Deforming NURBS Surfaces to Target Curves for Immersive VR Sketching

Junghoon KWON†, Jeongin LEE†, Nonmembers, Harksu KIM†, Member, Gilsoo JANG††, and Youngho CHAI†, Nonmembers

SUMMARY Designing NURBS surfaces by manipulating control points directly requires too much trial and error for immersive VR applications. A more natural interface is provided by deforming a NURBS surface so that it passes through a given target point, and by repeating such deformations we can make the surface follow one or more target curves. These deformations can be achieved by modifying the pseudo-inverse matrix of the basis functions, but this matrix is often ill-conditioned. However, the application of a modified FE approach to the weights and control points provides controllable deformations, which are demonstrated across a range of example shapes.

1. Introduction

The manufacturing of a product typically begins with a conceptual sketch by a designer, and 3D models are then constructed in a separate modeling phase. Shape modeling programs are insufficiently flexible to allow full play to the intuition of a designer, which is the main reason for a separate modeling process. Recently, there have been many attempts to integrate these processes to allow a model to be created by virtual sketching. The use of more flexible devices, such as a tablet or a 3D mouse, may be part of a solution [1]–[4]. But, a tablet is a 2D device and a 3D mouse is difficult to handle. Therefore, most attempts to facilitate conceptual design are based on immersive virtual environments [5]–[9].

We will describe an approach to the integration of sketching and modeling which is based on a semi-immersive virtual environment with an infrared tracking camera, as shown in Fig. 1. A conceptual designer generally outlines surfaces by a number of curves, shown in perspective in their sketches, and then adds further curves to make the shape of the model more explicit. This suggests that a conceptual sketch might be input as a series of curves in 3D space, and might then be augmented and transformed to create a more complete shape model.

We propose a VR-based sketching system based on the creation and deformation of curves. Our system has some similarities with all of those to be discussed in the related work, but our new calligraphic deformation procedure makes its functionality quite distinct. The deformation process operates on intuitively created curves. The output of the system requires some further improvement in accuracy, but the direct generation of a 3D conceptual model streamlines the design process.

This paper suggests an intuitive NURBS surface deformation algorithm that can be adopted by an immersive virtual sketching system. The concept of a 3D target curve is first introduced before the adjustable range of manipulation is applied through the free-form deformation method to show a NURBS surface manipulation. This deformation method can apply changes to the NURBS surface through the simple input of curves in the immersive virtual sketching system. Our algorithm allows a concept designer, who is used to a 2D drawing environment, to manipulate a NURBS surface freely by drawing curves on a 3D surface. This NURBS surface can be deformed to follow a specified target curve so as to make the process of 3D modeling easy and efficient.

In Sect. 2 we discuss related work, and in Sect. 3 we explain the deformation of a 3D target curve. The use of the modified FEM to produce the deformations is explained in Sect. 4. We conclude this paper in Sect. 5.

2. Related Work

We will now review some of the systems that have recently been proposed for constructing 3D models using VR techniques. 3-Draw [10] and FreeDrawer [6] are sketching systems which demonstrate that developing 3D models on the computer by drawing directly in 3D space is natural and quick. In Surface Drawing [7], the user creates surfaces by moving a hand, instrumented with a special glove, through space, within a semi-immersive interactive...
environment with a 3D display. The user’s hand guides a plane which is used to construct a surface. Another system, HoloSketch [11], supports several types of 3D drawing objects and animations in 3D space.

The user interfaces of these systems consist of a small, keeping number of low-level interactions and a sparse command set, with the aim of allowing the user to express their thinking rapidly; but this means that many of the sophisticated functionalities needed to build complex 3D models are necessarily excluded. For instance, the 3-Draw system can only create lines; HoloSketch is only good at constructing models that are made of its inbuilt primitives, but it cannot readily be extended to a larger class shapes; and Surface Drawing cannot produce neatly formed 3D models, because there is no way of correcting a user’s erratic hand movements.

