algorithm, nowadays one of the most popular because of its superior performance. We believe that the coding effort for the system shown in Fig. 2 is minimal and that there is a substantial reward in algorithmic performance when compared to previous implementations.

Another application is the use of second order R-filter as a Gaussian filter. This substitution is computationally very attractive, especially for larger standard deviations. The quality of the approximation to a true Gaussian, which should be sufficient for most applications (cf. Fig. 1), can be further improved through the use of repeated convolutions (a consequence of the central limit theorem). Such an approach is suitable for the generation of multiresolution or scale-space signal representations [20], [21]. It is certainly an interesting alternative to other suggested approximation methods [22].

Our implementation of R-filters is fully recursive and some care has to be taken to avoid propagation of roundoff errors. The simplest approach, which is the one that we selected, is to use floating point arithmetic. In image processing applications where it is often necessary to save memory storage, it is sufficient to use one auxiliary 1-D real array to store intermediate filtering results. The final output of the row or column filters can be truncated and stored in standard byte or integer format. Fixed point realizations are also conceivable, provided that an error analysis be performed to determine the appropriate number of bits per sample needed to maintain the error within an acceptable range [23].

Finally, we would like to mention that the availability of fast R-filtering techniques is crucial for the design of a new class of iterative algorithms for solving linear and space variant regularization problems. It is the investigation of such filtering-based algorithms that initially motivated the present study. We are currently studying the convergence properties of such schemes. They appear to be superior to the conventional Gauss–Seidel approach, particularly for large values of λ. Further, a problem that could benefit from this approach is the area-based estimation of optical flow in the Horn and Schunk formulation [11].

V. Conclusion

In this correspondence, we have investigated the properties of R-filters, a special class of smoothing operators with an adjustable scale parameter λ. These operators provide a convenient way of solving approximation problems with certain regularization constraints. We have developed general analysis and design techniques and applied them to the study of R-filters associated with the first and second difference operators. These filters have been fully characterized in terms of their impulse response, equivalent window size, and filter coefficients, expressed as functions of the regularization parameter λ.

The R-filters that have been described here have two essential features:

1) they can be implemented recursively with a small number of operations per sample value ($2n$ operations for a one-dimensional nth order R-filter);

2) their smoothing window can be tuned to any scale through a single parameter with no effect on execution speed.

Due to these properties, R-filters stand as attractive alternatives to standard moving average and Gaussian smoothers currently used in a wide variety of image processing and computer vision applications.

References

[1] A. N. Tikhonov and V. Y. Arsenin, *Solution of Ill-Posed Problems.* Washington, DC: Winston, 1977.
[2] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature,* vol. 317, pp. 314–319, 1985.
[3] M. Bertero, T. A. Poggio, and V. Torre, "Ill-posed problems in early vision," *Proc. IEEE,* vol. 76, pp. 869–889, Aug. 1988.
[4] W. K. Pratt, *Digital Image Processing.* New York: Wiley, 1978.
[5] C. H. Reinsh, "Smoothing by spline functions," *Numer. Math.,* vol. 10, pp. 177–183, 1967.
[6] T. Poggio, H. Voorhees, and A. Yuille, "Regularizing edge detection," Artificial Intell. Lab., Massachusetts Inst. Technol., Cambridge, Memo 776, 1984.
[7] V. Torre and T. A. Poggio, "On edge detection," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-8, pp. 147–163, Mar. 1986.
[8] T. Poggio, H. Voorhees, and A. Yuille, "A regularized solution to edge detection," *J. Complexity,* vol. 4, pp. 106–123, 1988.
[9] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London B,* vol. 207, pp. 187–217, 1980.
[10] D. Marr, *Vision.* San Francisco, CA: Freeman, 1982.
[11] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intell.,* vol. 17, pp. 185–203, 1981.
[12] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-8, pp. 679–697, 1986.
[13] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," *Int. J. Comput. Vision,* vol. 1, pp. 167–187, 1987.
[14] R. M. Haralick, "Digital step edges from zero crossing of second directional derivatives," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-6, pp. 58–68, Jan. 1984.
[15] M. Unser, G. Pelle, P. Brun, and M. Eden, "Automated extraction of serial myocardial borders from *M*-mode echocardiograms," *IEEE Trans. Med. Imaging,* vol. MI-8, pp. 96–103, Mar. 1989.
[16] P. J. Burt, "Fast algorithms for estimating local image properties," *Comput. Graphics Image Processing,* vol. 21, pp. 368–382, 1983.
[17] M. Unser and M. Eden, "Multiresolution feature extraction and selection for texture segmentation," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 11, pp. 717–728, July 1989.
[18] J. S. Lee, "Digital image enhancement and noise filtering by the use of local statistics," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-2, pp. 165–168, 1980.
[19] M. Unser, "Improved restoration of noisy images by adaptive least-squares post-filtering," *Signal Processing,* vol. 20, pp. 3–14, May 1990.
[20] A. P. Witkin, "Scale-space filtering," in *Proc. 4th Int. Joint Conf. Artificial Intelligence,* 1983, pp. 1019–1022.
[21] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the Gaussian kernel for scale-space filtering," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-8, pp. 26–33, Jan. 1986.
[22] W. M. Wells, "Efficient synthesis of Gaussian filters by cascaded uniform filters," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-8, pp. 234–239, Mar. 1986.
[23] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975.

Fast B-Spline Transforms for Continuous Image Representation and Interpolation

Michael Unser, Akram Aldroubi, and Murray Eden

***Abstract*—This correspondence describes efficient algorithms for the continuous representation of a discrete signal in terms of B-splines (direct B-spline transform), and for interpolative signal reconstruction (indirect B-spline transform) with an expansion factor *m*.Expressions for the *z*-transforms of the sampled B-spline functions are determined and a convolution property of these kernels is established. It is shown that both the direct and indirect spline transforms involve linear operators that are**

Manuscript received June 14, 1989; revised July 19, 1990. Recommended for acceptance by R. J. Woodham.

M. Unser is with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, MD 20892, and INSERM, Unité 2, Hôpital Henri-Mondor, F-94010 Créteil, France.

A. Aldroubi and M. Eden are with the Biomedical and Instrumentation Program, National Institutes of Health, Bethesda, MD 20892.

IEEE Log Number 9040953.

translation invariant and are implemented efficiently by linear filtering. Fast computational algorithms based on the recursive implementation of these filters are proposed. A B-spline interpolator can also be characterized in terms of its transfer function and its global impulse response (cardinal spline of order *n*). The case of the cubic spline is treated in greater detail. The present approach is compared with previous methods that are reexamined from a critical point of view. We conclude that, contrary to the claims of several authors, B-spline interpolation correctly applied does not result in a loss of image resolution and that this type of interpolation can be performed in a very efficient manner.

