Conceptual Free-Form Styling on the Responsive Workbench

Gerold Wesche Marc Droske*
GMD – German National Research Center for Information Technology
Institute for Media Communication
Sankt Augustin, Germany
gerold.wesche@gmd.de

ABSTRACT

A two-handed 3D styling system for free-form surfaces in a table-like Virtual Environment, the Responsive Workbench (RWB)™, is described. Intuitive curve and surface deformation tools based on variational modeling and interaction techniques adapted to 3D VR modeling applications are proposed. The user draws curves (cubic B-splines) directly in the Virtual Environment using a stylus as an input device. The curves are connected automatically, such that a curve network develops. A combination of automatic and user-controlled topology extraction modules creates the connectivity information. The underlying surface model is based on B-spline surfaces, or, alternatively, uses multisided patches [20] bounded by closed loops of curve pieces.

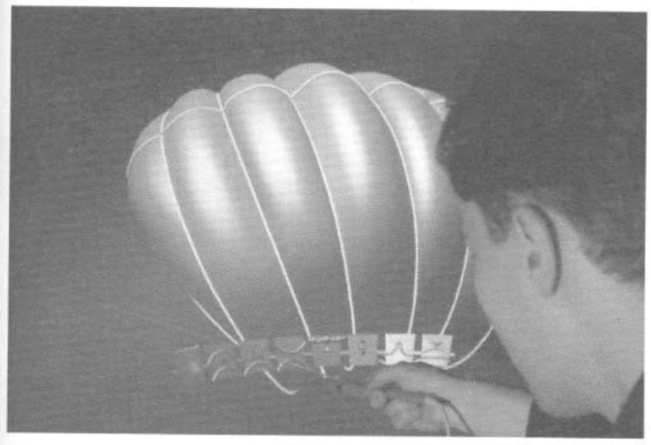

Figure 1: 3D styling of surfaces on the Responsive Workbench (RWB)™

1. STYLING IN VIRTUAL ENVIRONMENTS

Many designers would like sketching free-form shapes quickly in the conceptual design phase without using a complex CAD system [6]. Projection-based Virtual Environments like the Responsive Workbench (RWB)™ [1] [19] offer the following advantages for sketching applications over traditional desktop interfaces:

1. The three-dimensionality of perception in combination with 3D interaction gives an immediate understanding of the shape.
2. Regions can be identified, localized, and selected directly in space.
3. The large, high-resolution projection plane enables the representation of objects on a scale which corresponds to the working region of both hands.

Our ambition is to support free-form styling capabilities in a sketching system on the Responsive Workbench. We try to overcome the involved complexity regarding the user interface by exploiting three-dimensionality for the development of an alternative tool selection approach. Moreover, we hide the mathematical complexity from the designer by developing intuitive deformation techniques for curves and surfaces relying on familiar metaphors. Our basic approach is constructing curve nets like in [29] by drawing cubic B-spline curves directly in space and creating surfaces inside the loops of the curve net, see Figure 1.

The paper is organized as follows: After summarizing previous and related work in 1.1, we discuss two alternative, but related surface modeling approaches in 1.2. Section 2 summarizes the principles and previous work for variational modeling. In sections 3 and 4, we develop our tools based on this technique for curves and surfaces, respectively. We integrated the tools into our modeler described in section 5. In addition, we propose two interaction techniques for modeling applications on the workbench. We give ideas for future work in 6 and draw conclusions in 7.

---

*now working at University of Bonn, Germany

[1] Responsive Workbench (RWB) is a trademark of GMD, Germany, 1996.

VRST 2000 Seoul, Korea

1.1 Related work

In recent years, research work on modeling in Virtual Environments has gained much interest. Most contributions which provide free-form sketching capabilities follow a direct approach. They develop methods how users can quickly create and deform surfaces as presented in [23, 30, 34]. They have only a few drawing restrictions which usually involves that it can be very difficult to derive closed form representations from the user input. In our approach, surfaces are created indirectly by drawing the primal curves of objects like in [29]. An advantage of this is that we can create surfaces of closed form representation interactively, which is an important aspect in a Virtual Environment. On the other hand, users have to accept some topological restrictions during the shape design.

