Multiresolution Curve Editing With Linear Constraints*

Gershon Elber
Computer Science Department,
Technion, Haifa 32000, Israel
e-mail: gershon@cs.technion.ac.il

The use of multiresolution control toward the editing of freeform curves and surfaces has already been recognized as a valuable modeling tool [1–3]. Similarly, in contemporary computer aided geometric design, the use of constraints to precisely prescribe freeform shape is considered an essential capability [4,5]. This paper presents a scheme that combines multiresolution control with linear constraints into one framework, allowing one to perform multiresolution manipulation of nonuniform B-spline curves, while specifying and satisfying various linear constraints on the curves. Positional, tangential, and orthogonality constraints are all linear and can be easily incorporated into a multiresolution freeform curve editing environment, as will be shown. Moreover, we also show that the symmetry as well as the area constraints can be reformulated as linear constraints and similarly incorporated. The presented framework is extendible and we also portray this same framework in the context of freeform surfaces. [DOI: 10.1115/1.1430679]

Keywords: CAD, Curves & Surfaces, Geometric Modeling, Wavelets

1 Introduction

Building intuitive interactive editing capabilities of freeform curves and surfaces into contemporary CAD systems has been an elusive task for the geometric design community, for a long period of time. While freeform curves were introduced to the computer graphics world almost three decades ago, the editing process of freeform geometry continues to be considered a difficult task. Furthermore, direct (control point) manipulation has been recognized as a powerful computer graphics tool, and yet its locality appears to be an Achilles' heel of the B-spline representation, the most commonly used representation of freeform shapes in contemporary geometric modeling systems. In a single direct manipulation operation, one is unable to apply a modification to the B-spline shape that affects more than a local neighborhood.

The hidden capabilities in multiresolution control and editing of freeform geometry have been revealed in [1,2], in the context of uniform B-spline curves and surfaces [1,2] allow the user to directly and interchangeably affect the freeform shape globally as well as locally, in several resolution levels. In [3], a similar solution has been presented that supports the multiresolution editing of nonuniform B-spline curves and surfaces.

The work of [1–3] computes the orthogonal projections of the freeform geometry into lower dimensional spaces, employing a wavelet decomposition of uniform and nonuniform B-spline representations. While fairly simple to compute for the case of uniform knot sequences, this decomposition in the nonuniform case is computationally intensive. Fortunately, one can recognize that the explicit orthogonal decomposition is not really necessary [6], alleviating these computational difficulties in the nonuniform case.

In [7], a multiresolution curve editor that is based on a non-orthogonal decomposition has been presented. The major deficiency of this non-orthogonal decomposition lays in the possibility of the user to conduct many high resolution, fine operations, that can be represented as few low resolution operations. We will discuss this some more in Section 2.

In [5], a surface editing system that satisfies zero dimensional constraints such as positions, tangents, and normals, has been presented. The constraints, being linear, are efficiently solved, allowing for the interactive manipulation of the freeform geometry. [5] also considers *transfinite constraints* where the constraints might have a non zero dimensionality. While some cases might be of finite dimension, such as the containment of a polynomial curve in a polynomial surface when posed as a composition, other cases might necessitate an approximation.

The satisfaction of nonlinear constraints is more difficult than the satisfaction of linear constraints due to the imposed computational demands. Nonlinear constraints that are commonly considered are second order differential constraints such as convexity [8], enclosed volume [9], and first and second order fairing constraint, typically in the form of strain and stress surface shape optimization functionals [5].

The exploitation of first and second differential order constraints, in real time, is also highly intensive computationally. In [10], an interactive surface editing system that supports real time surface manipulation with convexity/developability constraints is reported, with the aid of a careful pre-computation of the curvature fields.

This work presents a two-fold. First, a special attention is given to the class of linear constraints that one can employ. Positional, tangential, and orthogonality constraints are already known to be part of this class. In addition, symmetry constraints are shown to be part of this class of linear constraints. Furthermore, we also show that the enclosed area of a closed curve can be posed as a linear constraint. Then, a synergetic view of the two methodologies of multiresolution control and linear constraints is considered. We combine the revealed capabilities of multiresolution control with the ability of linear constraints to prescribe precise freeform geometry, all in the framework of interactive editing of nonuniform B-spline curves.

Our implementation as well as all the examples shown as part of this work were based on the IRIT solid modeling system [11] that is developed at the Technion, Israel Institute of Technology.

This paper is organized as follows. Section 2 reviews the concepts of multiresolution control, following [1–3,7], that we have been exploiting. Section 3 considers the linear constraints that can be employed as part of this work, while in Section 4, we show how to fuse the two methodologies of multiresolution control and linear constraints into a single synergetic framework. Some examples that demonstrate the expected benefits of this synergy are presented in Section 5. We consider the extensions of this work to

*The research was supported in part by the Fund for Promotion of Research at the Technion, Haifa, Israel.

Contributed by the Computer Aided Product Development (CAPD) Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received Aug. 2001; revised manuscript received Oct. 2001. Associate Editors: D. Anderson, K. Lee.

3D curves on surface in Section 6, to freeform surfaces in Section 7 and to higher order moments in Section 8. Finally, we conclude in Section 9.

2 Multiresolution Editing of Curves

Let

$$C(t)=\sum_{i=0}^{n-1} P_i B_{i,\boldsymbol{\tau},k}(t), \tag{1}$$

be a planar nonuniform B-spline curve of order k, with n control points $P_i=(x_i, y_i)$, that is defined over the knot sequence $\boldsymbol{\tau}=\boldsymbol{\tau}_0$ of length $n+k$,

$$\boldsymbol{\tau}=\{t_0, t_1, \cdots, t_{k-1}, \cdots, t_n, \cdots, t_{n+k-1}\},$$

where the knots t_k through t_{n-1} are denoted the *interior knots*. Let Ψ_0 be the piecewise polynomial function space that is induced by $\boldsymbol{\tau}=\boldsymbol{\tau}_0$. A piecewise polynomial function, $f(t)\in\Psi_0$, is potentially discontinuous, only at the interior knots of Ψ_0.

Let $\boldsymbol{\tau}_i$ be a knot sequence formed by the first and last k knots of $\boldsymbol{\tau}_0$ and a subset of the interior knots. Further, let $\boldsymbol{\tau}_{i+1}\subset\boldsymbol{\tau}_i$. Such a hierarchy of knot vectors, $\{\boldsymbol{\tau}_i\}$, presents the following properties:

- The B-spline curves of order k defined over $\boldsymbol{\tau}_i$ have the domain of $[t_{k-1}, t_n)$, $\forall i$.
- The piecewise polynomial function space, Ψ_{i+1}, defined by $\boldsymbol{\tau}_{i+1}$, is strictly contained in Ψ_i.

In this work, $\boldsymbol{\tau}_{i+1}$ is selected to hold a half of the interior knots of $\boldsymbol{\tau}_i$, using every other interior knot of $\boldsymbol{\tau}_i$.

Clearly, the domain of all the curves in all the subspaces of Ψ_i is the same because we kept the first and last k knots unmodified. Consider $t_j\in\boldsymbol{\tau}_i$ with multiplicity $m_{j,i}$ such that $t_j\notin\boldsymbol{\tau}_{i+1}$. Then, a piecewise polynomial function $g(t)\in\Psi_{i+1}$ is C^∞ at t_j whereas $h(t)\in\Psi_i$ is at least $C^{k-m_{j,i}-1}$ continuous at t_j. Alternatively, consider the case of $t_j\in\boldsymbol{\tau}_{i+1}$ but the multiplicity of t_j in $\boldsymbol{\tau}_{i+1}$, $m_{j,i+1}$, is lower than the multiplicity of t_j in $\boldsymbol{\tau}_i$, $m_{j,i}$. Then, the continuity of $g(t)\in\Psi_{i+1}$ at t_j is at least $C^{k-m_{j,i+1}-1}$, whereas the continuity of $h(t)\in\Psi_i$ at t_j is at least $C^{k-m_{j,i}-1}$.