Index Terms—**B-splines, continuous representation, image reconstruction, interpolation, polynomial splines, recursive filter, transform.**

I. Introduction

Image interpolation plays a central role in many applications [1], [2]. It is required for resolution conversion to adapt to the characteristics of a particular display device. Efficient scaling mechanisms are needed on modern display workstations to allow users to vary the size of images interactively, to concentrate on some detail, or to get a better overview. Image interpolation is also used in some coding schemes [3]. Other utilizations include geometrical transformations [1] and image registration [4], [5] where it is necessary to resample the image to an undistorted or reference coordinate system.

The principle that is common to all interpolation schemes is to determine the parameters of a continuous image representation from a set of discrete points. The simplest approaches are the 0 order (nearest neighbor) and 1 first order (bilinear) interpolations [1]. A more refined technique is the cubic B-spline interpolation method introduced by Hou and Andrews [6]. Because of its computational complexity, researchers have suggested several alternative cubic convolution techniques [7], [8]. Chen and deFigueiredo have proposed generalized spline filters based on partial differential equations [9].

B-spline interpolation methods have not been very popular for two principal reasons. First, the algorithms for the determination of the spline coefficients described in the signal and image processing literature are relatively inefficient. Second, it is commonly believed that the use of B-splines causes image resolution degradation [1], [2]. This misinterpretation is in essence a consequence of incorrect implementation. In some instances [1], [2], an image interpolation function is obtained by substituting the initial pixel values for the B-spline coefficients. The algorithm recently described in [10] is another example of a flawed implementation scheme that produces blurring.

The purpose of this correspondence is to present a theoretical analysis of B-spline signal representations from a signal processing point of view. The main advantage is that both the direct spline transform (the process of determining the expansion coefficients) and the indirect spline transform (the process of reconstructing the original sampled values with an optional interpolation) can be interpreted as simple filtering operations. We can derive fast computational algorithms by studying the recursive structure of these filters. We will also describe simple procedures to characterize a B-spline interpolator of any order in terms of its impulse and frequency responses. These concepts will be illustrated by the design of recursive filters for direct and indirect cubic spline transform. We will also compare our algorithm with previous approaches and highlight some of the points that have been a common source of misinterpretation. In particular, we will show that the method of Sankar *et al.* does not produce a signal interpolation satisfying the B-spline representation and that it is in effect equivalent to simple spatial smoothing by an iterated moving average.

II. B-Splines Basis Functions

The essential property of B-splines of order n is to provide a basis of the subspace of all continuous piecewise polynomial functions of degree n with derivatives up to order $n-1$ that are continuous everywhere on the real line [11], [12]. In the case of equally spaced integral knot points, any function $\phi^n(x)$ of this subspace can be represented as

$$\phi^n(x) = \sum_{i=-\infty}^{+\infty} c(i)\beta^n(x-i) \quad (2.1)$$

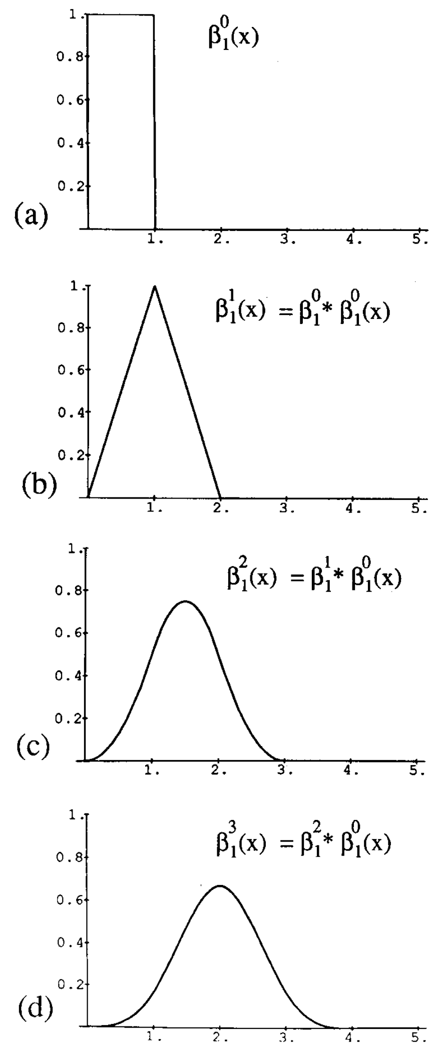

Fig. 1. Convolution property of the continuous B-spline functions. (a) Zero order spline, (b) first order spline, (c) quadratic spline, (d) cubic spline.

where $\beta^n(x)$ denotes the normalized B-spline of order n, defined below; n is also the degree of the piecewise polynomials that are connected at the knot points. The function $\phi^n(x)$ is uniquely determined by its B-spline coefficients $\{c(i)\}$. It is essentially the smoothness property of these functions as well as the compact support of the basis functions that make the use of B-spline representations attractive. The crucial step in B-spline interpolation is to determine the coefficients of this expansion such that $\phi^n(x)$ matches the values of some discrete sequence $\{f(k)\}$ at the knot points: $\phi^n(k) = f(k)$ for $(k = -\infty, \cdots, +\infty)$. In the mathematical literature, this problem is commonly referred to as the cardinal spline interpolation problem. The existence and uniqueness of the solution, as well as other fundamental mathematical results have been established by Schoenberg [13], [14].

In this section, we first recall the principal properties of continuous B-splines. We then investigate the properties of a class of discrete B-splines that are obtained by sampling continuous basis functions subject to a uniform dilatation. These preliminary results will be used extensively in Section III for the design of efficient algorithms for the direct and indirect B-spline transform.

A. B-Spline with Equally Spaced Knots

The normalized B-spline functions of order n with $n + 2$ equally spaced knots $\{0, 1, 2, \cdots, (n+1)\}$ are defined as [12, p. 135]:

$$\beta^n(x) = \sum_{j=0}^{n+1} \frac{(-1)^j}{n!} \binom{n+1}{j} (x-j)^n \mu(x-j) \tag{2.2}$$

where $\mu(x)$ is the step function

$$\mu(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases}$$

and where $\binom{n+1}{j}$ are the binomial coefficients:

$$\binom{n+1}{j} = \frac{(n+1)!}{(n+1-j)!\,j!} = \binom{n}{j} + \binom{n}{j-1}.$$

It is convenient to introduce a subscript notation to represent expanded basis functions by a factor m: $\beta_m^n(x) = \beta^n(x/m)$. These functions satisfy the convolution property

$$\beta_m^n(x) = \frac{1}{m}\beta_m^{n-1} * \beta_m^0(x) = \frac{1}{m^n} \underbrace{\beta_m^0 * \cdots * \beta_m^0}_{n+1 \text{ times}}(x) \tag{2.3}$$

as illustrated by Fig. 1. In some instances, it may be preferable to use symmetric basis functions centered on the origin. This is achieved by introducing the appropriate offset: $x_0 = (n+1)/2$. We have chosen here to stick with definition (2.2) because it simplifies subsequent derivations. Most of our results are directly applicable to symmetric basis functions by simply replacing all causal operators by their shifted symmetric counterparts.