Desktop and Head Mounted Display Systems

In 1976, Clark [3] build a system which used a HMD and some buttons to design bicubic patches rendered as line-drawings. To our knowledge, the first work proposing the idea of two-handed interaction and editing curve nets is the "3-Draw"-system from Sachs et al. [29], which is a desktop oriented system. One hand controls a tablet whereas the other hand draws curves on the tablet. Both hands are electromagnetically tracked. The system lacks a surface model, so that only the curve net is shown. Butterworth et al. [1] developed a modeling application designed for a Head Mounted Display (HMD) environment. The user assembles objects out of solid primitives like spheres or cylinders. With the "Fast Shape Designer", van Dijk [35] developed a similar system compared to that of Sachs et al. [29]. They use an irregular network of NURBS curves. The surfaces supported are 3 sided or 4 sided tensor product surfaces. To extract the topology for such kind of applications, Elsas et al. [36] implement a search method to generate a set of patches out of a sketched curve network. Usoh et al. [34] describe an idea for deforming surfaces in a HMD environment. They define spline-based surfaces by sweeping them out with the hand and manipulate them. They are able to deform them by simulating deformations. The "THRED" system by Shaw, Green et al. [7, 31] describe a simple two-handed desktop free-form editor. The user interacts with the model with both hands, which are tracked. "JDCAD" by Liang and Green [22] is a 3D solid modeling and animation system. Zeleznik et al. [41] describe a gesture based design system for 3D objects. It does not yet support free-form curves or surfaces. Matsuda et al. [25] follow a similar approach. They try to extract object descriptions out of ambiguous line segments typically used in paper drawings when an object is roughly sketched. The object descriptions are based on polyhedrons. The "Teddy" sketching interface [15] allows designing free-form models quickly and easily. From several 2D strokes, plausible polygonal surfaces are constructed. The "Skin" approach [23] proposes a new particle-based surface representation for sculpting free-form surfaces. Users interactively guide the particles to form triangulations with properties that make them suitable for subdivision. Several contributions deal with virtual assembly like [12, 17]. Some systems even support force feedback like the commercially available FreeForm™[2] modeling package [16], which is coupled to a PHANToM™[3] haptic device. This system uses an internal volumetric representation which requires a conversion if smooth surface representations are preferred.

Projection-based systems

Dani and Gadh [5] present a desktop VR system for design, called "COVIRDS", by using a combination of hand gestures, voice input, and keyboard input to create and manipulate a 3D artifact. They have extended their work for projection-based VR systems. An approach which relies on pictographic gestures describing superquadrics has been realized by Nishino et al. [26] in a 3D modeling system running in a projection wall environment. It has even been extended to support collaborative 3D modeling in a distributed Virtual Environment [27]. Cutler et al. [4] realized two-handed interaction for the Responsive Workbench (RWB)™ based on observations of how the two hands work together in daily life situations [11]. The approach of Krause and Lüddemann [18] about virtual clay modeling supports spline based virtual tools which are used to take away virtual material from a voxel volume. Their system is implemented on the workbench. W. Buxton et al. [2] contribute a styling approach motivated by the automotive design process. They explore the traditional and current uses of large displays in this field and present new applications, including digital tape drawing, that make innovative use of large displays. Gregory et al. [8] couple a small PHANToM™ force feedback device with a multiresolution mesh editor which allows users to naturally create complex forms. The appearance can be further enhanced by directly painting onto the surface. In contrast to [16], their work is based on a surface representation. The calculation of the surface deformation relies on a simple heuristic. A direct approach for surface sketching is "Surface Drawing" by Schkolne and Schröder [30]. Polygonal surfaces are created by moving a hand, instrumented with a glove, through space on the workbench. This modeler seems to fulfill all the requirements identified in a workshop analysing the needs of designers in Virtual Environments [6], like intuitive interaction and direct transfer of the design intent, although it was not part of the user tests. However, they do not support a closed form representation of smooth surfaces. This would involve topological restrictions or special treatment at the boundaries to fulfil continuity requirements, which may not be desirable in such a direct approach.

Other related work

Hummels et al. [13, 14] describe possible scenarios and gesture based 3D modeling concepts for a car styling environment. The interplay between both hands is used trying to extract the description of shapes.

1.2 Modeling free-form curves and surfaces

Two related approaches exist that can be used as a basis to develop intuitive modeling tools on top of them.

Variational methods try to create fair shapes by minimizing the energy of a curve or surface under given constraints. They have been introduced for surfaces by Welch and Witkin [38]. Wesselink and Veltcamp [37, 39, 40] present variational modeling techniques and tools for curves and surfaces.

---

[2] FreeForm is a Trademark of SensAble Technologies.

[3] PHANToM is a Trademark of SensAble Technologies.

Greiner and Seidel [9, 10] discuss approaches for energy functionals used in minimization methods.

The usual way these methods work is that a designer first has to define positional or directional constraints of an object and after that has to activate an energy minimization step to produce a shape. The process is then repeated until a desired shape has been found. Althoug this approach produces fair, pleasant shapes, is has one drawback. The minimization step applied in a usual way is a global smoothing process which wipes out details that may be desired features.

The second approach relies on the simulation of elastic deformations when a force is operating on a curve or a surface. If variational modeling is already available, linear elastic deformations for small forces can be implemented in a relatively easy way. In this case, the Hookean law holds and the energy matrix can be used as the stiffness matrix of the system. The full dynamic case is more complicated (and involves more computation time). Work in this field is presented by Qin et al. [28] for Catmull-Clark surfaces and by Terzopoulos [32] for NURBS curves and surfaces.

This paper presents intuitive curve and surface deformation tools based on variational modeling techniques by combining energy terms multiplied with weight functions (implemented for curves). Our main contribution in this field is to present a method how energy terms introduced by [37, 39, 40] could be combined to define intuitive, interactive tools which can be applied locally or globally. Such tools demonstrate their usefulness especially in a Virtual Environment. Therefore, we present a sketching system for simple objects constructed from curve networks and free-form surfaces. The effects of the tools presented in this paper are similar to the already established elastic deformations and therefore could be seen as an alternative approach. However, the way they are applied is different.

