A Virtual Surface Modeler for Direct and Regional Free Form Manipulation

Juli Yamashita, Hiroshi Yokoi, Yukio Fukui, and Makoto Shimojo

National Institute of Bioscience and Human-Technology, AIST, MITI
1-1, Higashi, Tsukuba, Ibaraki 305, Japan
E-Mails: juli@nibh.go.jp, fukui@nibh.go.jp, hyokoi@nibh.go.jp, and shimojo@nibh.go.jp

[Abstract] This paper presents a new virtual surface modeler "ViSurf," which allows users to deform a free formed surface directly with haptic feedback. To sculpt a surface, a user only needs to "push" it directly wherever s/he likes; then the surface will be deformed as it is pushed. ViSurf's direct form manipulation interface is based on a newly developed technique called Regional Direct Deformation Method (RDDM), which automatically controls form defining parameters to obtain appropriate deformation of user-specified region when the cursor collides with a form. RDDM is independent of underlying form representing parameters; users are free from worrying about inserting new knots to obtain deformations. RDDM provides various kinds of deformations; to distinguish and utilize these deformations easily and properly, we will introduce "tools," such as "trowels" that create flat areas. A tool represents the characteristics of its deformation, such as region and shape. ViSurf also offers a haptic virtual reality with a newly developed force feedback device used as its 3-dimensional input/output device. A user can feel haptic feedback on one's hand when s/he "pushes" a surface, that gives a user more certain feeling in form deformation process.
[Key Words] CAD, Direct Manipulation, Tools, Free Forms, Force Feedback, and Virtual Reality.

1. Introduction

1.1. The Problem

Direct manipulation is one of the most important paradigms in human-computer interaction, especially in manipulating free forms on CAD (Computer Aided Design) systems which need intuitive interface. Direct form deformation interface becomes much more important when virtual reality (VR) technologies are applied to CAD. However, existing CAD systems do not offer direct manipulation interface of free forms represented parametrically. A user can not touch forms themselves, but can deform them indirectly by altering parameters that define forms, such as control points and weights. Such indirect ways of deformation have many problems. For example, parameters that users can utilize to alter forms differ according to form representations. In addition, different kinds of parameters have different effects in deformation. Designers have to learn how many kinds of parameters they have and the effect of each parameter; therefore, each time they deform forms, users have difficulty in determining which parameters should be used and how far they should be changed in order to alter shapes as they want. A method of deforming free forms directly is strongly needed.

1.2. Review of Existent Direct Manipulating Techniques

In recent years, several direct free form manipulating techniques and systems have been proposed to release designers from the pain of indirect form deformation methods. Although, such techniques and systems are not adequate, users still have to be aware of underlying parameters that define forms. Let us review existent techniques briefly to clarify their problems from the stand point of interactive form design.

(1) Constraint Based Approach

Constraint based approaches [Bart89, Welc91, Welc92, Fowl93, Hsu92] provide users with point and curve constraints as handles for direct manipulation. A user can set such constraints as points and curves on a form wherever they want, and the form will be deformed to satisfy these constraints. In other words, you can drag a point on a surface to deform it. Then the system solves inverse functions of form representations and determines new parameters to obtain the deformation. Local geometry manipulation [Geor92] is similar to constraint based techniques, which allows users to manipulate surface geometries, such as tangent vectors and curvature, at any point of a curved surface.

A limitation of these approaches is their locality; the region deformed is determined by the underlying parameters that define the form because they solve inverse functions of form representations. Only [Hsu92] appears to be able to control the region of deformation by the resolution of FFD [Sede86] control lattice around objects to be deformed, however, the reconfiguration of the lattice is still being researched.

Dragging and solving inverse functions cause another problem. Sometimes users may get undesirable deformation, especially when they drag a point near special points like cusps. To avoid the problem, we need a new way to utilize shapes, which is independent of parameter values.

Point and curve constraints are not sufficient to sculpt free formed surfaces. For example, a plane part on a free form can not be defined only with point and curve constraints. Surface constraints should be introduced.

