Cubic Splines for Image Interpolation and Digital Filtering

HSIEH S. HOU, MEMBER, IEEE, AND HARRY C. ANDREWS, SENIOR MEMBER, IEEE

Abstract–This paper presents the use of *B*-splines as a tool in various digital signal processing applications. The theory of *B*-splines is briefly reviewed, followed by discussions on *B*-spline interpolation and *B*-spline filtering. Computer implementation using both an efficient software viewpoint and a hardware method are discussed. Finally, experimental results are presented for illustrative purposes in two-dimensional image format. Applications to image and signal processing include interpolation, smoothing, filtering, enlargement, and reduction.

I. Introduction

IN RELATION to the many applications of interpolation in signal processing (see [1]), the need for sampling rate adaption constantly arises in image processing. Examples of such applications are image resolution conversion and image change of scale. Rigorously speaking, the process of decreasing the data rate is called decimation and increasing the data samples is termed interpolation. However, in the following discussions we have not attempted to distinguish these two operations, nor the difference between resolution conversion and image scaling because they all can be treated by the same mathematical analysis. Conceptually, the resolution conversion process can be regarded as a two-step operation. First, the discrete data is reconstructed (or interpolated) into a continuous curve, then it is sampled at a different sampling rate. This is shown in Fig. 1. Nevertheless, the above steps are only a mental picture for illustrating the underlying principle. In real digital processing, the procedure of reconstruction by interpolation and sampling at a different rate can be done in one operation. (There is never a continuous curve existing inside a digital processor.)

In this paper we have focused our attention on the first box in Fig. 1, i.e., the interpolation process. In principle we are seeking a smooth continuous curve passing through a set of discrete data at certain given spatial points. Mathematically speaking, the interpolated continuous function in one dimension is

$$\hat{f}(\xi) = \sum_{k=1}^{K} c_k s_k(\xi) \tag{1}$$

where c_k are the coefficients to be determined from the input data, $s_k(\xi)$ are the chosen basis functions, and K is the number of given data points.

Even though the use of spline interpolation [2]-[12] for engineering applications has gained recent momentum, results have been widely spread amongst various branches of electrical engineering. The most popular ones, just to mention a few, are to estimate probability density functions [13]-[15] and power spectral density functions [16], and to interpolate noisy data [17]-[20]. Schafer and Rabiner [1] have discussed sampling and interpolation operations from a more general digital signal processing point of view, in which the spline interpolation has not been mentioned. Ostrander [21] has briefly discussed and demonstrated the use of spline transforms for correcting the discrete Fourier transformed data. Caprihan [22] has used cubic natural splines to design finite-duration filters. Furthermore, Hou and Andrews have adopted spline interpolation for image restoration [23].

In this paper we first concentrate our discussion on how to choose an optimum and yet easy to implement basis function for interpolation. Such a function is the so-called cubic *B*-spline which can give smooth interpolation for the given discrete data. The theory of spline interpolation and data smoothing using spline filtering is then discussed. This theory has been implemented by both software and hardware for image processing applications. Both methods are iterative procedures which are particularly suited for digital design. Finally, results from computer simulation with application to image magnification and minification are presented for comparison with those obtained by using other interpolation schemes.

II. Basic Concept of Spline Interpolation

From a numerical analysis point of view, the classical polynomial interpolation approaches [1], e.g., Lagrange interpolation [24], at an increasing set of data points all involve the use of polynomials of increasingly higher degree. That approach has several severe limitations. First, we cannot guarantee that a sequence of Lagrange interpolations to a continuous function $f(\xi)$ will converge uniformly to $f(\xi)$. In fact, for any sequence of sets of interpolation points, there exists a continuous function $f(\xi)$ such that the sequence of Lagrange interpolations to $f(\xi)$ at these points diverges. Second, while the sequence of interpolations may in fact converge to $f(\xi)$, approximating $f'(\xi)$ by the derivatives of its interpolation can be extremely inaccurate. These problems can be intuitively linked to two facts concerning polynomial interpolation. First, polynomials have a notorious ability to "wiggle," that is, pinning a

Manuscript received May 10, 1976; revised December 27, 1977 and July 11, 1978. This work was partially supported by the Advanced Research Projects Agency of the Department of Defense under Contract F-33615-76-C-1203, ARPA Order 3119, and monitored by Wright Patterson Air Force Base, Dayton, OH.

H. S. Hou is with the Xerox Corporation, El Segundo, CA 90245.

H. C. Andrews is with the Department of Electrical Engineering and Computer Science, University of Southern California, Los Angeles, CA 90007.

0096-3518/78/1200-0508$00.75 © 1978 IEEE

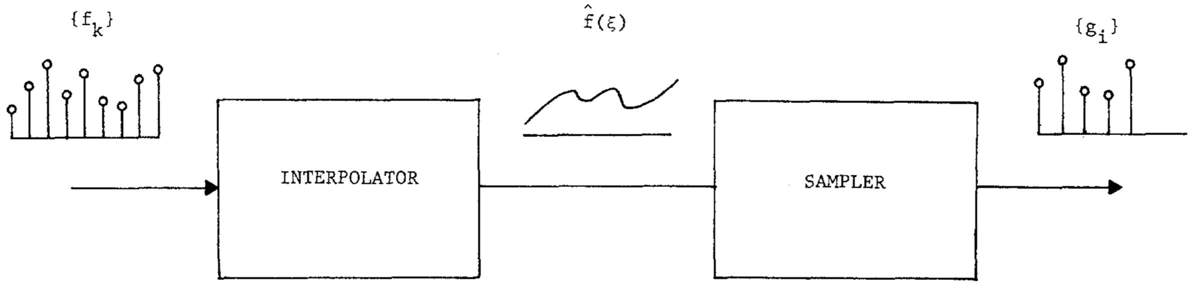


Fig. 1. The resolution conversion process.

polynomial down at a few points for a slowly varying function may not produce, in any sense, a good uniform approximation to the function or its derivatives. This is based on a fundamental property that an nth degree polynomial must have n roots. Second, polynomials are analytic functions. As such, their behavior everywhere is totally determined by their behavior in any interval. Thus, polynomial interpolation is in no sense a "local" procedure. That is, if the function to be interpolated varies rapidly in some part of the region of interest, the effect of this on the interpolation would be felt everywhere.

On the other hand, from the sampling theorem [25] one may attempt to use the Cardinal spline [26] as the basis functions, i.e., let

$$s_k(\xi) = \frac{\sin 2\pi\Omega(\xi - \xi_k)}{2\pi(\xi - \xi_k)} \equiv \text{sinc}\,(\xi - \xi_k) \tag{2}$$

where Ω is the one-sided bandwidth of $f(\xi)$. And one then concludes that from the sampling theorem

$$\hat{f}(\xi) = \sum_{k=-\infty}^{\infty} f_k s_k(\xi) \tag{3}$$

is a perfect reconstruction of $f(\xi)$ if it was originally sampled at or above the Nyquist rate. However, there are many difficulties in doing this. First, most natural images are not diffraction limited. Second, the Cardinal spline, though being analytic, behaves like an infinite degree polynomial whose supports are not local, which poses a computational problem. If truncations are made on the upper and lower limits of the summation in (3), oscillations known as Gibb's phenomenon will show up in $\hat{f}(\xi)$. Third, the interpolation formula in (3) has implicitly imposed a restriction that the discrete data $\{f_k\}$ must be equally spaced.