We would like to consider a modification to curve $C(t)$ that starts from $t_s=t_{j_1}$ and ends at $t_e=t_{j_2}$, where $t_{j_1}, t_{j_2}\in\boldsymbol{\tau}_0$. In Ψ_0, this entails the potential updates of all control points P_i, $i=j_1-k+1,\cdots,j_2-1$. The possible need for an update of an arbitrary large number of control points exposes the Achilles' heel of the B-spline representation; the locality property of the representation hinders any attempt to control the shape of the curve at various other resolutions.

Assume the existence of a subspace Ψ_i such that t_{j_1} and t_{j_2} are adjacent to each other, or no other interior knot exists between them. In Ψ_i, the same modification of the curve from $t_s=t_{j_1}$ to $t_e=t_{j_2}$ entails the potential update of exactly k control points, P_i, $i=j_1-k+1,\cdots,j_1$. Let $t_{j_1}<t_u<t_{j_2}$ be the point on $C(t)$ that is selected by the user for the interactive modification and let $\vec{\mathcal{V}}_m$ be the modification vector of $C(t_u)$. That is, $C(t_u)$ should be translated to $C(t_u)+\vec{\mathcal{V}}_m$. Typically, $\vec{\mathcal{V}}_m$ will be computed as the difference between the old and the new mouse positions, in a select-and-drag operation on the curve. Let $\Delta\boldsymbol{\tau}_i(t)\in\Psi_i$, be a polynomial function such that $\Delta\boldsymbol{\tau}_i(t_u)=1$. Then, $C(t)+\vec{\mathcal{V}}_m\Delta\boldsymbol{\tau}_i(t)$ satisfies the modification requirement at $t=t_u$. Further, due to the fact that $\Delta\boldsymbol{\tau}_i(t)$ is a single polynomial between t_{j_1} and t_{j_2}, the effect of the modification will always span t_{j_1} to t_{j_2} and hence will be at the proper resolution.

Having only one constraint on $\Delta\boldsymbol{\tau}_i(t)$, as $\Delta\boldsymbol{\tau}_i(t_u)=1$, there are typically infinitely many solutions to the construction problem of $\Delta\boldsymbol{\tau}_i(t)$. Nonetheless, we also seek to *minimize the change* in $C(t)$ as the result of applying $\Delta\boldsymbol{\tau}_i(t)$. Having support at the proper resolution of Ψ_i that modifies $C(t)$ from t_s to t_e, we would also like to minimize the change outside of this domain. One simple and direct approach at efficiently constructing such a minimal change solution can use the values of the supporting B-spline basis functions at t_u, in Ψ_i, normalized so that $\Delta\boldsymbol{\tau}_i(t_u)$ is indeed one,

$$\Delta\boldsymbol{\tau}_i(t)=\frac{1}{\sigma}\sum_{i=0}^{m} B_{i,\boldsymbol{\tau},k}(t_u)B_{i,\boldsymbol{\tau},k}(t),$$

with,

$$\sigma=\sum_{i=0}^{m}(B_{i,\boldsymbol{\tau},k}(t_u))^2=\sum_{i=J-k+1}^{J}(B_{i,\boldsymbol{\tau},k}(t_u))^2, \quad t_J\leq t_u<t_{J+1},$$

assuming any function in Ψ_i, including $\Delta\boldsymbol{\tau}_i(t)$, has $m+1$ coefficients. Here, only the k basis functions that are non zero at t_u contribute to $\Delta\boldsymbol{\tau}_i(t)$.

$\Delta\boldsymbol{\tau}_i(t)$ could always be raised to $\Delta\boldsymbol{\tau}_0(t)\in\Psi_0$ by inserting the knots [12] of $\{\boldsymbol{\tau}_0\backslash\boldsymbol{\tau}_i\}$. Then, a simple addition of $\Delta\boldsymbol{\tau}_0(t)$ to the original curve, $C(t)\in\Psi_0$,

$$C(t)\leftarrow C(t)+\vec{\mathcal{V}}_m\Delta\boldsymbol{\tau}_0(t), \tag{2}$$

can be materialized as the addition of the respective coefficients of $C(t)$ and $\Delta\boldsymbol{\tau}_0(t)$ (times $\vec{\mathcal{V}}_m$).

In general, however, the specified domain of influence will fall between knots. Having $t_s, t_e\notin\boldsymbol{\tau}_0$, one is required to either insert the new knots, t_s and t_e, into $\boldsymbol{\tau}_0$, or else to approximate the domain of influence. Recognizing the obvious overhead of drastically increasing the number of knots in the modified curve, as the editing process of the curve progresses, an approximation is employed in this work. Let $t_{j_1-1}\leq t_s<t_{j_1}$ and $t_{j_2}\leq t_e<t_{j_2+1}$ be the start and end parameters of the specified domain of influence. Moreover, let Ψ_i be a space such that t_{j_1} and t_{j_2} are adjacent to each other in $\boldsymbol{\tau}_i$ and similarly let Ψ_{i+1} be a space such that t_{j_1-1} and t_{j_2+1} are adjacent to each other in $\boldsymbol{\tau}_{i+1}$. The approximated solution is formed by linearly blending between the reconstructed $\Delta\boldsymbol{\tau}_i(t)$ and $\Delta\boldsymbol{\tau}_{i+1}(t)$ functions. Moreover, and while this blended solution continues to be in Ψ_i, one can independently solve for $\Delta\boldsymbol{\tau}_i(t)$ and $\Delta\boldsymbol{\tau}_{i+1}(t)$.

3 Linear Constraints of Freeform Curves

Constraints are an important tool in design. When a curve must interpolate a certain location or be perpendicular to another curve at some other location, linear constraints can satisfy these demands, all while the user is free to manipulate the curve as he or she see fits. In Section 3.1, we review the basic linear constraints that are traditionally employed. In Section 3.2, we consider the linear symmetry constraint whereas in Section 3.3, the enclosed area is formulated as a (bi)linear constraint.

3.1 Basic Linear Constraints. Recall curve,

$$C(t)=\sum_{i=0}^{n-1} P_i B_{i,\boldsymbol{\tau},k}(t),$$

from Eq. (1). A positional constraint, P, at some parameter value t_p is linear in the control points of the curve, P_i, as it reduces to,

$$P=C(t_p)=\sum_{i=0}^{n-1} P_i B_{i,\boldsymbol{\tau},k}(t_p). \tag{3}$$

Similarly, a tangential constraint, T, at some parameter value t_t, is reduced to the following linear equation in the control points,

$$T=C'(t_t)=(k-1)\sum_{i=0}^{n-2}\left(\frac{P_{i+1}-P_i}{t_{i+k-1}-t_i}B_{i,\boldsymbol{\tau},k-1}(t_t)\right). \tag{4}$$

A normal or orthogonality constraint can be satisfied by constraining the normal field of the planar curve $C(t)=(x(t),y(t))$ as $N(t)=(-y'(t),x'(t))$. Hence, an orthogonality constraint is equivalent to a tangential constraint, for planar curves, with minor differences. Equation (4) completely constrains $C'(t_t)$. In contrast, the tangency constraint could also be prescribed as,

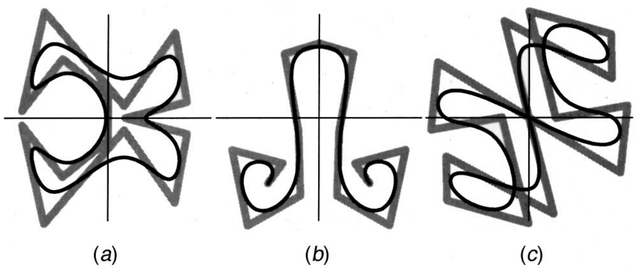

Fig. 1 Symmetries of curves: (*a*) *X*-symmetry, (*b*) *Y*-symmetry, (*c*) Circular-symmetry

$$0=\langle C'(t_t),N\rangle, \quad (5)$$

where N is the direction that $C(t)$ should be orthogonal to, at t_t. In (5), only the direction of $C'(t_t)$ is prescribed whereas in (4), the magnitude of the vector is fully constrained as well. This difference is similar to the difference between G^1 and C^1 continuity.

Unlike Constraints (3) and (4), the inner product constraint of (5) necessitates the derivation of the solution of all axes, simultaneously. Hence, instead of independently solving two sets of constraints (for planar curves), one is forced to simultaneously solve one set of twice as many constraints, satisfying both the x and the y requirements.