(2) Simulation Based Approach

In the context of virtual reality (VR) technique and animation, physically-based models [Terz88, Pent90, Essa92] and voxel based systems [Galy91, for example] are natural selections. These approaches simulate physical features of real materials, such as, elasticity and volume preservation, often by way of FEM (finite element method).

The largest problem of these methods is calculation cost; fine physical simulation consumes too much computational power to keep good interactivity. Even though users may have enough computational power, it is doubtful whether such physical features really help form design or not; for example, volume preservation will give both desirable dents of shapes as well as undesirable rises. Elasticity must be an obstacle to form design that is an essentially inelastic deformation process.

Another problem is their locality. Region control of deformation is also difficult, especially with voxel based systems. To realize the deformation of wider area of voxel data such as bending, there has been an attempt to bridge the gap between voxel and parametric form representations [True92], which is not a complete success yet. [Terz91] gives an approach to locally and globally deformable models by introducing a special form representation which is a combination of superquadrics and spline. Generalizing the method to widely used parametric representations such as B-Spline and Bezier is its future problem.

1.3. Scope of this Paper

The objective of this paper is to provide users with a direct form deforming interface which fulfils the following requirements:

- simple and easy to understand,
- deformation regions are controllable,
- independent of form underlying parameters' conditions, and
- highly interactive utilizing not only visual, but haptic feedback to users.

To achieve this goal, we will present a new direct virtual surface modeler "ViSurf." On ViSurf, a user can deform a virtual form by just pushing it with a cursor, and the form will be deformed as it is pushed. At the same time, the user gets force feedback on his/her hand when s/he touches and pushes the form. ViSurf also offers "tools," which give their own deformation characteristics such as shapes and regions. For example, a "trowel" tool makes a region flat. The interface is based on a new direct form deforming technique named Regional Direct Deformation Method (RDDM), which automatically calculates new parameters from old parameters, user's deforming actions, and the tool used for deformation. RDDM offers various kinds of deformations with region control, which can be represented as tools. ViSurf also has a newly developed input device with force feedback to provide a user with haptic feedback on his/her hand when s/he "pushes" a form.

2. ViSurf System

ViSurf system consists of two modules (Figure 1). One is a modeler with direct deformation interface based on RDDM (Regional Direct Deformation Method), that contains data of virtual forms represented in Non Uniform B-Spline curves and surfaces. (This paper assumes that readers have enough knowledge about B-Spline. Please refer to [Fari88] etc., if needed.) The other is an input device with force feedback which is a 3-dimensional cursor as well as an output device of haptic virtual reality.

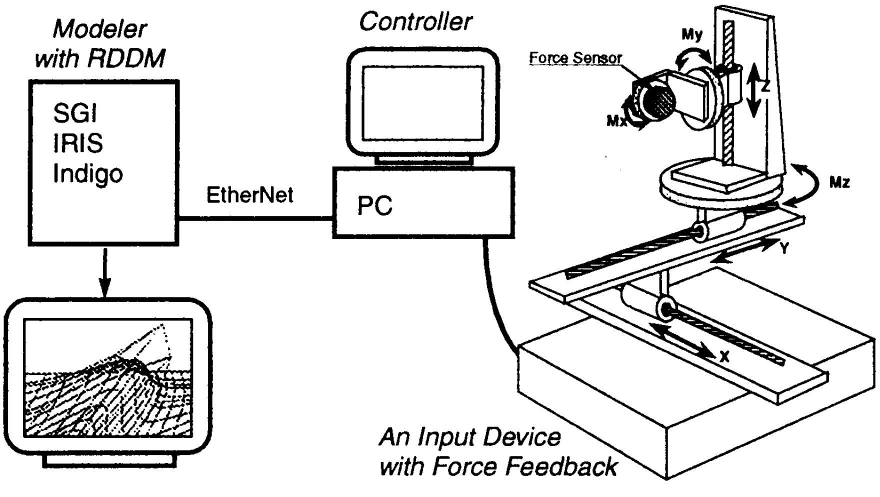

Figure 1. System configuration of ViSurf ViSurf system consists of two modules connected by EtherNet; a modeler with direct deformation interface based on RDDM (Regional Direct Deformation Method) and an input device with force feed back.