B. Discrete B-Splines

We define discrete B-splines by sampling continuous spline functions. More specifically, the discrete B-spline of order n with a horizontal expansion factor m is given by

$$b_m^n(k) \triangleq \beta^n\left(\frac{k}{m}\right) = \frac{1}{m^n} \sum_{j=0}^{n+1} \frac{(-1)^j}{n!} \binom{n+1}{j} (k - jm)^n \mu(k - jm). \tag{2.4}$$

We will characterize these functions in terms of their z-transforms. For this purpose, we first need to determine the transform of the power series: $\{k^n; k = 0, \cdots, +\infty\}$. Let us define

$$P^n(z) = \sum_{k=0}^{+\infty} k^k z^{-k} = \sum_{k=-\infty}^{+\infty} k^n z^{-k} \mu(k). \tag{2.5}$$

It is straightforward to establish the recurrence equation

$$P^n(z) = -z \frac{\partial P^{n-1}(z)}{\partial z}. \tag{2.6}$$

By using the fact that the z-transform of $\mu(k)$ is

$$P^0(z) = \frac{1}{1 - z^{-1}},$$

we are able to evaluate the transform of all subsequent power series recursively. The results of these computations for $n = 0$ to 5 are summarized in Table I. We can easily show that the general term has the form

$$P^n(z) = \frac{A^n(z)}{(1 - z^{-1})^{n+1}}$$

where the numerator $A^n(z)$ is some polynomial in z^{-1}. The z-transform of (2.4) is found by making use of the shift theorem and substituting the expression for $P^n(z)$:

$$B_m^n(z) = \frac{1}{m^n} \sum_{j=0}^{n+1} \frac{1}{n!} \binom{n+1}{j} (-1)^j z^{-jm} \frac{A^n(z)}{(1 - z^{-1})^{n+1}}.$$

By noticing that $(-1)^j z^{-jm}$ is also equal to $(-z^{-m})^j$ and recalling that

$$(x+1)^{n+1} = \sum_{j=0}^{n+1} \binom{n+1}{j} x^j$$

we finally get

$$B_m^n(z) = \frac{1}{m^n} \frac{A^n(z)}{n!} \left(\frac{1 - z^{-m}}{1 - z^{-1}}\right)^{n+1}.$$

An equivalent but more useful expression is given by

$$B_m^n(z) = \frac{1}{m^n} B_1^n(z) \left(B_m^0(z)\right)^{n+1} \tag{2.7}$$

where

$$B_1^n(z) = \frac{A^n(z)}{n!} = \sum_{k=0}^{n+1} b_1^n(k) z^{-k} \tag{2.8}$$

is the z-transform of the discrete signal obtained by sampling the B-spline at its knots and where

$$B_m^0(z) = \frac{1 - z^{-m}}{1 - z^{-1}} = \sum_{k=0}^{m-1} z^{-k} \tag{2.9}$$

is the z-transform of a rectangular pulse of length m. Equation (2.7) clearly shows that we have a similar convolution relation as for the continuous case, that is:

$$b_m^n(k) = \frac{1}{m^n} \underbrace{\left(b_m^0 * b_m^0 * \cdots * b_m^0\right)}_{n+1 \text{ times}} * b_1^n(k). \tag{2.10}$$

This relationship is illustrated in Fig. 2 for the discrete cubic B-spline. Equations (2.7) and (2.10) are quite general and can be shown to be also valid for shifted sampled basis functions, in particular symmetrical ones. Comparison of (2.10) to its counterpart for the continuous case (2.3) reveals an interesting difference. In (2.10), there is an additional convolution with a normalized kernel $\{b_1^n(k)\}$. This operation is required to guarantee that discrete B-spline provide the same values as the continuous basis functions at the node points, as shown in Fig. 2. Interestingly enough, the convolution with $\{b_1^n(k)\}$ provides a weighting scheme of the type used in the Newton–Cotes quadrature formulae for numerical integration by summation [15]. For $n = 2$, we have the simple trapezoidal rule, and for $n = 3$ we have the coefficients for Simpson's rule of integration. The need for such a correction is not so surprising if one recalls that in the discretization process, all convolution integrals are replaced by summations. Another interpretation of (2.10) is that a uniform stretching of $b_1^n(k)$ can be obtained by convolving this function $(n+1)$ times with a moving average filter of size m where m is the expansion factor (cf. Fig. 2).

The Fourier transform of the discrete B-spline of order n is obtained by replacing z with $e^{j2\pi f}$ in (2.6), which after some manipulations, yields

$$|B_m^n(f)| = \frac{1}{m^n} \left| \sum_{k=0}^{n+1} b_1^n(k) e^{-j2\pi fk} \right| \left(\frac{\sin(\pi m f)}{\sin(\pi f)}\right)^{n+1}. \tag{2.11}$$

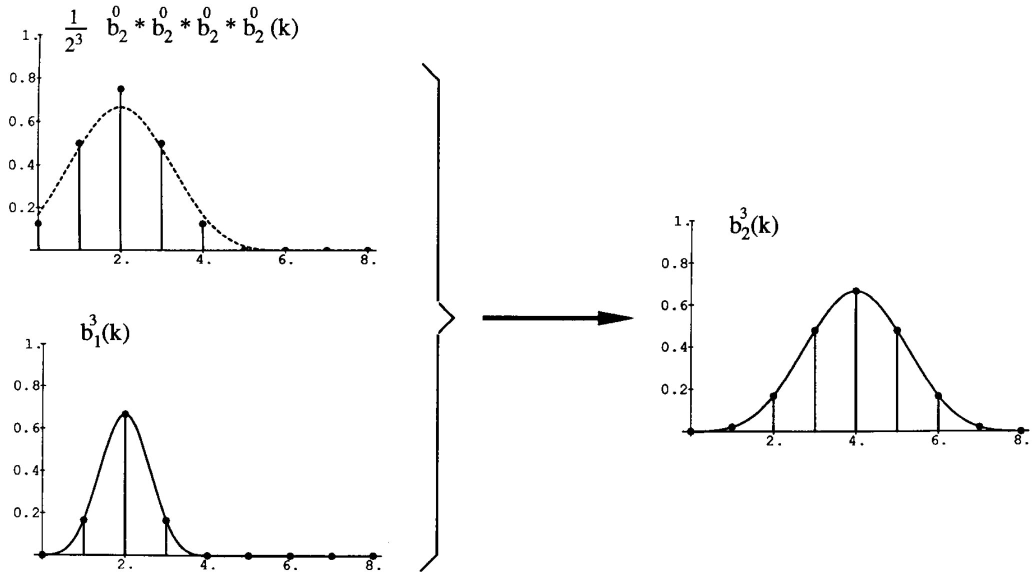

Fig. 2. Convolution property for the discrete cubic B-spline with an up-sampling factor $m = 2$. The corresponding continuous cubic spline functions are superimposed in solid lines when there is a perfect match and in dashed lines otherwise.

