulting from every kernel in a given experiment (either rotation or translation) applied to the slices (either transversal or sagittal) of a given group of five datasets from any of the eight different modalities, were averaged. In order to correct for possible intrinsic differences in the dynamic range of grey values between the images within a group of five, the individual error figures were normalized with respect to the dynamic range of their corresponding image, before being averaged.

It was observed that regardless of image modality (CT, PD-w MRI, T1-w MRI, T2-w MRI, PET, SPECT, 3DRA, XRA), slice direction (transversal, sagittal), type of experiment (rotation, translation), or figure of merit (RMSE, LAE), spline interpolation constitutes the best trade-off between accuracy and computational cost. That is to say, none of the other approaches is more accurate and at the same time computationally cheaper. The scatter plots resulting from the different experiments carried out on CT data are shown in Fig. 6.7. For all modalities, the scatter plots showing

averaged, normalized RMSEs resulting from the rotation experiment carried out on transversal slices are presented in Fig. 6.8. As can easily be seen from these plots, the results of spline interpolation constitute a “lower boundary” in all cases.

It is important to note that in the timing experiments, kernel values were determined by exact computations during convolution. In practice, interpolation operations can be accelerated by using look-up tables of densely sampled, precomputed kernel values, as has been pointed out by *e.g.* Wolberg [420]. In principle, the additional errors due to the spatial quantization of a kernel can be reduced to any level simply by increasing the density of kernel samples. When using this approach, the relative computational cost of convolution kernels is determined solely by their spatial support. This implies that although spline interpolation seems the best approach in the case of exact computations (as suggested by the plots in Figs. 6.7 and 6.8), it does not necessarily have to be so when using look-up tables. Therefore it makes sense to mutually compare the accuracy of interpolation approaches of which the corresponding convolution kernels have equal spatial support.

To this end, the averaged, normalized error figures resulting from all kernels in the different experiments were analyzed separately for  $m = 1, 2, 3, 4$ , and 5. In order to limit the extent of this analysis, kernels with non-integer values of  $m$  (the even-degree piecewise polynomial kernels) were included in the group corresponding to the smallest larger integer value of  $m$  (*e.g.*, the quartic Lagrange central interpolation kernel, for which  $m = 2.5$ , was included in the group of kernels for which  $m = 3$ ). It was observed that regardless of image modality, slice direction, type of experiment, or FOM, given the value of  $m$ , the corresponding B-spline kernel performed either comparably to, or considerably better than all other kernels with the same spatial support. Therefore, we decided to present only the errors resulting from spline interpolation and to indicate whether or not these errors were statistically significantly smaller.

The averaged, normalized RMSEs and LAEs introduced by spline interpolation in the rotation and subpixel translation experiments applied to the test images of different modalities and slice directions (Slc), either transversal (Tr) or sagittal (Sa), are shown in Tables 6.3 and 6.4. Recall from Section 6.3.4 that the relation between the spatial support parameter  $m$  and the degree  $n$  of the B-spline kernels is  $n = 2m - 1$ . For every modality, slice direction, type of experiment, and FOM, the normalized errors figures resulting from spline interpolation in the five images were compared pairwise to the figures resulting from all other methods with the same spatial support for the corresponding convolution kernel. By using a paired  $t$ -test [6], the errors resulting from spline interpolation were found to be statistically significantly smaller ( $p < 0.05$ ), under the null hypothesis that the methods should yield similar results, except in those cases marked by the “★” symbol in Table 6.4.

## 6.5 Discussion

In the literature, several alternative approaches have been proposed for the evaluation or comparison of the accuracy of interpolation methods. In this section, we first discuss these approaches and explain why we have chosen the strategy described in the previous section. We also discuss the results of the present evaluation.