2. 1. Formulating Regional Direct Deformation Method (RDDM)

2. 1. 1. Introducing Direct Deformation Method (DDM)

Direct Deformation Method (DDM) [Yama93] gives the basic concept of ViSurf's direct form manipulation interface, "just push and deform." DDM's idea is simple; when a cursor "pushes" a form, or, collides with a form, then the system automatically calculates new parameters of a deformed form according to "deforming function" f as follows:

$$\{\text{New parameters}\}=f\{\{\text{Old parameters}\},\{\text{Cursor's loci}\}\}$$

then the form is deformed as it is pushed. For more detailed description of DDM, please refer to [Yama93].

As shown in Figure 2, DDM gives various kinds of deformations. For example, Fig. 2-a shows the deforming effect by moving the nearest control point to the pushed point of the curve; it means that DDM gives such deforming functions that are equivalent to one parameter's change. Altering many parameters at one time gives deformations of wider area (Figure 2-b). DDM can also offers deformations with restrictions and adding/deleting parameters. Fig. 2-c is an angle made by inserting new knots and by introducing restrictions to control points. DDM is effective for 3D surface deformation as well (Fig. 2-d).

2. 1. 2. Introducing Regional DDM (RDDM)

Since DDM is a general framework of direct manipulation, the region it affects is implicitly defined by its deforming function and underlying form parameters. To deform user-specified regions of free forms, we introduce Regional DDM (RDDM), which is a subset of DDM with regional constraints. RDDM's algorithm for deformation of a B-Spline curve of degree d is as follows (see Figure 3):

(1) Specify a region of a curve C by the user. The specified region C_r can be described by a parameter $t \in [t_b, t_e]$, where $t_b < t_e$.

(2) Insert new knots at t_b and t_e. Each curve segment begins at t_b and t_e consists of control points $\{V_i, V_{i+1}, ..., V_{i+d}\}$ and $\{V_j, V_{j+1}, ..., V_{j+d}\}$, respectively, where d is the degree of the curve. To guarantee a regional deformation between t_b and t_e, only those control points from V_{i+d} to V_{j-1} can be moved. If there is not enough control points, the system has to automatically insert some new knots to ensure movable control points.