Existing geometric deformation algorithms fall into two categories: indirect manipulation methods that deform a grid that supports a surface, and direct manipulation methods in which the surface of the model is changed directly. Free-form deformation (FFD) [12] using a grid, deformation using an extended grid [13], and deformation using an arbitrary grid [14] are some of the methods used for the indirect grid manipulation.

Direct manipulation of free-form deformations (DMFFD) [15] is a direct manipulation algorithm which allows a user to edit a NURBS model freely. This algorithm suggests an inverse operation in which changes to NURBS control points are determined be inverting direct changes to points or areas on a NURBS surface. Simple constrained deformation [16], deformation based on a curved surface [17], and area deformation using a numerical formula [18] are different methods of applying changes to the area to be manipulated. Curved surface modeling through the analysis of curves [19] is a method of applying partial changes to a 3D model by drawing curves on a 2D surface.

Direct manipulation of free-form surfaces can take place in an immersive virtual sketching system. Control point repositioning method [20] is one of the well-known shape modification tools and used in a virtual reality environment [21]. However, it is not suitable for a 3D spatial sketching paradigm because it requires selecting the area of the surface to be manipulated and it cannot express the details of the manipulation of a NURBS surface since it only changes the fixed area specified by the control point and basis function. And deformation by means of a pseudo-inverse may cause some distortion of the NURBS surface when the area to be manipulated is smaller than a minimum size determined by the control points and the degree of the surface.

More detailed deformations of NURBS surfaces can be achieved by modified finite-element techniques, which include the direct manipulation of an FE mesh [22] and the combination of FE analysis with the B-spline basis function [23]–[27]. However, these methods are also unsuitable for immersive virtual sketching, since it is difficult to control the deformation by inputting curves.

3. Surface Deformation Using a 3D Target Curve

A designer specifies a 3D target curve [28], [29] near to a NURBS surface, which is then deformed so that a part of the surface includes the curve. This approach is superior to the use of the NURBS control grid because it accords more clearly with a designer’s intuitive abilities.

3.1 Deformation to Target Points Using DMFFFD

The method of direct manipulation of free from deformation (DMFFFD) allows us to deform a NURBS surface so that it includes a given point. A NURBS surface is defined in terms of a bidirectional net of control points in the u, v directions, as follows [20].

\[
S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) P_{i,j}
\]

(1)

If a control point undergoes a known displacement, the resulting deformation of the surface can be expressed as

\[
\Delta q_0 = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u_0) N_{j,q}(v_0) \Delta P_{i,j}
\]

(2)

where \(\Delta P_{i,j}\) is the displacement of control point \((i, j)\), \(N_{i,p}(u_0)\) and \(N_{j,q}(v_0)\) are the basis functions of \(q_0\) in the \(u\) and \(v\) directions respectively. The original surface can now be deformed to include the target point by calculating the displacements of the control points using the pseudo-inverse matrix \(N^+\) as follows:

\[
\Delta P_{i,j} = N^+_{i,p}(u_0) N^+_{j,q}(v_0) \Delta q_{k,l}
\]

(3)

3.2 Input of the Target Curve and the Deformation Process

To deform a surface, the user draws the curve that they want the surface to include. An example target curve is shown in Fig. 2 (a). A set of points on the target curve is used as the target points for the NURBS surface deformation process described above.

The details of this process are as follows:

1. The piece of surface to be deformed is sampled, and a control grid that describes it in NURBS form is constructed. This is the NURBS patch that will be deformed. The numbers of points sampled determines the accuracy of the deformation. A small numbers of points can lead to divergent values of the pseudo-inverse matrix, but too many points requires unacceptably long computation times.

Fig. 2   Deforming a NURBS surface to include a target curve.
2. The user creates a curve using a wand. The curves drawn by the wand consist of a list of points and directions. Each point on the input curve is projected to the nearest point on the NURBS surface to form the projected target curve.