These above considerations lead us rather naturally to the idea of interpolating a function by piecewise polynomials, i.e., by analytic functions which are a piecewise polynomial of fixed degree. The whole class of piecewise polynomials are called splines. The spline interpolation not only alleviates the difficulties, as we have mentioned previously, suffered by the classical polynomial approach, but also minimizes the least squares errors of the desired function values and its derivatives at the interpolation points. In other words, among the many interpolating functions passing through the data points only the spline interpolation gives the smoothest,[1] which is also the best (in a least squares sense) approximation.

III. Properties of Spline Basis Functions

In this paper we are particularly interested in the B-spline functions [4]-[12] because they are smooth and span a finite number of data points, i.e., their support is local. Because of this property we can use them as basis functions in the interpolation formula in (1). We shall define them mathematically in the following paragraph.

Definition: Let π: $\xi_0 < \xi_1 < \cdots < \xi_n < \xi_{n+1}$ be a partition of the interval $[\xi_0, \xi_{n+1}]$ on a real axis. A B-spline of degree n on π is, by definition, the following piecewise polynomial:

$$B_n(\xi; \xi_0, \xi_1, \xi_2, \cdots, \xi_{n+1}) = (n+1) \sum_{k=0}^{n+1} \frac{(\xi - \xi_k)^n\, U(\xi - \xi_k)}{\omega(\xi_k)} \tag{4}$$

where

$$\omega(\xi_k) = \prod_{\substack{j=0 \\ j \neq k}}^{n+1} (\xi_k - \xi_j)$$

$$U(\xi - \xi_k) = \begin{Bmatrix} (\xi - \xi_k)^0 & \text{for } \xi > \xi_k \\ 0 & \text{for } \xi \leqq \xi_k \end{Bmatrix},$$

i.e., a unit step function

and

$$n = 0, 1, 2, \cdots.$$

A sketch of the first four lower order B-splines for the uniformly spaced data points is shown in Fig. 2, where $\Delta = \xi_k - \xi_{k-1}$. Furthermore, for the uniformly spaced data points (knots), one can show from (4) that $B_1 = B_0 * B_0$, $B_2 = B_0 * B_0 * B_0$, $B_3 = B_0 * B_0 * B_0 * B_0, \cdots$, where $*$ denotes convolution.

Evidently the B_0 is a sample and hold function such that the interpolation becomes replication with possible discontinuities at the knots. The interpolation by B_1 becomes piecewise straight line connections between the knots. Likewise the interpolation by B_2 is a graph composed of a sequence of parabolas which join at the knots continuously together with their slopes. Finally, the interpolation by B_3 is composed of a sequence of third degree piecewise polynomials which join at the knots continuously together with their slopes and curvatures.[2]

Now it becomes obvious that interpolation by using B_0 and

[1] Here the term "smoothest" means that the norm of the $(n-1)$th derivative of the nth order spline between the data points is the smallest.

[2] As a reminder, the curvature of a plane curve is defined as $f''(\xi)/\{1 + [f'(\xi)]^2\}^{3/2}$.

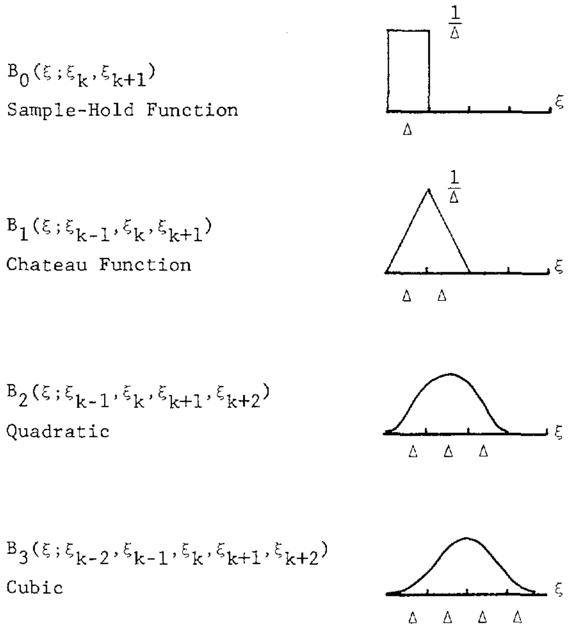

$B_0(\xi;\xi_k,\xi_{k+1})$
Sample-Hold Function
$\frac{1}{\Delta}$
Δ
ξ
$B_1(\xi;\xi_{k-1},\xi_k,\xi_{k+1})$
Chateau Function
$\frac{1}{\Delta}$
Δ Δ
ξ
$B_2(\xi;\xi_{k-1},\xi_k,\xi_{k+1},\xi_{k+2})$
Quadratic
Δ Δ Δ
ξ
$B_3(\xi;\xi_{k-2},\xi_{k-1},\xi_k,\xi_{k+1},\xi_{k+2})$
Cubic
Δ Δ Δ Δ
ξ

Fig. 2. *B*-spline functions.

B_1 does not yield satisfactory results. On the other hand, when the order of spline increases beyond three, it spans more knots. In this case it behaves more like normal polynomial interpolation and the local basis advantage evaporates. Therefore, from a smooth interpolation and easy implementation point of view, the cubic spline is a good choice for the basis function. The cubic *B*-spline being piecewise analytic and spanning five knots is very flexible. Furthermore, it has the advantage of offering good approximation of the function values and its first and second derivatives at the knots. For most engineering interpolation problems, this is a good approximation. Therefore, in the rest of this paper we only discuss cubic *B*-spline functions.

To be explicit, from (4) the cubic *B*-spline for uniformly spaced knots is given by

$$\begin{aligned} s(\xi - \xi_k) &\equiv B_3(\xi; \xi_{k-2}, \xi_{k-1}, \xi_k, \xi_{k+1}, \xi_{k+2}) \\ &= [(\xi - \xi_{k-2})^3\, U(\xi - \xi_{k-2}) - 4(\xi - \xi_{k-1})^3 \\ &\quad \cdot U(\xi - \xi_{k-1}) + 6(\xi - \xi_k)^3\, U(\xi - \xi_k) \\ &\quad - 4(\xi - \xi_{k+1})^3\, U(\xi - \xi_{k+1}) \\ &\quad + (\xi - \xi_{k+2})^3\, U(\xi - \xi_{k+2})]\, \frac{1}{6\Delta^4}. \end{aligned} \tag{5}$$

A normalized cubic *B*-spline is plotted in Fig. 3.