To complete our discussion on the basic linear constraints, we should also consider higher order derivatives that are also linear in the control points of the curve and hence can be equally employed. Nonetheless, second order, curvature, constraints are likely to be the only higher order constraints that one might exploit, in practice.

3.2 Symmetry Constraints. An additional linear yet useful constraint we would like to consider as part of this work is *X*-, *Y*-, or Circular-symmetry. Clearly, one can handle one half of the shape only to be reflected to yield the symmetry. Nevertheless, continuity along the joint of the two halves must be then preserved and knots must be unnecessarily introduced, along the symmetry line. Moreover, such a reflection of the curve induces a need to treat the symmetry joint location as a special case, an excessive requirement from the curve editing tool which we would like to refrain from. Hence, we would rather express the symmetry as a linear constraint. Assume the domain of curve $C(t)$ is $t\in[0,1]$ and let $\boldsymbol{\tau}$ be a symmetric knot sequence. That is, $t_i=1.0-t_{n+k-i-1}$, $\forall\, t_i\in\boldsymbol{\tau}$.

Consider the major axes. Then, we say that $C(t)=(x(t),y(t))$ is (See Fig. 1),

$$\begin{aligned}
&X\text{-sym. if } x(t)=x(1-t) \text{ and } y(t)=-y(1-t),\\
&Y\text{-sym. if } y(t)=y(1-t) \text{ and } x(t)=-x(1-t),\\
&\text{Circ.-sym if } x(t)=-x(1-t) \text{ and } y(t)=-y(1-t). \quad (6)
\end{aligned}$$

These symmetry constraints reduce to similar constraints over the control polygon. Having a symmetric knot sequence, we have, $B_{i,\boldsymbol{\tau},k}(1.0-t)=B_{n-1-i,\boldsymbol{\tau},k}(t)$. Then,

$$\begin{aligned}
C(1.0-t)&=\sum_{i=0}^{n-1} P_iB_{i,\boldsymbol{\tau},k}(1.0-t)=\sum_{i=0}^{n-1} P_iB_{n-1-i,\boldsymbol{\tau},k}(t)\\
&=\sum_{i=0}^{n-1} P_{n-1-i}B_{i,\boldsymbol{\tau},k}(t). \quad (7)
\end{aligned}$$

Now, for example, consider the *X*-symmetry case. If $x(t)=x(1-t)$, from Eq. (7) we have,

$$\sum_{i=0}^{n-1} x_iB_{i,\boldsymbol{\tau},k}(t)=\sum_{i=0}^{n-1} x_{n-1-i}B_{i,\boldsymbol{\tau},k}(t),$$

or

$$0=\sum_{i=0}^{n-1} (x_i-x_{n-1-i})B_{i,\boldsymbol{\tau},k}(t),$$

which immediately reduces to the set of linear constraints of the form,

$$0=x_i-x_{n-1-i}, \quad i=0,\cdots,\left|\frac{n}{2}\right|-1, \quad (8)$$

due to the independence of the B-spline basis functions. Therefore, the symmetry constraint of $x(t)=x(1-t)$ as reduced to Eq. (8) could be posed as $\lfloor n/2\rfloor$ linear constraints in the coefficients of the X axis, x_i.

Similarly, a constraint of the form $y(t)=-y(1-t)$ could be reduced to $\lceil n/2\rceil$ linear constraints in the coefficients of the Y axis, y_i, as,

$$0=y_i+y_{n-1-i}, \quad i=0,\cdots,\left|\frac{n}{2}\right|-1. \quad (9)$$

Obviously, the axes of symmetry need not be the major axes and, for example, a constraint of *X*-symmetry around the horizontal line $y=Y_0$ is reduced to a set of $\lceil n/2\rceil$ constraints of the form $2Y_0=y_i+y_{n-1-i}$.

3.3 Area Constraint of a Closed Planar Curve. We now consider the area as a constraint on a closed freeform planar curve. The enclosed area or volume of a closed curve or a closed surface, respectively, were considered in the context of vision [13] as well as geometric modeling [14]. Presented as a non linear problem, approximation methods are typically employed toward the computation of the enclosed property.

Let $C(t)=(x(t),y(t))$ be a regular closed planar parametric curve. Employing Green's theorem, the (signed) area, $\mathcal{A}$, enclosed by $C(t)$ equals (See, for example [13,14]),

$$\mathcal{A}=\frac{1}{2}\oint -x'(t)y(t)+x(t)y'(t)dt=\frac{1}{2}\oint |C(t)\times C'(t)|dt, \quad (10)$$

where $|C(t)\times C'(t)|$ denotes the cross product's determinant. Having the x and y components of this cross product vanish to zero because the curve is planar, this determinant is reduced to the z component only. Hereafter, we use $C(t)\times C'(t)$ to denote this scalar z component.

A geometric interpretation of Eq. (10) can be found in Fig. 2. Herein, we are interested in evaluating this equation as efficiently as possible when $C(t)$ is a B-spline curve. Moreover, we are pursuing this computation in the context of (linear) constraint satisfaction, in real time interaction.

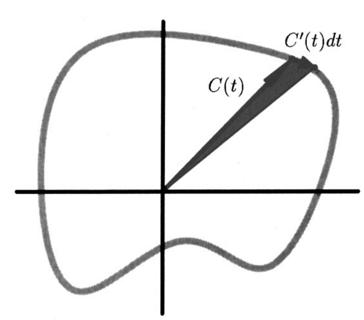

Fig. 2 The area of a closed parametric curve. The differential area in gray equals to $\frac{1}{2}|C(t)\times C'(t)|dt$ and the enclosed area by the curve is the result of integrating this differential area over the entire parametric domain of the curve. See Eq. (10)

Let $C(t)$ be a B-spline curve, $C(t)=\Sigma_{i=0}^{n-1}P_iB_{i,k}(t)$, where $P_i=(x_i,y_i)$ and $B_{i,k}(t)$ is the i'th B-spline basis function of order k. Then, the enclosed area can be rewritten as the *bilinear form* of,

$$
\begin{aligned}
2\mathcal{A}&= \oint -\sum_i x_iB'_{i,k}(t)\sum_j y_jB_{j,k}(t)\\
&\quad+\sum_i x_iB_{i,k}(t)\sum_j y_jB'_{j,k}(t)dt\\
&=\sum_i x_i\sum_j y_j \oint -B'_{i,k}(t)B_{j,k}(t)+B_{i,k}(t)B'_{j,k}(t)dt\\
&=[x_0,x_1,\cdots,x_{n-1}]\\
&\quad\begin{bmatrix}\phi_{0,0} & \phi_{0,1} & \cdots & \phi_{0,n-1}\\ \phi_{1,0} & \phi_{1,1} & \cdots & \phi_{1,n-1}\\ \vdots & \vdots & \ddots & \vdots\\ \phi_{n-1,0} & \phi_{n-1,1} & \cdots & \phi_{n-1,n-1}\end{bmatrix}\begin{bmatrix}y_0\\ y_1\\ \vdots\\ y_{n-1}\end{bmatrix}\\
&=\mathbf{X\Phi Y},
\end{aligned}
\tag{11}
$$

where

$$
\begin{aligned}
\phi_{ij}&= \oint -B'_{i,k}(t)B_{j,k}(t)+B_{i,k}(t)B'_{j,k}(t)dt\\
&= \oint -(k-1)\left(\frac{B_{i,k-1}(t)}{t_{i+k-1}-t_i}-\frac{B_{i+1,k-1}(t)}{t_{i+k}-t_{i+1}}\right)B_{j,k}(t)\\
&\quad+(k-1)\left(\frac{B_{j,k-1}(t)}{t_{j+k-1}-t_j}-\frac{B_{j+1,k-1}(t)}{t_{j+k}-t_{j+1}}\right)B_{i,k}(t)dt.
\end{aligned}
\tag{12}
$$

Equation (11) sheds some light on our objectives. Assume the y_i coefficients of the constrained curve are fixed. Then, the area constraint is linear in the x_i coefficients! Similarly, if the x_i coefficients of the constrained curve are fixed, the area constraint is linear in the y_i coefficients.