3. The projected target curve is expanded on the NURBS surface to create the area of deformation, shown in Fig. 3. If the distance from a projected point to a target point is \( h \), and the width of the deformation area is \( w \), then a cubic interpolation will satisfy these two equations:

\[
ar^3 + br^2 + cr + d = h
\]

\[
a(r - w)^3 + b(r - w)^2 + c(r - w) + d = 0
\]

The \( u, v \) measures of the sampled points are set as \( r \), and the \( x, y \) and \( z \) values of the coordinates of the deformed points are calculated separately. To obtain a displacement of \( r \), the constants \( a, b, c \) and \( d \) have to be calculated, as follows:

\[
\begin{bmatrix}
a \\ b \\ c \\ d
\end{bmatrix} =
\begin{bmatrix}
r^3 & r^2 & r & 1 & h \\ 3r^2 & 2r & 1 & 0 & 0 \\ (r - w)^3 & (r - w)^2 & r - w & 1 & 0 \\ 3(r - w)^2 & 2(r - w) & 1 & 0 & 0
\end{bmatrix}^{-1}
\]

The deformations of all the sampled points within the deformation area are interpolated in the same way.

4. The sampled points which are connected to projected points are deformed to the target points, and the other sampled points inside the deformation area are interpolated by a fraction of the maximum deformation \( h \), which diminishes with distance \( w \), to achieve a smooth interpolated deformation of the sampled points.

5. The displacement of the NURBS surface is then determined from the pseudo-inverse matrix of the basis function. And the deformed NURBS surface is obtained by applying the displaced control points to the original surface.

The displacement of the sampled points inside the deformation area has to be modified to achieve a smooth deformation (see item 4 above). If the sampled NURBS surface has the same or higher degree than the NURBS surface, then the pseudo-inverse matrix should not be divergent, and the boundary of the displaced surface will be continuous with the original NURBS surface. Figure 4 shows how the deformation of the control points is affected by the location and direction of the target points. The bold black points and lines are target points and their directions, the small points on the NURBS surface are the sampled points, and the circular area on the NURBS surface is the deformation area. The translucent radial shape on the NURBS surface under the target point indicates the target deformation for the sampled points. Since the process is axis-independent, the deformation is possible regardless of the location and direction of the target point.

3.3 Application of the Results, and Some Problems

Figure 5 shows a deformation controlled by two overlapping input curves. This example demonstrates a different deformed surface is produced for the same target curve if the deformation area is different. Even if the two inputs overlap with each other, the deformation process is well-behaved. Figure 6 shows an attempt to make the shape of a face by successive deformations of a NURBS surface to follow a sequence of target curves. Figure 7 compares direct modification of a NURBS surface using the DMFFD.
method with our method. The surface is a cubic NURBS with $7 \times 7$ control grid. The deformation of a NURBS surface is influenced by the basis function, and the displacement of one control point produces a deformation over an area of $4 \times 4$ control points. Both methods require the inversion of a matrix to determine the location of the control points. The circular deformation used in this example required the location of each control point to be calculated using a pseudo-inverse matrix. With the conventional technique, the deformation area is fixed, as shown in Fig. 7 (a), and it is impossible to control its size. Therefore, the locations of surrounding control points have to be changed and the sharpness of the deformation is limited.

The proposed method allows a sharper deformation because a narrow deformation area can be defined, as shown in Fig. 7 (b). By modification of the deformation area, the movement of the surrounding control points can be controlled to a certain extent. If the deformation area is narrower than the boundary of a segment of the surface, a sharper deformation is possible, but the pseudo-inverse matrix may be inaccurate. Figure 7 (b) shows this situation, and the distortion that occurs in the boundary area.

4. A Modified Finite-Element Method for the Deformation of a NURBS Surface

We will now describe an improved method of deforming a NURBS surface so that it passes through a target curve using a modified finite-element method. In this approach, the elements and the basis function and control points of a B-spline, instead of the shape functions and nodes [23]–[27] used for stress analysis etc. Because the finite elements are already available, and there is no need to divide a shape model, surface deformations can be performed in real time. The FE approach also allows the deformation area to be smaller than the sub-patches between the knots of a NURBS surface, which cannot be handled accurately by the DMFFD method.