TABLE I
z-TRANSFORMS OF POWER SERIES AND DISCRETE B-SPLINES FOR $n = 0$ TO 5

n	$P^n(z) = \sum_{k=0}^{+\infty} k^n z^{-k}$	$A^n(z)$	$B_n^1(z)$
0	$\frac{A^0(z)}{(1-z^{-1})}$	1	1
1	$\frac{A^1(z)}{(1-z^{-1})^2}$	z^{-1}	$\frac{1}{z}$
2	$\frac{A^2(z)}{(1-z^{-1})^3}$	$z^{-1} + z^{-2}$	$\frac{1+z^{-1}}{2z}$
3	$\frac{A^3(z)}{(1-z^{-1})^4}$	$z^{-1} + 4z^{-2} + z^{-3}$	$\frac{z+4+z^{-1}}{6z^2}$
4	$\frac{A^4(z)}{(1-z^{-1})^5}$	$(z^{-1} + z^{-2})(1 + 10z^{-1} + z^{-2})$	$\frac{(1+z^{-1})(z+10+z^{-1})}{24z^2}$
5	$\frac{A^5(z)}{(1-z^{-1})^6}$	$z^{-1} + 26z^{-2} + 66z^{-3} + 26z^{-4} + z^{-5}$	$\frac{z^2+26z+66+26z^{-1}+z^{-2}}{120z^3}$

III. SIGNAL REPRESENTATION AND INTERPOLATION

In this section, we describe fast algorithms for the determination of the expansion coefficients in (2.1) (direct B-spline transform) and for signal reconstruction and interpolation from its spline coefficients (indirect B-spline transform). These algorithms are all based on recursive filtering and will be analyzed in term of their impulse response and transfer function.

A. Direct B-Spline Transform

Let us consider a discrete signal $\{f(k)\}$ defined on $k = -\infty, \cdots, +\infty$. We seek the interpolating function $\phi^n(x)$ of the form (2.1) such that

$$f(k) = \phi^n(k) = \sum_{i=-\infty}^{+\infty} c(i) b_1^n(k-i). \tag{3.1}$$

This expression describes a convolution and can also be written as

$$f(k) = b_1^n * c(k) \tag{3.2}$$

where $b_1^n(k)$ is the finite impulse response of the operator that we shall refer to as the indirect spline filter of order n. By taking the z-transform, we get

$$F(z) = B_1^n(z)C(z) \tag{3.3}$$

where $B_1^n(k)$ is given by (2.8). This result suggests that the spline coefficients $\{c(k)\}$ can be determined simply by inverse filtering. The corresponding linear space invariant operator $\{s^n(k)\}$ is called the direct spline filter of order n. Its transfer function is given by

$$S^n(z) = B_1^n(z)^{-1} = \frac{1}{\sum_{k=0}^{n+1} b_1^n(k) z^{-k}}. \tag{3.4}$$

TABLE II
TRANSFER FUNCTIONS OF SYMMETRIC DIRECT AND BASIC INDIRECT B-SPLINE FILTERS

n	Indirect B-Spline Filter	Direct B-Spline Filter	Poles
0	1	1	–
1	1	1	–
2	$\frac{z+6+z^{-1}}{8}$	$\frac{8}{z+6+z^{-1}}$	$\alpha_1 = \sqrt{8}-3$ $\alpha_1 = 1/\alpha_1$
3	$\frac{z+4+z^{-1}}{6}$	$\frac{6}{z+4+z^{-1}}$	$\alpha_1 = \sqrt{3}-2$ $\alpha_2 = 1/\alpha_1$
4	$\frac{z^2+76z+230+76z^{-1}+z^{-2}}{384}$	$\frac{384}{z^2+76z+230+76z^{-1}+z^{-2}}$	$\alpha_1 = -0.01372$ $\alpha_2 = -0.36134$ $\alpha_3 = 1/\alpha_1,\ \alpha_4 = 1/\alpha_2$
5	$\frac{z^2+26z+66+26z^{-1}+z^{-2}}{120}$	$\frac{120}{z^2+26z+66+26z^{-1}+z^{-2}}$	$\alpha_1 = -0.04309$ $\alpha_2 = -0.43057$ $\alpha_3 = 1/\alpha_1,\ \alpha_4 = 1/\alpha_2$

Clearly, $S^n(z)$ is the response of a recursive infinite impulse response (IIR) filter. It is therefore important to check its stability which amounts to determining the roots of $B_1^n(z)$ and verifying whether or not these lie on the unit circle. By studying the operators given in Table I, we see that only the filters corresponding to odd orders are stable. For n even, we always have a pole at $z = 1$ which makes the present approach unsuitable. We note, however, that this problem is easily overcome by introducing a phase shift of $1/2$ in the definition of the sampled even B-splines, which is needed in the case for the construction of symmetric basis functions. There is no major difficulty in making this modification. For reference we present in Table II the transfer functions of all direct and indirect symmetric B-spline filters up to order five.

The explicit determination of the impluse response of the direct B-spline filter can be obtained by decomposing $S^n(z)$ into simple partial fractions. Such a decomposition can also result in efficient filter realizations by expressing the transfer functions as a sum (or a product) of a causal and anticausal system. These points will be illustrated in Section III-E with the detailed analysis of the direct cubic B-spline filter.

B. Indirect B-Spline Transform

The indirect B-spline transform governs the process of reconstructing discrete signal values from the B-spline representation. In most applications, one is interested in filling in a signal at a higher sampling rate, which is one particular form of interpolation. The indirect transformation or reconstruction method presented here uses an integral up-sampling factor m. The requirement of m being an integer is not a major limitation since any rational sampling rate can be obtained from a succession of integral interpolations and decimations, although this is not likely to be the most computationally efficient technique.

Signal reconstruction or interpolation with an up-sampling factor m amounts to determining the set of discrete values:

$$f_m(k') = \phi\left(\frac{k'}{m}\right) = \sum_{i=-\infty}^{+\infty} c(i) b_m^n(k' - im). \qquad (3.5)$$

This operation can also be interpreted as a magnification with a zooming or expansion factor m. A basic operation is the up-sampling of a signal $\{f(k)\}$ by a factor m which produces the new sequence $\{[f]_{\uparrow m}(k')\}$ defined as

$$[f]_{\uparrow m}(k') = \begin{cases} f(k) & \text{for } k' = mk \\ 0 & \text{otherwise} \end{cases} \leftrightarrow F(z^m). \qquad (3.6)$$

By substituting $[c]_{\uparrow m}(k')$ in (3.5), we get the convolution equation:

$$f_m(k') = b_m^n * [c]_{\uparrow m}(k'). \qquad (3.7)$$

Taking the z-transform and using (2.6) we find

$$F_m(z) = B_1^n(z) \frac{1}{m^n} \left(B_m^0(z)\right)^{n+1} C(z^m) \qquad (3.8)$$

which clearly indicates that signal interpolation can be achieved from a cascade of $n + 1$ moving average filters of size m and an indirect spline filter $\{b_1^n(k)\}$. When $m = 1$ this operation is the inverse of the procedure described in Section III-A.