This crucial view of the area equation allows us not only to enforce a prescribed area as a linear constraint, but also to precompute all the coefficients of the $\mathbf{\Phi}$ matrix. A typical editing session of a freeform shape starts with a prescription of the specific function space, Ψ_0, or knot sequences. Only then, the shape is modified, for example via a direct or a control point select-and-drag operation. Once the function space, Ψ_0, of the curve is prescribed, the $\mathbf{\Phi}$ matrix can be clearly computed. The computation of integrals and products of nonuniform B-spline basis functions is considered, for example, in [3].

3.3.1 Area Constraint of Linear B-spline Curves. It is interesting to examine this derived area constraint in the context of linear B-spline curves. Then, the B-spline curve is reduced to a polygon and hence we expect the constraint to reduce to the equation of the area of a polygon with n vertices. For the linear B-spline case where $k=2$, Eq. (12) becomes,

$$
\begin{aligned}
\phi_{ij}&= \oint -\left(\frac{B_{i,1}(t)}{t_{i+1}-t_i}-\frac{B_{i+1,1}(t)}{t_{i+2}-t_{i+1}}\right)B_{j,2}(t)\\
&\quad+\left(\frac{B_{j,1}(t)}{t_{j+1}-t_j}-\frac{B_{j+1,1}(t)}{t_{j+2}-t_{j+1}}\right)B_{i,2}(t)dt.
\end{aligned}
\tag{13}
$$

Clearly if $|i-j|\geqslant 2$ and due to the final support of the B-spline basis functions, Eq. (13) is zero. Moreover, if $i=j$ and due to the antisymmetry of the integrand in Eq. (13), ϕ_{ii} is also zero. Hence, we only need to derive $\phi_{i,i\pm1}$:

$$
\begin{aligned}
\phi_{i,i+1}&= \oint -\left(\frac{B_{i,1}(t)}{t_{i+1}-t_i}-\frac{B_{i+1,1}(t)}{t_{i+2}-t_{i+1}}\right)B_{i+1,2}(t)\\
&\quad+\left(\frac{B_{i+1,1}(t)}{t_{i+2}-t_{i+1}}-\frac{B_{i+2,1}(t)}{t_{i+3}-t_{i+2}}\right)B_{i,2}(t)dt\\
&= \oint \frac{B_{i+1,1}(t)}{t_{i+2}-t_{i+1}}B_{i+1,2}(t)+\frac{B_{i+1,1}(t)}{t_{i+2}-t_{i+1}}B_{i,2}(t)dt\\
&= \oint \frac{B_{i+1,1}(t)}{t_{i+2}-t_{i+1}}(B_{i+1,2}(t)+B_{i,2}(t))dt\\
&=\int_{t_{i+1}}^{t_{i+2}}\frac{1}{t_{i+2}-t_{i+1}}dt=1.
\end{aligned}
\tag{14}
$$

$\phi_{ij}=-\phi_{ji}$ (See Eq. (12)) or Φ is an antisymmetric matrix. Hence, $\phi_{i,i-1}=-1$, and we have shown that the area of a closed polygon $\mathcal{P}=\{P_i\}_{i=0}^{n-1}$, $P_i=(x_i,y_i)$ equals,

$$
\mathcal{A}=\frac{1}{2}\left[x_0,x_1,\cdots,x_{n-1},x_0\right]
\begin{bmatrix}0 & 1 & 0 & 0 & \cdots & 0\\ -1 & 0 & 1 & 0 & \cdots & 0\\ 0 & -1 & 0 & 1 & \cdots & 0\\ 0 & 0 & -1 & 0 & \cdots & 0\\ \vdots & \vdots & \vdots & & \ddots & \vdots\\ 0 & 0 & \cdots & -1 & 0 & 1\\ 0 & 0 & 0 & \cdots & -1 & 0\end{bmatrix}
\begin{bmatrix}y_0\\ y_1\\ \vdots\\ y_{n-1}\\ y_0\end{bmatrix},
\tag{15}
$$

which is the same as,

$$
\mathcal{A}=\frac{1}{2}\left(\begin{vmatrix}x_0 & x_1\\ y_0 & y_1\end{vmatrix}+\begin{vmatrix}x_1 & x_2\\ y_1 & y_2\end{vmatrix}+\cdots+\begin{vmatrix}x_{n-1} & x_0\\ y_{n-1} & y_0\end{vmatrix}\right),
\tag{16}
$$

as, for example, in [15]. This linear B-spline case allows one to edit closed planar polygonal domains while coercing the enclosed area of the polygon to be the same throughout the editing process, via a *bilinear constraint* over the vertices of the polygon.

3.3.2 Area Constraint in Interactive Multiresolution Editing. Unlike other linear constraints, the area contribution of modification $\Delta\boldsymbol{\tau}_i(t)$ depends on the current shape of curve $C(t)$ as well. Let $C(t)=(x(t),y(t))=(\Sigma x_iB_i(t),\Sigma y_iB_i(t))$ be the current closed curve under editing with area $\mathcal{A}$. Let $\Delta\boldsymbol{\tau}_i(t)=(\delta_x(t),\delta_y(t))=(\Sigma\delta_{x_i}B_i(t),\Sigma\delta_{y_i}B_i(t))$ be the modification one would like to add to $C(t)$ as,

$$
D(t)\Leftarrow C(t)+\Delta\boldsymbol{\tau}_i(t).
$$

The area, $\hat{\mathcal{A}}$, of the modified curve, $D(t)$, equals,

$$
\begin{aligned}
2\hat{\mathcal{A}}&= \oint D(t)\times D'(t)dt\\
&= \oint (C(t)+\Delta\boldsymbol{\tau}_i(t))\times(C(t)+\Delta\boldsymbol{\tau}_i(t))'dt\\
&=2\mathcal{A}+ \oint C(t)\times\Delta'\boldsymbol{\tau}_i(t)+\Delta\boldsymbol{\tau}_i(t)\times C'(t)\\
&\quad+\Delta\boldsymbol{\tau}_i(t)\times\Delta'\boldsymbol{\tau}_i(t)dt.
\end{aligned}
$$

In order to keep the total area of $D(t)$ the same as $C(t)$, we require $\hat{\mathcal{A}}=\mathcal{A}$, or,

$$
\begin{aligned}
0 &= \oint C(t)\times\Delta'\boldsymbol{\tau}_i(t)+\Delta\boldsymbol{\tau}_i(t)\times C'(t)+\Delta\boldsymbol{\tau}_i(t)\times\Delta'\boldsymbol{\tau}_i(t)dt\\
&= \oint x(t)\delta_y'(t)-\delta_x'(t)y(t)+\delta_x(t)y'(t)-x'(t)\delta_y(t)\\
&\quad+\delta_x(t)\delta_y'(t)-\delta_x'(t)\delta_y(t)dt\\
&= \oint \delta_x(t)\overline{y'}(t)-\delta_x'(t)\overline{y}(t)+x(t)\delta_y'(t)-x'(t)\delta_y(t)dt,
\end{aligned}
\tag{17}
$$

where

$$
\overline{y}(t)=y(t)+\delta_y(t).
$$

Equation (17) can also be written as,

$$
\oint \delta_x(t)\overline{y'}(t)-\delta_x'(t)\overline{y}(t)dt=\oint x'(t)\delta_y(t)-x(t)\delta_y'(t)dt.
\tag{18}
$$

Assume $\delta_y(t)$ has already been derived. Then, the right hand side of Eq. (18) is completely known and so is $\overline{y}(t)$. In other words, Eq. (18) is linear in $\delta_x(t)$ that can, with the aid of Eq. (11), be rewritten as,

$$
[\delta_{x_0},\delta_{x_1},\cdots,\delta_{x_{n-1}}]\boldsymbol{\Phi}\begin{bmatrix} y_0+\delta_{y_0}\\ y_1+\delta_{y_1}\\ \vdots\\ y_{n-1}+\delta_{y_{n-1}}\end{bmatrix}
=[x_0,x_1,\cdots,x_{n-1}]\boldsymbol{\Phi}\begin{bmatrix}\delta_{y_0}\\ \delta_{y_1}\\ \vdots\\ \delta_{y_{n-1}}\end{bmatrix}.
\tag{19}
$$

Therefore, and during interactive control point or curve select-and-drag operations, the y axis of $\Delta\boldsymbol{\tau}_i(t)$ can be solved for all the δy_i coefficients satisfying all constraints, excluding and ignoring the area constraint. Then, the x axis of $\Delta\boldsymbol{\tau}_i(t)$ is solved for all the δx_i coefficients satisfying all constraints, including the linear area constraint of Eq. (19). The right hand side of Eq. (19) is fully known, having the δ_{y_i} coefficients, and hence Eq. (19) adds one more linear (area) constraint in δx_i to the existing set of linear constraints.