4.1 The Modified FE Method

Using the FE approach, the target curve is created in the way already described. The gradual NURBS surface deformation gives a change to modify a possible unwanted deformation by changing the direction of movement of control points in the middle of the deformation. Therefore, the deformation is accomplished over several iterations, and is completed when the NURBS surface contains the target curve. Figure 8 shows the FE elements created for NURBS surfaces with quadratic and cubic basis functions. The modified FE method uses a global mass matrix and a stiffness matrix with elements which reflect the extent to which the elements overlap.

As for NURBS surface deformation, the direct stiffness method is used to utilize modified FEM. Galerkin’s method:

$$R(x, y) = \left[ \frac{\partial}{\partial x} \left( k_1 \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_2 \frac{\partial u}{\partial y} \right) - \rho \frac{\partial^2 u}{\partial t^2} + f \right]$$

where $\rho$ is the mass density, $k_1$ and $k_2$ are the stiffness coefficients in the $x$ and $y$ directions, and $f$ is an external force. This is a 2D wave equation. In order to make an elemental description, a residual $R(x, y)$ must be determined by Galerkin’s method:

$$R(x, y) = \left[ \frac{\partial}{\partial x} \left( k_1 \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_2 \frac{\partial u}{\partial y} \right) - \rho \frac{\partial^2 u}{\partial t^2} + f \right]$$

in which the integration over all regions of weighted values of this residual is minimized over all regions as follows:

$$\int_{y_1}^{y_{n+1}} \int_{x_1}^{x_{n+1}} R(x, y) w(x, y) dxdy = 0$$

where $x_1$ and $y_1$ are the coordinates of the element’s first node, $x_{n+1}$ and $y_{n+1}$ are the coordinates of the element’s final node, and $w(x, y)$ is a weight function. By substituting Eq. (8) into Eq. (9) and partially integrating [31], we obtain the elemental description of the 2D wave equation:

$$\int_{y_1}^{y_{n+1}} \int_{x_1}^{x_{n+1}} \left( \frac{\partial w}{\partial x} k_1 \frac{\partial u}{\partial x} + \frac{\partial w}{\partial y} k_2 \frac{\partial u}{\partial y} + \rho w \frac{\partial^2 u}{\partial t^2} \right) dxdy = \int_{y_1}^{y_{n+1}} \int_{x_1}^{x_{n+1}} \left( w k_2 \frac{\partial u}{\partial y} \right) dx + \int_{y_1}^{y_{n+1}} \left( w k_1 \frac{\partial u}{\partial x} \right) dy$$

This equation can be rearranged in terms of $u(t)$, as follows:
\[ \int_{y_1}^{y_{n+1}} \int_{x_1}^{x_{n+1}} \left[ \rho N N^T \dddot{\boldsymbol{u}}(t) + \frac{\partial}{\partial x} N k_1 \frac{\partial}{\partial x} [N^T \boldsymbol{u}(t)] \right. \\
+ \left. \frac{\partial}{\partial y} N k_2 \frac{\partial}{\partial y} [N^T \boldsymbol{u}(t)] \right] dxdy = \text{R.H.S.} \] (11)

This equation can then be converted from a local global coordinate system:

\[ \int_{y_1}^{y_{n+1}} \int_{x_1}^{x_{n+1}} \rho N N^T \dddot{\boldsymbol{u}}(t) |J| dudu \\
+ \int_{y_1}^{y_{n+1}} \int_{x_1}^{x_{n+1}} N \left[ \frac{\partial}{\partial x} \frac{\partial}{\partial y} \right] J^{-T} \\
\left[ k_1 \quad 0 \right. \\
\left. 0 \quad k_2 \right] J^{-1} \left[ \frac{\partial}{\partial x} \frac{\partial}{\partial y} \right] N^T \boldsymbol{u}(t) |J| dudu = \text{R.H.S.} \] (12)