These results are also applicable to B-splines centered on the origin (cf. Table II) provided that the appropriate phase shifts are introduced. The easiest way to produce a global response that is symmetric is to use individual convolution operators that are also symmetric. The only difficulty arises when m is even, in which case we can combine moving averaging filters in pairs to produce a globally symmetric response (e.g., one moving average filter is centered on $m/2$ while the other is centered on $(m/2) + 1$).

C. Global System Analysis

The block diagram of the global system performing this whole sequence of operations is shown in Fig. 3. The spline coefficients are obtained by prefiltering (direct B-spline transform). The next three operations (up-sampling by a factor m, a multiplication by m, and a moving average filter of length m) are in fact equivalent to performing a 0 order signal interpolation, which amounts to simply repeating the current sample value m times. The signal is then smoothed by iterating a moving average filter $\left(M_m(z) = B_m^0(z)/m\right)$ n times. The final operation is the postfiltering (basic indirect B-spline transform).

To facilitate the analysis, it is convenient to consider an equivalent system where the initial input signal as well as the impulse

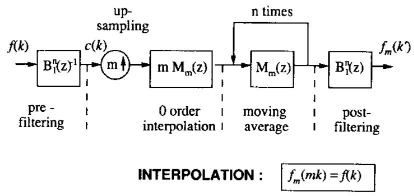

Fig. 3. Block diagram of an nth order B-spline interpolator with an upsampling factor m.

response of the direct spline filter are up-sampled by a factor m. The whole succession of operations is then described by the convolution equation

$$f_m(k') = b_m^n * [s^n]_{\uparrow m} * [f]_{\uparrow m}(k') = h_m^n * [f]_{\uparrow m}(k') \qquad (3.9)$$

where global impulse response of the system $h_m^n(k') = b_m^n * [s^n]_{\uparrow m}(k')$ is the sampled cardinal spline of order n (cf. Section III-E). The global transfer function is given by

$$H_m^n(z) = \frac{B_m^n(z)}{B_1^n(z^m)} = \frac{1}{m^n}\frac{B_1^n(z)\left(B_m^0(z)\right)^{n+1}}{B_1^n(z^m)}. \qquad (3.10)$$

The system may also be characterized by its frequency response which is given by

$$|H_m^n(f)| = \frac{1}{m^n}\frac{\left|\sum_{k=0}^{n+1} b_1^n(k)e^{-j2\pi f}\right|}{\left|\sum_{k=0}^{n+1} b_1^n(k)e^{-j2m\pi f}\right|}\left(\frac{\sin(\pi m f)}{\sin(\pi f)}\right)^{n+1}. \qquad (3.11)$$

As far as signal reconstruction is concerned, this filter should provide a reasonable approximation of a perfect lowpass filter which, according to Shannon's sampling theorem, is required for perfect interpolation of a bandlimited signal.

It is interesting at this stage to investigate the computational complexity of the system that we are proposing. The direct B-spline filter has $2[n/2] + 1$ coefficients[1] and can be implemented recursively with a computational cost of $2[n/2]$ adds and $2[n/2]$ multiplies per sample value, as illustrated for $n = 3$ in Section III-E. This part of the algorithm should be performed using real (fixed or floating point) arithmetics. The signal reconstruction, on the other hand, can be evaluated using integer operations exclusively. In our system, the moving average filter is implemented recursively using a simple update strategy (e.g., two additions per moving sum) [16], and the final normalization is only performed once. Accordingly, the computational cost of the interpolation step is $(3n + 1)$ adds and $2[n/2] + 1$ multiplies per reconstructed sample point.

D. B-Splines in Higher Dimensions

Although all our results were derived for one-dimensional signals, they are directly applicable to higher-dimensions through the use of tensor product polynomial splines [12]. The corresponding basis functions are obtained from the product of one-dimensional splines defined for each index variable. Since all basis functions are separable, the corresponding linear direct and indirect transformations are also separable [1]. This implies that the spline coefficients can be determined by successive one-dimensional direct B-spline filtering along the coordinates. The same strategy is also applicable for signal reconstruction or interpolation by indirect spline filtering.

E. Cubic B-Spline Filter

To illustrate these results, we consider the design of the direct cubic B-spline filter. The z-transform of the sampled symmetric cubic B-spline is

$$B_1^3(z) = \frac{z + 4 + z^{-1}}{6}. \qquad (3.12)$$

The transfer function of the direct B-spline filter (the inverse filter) can be factorized as

$$S^3(z) = \frac{6}{z + 4 + z^{-1}} = \frac{-6\alpha}{(1 - \alpha z^{-1})(1 - \alpha z)} \qquad (3.13)$$

where $\alpha = \sqrt{3} - 2$ is the smallest (in absolute value) root of the polynomial $z^2 + 4z + 1$; the other root is $1/\alpha$ due to the reverse symmetry of the coefficients. $S^3(z)$ is the transfer function of a stable symmetric infinite impulse response filter and is further decomposed as

$$S^3(z) = \frac{-6\alpha}{(1 - \alpha^2)}\left(\frac{1}{(1 - \alpha z^{-1})} + \frac{1}{(1 - \alpha z)} - 1\right) \qquad (3.14)$$

where the term with z^{-1} (resp. z) in the denominator is the transfer function of a simple first order causal (resp. anticausal) filter. It follows that the impulse response is given by

$$s^3(k) = \frac{-6\alpha}{(1 - \alpha^2)}\alpha^{|k|} \qquad (3.15)$$

which is a symmetric decreasing exponential with an alternating sign change. This filter is implemented efficiently using either (3.13) or (3.14). In the latter case, the recursive filter equations are

$$\begin{cases} c^+(k) = f(k) + b_1 c^+(k-1), & (k = 2, \cdots, K) \\ c^-(k) = f(k) + b_1 c^-(k+1), & (k = K-1, \cdots, 1) \\ c(k) = b_0\left(c^+(k) + c^-(k) - f(k)\right) & \end{cases} \qquad (3.16)$$

where $b_0 = -6\alpha/(1 - \alpha^2)$ and $b_1 = \alpha = -0.2679$. In addition, we have to impose some boundary conditions. For practical convenience and to avoid discontinuities, we have chosen to extend the signal by its mirror image. In this circumstance, we have the initial values

$$\begin{cases} c^+(1) = \sum_{k=1}^{k_0} \alpha^{|k-1|} f(k) \\ c^-(K) = c^+(K) \end{cases} \qquad (3.17)$$

where k_0 is such that $\alpha^{|k_0|}$ is below some prescribed level of precision. Obviously, the indirect B-spline filter should use the same type of boundary conditions to insure that the procedure is reversible. We note that with these particular boundary conditions, we are indirectly imposing a zero first order derivative at the end points.