So far, the $\boldsymbol{\Phi}$ matrix has been derived for curves in the original space, Ψ_0. Nevertheless, and during the multiresolution editing process, the modification curve, $\Delta\boldsymbol{\tau}_i(t)$, might be in a different space, Ψ_i. Let $\mathbf{A}_{mn}$ be the refinement (alpha) matrix [12] of a curve in Ψ_i to a curve in Ψ_0. $\mathbf{A}_{mn}$ is a matrix of size $m\times n$, $n>m$, refining a B-spline curve in Ψ_i with m control points into an identical B-spline curve in Ψ_0 with n control points. In essence, $\mathbf{A}_{mn}$ inserts the knots of $\{\boldsymbol{\tau}_0\backslash\boldsymbol{\tau}_i\}$. Then, for $\Delta\boldsymbol{\tau}_i(t)=(\delta_x(t),\delta_y(t))\in\Psi_i$ and $C(t)=(x(t),y(t))\in\Psi_0$, Eq. (19) becomes,

$$
[\delta_{x_0},\delta_{x_1},\cdots,\delta_{x_{m-1}}]\mathbf{A}_{mn}\boldsymbol{\Phi}\left[\begin{bmatrix} y_0\\ y_1\\ \vdots\\ y_{n-1}\end{bmatrix}+\mathbf{A}_{mn}^T\begin{bmatrix}\delta_{y_0}\\ \delta_{y_1}\\ \vdots\\ \delta_{y_{m-1}}\end{bmatrix}\right]
=[x_0,x_1,\cdots,x_{n-1}]\boldsymbol{\Phi}\mathbf{A}_{mn}^T\begin{bmatrix}\delta_{y_0}\\ \delta_{y_1}\\ \vdots\\ \delta_{y_{m-1}}\end{bmatrix}.
\tag{20}
$$

Needless to say, the role of the x and y axes could and should be interchanged. In practice, and as the user is dragging the modified point on the curve, numerous location-event are generated by the input device. We exchange the axis that is employed toward the satisfaction of the area constraint in each such location event, resulting in a fair looking behavior that favors no special axis.

4 Constrained Multiresolution Freeform Curve Editing

The combination of multiresolution control and linear constraints could be accomplished once one realizes how to properly construct a modification function $\Delta\boldsymbol{\tau}_i(t)$ (See Eq. (2)) such that,

$$
\begin{aligned}
&\Delta(t_u)=1,\ t_u \text{ is a user specified modification location,} && \text{(see Eq. (2))},\\
&\Delta(t_p)=0,\ \text{at each positional constraint, } t_p, && \text{(see Eq. (3))},\\
&\Delta'(t_t)=0,\ \text{at each tangential constraint, } t_t, && \text{(see Eq. (4))},\\
&\Delta(t)\ \text{Is symmetric to following any symmetry constraint,} && \text{(see Eq. (6))},\\
&\Delta(t)\ \text{Has a zero area contribution to } C(t), && \text{(see Eq. 19))},
\end{aligned}
\tag{21}
$$

Much like the unconstrained multiresolution curve editing case, if one could construct such a $\Delta\boldsymbol{\tau}_i(t)$ function that satisfies all Constraints (21), then if $C(t)$ satisfies a positional constraint at t_p, so will $C(t)+\vec{\mathcal{V}}_m\Delta\boldsymbol{\tau}_i(t)$ because $C(t_p)+\vec{\mathcal{V}}_m\Delta\boldsymbol{\tau}_i(t_p)=C(t_p)$. Similarly, a tangential constraint will also be preserved under this the modification of $\Delta\boldsymbol{\tau}_i(t)$ due to the fact that $C'(t_t)+\vec{\mathcal{V}}_m\Delta'_{\boldsymbol{\tau}i}(t_t)=C'(t_t)$. Moreover, if $C(t)$ and $\vec{\mathcal{V}}_m\Delta\boldsymbol{\tau}_i(t)$ are both symmetric, so is their sum, and finally the area that is contributed by $\Delta\boldsymbol{\tau}_i(t)$ to $C(t)$ is zero.

Hence, the construction of such a $\Delta(t)$ function is the key question in the possible synergy of multiresolution editing control and linear constraints' control. Before we go ahead and attempt to satisfy all these constraints, we must realize that these sets of constraints might be under-determined, exactly determined, or over determined. Having m constraints can result in an over-determined system of equations in Ψ_i but an under-determined system of equations in Ψ_j, $j<i$.

Clearly, an over-determined set of constraints can only be approximated. However, having an under-determined system suggests an infinite family of solutions, much like in the unconstrained multiresolution case. In order to minimize the global affect on the whole curve, we once again employ a solution for the under-determined linear system of equations that minimizes the global change. Herein, the derived solution is the one that minimizes the change in the control points of the $C(t)$ curve, in L_2 sense. Two possible approaches that can be employed to achieve an L_2 minimizing solution are either the singular valued decomposition (SVD) or the QR factorization [16] of linear systems of equations. Interestingly enough, the QR factorization is employed by [5], an approach taken by this work as well, due to efficiency reasons.

The space of Ψ_i might be incapable of satisfying all the existing constraints. More degrees of freedom, in the form of more knots, must be employed. Toward this end, we can construct a hierarchy of subspaces, $\{\Psi_i\}$, such that $\Psi_{i+1} \subset \Psi_i$. Each knot sequence $\boldsymbol{\tau}_{i+1}$ can employ, for example, half of the interior knots of $\boldsymbol{\tau}_i$, skipping every second interior knot. Having this clear hierarchy, the user can interactively reduce the domain of influence, effectively going to more and more fine subspaces, until a subspace is found where a complete satisfaction of all the constraints can be achieved. Section 5 presents several examples of multiresolution editing and control with linear constraints that follow the proposed synergetic methodology.

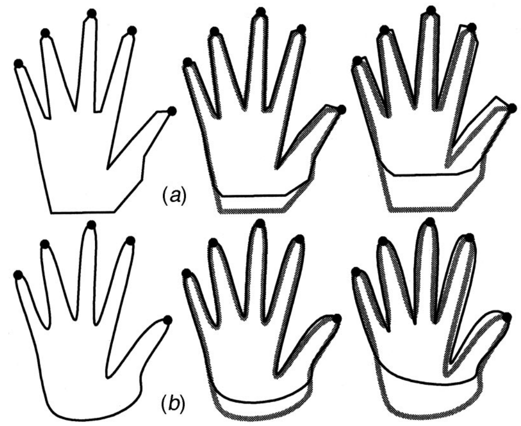

Fig. 4 An outline of a hand is manipulated from below while the finger-tips are anchored and the total area is preserved. As a result, the width of the finger is adapted to the changes from below while the fingertips are stationary. Both linear (*a*) and cubic (*b*) curves are shown. Starting with the original curve that is shown on the left, the modifications are shown in the middle and in the right side in black with the original curve shown in thick gray color.

5 Examples

The presented synergy between multiresolution control and linear constraints is equally applicable to *periodic* B-spline curves. The manipulated periodic curves necessitate a more careful budgeting of indices of the linear constraints as the system of constraints is now derived modulo the number of actual coefficients in the curve. Nonetheless, nothing is conceptually different when periodic curves are employed and in this section we mostly employ periodic curves.

All the examples presented in this section are the results of interactive sessions where the user attempted to directly manipulate the nonuniform B-spline curves, at various resolutions, and with a whole variety of linear constraints. The current resolution level and domain of influence is set by picking a point using the mouse at the center of the desired domain on the curve, and narrowing and/or expanding the domain of influence using the left and right keyboard keys.