where \( J \) is the Jacobian matrix. The mass matrix and the stiffness matrix of each element is formulated as follows:

\[ M_e = \int_{e} \rho N N^T |J| dudu \] (13)

\[ K_e = \int_{e} N \left[ \frac{\partial}{\partial x} \frac{\partial}{\partial y} \right] J^{-T} \\
\left[ \begin{array}{cc}
    k_1 & 0 \\
    0 & k_2 \end{array} \right] J^{-1} \left[ \frac{\partial}{\partial x} \frac{\partial}{\partial y} \right] N^T |J| dudu \] (14)

In the modified FE method, a shape function can be expressed as the tensor product of the NURBS basis functions in the \( u \) and \( v \) directions:

\[ N_{ri}(n+1,j) = B_i(u)B_j(v)c_{im} \] (15)

The mass and stiffness matrix of each element are made by applying this shape function to Eqs. (13) and (14). The global mass matrix and stiffness matrix can be obtained as shown in Fig. 8, using the direct stiffness method. Finally, the 2D wave equation is reformulated as a linear differential equation which includes the global mass matrix and stiffness matrix.

\[ M_g \dddot{\boldsymbol{u}}_g(t) + K_g \ddot{\boldsymbol{u}}_g(t) = f_g + b_g \] (16)

where \( f_g \) is the global external force and \( b_g \) is the global boundary condition. However, we set \( f_g \) and \( b_g \) to zero. Then, Eq. (16) can be rewritten as follows:

\[ \dddot{\boldsymbol{u}}_g(t) = -M_g^{-1} K_g \ddot{\boldsymbol{u}}_g(t) \] (17)

The real-time displacement of each node can be obtained by numerical integration of this equation, which provides a displacement that allows real-time simulation. Depending on the velocity control algorithm and the target deformation points, deformation begins with a different initial velocity for each node. As shown in Fig. 9, the deformation control zone is determined by dividing up the sub-patch delineated by the four control points (\( c_1, c_2, c_3, \) and \( c_4 \)) into four parts. One control zone is the shaded area (\( c_1, p_2, p_3, \) and \( p_4 \)) in Fig. 9, where \( c_1 \) is the nearest control point to the target deformation point \( T_p \).

The initial velocity of \( c_1 \), which is the control-point nearest target deformation point, is given, and the velocities of the other three control points are determined in proportion to the distance between the target deformation point and bisectors of each control point and \( c_1 \). In order to perform deformations in real time, the size of the FE problem should be kept as small as possible, by involving only the part of the surface that is to be deformed. In general, the more the control points of deformation area, the bigger the mass and stiffness matrices, In this paper, a \( 7 \times 7 \) control point grid generates a \( 50 \times 50 \) size of the linearized A matrix, which can be calculated in real-time deformation with a Pentium 4 (3.2 GHz). If the size of control grid net increases to \( n \times n \), the size of the A matrix is also increased to \( 4(n - 2)^4 \).

Figure 10 shows the relation ship between the size of control points and the size of the finite elements to calculate. We see that the shape of the global deformation depends on the initial velocity, and higher velocities lead to a shaper deformation, as shown in Fig. 11.

