[6] Finkelstein, A., and Salesin, D. Multiresolution curves. *Computer Graphics 28*, 4 (1994), 261–268.

[7] Forsey, D., and Bartels, R. Hierarchical B-spline refinement. *Computer Graphics 22*, 4 (1988), 205–212.

[8] Fowler, B. Geometric manipulation of tensor product surfaces. In *Proceedings of the Symposium on Interactive Computer Graphics* (1992), ACM.

[9] Gortler, S. Private communication.

[10] Gortler, S., and Cohen, M. Variational modeling with wavelets. Tech. Rep. CS-TR-456-94, Princeton University, Computer Science Department, 1994.

[11] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. *Numerical Recipes in C*. Cambridge University Press, 1988.

[12] Welch, W., and Witkin, A. Variational surface modeling. *Computer Graphics 26*, 2 (1992).

approach should be combined with precise designing tools. Adding the ability to merge curves using Boolean operations will allow one to merge a circle with the tip of the wing, making it feasible to precisely design the cross-section of the missile. The techniques presented herein with the aid of the NUBS representation can be easily enhanced to support these features.

The method presented here is incapable of providing a continuous resolution control. The use of linear interpolation between two adjacent levels alleviates this somewhat, but does not solve it. A complete continuous control would require introducing new knot sequences into the modified curve, knot sequences that would change dynamically as the user employs the continuous resolution control. This issue deserves separate investigation.

The development of an efficient multiresolution decomposition for rational NURBs curves is an intriguing question, considering the complexity that will be introduced into the algebraic summation, so easily computable for polynomials. Providing such a decomposition will make multiresolution methods applicable to the vast majority of curve representations used by contemporary solid modeling systems.

References

[1] Celniker, G., and Gossard, D. Deformable curve and surface finite-elements for free-form shape design. *Computer Graphics 25*, 4 (1991).

[2] Cobb, E. *Design of Sculptured Surfaces using the B-spline Representation.* PhD thesis, University of Utah, June 1984.

[3] Conner, D., Snibble, S., Herndon, K., Robbins, D., Zeleznik, R., and Van-Dam, A. Three-dimensional widgets. In *Proceedings of the Symposium on Interactive 3D Graphics* (1992).

[4] Daehlen, M., and Lyche, T. Decomposition of splines. In *Mathematical Methods in Computer Aided Geometric Design II*, Lyche and Schumaker, Eds. Academic Press, 1992.

[5] Elber, G. and Gotsman, C. Multiresolution control for nonuniform B-spline curve editing: Demonstration Videocassette, (VT00758), Hewlett-Packard Laboratories, 1994.

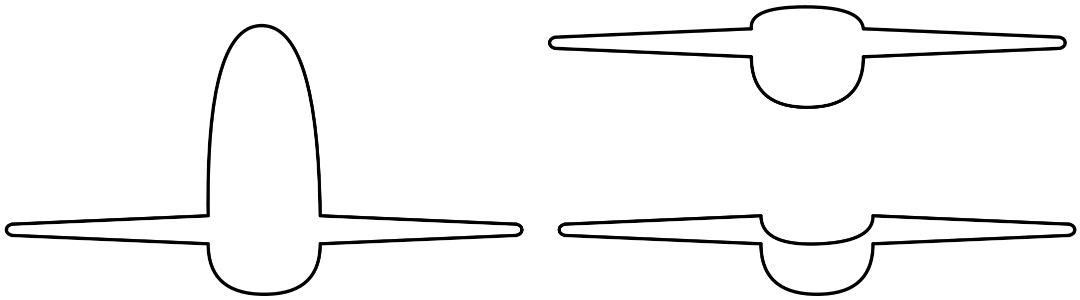

Figure 5: Three curves all derived from the same airplane cross-section, freely editing the fuselage while not crossing the C^1 discontinuity to the wings.

curve to the original curve directly. The multiresolution decomposition need not be computed explicitly and only the subspaces of the decomposition, induced by the decimated knot sequences, need be defined. Such an approach does not allow the display of the multiresolution decomposition of the edited curve and, as such, its usefulness should be investigated.