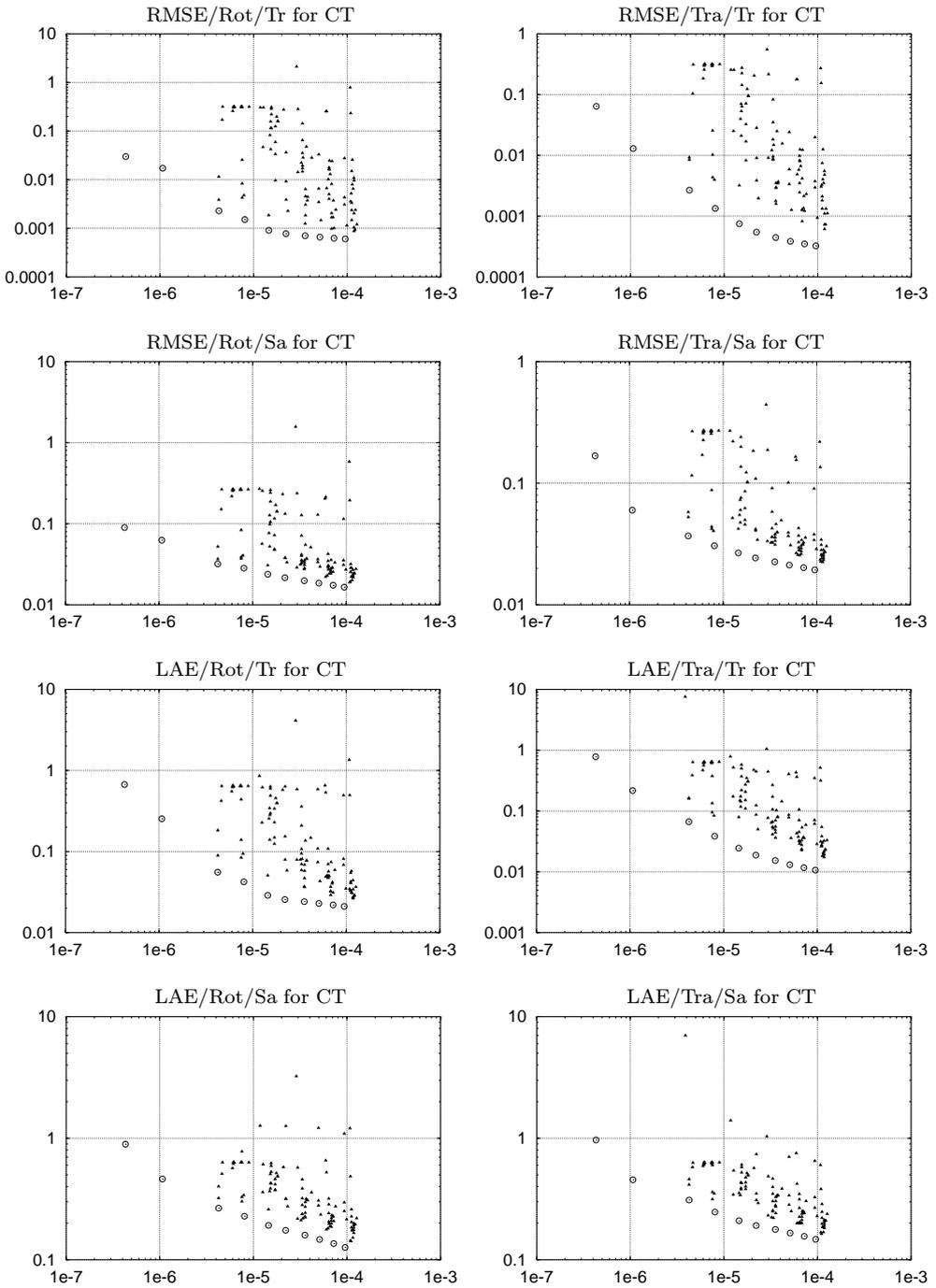
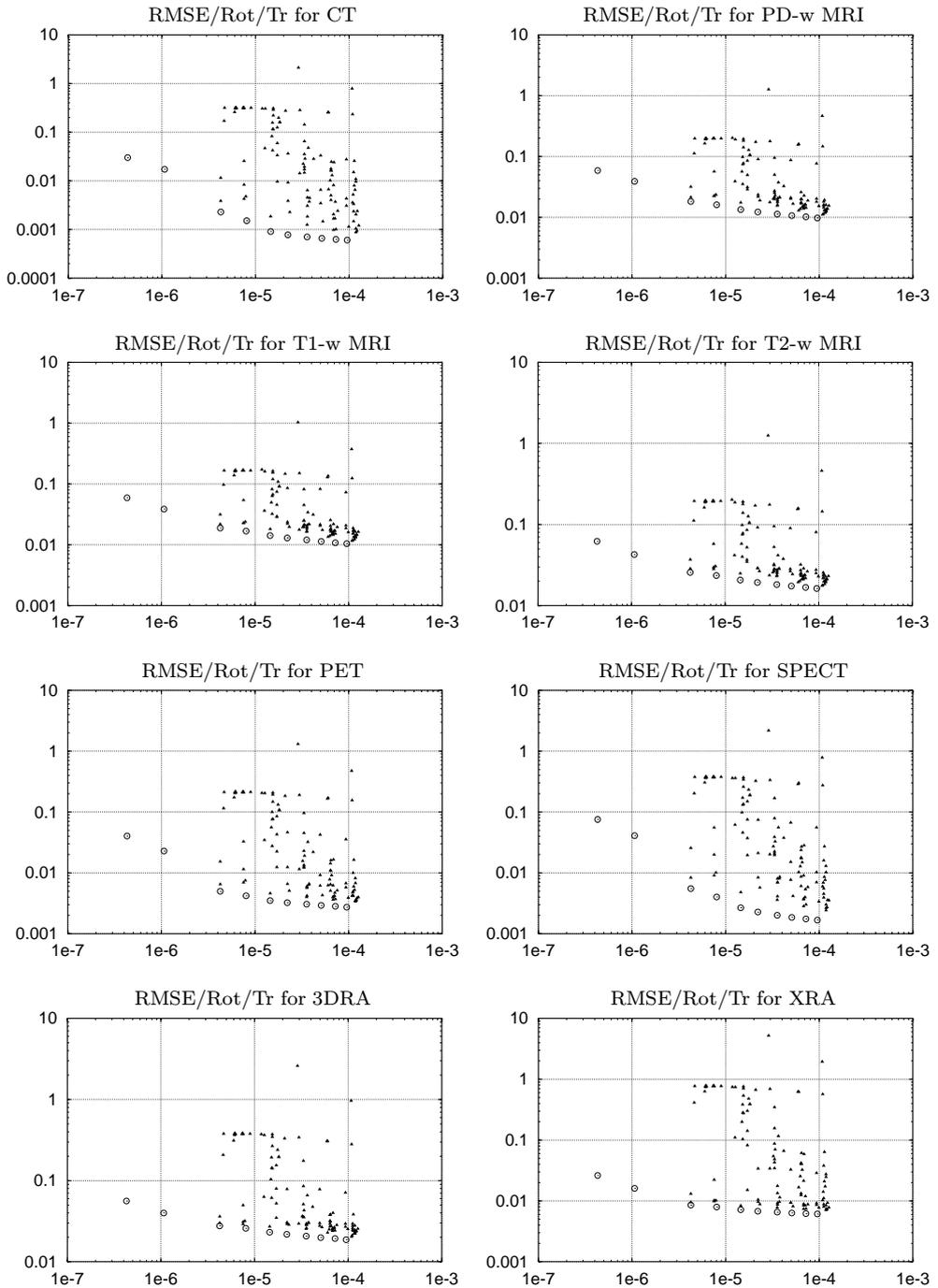


Figure 6.7. For explanation, see Page 135.



**Figure 6.8.** For explanation, see Page 135.

Averaged RMSEs (subpixel translation experiments)						
Modality	Slc	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
CT	Tr	1.30%	0.13%	0.05%	0.04%	0.03%
	Sa	5.99%	3.06%	2.43%	2.12%	1.94%
PD-w MRI	Tr	3.23%	1.45%	1.08%	0.90%	0.78%
	Sa	6.58%	3.56%	2.82%	2.46%	2.23%
T1-w MRI	Tr	3.22%	1.48%	1.11%	0.92%	0.80%
	Sa	6.59%	3.66%	2.97%	2.63%	2.42%
T2-w MRI	Tr	3.57%	1.96%	1.54%	1.31%	1.16%
	Sa	6.23%	3.84%	3.07%	2.65%	2.39%
PET	Tr	1.72%	0.34%	0.25%	0.22%	0.20%
	Sa	8.76%	4.16%	3.30%	2.88%	2.62%
SPECT	Tr	2.94%	0.33%	0.18%	0.14%	0.12%
	Sa	3.39%	0.75%	0.59%	0.50%	0.43%
3DRA	Tr	3.52%	2.28%	1.86%	1.61%	1.45%
	Sa	3.47%	2.02%	1.61%	1.39%	1.24%
XRA		1.26%	0.61%	0.48%	0.41%	0.37%

Averaged RMSEs (rotation experiments)						
Modality	Slc	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
CT	Tr	1.72%	0.15%	0.08%	0.07%	0.06%
	Sa	6.28%	2.84%	2.16%	1.85%	1.65%
PD-w MRI	Tr	3.90%	1.61%	1.22%	1.07%	0.98%
	Sa	7.27%	3.60%	2.84%	2.49%	2.27%
T1-w MRI	Tr	3.86%	1.68%	1.30%	1.13%	1.04%
	Sa	7.26%	3.66%	2.94%	2.62%	2.42%
T2-w MRI	Tr	4.27%	2.36%	1.94%	1.74%	1.63%
	Sa	6.93%	4.05%	3.31%	2.95%	2.73%
PET	Tr	2.28%	0.42%	0.32%	0.29%	0.27%
	Sa	8.59%	3.62%	2.81%	2.43%	2.19%
SPECT	Tr	4.07%	0.40%	0.23%	0.19%	0.17%
	Sa	4.96%	0.75%	0.51%	0.41%	0.35%
3DRA	Tr	4.00%	2.58%	2.18%	1.99%	1.87%
	Sa	4.34%	2.67%	2.24%	2.03%	1.91%
XRA		1.60%	0.79%	0.68%	0.63%	0.61%

**Table 6.3.** Averaged, normalized RMSEs introduced by linear ( $m = 1$ ), cubic ( $m = 2$ ), quintic ( $m = 3$ ), septic ( $m = 4$ ), and nonic ( $m = 5$ ) spline interpolation in the subpixel translation (**top**) and rotation (**bottom**) experiments.