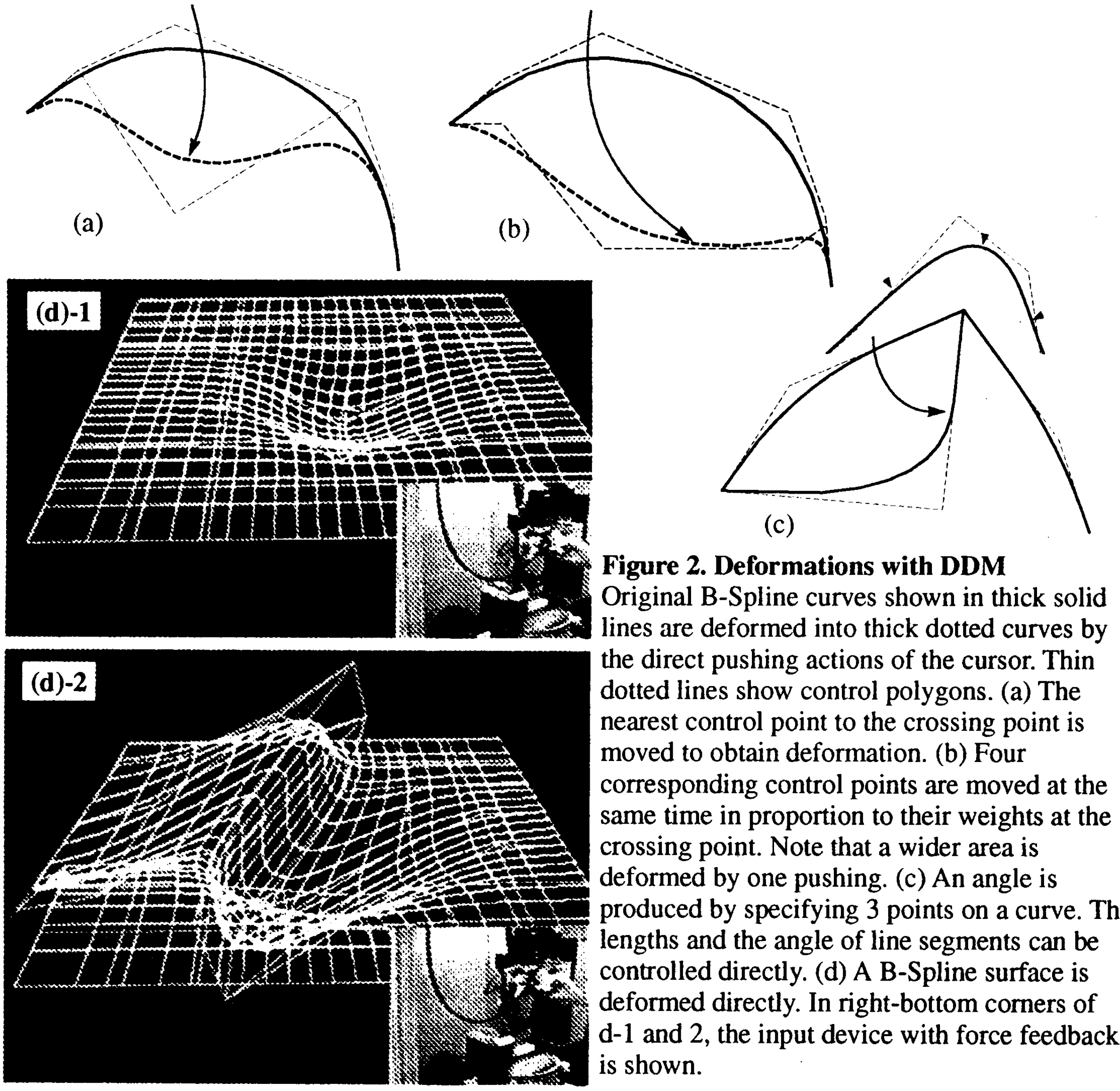

Figure 2. Deformations with DDM
Original B-Spline curves shown in thick solid lines are deformed into thick dotted curves by the direct pushing actions of the cursor. Thin dotted lines show control polygons. (a) The nearest control point to the crossing point is moved to obtain deformation. (b) Four corresponding control points are moved at the same time in proportion to their weights at the crossing point. Note that a wider area is deformed by one pushing. (c) An angle is produced by specifying 3 points on a curve. The lengths and the angle of line segments can be controlled directly. (d) A B-Spline surface is deformed directly. In right-bottom corners of d-1 and 2, the input device with force feedback is shown.

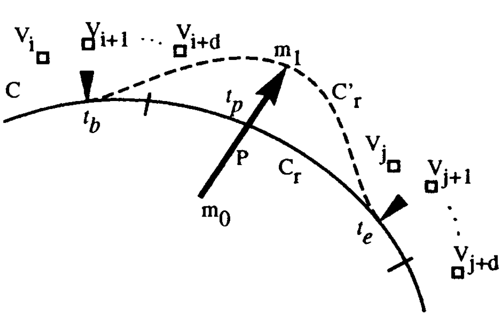

Figure 3. Principle of Regional DDM (RDDM)
A user-specified region C_r, which is a part of B-Spline curve C shown in solid line, is deformed into C'_r, the dotted curve, by the pushing action. To guarantee regional deformation, RDDM creates new parameters and gives restrictions on them. In this example, new knots are inserted at t_b and t_e, and some more between them if necessary. RDDM can provide various kinds of deformation shapes and users can deform a region without taking care of underlying form defining parameters.

(3) Check the cursor crossing. If it crosses the curve between t_b and t_e, then change parameters $\{ V_k \mid i+d \leq k \leq j-1 \}$ and $\{ t \mid t_b \leq t \leq t_e \}$ to deform the curve C_T according to the crossing motion of the cursor.

Figure 4 shows examples of deformations with RDDM. Note that the system automatically inserts new knots to guarantee regional deformation and that deformations of various shapes can be offered to the same region.