One perhaps has noticed that the *B*-spline basis functions defined in (4) and Fig. 2 are strictly positive. This property is very attractive for image processing applications because in ordinary pictures the picture elements (pixels), being light intensities or reflectances, should always be nonnegative quantities. In case the discrete data are noisy or have certain amounts of fluctuation, it is often desirable to use positive basis functions for the interpolation such that the positivity of the interpolated image is guaranteed.

Moreover, we notice from (5) that for uniformly spaced discrete data, the cubic *B*-spline is shift-invariant. Using it as a basis function, (1) can be regarded as a convolution sum because, with very little error, we can extend both the lower and the upper summation limits to $-\infty$ and $+\infty$, respectively. This is due to the local support property of cubic *B*-splines. In other words, the interpolation operation, using cubic *B*-spline basis functions as defined in (1), now becomes a linear, shift-invariant filtering operation and the cubic *B*-spline basis function becomes the filter impulse response. The filtering property of cubic *B*-splines is further studied in Section V preceded by the development of interpolation formulas in Section IV.

IV. Theory of *B*-Spline Interpolation

Granting that (5) is the basis function for interpolation we are going to use, we shall develop an interpolating formula in this section owing to the local support and shift-invariant properties of the cubic *B*-spline.

Let us consider $\hat{f}(\xi)$ at

$$\xi = \xi_k + x\Delta \qquad 0 \leqq x \leqq 1. \tag{6}$$

From the definition of the cubic *B*-spline in (5) and illustrated in Fig. 4, $\hat{f}(\xi)$ in (1) becomes

$$\begin{aligned} \hat{f}(\xi) = \frac{1}{6\Delta^4} \{ & c_{k-1}[(\xi - \xi_{k-3})^3 - 4(\xi - \xi_{k-2})^3 \\ & + 6(\xi - \xi_{k-1})^3 - 4(\xi - \xi_k)^3] \\ & + c_{k+0}[(\xi - \xi_{k-2})^3 - 4(\xi - \xi_{k-1})^3 + 6(\xi - \xi_k)^3] \\ & + c_{k+1}[(\xi - \xi_{k-1})^3 - 4(\xi - \xi_k)^3] \\ & + c_{k+3}[(\xi - \xi_k)^3]\}. \end{aligned} \tag{7}$$

Substituting (6) into the above, we have

$$\begin{aligned} \hat{f}(\xi_k + x\Delta) &= \frac{1}{6\Delta} \{c_{k-1}[(3 + x)^3 - 4(2 + x)^3 \\ &\quad + 6(1 + x)^3 - 4x^3] \\ &\quad + c_{k+0}[(2 + x)^3 - 4(1 + x)^3 + 6x^3] \\ &\quad + c_{k+1}[(1 + x)^3 - 4x^3] + c_{k+2}x^3\} \\ &= \frac{1}{6\Delta} \sum_{i=0}^{3} b_i x^{3-i} \qquad 0 \leqq x \leqq 1 \end{aligned} \tag{8}$$

where

$$\boldsymbol{b} = \Omega \boldsymbol{c}_k$$

$\boldsymbol{b} = [b_0, b_1, b_2, b_3]^t$, $\boldsymbol{c}_k = [c_{k-1}, c_k, c_{k+1}, c_{k+2}]^t$ and

$$\Omega = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}. \tag{9}$$

Equation (7) allows us to find the interpolation at any point between knots.

In particular, at the node point $\xi = \xi_k$, i.e., $x = 0$, (8) gives

$$\hat{f}(\xi_k) = \frac{1}{6\Delta}(c_{k-1} + 4c_k + c_{k+1}). \tag{10}$$

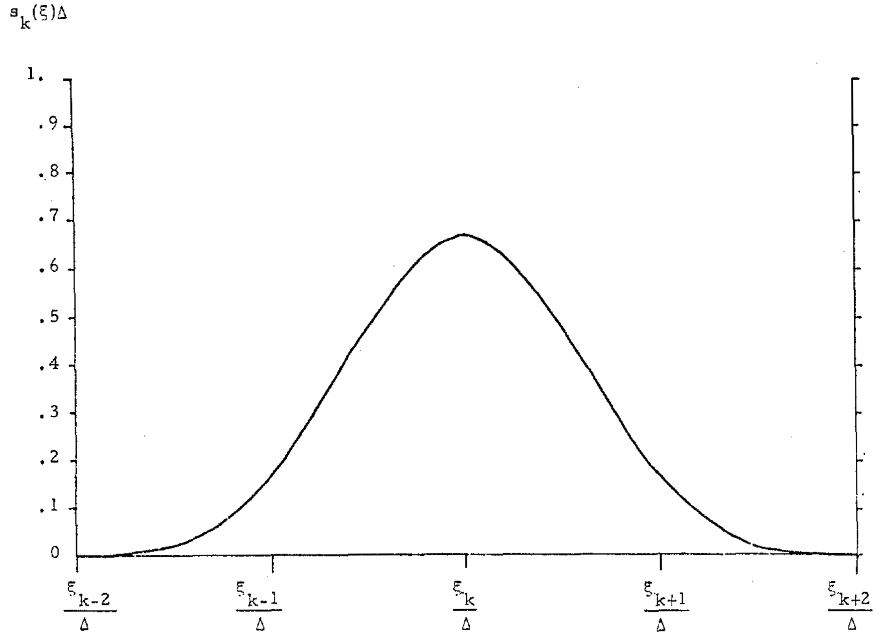


Fig. 3. The normalized cubic *B*-spline.

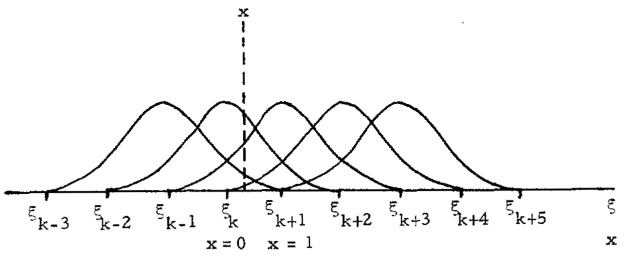


Fig. 4. $\hat{f}(\xi)$ interpolated by cubic *B*-splines.