5 The Videotape

The VHS/NTSC videotape accompanying this paper [5] demonstrates the methods proposed here, which we implemented in an interactive X/Motif-based curve editing system.

The videotape shows multiresolution manipulation of four curves, three of which appear in Figures 1 (star shaped curve), 2 (signature curve), and 3 (cross-section of an airplane). For the star shaped curve and the signature curve, we show the different low-resolution curves obtained on a continuous resolution scale, and the effect obtained by editing at various resolutions. In the airplane curve as well as the fourth curve (a cross-section of a face), we also show adaptive refinement and local control capabilities.

6 Conclusion

We have presented a scheme for computing multiresolution decompositions, and their use for the editing of NUBS. Extending the approach of [6] from uniform to nonuniform B-spline curves not only makes it more practical for existing CAD systems to use, but also provides the mechanism for local refinement and adaptive local curve manipulation.

Editing the cross-section and adding the missiles in Figure 5, it is quite clear that this

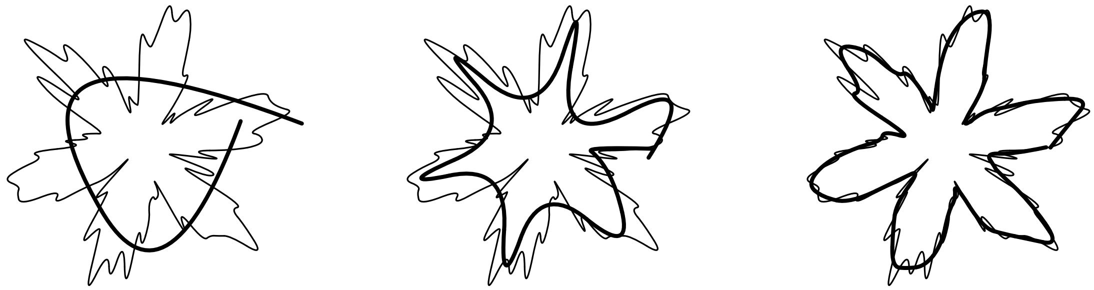

Figure 4: Three levels of a multiresolution decomposition of the closed curve as in Figure 1, with open end conditions. The multiresolution decomposition introduces open curves. The original curve is shown in thin lines throughout while the low resolution versions are shown in thick lines.

Unlike the open or floating end condition, the knots are not independent any more. Equation (5) is not satisfied when we arbitrarily remove interior knots because the end condition knots, $t_j, \quad 0 \leq j \leq n-1$ or $l_i \leq j \leq l_i + n - 1$, were previously unmodified. When interior knots are removed by Algorithm 1, one must translate knots $t_j, \quad 0 \leq j \leq n-2$ or $l_i + 1 \leq j \leq l_i + n - 1$, so that Equation (5) holds. Knots t_{n-1} and t_{l_i} must remain unmodified to preserve the parametric domain of the curve.

4.2 Maintaining Discontinuities

Dealing with NUBS, one can easily handle multiple knots and represent and preserve discontinuities in the edited curve for all the resolutions of the decomposition, by coercing all knot vectors, τ_i, to preserve the discontinuity. Figure 5 shows the same airplane cross-section as Figure 3(a), in which the fuselage and the wings are only C^0 continuous. In Figure 5, the C^1 discontinuities are preserved and the user is provided with optional control that does not cross C^1 discontinuities, as the figure demonstrates.

4.3 Optimization

Once a multiresolution curve is modified at a certain resolution, the B-spline curve is computed as an algebraic summation, as in Equation (1). This process involves the refinement of the low resolution curves, $C_i(t), \quad 0 \leq i < k$, so they are all in the same space V_k. Then, vector addition can take place in V_k. This process was found to be sufficiently fast to be exploited in interactive design. However, one can optimize this computation [9] by adding the modified low-resolution

