COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING **52**, 264–275 (1990)

B-Spline Curves and Surfaces Viewed as Digital Filters

ARDESHIR GOSHTASBY

Electrical Engineering & Computer Science Department, University of Illinois at Chicago, Chicago, Illinois 60680

FUHUA CHENG

Computer Science Department, University of Kentucky, Lexington, Kentucky 40506

AND

BRIAN A. BARSKY

Computer Science Division-EECS, University of California at Berkeley, Berkeley, California 94720

Received March 29, 1988; accepted September 28, 1989

In this paper, we show that *B*-spline curves and surfaces can be viewed as digital filters. Viewing *B*-spline problems as digital filters allows one to predict some properties of the generated curves and surfaces. We find that even-order *B*-splines and odd-order *B*-splines behave differently when used in curve and surface interpolation. Even-order *B*-splines generate smoother curves and surfaces than do odd-order *B*-splines. © 1990 Academic Press, Inc.

1. INTRODUCTION

B-spline curves and surfaces have wide applications in image processing [1, 2] and computer aided geometric design [3, 4]. Although *B*-splines have been used as approximation tools in the past to construct digital filters [1], in this paper, we show that *B*-splines themselves form a class of digital filters. A *B*-spline curve is defined by the control vertices of its control polygon, and a *B*-spline surface is defined by the control vertices of its control graph [4]. Determining the control vertices of a *B*-spline curve or surface for interpolation, however, requires the solution of a large system of equations [12–14]. In this paper, we show that *B*-spline curves and surfaces can be viewed as digital filters and, as a result, methods developed for digital filters can be used to determine the control vertices of *B*-spline curves and surfaces for interpolation without solving a system of equations. We determine the control vertices of *B*-spline curves and surfaces by an inverse filtering operation. Also, viewing *B*-splines as digital filters enables the prediction of the *smoothness* of the generated curves and surfaces in advance.

In the following discussion, we use the term *joint* to refer to a point on a *B*-spline curve where two adjacent curve segments join. We use the same term to refer to a point on a *B*-spline surface where four adjacent patches join. We will refer to a *B*-spline curve or surface by its *order* rather than its *degree*. A *B*-spline curve of order k is of degree $k-1$, and a *B*-spline surface of order $k \times l$ is of degree $(k-1) \times (l-1)$. In this paper, the *smoothness* of a curve or a surface is defined in terms of the spatial frequency content rather than the degree of differentiability of the curve or surface. The smoother a curve or surface, the less the amount of high spatial frequencies in the curve or surface.

0734-189X/90 $3.00

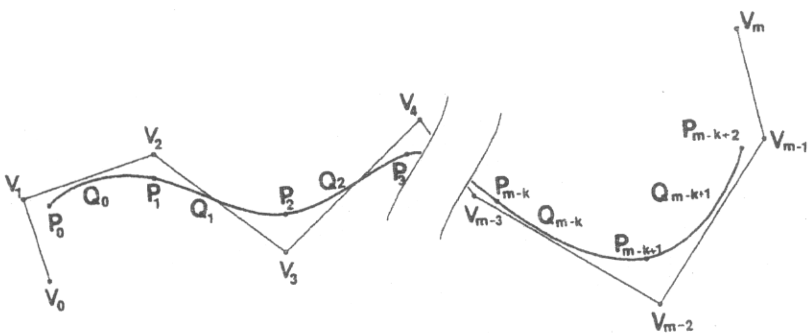
2. B-SPLINE CURVES

A uniform B-spline curve may be defined in piecewise form. The ith segment of a B-spline curve of order k is defined by [4, pp. 19–46, 173–210]

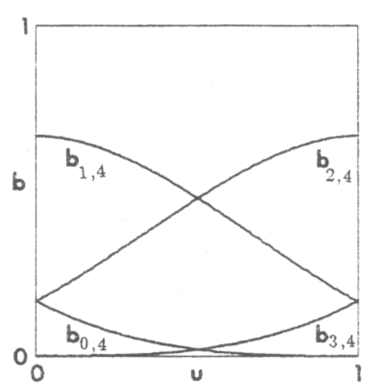
$$\mathbf{Q}_i(u) = \sum_{r=0}^{k-1} \mathbf{V}_{i+r} b_{r,k}(u), \qquad 0 \le u < 1, \quad i = 0,1,\ldots,m-k+1, \tag{1}$$

where $\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_m$ are the control vertices of the B-spline curve and $b_{r,k}(u)$ is the rth B-spline *basis segment* of order k. A uniform B-spline curve of order $k = 4$ (cubic) is shown in Fig. 1a, and the B-spline basis segments of order four are shown in Fig. 1b. The position of the ith joint, $\mathbf{P}_i$, on the B-spline is determined by setting $u = 0$ in (1). That is

$$\mathbf{P}_i = \mathbf{Q}_i(0) = \sum_{r=0}^{k-1} \mathbf{V}_{i+r} b_{r,k}(0) \qquad \text{for } i = 0,1,\ldots,m-k+1. \tag{2}$$



(a)



(b)

FIG. 1. (a) A cubic B-spline curve. $\mathbf{V}$, $\mathbf{P}$, and $\mathbf{Q}$ are the control vertices, the joints, and the curve segments, respectively, of the B-spline. (b) B-spline basis segments of order four.