Averaged LAEs (subpixel translation experiments)						
Modality	Slc	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
CT	Tr	21.61%	3.86%	1.89%	1.31%	*1.06%
	Sa	45.57%	24.71%	19.17%	16.63%	14.73%
PD-w MRI	Tr	42.35%	19.36%	13.27%	10.92%	9.31%
	Sa	48.36%	32.67%	*24.90%	*20.16%	16.98%
T1-w MRI	Tr	44.86%	21.69%	15.59%	12.17%	10.00%
	Sa	52.70%	31.93%	*25.75%	*21.71%	*18.92%
T2-w MRI	Tr	42.66%	22.03%	16.26%	13.23%	*11.11%
	Sa	38.58%	27.31%	*20.42%	*16.07%	*14.26%
PET	Tr	11.21%	2.38%	1.75%	1.39%	1.16%
	Sa	34.94%	18.84%	14.75%	11.82%	10.57%
SPECT	Tr	13.91%	2.10%	1.37%	1.02%	0.81%
	Sa	22.42%	7.19%	4.57%	3.54%	*2.97%
3DRA	Tr	38.77%	19.77%	14.07%	*11.29%	*9.46%
	Sa	29.26%	14.11%	9.53%	7.89%	6.89%
XRA		37.75%	20.56%	13.96%	*10.57%	*8.60%

Averaged LAEs (rotation experiments)						
Modality	Slc	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
CT	Tr	25.32%	4.23%	2.55%	*2.30%	*2.11%
	Sa	46.17%	22.82%	17.49%	14.69%	*12.66%
PD-w MRI	Tr	47.50%	19.13%	14.03%	*12.14%	*11.27%
	Sa	51.72%	*33.65%	*26.76%	*22.47%	*19.76%
T1-w MRI	Tr	47.32%	23.95%	*17.85%	*14.69%	*12.99%
	Sa	54.01%	29.45%	24.12%	21.34%	*18.86%
T2-w MRI	Tr	50.70%	*25.32%	18.51%	*15.69%	*14.07%
	Sa	49.75%	*33.51%	*26.55%	*22.13%	*19.63%
PET	Tr	14.33%	2.92%	2.36%	2.16%	2.04%
	Sa	35.86%	*18.06%	12.82%	10.25%	8.97%
SPECT	Tr	18.95%	2.33%	1.72%	1.41%	1.21%
	Sa	23.97%	*8.28%	5.37%	*4.18%	*3.49%
3DRA	Tr	43.76%	*24.03%	*17.39%	14.35%	13.35%
	Sa	41.56%	*25.42%	*20.01%	*17.11%	*15.34%
XRA	Tr	53.95%	*36.07%	*28.75%	*24.97%	*22.79%

**Table 6.4.** The averaged, normalized LAEs introduced by linear ( $m = 1$ ), cubic ( $m = 2$ ), quintic ( $m = 3$ ), septic ( $m = 4$ ), and nonic ( $m = 5$ ) spline interpolation in the subpixel translation (**top**) and rotation (**bottom**) experiments. See Section 6.4.2 for details, including the meaning of the “\*” symbol.

**Figure 6.7 (Page 131).** Scatter plots showing interpolation error (ordinates) *versus* computational cost (abscissae) for the different interpolation kernels applied to CT data. The label “A/B/C” on top of each plot provides details concerning the results shown, where A indicates the type of interpolation error (RMSE or LAE), B the type of experiment (rotation (Rot) or translation (Tra)), and C the type of slice (transversal (Tr) or sagittal (Sa)) on which the experiment was carried out. For every kernel in each plot, the presented FOM is an average of the individual FOMs (expressed as fractions of the dynamic range of grey values) resulting from the five datasets. Notice that the computational costs (shown here in seconds per voxel) were obtained from separate experiments. Open circles indicate the results of spline interpolation, where the left-most circle corresponds to zeroth-degree and the right-most to ninth-degree spline interpolation. See Section 6.4.2 for details.

**Figure 6.8 (Page 132).** Scatter plots showing interpolation error (ordinates) *versus* computational cost (abscissae) for the different interpolation kernels applied to all modalities (see the label on top of each plot) incorporated in this study. For every kernel in each plot, the FOM presented on the ordinate axis is an average of the RMSEs (expressed as fractions of the dynamic range of grey values) resulting from the rotation experiment carried out on five transversal slices. Notice that the computational costs (shown here in seconds per voxel) were obtained from separate experiments. As in Fig. 6.7, open circles indicate the results of spline interpolation, where the left-most circle corresponds to zeroth-degree and the right-most to ninth-degree spline interpolation. See Section 6.4.2 for details.

### 6.5.1 Discussion of Evaluation Strategies

A frequently used approach to the evaluation of interpolation kernels is to compare the spatial and spectral properties of these kernels to those of the sinc function, either by discussing their low-frequency band-pass and high-frequency suppression capabilities [228, 286], or by using such metrics as “sampling and reconstruction (SR) blur” [284, 332], “smoothing”, “post-aliasing”, or “overshoot” [235], “truncation error”, or “non-sinc error” [227], to mention but a few. These approaches are based on the fundamental assumption that in all cases, the sinc function is the optimal interpolation kernel. As such, they provide insight in the theoretical behavior of these kernels as low-pass filters. However, the conclusions of such evaluations are often not easily translated to specific image processing tasks. Alternatively, interpolation kernels may be compared by subjective visual inspection of image quality, after having used the kernels to perform certain resampling operations [70, 159, 286, 338, 420], or by analyzing their abilities to reconstruct certain mathematical test functions [180, 352]. However, given an image processing task, the most useful evaluation is obtained by applying the kernels to perform that task and then to compare the results to what is considered the gold standard.

In a recently published paper by Grevera & Udupa [128], an elaborate comparison of a number of well-known scene-based and object-based interpolation methods was presented. In the evaluation, 3D medical images from different modalities were first subsampled in the slice direction with a factor of two. Next, the subsampled images

were supersampled with the same factor in order to restore the original dimensions, where the supersampling was carried out by using the different interpolation methods. The subsampled-supersampled images were then compared to their originals by using different FOMs. We note that this evaluation approach was designed to assess the performance of interpolation methods for a specific task: increasing the number of slices for the purpose of improved 3D object quantification or visualization. The conclusions of this study can not simply be generalized to other interpolation problems, such as those occurring in *e.g.* image registration. In addition, two properties of this evaluation strategy are questionable. First, it is known from Fourier analysis that subsampling introduces aliasing artifacts which are not easily corrected by interpolation. Because of the low spatial resolution, this is especially true for the slice direction. These aliasing errors may have influenced the results and conclusions. For example, in some cases the cubic convolution kernel resulting from the flatness constraint (referred to as the modified cubic spline) performed statistically significantly worse than linear interpolation, while it is known from many other studies [180, 249, 278, 285, 286, 420] (including the present one) that the former kernel is generally superior. Second, the evaluation does not assess the performance of entire kernels, but only of a few distinct function values of these kernels. For example, in the evaluation of the cubic convolution kernel, only the values at  $x = -1.5, -0.5, 0.5, 1.5$  are taken into consideration. This implies that any other function that has the same values at these points would have given the same results.

A frequently used alternative approach to study the performance of interpolation kernels for the purpose of applying geometrical transformations, is to apply these transformations to a number of test images, followed by the inverse transformation so as to bring the images back in their original position [65, 134, 213, 227, 249, 278]. Ideally, the forward-backward transformed images should be identical to their respective originals, so that a quantitative performance measure can be based on the grey-value differences between the images. Although this approach may be of value when comparing certain families of interpolation kernels, its use is limited in the case of a large number of fundamentally different kernels, since the negative effects of a kernel in the forward transformation may be canceled out by the backward transformation. This occurs *e.g.* when employing a nearest-neighbor interpolation scheme in a forward-backward subpixel translation operation. While we know that this type of interpolation yields very large errors in the forward transformation, the backward transformed image is nevertheless exactly identical to the original image.