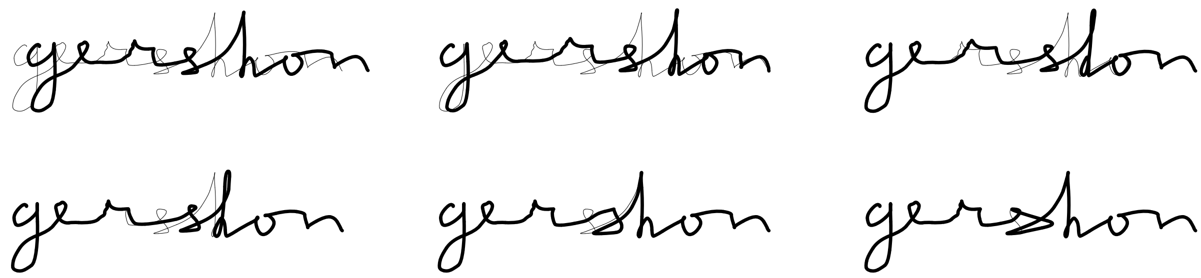

Figure 2: Editing the same curve location, the center of the s in the signature, at different resolution levels, from the lowest (top left) to the highest resolution (right bottom). The original curve is shown in thin lines and the low resolution curves are displayed in thick lines.

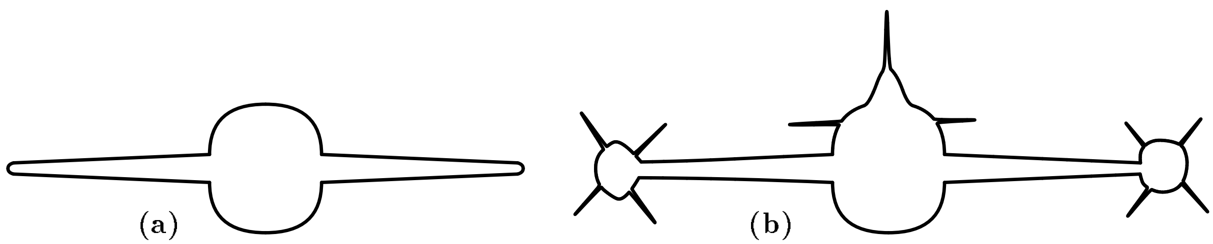

Figure 3: (a) Low resolution quadratic B-spline curve with 22 control points, representing a cross-section of a airplane. (b) The curve of (a) is locally refined to create the degrees of freedom that are necessary to interactively model the missiles at the wing tips, the elevators and the steering wings, as well as the cockpit.

4 Extensions

4.1 Periodic Closed Curves

Figure 1 shows some levels of the multiresolution decomposition of a closed curve. Closed curves are common in modeling. For example, cross-sectional curves are frequently employed in surface constructions. It is essential that low-resolution versions of a given closed curve also be closed. When using B-spline curves with open or floating end conditions, the curves, $C_i(t)$, of the multiresolution decomposition are not guaranteed to be closed and, in general, are not. Figure 4 shows some curves from a decomposition similar to the one of Figure 1 but with open end conditions. Editing one end of the curve at any resolution that affects the end points will almost certainly break the curve open. Using periodic B-spline curves one can solve this problem.

Consider a periodic curve $C_k(t)$ of order n and l_k control points. The knot removal procedure, when τ_i is constructed from τ_{i+1}, must satisfy the following, for all τ_i:

$$t_{j+1} - t_j = t_{l_i+j+2-n} - t_{l_i+j+1-n}, \quad t_{l_i+j+1} - t_{l_i+j} = t_{n+j} - t_{n+j-1}, \quad 0 \le j \le n-2. \qquad (5)$$

device and dragged in the preferred direction, M_0. P_0 combined with $M_0\mathcal{S}$ are applied to the lower-resolution curve $C_j(t)$ at the preselected resolution level j.

Assume $\mathcal{S} \equiv 1$, i.e. no sensitivity control. By adding M_0 to all the control points of $C_j(t)$, we form a translated curve $\hat{C}_j(t)$,

$$\hat{C}_j(t) = C_j(t) + M_0 = \sum_{i=0}^{l_j-1} P_i B_i(t) + M_0 = \sum_{i=0}^{l_j-1} (P_i + M_0) B_i(t) \tag{4}$$

and, in particular for some $t = t_0$, $\hat{C}_j(t_0) = C_j(t_0) + M_0$.