The last joint, $\mathbf{P}_{m-k+2,}$ is obtained by setting $u = 1$ and $i = m - k + 1$ in Eq. (1):

$$\mathbf{P}_{m-k+2} = \mathbf{Q}_{m-k+1}(1) = \sum_{r=0}^{k-1} \mathbf{V}_{m-k+1+r} b_{r,k}(1). \quad (3)$$

In this paper, we view a uniform B-spline curve as a signal. We assume that the given interpolating points are sample points from the signal taken at equal spacing. This implies that the underlying B-spline should have unique knots that are equally spaced. With this knot arrangement, $b_{k-1,k}(0) = 0$ (see Fig. 1b) and this allows relation (2) to be rewritten as

$$\mathbf{P}_i = \sum_{r=0}^{k-2} \mathbf{V}_{i+r} b_{r,k}(0) \qquad \text{for } i = 0, 1, \ldots, m-k+1. \quad (4)$$

Similarly, since the knots are unique and uniformly spaced, $b_{0,k}(1) = 0$ (see Fig. 1b); consequently, we can write (3) as

$$\mathbf{P}_{m-k+2} = \sum_{r=1}^{k-1} \mathbf{V}_{m-k+1+r} b_{r,k}(1) \quad (5)$$

and, by performing a change of indices ($r' = r - 1$), we obtain

$$\mathbf{P}_{m-k+2} = \sum_{r'=0}^{k-2} \mathbf{V}_{m-k+2+r'} b_{r'+1,k}(1). \quad (6)$$

Since B-spline basis segments with uniform knots have the property that $b_{r+1,k}(1) = b_{r,k}(0)$, this can be substituted into (6) to yield

$$\mathbf{P}_{m-k+2} = \sum_{r'=0}^{k-2} \mathbf{V}_{m-k+2+r'} b_{r',k}(0). \quad (7)$$

Rewriting r' as r, and combining (4) and (7), we obtain

$$\mathbf{P}_i = \sum_{r=0}^{k-2} \mathbf{V}_{i+r} b_{r,k}(0), \qquad i = 0, 1, \ldots, m-k+2. \quad (8)$$

Now, relation (8) shows a convolution operation which may be written as

$$\mathbf{P} = \mathbf{V} * \mathbf{H}_k, \quad (9)$$

where $\mathbf{H}_k = [b_{0,k}(0)\ b_{1,k}(0)\ \ldots\ b_{k-2,k}(0)]$, $\mathbf{V} = [\mathbf{V}_0\ \mathbf{V}_1\ \ldots\ \mathbf{V}_m]$, $\mathbf{P} = [\mathbf{P}_0\ \mathbf{P}_1\ \ldots\ \mathbf{P}_{m-k+2}]$ for an open curve, and $\mathbf{P} = [\mathbf{P}_0\ \mathbf{P}_1\ \ldots\ \mathbf{P}_m]$ for a closed curve, and $*$ is the convolution operation [5, pp. 145–150]. Note that in the convolution operation, the number of elements determined for $\mathbf{P}$ is the same as the number of elements given in $\mathbf{V}$. For an open curve where the number of elements in $\mathbf{P}$ is smaller than the number of elements in $\mathbf{V}$, the computed first $\lfloor k/2 - 1 \rfloor$ and last $\lceil k/2 - 1 \rceil$

values of (9) should be discarded. We see that the relation between the control vertices and the joints of a *B*-spline curve is analogous to the relation between a sampled signal and its filtered counterpart, with $\mathbf{H}_k$ being the point-spread function of the filter. (The point-spread function of a filter shows the response of the filter to an impulse; a filter is totally defined by its point-spread function.) This implies that some of the techniques that have been developed for digital filtering can be applied to *B*-splines as well.