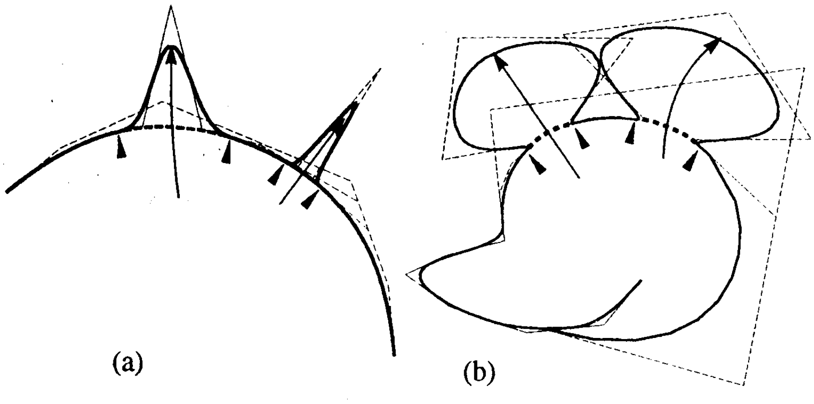

Figure 4. Deformation with RDDM Thin dotted lines are control polygons and thin solid curves with arrowhead show cursor loci. (a) Deformation of user-specified regions. New control points are created by RDDM to give deformation of narrower regions. (b) A variation of deformation shape. The specified region swells out when pushed by the cursor. The effect is used to create ears of an animal face.

2. 2. Introducing Tools

When a user needs deformation of the same size and shape many times, it must be time-consuming to specify a region before each deformation. It must also be difficult to choose a proper deformation shape among various kinds of deformations provided by RDDM. To solve these problems, we propose an interface with "virtual tools." Example tools and their deformation effects are shown in Figure 5. The size and the shape of a tool represents the region and shape of deformation, respectively. For example, a flat "trowel" produces a plane part of its size (Figure 5-a). A user only needs to push a form with a tool to obtain a regional deformation according to its size and shape.

The following outlines the algorithm of a 2D trowel shown in Fig. 5-a:

(1) Track the position of a tool in use and check whether it collides with a form. If it does, then go to #2 process. Note that the collision detection algorithm is specific to each tool. In the example of trowel, the tool is expressed as a line segment l and the system checks collision with a B-Spline curve C.

(2) Calculate new form defining parameters of the deformed form according to the deforming function f_{TOOL} specific to each tool. Generally, new parameters can be described as follows:

$$\{\text{New form parameters}\} = f_{TOOL}\{\{\text{Old form parameters}\}, \{\text{Tool's loci}\}\}$$

[Notes]

- A tool's "locus" includes both its location and direction.
- The region and the shape of deformation are defined by f_{TOOL}.
- Algorithms that realize f_{TOOL} differ according to the representation of the form to be deformed.
- The same deforming effect can be realized in several ways, or, several tools may offer the same deformation.

In the example of a trowel, suppose l crosses C at points Pt1 and Pt2, whose knot value are t1 and t2, where t1 < t2. Main part of the algorithm of f_{TROWEL}, the deforming function of a trowel, can be outlined like this:

```
for ( val = t1 , t2 )
    if ( val != knot point) { insert a knot value val into the knot vector of C }
n1 = curve segment number of t1 ;
n2 = curve segment number of t2 + degree of C ;
for ( i = n1;  i < n2;  i++)  V[i] = ( Pt1 * ( n2 - i + 1 ) + Pt2 * ( i - n1 ) ) / ( n2 - n1 + 1 ) ;
```

Then the region pushed by the trowel becomes straight.

(3) Display the deformed form with new parameters and return to #1.

Fig. 5-b, c and d show deformations of B-Spline surfaces of degree 3 by 3D tools.