2. BASICS OF VARIATIONAL MODELING FOR CURVES AND SURFACES

Variational modeling is a modeling approach based on minimizing a "fairness" functional defined over a curve or surface. The goal is to achieve a fair, pleasant shape. Although there is not an exact definition of what is meant by "fair", the following criterions can be found in the literature: Inflection points, flat spots, and buckles or bumps are not desirable. The shape should be somehow visually pleasing. As a functional which measures the fairness, a combination of bend and stretch energy approximations can be used.

2.1 Notation

Throughout the paper, we use the following notation. Vectors are assumed to be columns and are denoted by boldface lower case characters, like $\mathbf{v}$, whereas matrices are denoted as uppercase boldface characters, e.g., $\mathbf{M}$. Scalars are denoted by italicized characters, e.g., w or E. The dot product between two vectos $\mathbf{x}$ and $\mathbf{y}$ is written $\mathbf{x}^T\mathbf{y}$. Derivatives are denoted by points or by subscripts, e.g., $\dot{\mathbf{x}}$ or $\mathbf{s}_u$.

2.2 Curves

For the following, we assume that our curves or their segments can be described in the following representation:

$$\mathbf{c}(t) = \sum_{i=1}^{n} \mathbf{p}_i N_i(t) \tag{1}$$

is a curve with n control points $\mathbf{p}_i$ and basis functions $N_i(t)$ defined over a parameter t. In this paper, we assume this representation when we use the term "curve". We implemented our tools for non-uniform cubic B-spline curves.

The energy functional approximations for the bend energy

$$E_b = \int_{t_0}^{t_1} \|\ddot{\mathbf{x}}_{min}(t)\|^2 dt \tag{2}$$

and the stretch energy

$$E_s = \int_{t_0}^{t_1} \|\dot{\mathbf{x}}_{min}(t)\|^2 dt \tag{3}$$

for curves in their simple forms as given in [40] are then quadratic in the control points, so that (6) can be solved efficiently. t_0 and t_1 denote the curve parameter interval where the curve $\mathbf{x}_{min}$ is supposed to be minimized. E_b and E_s are parts of a weighted sum, that describes the *internal* energy of a curve,

$$E_i = w_b E_b + w_s E_s. \tag{4}$$

To allow more control over the shape of a curve, attractors have been introduced by [39, 40]. Attractors are weigthted energy terms which contribute to the *external* energy E_e of a curve. As their name indicates, these energy terms can attract a curve towards a given point, line, plane or towards another curve.

$$E = w_i E_i + w_e E_e \tag{5}$$

depicts the total energy of the curve. The quadratic parts of E can be assembled into an energy matrix $\mathbf{A} \in \mathbb{R}^{3n\times 3n}$ for the curve. $\mathbf{A} = \mathbf{A}_i + \mathbf{A}_e, \mathbf{A}_i, \mathbf{A}_e \in \mathbb{R}^{3n\times 3n}$, where $\mathbf{A}_i$ contains the internal energy and $\mathbf{A}_e$ is the external energy matrix. If all the control points $\mathbf{p}_i$ are collected in the concatenation vector $\mathbf{c} \in \mathbb{R}^{3n}$, a quadratic programming problem with linear constraints

$$\text{minimize} \quad \mathbf{c}^T\mathbf{A}\mathbf{c} + \mathbf{b}^T\mathbf{c} + k \tag{6}$$

$$\text{such that} \quad \mathbf{D}\mathbf{c} = \mathbf{e}, \tag{7}$$

is obtained where $\mathbf{b}$ contains the linear contributions of the energy terms in E and k contains the constant parts. (7)

with $\mathbf{D} \in \mathbb{R}^{m \times 3n}, \mathbf{e} \in \mathbb{R}^m$ comprises the m linear constraints, which can be used to fix the first and last control points influencing the curve segment subject to minimization. This problem can be solved efficiently by various means, see [39].

2.3 Surfaces

In this paper, we assume the following representation when we use the term "surface". We implemented the tools for uniform bicubic B-spline surfaces. Let

$$\mathbf{s}(u,v) = \sum_{i=1}^{n} \mathbf{p}_i N_i(u,v) \tag{8}$$

describe a surface $\mathbf{s} : \Omega \to \mathbb{R}^3$ with with n control points $\mathbf{p}_i$ and bivariate basis functions $N_i(u,v)$ defined over the rectangular domain $\Omega \subset \mathbb{R}^2$ and $(u,v) \in \Omega$. As in the curve case, a weighted sum E_i of simple approximations of the *thin plate* or bend energy E_b and of the stretch energy E_s is commonly used as the internal energy contribution to the fairness functional:

$$E_b = \int_\Gamma (\mathbf{s}_{uu}^T \mathbf{s}_{uu} + 2\mathbf{s}_{uv}^T \mathbf{s}_{uv} + \mathbf{s}_{vv}^T \mathbf{s}_{vv})\, du dv, \tag{9}$$

$$E_s = \int_\Gamma (\mathbf{s}_u^T \mathbf{s}_u + \mathbf{s}_v^T \mathbf{s}_v)\, du dv, \tag{10}$$

$$E_i = w_b E_b + w_s E_s. \tag{11}$$

Here, $\Gamma \subseteq \Omega$ denotes the region for the optimization (6). Similar to (5), attractors contribute to an external energy E_e and can be used to control the shape resulting from the minimization process. The total energy of the surface becomes, like in the curve case

$$E = w_i E_i + w_e E_e. \tag{12}$$

The control points $\mathbf{p}_i$ can be concatenated into a vector $\mathbf{c}$, analogous to the curve case, and a minimization problem like (6) under linear constraints like (7) has to be solved. The constraints can be used to fix the control points of the boundary.