4.2 Comparison of Direct and FE-Based Deformation

Moving a single point on a NURBS surface with a cubic basis function causes the displacement of at least 16 control
points. This reduces the attractiveness of surface deformation as a shape design tool. Direct manipulation of free-form deformations (DMFFD) has similar problems, associated with the choice of deformation area. DMFFD performs well on free-form surfaces with a relatively large number of control points, but distortions occur in surfaces with a few control points, such as that shown in Fig. 7 (b). This is why we developed the modified FE approach. Figure 12 compares DMFFD and the modified FE deformations to a single target point. The sample’s deformation target height is 5. The target deformation, as shown in Fig. 12 (a), was created by moving relevant control points manually to make the height of 5. The DMFFD algorithm suggested in this paper allows the range of the deformation to be adjusted. The result from DMFFD, as shown in Fig. 12 (b), contains distortion in the area of the surface that should remain undeformed. Figure 12 (c) shows the FE technique achieves a closer approach to the target deformation. Figure 13 shows the curvature difference [32] between (a) and (b) and (a) and (c) in Fig. 12. This fairness measure confirms that the FE method achieves a better result than DMFFD. Figure 14 shows deformations to three target points of the same height. Successive deformations are made to the three points in order. Here we are looking to see how far the second deformation affects the accuracy of the first deformation. DMFFD changes the initial deformation by a similar amount to the extent of the second deformation. Thus, in Fig. 14 (c), the additional displacement of point 2 makes it higher than point 3. In contrast, the modified FE technique does not affect the position of point 2 during the second deformation, as shown in Fig. 14 (e). In addition, the DMFFD approach produces a distorted area at the boundary of the deformation.

Figure 15 shows a deformation with a T-shaped target curve. The top of the ‘T’ is created first, and then the vertical stroke. Repeated trial and error is needed to achieve this deformation using DMFFD because unexpected shapes occur where the two parts of the ‘T’ meet. Also, the final shape is slightly different from the desired deformation, due to the inaccuracy of the pseudo-inverse matrix. The modified FE technique only requires one pass and the result is a much better approximation to the desired shape.

As for an overlapping part in deformation by modified FEM, the deformation was sometimes performed by setting a velocity of existing control points in the counter direction of deformation to satisfy the target shape, which is an advantage of control point velocity control in the gradual deformation. The graphs in Fig. 16 compare the heights of the ridges that form the two parts of the T-shaped deformation. Figure 17 shows a plateau-shaped deformation. But with the DMFFD technique the top of the plateau is not flat. A better result could be achieved with a surface with many control points, but the plateau shape is hard to be created with few control points and a deformation area that occupies a single cell of the control grid. In this case, again, the modified FE technique achieves a much closer approximation to the desired shape.
4.3 Applications of the Modified FE Method

The deformation techniques that we have described are not only suitable for interaction using a menu-driven interface, but also in virtual environments in which the user uses a wand to draw or correct the shape of an object. Figure 18 shows a shape produced in a projection environment. Figure 19 shows a face shape produced using the modified FE method. Figure 20 shows a more complicated gargoyle face, also produced using the modified FE method. The modified FE deformation technique can easily be extended to multiple patches to create a more complicated shape. Figure 21 shows the deformation of a car hood constructed from two connected patches.

5. Conclusions

We have proposed two NURBS surface deformation algorithms that use target curves to provide a more intuitive interface for 3D modeling within an immersive VR sketching system. Our first algorithm deforms NURBS surfaces through manipulating the pseudo-inverse matrix of the NURBS basis function. A deformed surface can be deformed again, so that it can be made to include a target curve made up of several points. However, this technique suffers from the ill-conditioning of the pseudo-inverse matrix. We therefore went on to propose a method of NURBS surface deformation which uses a modified FE approach, in which the B-spline control points and basis functions replace the nodes and shape functions of the conventional FE method. Our technique includes a method of controlling the velocity of the control points which avoids the shortcomings of direct deformation. Our technique produces a gradual deformation by numerical integration, and enables a wide range of shapes to be generated without the distorted boundary or trial and error.
Immersive VR sketching is expected to overcome the problems of 2D interfaces which force users to understand a rigid mathematical structure and to use a complicated toolset. VR sketching can be implemented as a small set of tools, making it accessible to both experts and beginners, but it is not sophisticated and effective in complicated projects. The drawing technique of the designer is critical in the construction of an effective and original shape model. Our technique permits designer to extend classical drawing techniques within an immersive VR environment.