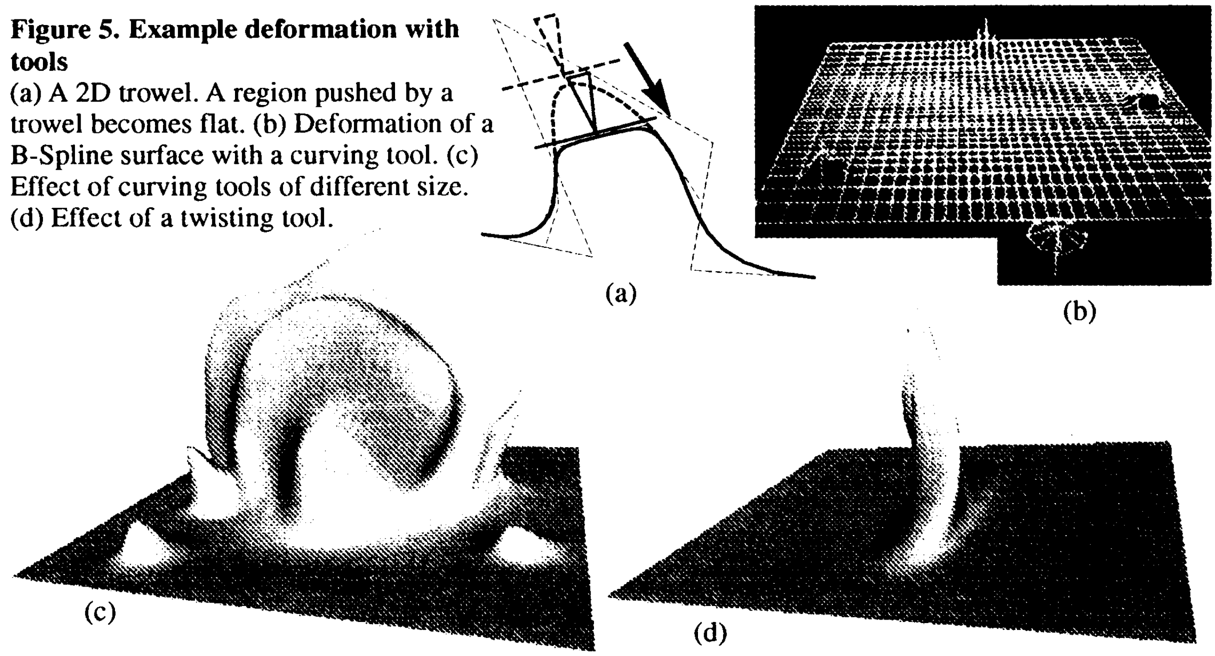

Figure 5. Example deformation with tools
(a) A 2D trowel. A region pushed by a trowel becomes flat. (b) Deformation of a B-Spline surface with a curving tool. (c) Effect of curving tools of different size. (d) Effect of a twisting tool.