3. INTUITIVE CURVE STYLING TOOLS

With such tools at hand, designers are able to influence the minimization process in a way that the object adopts a desired shape. However, the possibilities to deform a curve or surface in this way are restricted. Within the region in which the minimization occurs, details tend to decrease and previously defined features are being ignored. Therefore it is necessary to add energy terms which can *preserve* the shape of an object. The following energy terms could be adapted for this task:

$$E_{p0} = \int_{t_0}^{t_1} \|\dot{\mathbf{x}}_{min}(t) - \dot{\mathbf{x}}_{ref}(t)\|^2 dt \tag{13}$$

$$E_{p1} = \int_{t_0}^{t_1} \|\ddot{\mathbf{x}}_{min}(t) - \ddot{\mathbf{x}}_{ref}(t)\|^2 dt \tag{14}$$

In our experiments, a combination

$$E_p = w_{p0} E_{p0} + w_{p1} E_{p1} \tag{15}$$

produces the best results. These energy terms tend to minimize the variations between two curves. If $\mathbf{x}_{ref}(t)$ is initialized with the original curve and $\mathbf{x}_{min}(t)$ is set to the results of the minimization, features of the original curve can be preserved, which can be controlled by using corresponding weights.

We introduce *locality* of the energy minimization by weight functions defined over the same parametric domain as the curve. We define "local" bend and stretch energy terms E_{bl} and E_{sl} as follows:

$$E_{bl} = \int_{t_0}^{t_1} w_{bl}(t) \|\ddot{\mathbf{x}}_{min}(t)\|^2 dt, \tag{16}$$

$$E_{sl} = \int_{t_0}^{t_1} w_{sl}(t) \|\dot{\mathbf{x}}_{min}(t)\|^2 dt, \tag{17}$$

where $w_{bl}(t)$ and $w_{sl}(t)$ are user defined weight functions to control the amount of smoothing along the curve.

With these features, it is easy to combine energy terms to develop new interactive, intuitive curve deformation tools. In the following paragraphs, we give some examples how this can be achieved.

3.1 A curve smoother

An interactive curve smoother can be assembled from the following energy terms.

$$E = E_{smooth} = w_b E_{bl} + w_s E_{sl} + w_p E_p, \tag{18}$$

where $w_{bl}(t)$ and $w_{sl}(t)$ are chosen to be an exponential weight function of gaussian shape with it's maximum at the parametric location t_0 where the designer points at the curve:

$$w_{bl}(t) = w_{sl}(t) = e^{-a(t-t_0)^2} \tag{19}$$

$a > 0$, w_b, and w_s have to be chosen by the designer. This interactive smoother works in the following way. First, a curve $\mathbf{x}_{min}$ is selected that is supposed to be smoothed. In each frame of the application process, the curve parameter t_0 is determined from the location of the input device which should be reasonable close to the curve. The current state of $\mathbf{x}_{min}(t)$ is copied into $\mathbf{x}_{ref}(t)$ to calculate a new E_{smooth} which defines the matrix $\mathbf{A}$ for the minimization step. The solution produces a curve which is slightly smoother at the location of the pointer, but still retains it's details beside the maximum of the gaussian. This curve is taken as $\mathbf{x}_{ref}(t)$ for the next application step. The curve is gradually smoothed out depending on the weights used. In fact, the smoother is designed to be applied in a similar way as a flat iron would be. The pointer is moved along the curve until the desired shape has been achieved. The smoothing effect decreases over time as long as the curve is active. Figure 2 shows an example.

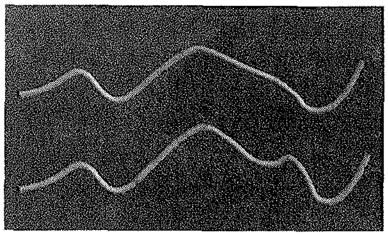

Figure 2: Smoothing a curve locally

3.2 A curve sharpener

The curve sharpener can be interpreted as the inverse operation to the curve smoother. It seems to be peculiar to create a sharpening operation with variational methods. However, it is easy to implement the desired effect by simply choosing negative weights w_b and w_s in equation (18) for the weight function of the internal energy. The curve sharpener is applied in the same way as the curve smoother such that initial details on the curve can be elaborated. As opposed to the smoother, the sharpening effect increases over time. This means that the whole process does not converge. Usually the sharpening tool is deactivated before this critical situation occurs. Figure 3 shows a curve which has been locally sharpened.