In the research described in this chapter, we used an alternative evaluation strategy. Rather than analyzing the spatial and spectral properties of interpolation kernels compared to the sinc function, we studied the actual performance of these kernels for the specific task of applying geometrical transformations to real medical image data. The strategy is a refined version of an approach used by Unser *et al.* [382], who considered rotation over  $16 \times 22.5^\circ = 360^\circ$ . The approach is entirely objective in the sense that it does not involve artificially created gold standards. It circumvents the aforementioned problems with other approaches: the test images are treated at their intrinsic resolution, thereby avoiding additional aliasing artifacts due to subsampling. Furthermore, by taking into consideration a large number of different rotation angles and translation vectors, interpolation errors are contributed to by the entire shape

of the kernels, not just by a limited number of kernel values.<sup>5</sup> Finally, only forward transformations are applied in order to better avoid cancellation of errors.

### 6.5.2 Discussion of the Results

As follows from the results presented in Section 6.4.2, the RMSEs introduced by spline interpolation were statistically significantly smaller than those caused by all other convolution-based interpolation approaches, regardless of image modality (CT, PD-w MRI, T1-w MRI, T2-w MRI, PET, SPECT, 3DRA, or XRA), slice direction (transversal or sagittal), or type of transformation (rotation or translation). However, this was not always the case for the LAEs. Although according to this figure of merit, linear interpolation (first-degree spline interpolation) performed statistically significantly better than all other kernels with a spatial support of two grid intervals ( $m = 1$ ), cubic spline interpolation did not perform statistically significantly better than cubic convolution and cubic Lagrange interpolation in the cases marked by the “ $\star$ ” symbol in Table 6.4 (“ $m = 2$ ” columns). In the higher-degree non-significant cases, especially the Welch, Cosine, Kaiser, and Lanczos windowed sinc kernels (in that order) performed comparably to spline interpolation. However, since these alternative methods did not perform statistically significantly better than spline interpolation, nothing is lost by using spline interpolation in these cases.

An explanation for the superiority of spline interpolation in the vast majority of cases may be obtained from approximation theory: it has been shown recently by Blu & Unser [20, 21] that spline interpolation has the largest possible order of approximation, given the spatial support of the B-spline convolution kernel. This implies that given the samples of any originally continuous input image, the interpolated image resulting from splines converges most rapidly to the original image as the inter-sample distance vanishes. Although there exist other interpolation kernels with this property, such as the Lagrange central interpolation kernels, splines have the unique additional property that they also yield the smoothest interpolant: in contrast with all other approaches considered in this chapter,  $n$ th-degree spline interpolation results in an interpolant which is  $n - 1$  times continuously differentiable. See Table 6.5 for an overview of the convolution kernels incorporated in this study and their corresponding properties as mentioned in this paragraph: spatial support, smoothness or regularity, and the rate of convergence of the resulting interpolant.

In order to give an impression of the errors introduced by spline interpolation of different degrees, the results of the rotation experiment for a transversal slice of a CT and a T1-weighted MRI dataset, as well as a sagittal slice of a PET dataset, are shown in Figs. 6.9, 6.10, and 6.11, respectively. (Notice that in the latter figure, the

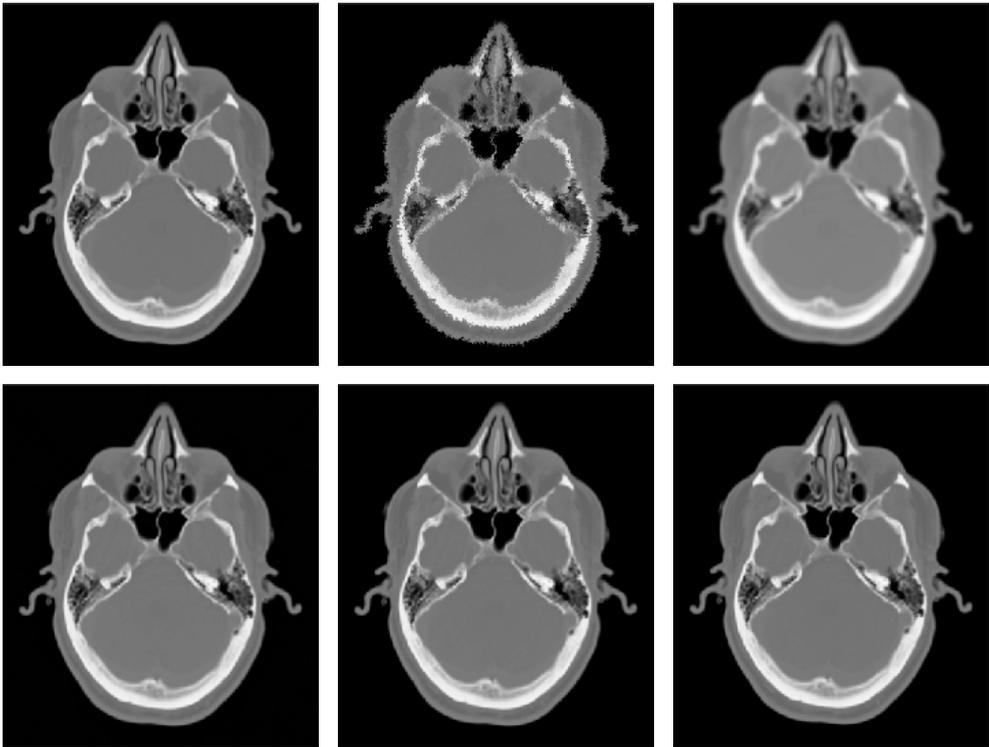
<sup>5</sup>We note that in this evaluation we have considered only subpixel translations over less than 0.5 pixels and rotation angles smaller than  $45^\circ$ . We claim that this is sufficient to demonstrate the performance of the interpolation kernels. It can easily be seen that when performing a translation over  $k + \gamma$  pixels, with  $k \in \mathbb{Z}$  and  $\gamma \in [0, 1) \subset \mathbb{R}$ , the required kernel values are determined by  $\gamma$ , not by  $k$ . Furthermore, due to the symmetry of all kernels, a translation over  $0.5 \leq \gamma < 1.0$  pixels involves the same kernel values as a translation over  $1.0 - \gamma$  pixels. Similarly, when rotating an image around its center over  $90\kappa + \varphi$  degrees, with  $\kappa \in \mathbb{Z}$  and  $\varphi \in [0, 90) \subset \mathbb{R}$ , the required kernel values are determined by  $\varphi$ , not by  $\kappa$ , and due to the symmetry of the operation, a rotation over  $45 \leq \varphi < 90$  degrees involves the same kernel values as a rotation over  $90 - \varphi$  degrees.