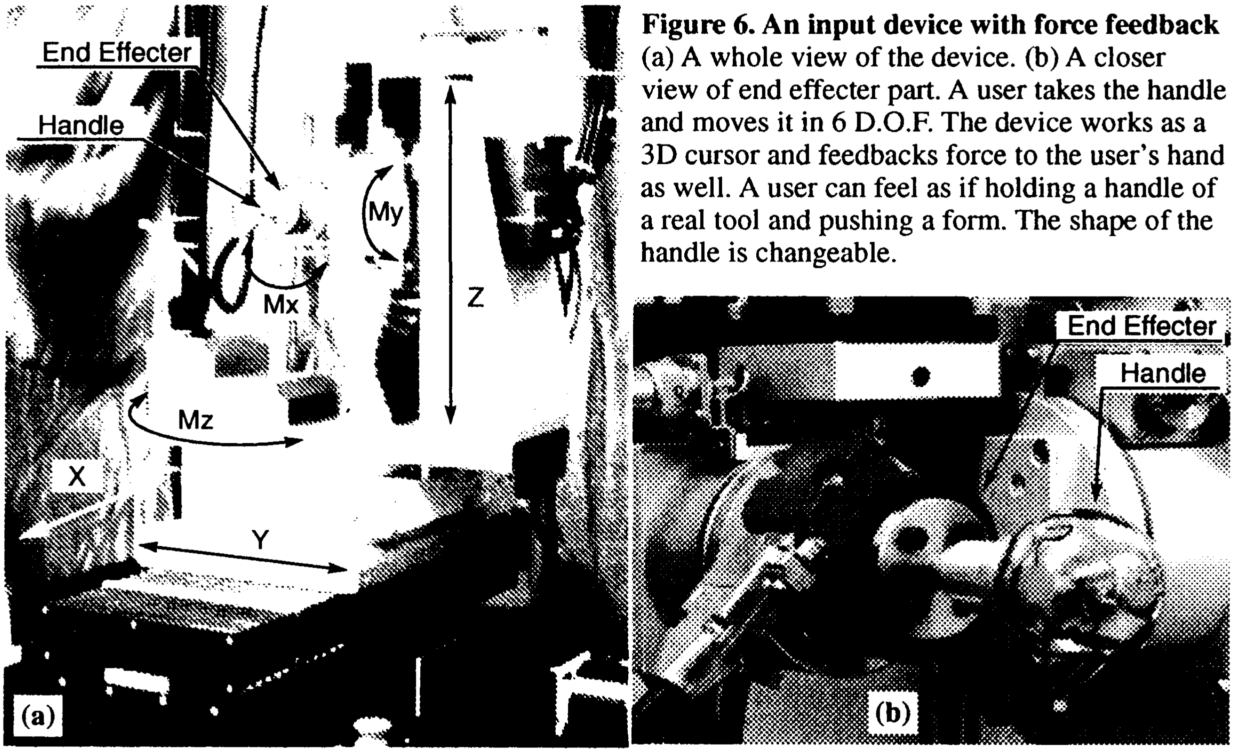

Figure 6. An input device with force feedback
(a) A whole view of the device. (b) A closer view of end effecter part. A user takes the handle and moves it in 6 D.O.F. The device works as a 3D cursor and feedbacks force to the user's hand as well. A user can feel as if holding a handle of a real tool and pushing a form. The shape of the handle is changeable.

2. 3. Input Device with Force Feedback

ViSurf has a newly developed input device with force feedback, which will be introduced briefly here. (For more detailed description, please refer to [Yoko94] in this proceedings.) The device is a 3-dimensional cursor and gives haptic feedback on the user's hand when it touches a virtual surface. The device is a Cartesian type manipulator (see Figure 6) of 6 D. O. F. (degrees of freedom) at the end effecter position. It can move along X, Y, and Z axes and can rotate around each axis. The device has force sensors of 6 D. O. F. at its end effecter to pick up the force applied by the user's hand; then the detected force data are input to its controller that adopts velocity and position control to drive the device freely and to feedback appropriate force to the user.

3. Discussion and Future Work

The most distinctive features of ViSurf's direct deformation interface are its simplicity and flexibility. A user can push a form wherever s/he likes directly to deform it and does not need to take care of underlying parameters that define the form. RDDM offers various kinds of deformations that can be represented by tools. A user can select proper tools easily because the shape and the size of a tool represent the size and the shape of its deformation. ViSurf also has a potential to be representation-independent, if tools with the same deforming effect can be developed on other representations than B-Spline.

ViSurf's tool-based interface has 3 large merits:

(1) It enables introducing surface constraints to free forms. To create a flat region on a free form, pushing it with a "trowel" is much more intuitive compared to applying point and curve constraints. Not only a trowel, but other tools can be "casted" to introduce shape constraints.

(2) It opens a door to description of form deformation process by tools and their loci. Once the process is described, you can edit it later to get another deformation by changing tools and editing their loci. You can record and share the process of your design with others, possibly via networks.

(3) It is suitable for object-oriented programming. A tool is an instantiation of a particular class of deformation provided by RDDM. The region and the shape of a tool are its instance variables and methods. When a tool detects collision with a form, the tool sends a message to tell the form how much to be deformed. In addition, the idea of object-orientation is suitable for massively parallel computing technology. To improve interactivity of ViSurf, we can reconstruct our system according to object-oriented programming manner to take the advantage of parallel computing.

ViSurf's 3D input device with force feedback improves direct manipulation interface a great deal because a user can feel the tool touching a surface. Hardness of a surface, or how much force a user feels to deform it, is controllable. Now we have an interesting and challenging research field of multi-sensory feedback: How much force should be feedbacked for comfortable deformation? How different and how helpful are visible, audible, and haptic feedbacks? etc. In addition, cooperation with the force feedback device has great potential, such as showing design restrictions haptically.