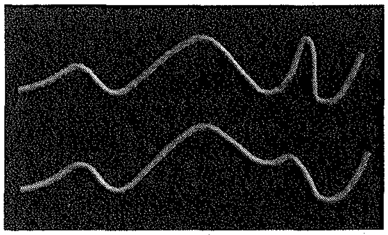

Figure 3: Sharpening a curve locally

3.3 A curve puller

Several attractors have already been proposed by [40] to pull a curve segment to a given point $\mathbf{p}$, line or plane or pull a point at the location t_0 on the curve towards another given point $\mathbf{p}$. As an example, we list the "point to point attractor":

$$E_{pp} = ||\mathbf{x}_{min}(t_0) - \mathbf{p}||^2. \tag{20}$$

To create a curve pulling tool which preserves the details and features of the selected segment (Figure 4), we combine E_{pp} with a shape preserving term E_p selected from equations (13), (14), or (15):

$$E = E_{pull} = w_{pp}E_{pp} + w_pE_p. \tag{21}$$

The way the curve puller works is a little bit different from the smoothing and sharpening procedures. When the puller is activated with a curve $\mathbf{x}_{min}(t)$, $\mathbf{x}_{ref}(t)$ in E_p is initialized with this curve to preserve it's details. In the following frames of the application process, only $\mathbf{p}$ in (20) is updated with the current pointer position.

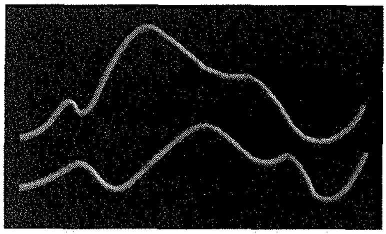

Figure 4: Pulling a curve segment (to the upper left)

3.4 Advantages

We note that these deformation tools have similar effects than the simulation of a force application onto a curve. However, the way of controlling smoothing and sharpening is rather indirect compared to the direct pulling or pushing process. Especially for larger deformations the amount of arm movement can be restricted to pointing to the location where the effect is intended to increase. Moreover, it seems to be easier to control the exactness of the smooting process of a curve in this way rather than trying this with direct deformations. By applying the intuitive "flat-iron-metaphor", the location of smoothing out the curve can be distributed in a very elegant way.

4. INTUITIVE SURFACE STYLING TOOLS

Similar styling tools can be developed for surfaces, since they can be represented in an analogous canonical representation (8). For surfaces, we use a different approach to implement shape-preserving behaviour. From the internal energy and the control points we compute the equilibrium $\mathbf{b}_q = \mathbf{A}_i\mathbf{c}$ and add it to the right hand side of the linear system used to solve the minimization problem (6). In (6), we set

$$\mathbf{b} = \mathbf{b}_e - w_q\mathbf{b}_q, \tag{22}$$

where $\mathbf{b}_e$ contains the linear contributions of all attractors in the system. The weight w_q controls how much the result approaches the minimal energy surface. $w_q = 1$ results in the original surface, whereas $w_q = 0$ causes the minimal surface (if we assume $\mathbf{b}_e = \mathbf{0}$).

4.1 A surface smoother

The surface smoother is designed in a way that the grade of the smoothness can be controlled by stopping the energy minimization when the desired shape is achieved. In contrast to the curve case, weighted energy functions are not implemented. The energy term is simply the combination of the bend and stretch engergies:

$$E = E_{smooth} = w_i E_i \tag{23}$$

In addition, an equilibrium is calculated to delay the smoothing process over time. In each frame of the application process, w_q in (22), with an initial value of 1, is multiplied by a factor $f < 1$. The result becomes the new w_q, such that the influence of $\mathbf{b}_q$ is decreasing. Note that $\mathbf{b_e} = \mathbf{0}$ in this case. Because the energy matrix $\mathbf{A}$ needs to be initialized only once, the smoother can be used interactively. In each frame, after calculating $\mathbf{b}_q$ and $\mathbf{b}$, only the solving step needs to be performed.

4.2 A surface sharpener

A sharpening tool can be derived from the smoother by simply using an $f > 0$. This causes the whole surface to get "inflated", until stopped by the user in order to avoid that the process diverges.

4.3 A surface puller

The surface puller can be used to deform a surface region similar to the curve case. We need a shape preserving term, $\mathbf{b}_q$ from (22), an internal energy term, and a "point to point attractor" for surfaces, as defined in [39],

$$E_{pp} = \|\mathbf{s}(u_0, v_0) - \mathbf{p}\|^2, \tag{24}$$

where (u_0, v_0) is the parametric location of the surface point which is supposed to be attracted towards $\mathbf{p}$. The total energy in case of surface pulling becomes

$$E = E_{pull} = w_i E_i + w_{pp} E_{pp}. \tag{25}$$

Recall that $\mathbf{b}_q$ is computed only from E_i, the quadratic contribution from E_{pp} does not influence the equilibrium $\mathbf{b}_q$. It has to be initialized when the puller starts working and retains it's value in the following frames. Only E_{pp} and $\mathbf{b}$ have to be updated before the solution can be calculated.

5. MODELING ON THE RESPONSIVE WORKBENCH

5.1 Modeling curve networks

In our system, cubic B-spline curves can be edited freely in space, but new curves have to be woven into the existing curve network. The curve is first drawn directly in space without any constraints. This is not as difficult as it might appear. As opposed to drawing attempts with the mouse, a drawing sweep on the workbench is a large scale gesture, which results in pretty smooth curves. We use this method successfully even without doing any software filtering of the data points, merely by applying the standard cubic B-spline interpolation. The object in Figure 1 has been designed with this technique on the workbench. On the net curves to which the new curve comes close, intersection points with the new curve are estimated. The final curve is an approximation of the new curve which interpolates the intersection points. A similar technique is used for maintaining the connectivity in the net when a single net curve is changed.