Kernel	Support	Smoothness	Convergence	Notes
Nearest-neighbor	1	–	$\mathcal{O}(\Delta^1)$	
Linear	2	$C^0$	$\mathcal{O}(\Delta^2)$	
Lagrange	$n + 1$	$C^0$	$\mathcal{O}(\Delta^{n+1})$	$n \geq 1$ odd
	$n + 1$	–	$\mathcal{O}(\Delta^{n+1})$	$n \geq 2$ even
Dodgson	3	$C^0$	$\mathcal{O}(\Delta^2)$	
Generalized convolution	$n + 1$	$C^{n-2}$	$\mathcal{O}(\Delta^3)$	$n \geq 3$ odd, $\alpha = \alpha_b$
	$n + 1$	$C^{n-2}$	$\mathcal{O}(\Delta^1)$	$n \geq 3$ odd, $\alpha \neq \alpha_b$
B-spline	$n + 1$	$C^{n-1}$	$\mathcal{O}(\Delta^{n+1})$	requires prefiltering
Bartlett, Blackman-Harris, Hamming windowed sinc	$2m$	$C^0$	$\mathcal{O}(\Delta^0)$	
Gaussian windowed sinc	$2m$	$C^0$	$\mathcal{O}(\Delta^0)$	$\forall \alpha \in \mathbb{R}^+$
Kaiser windowed sinc	$2m$	$C^0$	$\mathcal{O}(\Delta^0)$	$\forall \alpha \in \mathbb{R}^+, \alpha < \infty$
Cosine, Lanczos, Welch windowed sinc	$2m$	$C^1$	$\mathcal{O}(\Delta^0)$	
Blackman, Bohman, Hann windowed sinc	$2m$	$C^2$	$\mathcal{O}(\Delta^0)$	

**Table 6.5.** The convolution kernels described in Section 6.3 and some of their properties: spatial support, smoothness or regularity, and the rate of convergence of the resulting interpolant. In the second-last column, “ $\mathcal{O}$ ” denotes Landau’s order symbol and  $\Delta$  is the inter-sample distance as described in Section 6.2. See Section 6.5.2 and Appendix 6.A for more details.

displayed images were scaled so as to visually correct for the voxel anisotropy.) As was to be expected from the figures in Tables 6.3 and 6.4, the interpolation errors in the CT image are smallest. We note that the errors made in the rotation and subpixel translation experiments are cumulative errors. That is, they are considerably larger than the errors in practical interpolation problems of the same nature; one usually does not perform *e.g.* rotation by successive intermediate rotations. Nevertheless, the experiments give a representative impression of the average relative performance of the different interpolation kernels.

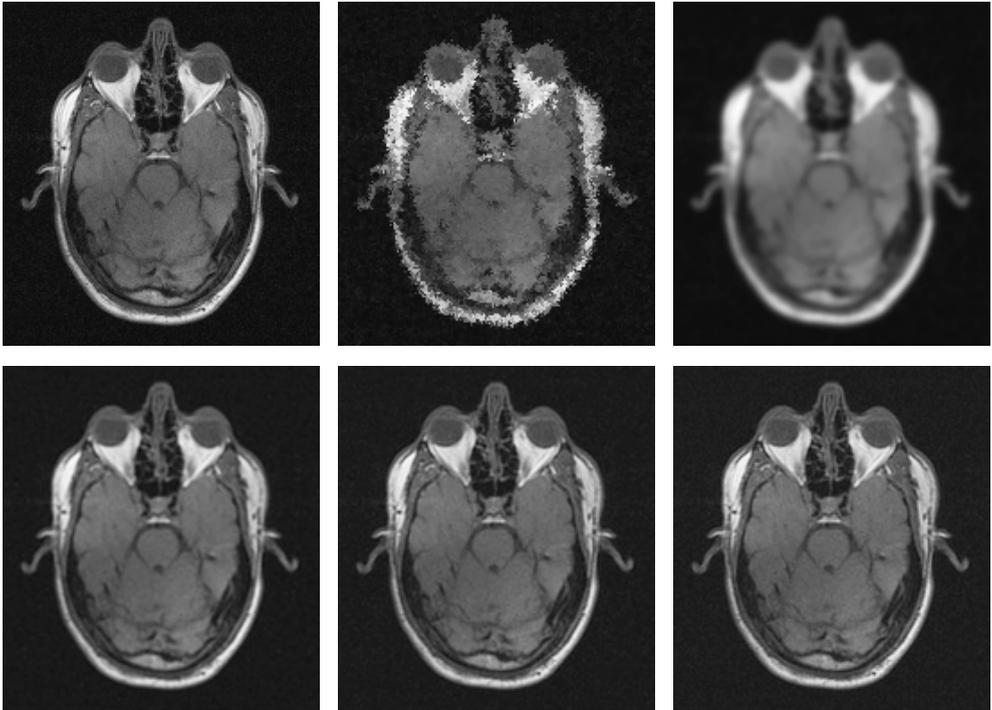
As can be observed from the results of the subpixel translation experiment shown in Tables 6.3 and 6.4, the errors in the through-plane direction can be much larger than those made in the in-plane direction in images with a relatively large voxel anisotropy (in our experiments notably the CT, MRI, and PET images). This can be explained from sampling theory: the lower the sampling frequency, the more pre- and post-aliasing artifacts can be expected to be introduced by sampling and non-ideal reconstruction operations. The results indicate that in order to reduce interpolation



**Figure 6.9.** Visual impression of the errors resulting from the rotation experiment carried out on a transversal slice of a CT dataset (**top left**), when using (**from top middle to bottom right**) nearest-neighbor or zeroth-degree spline interpolation, linear or first-degree spline interpolation, and cubic, quintic, and septic spline interpolation, respectively.

errors when performing 3D geometrical transformations, it is inefficient to choose a larger (more expensive) kernel for the in-plane interpolations, if nothing is done to considerably improve the through-plane interpolations. To give an example, for the CT and PET images considered in this evaluation, about ninth-degree spline interpolation was required in the through-plane direction in order to have similar RMSEs as linear interpolation in the in-plane direction (Table 6.3). For the other modalities, the difference between in-plane and through-plane interpolation errors was less drastic, due to the smaller voxel anisotropy.

Finally, a note concerning the computational cost of spline interpolation. As explained in Section 6.3.4, interpolation by means of B-spline convolution kernels requires prefiltering of the raw image data for all degrees  $n \geq 2$ . Although the timing experiments indicated that spline interpolation (including the prefiltering) is computationally cheaper than windowed sinc interpolation, it is somewhat more expensive than the alternative piecewise polynomial schemes. When using look-up tables, as discussed in Section 6.4.2, the required prefiltering causes spline interpolation to be

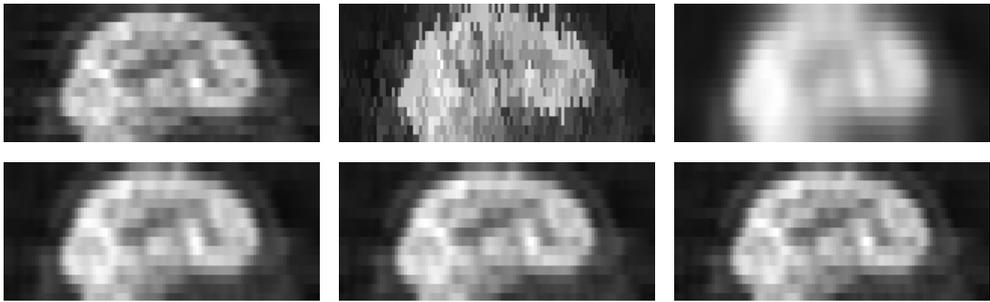


**Figure 6.10.** Visual impression of the errors resulting from the rotation experiment carried out on a transversal slice of a T1-weighted MRI dataset (**top left**), when using (**from top middle to bottom right**) nearest-neighbor or zeroth-degree spline interpolation, linear or first-degree spline interpolation, and cubic, quintic, and septic spline interpolation, respectively.

the computationally most expensive approach. However, since the prefiltering operations can always be carried out separably, their computational cost becomes relatively small in higher-dimensional interpolation problems. Moreover, in applications where many transformations have to be applied to the original image, such as in registration and visualization, the prefiltering needs to be carried out only once, so that the additional cost becomes negligible. Therefore, in the plots shown in Figs. 6.7 and 6.8, only the computational costs of the actual convolution operations were used.