A slightly more economical alternative is an implementation based on the product decomposition (3.13):

$$\begin{cases} d^+(k) = f'(k) - \alpha d^+(k-1), & (k = 2, \cdots, K) \\ c(K) = \frac{-\alpha}{1-\alpha^2}\left(2d^+(K) - f'(K)\right) & \\ c(k) = \alpha\left(c(k+1) - d^+(k+1)\right), & (k = K-1, \cdots, 1) \end{cases} \qquad (3.18)$$

with $f'(k) = 6f(k)$ and $d^+(1) = 6c^+(1)$. The second equation is borrowed from the sum decomposition and is required to obtain a correct initialization of the backward recursion. In this case, the number of operations is two floating point additions and two multiplications per sample point. An additional integer multiplication is also necessary for proper scaling. We note, however, that scaling can either be avoided through the use of nonnormalized basis functions, or at least performed only once in a separable multidimensional implementation. In terms of complexity, this approach has to be compared with more standard numerical analysis methods for determining cubic splines based on the solution of a corresponding tridiagonal system of equations [11], [17]. An efficient solution through Gaussian elimination or LU factorization requires at least four additions and four multiplications per sample point, as well as some additional intermediate storage for the matrix coefficients [17], [18, p. 156].

[1] $[x]$ denotes the integer truncation of x.

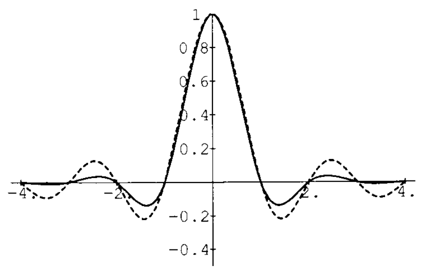

Fig. 4. Comparison of the cardinal cubic spline function (solid line) and ideal sinc interpolation kernel (dashed line).

Cubic B-spline interpolation is best understood by expressing the interpolating function in terms of the initial sample values $\{f(k)\}$:

$$\phi^n(x) = \sum_{i=-\infty}^{+\infty} c(i)\beta^3(x-i) = \sum_{k=-\infty}^{+\infty} f(k)\eta^3(x-k) \qquad (3.19)$$

where $\eta^3(x)$ is the global continuous impulse response of the system; Lancaster and Salkauskas have referred to this function as the cubic cardinal spline [19]. By recalling that $c(k) = s^3 * f(k)$ and using (3.15), it is rather straightforward to show that

$$\eta^3(x) = \frac{-6\alpha}{(1-\alpha^2)} \sum_{k=-\infty}^{+\infty} \alpha^{|k|}\beta^3(x-k). \qquad (3.20)$$

This function is shown in Fig. 4 and appears to be quite similar to a sinc function that has been superimposed in dotted lines. The essential interpolation property of these functions stems from the fact that they are equal to zero at all the knot points except the origin. Clearly, the cubic cardinal spline tends to vanish more rapidly indicating that cubic B-spline interpolation is less influenced by distant data points.

We have implemented these algorithms in Fortran on a personal computer for biomedical image processing applications. Fig. 5 illustrates the use of a separable direct bicubic spline filter on a standard image. This operator computes the direct B-spline transform which provides the coefficients for the bicubic spline functions centered on each spatial location. The transform is slightly sharper than the initial image because, as we have noted, the filter enhances high frequency components. Thanks to the efficiency of the recursive algorithm, the amount of computation is low. On our low-end workstation (standard 16 Mhz Apple Macintosh II), the direct spline filtering of a 256×256 image is performed in fewer than 8 s and the indirect filtering in fewer than 5 s. For comparison, the CPU time of three moving average iterations, which are required for image magnification by bicubic spline interpolation, is of the order of 5 s. We have tested the reversibility of the transformation and found the absolute difference between initial and final pixel values to be less than two gray level intensity values (full range 0–255), although the processing results are truncated and stored in integer or byte arrays. We note, however, that we have used an auxiliary one-dimensional real array to store the intermediate 1-D recursive filtering results. We have also tried to perform the normalization at the very end of the computation.

IV. Comparative Evaluation

A. Results

In Fig. 6, we show the result of several interpolation techniques applied to the magnification of a detail of the standard girl image with a zooming factor 8. The first two methods are the simple nearest neighbor [Fig. 6(a)] and bilinear [Fig. 6(b)] interpolations, which are equivalent to 0 and 1 order B-spline interpolation, respectively. The other two images are obtained with our bicubic spline interpolation

Fig. 5. Direct cubic B-spline transform by recursive filtering. (a) Initial 208×222 girl image, (b) cubic B-spline transform.

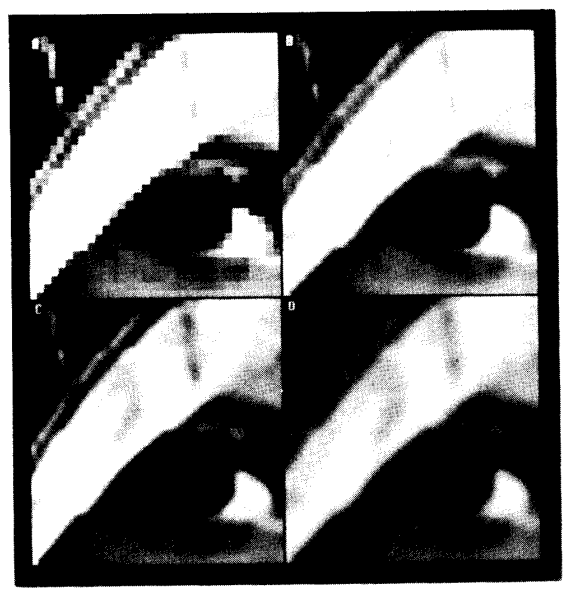

Fig. 6. Comparison of interpolation algorithms for the magnification with a zooming factor 8 of a 32×32 detail of the standard girl image. (a) 0 order interpolation, (b) bilinear interpolation, (c) bicubic spline interpolation, (d) smoothing by indirect bicubic spline transform.

[Fig. 6(c)], and with the reconstruction part of the algorithm applied to the input image directly [Fig. 6(d)] (indirect transform only).