The cross section in Fig. 3, possibly of a fuselage of a plane, is constrained to present a constant area, and hence eventually a fixed volume of the plane. Furthermore, and for obvious reasons, the curve is also constrained to be *Y*-symmetric. Shown in the figure are several cross sections that were all derived from the original curve, in gray, in few seconds, while preserving both the fixed area and the *Y*-symmetry constraints.

The shape of a hand in Fig. 4 is constrained to present a constant area, while the five fingertips are anchored via positional constraints. The hand's outline is manipulated from below affecting the width of the fingers in the attempt to preserve the total enclosed area. Both linear and cubic B-spline cases are considered and displayed.

The closed cross sections in Fig. 5 are also constrained to present a constant area. A single point is selected and dragged, resulting in the motion of the entire shape due to the preservation of this area constraint. Furthermore, other linear constraints could be simultaneously applied as is demonstrated in this Fig. 5.

A curve in the shape of a butterfly is directly manipulated in Fig. 6. This butterfly is a cubic B-spline curve with 49 control points. Shown in Fig. 6 are several samples of editing the curve at different resolutions, with and without constraints. A *Y*-symmetry constraint guarantees the general symmetry of the butterfly while a positional and a tangential constraint are placed on one wing (affecting both wings due to the symmetry constraint).

In Fig. 7, a curve is constrained to present a constant width by constraining the position as well as the tangent at the two extreme width locations. In addition, the curve is constrained to enclose a fixed area and be *Y*-symmetric throughout the editing process.

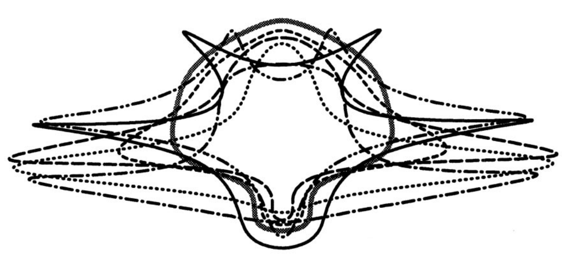

Fig. 3 In thick gray, a periodic planar B-spline cross section is shown that is constrained to be *Y*-symmetric as well as to present a fixed area. All the other cross sections were derived from it in few seconds via direct manipulation while the area as well as the *Y*-symmetry are preserved. The curve is a cubic periodic curve with twelve control points.

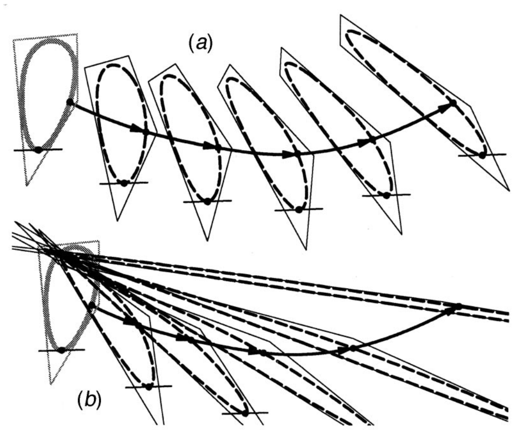

Fig. 5 A periodic planar cubic B-spline curve in gray is directly manipulated and dragged at the selected point to the right along the (solid curved) path while preserving the enclosed area. Several snapshots are shown. In (*a*), and in addition to the area constraint, a tangential constraint is preserved at the bottom of the shape. In (*b*), a third, additional, positional constraint is added to the top left side of the curve anchoring the shape to interpolate that location throughout this direct manipulation stage.

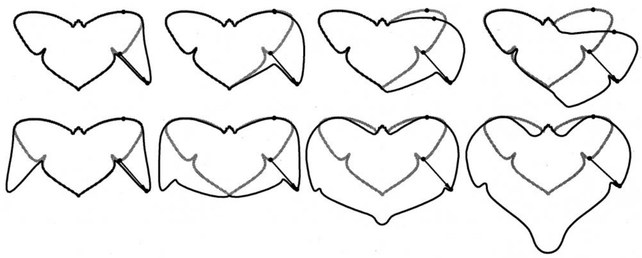

Fig. 6 A curve in the shape of a butterfly (in gray) is directly manipulated. A point on the bottom right side of the shape is moved in the bottom right direction, following the arrows. A positional as well as a tangential constraint are both placed at the top right of the butterfly, that is also constrained to be *Y*-symmetric. The top row shows the result of multiresolution editing in several resolutions without any constraints whereas the bottom row shows the same sequence of multiresolution operations with the constraints activated.

Fig. 7 A *Y*-symmetric curve with a constant area has a positional constraint on the left and a tangential constraint on the right. Several direct manipulation operations are performed while, effectively, these two constraints keep the curve at a *constant width*, throughout. The previous operation is shown in gray and the new one is shown in black, from left to right. These examples were created in a few seconds.

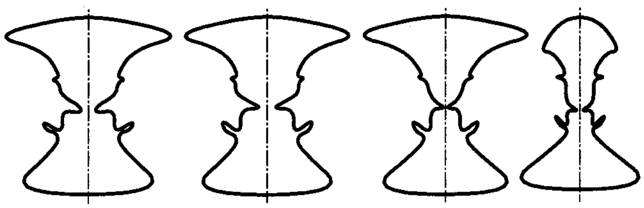

Fig. 8 Several examples of the illusion of two faces versus a vase. These examples were created in few minutes using a quadratic B-spline curve with 32 control points that is constrained to be *Y*-symmetric.

Shown in the figure are four steps of direct manipulations of the curve via a select-and-drag operation, all while the constraints are completely satisfied.

The infamous illusion of two faces versus a vase has several examples in Fig. 8. These examples were created using a quadratic periodic B-spline curve in few minutes, with the aid of a *Y*-symmetry constraint.

6 Extensions to 3D Curves on Surfaces

Curves need not be planar. In many cases, curves are constructed in three-dimensional space or are the result of operations between surfaces such as surface surface intersection (SSI). The work presented in the previous sections could be extended to curves on surfaces. A curve C could be defined on surface $S_1(u,v)$ as the intersection curve with another surface $S_2(r,t)$. Alternatively, C could be created via a *sketching operation directly on* S_1. The ability to hand-draw a curve is appealing in the preliminary stages of the design, when a rough yet fast construction of a prototype of the geometry is important. The curves could be sketched on the plane, but also, they could be sketched on an arbitrary surface. In the planar case, the mouse location is computed in the drawing plane. Similarly, and given a surface to sketch on, S, a ray from the mouse location is intersected against surface S, selecting the closest intersection point, if multiple intersection points are found.

Figure 9 shows a sketched curve in the shape of the letter 'S' on the body of the Utah teapot. The same curve could also be used to trim out the interior region of the 'S' shape from the surface of the body, by serving as a trimming curve of the surface.

While the mouse is dragged over the surface, sample points in the parametric domain of the curve are computed and stored. One could attempt multiresolution computation with constraints in Euclidean space. Nonetheless, such a process would be difficult to control due to the need to continuously keep the curve on the surface. In contrast, constraining the curve to lay exactly on the surface is cost free if the same computation is conducted in the parametric space of the surface. Further, clipping the curve to the surface domain is also simple as it amounts to clipping the parametric curve to the parametric domain of the surface.

Let $c(t)=(u(t),v(t))$ be the curve in the parametric domain of surface S that undergoes multiresolution editing. Its Euclidean equivalent is $C(t)=S(c(t))=S(u(t),v(t))$. $C(t)$ could be derived from $c(t)$ with the aid of composition computation over piecewise rational freeform surfaces [7,17]. While feasible, this process was found to be too slow for the interaction stages and hence $S(c(t))$ is evaluated at fixed samples to provide an approximated drawing at interaction rates. The exact composition computation could be employed, as required, once the interaction is completed. Figures 10 and 11 present the results of this exact composition computation.