## 6.6 Conclusions

In this chapter, we presented the results of a quantitative evaluation of sinc-approximating kernels for convolution-based medical image interpolation. The evaluation comprised the application of geometrical transformations (rotations and subpixel translations) to medical images from different modalities (CT, MRI, PET, SPECT, 3DRA, and XRA), by using the different kernels. The interpolation errors in the result-



**Figure 6.11.** Visual impression of the errors resulting from the rotation experiment carried out on a sagittal slice of a PET dataset (**top left**), when using (**from top middle to bottom right**) nearest-neighbor or zeroth-degree spline interpolation, linear or first-degree spline interpolation, and cubic, quintic, and septic spline interpolation, respectively.

ing transformed images were analyzed by computing the root-mean-square and the largest absolute deviation from the corresponding reference images. The evaluation was designed in such a way that the original images could be used as references. A total of 126 different kernels were evaluated. These included piecewise polynomial kernels (nearest-neighbor, linear, Lagrange, generalized convolution, and B-spline kernels) and a large number of windowed sinc kernels (Bartlett, Blackman, Blackman-Harris, Bohman, Cosine, Gaussian, Hamming, Hann, Kaiser, Lanczos, Rectangular, and Welch windows), with spatial supports ranging from 2 to 10 grid intervals.

The combined results of accuracy and timing experiments showed that regardless of image modality, slice direction (transversal or sagittal), type of transformation (rotation or translation), or figure of merit (RMSE or LAE), spline interpolation constitutes the best trade-off between accuracy and computational cost. That is to say, none of the other approaches included in this study was more accurate and at the same time computationally cheaper. In addition, pairwise comparisons of the error figures resulting from kernels with equal spatial support indicated that spline interpolation is statistically significantly better in the vast majority of cases. Therefore we conclude that spline interpolation is to be preferred over all other methods.

The results also revealed that, especially in images with a relatively large voxel anisotropy (in our experiments notably the CT, MRI, and PET images), the errors caused by interpolation in the through-plane direction are considerably larger than those resulting from interpolation in the in-plane direction. This implies that in general, it requires higher-degree spline interpolation in the through-plane direction in order to have similar errors as linear interpolation in the in-plane direction.

When comparing different degrees of spline interpolation, it can be concluded that cubic spline interpolation results in a considerable (28%–91%) reduction of interpolation errors as compared to linear interpolation (first-degree spline interpolation). Even better results (66%–98% reduction) are obtained with higher-degree spline interpolation, albeit at a considerable increase in computational cost.

## 6.A Appendix: Piecewise Polynomial Interpolators and B-Splines

In Section 6.5.2, some of the theoretical properties of the interpolation kernels described in Section 6.3 were discussed briefly and summarized in Table 6.5. By comparing the figures presented in this table, it was concluded that the interpolants obtained by B-spline interpolation are smoother and converge faster to the original continuous images than those resulting from any of the other convolution-based interpolation methods described in this chapter.

When using B-splines of at most degree  $n$ , it is possible to reproduce any polynomial of the same degree, *i.e.*, including the alternative piecewise polynomial kernels mentioned in this chapter. The purpose of this appendix is to give more insight in the properties of the alternative piecewise polynomial kernels by explicitly expressing these kernels in terms of B-splines. The relations presented here were obtained by computing the Fourier transform of the kernels and by factoring out  $\text{sinc}^{n+1}(f)$ , which corresponds to  $\beta^n(x)$  in the spatial domain.

As already mentioned in Section 6.3.4, the zeroth-degree and first-degree B-splines are identical to, respectively, the nearest-neighbor and linear interpolation kernel:

$$\zeta(x) = \beta^0(x), \quad (6.21)$$

and

$$\phi(x) = \beta^1(x). \quad (6.22)$$

For the Lagrange central interpolation kernels described in Section 6.3.2, the following relations can be derived:

$$\lambda^1(x) = \beta^1(x), \quad (6.23)$$

$$\lambda^2(x) = \left(1 - \frac{1}{8} \frac{\partial^2}{\partial x^2}\right) \beta^2(x), \quad (6.24)$$

$$\lambda^3(x) = \left(1 - \frac{1}{6} \frac{\partial^2}{\partial x^2}\right) \beta^3(x), \quad (6.25)$$

$$\lambda^4(x) = \left(1 - \frac{5}{24} \frac{\partial^2}{\partial x^2} + \frac{3}{128} \frac{\partial^4}{\partial x^4}\right) \beta^4(x), \quad (6.26)$$

$$\lambda^5(x) = \left(1 - \frac{1}{4} \frac{\partial^2}{\partial x^2} + \frac{1}{30} \frac{\partial^4}{\partial x^4}\right) \beta^5(x), \quad (6.27)$$

$$\lambda^6(x) = \left(1 - \frac{7}{24} \frac{\partial^2}{\partial x^2} + \frac{259}{5760} \frac{\partial^4}{\partial x^4} - \frac{5}{1024} \frac{\partial^6}{\partial x^6}\right) \beta^6(x), \quad (6.28)$$

$$\lambda^7(x) = \left(1 - \frac{1}{3} \frac{\partial^2}{\partial x^2} + \frac{7}{120} \frac{\partial^4}{\partial x^4} - \frac{1}{140} \frac{\partial^6}{\partial x^6}\right) \beta^7(x), \quad (6.29)$$

$$\lambda^8(x) = \left(1 - \frac{3}{8} \frac{\partial^2}{\partial x^2} + \frac{47}{640} \frac{\partial^4}{\partial x^4} - \frac{3229}{322560} \frac{\partial^6}{\partial x^6} + \frac{35}{32768} \frac{\partial^8}{\partial x^8}\right) \beta^8(x), \quad (6.30)$$

$$\lambda^9(x) = \left(1 - \frac{5}{12} \frac{\partial^2}{\partial x^2} + \frac{13}{144} \frac{\partial^4}{\partial x^4} - \frac{41}{3024} \frac{\partial^6}{\partial x^6} + \frac{1}{630} \frac{\partial^8}{\partial x^8}\right) \beta^9(x). \quad (6.31)$$

As can be observed from Eqs. (6.23)–(6.31), any  $n$ th-degree Lagrange central interpolation kernel can be expressed in terms of a B-spline of the same degree and its derivatives. The presence of higher-order derivatives does not influence the approximation order of the composition; it equals that of the corresponding B-spline. This can be confirmed by testing the so called Strang-Fix conditions, as described by Blu & Unser [20, 21] and also by Thévenaz *et al.* [367]. The derivatives do, however, reduce the smoothness of the composition. Since the kernels are symmetric, only even-order derivatives are involved, which explains why the odd-degree kernels are continuous and the even-degree kernels are not (see also Table 6.5).