Written in vector form for $\hat{f}(\xi)$ at the node points, we have from (8) and (10),

$$\hat{f} = Ec \tag{11}$$

where

$$E = \frac{1}{6\Delta}\begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & \bigcirc & \\ & 1 & 4 & 1 & \\ & \ddots & \ddots & \ddots & \ddots \\ \bigcirc & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix} \tag{12}$$

$$\hat{f} = [\hat{f}(\xi_1), \hat{f}(\xi_2), \cdots, \hat{f}(\xi_k)]^t$$

and

$$c = [c_1, c_2, \cdots, c_k]^t.$$

Evidently, matrix E is strictly diagonally dominant with positive real diagonal entires, hence, it is positive definite. In addition, it is banded, real, and symmetric. The inversion of E should then be easy to find. We should like to emphasize here that $\hat{f}$ at the nodes given by (11) are the given sampled data $\hat{f}$. It is therefore valid to write

$$f = Ec \tag{13}$$

where

$$f = [f(\xi_1), f(\xi_2), \cdots, f(\xi_k)]^t.$$

Extending the above one-dimensional case to two dimensions, (1) can be written as [12]

$$\hat{f}(\xi, \eta) = \sum_{k=1}^{K} \sum_{l=1}^{L} c_{kl} s_k(\xi)\, s_l(\eta). \tag{14}$$

Let us consider $f(\xi, \eta)$ at the point $(\xi = \xi_k + x\Delta, \eta = \eta_l + y\Delta)$, where $0 \leqq x \leqq 1$ and $0 \leqq y \leqq 1$. Following (8),

$$\begin{aligned}\hat{f}(x,y) = \frac{1}{6\Delta}\{&\hat{f}_{l-1}(x)\,[(3+y)^3 - 4(2+y)^3 \\ &+ 6(1+y)^3 - 4y^3] \\ &+ \hat{f}_l(x)\,[(2+y)^3 - 4(1+y)^3 + 6y^3] \\ &+ \hat{f}_{l+1}(x)\,[(1+y)^3) - 4y^3] + f_{l+2}(x) \cdot y^3\}\end{aligned} \tag{15}$$

where from (8)

$$\hat{f}_j(x) = \frac{1}{6\Delta} \sum_{i=0}^{3} b_{ij} x^{3-i} \qquad j = l-1, l, l+1, l+2 \tag{16}$$

and

$$\boldsymbol{b}_j = \Omega \boldsymbol{c}_j$$

where

$$\boldsymbol{b}_j = (b_{0j}, b_{1j}, b_{2j}, b_{3j})^t$$

$$\boldsymbol{c}_j = (c_{k-1,j}, c_{k,j}, c_{k+1,j}, c_{k+2,j})^t$$

and Ω is given by (9).

In particular, considering (15) at the node point (ξ_k, η_l), i.e., $x = 0$ and $y = 0$, we have

$$\hat{f}(\xi_k, \eta_l) = \frac{1}{36\Delta^2}\,[c_{k-1,\,l-1} + 4c_{k,\,l-1} + c_{k+1,\,l-1}) + 4(c_{k-1,\,l} + 4c_{k,\,l} + c_{k+1,\,l}) + (c_{k-1,\,l+1} + 4c_{k,\,l+1} + c_{k+1,\,l+1})] \tag{17}$$

for all $k = 1, 2, \cdots, K$ and $l = 1, 2, \cdots, L$.

Written in matrix form, (17) becomes

$$F = ECE \tag{18}$$

where F is a matrix composed of the object samples at the knots, C is the matrix with elements c_{kl}, and E is given by (12).

V. Theory of B-Spline Filtering

To gain some physical insights on the linear filtering property of the cubic B-spline function, we next turn to the spatial frequency domain. Taking the Fourier transform on both sides of (5) and using the shift invariant property of the B-spline function, we have

$$\hat{F}(\omega) = \mathcal{T}[s_0(\xi)] \sum_{k=1}^{K} c_k e^{-j\omega\xi_k} \tag{19}$$

where

$$s_0(\xi) = B_3(\xi; \xi_{-2}, \xi_{-1}, \xi_0, \xi_1, \xi_2)$$

and $B_3(\xi: \xi_{-2}, \xi_{-1}, \xi_0, \xi_1, \xi_2)$ is defined in (5). Now for a uniform partition with mesh size Δ, $\xi_k = k\Delta$ and from Fig. 2

$$\mathcal{T}[s_0(\xi)] = \left[\text{sinc}\left(\frac{\omega\Delta}{2}\right)\right]^4 \triangleq \hat{W}_0(\omega). \tag{20}$$

The window function $\hat{W}_0(\omega)$ is known as a Parzen window [27] which is nonnegative, has finite second moment, and it can be shown that $\hat{W}_0(\omega) \to 0$ as ω^{-4}. Hence, (19) becomes

$$\hat{F}(\omega) = \hat{W}_0(\omega) \sum_{k=1}^{K} c_k e^{-jk\omega\Delta}. \tag{21}$$

One interpretation of (21), as shown in Fig. 5, is therefore a string of impulses passing through a Parzen window filter with transfer function $\hat{W}_0(\omega)$. The output of the filter is the smoothed interpolation $\hat{f}(\xi)$. Another interpretation of (21) is based on the adoption of windowing in the discrete Fourier transform. Suppose now the Fourier transform of a discrete data set $\{f_k\}$ is desired. However, errors often result in the Fourier transform if a direct discrete Fourier transform is carried out. The errors are commonly caused by undersampling and truncation on $\{f_k\}$. One way to reduce these errors is to perform a continuous Fourier transform on $\hat{f}(\xi)$, where $\hat{f}(\xi)$ is given by (1). The resultant Fourier transform is of course (21), of which the coefficients $\{c_k\}$ are related to $\{f_k\}$ through (13).

Furthermore, as compared with other existing window filters, the Parzen window filter is optimum in the sense that the power spectrum of $\hat{f}(\xi)$ has the smallest variance [27] for a random sequence $\{c_k\}$ as shown in (21). One would expect good smoothing results for using such as an interpolation filter. In fact, this smoothing property can be clearly shown if we consider a continuous input $f(\xi)$ to the Parzen window filter. The filter output is then

$$\bar{f}(\xi_k) = \int_{\xi_{k-2}}^{\xi_{k+2}} f(\xi)\, s_0(\xi - \xi_k)\, d\xi. \tag{22}$$

Applying the generalized mean value theorem [28] since $s_0(\xi - \xi_k)$ is one-sign in the interval (ξ_{k-2}, ξ_{k+2}) and using the unity normalization property of B-spline, we have from (22)

$$\bar{f}(\xi_k) = f(\xi')$$

where ξ' is a point in (ξ_{k-2}, ξ_{k+2}). If $f(\xi)$ happens to be symmetric in this interval, then $\bar{f}(\xi_k)$ is also the median value of $f(\xi)$ in that interval.

In image processing practice, we like to impose an upper bound on the interpolated result. This is because most physical imaging systems have a limit on the maximum brightness of an image, e.g., film saturation, cathode ray tube phosphor heating, etc. Thus, for a given upper bound on the sampled data f, this implies that the coefficients c_i are bounded according to (13), and it will be shown in the following that the continuous output of the Parzen window filter is also bounded. Obviously, this is due to the smoothing nature of the filter. Furthermore, we can show that the second derivative estimate is also bounded; this is in accordance with our constant usage of the second derivative as a measure of smoothness in the cubic B-spline function. The two-dimensional notation for imagery, however, is unnecessarily complicated and consequently, we will stick to the one-dimensional formulation.