In future work, we believe that deformation much closer to the intention of a user can be realized through a more efficient interpolation method, and expressed in a different form by changing the knot vector or weights of a NURBS curve. We also think that our modified FE technique could be extended to support naturalistic dynamic deformations by including real material properties using the different control strategies of the velocities of the control points in its formulation.

Acknowledgments

This work is funded by KOSEF (Korea Science and Engineering Foundation). (No.R01-2007-000-20283-0)

References

[1] D.H. Kim and M.J. Kim, “A new modeling interface for the pen-input displays,” Comput. Aided Des., vol.38, no.4, pp.210–223, 2006.
[2] P. Company, A. Piquer, M. Contero, and F. Naya, “A survey on geometrical reconstruction as a core technology to sketch-based modeling. Special issue on sketch-based interfaces and modeling,” Comput. Graph., vol.29, no.6, pp.892–904, 2005.
[3] R. Juchmes, P. Leclercq, and S. Azar, “A freehand-sketch environment for architectural design supported by a multi-agent system. Special issue on sketch-based interfaces and modeling,” Comput. Graph., vol.29, no.6, pp.905–915, 2005.
[4] L. Eggli, C.Y. Hsu, B.D. Brudelin, and G. Elbert, “Inferring 3D models from freehand sketches and constraints,” Comput. Aided Des., vol.29, no.2, pp.101–112, 1997.
[5] O. Bimber, L.M. Encarnacao, and A. Stock, “Calligraphic interfaces: A multi-layered architecture for sketch-based interaction within virtual environments,” Comput. Graph., vol.24, pp.851–867, 2000.
[6] G. Wesche and H. Seidel, “FreeDrawer - A free-form sketching system on the responsive workbench,” Proc. VRST, 2001, pp.167–174, Banff, Alberta, Canada, 2001.
[7] S. Schkolne, M. Pruet, and P. Schroder, “Surface drawing: Creating organic 3D shapes with the hand and tangible tools,” Proc. SIGCHI 2001, pp.261–268, 2001.
[8] M. Fiorentino, R. Amicis, G. Monno, and A. Stork, “Spacesdesign: A mixed reality workspace for aesthetic industrial design,” Proc. Mixed and Augmented Reality 2002, pp.86–96, 2002.
[9] F.W.B. Li, R.W.H. Lau, and E.F.C. Ng, “VSculpt: A distributed virtual sculpting environment for collaborative design,” IEEE Trans. Multimedia, vol.5, no.4, pp.570–580, 2003.
[10] E. Sachs, A. Roberts, and D. Stoops, “3-Draw: A tool for designing 3D shapes,” IEEE Comput. Graph. Appl., vol.11, no.6, pp.18–26, 1991.
[11] M.F. Deering, “HoloSketch: A virtual reality sketching/animation tool,” ACM Trans. Computer-Human Interaction, vol.2, no.3, pp.220–238, 1995.
[12] T.S. Sederberg and S.R. Parry, “Free-form deformation of solid geometric models,” Proc. SIGGRAPH ’86, Comput. Graph., vol.20, no.4, pp.151–160, 1986.
[13] S. Coquillart, “Extended free-form deformation: A sculpturing tool for 3D geometric modeling,” Proc. SIGGRAPH ’90, Comput. Graph., pp.187–196, 1990.
[14] R. MacCracken and K.I. Joy, “Free-form deformations with lattices of arbitrary topology,” Proc. SIGGRAPH ’92, Comput. Graph., pp.181–188, 1996.
[15] W.M. Hsu, J.F. Hughes, and H. Kaufman, “Direct manipulation of free-form deformations,” Proc. SIGGRAPH ’92, Comput. Graph., vol.26, no.2, pp.177–184, 1992.
[16] P. Borrel, “Simple constrained deformation for geometric modeling and interactive design,” ACM Transactions on Graphics, vol.13, no.2, pp.137–155, 1994.
[17] J.M. Zheng, K.W. Chan, and I. Gibson, “Surface feature constraint deformation for free-form surface and interactive design,” Proc. 5th ACM Symposium on Solid Modeling and Applications 1999, pp.223–233, 1999.
[18] J. Hua and H. Qin, “Free-form deformation via sketching and manipulating scalar fields,” Proc. 8th ACM Symposium on Solid Modeling and Application 2003, pp.328–333, 2003.