To determine the joints of a *B*-spline curve, we observe that since relation (9) is a convolution operation, we may replace it by the following operation [5, pp. 173]. Let $\mathcal{F}(\mathbf{V})$ and $\mathcal{F}(\mathbf{H}_k)$ represent the Fourier transforms of $\mathbf{V}$ and $\mathbf{H}_k$, respectively. Then the joints of the *B*-spline can be obtained by multiplying $\mathcal{F}(\mathbf{V})$ and $\mathcal{F}(\mathbf{H}_k)$ point-by-point and determining the inverse Fourier transform of the result. That is,

$$\mathbf{P} = \mathcal{F}^{-1}\{\mathcal{F}(\mathbf{V}) \cdot \mathcal{F}(\mathbf{H}_k)\}, \tag{10}$$

where the dot represents the point-by-point multiplication operation, and $\mathcal{F}^{-1}$ indicates the inverse Fourier transform operation. Note that $\mathbf{H}_k$ and $\mathbf{V}$ are required to be of the same length. Since the lengths of $\mathbf{H}_k$ and $\mathbf{V}$ are $k-1$ and $m \geq k$, respectively, $\mathbf{H}_k$ has fewer elements than $\mathbf{V}$ does. However, we can add an appropriate number of zeros to $\mathbf{H}_k$ to make $\mathbf{V}$ and $\mathbf{H}_k$ have the same length. We can use the fast Fourier transform (FFT) algorithm [6] to compute the *B*-splines efficiently. Note that relation (10) holds for a closed curve where the number of control vertices is equal to the number of joints. For an open curve, the obtained first $\lfloor k/2 - 1 \rfloor$ and last $\lceil k/2 - 1 \rceil$ joints should be discarded.

3. DETERMINING THE CONTROL VERTICES OF UNIFORM *B*-SPLINE CURVES FOR INTERPOLATION

In the following, we determine the control vertices $\mathbf{V}_r$, $r = 0, 1, \ldots, m$ of a uniform *B*-spline curve to interpolate a given set of points $\mathbf{P}_i$, $i = 0, 1, \ldots, m-k+2$. If we assume that $\mathbf{P}_i$ represents the ith joint of a *B*-spline curve, then the problem to be solved is an inverse problem where we are given the joints of a *B*-spline curve and must determine its control vertices. This problem may be solved by substituting $\mathbf{P}_i$ for $\mathbf{Q}(i)$ in (1), setting $u = i$, and solving the resulting system of $m-k+3$ linear equations for $i = 0, 1, \ldots, m-k+2$, as is done in [7]. Since the number of unknowns is $m+1$ and we have only $m-k+3$ equations, extra constraints need to be imposed so that the system can be solved. One such constraint is to repeat the end control vertices. Barsky [8] discusses different end conditions that can be used as constraints.

To avoid having to solve the large system of equations that arises when the number of given joints is large, Yamaguchi [9] suggests using an iterative algorithm to determine the control vertices. The following discussion describes an alternative method based on the idea of digital filters which determines the control vertices of a *B*-spline curve without solving a system of equations.

To determine the control vertices, we obtain the Fourier transform of both sides of Eq. (10). That is,

$$\mathcal{F}(\mathbf{P}) = \mathcal{F}(\mathbf{V}) \cdot \mathcal{F}(\mathbf{H}_k) \tag{11}$$

which implies a deconvolution operation

$$\mathbf{V} = \mathscr{F}^{-1}\left\{ \frac{\mathscr{F}(\mathbf{P})}{\mathscr{F}(\mathbf{H}_k)} \right\}. \tag{12}$$

Therefore, to determine the control vertices **V**, we compute the Fourier transform of **P**, compute the Fourier transform of $\mathbf{H}_k$, divide the former by the latter point-by-point, and find the inverse transform of the result.

The FFT algorithm may be used to carry out this computation [6]. Note that this method does not require any end conditions and that it works for closed curves where the number of control vertices is equal to the number of joints. If an open curve is used, the first $\lfloor k/2 - 1 \rfloor$ and the last $\lceil k/2 - 1 \rceil$ control vertices of the B-spline curve will not be found. If some end conditions are provided, such as multiply defining the end control vertices, then all the control vertices will be determined. Note that computation of (12) is bases on the requirement that all Fourier transform coefficients of the underlying filter be non-zero to avoid a zero in the denominator of (12).

Regarding computational complexity, since the FFT can be carried out in $O(m \log m)$ operations for m input points [6], solving (12) requires only $O(m \log m)$ operations. This complexity is independent of the order of B-spline. It is possible to find the control vertices of an open uniform B-spline curve of order four in $O(m)$ operations for m points by solving a system of equations directly or iteratively [7, 9, 12, 13]. This is because a three-term linear first-order recurrence relation holds for B-splines of order four; therefore, a tridiagonal system of equations can be constructed which allows an especially efficient algorithm to be designed. However, this three-term linear first-order recurrence relation does not hold for closed B-spline curves or B-spline curves of higher order. The above approach, therefore, provides an efficient algorithm for the general case.