Making use of (21) in the following Fourier transform relationship, we have

$$[\hat{f}(\xi)]^2 = \left[\frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{F}(\omega)\, e^{j\omega\xi}\, d\omega\right]^2 = \left[\frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{W}_0(\omega)\, e^{j\omega\xi} \sum_{k=1}^{K} c_k e^{-jk\omega\Delta}\, d\omega\right]^2$$

where $\hat{W}_0(\omega)$ is defined in (20). From Schwarz's inequality we have

$$[\hat{f}(\xi)]^2 \leqq K_1^2 \int_{-\infty}^{\infty} \left|\sum_{k=1}^{K} c_k e^{j\omega(\xi - k\Delta)}\right|^2 d\omega \tag{23}$$

where

$$K_1^2 = \frac{1}{(2\pi)^2}\int_{-\infty}^{\infty} |W_0(\omega)|^2\, d\omega.$$

Then applying Parseval's formula to the right-hand side of (23), we get

$$[\hat{f}(\xi)]^2 \leqq K_1^2 \int_{-\infty}^{\infty}\left[\sum_{k=1}^{K} c_k^2\, \delta(\xi - k\Delta)\right] d\xi = K_1^2 R_c^2 \qquad \text{for any } \xi$$

where

$$R_c^2 = \sum_{k=1}^{K} c_k^2.$$

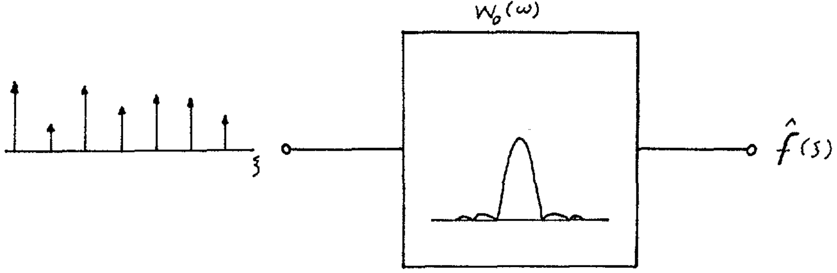


Fig. 5. The Parzen window filter.

Thus, if we impose that $\hat{f}(\xi) < M$, one of the possible bounds on the c_i's is

$$\left[\sum_{k=1}^{K} c_k^2\right]^{1/2} = M/K_1. \tag{24}$$

Geometrically, this means that if $\hat{f}(\xi)$ is within a sphere in the spline coefficient space, the radius of that sphere is proportional to the Euclidean norm in that space.

Furthermore, $\hat{f}''(\xi)$ is also bounded in the same above setting. This can be easily shown as we notice that

$$\hat{f}''(\xi) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \omega^2 \hat{F}(\omega)\, e^{j\omega\xi}\, d\omega.$$

Substituting (21) into the above equation we get

$$[\hat{f}''(\xi)]^2 = \left[\frac{1}{2\pi}\int_{-\infty}^{\infty} \omega^2 \hat{W}_0(\omega)\, e^{j\omega\xi} \sum_{k=1}^{K} c_k\, e^{-jk\omega\Delta}\, d\omega\right]^2.$$

Then applying Schwarz's inequality,

$$[\hat{f}''(\xi)]^2 \leqq K_2^2 \int_{-\infty}^{\infty} \left| \sum_{k=1}^{K} c_k\, e^{j\omega(\xi - k\Delta)} \right|^2 d\omega$$

where

$$K_2^2 = \frac{1}{(2\pi)^2}\int_{-\infty}^{\infty} \omega^4\, |W_0(\omega)|^2\, d\omega.$$

K_2^2 has a finite value because $\hat{W}_0(\omega) \rightarrow 0$ as ω^{-4}. Finally,

$$[\hat{f}''(\xi)]^2 \leqq K_2^2 R_c^2$$

or

$$|\hat{f}''(\xi)| \leqq \frac{K_2}{K_1} M \tag{25}$$

if $\hat{f}(\xi) \leqq M$ and $R_c = M/K_1$.

The above relationships have clearly shown that $\hat{f}''(\xi)$ is bounded if $\hat{f}(\xi)$ is or vice-versa.

VI. Computer Implementation

As shown above, (1) and (15) constitute the fundamental interpolation formulas for using cubic B-splines in one and two dimensions. Those formulas also allow us to perform noninteger factor image resolution conversion, magnification, and minification. Nevertheless, to develop a fast interpolation algorithm by taking advantage of the local basis property we should like to derive two methods which can be implemented by software and hardware.

A. Software Method

It is evident that a three-dimensional plot can be generated from (15) for $\hat{f}(x, y)$ at each (x, y). However, this multiple dimension table-look-up method by using (15) can be further simplified in the following fashion. The coefficient C matrix was obtained from an inverse interpolating operation on the input data F, i.e., from (18)

$$C = E^{-1} F E^{-1} \tag{26}$$

where matrix E is given in (12). We notice that E is positive, symmetric, tridiagonal, and diagonally dominant. Even so, to invert E, classical methods such as the Cholesky decomposition method [29] can be applied since E is also positive definite, and there exists a simple algorithm for computing E^{-1} [30] which states as such for $E^{-1} = (\nu_{k,l})_{K \times K}$,

$$\nu_{k,l} = \nu_{l,k} = \frac{(-1)^{k+l} \mu_l \mu_{K-k+1}}{4\mu_K - \mu_{K-1}} \quad \text{for} \quad l \leqq k \tag{27}$$

where

$$\mu_1 = 1$$

$$\mu_2 = 2$$

$$\mu_i = 4\mu_{i-1} - \mu_{i-2} \quad \text{for} \quad i = 3, 4, \cdots, K. \tag{28}$$

Having found the coefficients C from the input data F, the one-dimensional interpolation formula in (7) is applied to every row of C then to every column of C. In fact, the simulation results shown in Section VII were obtained in this way.

B. Hardware Method

In this section we shall describe a scheme to realize the cubic B-spline interpolator by a physically realizable digital filter. First let us consider the one-dimensional interpolation relation in (1) which is a discrete convolution sum for the discrete data $\{c_i\}$. In digital processing the continuous coordinate ξ must be digitized, say $\xi = n\delta$ where n is a running integer and δ is the physical spacing between adjacent digitized data. By doing so, the z transform of $\hat{f}$ is equal to the product of each of the z transforms of c and s_0. Now we would like to derive the z transform of s from the basic relation in (5).