Dodgson's quadratic convolution kernel described in Section 6.3.3 can be expressed in terms of first- and second-degree B-splines as follows:

$$\psi^2(x) = 2\beta^2(x) - \frac{1}{2} (\beta^1(x - \frac{1}{2}) + \beta^1(x + \frac{1}{2})). \quad (6.32)$$

For the odd-degree generalized convolution kernels described in that same section, the following relations exist:

$$\psi^3(x) = 3\beta^3(x) - (\beta^2(x - \frac{1}{2}) + \beta^2(x + \frac{1}{2})), \quad (6.33)$$

$$\begin{aligned} \psi^5(x) = -\frac{3}{8} \int_{-\infty}^x \int_{-\infty}^{t_1} & \left( 110\beta^3(t) + \right. \\ & 15 (\beta^3(t-1) + \beta^3(t+1)) - \\ & 67 (\beta^2(t - \frac{1}{2}) + \beta^2(t + \frac{1}{2})) - \\ & \left. 3 (\beta^2(t - \frac{3}{2}) + \beta^2(t + \frac{3}{2})) \right) dt dt_1, \end{aligned} \quad (6.34)$$

$$\begin{aligned} \psi^7(x) = \frac{5}{578} \int_{-\infty}^x \int_{-\infty}^{t_3} \int_{-\infty}^{t_2} \int_{-\infty}^{t_1} & \left( 87038\beta^3(t) + \right. \\ & 26880 (\beta^3(t-1) + \beta^3(t+1)) + \\ & 497 (\beta^3(t-2) + \beta^3(t+2)) - \\ & 62512 (\beta^2(t - \frac{1}{2}) + \beta^2(t + \frac{1}{2})) - \\ & 8313 (\beta^2(t - \frac{3}{2}) + \beta^2(t + \frac{3}{2})) - \\ & \left. 71 (\beta^2(t - \frac{5}{2}) + \beta^2(t + \frac{5}{2})) \right) dt dt_1 dt_2 dt_3, \end{aligned} \quad (6.35)$$

$$\begin{aligned}
\psi^9(x) = \frac{-35}{684232} \int_{-\infty}^x \int_{-\infty}^{t_5} \cdots \int_{-\infty}^{t_1} & \left( 293240340\beta^3(t) + \right. \\
& 126402003 (\beta^3(t-1) + \beta^3(t+1)) + \\
& 8421606 (\beta^3(t-2) + \beta^3(t+2)) + \\
& 34461 (\beta^3(t-3) + \beta^3(t+3)) - \\
& 228578875 (\beta^2(t-\frac{1}{2}) + \beta^2(t+\frac{1}{2})) - \\
& 50983407 (\beta^2(t-\frac{3}{2}) + \beta^2(t+\frac{3}{2})) - \\
& 1912129 (\beta^2(t-\frac{5}{2}) + \beta^2(t+\frac{5}{2})) - \\
& \left. 3829 (\beta^2(t-\frac{7}{2}) + \beta^2(t+\frac{7}{2})) \right) dt dt_1 \cdots dt_5.
\end{aligned} \tag{6.36}$$

Notice that Eqs. (6.33)–(6.36) hold true only for  $\alpha = \alpha_b$ . Contrary to the Lagrange central interpolation kernels, any  $n$ th-degree generalized convolution kernel is composed of (integrated versions of) second- and third-degree B-splines. The integration operations do not influence the approximation order of the composition; for all kernels, it equals that of the second-degree B-spline, as can be confirmed by testing the aforementioned Strang-Fix conditions. Integration does, however, increase the smoothness of the resulting kernels (see also Table 6.5).

## 6.B Appendix: Implementation of Direct B-Spline Filters

Since the conclusion of the evaluation presented in this chapter is that spline interpolation is to be preferred in general, it may be useful to provide some more details concerning the efficient implementation of this type of interpolation. The outline of the treatise presented here was first described by Unser *et al.* [379–381].

As explained in Section 6.3.4,  $n$ th-degree spline interpolation is carried out separably in every dimension by means of convolution according to:

$$\sum_{k=-\infty}^{+\infty} c(k)\beta^n(x-k), \quad x \in \mathbb{R}, \tag{6.37}$$

where  $\beta^n$  is the  $n$ th-degree B-spline as defined in Eq. (6.16), and  $c(k)$ ,  $k \in \mathbb{Z}$ , are the so called B-spline coefficients, to be determined. In order for the convolution (6.37) to actually interpolate given samples  $s(k)$ ,  $k \in \mathbb{Z}$ , the following must hold:

$$\sum_{l=-\infty}^{+\infty} c(l)\beta^n(k-l) = s(k), \quad \forall k \in \mathbb{Z}. \tag{6.38}$$

The  $z$ -transform of Eq. (6.38) reads  $C(z)B^n(z) = S(z)$ , and consequently, the B-spline coefficients can be found by evaluating

$$C(z) = (B^n(z))^{-1}S(z). \tag{6.39}$$

In this equation,  $(B^n(z))^{-1}$  is called the direct B-spline filter of degree  $n$ . Since, by definition,  $B^n(z) = \sum_{k \in \mathbb{Z}} \beta^n(k) z^{-k}$ , this filter can be obtained simply by inserting Eq. (6.16). When doing so, it turns out that  $(B^n(z))^{-1} = 1$  for  $n = 0$  and  $n = 1$ , which implies that in these cases  $C(z) = S(z)$ , that is to say,  $c(k) = s(k)$ . For any  $n \geq 2$ , however,  $(B^n(z))^{-1}$  is a digital high-pass filter that corrects for the blurring introduced by the corresponding B-spline convolution kernel, due to the fact that this kernel does not possess the interpolation property expressed in Eq. (6.4).

In order to obtain an efficient implementation of the direct B-spline filter corresponding to any  $n \geq 2$ , the best approach is to factorize it. Factorization of  $(B^n(z))^{-1}$  involves the computation of its poles. Due to the fact that all B-spline kernels are symmetric, we have that  $(B^n(z))^{-1} = (B^n(z^{-1}))^{-1}$  for any degree  $n$ . This implies that all poles come in reciprocal pairs, and the filter can be written as

$$(B^n(z))^{-1} = c_n \prod_{i=1}^{\lfloor n/2 \rfloor} G(z; z_i), \quad (6.40)$$

where

$$c_n = \frac{1}{\beta^n(\lfloor n/2 \rfloor)}, \quad (6.41)$$

and

$$G(z; z_i) = \frac{1}{z^{-1}(z - z_i)(z - z_i^{-1})} = \frac{-z_i}{(1 - z_i z^{-1})(1 - z_i z)} \quad (6.42)$$

is the factor corresponding to the pole pair  $\{z_i, z_i^{-1}\}$ , with  $|z_i| < 1$ . Since the poles of  $(B^n(z))^{-1}$  are the zeroes of  $B^n(z)$ , they are obtained by solving  $B^n(z) = 0$ . Notice that for  $n \geq 6$ , this can only be done numerically (Abel's theorem [2]), since in those cases the degree of the resulting algebraic equation is larger than four. Numerical representations of the poles  $|z_i| < 1$  of the direct B-spline filters of degree  $n = 0, 1, \dots, 9$ , as used in this chapter, are shown in Table 6.6.

Efficient implementation of the factors  $G(z; z_i)$  is obtained by a further factorization according to:

$$G(z; z_i) = G^-(z; z_i)G^+(z; z_i), \quad (6.43)$$

with

$$G^+(z; z_i) = \frac{1}{(1 - z_i z^{-1})} \quad \text{and} \quad G^-(z; z_i) = \frac{-z_i}{(1 - z_i z)}. \quad (6.44)$$

By using the  $z$ -transform property  $s(k+l) \xleftrightarrow{z} z^l S(z)$ , it can easily be derived that  $G(z; z_i)$ , implemented by successively applying  $G^+(z; z_i)$  and  $G^-(z; z_i)$  described above, gives rise to the following recursive filters in the spatial domain:

$$s^+(k) = s(k) + z_i s^+(k-1), \quad (\text{causal filter}) \quad (6.45a)$$

$$s^-(k) = z_i (s^-(k+1) - s^+(k)), \quad (\text{anti-causal filter}) \quad (6.45b)$$

where the  $s(k)$  are input samples, the  $s^+(k)$  are intermediate output samples resulting from the causal filter, and the  $s^-(k)$  are the final output samples resulting from the subsequent application of the anti-causal filter.