4. SMOOTHNESS OF *B*-SPLINE CURVES

We will define the *smoothness* of a curve in terms of the spatial frequency content of the curve. In signal processing, it is well known that any signal may be decomposed into a sum of sinusoidal waves having different magnitudes and different frequencies [11, p. 59]. The magnitude of the sinusoidal wave at a given frequency determines the amount of that frequency in the signal. A curve can be considered a spatial signal and it can be represented by a sum of sinusoidal curves with different spatial frequencies. A smooth curve has large magnitude coefficients for its low spatial frequencies and small or zero magnitude coefficients for its high spatial frequencies.

If we assume that the generated B-spline curve is a signal after being filtered, then two factors determine the spatial frequency content of the curve: the spatial frequency content of the original signal and the spatial frequency response of the filter. The spatial frequency content of the original signal is out of our control, but we can control the spatial frequency response of the filter by selecting the order of the filter. In this manner, the smoothness of the generated curve can be controlled.

The spatial frequency response of a filter may be determined by studying the *transfer function* of the filter [11, pp. 27–54]. The transfer function of a filter is

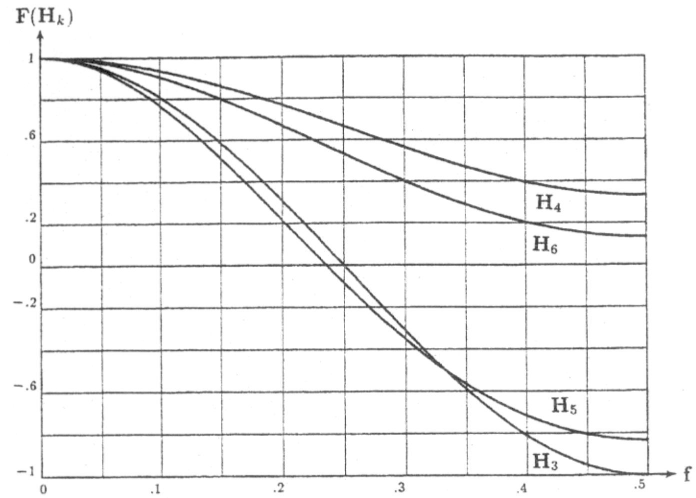


FIG. 2. Transfer function of filters $\mathbf{H}_3$, $\mathbf{H}_4$, $\mathbf{H}_5$, and $\mathbf{H}_6$. Odd-order filters and even-order filters behave very differently. The even-order filters work like low-pass filters.

obtained from the Fourier transform of the filter and describes the behavior of the filter under all input frequencies. Figure 2 shows the transfer functions of filters generating *B*-spline curves of orders 3, 4, 5, and 6.

Note that the even-order filters behave differently than odd-order filters. Specifically, the even-order filters behave like a low-pass filter, passing zero-frequency signals with no attenuation and other signals with more and more attenuation as the frequency of the signal increases. Also note that as the order of the even-order filters increases, more of the high frequencies are attenuated. This means that higher order *B*-splines generate smoother curves.

The odd-order filters attenuate the low and high frequencies only slightly. Therefore, high frequencies in the input pass through the filter with very little attenuation; however, as the order of odd-order *B*-splines increases, more of the high frequencies are attenuated. Again, therefore; smoother curves are obtained as the order of the *B*-splines increases. This phenomenon is seen in Fig. 3 which shows *B*-spline curves of orders 3, 4, 5, and 6 interpolating the same set of points.

Although the odd-order filters attenuate the high frequencies only slightly, they do attenuate the mid-frequencies considerably. Consider, for instance, $\mathbf{H}_3$. This filter has a frequency, $f = \frac{1}{4}$ (see Fig. 2), such that the amplitude of the output is zero regardless of the amplitude of the corresponding input frequency. Therefore, $\mathbf{H}_3$ will totally remove frequency $\frac{1}{4}$ from the input and attenuate frequencies around $f = \frac{1}{4}$ considerably. It follows then, that once a signal is passed through an odd-order filter, information about the original signal—at least at one frequency—is lost, and it becomes impossible to reconstruct the original signal.

In terms of *B*-spline curves, this means that given the control vertices of an odd-order *B*-spline curve, we can determine the curve. However, *given the joints of an odd-order B-spline curve, we may not be able to determine the control vertices of the curve*. This is also obvious from relation (12) where, for example, for a curve of order three at frequency $f = \frac{1}{4}$, the denominator becomes zero and we cannot

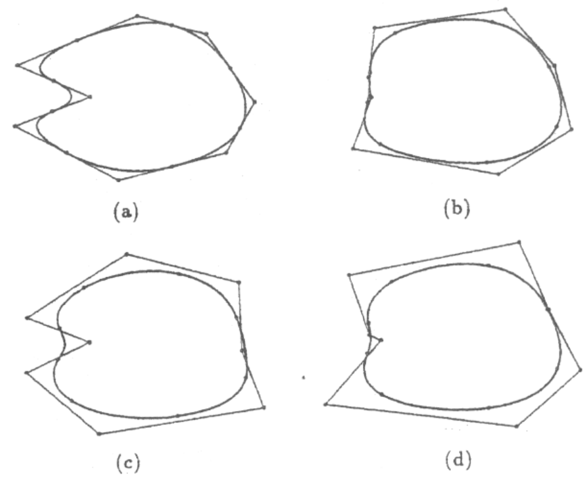


FIG. 3. (a), (b), (c), and (d), are *B*-spline curves of orders 3, 4, 5, and 6, respectively, interpolating the same set of points.