[19] S. Zelinka and M. Garland, “Mesh modeling with curve analogies,” Proc. 12th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2004), IEEE Computer Society Press 2004, pp.94–98, 2004.
[20] L. Piegl and W. Tiller, The NURBS Book, Second ed., Springer, 1997.
[21] A. Liverani and G. Piraccini, “NURBS surface shaping in a virtual reality environment,” Proc. 23rd IASTED International Conference on Modeling, Identification and Control, pp.473–478, 2004.
[22] N. Frisch and T. Ertl, “Deformation of finite element meshes using directly manipulated free-form deformation,” Proc. ACM Symposium on Solid Modeling and Applications, pp.249–256, 2002.
[23] J.C. Edward, Interactive synthetic environments with force feedback, Ph.D. thesis, Iowa State University, Ames, Iowa, 1998.
[24] D. Terzopoulos and H. Qin, “Dynamic NURBS with geometric constraints for interactive sculpting,” ACM Transactions on Graphics vol.13, no.2, pp.103–136, 1994.
[25] X. Zhou and J. Lu, “NURBS-based Galerkin method and application to skeletal muscle modeling,” Proc. 2005 ACM Symposium on Solid and Physical Modeling 2005, pp.71–78, 2005.
[26] K. Holling, Finite Element Methods with B-splines, Philadelphia, SIAM, 2003.
[27] W. Pöschl, “B-spline finite elements and their efficiency in solving relativistic mean field equations,” Computer Physics Communications, vol.112, no.1, pp.42–66, 1998.
[28] V. Cheutet, C.E. Catalano, J.P. Pernot, F. Balcédieno, F. Giannini, and J.C. Leon, “3D sketching for aesthetic design using fully free-form deformation features. Special issue on sketch-based interfaces and modeling,” Comput. Graph. 2005, vol.29, no.6, pp.916–930, 2005.
[29] P. Michalik, D.H. Kim, and B.D. Bruderlin, “Sketch- and constraint-based design of B-spline surfaces,” Proc. 7th ACM Symposium on Solid Modeling and Applications 1999, pp.297–304, 2002.
[30] B. Brickman, A First Course in the Finite Element Method, 1990.
[31] E.W. Swokowski, Calculus with Analytic Geometry (Second Alternate Edition). PWS Kent, 1998.
[32] M. Meyer, M. Desbrun, P. Schroder, and A. Barr, “Discrete differential-geometry operator for triangulated 2-manifolds,” Visualization and Mathematics III 2003, pp.35–57, 2003.
Junghoon Kwon is a senior researcher in the mechanical engineering department of Seoul National University, Seoul, Korea. His research interests include geometry deformation, human body modeling and virtual reality applications. Kwon has a Ph.D. in image engineering from Chung-Ang University.

Jeongin Lee is a researcher in the M&SOFT, Inc., Seoul, Korea. His research interests include geometry deformation, GPS system and virtual reality applications. Lee has a MS in image engineering from Chung-Ang University.

Harksu Kim is a Ph.D. candidate in the graduate school of advanced imaging science of Chung-Ang University, Seoul, Korea. His research interests include object tracking, multiple view geometry and virtual sketch applications. Kim has a MS in computer engineering from Keon-Yang University.

Gilsoo Jang is a professor in the electrical engineering department of Korea University, Seoul, Korea. His research interests include power system modeling, distributed power system and virtual reality applications. Jang has a Ph.D. in electrical engineering from Iowa State University.

Youngho Chai is a professor in the graduate school of advanced imaging science of Chung-Ang University, Seoul, Korea. His research interests include spatial sketching, virtual collaborative design and computational geometries. Chai has a Ph.D. in mechanical engineering from Iowa State University.