In Fig. 10, multiresolution editing operations with no constraints are applied to the 'S' curve on the body of the Utah teapot. Three different levels of resolution are employed resulting in three

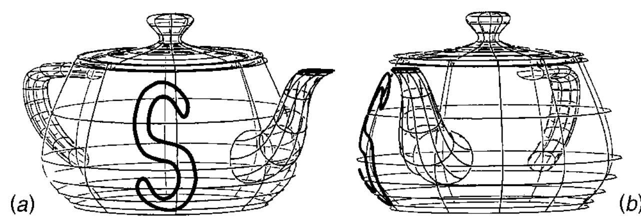

Fig. 9 Multiresolution with combination with constraints could also be applied to non planar curves on surfaces. Here, two views of a curve in the shape of the letter "S" are shown on the body of the Utah teapot, potentially serving as a trimming curve for the surface.

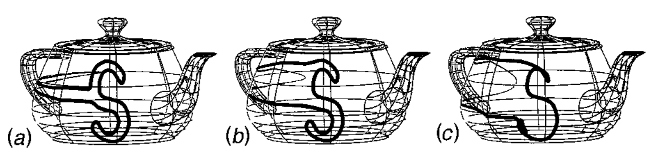

Fig. 10 Multiresolution editing without constraints is shown for the curve on the body of the Utah teapot in Fig. 9. Three resolution levels of a single select-and-drag operation and no constraints are shown.

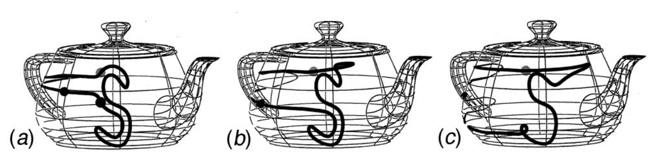

Fig. 11 Multiresolution editing with constraints is show for the curve on the body of the Utah teapot in Fig. 9. Three resolution levels of a single select-and-drag operation are shown along with two tangent constraint (black points) and one positional constraint (gray point).

different domains that are affected by a similar select-and-drag operation. Note the way the dragged curve is wrapped around the body.

In Fig. 11, multiresolution editing operations with constraint are applied to the "S" curve on the body of the Utah teapot. Two tangential constraints (black points) and one positional constraints (gray points) are placed. Again, three different levels of resolution are employed resulting in three different domains of the curve that are affected by a similar select-and-drag operation. Note the way the dragged curve is wrapped around the body and compare with Fig. 10. While a curve is dragged on a general surface, the tangent of the curve must be contained in the tangent plane of the surface at that location. A new surface position with new tangent plane might no longer contained some prescribed tangential constraint of the curve. One partial remedy could be the projection of the tangential constraint of the curve onto the tangent plane of the surface, seeking out the best possible approximation.

7 Extensions to Freeform Surfaces

The work presented here could also be extended to support the same synergetic view of multiresolution editing control and linear constraints for surfaces or even multivariate functions. Multiresolution control of surfaces has already been demonstrated [1–3] and the power of linear constraints in shape design has also been recognized [5]. The extension of the basic linear constraints from Section 3.1 to surfaces is simple. The symmetry constraint could also be considered in the context of surfaces along the same lines. Given surface $S(u,v)=\Sigma_i\Sigma_j P_{ij}B_{i,k_u}(u)B_{j,k_v}(v)$, where $P_{ij}=(x_{ij},y_{ij},z_{ij})$, one can extend the notion of symmetry and define $S(u,v)=(x(u,v),y(u,v),z(u,v))$ to be (Compare with Eq. (6)),

$$
\begin{aligned}
XY\text{-symmetric} \quad & \text{if} && x(u,v)=x(1-u,v) \\
& \text{and} && y(u,v)=y(1-u,v) \\
& \text{and} && z(u,v)=-z(1-u,v), \\
XZ\text{-symmetric} \quad & \text{if} && x(u,v)=x(1-u,v) \\
& \text{and} && z(u,v)=z(1-u,v) \\
& \text{and} && y(u,v)=-y(1-u,v), \\
YZ\text{-symmetric} \quad & \text{if} && y(u,v)=y(1-u,v) \\
& \text{and} && z(u,v)=z(1-u,v) \\
& \text{and} && x(u,v)=-x(1-u,v),
\end{aligned}
\tag{22}
$$

for all v and similarly for v for all u. This form of surface symmetry is clearly reducible to a set of linear constraints on the coefficients of the surface, extending the curves' symmetry case presented in Section 3.2.

The presented view of an enclosed area as a bilinear constraint could also be extended to handle the enclosed volume of freeform surfaces. The decomposition of the area constraint into a bilinear form is similarly extendible to the volume enclosed by a parametric B-spline surface. Following [14], the signed volume V, enclosed by parametric surface $S(u,v)$ equals,

$$
V=\int_U z\left(\frac{\partial x}{\partial u}\frac{\partial y}{\partial v}-\frac{\partial x}{\partial v}\frac{\partial y}{\partial u}\right)du\,dv, \tag{23}
$$

where U is the parametric domain of

$$
S(u,v)=\sum_i\sum_j P_{ij}B_{i,k_u}(u)B_{j,k_v}(v),
$$

and $P_{ij}=(x_{ij},y_{ij},z_{ij})$. Then,

$$
\begin{aligned}
V=&\int_U \sum_{l_u}\sum_{l_v} z_{l_u,l_v}B_{l_u,k_u}(u)B_{l_v,k_v}(v) \\
&\left(\sum_{i_u}\sum_{i_v} x_{i_u,i_v}B'_{i_u,k_u}(u)B_{i_v,k_v}(v)\right. \\
&\sum_{j_u}\sum_{j_v} y_{j_u,j_v}B_{j_u,k_u}(u)B'_{j_v,k_v}(v) \\
&-\sum_{i_u}\sum_{i_v} x_{i_u,i_v}B_{i_u,k_u}(u)B'_{i_v,k_v}(v) \\
&\left.\sum_{j_u}\sum_{j_v} y_{j_u,j_v}B'_{j_u,k_u}(u)B_{j_v,k_v}(v)\right)du\,dv \\
=&\sum_{i_u}\sum_{i_v} x_{i_u,i_v}\sum_{j_u}\sum_{j_v} y_{j_u,j_v}\sum_{l_u}\sum_{l_v} z_{l_u,l_v} \\
&\int_U B_{l_u,k_u}(u)B_{l_v,k_v}(v) \\
&(B'_{i_u,k_u}(u)B_{i_v,k_v}(v)B_{j_u,k_u}(u)B'_{j_v,k_v}(v) \\
&-B_{i_u,k_u}(u)B'_{i_v,k_v}(v)B'_{j_u,k_u}(u)B_{j_v,k_v}(v))du\,dv.
\end{aligned}
\tag{24}
$$

Hence, the volume enclosed by a parametric B-spline surface, in Eq. (23), is reducible to a tri-linear form in the x_{i_u,i_v}, y_{j_u,j_v} and z_{l_u,l_v} coefficients of the surface S. In a similar way to Eq. (11), one can a-priori compute the integral of the products of the basis functions in matrix $\mathbf{\Phi}$, with

$$
\begin{aligned}
&\phi_{i_u,i_v,j_u,j_v,l_u,l_v} \\
&\quad=\int_U B_{l_u,k_u}(u)B_{l_v,k_v}(v) \\
&\qquad(B'_{i_u,k_u}(u)B_{i_v,k_v}(v)B_{j_u,k_u}(u)B'_{j_v,k_v}(v) \\
&\qquad-B_{i_u,k_u}(u)B'_{i_v,k_v}(v)B'_{j_u,k_u}(u)B_{j_v,k_v}(v))du\,dv.
\end{aligned}
$$

With the aid of this tri-linear form in the coefficient of the surface, during the surface interactive manipulation, one is required to solve for the linear constraint of the volume in either the x, the y, or the z coefficients of the surface, in alternating order.

In [5], transfinite constraints are defined as integral constraints, and the coercion of a curve to be contained in a surface is presented as one example. These constraints are not always of infinite dimension and, for example, this coercion of a curve on a surface could be posed as the composition [7,17] constraint of a curve of degree n in a surface of degrees $m\times m$, resulting in an interpolation problem of a polynomial function with degree $2mn$. Hence, such a constraint could be embedded into the paradigm presented herein, while enforcing $2mn+1$ linear constraints, much like the set of linear constraints in the case of imposing a symmetry constraint, that was presented in this work.