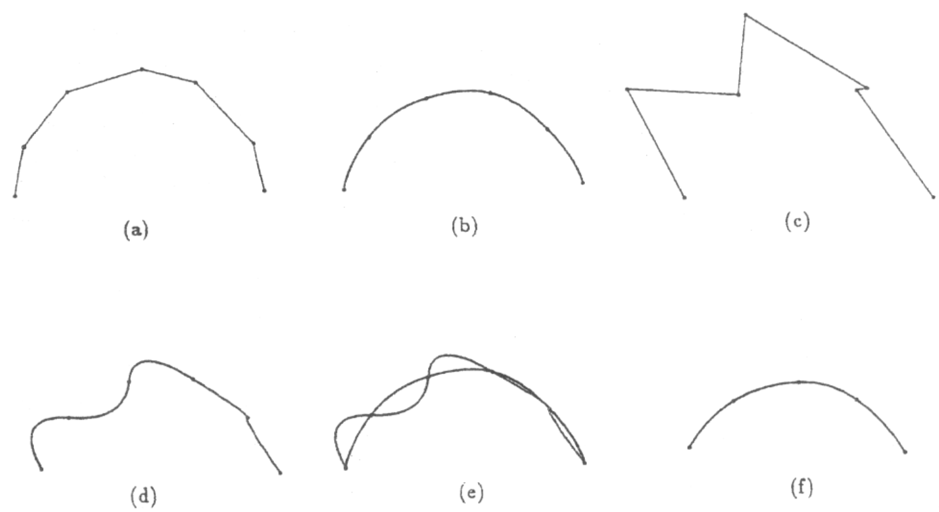


FIG. 4. (a) Seven control vertices. (b) *B*-spline curve of order three corresponding to (a). (c) Control vertices obtained from the inverse process using the joints of (b). (d) *B*-spline curve of order three obtained from control vertices of (c). (e) This figure shows that curves (b) and (d) have the same joints although they have different control vertices as shown in (a) and (c). (f) *B*-spline curve of order four obtained from control vertices in (a). If joints of (f) are used in the inverse process for a *B*-spline of order four, the control vertices of (a) would be uniquely obtained.

determine the control vertices of the *B*-spline from its joints. Figure 4 exhibits this fact. Consider the control vertices given in Fig. 4a. These control vertices define the *B*-spline curve of order three shown in Fig. 4b. If we consider the inverse problem, where the joints of a *B*-spline curve of order three are given as shown in Fig. 4b, we cannot determine the original control vertices because information about frequency $f = \frac{1}{4}$ is lost. However, assuming that the original signal did not contain any frequency equal to $\frac{1}{4}$, then we can obtain the control vertices as shown in Fig. 4c. The control vertices shown in Fig. 4c generate a *B*-spline curve of order three (see Fig. 4d) that has the same set of joints obtained from control vertices shown in Fig. 4a. In Fig. 4e, the two curves are overlaid to show that they have the same joints.

In curve interpolation, however, we see that although it would be desirable to obtain the curve of Fig. 4b. Unfortunately the inverse process determines the curves of Fig. 4d, which does not contain frequency $f = \frac{1}{4}$.

Therefore, although the joints of an odd-order *B*-spline curve can be determined uniquely from the given control vertices, the control vertices of the curve may not be determined from its joints. This suggests that where there is an interpolation problem, we should avoid using the odd-order *B*-splines and instead use the even-order *B*-splines. The relation between the control vertices and the joints of an even-order *B*-spline is unique. For the control vertices shown in Fig. 4a, we obtain a *B*-spline curve of order four as shown in Fig. 4f, and for the joints shown in Fig. 4f, we uniquely obtain the control vertices shown in Fig. 4a.