Given a user-selected location on the curve P_0, the parameter value t_0 minimizing $\|C_k(t_0) - P_0\|_2$ is identified. The basis functions contributing to $C_k(t_0)$ are then isolated and their contribution computed. At most n basis function are nonzero at t_0. Each control point is then translated in the direction of $M_0\mathcal{S}$ by an amount proportional to its contribution to $C_k(t_0)$.

Clearly, because we do not translate all points by M_0, the new curve will not be equal to $C_k(t_0) + M_0$ at t_0. However, by weighing the contribution of each control point to $C_k(t_0)$, we more fairly distribute the modification the user introduces to the curve at t_0. While we lose the interpolation ability, we found that the use of a sensitivity control, $\mathcal{S}$, to scale M_0 and the level of interaction involved more than compensates for this loss.

Figure 2 shows a signature curve modified at different resolution levels. Editing was performed at the same location, P_0, the center of the s character, with the same translation amount M_0. As in [6], we linearly interpolate between two neighboring resolutions to provide a notion of continuous resolution control. Clearly, a linear combination of a curve $C_i(t) \in V_i$ and a curve $C_{i+1}(t) \in V_{i+1}$ results in a new curve still in V_{i+1}. However, this simple emulation gives the practical look and feel of continuous behavior and was found to be more than satisfactory.

The ability to handle multiresolution NUBS curves allows one to not only edit and manipulate them at different resolutions, but also to provide local refinement control, inserting new knots into the curve in the neighborhood of the domain to be manipulated. A fixed number of knots are heuristically inserted in the neighborhood of the refined location while preserving continuity as much as possible by prohibiting knot multiplicities. Figure 3(a) shows a low resolution cross-section of a airplane, edited and refined to form missiles at its wing tips as well as its elevators, steering wings, and cockpit in Figure 3(b).

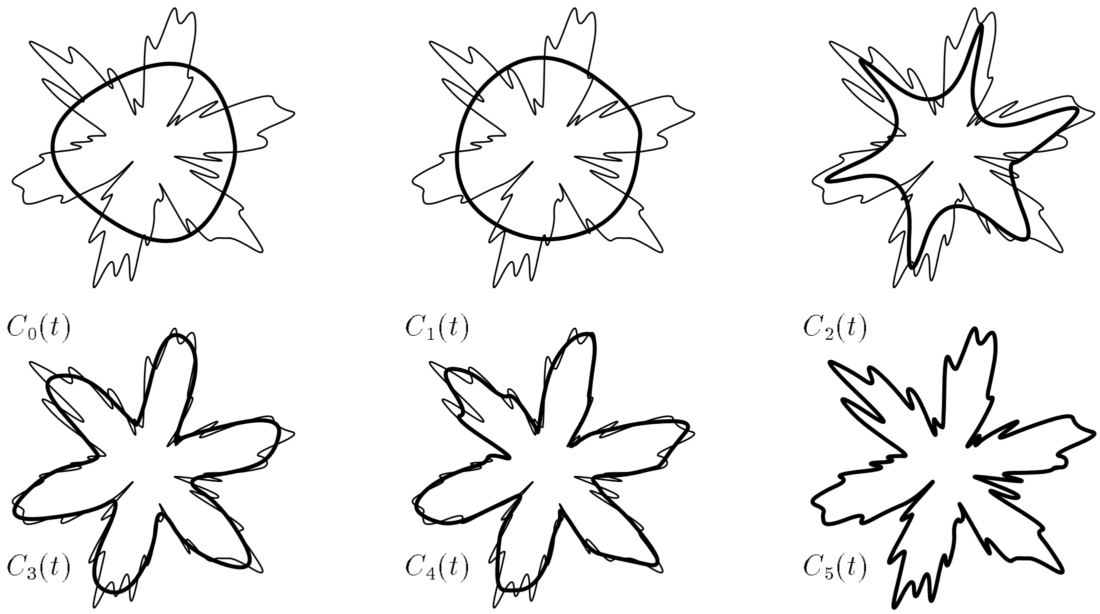

Figure 1: Six levels, $C_i(t)$, $0 \leq i \leq 5$, of the multiresolution decomposition of a B-spline star curve $C_5(t)$ of order 3 defined using 100 control points. The original curve is shown in thin lines throughout while $C_i(t)$'s are shown in thick lines.

interactive speeds. Figure 1 shows the multiresolution decomposition for a star shaped curve, six levels in all. The original quadratic B-spline curve had 100 control points.