8 Extensions to Higher Order Moments

The tri-linear volume computation in Section 7 is a special case of an evaluation of a zero order moment. One can extend this scheme to a multi-linear evaluation of higher order moments. Here, we briefly present how first and second order moments of objects bound by freeform parametric B-spline surface could be efficiently evaluated.

Reconsider Eq. (23). The moment, $\mathcal{M}$, of order $i+j+k$ equals (See also [13]),

$$
\mathcal{M}=\int_U \kappa x^i y^j z^k z\left(\frac{\partial x}{\partial u}\frac{\partial y}{\partial v}-\frac{\partial x}{\partial v}\frac{\partial y}{\partial u}\right)du\,dv, \tag{25}
$$

where $\kappa \in \mathbb{R}$ depends on k and for example for $k=0$, $\kappa=1$.

In a similar way to (24), one can expand Eq. (25) for first order moments. Consider the first order x moment, i.e. $i=1, j=k=0$. Then (y and z are similarly computed),

$$\mathcal{M}_x = \int_U xz\left(\frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u}\right)du\,dv$$
$$= \int_U \sum_{m_u}\sum_{m_v} x_{m_u,m_v} B_{m_u,k_u}(u) B_{m_v,k_v}(v)$$
$$\sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v} \phi_{i_u,i_v,j_u,j_v,l_u,l_v}\,du\,dv$$
$$= \sum_{m_u}\sum_{m_v} x_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v}$$
$$\int_U \phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v}\,du\,dv \tag{26}$$

where

$$\phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v}(u,v) = \phi_{i_u,i_v,j_u,j_v,l_u,l_v}(u,v) B_{m_u,k_u}(u) B_{m_v,k_v}(v).$$

The second order moment assumes the following multi-linear multiplicative form (again for x only, $i=2, j=k=0$):

$$\mathcal{M}_{xx} = \int_U \sum_{n_u}\sum_{n_v} x_{n_u,n_v} B_{n_u,k_u}(u) B_{n_v,k_v}(v)$$
$$\sum_{m_u}\sum_{m_v} x_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v} \sum_{i_u}\sum_{i_v} x_{i_u,i_v}$$
$$\sum_{j_u}\sum_{j_v} y_{j_u,j_v} \phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v}\,du\,dv$$
$$= \sum_{n_u}\sum_{n_v} x_{n_u,n_v} \sum_{m_u}\sum_{m_v} x_{m_u,m_v} \sum_{l_u}\sum_{l_v} z_{l_u,l_v}$$
$$\sum_{i_u}\sum_{i_v} x_{i_u,i_v} \sum_{j_u}\sum_{j_v} y_{j_u,j_v}$$
$$\int_U \phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v}\,du\,dv \tag{27}$$

where

$$\phi^2_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v,n_u,n_v}(u,v) = \phi^1_{i_u,i_v,j_u,j_v,l_u,l_v,m_u,m_v}(u,v) B_{n_u,k_u}(u) B_{n_v,k_v}(v).$$

These high order moments are no longer linear in all degrees of freedoms as in the volume or zero order moment case. Yet, the multiplicative nature of the final expressions of (26) and (27) and the fact that some degrees of freedom remain linear, suggest that one might be able to benefit from this representation and embed them as constraints, much like the volume computation presented in Section 7. This question is open for future research. For further details on this efficient moment evaluation scheme, see [18].

9 Conclusions

A possible synergy between two important freeform curve editing paradigms has been demonstrated. Multiresolution editing control could be smoothly integrated with a large variety of linear constraints. Furthermore, we have introduced two additional linear constraints, the symmetry constraint that is linear but introduces a set of $n/2$ linear constraints, and the area constraint that as a bilinear form could be employed as an interchangeable linear constraint as well.

Herein, we have presented the ability to apply one symmetry constraints only. It might be desirable to apply several symmetry constraints, simultaneously. Then, each additional symmetry constraint reduces by half the number of degrees of freedom of the shape. For example, a curve with an X- as well as a Y-symmetry constraint would end up with $n/4$ degrees of freedom left.

In Section 3.1, we have presented a relaxed set of simple linear constraints that impose G^1 continuity instead of C^1 continuity. The fact that these G^1 style constraints necessitate a simultaneous solution of the X and Y axes, render them impossible to use in juxtaposition with the area (for curves) or volume (for surfaces) constraints. This deficiency deserves some more considerations and hopefully could be resolved in a form of a two stage simultaneous solution, relaxing one axes after the other.

Rational curves are typically supported by neither multiresolution editing and control nor by linear constraints. While it is feasible for both paradigms to support the rational representation, the computational overheads are significant. The expected benefits of using rational forms instead of polynomial forms in free style shaping and modeling should be weighed against these anticipated overheads.

We are hopeful that the synergy presented in this work will further alleviate the difficulties that the geometric design community is facing, in attempting to provide interactive as well as intuitive tools to manipulate freeform curves and surfaces.

10 Acknowledgment

I would like to thank Octavian Soldea for his aid and work on the problem of efficient evaluation of higher order moment.

References

[1] Finkelstein, A., and Salesin, D. H., 1994, ''Multiresolution Curves,'' Siggraph-Comput. Graph., pp. 261–68.
[2] Gortler, S. J., 1995, ''Wavelet Methods for Computer Graphics,'' Ph.D. dissertation, Princeton University.
[3] Kazinnik, R., and Elber, G., 1997, ''Orthogonal Decomposition of Nonuniform B-spline Spaces Using Wavelets,'' Computer Graphics *forum*, **16**, No. 3, pp. 27–38.
[4] Gleicher, M., 1992, ''Integrating Constraints and Direct Manipulation. Computer Graphics,'' **25**, No. 2, pp. 171–174, Symposium on Interactive 3D graphics.
[5] Welch, W., and Witkin, A. 1992, ''Variational Surface Modeling'' Computer Graphics, Siggraph, **26**, pp. 157–166.
[6] Gortler, S. private communications.
[7] Elber, G., 1995, ''Symbolic and Numeric Computation in Curve Interrogation,'' Computer Graphics *forum*, **14**, No. 1, pp. 25–34.
[8] Kaklis, P. D., and Sapidis, N. S., 1995, ''Convexity-Preserving Interpolatory Parametric Splines of Nonuniform Polynomial Degree, Computer Aided Geometric Design, **12**, No. 1, pp. 1–26.
[9] Rappaport, A., Sheffer, A., and Bercovier, M., 1996, ''Volume-Preserving Free-Form Solids, IEEE Transactions on Visualization and Computer Graphics, **2**, No. 1, pp. 19–27.
[10] Plavnik, M., and Elber, G., 1998, ''Surface Design Using Global Constraints on Total Curvature,'' The VIII IMA Conference on Mathematics of Surfaces, Birmingham.
[11] IRIT 8.0 User's Manual, October 2000, Technion. http://www.cs.technion.ac.il/~irit.
[12] Cohen, E., Lyche, T., and Riesenfeld, R. F., 1980, ''Discrete B-splines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics,'' CGIP, **14**, No. 2, pp. 87–111.
[13] Eberly, D., and Lancaster, J., 1991, ''On Gray Scale Image Measurements: I. Arc Length and Area,'' CVGIP: Graphical Models and Image Processing, **53**, No. 6, pp. 538–549.
[14] Gonzalez-Ochoa, C., Mccammon, S., and Peters, J., 1998, ''Computing Moments of Objects Enclosed by Piecewise Polynomial Surfaces, ACM Trans. Graphics, **17**, No. 3, pp. 143–157.
[15] Pearson, C. E., 1990, Handbook of Applied Mathematics, Selected Results and Methods, Van Nostrand Reinhold publishing, Second Edition.
[16] Golub, G. H., and Van Loan, C. F., 1996, Matrix Computation, The John Hopkins University Press, Baltimore and London, Third Edition.
[17] Liu, W., and Mann, S., 1997, ''An Optimal Algorithm for Expanding the Composition of Polynomials,'' ACM Trans. Graphics, **16**, No. 2, pp. 155–178.
[18] Soldea, P., Elber, G., and Rivlin, E., Exact and Efficient Computation of Moments of Free-Form Surface and Trivariate Based Geometry. Accepted for publication in Computer Aided Design.