5. *B*-SPLINE SURFACES

A tensor product uniform *B*-spline surface of order $k \times l$ is defined in piecewise form over a grid of control vertices $\mathbf{V}_{rs}$, $r = 0, 1, \ldots, m$; $s = 0, 1, \ldots, n$ by [4, pp. 46–66]:

$$\mathbf{Q}_{ij}(u, v) = \sum_{r=0}^{k-1} \sum_{s=0}^{l-1} \mathbf{V}_{i+r, j+s} b_{r,k}(u) b_{s,l}(v), \qquad 0 \le u < 1, \quad 0 \le v < 1,$$

$$i = 0, 1, \ldots, m - k + 2, \text{ and } j = 0, 1, \ldots, n - l + 2. \tag{13}$$

Following a development analogous to that in Section 2 for *B*-spline curves, we obtain the following relation between the *ij*th joint and the control vertices of a *B*-spline surface,

$$\mathbf{P}_{ij} = \sum_{r=0}^{k-2} \sum_{s=0}^{l-2} \mathbf{V}_{i+r, j+s} b_{r,k}(0) b_{s,l}(0), \qquad i = 0, 1, \ldots, m - k + 2,$$

$$j = 0, 1, \ldots, n - l + 2. \tag{14}$$

Assuming the given control vertices are sampled data from a two-dimensional signal, then from the digital filtering point of view, relation (14) represents a convolution operation of the sampled data with the operator,

$$\mathbf{H}_{kl} = \begin{pmatrix} b_{0,k}(0)b_{0,l}(0) & b_{0,k}(0)b_{1,l}(0) & \cdots & b_{0,k}(0)b_{l-2,l}(0) \\ b_{1,k}(0)b_{0,l}(0) & b_{1,k}(0)b_{1,l}(0) & \cdots & b_{1,k}(0)b_{l-2,l}(0) \\ \vdots & \vdots & \ddots & \vdots \\ b_{k-2,k}(0)b_{0,l}(0) & b_{k-2,k}(0)b_{1,l}(0) & \cdots & b_{k-2,k}(0)b_{l-2,l}(0) \end{pmatrix}. \tag{15}$$

$\mathbf{H}_{kl}$ represents the point-spread function of a two-dimensional filter. Relation (14) may be written in convolution form as

$$\mathbf{P} = \mathbf{V} * \mathbf{H}_{kl}, \tag{16}$$

where $\mathbf{P}$ is the grid of joints, $\mathbf{V}$ is the grid of control vertices, $\mathbf{H}_{kl}$ is the operator of (15), and $*$ is the two-dimensional convolution operation. This relation can be written by defining $\mathbf{W}_{i+r,j}$ as

$$\mathbf{W}_{i+r,j} = \sum_{s=0}^{l-2} \mathbf{V}_{i+r,j+s} b_{s,l}(0), \tag{17}$$

where $\mathbf{W}_{i+r,j}$ represents the jth joint of a B-spline curve obtained from the control vertices in the $(i+r)$th row of the grid. Substituting (17) into (14), we obtain

$$\mathbf{P}_{ij} = \sum_{r=0}^{k-2} \mathbf{W}_{i+r,j} b_{r,k}(0), \qquad \text{for } i = 0,1,\ldots,m-k+2, \quad j = 0,1,\ldots,n-l+2. \tag{18}$$

From the digital filtering point of view, this means that instead of convolving the sample points with a two-dimensional operator, we may convolve each column of the data with a one-dimensional operator and then apply another one-dimensional operator row-by-row to the obtained result.

Filtering in one dimension rather than two dimensions considerably speeds up the computations. The FFT algorithm could be used to reduce computation time even further. Using the Fourier transform operation, from relation (16) we can obtain the relation between the control vertices and the joints of a B-spline surface as

$$\mathbf{P} = \mathscr{F}^{-1}\{\mathscr{F}(\mathbf{V}) \cdot \mathscr{F}(\mathbf{H}_{kl})\}. \tag{19}$$

Now, assuming

$$\mathbf{H}_k = [b_{0,k}(0) \ b_{1,k}(0) \ \cdots \ b_{k-2,k}(0)] \tag{20}$$

and

$$\mathbf{H}_l = [b_{0,l}(0) \ b_{1,l}(0) \ \cdots \ b_{l-2,l}(0)] \tag{21}$$

and, since $\mathbf{H}_{kl}$ separates into $\mathbf{H}_k$ and $\mathbf{H}_l$, we can compute (19) in two stages, first by performing a row-by-row convolution,

$$\mathbf{W}_{i*} = \mathcal{F}^{-1}\{\mathcal{F}(\mathbf{V}_{i*}) \cdot \mathcal{F}(\mathbf{H}_l)\} \tag{22}$$

and then by performing a column-by-column convolution,

$$\mathbf{P}_{*j} = \mathcal{F}^{-1}\{\mathcal{F}(\mathbf{W}_{*j}) \cdot \mathcal{F}(\mathbf{H}_k)\}, \tag{23}$$

where $\mathbf{V}_{i*} = [\mathbf{V}_{i0}\ \mathbf{V}_{i1}\ \cdots\ \mathbf{V}_{in}]$, and $\mathbf{W}_{i*} = [\mathbf{W}_{i0}\ \mathbf{W}_{i1}\ \cdots\ \mathbf{W}_{in}]$ and $\mathbf{W}_{*j} = [\mathbf{W}_{0j}\ \mathbf{W}_{1j}\ \cdots\ \mathbf{W}_{mj}]$ are, respectively, the ith row and the jth column of the result after the row-by-row convolution. $\mathbf{P}_{*j} = [\mathbf{P}_{0j}\ \mathbf{P}_{1j}\ \cdots\ \mathbf{P}_{mj}]$ is the jth column of the result after the column-by-column convolution.