From the basic definition of one-sided z transform

$$Z(s) = \sum_{n=0}^{\infty} s(n)\, z^{-n}$$

we obtain the z transform of n^3 as [31]

$$Z(n^3) = z^{-1} \frac{d}{dz^{-1}} \left[\frac{z^{-1} + z^{-2}}{(1 - z^{-1})^3}\right]$$
$$= \frac{z^{-1}}{(1 - z^{-1})^4} [1 + 4z^{-1} + z^{-2}].$$

Now letting $\Delta = m\delta$, where $m \geqslant 1$ is a fixed integer and making use of the shifting theorem in z transform to (5), we get the z transform of s_0

$$S_0(z) = \frac{1}{6m\delta} Z(n^3) [z^{2m} - 4z^m + 6 - 4z^{-m} + z^{-2m}]. \quad (29)$$

Evidently, the z transform of s_0 given by (20) is also the transfer function of the digital interpolator we have sought for, which can be written in the following short-hand form,

$$S_0(z) = \frac{\hat{F}(z)}{C(z)} = K_m H_1(z) H_2(z) H_3^4(z) \quad (30)$$

where

$$K_m = \frac{1}{6m\delta}$$

$$H_1(z) = (1 + 4z^{-1} + z^{-2})/z^{-1} \quad (31)$$

$$H_2(z) = z^{2(m-1)} \quad (32)$$

$$H_3(z) = (1 - z^{-m})/(1 - z^{-1})$$
$$= \sum_{k=0}^{m-1} z^{-k}. \quad (33)$$

$\hat{F}(z)$ and $C(z)$ denote the z transforms of $\hat{f}$ and c.

In the above formulation, i.e., (20)-(33), we have assumed that m is a positive integer and $m \geqslant 1$. This is because we only consider the integer factor up-conversion or the integer factor magnification here. For this reason we shall call the circuit shown in Fig. 6 an up-converter. For a down-converter, the role of input and output in the above formulation shall be interchanged and its transfer function is the inverse of (30). For noninteger resolution conversion we need to connect the up-converter and down-converter in cascade. For example, the up-converter will give $1:m$ conversion and the down-converter will deliver $p:1$ conversion, then the over-all conversion ratio is $1:m/p$ where both m and p are integers, but they may not have a common divisor.

The digital interpolator described by (30) can be viewed as six digital filters, i.e., one $H_1(z)$, one $H_2(z)$, and four $H_3(z)$ connected in cascade as depicted in Fig. 6. Each one can be easily implemented by present digital hardware such as adders, shifted registers, and read-only memory (ROM) or programmable read-only memory (PROM).

VII. Experimental Results

In this section we shall present the experimental results of picture enlargement and reduction by using different interpolation schemes. The purpose is to compare the cubic B-spline interpolation results for both image magnification and minification with those obtained from other commonly familiar methods.

A. Image Magnification

Three methods have been simulated for image enlargement; they are replication, bilinear, and cubic B-spline interpolations.

By replication we mean that each pixel is repeated inside an $m \times m$ square, where m is the linear magnification factor. In other words, the interpolating basis function is the sample-hold function B_0, as shown in Fig. 2.

By bilinear interpolation we mean that we operate on every row of data and then every column of data by the following interpolating formula:

$$\hat{f}(k + r) = r\hat{f}(k + 1) + (1 - r)\hat{f}(k) \quad (34)$$

for $0 \leq r \leq 1$. In other words, the interpolating basis function is the Chateau function B_1 as shown in Fig. 2.

The computer simulation results obtained by using the above three interpolation schemes are shown in Figs. 7 and 8. The pictures displayed in Figs. 7 and 8 are 512 by 512 sizes enlarged from 64 by 64 originals. Among those, images in Figs. 7(a) and 8(a) are obtained from the replication interpolation, in Figs. 7(b) and 8(b) from the bilinear interpolation, and in Fig. 7(c) and (d) and Fig. 8(c) and (d) from the cubic B-spline interpolation. The difference between those in Figs. 7(c) and 8(c) and in Figs. 7(d) and 8(d) is that in Figs. 7(d) and 8(d) we have used the original 64 by 64 image pixels in lieu of the correct 64 by 64 C coefficients.

In comparing those results from the three different interpolation procedures, the replication has resulted in "jaggies" along the edges. In the high resolution region the bilinear and cubic B-spline interpolations have shown some differences, demonstrated in Fig. 8(b) and (c). Some high-frequency detail loss can be noted in Fig. 8(b) for which the bilinear interpolation was used. Understandably, the superior performance of cubic B-spline interpolation is due to some negative coefficients in C which can preserve the resolution and the contrast of the original image. This becomes apparent if we compare images in Figs. 7(c) and 8(c) with those in Figs. 7(d) and 8(d). On the other hand, the noise in the original girl picture can be filtered out if the original image pixels at the knots in lieu of the C coefficients have been used in the cubic B-spline interpolation formula. This is clearly demonstrated in Fig. 7(d) as compared with the noisy picture in Fig. 7(c).

B. Image Minification

In this section we shall show some simulation results by using different methods for binary (i.e., black-white) image minification. Figs. 9 and 10 show the original binary image prior to minification. The reduced image by using cubic B-spline interpolation is shown in Fig. 11. For the purpose of comparison, we have used the truncated sinc functions as defined in (2) as basis functions. The reduction results are shown in Fig. 12(a)-(c), respectively, for different truncation intervals. Fig. 12(a) is for the sinc function truncated to the first zero crossing, 12(b) for the second zero crossing and 12(c) for the third zero crossing. These truncated sinc functions are shown in Fig. 13. It is evident from Fig. 12(a) that the blurring of the reduced images is due to the very low-pass nature of the truncated sinc in Fig. 13(a). Furthermore, the cavities and the missing lines in Fig. 12(b) and (c) are due to the negative overshoots in the

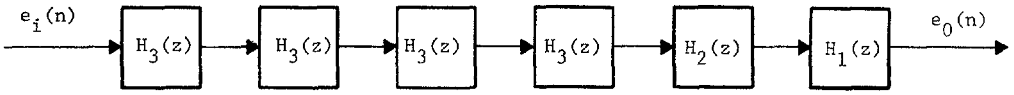


Fig. 6. Block diagram of digital interpolator $S_0(z)$.

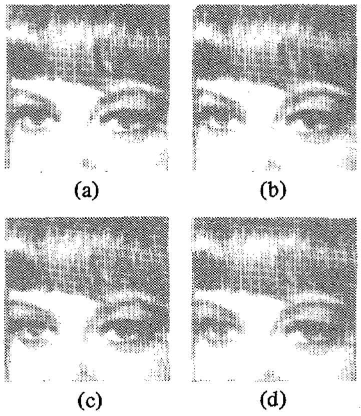


Fig. 7. Enlargement by interpolation comparisons (original is 64 × 64, output is 512 × 512). (a) Replication. (b) Bilinear interpolation. (c) Cubic *B*-spline interpolation. (d) Cubic *B*-spline interpolation with incorrect coefficients.

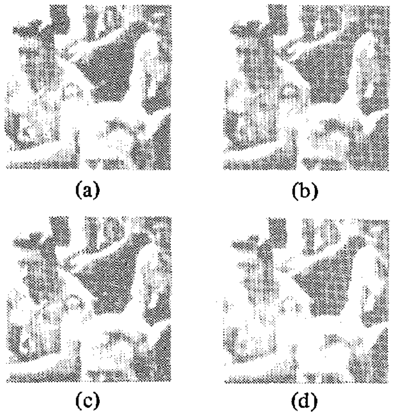


Fig. 8. (a) Replication. (b) Bilinear interpolation. (c) Cubic *B*-spline interpolation. (d) Cubic *B*-spline interpolation with incorrect coefficients.