5.2 Extracting the topology

For the extraction of the topology, we use a combination of an automatic loop searching module based on geometric criteria and the explicit selection of successive curve pieces that are supposed to form a loop. A closed loop is a set of successive curve pieces that surround a surface.

5.3 Creating the surfaces

A method of generating surfaces from a network of curves that has arbitrary topology has been proposed by Kuriyama [20] for the quick input of surfaces in 3D-CAD systems during the initial stage of design.

From m surrounding curves, a patch is computed by transfinite interpolation of neighboring curves and then blending the subsurfaces. The patches have the nice property of G^n continuity with neighbouring patches if all curves surrounding it have C^n continuity.

These surfaces are supported in our system. However they cannot be deformed directly. Therefore we implemented the surface styling tools from section 4 for bicubic B-spline surfaces and implemented them already for Catmull-Clark surfaces. We are currently integrating the Catmull-Clark surfaces into our system.

5.4 Two-handed modeling

The non-dominant hand assists the modeling hand in drawing curves by controlling the position and orientation of a modeling coordinate system dynamically or by fixing it at a preferred place, following the suggestions from Shaw et al. [7, 31]. In this way, it is easier to deal with space curves. Alternatively, the non-dominant hand controls a virtual transparent drawing plane onto which curves are projected, or it applies symmetry planes globally or locally.

5.5 Manual tool selection

We implemented 3D context-based menues which are variable in position to support modeling applications on the workbench. The goal is to integrate the selection of tools in the workflow in non disturbing way. Object-specific sets of tools can be associated with the existing classes in the system.

To activate a toolbar, the user has to localize an object of the associated class with the pointer. By pressing the pointer-button, a toolbar appears, awaiting a tool selection. The

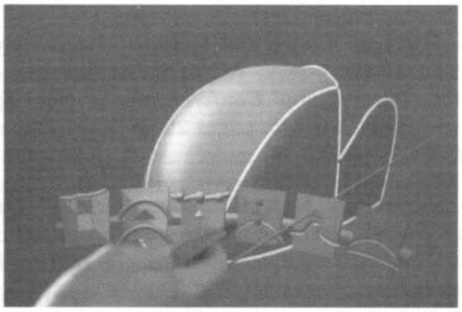

Figure 5: Selecting a tool from the curve toolbar.

toolbar coordinate system is defined such that it's center coincides with the center of rotation of the wrist. In this way, the wrist position is connected to the toolbar so that it can follow the hand. The rotational degrees of freedom of the toolbar are not connected. Heading and pitch are frozen, whereas roll, the rotation about the pointer axis, is set to 0. We achieve a toolbar which is fixed relative to the wrist position but can still be moved. To select a tool, the hand has to turn until the pick ray emanating from the pointer hits the corresponding icon (Figure 5), after that the button should be released. Turning the hand does not affect the toolbar position, so the whole process is comfortable for the user. If the virtual toolbar is hidden by another object, it can be dragged out easily. To properly isolate the toolbar movement from turning the hand around the wrist, it is necessary to define an individual shift vector in the receiver coordinate system of the tracking hardware for each user seperately. We found that a curved horizontal arrangement of the tools corresponds very well to turning the hand, if the radius of the configuration equals the distance of the wrist to the toolbar.

The technique described is similar to Liang's ring menue [21], which has proven to be very useful. In addition to their solution, our technique allows a moving menue, following the hand. The initial orientation of the menue relative to the user can be influenced by the direction the user points at the object. We found the flexibility of the menue position and it's initial orientation to be very useful in Virtual Environments with large workspaces.

6. FUTURE WORK

6.1 Gesture based tool selections

In the last section, we used the pick ray emanating from the pointer to identify an object and therefore an associated set of tools. To reduce this set further or even to select a certain tool, we can exploit three-dimensionality by evaluating the position and orientation of the pointer relative to a coordinate system defined on the object in a reference point. This point could be the intersection of the object with the pick ray or another point close to the pointer position. For a curve, the Frenet Frame may be used, for a surface the system of directional derivatives could be a reasonable choice.

We explain our approach by means of the curve, because the tool set for curves can contain many related and unrelated tools for editing, copying or deleting curves. Our objective is to find intuitive criterions for the position and orientation of the pointer relative to the curve which could be interpreted as an intention of the user to perform a certain manipulation on the curve. We evaluate the angle $\alpha, 0^\circ \leq |\alpha| < 180^\circ$ of the pointer to the curve tangent and the distance d of the pointer's tip to the reference point $\mathbf{p}$ on the curve and select a tool according to table 1. Figure 6 demonstrates how the tool selection depends on the geometric properties.