The inverse problem, where the joints of a *B*-spline surface are given and we have to determine the control vertices of the *B*-spline, can be easily solved by inverse filtering. From (16), we can derive

$$\mathbf{V} = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}(\mathbf{P})}{\mathcal{F}(\mathbf{H}_{kl})}\right\}. \tag{24}$$

The computation of (24) again may be carried out in two stages. Relations (23) and (22) are used to compute the first stage and the second stage, respectively, of the process as follows. From (23) we find

$$\mathbf{W}_{i*} = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}(\mathbf{P}_{i*})}{\mathcal{F}(\mathbf{H}_k)}\right\} \tag{25}$$

and from (22) we obtain

$$\mathbf{V}_{*j} = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}(\mathbf{W}_{*j})}{\mathcal{F}(\mathbf{H}_l)}\right\}. \tag{26}$$

Previously, Barsky [12–14] has determined the control vertices of a *B*-spline surface of order 4×4 from the joints of the surface by solving a system of linear equations. To avoid solving a large system of equations when the number of control vertices is large, an iterative method has also been proposed [15]. The computational complexity of these methods is $O(mn)$ multiplications while the computational complexity of the filtering method developed in this paper is $O(mn(\log m + \log n))$ multiplications. Note, however, that pervious methods are restricted to *B*-spline surfaces of order 4×4 while the proposed filtering method is general and applies to *B*-spline surfaces of any order with the same computational complexity.

From the discussion in the previous section, we can conclude that for surfaces, as for curves, the inverse problem cannot be solved uniquely when using odd-order *B*-splines, but can always be solved uniquely when using even-order *B*-splines.

The smoothness (spatial frequency content) of a generated *B*-spline surface can be determined by studying the transfer function of the applied operator $\mathbf{H}_{kl}$. This

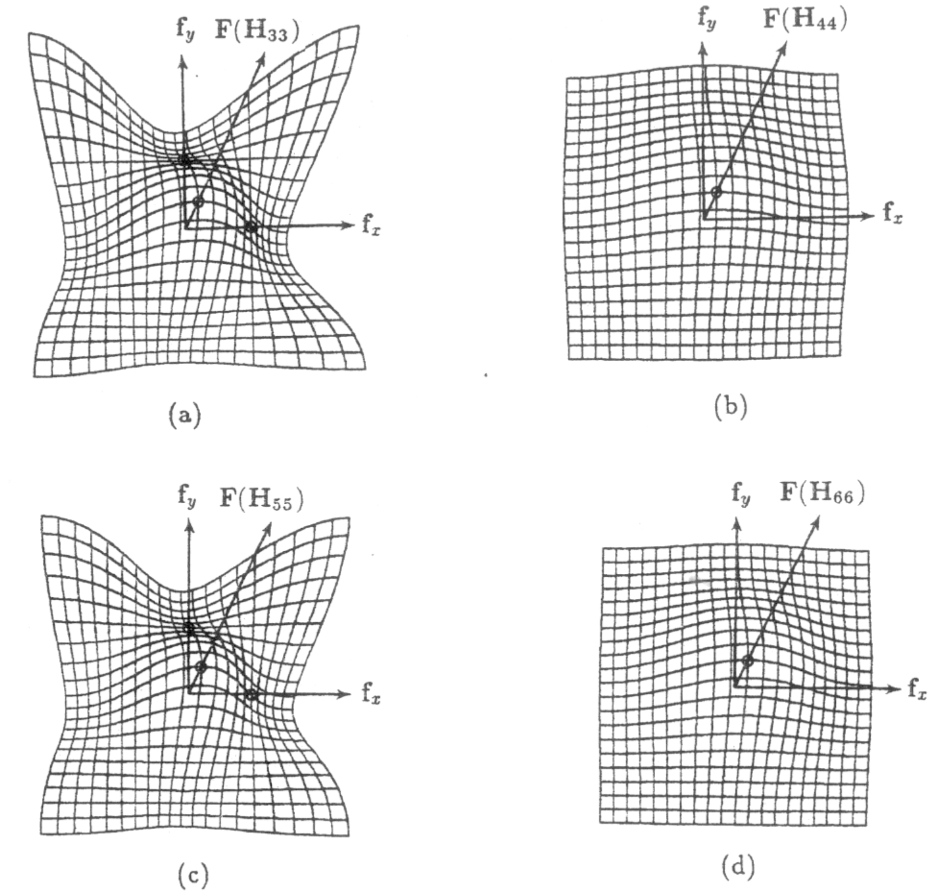


FIG. 5. (a), (b), (c), and (d) are the transfer functions of filters $\mathbf{H}_{33}$, $\mathbf{H}_{44}$, $\mathbf{H}_{55}$, and $\mathbf{H}_{66}$, respectively. The small circles show the intersections of the axes with the transfer functions of the filters.