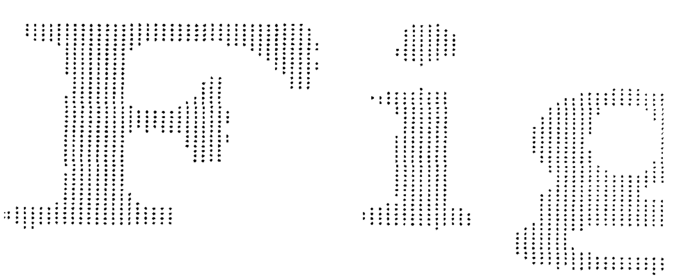


Fig. 9. Original text for minification comparison.

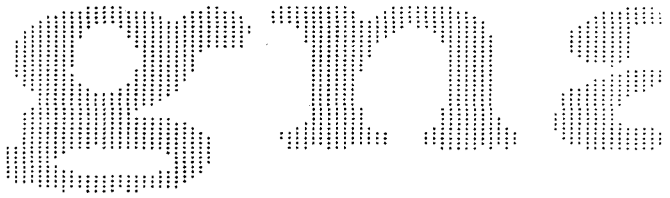


Fig. 10. Original text for minification comparison.

truncated sinc functions in Fig. 13(b) and (c), which also cause reduced oscillation of pixels. In addition, because of this oscillating property, we have found that the reduced images by using sinc function interpolation are sensitive to the quantization level on the output data. On the other hand, the reduced images by using cubic *B*-spline interpolation is insensitive to the change of quantization level. For the results as shown here, the quantization level in either direction is halfway between white and black and the reduction ratio is 0.647. As predicted, the best image is resulted from using the cubic *B*-spline interpolation. In other words, the cubic *B*-spline interpolation tends to give a faithful minification of the original image.

As we have mentioned in Section VI, that for minification the roles of input and output are interchanged as compared with the case of magnification. The simulation results shown here were obtained with the basis functions spanned over the output knots.

VIII. Conclusion

From an engineering analysis and application point of view, we have studied the interpolation and smoothing property of the cubic *B*-spline function. Its local support and shift-invariant properties offer very attractive computational procedures. Computing algorithms implemented by software and hardware have been discussed. Computer simulations with applications to image magnification, minification enlargement, and noise smoothing have shown that the cubic *B*-spline interpolation is superior to other interpolation methods.

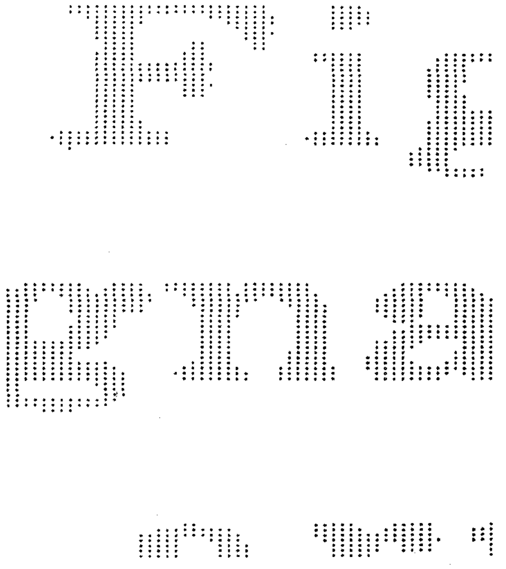
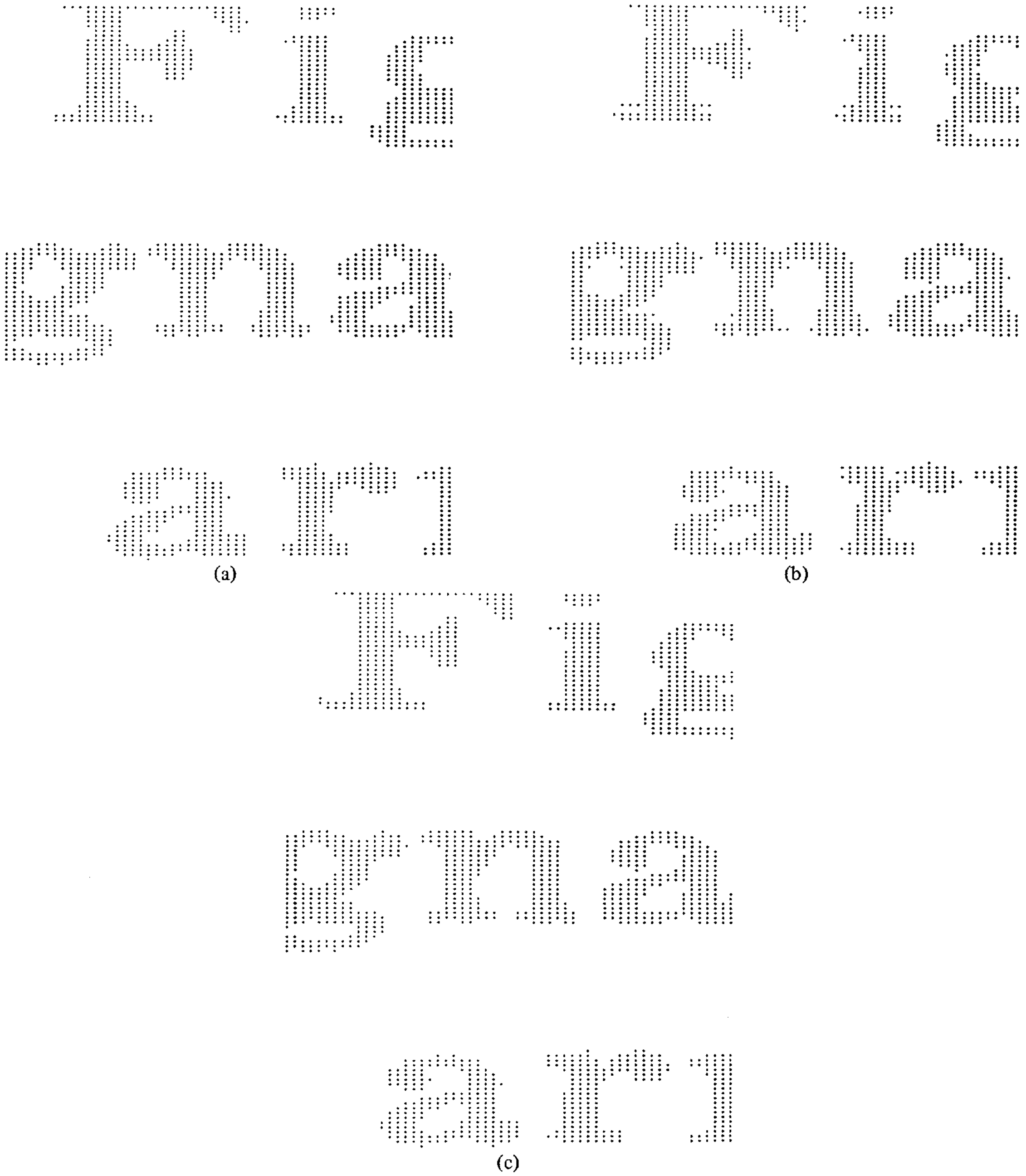


Fig. 11. 0.647 reduction of Figs. 9 and 10 by cubic *B*-spline interpolation.