The four image interpolations were displayed to a group of subjects (adult male staff members of BEIP) to assess their quality. No special pains were made to establish a consistent background or light level. The subjects were placed 2.5 meters from a page of white paper containing all four of the images (Fig. 6). The actual size of the images were 11 cm. The viewing distance was chosen so that the visual angle of each image (about 2.5°) was about the same as that of the same scene (a segment of the girl's eye) when the "girl" picture is viewed at a distance of about 30 cm. The subjects were asked to rank order the images in descending order of being "lifelike."

The results for 20 subjects were unambiguous. All but one rated image *C* first. The subsequent ranks in descending order were *B* (18 out of 20), *D* (14 out of 20), and *A* (14 out of 20). These rankings were remarkably consistent. Several subjects pointed out that *B* was exhibiting some noticeable striking artifacts along the image contours, especially for diagonal lines. *D* was usually described as being blurred with a loss of some details.

However, it should be noted that viewing distance is crucial. A subset of subjects were shown the same set of images at a distance

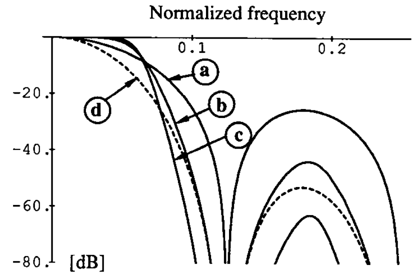

Fig. 7. Frequency responses of various discrete interpolation kernels (zooming factor 8). (a) First order interpolation, (b) cubic spline interpolation, (c) quintic spline interpolation, (d) cubic spline smoothing.

of about 6 meters. In every case they commented that image A was sharper. It seems plausible that at the shorter distance the quantization edges of image A are very distracting. Of the other three, it would appear that quality falls off with decreasing energy at high spatial frequency.

The global vertical or horizontal frequency responses of the corresponding interpolation operators are shown in Fig. 7. These were evaluated from (3.11) (and (2.11) for the cubic spline smoother) and correspond to the Fourier transform of the kernels obtained by sampling the following functions: 1) cardinal spline (or B-spline) of order 1 [Fig. 1(b)], 2) cubic cardinal spline (Fig. 4), 3) cubic B-spline [Fig. 1(d)]. The response of the quintic B-spline filter has also been included to illustrate the tendency of B-spline interpolators to more closely approximate an ideal lowpass filter as n increases: the attenuation in the stop band is increased by approximately 20 dB with each increment while the response tends to become more nearly uniform in the bandpass region. On the other hand, the indirect cubic B-spline filter used on its own produces the greatest amount of smoothing in the pass zone, although its stop band performance is relatively good. This observation stresses the importance of prefiltering (direct spline transform) to counterbalance the smoothing effect of the indirect spline filter.

B. The Sankar and Ferrari Approach

Sankar and Ferrari have recently proposed what they claim to be an efficient algorithm for computing B-spline image representation and interpolation [10]. The main feature of their algorithm is that it uses additions only, which is a rather intriguing result. We have analyzed their system in detail and note that it does not guarantee precise signal reconstruction at the knot points.

The discrete basis functions used by Sankar *et al.* are wider than the usual ones by a factor p and are different from those described in Section II by reason of the omission of the correction kernel $b_1^n(k)$ in the convolution. It follows immediately that they do not even match the continuous B-splines at the knot points. Sankar *et al.*'s Toeplitz inverter is in fact equivalent to the digital filter with the transfer function

$$S'(z) = \frac{1}{B_p^0(z)^{n+1}} = \left(\frac{1 - z^{-1}}{1 - z^{-p}}\right)^{n+1}. \tag{4.1}$$

Their reconstruction algorithm with a zooming factor m is obtained from a sequence of $(n + 1)$ moving average filters of size $(p \times m)$. Therefore, the impulse response of the global system, which should be compared to (3.11), is

$$\begin{aligned} H'(z) &= c_n S'(z^m)\left(B_{pm}^0(z)\right)^{n+1} \\ &= c_n \left(\frac{1 - z^{-m}}{1 - z^{-mp}}\right)^{n+1} \left(\frac{1 - z^{-mp}}{1 - z^{-1}}\right)^{n+1} \end{aligned}$$

where c_n is some constant normalization factor. Obviously, the terms in $\left(1 - z^{-mp}\right)^{n+1}$ cancel and the global response simplifies to

$$H'(z) = c_n\left(\frac{1 - z^{-m}}{1 - z^{-1}}\right)^{n+1}. \tag{4.2}$$

It follows that the system that they are describing is simply performing $n + 1$ moving averages of size m, a procedure that can be carried out more directly. These authors are obviously correct when they conclude that their cubic B-spline filter induces blur in the resulting interpolated image. Clearly, this approach does not preserve the initial signal values at the knot points and does therefore not qualify as a valid interpolation algorithm. Incidentally, this method is very similar to that of applying our indirect B-spline transform to the initial signal values instead of applying it to the B-spline coefficients [cf. Fig. 6(d)].

C. Discussion

We believe our results show that a higher order B-spline interpolation correctly applied does not produce increased signal blurring. This is a point that has been overlooked by some authors and has been treated incorrectly in several text books or papers [1], [2]. It appears that the higher the order of the spline interpolator, the better it approximates an ideal reconstruction filter (sinc). Increasing n has the effect of amplifying the frequency attenuation in the stop band, which reduces aliasing. At the same time, as a result of prefiltering (direct B-spline transform), the frequency is flattened in the bandpass region which appears to minimize the loss of resolution.

Our reason for including the cubic spline smoothing method [Fig. 6(d)] in the comparison is that this particular approach is often referred to, incorrectly, as the bicubic spline interpolation [1], [2]. In the early paper by Hou and Andrews [6], this technique was proposed as an alternative to the full procedure, mainly because these authors did not have an efficient way of determining the spline coefficients. They were relying on a method using matrix multiplication and requiring the explicit inversion of a tridiagonal Toeplitz matrix, which turns out to be a rather time consuming task. In their comparison of various interpolating methods, Parker and Kenyon have mistakenly used the frequency response of the cubic B-spline [cf. Fig. 7(d)] to characterize the performance of the cubic B-spline interpolator [2]. Consequently, they conclude incorrectly that B-spline interpolation induces excessive smoothing and that therefore other cubic convolution schemes [7], [8] should be preferred. Obviously, their reasoning applies only to the method used to produce Fig. 6(d) and not 6(c). A correct analysis should consider the cardinal cubic B-spline (cf. Fig. 4) as the interpolation kernel and base its conclusions on the examination of the global frequency response of the system [Fig. 7(b)].