operator depends only on the order of the *B*-spline surface, which in turn determines the smoothness of the surface. Figure 5 shows transfer functions of *B*-spline surfaces of orders 3×3, 4×4, 5×5, and 6×6. Again we see that odd-order and even-order *B*-splines behave differently. The even-order *B*-splines work like a low-pass filter while the odd-order *B*-splines generate surfaces containing low magnitude mid-frequencies but high magnitude low and high frequencies. As the order of a *B*-spline surface increases, both odd-order and even-order *B*-splines generate smoother surfaces.

6. CONCLUSION

This paper has demonstrated that *B*-spline curves and surfaces can be viewed as digital filters. If we assume that the control vertices of a *B*-spline curve or surface represent samples from a one-dimensional or a two-dimensional signal with a sampling rate equal to or greater than twice the highest frequency in the signal (Nyquist rate), then we can consider the joints of the *B*-spline as samples from the signal after it is filtered. Thus, to determine the control vertices of a *B*-spline curve or surface that interpolates a set of points, we can use the inverse filtering operation instead of solving a large system of equations.

In studying the behavior of *B*-splines as digital filters, we found that the even-order *B*-splines work like low-pass filters and generate smoother curves and surfaces as the order of the *B*-spline increases. The odd-order *B*-splines, on the other hand, work differently, generating curves and surfaces containing large-magnitude low and high frequencies but small-magnitude mid-frequencies. Therefore, the even-order *B*-splines are more appropriate for curve and surface interpolation than the odd-order *B*-splines.

In this paper, we determined digital filters that generate *B*-spline curves of order k and *B*-spline surfaces of order $k \times l$. In a similar manner, digital filters that generate other piecewise parametric curves and surfaces may be obtained. In such cases, curve and surface interpolation may be carried out by appropriate filtering operations.

REFERENCES

1. L. A. Ferrari, P. V. Sankar, and J. Sklansky, Efficient two-dimensional filters using *B*-spline functions, *Comput. Vision Graphics Image Process.* **35**, No. 2, 1986, 152–169.
2. H. S. Hou and H. Andrews, Cubic splines for image interpolation and digital filters. *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-26,** No. 6, 1978, 508–517.
3. B. A. Barsky, A description and evaluation of various 3D models, *IEEE Comput. Graphics Appl.* **4**, No. 1, 1984, 38–52.
4. R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann, Los Atlos, CA, 1987.
5. K. R. Castleman, *Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
6. E. O. Brigham and R. E. Morrow, The fast Fourier transform, *IEEE Spectrum* **4**, No. 12, 1962, 63–70.
7. B. A. Barsky and S. W. Thomas, TRANSPLINE—A system for representing curves using transformations among four spline formulations, *Comput. J.* **24**, No. 3, 1981, 271–277.
8. B. A. Barsky, End conditions and boundary conditions for uniform *B*-spline curve and surface representations, *Comput. Ind.* **3**, No. 1, 2, 1982, 17–29.
9. F. Yamaguchi, A new curve fitting method using a CRT computer display, *Comput. Graphics Image Process.* **7**, No. 3, 1978, 425–437.
10. G. W. Stewart, *Introduction to Matrix Computation*, Academic Press, New York, 1973.
11. R. W. Hamming, *Digital Filters*, Prentice–Hall, Englewood Cliffs, NJ, 1977.
12. B. A. Barsky, *A Method for Describing Curved Surfaces by Transforming between Interpolatory Spline and B-spline Representations*, Master's thesis, Cornell University, January 1979.
13. B. A. Barsky and D. P. Greenberg, Determining a set of *B*-spline control vertices to generate an interpolating surface, *Comput. Graphics Image Process.* **14**, No. 3, 1980, 203–226.
14. B. A. Barsky and D. P. Greenberg, An interactive surface representation system using a *B*-spline formulation with interpolation capability, *Comput. Aided Des.* **14**, No. 4, 1982, 187–194; Corrigendum, *Comput. Aided Des.* **15**, No. 3, 1983, 174.
15. F. Cheng and A. Goshtasby, *B-Spline Surface Interpolation Using SLOR Method with Parallel Relaxation*, Technical Report No. 94-87, Dept. of Computer Science, University of Kentucky, September 1987.