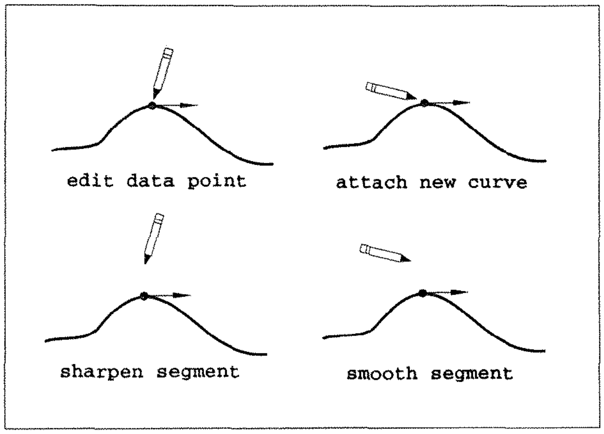

Figure 6: Holding the pointer for curve tool selection.

Note that the approach described works only in a 3D Virtual Environment. In this way we can exploit all properties of the RWB to integrate and support the selection of tools in 3D.

curve tools	$60^\circ \leq \|\alpha\| \leq 120^\circ$	$\|\alpha\| \leq 30^\circ$ or $\|\alpha\| \geq 150^\circ$
$d \leq 2cm$	edit data point	attach new curve
$2 < d \leq 4cm$	sharpen segment	smooth segment

Table 1: Gesture based selection of curve tools

6.2 Force feedback

The drawback of "interacting in the air", which occurs particularly with regard to elastic deformations, will be attacked by integrating a large version of the PHANToM^{TM} [24] device into the workbench environment. In contrast to this, controlling the smoothing and sharpening tools from sections 3 and 4 works very well even without force feedback since the corresponding arm movements are rather indirect.

7. CONCLUSION

This paper has presented curve and surface styling tools which can support the intuitive transfer of the design intent into virtual models. The tools are designed especially for free-form modeling applications in Virtual Environments. A modeler on the Responsive Workbench (RWB)^{TM} has been described that supports the quick input of surfaces by allowing the user to draw the primal curves of a model directly in 3D. Characteristics of 3D Virtual Environments have been exploited to develop approaches in the field of interface design, that help to reduce the complexity of interaction processes in modeling applications in 3D.

User studies have to evaluate the usefulness of the proposed techniques to suggest further improvements, especially for the manual and implicit tool selection mechanisms from sections 5.5 and 6.1.

Although reliable evaluations are missing, users reported that the toolbar technique from section 5.5 can alleviate interaction with menues compared to a fixed menue location on the workbench. This technique has been integrated in visualization projects for automotive and medical applications developed in our group.

8. ACKNOWLEDGEMENTS

Many thanks to our colleague I. Nikitin who programmed one of the user controlled loop selection methods by pointing at the loop directly in space. We also thank M. Göbel and M. Reiser who support this work and made it possible and H. P. Seidel for his valuable comments on the modeling system. The idea of the shell-like shape modeled for Figure 1 comes from F. Dombois. B. Fröhlich implemented general tools for 3D interaction, which have been used as a basis for the work described in 5.5. Many colleagues in our group are involved in the development of the AVANGO^{TM}[4] framework [33], on top of which the application described here has been realized.

9. REFERENCES

[1] J. Butterwoth, A. Davidson, S. Hench, and T. Olano. 3dm: A three dimensional modeler using a head-mounted display. *Communications of the ACM*, pages 55–62, June 1992.

[2] W. Buxton, G. Fitzmaurice, R. Balkrishnan, and G. Kurtenbach. Large displays in automotive design. *IEEE Computer Graphics and Applications*, pages 68–75, July/August 2000.

[3] J. H. Clark. Designing surfaces in 3-d. *Communications of the ACM*, 19(8):454–460, August 1976.

[4] L. Cutler, B. Fröhlich, and P. Hanrahan. Two-handed direct manipulation on the Responsive Workbench. In *Proc. Symposium on Interactive 3D Graphics*, 1997.

[5] T. H. Dani and R. Gadh. Creation of concept shape designs via a virtual reality interface. *Computer-Aided Design*, 29(8):555–563, 1997.

[6] J. Deisinger, R. Blach, G. Wesche, R. Breining, and A. Simon. Towards immersive modeling - challenges and recommendations: A workshop analysing the needs of designers. In *Eurographics Workshop on Virtual Environments 2000*, pages 145–156, 2000.

[7] M. Green, J. Liang, and C. Shaw. Interactive 3d geometrical modelers for virtual reality and design. In *Proc. International Conference on Virtual Systems and Multimedia '95*, pages 29–36, Gifu, Japan, September 1995.

[8] A. D. Gregory, S. A. Ehmann, and M. C. Lin. InTouch: Interactive multiresolution modeling and 3d painting with a haptic interface. In *IEEE VR 2000 Proceedings*, pages 45–52, 2000.

[9] G. Greiner, J. Loos, and W. Wesselink. Surface modeling with data dependent energy functionals. In *Eurographics '96*, volume 15, pages 97–110, 1996.

[10] G. Greiner and H.-P. Seidel. Automatic modeling of smooth spline surfaces. In N. Magnenat-Thalmann and V. Skala, editors, *Proc. WSCG '97*, pages 665–675, 1997.

[11] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19:486–517, 1987.

[12] R. Gupta, D. Whitney, and D. Zeltzer. Prototyping and design for assembly analysis using multimodal virtual environments. *Computer-Aided Design*, 29(8):565–574, 1997.