To the best of our knowledge, the present recursive filtering approach for the direct B-spline transform has not been described before. It is essentially this part of the algorithm that makes the use of B-spline interpolation practical for image processing. From this point of view, our algorithm is a substantial improvement in efficiency over the method of Hou and Andrews which relies on an explicit matrix multiplication for determining the cubic spline coefficients. Their reconstruction algorithm is essentially the same as ours, although they limited their considerations to the case of cubic splines. In Section III-E, we have also shown that the recursive algorithm for the cubic spline coefficients (3.18) offers a reduction of the computational complexity of approximately a factor of two over a carefully designed matrix approach using Gaussian elimination or LU factorization to efficiently solve the corresponding system of tridiagonal equations. For higher order splines, the gain of the present approach is even greater: $4[n/2]$ flops/sample for the recursive filtering approach, as opposed to $2n[n/2]^2$ flops/sample for the LU factorization alone plus $8[n/2]$ flops/sample for the subsequent solution of the banded system of equations using forward and backward substitution [18, pp. 150–151.]

Recent work by Toraichi *et al.* [3] provides a detailed analysis of a two-dimensional quadratic B-spline interpolator that is consistent with our results. The major difference is that they combine the coefficient estimation and interpolation tasks. They approximate the resulting interpolation kernel by a finite impulse response filter, which is computationally less satisfactory than the present approach.

The basic design technique described in Section III-E is directly applicable to the direct symmetric quadratic spline filter (cf Table II) by simply replacing α by $\sqrt{8} - 3$. It can also be used with minor modification for higher order splines, the only difference being that simple exponential impulse responses are replaced by weighted sums of exponentials. As the coefficients of the characteristic filter polynomial are all positive, the roots (if they are real) will always be negative. It follows that the cardinal spline of order n will always have an alternation of positive and negative cycles with a decaying amplitude, which establishes its similarity to a sinc function. From Table II, it is also clear that the computational complexity of the direct spline filters of order $2n$ and $2n + 1$ are essentially the same. It is therefore more advantageous from a practical point of view to use odd order splines that provide better interpolation performance for approximately the same computational cost (the overhead is one additional moving average iteration for image reconstruction when $m > 1$).

From our results, we conjecture that the cardinal spline of order n should approach an ideal interpolation sinc function as n tends to infinity. This proposition, however, remains to be demonstrated. Conversely, if there is a B-spline interpolation equivalent to the ideal interpolation procedure, it must be a spline of "infinite" order. This simply follows from the fact that an ideally reconstructed signal is bandlimited, which also implies that the function is analytical. Consequently, all its derivatives must be continuous everywhere. Clearly, this last condition is satisfied only for a spline of "infinite" order.

V. Conclusion

In this correspondence, we have described fast algorithms for the evaluation of the direct and indirect (with an upsampling factor m) B-spline transforms. The main result is that these transformations can be expressed as filtering operations that are most efficiently implemented recursively. The major advantage of these algorithms is their simplicity and their ease of implementation. It should make them attractive in a variety of image processing applications.

Our analysis shows, contrary to the claims of several authors, that B-spline interpolation does not result in a loss of resolution. Generally, increasing the order of the spline improves the quality of the interpolation in the sense that the resulting filter more closely approximates the ideal (sinc) interpolation function. However, the performance of the cubic spline interpolator, which guarantees the continuity of the function up to its second order derivative, seems to be sufficient for most practical applications.

Although we have considered the use of B-splines in the context of image interpolation, there are other potential applications. Dealing with a continuous image representation may facilitate the conception of other image processing algorithms. These could be designed to operate directly in the space of the B-spline coefficients. It is for example relatively easy to compute quantities such as gradients, Laplacians, or directional derivations, which are usually not well defined for discrete images, but that are very useful for tasks such as edge detection. B-splines also provide a simple way of translating many of the concepts of differential geometry for the study of digital images.

References

[1] W. K. Pratt, *Digital Image Processing.* New York: Wiley, 1978.
[2] J. A. Parker, R. V. Kenyon, and D. E. Troxel, "Comparison of interpolating methods for image resampling," *IEEE Trans. Med. Imaging,* vol. MI-2, pp. 31–39, 1983.
[3] K. Toraichi, S. Yang, M. Kamada, and R. Mori, "Two-dimensional spline interpolation for image reconstruction," *Pattern Recog.,* vol. 21, pp. 275–284, 1988.
[4] P. Van Wie and M. Stein, "A Landsat digital image rectification system," *IEEE Trans. Geosci. Electron.,* vol. 15, pp. 130–137, 1977.
[5] M. Unser, B. L. Trus, and M. Eden, "Iterative restoration of noisy elastically distorted quasi-periodic images," *Signal Processing,* vol. 17, pp. 191–200, July 1989.
[6] H. S. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-26, pp. 508–517, 1978.
[7] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-29, pp. 1153–1160, 1981.
[8] S. K. Park and R. A. Showengerdt, "Image reconstruction by parametric convolution," *Comput. Vision, Graphics, Image Processing,* vol. 20, pp. 258–272, Sept. 1983.
[9] T. C. Chen and R. J. P. deFigueiredo, "Two-dimensional interpolation by generalized spline filters based on partial differential equations," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-33, pp. 631–642, June 1985.
[10] P. V. Sankar and L. A. Ferrari, "Simple algorithms and architectures for B-spline interpolation," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 10, pp. 271–276, 1988.
[11] C. de Boor, *A Practical Guide to Splines.* New York: Springer-Verlag, 1978.
[12] L. L. Schumaker, *Spline Functions: Basic Theory.* New York: Wiley, 1981.
[13] I. J. Schoenberg, "Contribution to the problem of approximation of equidistant data by analytic functions," *Quart. Appl. Math.,* vol. 4, pp. 45–99, 112–141, 1946.
[14] ——, "Cardinal interpolation and spline functions," *J. Approximation Theory,* vol. 2, pp. 167–206, 1969.
[15] G. A. Korn and T. M. Korn, *Mathematical Handbook for Scientists and Engineers.* New York: McGraw-Hill, 1968.
[16] M. Unser, "Recursion in short time signal analysis," *Signal Processing,* vol. 5, pp. 229–240, May 1983.
[17] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes.* Cambridge, England: Cambridge University Press, 1986.
[18] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, MD: Johns Hopkins University Press, 1989.
[19] P. Lancaster and K. Salkauskas, *Curve and Surface Fitting: An Introduction.* London: Academic, 1986.

Small Sample Error Rate Estimation for k-NN Classifiers

Sholom M. Weiss

Abstract— **Small sample error rate estimators for nearest neighbor classifiers are reexamined and contrasted with the same estimators for three-nearest neighbor classifiers. The performance of the bootstrap estimators, e0 and 0.632B, is considered relative to leaving-one-out and other cross-validation estimators. Monte Carlo simulations are used to measure the performance of the error rate estimators. The experimental**

Manuscript received February 18, 1989; revised September 9, 1990. This work was supported in part by the National Institutes of Health under Grant P41-RR02230 and by the Office of Naval Research under Contract N00014-87-K-0398.

The author is with the Department of Computer Science, Rutgers University, New Brunswick, NJ 08903.

IEEE Log Number 9040950.