Unlike the wavelet basis function used for uniform B-splines, the number of coefficients obtained by our multiresolution decomposition of NUBS is greater than the number of coefficients representing the original curve $C_k(t)$, which is precisely l_k. However, this number is bounded by $2l_k$, since,

$$l_k + \frac{l_k}{2} + \frac{l_k}{4} + \cdots + \frac{l_k}{2^k} < 2l_k, \tag{3}$$

where, in fact, $\frac{l_k}{2^k}$ cannot be less than n. Hence, we at most double the number of coefficients.

3 Manipulation of Multiresolution Decompositions

A major reason for applying a multiresolution decomposition to B-spline curves is to obtain the ability to manipulate and edit the curve at different resolutions. In our implementation, we waive the exact interpolation or constraint based design paradigm [1, 12] in favor of a direct freeform editing style.

The user preselects the resolution j, $0 \leq j \leq k$, he/she would like to edit the curve at, and the sensitivity, $\mathcal{S}$, of the editing process. A point P_0 on the curve is then picked using the input

Algorithm 1 `Multiresolution decomposition of a B-spline curve.`

```
Input:
    C_k(t), a NUBS curve.
Output:
    C_0(t),  D_i(t),  0 ≤ i ≤ k − 1, the multiresolution decomposition of C_k(t).
Algorithm:
    τ_k ⇐ Knot sequence of C_k(t);
    for i = k − 1 to 0 step -1 do
        τ_{k−1} ⇐ half the knots of τ_k, preserving end conditions;
    C_0(t) ⇐ Least squares approximation of C_k(t) in V_o, defined over τ_o;
    for i = 1 to k do
    begin
        C_k(t) ⇐ C_k(t) − C_{i−1}(t);
        C_i(t) ⇐ Least squares approximation of C_k(t) in V_i, defined over τ_i;
        D_{i−1}(t) ⇐ C_i(t) − C_{i−1}(t);
    end;
```

solved using least-squares methods. Algorithm 1 summarizes the multiresolution decomposition process.

While the $C_i(t)$ are regular Euclidean curves, the $D_i(t)$ are vector fields curves that can be used to reconstruct $C_j(t)$,

$$C_j(t) = C_0(t) + \sum_{i=0}^{j-1} D_i(t), \tag{2}$$

In [4], the problem of knot sequence decimation was considered for data reduction purposes. Knots are selected for removal by weighing their possible effect on the curve. However, herein, one would like to minimize the local effect on the curve due to knot removals from level i to level $i+1$. Hence, consecutive knots should not be removed in one step. Removing every n'th knot, where n is the curve order, will cause the least change from one level to the next, yet affect the entire curve. As the degree of a Bézier or B-spline curve is increased, the curve is becoming smoother and smoother due to the low pass property of the basis functions of the representation. Therefore, as n increases, by selecting every n'th knot for removal, we remove knots at larger intervals yet the curve becomes smoother. In practice, we found that removing every second knot still retains a sufficient number of resolution levels to enable an effective multiresolution control. Moreover, the computational overhead required for the algebraic summation is kept at handle periodic NUBS curves and maintain discontinuities. Finally, we conclude and discuss other possible extensions in Section 6.