In addition, ViSurf system has the following points that need to be improved in the future:

- How to select a set of desirable deforming tools? A set of tools we provide is the basis of deformation that can be created. It is an interesting problem to find a minimal tool set for a given class of deformation. For designers, an environment to create their own tools as they want will be needed.
- More variation in tool usage. In the real world, a tool can be used in many ways; sometimes a designer uses only its edge or only a part of it. ViSurf does not allow such partial usage of tools, however, it is an interesting problem.
- Deformations with topological change are not realized. For example, it is not possible to penetrate a hole on a surface with ViSurf now. However, it is not a problem with form deforming interface but with form representations.
- Evaluation of usability. No standard way to evaluate system usability has been established yet, however, we want to measure how useful our system is and how effective haptic feedback is in form deformation. If a kind of benchmark test is developed, we can compare different systems for free form manipulation.

4. Summary

A new direct virtual free formed surface modeler "ViSurf" was presented. With ViSurf, a user can deform a surface directly by "pushing" it with a cursor/tool. The system provides various kinds of deformations which can be represented as tools; each tool has its own deformation characteristics, such as the region it affects and the shape it creates. The "push and deform" interface releases users from taking care of underlying parameters that define the form; instead of users, the system automatically creates new parameters according to RDDM (Regional Direct Deformation Method). ViSurf also has a 3-dimensional input device with force feedback, with which a user can feel force on her/his hand when s/he pushes a form. The haptic feedback gives a user more certain feeling in form deformation.

References

[Bart89] Bartels, R. H. and J. C. Beatty: "A Technique for the Direct Manipulation of Spline Curves", *Proc. of Graphics Interface '89,* 1989.

[Essa92] Essa, I. A., S. Sclaroff, and A. Pentland: "Physically-based Modeling for Graphics and Vision", *M.I.T. Media Lab. Perceptual Computing Group Tech. Report No. 184,* 1992.

[Fari88] Farin, G. E.: "Curves and Surfaces for Computer Aided Design", Academic Press, Boston, 1988.

[Fowl93] Fowler, B. and R. Bartels: "Constraint-Based Curve Manipulation", *IEEE Computer Graphics & Applications,* 1993.

[Galy91] Galyean, T. A. and J. F. Hughes: "Sculpting: An Interactive Volumetric Modeling Technique", *Proc. of SIGGRAPH '91,* 1991.

[Geor92] Georgiades, P. N. and D. P. Greenberg: "Locally Manipulating the Geometry of Curved Surfaces", *IEEE Computer Graphics & Applications,* 1992.

[Hsu92] Hsu, W. M., J. F. Hughes, and H. Kaufman: "Direct Manipulation of Free-Form Deformations", *ACM Computer Graphics,* vol. 26, No. 2, 1992.

[Pent90] Pentland, A., I. Essa, M. Friedmann, B. Horowitz, and S. Sclaroff: "The ThingWorld Modeling System: Virtual Sculpting by Modal Forces", *ACM Computer Graphics,* vol. 24, No. 2, 1990.

[Sede86] Sederberg, T. W. and S. R. Parry: "Free-Form Deformation of Solid Geometric Models", *Proc. of SIGGRAPH '86,* Vol.20, No.4, 1986.

[Terz88] Terzopoulos, D. and K. Fleischer: "Deformable models", *The Visual Computer,* 14, Springer-Verlag, 1988.

[Terz91] Terzopoulos, D. and D. Metaxas: "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 13, No. 7, 1991.

[True92] True, T. J. and J. F. Hughes: "Volume Warping", *Proc. of IEEE Visualization '92,* 1992.

[Welc91] Welch, W., M. Gleicher, and A. Witkin: "Manipulating Surfaces Differentially", *Proc. of Compugraphics '91,* Springer-Verlag, 1991.

[Welc92] Welch, W. and A. Witkin: "Variational Surface Modeling", *ACM Computer Graphics,* Vol.26, No.2, 1992.

[Yama93] Yamashita, J. and Y. Fukui: "A Direct Deformation Method", *Proc. of IEEE VRAIS'93,* 1993.

[Yoko94] Yokoi, H. and J. Yamashita, Y. Fukui, and M. Shimojo: "Developement of the Virtual Shape Manipulating System", *Proc. of ICAT '94,* 1994.