(a) (b) (c)

Fig. 12. 0.647 reduction of Figs. 9 and 10 by truncated sinc function interpolation. (a) Using Fig. 13(a). (b) Using Fig. 13(b). (c) Using Fig. 13(c).

References

[1] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, pp. 692-702, June 1973.

[2] I. J. Schoenberg, "Spline functions and the problem of graduation," *Proc. Nat. Acad. Sci.*, vol. 52, pp. 947-950, 1964.

[3] E. T. Whittaker, "On a new method of graduation," *Proc. Edinburgh Math. Soc.*, vol. 41, pp. 63-75, 1923.

[4] T. N. E. Greville, *Theory and Application of Spline Functions.* New York: Academic, 1969.

[5] M. Schultz, *Spline Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1973.

[6] I. J. Schoenberg, "On interpolation by spline functions and its minimal properties," in *Proceedings of the Conference on Approximation Theory*, P. L. Butzer and J. Korevaar, Eds. Basel: Verlag, 1964.

[7] H. B. Curry and I. J. Schoenberg, "On polya frequence functions, IV: The fundamental spline functions and their limits," *J. Analyse Math.*, vol. 17, pp. 71-167, 1966.

[8] T. N. E. Greville, "Spline functions, interpolation, and numerical quadrature," in *Mathematical Methods for Digital Computers*, vol. II, A. Ralston and H. S. Wilf, Eds. New York: Wiley, 1967.

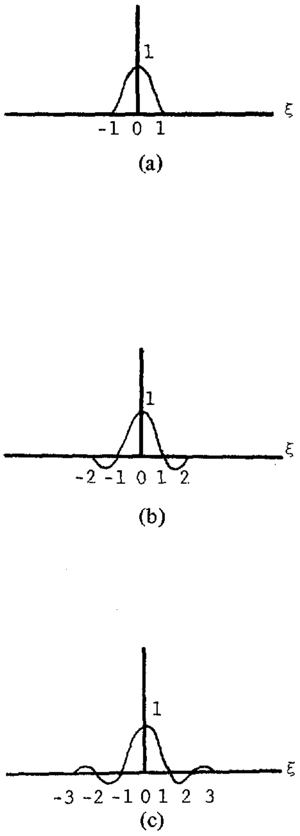
[9] T. Lynche and L. Schumaker, "Computation of smoothing and interpolating natural splines via local bases," Center for Numerical Analysis, University of Texas, Austin, Rep. 17, Apr. 1974.

[10] C. deBoor, "On uniform approximation by splines," *J. Approximation Theory*, vol. 1, pp. 219-235, 1968.

[11] J. H. Ahlberg, E. N. Nilson, and J. L. Walsh, *The Theory of Splines and Their Applications.* New York: Academic, 1967.

[12] M. J. Munteanu and L. L. Schumaker, "Some multi-dimensional spline approximation methods," *J. Approximation Theory*, vol. 10, pp. 23-40, 1974.

[13] G. Wahba, "A polynomial algorithm for density estimation," *Ann. Math. Statist.*, vol. 42, pp. 1870-1886, 1971.



(a)

(b)

(c)

Fig. 13. Truncated sinc basis functions.

[14] ——, Interpolating spline methods for density estimation, I: Equi-spaced knots," *Ann. Statist.*, vol. 3, pp. 30-48, 1975.

[15] J. O. Bennett, "Estimation on multivariate probability density functions using *B*-splines," Ph.D. dissertation, Rice University, Houston, TX, 1974.

[16] L. L. Horowitz, "The effects of spline interpolation on power spectral density," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 22-27, Feb. 1974.

[17] T. Pavlidis, "Optimal piecewise polynomial L_2 approximation of functions of one and two variables," *IEEE Trans. Comput.*, vol. C-24, pp. 98-102, Jan. 1975.

[18] M. L. Liou, "Spline fit made easy," *IEEE Trans. Comput.*, vol. C-25, pp. 522-527, May 1976.

[19] A. K. Cline, "Scalar- and planar-valued curve fitting using spline under tension," *Comm. Ass. Comput. Mach.*, vol. 17, pp. 218-220, 1974.

[20] H. Akima, "A method of bivariate interpolation and smooth surface fitting based on local procedures," *Comm. Ass. Comput. Mach.*, vol. 17, no. 1, pp. 18-20, 1974.

[21] L. E. Ostrander, "The Fourier transform of spline-function approximations to continuous data," *IEEE Trans. Audio Electroacoust.*, vol. AU-19, pp. 103-104, Mar. 1971.

[22] A. Caprihan, "Finite-duration digital filter design by use of cubic splines," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 204-207, Mar. 1975.

[23] H. S. Hou and H. C. Andrews, "Least squares image restoration using spline basis functions," *IEEE Trans. Comput.*, vol. C-26, pp. 856-873, Sept. 1977.

[24] Can be found in most numerical analysis books, in particular, in E. K. Blum, *Numerical Analysis and Computation: Theory and Practice.* Reading, MA: Addison-Wesley, 1972.

[25] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, pp. 10-21, Jan. 1949.

[26] E. T. Whittaker, "On the functions which are represented by the expansions of the interpolation theory," *Proc. Roy. Soc. Edinburgh*, vol. 35, pp. 181-194, 1915.

[27] G. M. Jenkins and D. G. Watts, *Spectral Analysis and Its Applications.* San Francisco, CA: Holden-Day, 1968.

[28] P. J. Davis and P. Rabinowitz, *Numerical Integration.* Waltham, MA: Blaisdell, 1967, p. 5.

[29] G. H. Golub, "Matrix decomposition and statistical calculations," in *Statistical Computation*, R. C. Milton and J. A. Nelder, Eds. New York: Academic, 1969, pp. 365-397.

[30] R. T. Gregory and D. L. Karney, *A Collection of Matrices for Testing Computational Algorithms.* New York: Wiley, 1969, p. 48.

[31] R. J. Schwarz and B. Friedland, *Linear Systems.* New York: McGraw-Hill, 1965, p. 246.

[32] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital filter for PCM encoded signals," U.S. Patent 3777130, Dec. 3, 1973.

[33] A. Peled and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 456-462, Dec. 1974.