2 Least Squares Multiresolution Decomposition

Let $C_k(t)$ be a B-spline curve of order n and l_k control points, defined over the knot vector τ_k, where $k \in \mathcal{Z}^+$ will be defined more precisely shortly. Denote by V_k the (linear) space induced by τ_k. Let $\tau_{k-1} \subsetneq \tau_k$. Clearly, the new space induced by τ_{k-1}, denoted V_{k-1}, is a strict subspace of V_k. Let $C_{k-1}(t) \in V_{k-1}$ be the least squares approximation in V_{k-1}, of $C_k(t)$, and their difference be the *detail* $D_{k-1}(t) = C_k(t) - C_{k-1}(t)$, $D_{k-1}(t) \in V_k$.

This process of decomposing a curve into two parts, one low resolution approximation and one high resolution detail, can be applied recursively. $C_k(t)$ could then be expressed as

$$C_k(t) = C_0(t) + \sum_{i=0}^{k-1} D_i(t), \tag{1}$$

where $C_0(t) \in V_0$ and $D_i(t) \in V_{i+1}$.

In order to construct a multiresolution decomposition of a NUBS curve as in Equation (1), the knot sequences τ_i, inducing the subspaces V_i, must first be defined. Given τ_k, construct a set of knot vectors τ_i, $0 \le i < k$, such that $\tau_i \subsetneq \tau_{i+1}$ and $2|\tau_i| \approx |\tau_{i+1}|$, where $|\cdot|$ denotes the size of the knot vector. The end conditions of the original curve must be preserved, hence the knots $t_j \in \tau_i$, $0 \le j < n$ or $l_i \le j < l_i + n$, $\forall 0 \le i < k$ are unmodified, where l_i denotes the number of control points defining $C_i(t)$ over τ_i. In general, $l_i = |\tau_i| - n$. This knot decimation process defines the function space hierarchy and is independent of the specific curve being decomposed.

For a B-spline curve $C_k(t)$ with knot vector τ_k of size 2^k, k subspaces will be constructed, each induced by approximately half the knots of the previous level. The lowest resolution approximation $C_0(t)$ is a single polynomial curve, i.e. τ_0 has no interior knots ($|\tau_0| = 2n$). Least squares techniques [11] are employed to find the curve $C_i(t) \in V_i$, defined over τ_i, best approximating $C_k(t)$. Indeed, $C_k(t)$ is uniquely determined using l_k constraints defined at its l_k node parameter values. Thus, the approximation problem may be reduced to a set of l_k linear equations in l_i unknowns by sampling $C_k(t)$ at its l_k node parameter values, which can be reasoning recursively k stages, we arrive at a space V_k of dimension $2^k n$, such that,

$$V_0 \subset V_1 \subset \cdots V_k \quad ,$$

and

$$V_k = V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{k-1} \quad .$$

Given a vector (B-spline curve) $C_k(t) \in V_k$, its low-resolution versions $C_0(t), .., C_{k-1}(t)$ are projections of $C_k(t)$ onto the lower dimensional spaces $V_0, .., V_{k-1}$. $C_k(t)$ can be expanded:

$$C_k(t) = C_0(t) + \sum_{i=0}^{k-1} D_i(t),$$

where $D_i(t)$ is $C_k(t)$'s projection on W_i. Multiresolution B-spline manipulation follows by decomposing a B-spline curve $C_k(t)$ to a low-resolution version $C_j(t)$ for some $0 \leq j \leq k$, and details $D_i(t)$, $j \leq i \leq k-1$, performing some operation on $C_j(t)$, and adding all $D_i(t)$, $j \leq i \leq k-1$ back in.

While uniform B-spline basis functions are a powerful mathematical representation, they cannot model many important geometric features, of which the most important are, undoubtedly, C^1 discontinuities. To solve this, we extend the work of [6] to nonuniform B-spline curves (NUBS). Retaining the flavor of that work, we decompose NUBS curves by a least squares approximation process to NUBS defined over the subspaces V_i, and their successive algebraic differences, as suggested in [4] for data reduction purposes. Decimation of nonuniform knot sequences is done in a manner similar to that of [4]. Manipulation is done at a lower resolution selected by the user, and then the extraneous detail is algebraically added back in.

The support of multiresolution NUBS allows us to handle general B-spline curves. This added flexibility is crucial not only because the nonuniform B-spline representation is common in contemporary geometric modeling systems, but also because we are now able to add details using an adaptive refinement process as well as properly maintain discontinuities.