[13] C. Hummels, A. Paalder, C. Overbeeke, P. Stappers, and G. Smets. Two-handed gesture-based car styling in a virtual environment. In *30th ISATA Proceedings, Mechatronics*, 1997.

[14] C. Hummels, G. Smets, and C. Overbeeke. An intuitive two-handed gestural interface for computer supported product design. In *Bielefeld Gesture Workshop*, September 1997.

[15] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3d freeform design. In *SIGGRAPH '99 Proceedings*, pages 409–416, 1999.

[16] Sensable Technologies Inc. FreeForm^{TM} modeling system. In *http://www.sensable.com/freeform*, 1999.

[17] S. Jayaram, H. I. Connacher, and K. W. Lynos. Virtual assembly using virtual reality techniques. *Computer-Aided Design*, 29(8):575–584, 1997.

[18] F. L. Krause and J. Lüddemann. Virtual clay modeling. In *Proceedings IFIP WG 5.2, Geometric Modeling for CAD*, 1996.

[19] W. Krüger, C. A. Bohn, B. Fröhlich, H. Schüth, W. Strauss, and G. Wesche. The Responsive Workbench: A virtual work environment. *IEEE Computer*, 28(7):42–48, 1995.

[20] S. Kuriyama. Surface modelling with an irregular network of curves via sweeping and blending. *Computer-Aided Design*, 26(8):597–606, 1994.

[21] J. Liang and M. Green. Geometric modeling using six degrees of freedom input devices. *Computers and Graphics*, 18(4), 1994.

---

[4] AVANGO is a trademark of GMD, Germany, 2000

[22] J. Liang and M. Green. JDCAD: A highly interactive 3d modeling system. *Computers and Graphics*, 18(4), 1994.

[23] L. Markosian, J. M. Cohen, T. Crulli, and J. Hughes. Skin: A constructive approach to modeling free-form shapes. In *SIGGRAPH '99 Proceedings*, pages 393–400, 1999.

[24] T. H. Massie and J. K. Salisburg. The phantom haptic interface: A device for probing virtual objects. In *Proc. 1994 ASME International Mechanical Engineering Congress and Exhibition*, pages 295–302, Chicago, 1994.

[25] K. Matsuda, S. Sugishita, Z. Xu, K. Kondo, H. Sato, and S. Shimada. Freehand sketch system for 3d geometric modeling. In *1997 International Conference on Shape Modeling and Applications*, pages 55–62, 1997.

[26] H. Nishino, K. Utsumiya, and K. Korida. 3d object modeling using spatial and pictographic gestures. In *VRST '98 Proceedings*, pages 51–58, Taipeh, Taiwan, November 1998.

[27] H. Nishino, K. Utsumiya, K. Korida, A. Sakamoto, and K. Yoshida. A method for sharing interactive deformations in collaborative 3d modeling. In *VRST '99 Proceedings*, pages 116–123, London, December 1999.

[28] H. Qin, C. Mandal, and B. C. Vermuri. Dynamic catmull-clark subdivision surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 4(3):215–229, July-September 1998.

[29] E. Sachs, A. Roberts, and D. Stoops. 3-draw: A tool for designing 3d shapes. *IEEE Computer Graphics & Applications*, pages 18–26, November 1991.

[30] S. Schkolne and P. Schröder. Surface drawing. Technical Report CS-TR-99-03, Caltech Department of Computer Science, 1999.

[31] C. Shaw and M. Green. THRED: A two-handed design system. *Multimedia Systems Journal*, 5(2), March 1997.

[32] D. Terzopoulos and H. Qin. Dynamic nurbs with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 13(2):103–136, April 1994.

[33] H. Tramberend. Avocado: A distributed virtual reality framework. In *IEEE VR 1999 Proceedings*, pages 14–21, 1999.

[34] M. Usoh, M. Slater, and T. I. Vassilev. Collaborative geometrical modeling in immersive virtual environments. In *3rd Eurographics Workshop on Virtual Environments*, Monte Carlo, 1996.

[35] C. van Dijk. Conceptual surface modeling for industrial design. In *IFIP Transactions B-9*, pages 271–278. Elsevier, North-Holland.

[36] A. van Elsas, A. van den Hout, J. Vergeest, and W. Bronsvoort. Automatic topology extraction from an irregular network of sketched 3d curves. In *28th ISATA Proceedings, Mechatronics*, 1995.

[37] R. C. Veltcamp and W. Wesselink. Modeling 3d curves of minimal energy. In *Eurographics '95*, volume 14(3), pages 97–110, 1995.

[38] W. Welch and A. Witkin. Variational surface modeling. In *SIGGRAPH '92 Proceedings*, volume 26(2), pages 157–166, 1992.

[39] W. Wesselink. *Variational Modeling of Curves and Surfaces.* Dissertation, Technical University Eindhoven, The Netherlands, 1996.

[40] W. Wesselink and R. C. Veltcamp. Interactive design of constrained variational curves. *Computer-Aided Geometric Design*, 12(5):533–546, August 1995.

[41] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. Sketch: An interface for sketching 3d scenes. In *SIGGRAPH '96 Proceedings*, pages 163–170, 1996.