Degree	Poles	Degree	Poles
$n = 0$	–	$n = 1$	–
$n = 2$	$z_1 = -1.71572875254 \cdot 10^{-1}$	$n = 3$	$z_1 = -2.67949192431 \cdot 10^{-1}$
$n = 4$	$z_1 = -3.61341225900 \cdot 10^{-1}$ $z_2 = -1.37254292973 \cdot 10^{-2}$	$n = 5$	$z_1 = -4.30575347100 \cdot 10^{-1}$ $z_2 = -4.30962882033 \cdot 10^{-2}$
$n = 6$	$z_1 = -4.88294589303 \cdot 10^{-1}$ $z_2 = -8.16792710762 \cdot 10^{-2}$ $z_3 = -1.41415180833 \cdot 10^{-3}$	$n = 7$	$z_1 = -5.35280430796 \cdot 10^{-1}$ $z_2 = -1.22554615192 \cdot 10^{-1}$ $z_3 = -9.14869480961 \cdot 10^{-3}$
$n = 8$	$z_1 = -5.74686909249 \cdot 10^{-1}$ $z_2 = -1.63035269297 \cdot 10^{-1}$ $z_3 = -2.36322946948 \cdot 10^{-2}$ $z_4 = -1.53821310642 \cdot 10^{-4}$	$n = 9$	$z_1 = -6.07997389169 \cdot 10^{-1}$ $z_2 = -2.01750520193 \cdot 10^{-1}$ $z_3 = -4.32226085405 \cdot 10^{-2}$ $z_4 = -2.12130690318 \cdot 10^{-3}$

**Table 6.6.** Numerical representations (12-significant decimals) of the poles  $|z_i| < 1$ ,  $i = 1, \dots, \lfloor n/2 \rfloor$ , of the direct B-spline filters of degree  $n = 0, 1, \dots, 9$ .

Since in practice these filters will be applied to images of finite extent, it remains to describe how to compute the initial values of the recursions given in Eqs. (6.45a) and (6.45b). That is, given samples  $s(k)$ ,  $k = 0, 1, \dots, K - 1$ , where  $K$  denotes the total number of samples, how to compute  $s^+(0)$  and  $s^-(K - 1)$ ? The former may be obtained by writing the application of  $G^+(z; z_i)$  to  $S(z)$  as a convolution, rather than a recursion, in the spatial domain:

$$s^+(k) = \sum_{l=-\infty}^{+\infty} g^+(l; z_i) s(k - l), \tag{6.46}$$

where  $g^+(k; z_i)$  is the inverse  $z$ -transform of  $G^+(z; z_i)$  which, by using the  $z$ -transform pair  $a^k u(k) \xleftrightarrow{z} z/(z - a)$ , which holds true for  $|z| > |a|$  (see *e.g.* Kwakernaak & Sivan [200, p. 492]), can be derived to be

$$g^+(k; z_i) = z_i^k u(k), \quad \text{with} \quad u(k) = \begin{cases} 1 & \text{if } k \geq 0, \\ 0 & \text{if } k < 0. \end{cases} \tag{6.47}$$

By substituting (6.47) into (6.46) and by using mirror-boundary conditions, that is,  $s(-k) = s(k)$ , which turns the original finite-extent signal into an infinite-extent one with period  $2K - 2$ , we have that

$$s^+(0) = \sum_{l=0}^{+\infty} z_i^l s(l) = \sum_{k=0}^{+\infty} (z_i^{2K-2})^k \sum_{l=0}^{2K-3} z_i^l s(l) = \frac{1}{1 - z_i^{2K-2}} \sum_{l=0}^{2K-3} z_i^l s(l). \tag{6.48}$$

The initial value for the anti-causal recursion,  $s^-(K-1)$ , may be obtained by writing the application of the compound filter  $G(z; z_i)$  to  $S(z)$  as a convolution in the spatial domain:

$$s^-(k) = \sum_{l=-\infty}^{+\infty} g(l; z_i) s(k-l), \quad (6.49)$$

where  $g(k; z_i)$  is the inverse  $z$ -transform of  $G(z; z_i)$ , which may be obtained by using the partial-fraction expansion of the right-hand side of Eq. (6.42):

$$G(z; z_i) = \frac{-z_i}{(1-z_i^2)} \left( \frac{1}{1-z_i z^{-1}} + \frac{1}{1-z_i z} - 1 \right), \quad (6.50)$$

in combination with the following  $z$ -transform pairs:  $a^k u(k) \xleftrightarrow{z} z/(z-a)$ , which holds for  $|z| > |a|$ , furthermore  $-a^k u(-k) \xleftrightarrow{z} a/(z-a)$ , which holds for  $|z| < |a|$ , and finally  $1 \xleftrightarrow{z} \delta(k)$ , which holds for all  $z \in \mathbb{C}$  (see *e.g.* Kwakernaak & Sivan [200, p. 492]). Together, this results in

$$g(k; z_i) = \frac{-z_i}{(1-z_i^2)} (z_i^k u(k) + z_i^{-k} u(-k) - \delta(k)). \quad (6.51)$$

By substituting (6.51) into (6.49) and by again using mirror-boundary conditions, that is,  $s(K-1+l) = s(K-1-l)$ , it can easily be derived that

$$s^-(K-1) = \frac{-z_i}{(1-z_i^2)} (2s^+(K-1) - s(K-1)). \quad (6.52)$$

In summary, interpolation by means of a B-spline kernel of degree  $n \geq 2$  requires prefiltering of the raw data in order to correct for the blurring nature of the kernel. This prefiltering is accomplished by carrying out the following operations on the given samples  $s(k)$ ,  $k = 0, 1, \dots, K-1$ :

- 1) Compute the initial value  $s^+(0)$  for the causal filter by evaluating the right-hand side of Eq. (6.48), using  $z_i = z_1$  as given in Table 6.6.<sup>6</sup>
- 2) Apply the causal filter, Eq. (6.45a), for  $k = 1, 2, \dots, K-1$ , using  $z_i = z_1$ .
- 3) Compute the initial value  $s^-(K-1)$  for the anti-causal filter by using  $z_i = z_1$  in evaluating Eq. (6.52).
- 4) Apply the anti-causal filter, Eq. (6.45b), for  $k = K-2, \dots, 1$ , using  $z_i = z_1$ .
- 5) Repeat steps 1) – 4) for the remaining poles  $z_i$ ,  $i = 2, \dots, \lfloor n/2 \rfloor$ , belonging to degree  $n$ , as given in Table 6.6.
- 6) Multiply the resulting samples with the factor  $c_n$  given in Eq. (6.41).

In the case of multiple dimensions, steps 1) – 6) must be repeated separately in all dimensions. That is, first to the rows, then to the *resulting* columns, *etc.*

<sup>6</sup>Notice that if the number of samples,  $K$ , is sufficiently large, the summation may be terminated earlier than at  $l = 2K - 3$ , since the contributions of the samples corresponding to larger values of  $l$  are negligible due to the exponential decay of  $z_i^l$ .