This paper is organized as follows. Section 2 describes the computation of the least squares approximation in the multiresolution decomposition of a NUBS curve. In Section 3, we discuss editing and manipulating multiresolution NUBS and in Section 4 we extend the support to were used as weights to affect the control points. This approach has a fundamental difficulty when two separate portions of the curve are close. Constraint based approaches were proposed recently [1, 12] to alleviate the difficulty in editing freeform shapes while matching engineering specifications. Direct and interactive manipulation tools of freeform curves and surfaces have also been investigated [3].

The piecewise polynomial B-spline representation is common in many contemporary geometric modeling systems. While this is a powerful mathematical tool with many desirable properties, these same properties impose undesirable constraints on the user. For example, the B-spline representation's most attractive property, namely, locality, does not allow the user to easily perform global operations on the object being modeled. An operation affecting the entire curve must be translated (usually manually, by the user) into a series of local operations, each affecting only a small portion of the curve. The ability to simultaneously perform both local and global operations at will would add significant functionality to any modeling system. It is common to call operations affecting the entire object *low-resolution* operations, and those affecting just a small portion of it *high-resolution* operations. It would be desirable to perform manipulation at any resolution level, on a continuous scale. Multiresolution decomposition of curves has been proposed as a tool that can provide simultaneous global and local curve control.

Multiresolution methods have been used in a variety of systems in one way or another [6, 7, 8, 10, 12]. The approach we present herein generalizes [6], where an extension of the standard uniform B-spline basis to a multiresolution uniform B-spline wavelet basis is introduced. For completeness' sake, we describe the underlying theory of that extension here.

Any B-spline curve on a unit interval knot sequence of length n can obviously be represented as a B-spline curve on a half-unit interval knot sequence, but not the opposite. This manifests in the latter linear space, denoted V_1, having dimension $2n$, while the former, denoted V_0, has dimension n. Geometrically, piecewise polynomials in V_0 are able to represent discontinuities at the unit knots whereas piecewise polynomials in V_1 are able to represent discontinuities at the half-unit knots as well as at the unit knots. The extra power of a half-unit knot sequence is conveyed by the complement space W_0, such that $V_0 \oplus W_0 = V_1$. Continuing this line of

Multiresolution Control for Nonuniform B-spline Curve Editing

Gershon Elber Craig Gotsman
CRC - HP Israel Science Center
Technion City,
Haifa 32000, Israel.
e-mail: {gershon|gotsman}@hp.technion.ac.il

Keywords: Least Squares, Direct Manipulation, Hierarchical Curve Representation.

Abstract

The piecewise polynomial B-spline representation is widely used throughout the CAGD community as the representation of choice. However, the locality of B-spline curves, while important in many respects, disables global control of the curve, preventing efficient and easy manipulation. Multiresolution representations for uniform B-spline curves have been recently proposed to alleviate this problem.

Herein, we extend the use of multiresolution representations to nonuniform B-spline (NUBS) curves, including periodic curves. Our method supports local nonuniform refinement and (dis)continuity preservation. The multiresolution decomposition of the freeform NUBS curve is computed using least-squares approximation, based on existing data reduction techniques.

The least-squares decomposition allows us to support NUBS curves, but it also imposes some preprocessing penalties in both time and space compared to techniques for multiresolution uniform B-spline curves. Nonetheless, the entire process is fast enough to enable interactive editing of complex NUBS curves, as is demonstrated by an interactive editor implemented to test our methods.

1 Introduction

It is widely agreed that the construction of efficient, intuitive, and interactive editors for geometric objects is a fundamental objective, yet difficult challenge, in the field of geometric modeling. Most freeform geometric modeling systems allow the user to work in the framework of a specific data model, e.g. Bezier, or nonuniform rational B-splines (NURBS) forms. This tightly constrains the set of geometric manipulation operations allowed, the man-machine interface, and the type of objects which can be modeled.

Different curve manipulation techniques have been proposed on many occasions. In [2], the Euclidean distances between the point of modification and the control points of a B-spline curve