Vol. 8 No. 4 PROGRESS IN NATURAL SCIENCE August 1998

Shape modifying and controlling with sensor glove in virtual reality *

MA Lizhuang (马利庄)[①], LAU Rynson[②], FENG Jieqing (冯结青)[①] and C. M. Ng[②]

(① State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310027, China; ② The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China)

Received April 16, 1997; revised January 5, 1998

Abstract A bicubic B-spline surface, so-called hand surface, is constructed, which interpolates or approximates the key data of a sensor glove or user's hand and is used to control and deform the shapes of surfaces and solids in a virtual reality (VR) environment. Because a corresponding mapping, a region filter function, an α blend and an incremental method are adopted, the shape modification becomes very easy and intuitive as if the user was using his hand to modify and control objects. Experimental results show the potential of the proposed method

Keywords: shape modification, sensor gloves, deformation, computer aided design.

The interactive design and modifications of shapes have been studied in both CAD/CAM and Computer Graphics. After the surface or object has been defined, subsequent modifications are likely necessary. One common way to modify the free-form surface's shape is to modify their control points one by one[1—3]. However, it is tedious to modify a complicate surface or object which is composed of a number of patches with many control points. Thus interactive tools to control a set of control points or sampled points are preferred, e.g. some deformation methods and linear transformation methods[4—6]. In virtual reality, human interacts and immerses with the objects in the environment. The most natural way of interactive tools for human being is the hand gesture When the shape of the hand including the fingers changes, one will hope the surfaces or objects to be controlled change their shapes accordingly (see fig. 1 and Plate I). Sensor gloves are VR tools that were designed to simulate the motion of the user's hand. Some of the commercial models are VPL DataGlove[7], Vertex CyberGlove[8], Mattel PowerGlove and Exos Dextrous Hand Master[9]. They all have sensors that measure some or all of the finger joint angles. They have three-dimensional sensors as well in order to track the user's wrist motion. These sensor gloves are powerful to interact on the object and input data by simulating user's hand motion and have been used successfully in current VR systems. Our idea is to combine both the advantages of sensor gloves and deformation tools so that the shape of virtual objects can be controlled and modified intuitively. A sensor glove is thus considered as a control surface with which surfaces and solids are modified accordingly. In this paper, a bicubic B-spline surface, so-called hand surface, is constructed, which interpolates or approximates the key data of a sensor glove or user's hand, e.g. finger joints and palm center. This surface is used to control and deform the shapes of surfaces and solids in the VR environment. By setting up a corresponding mapping between the surfaces or

* Project supported by the National Excellent Young Science Foundation of China (Grant No 69625304), Huo Ying-dong Foundation for Young Teachers and Zhao Guoang-biao Foundation for High Technology Developments.

solids to be controlled and the hand surface, the deformation can be handled in an incremental manner. The related surfaces or solids change accordingly while the sensor glove is changing its shape. For local modification, a region filter function $W(u, v)$ is introduced which imposes locality on the mapping/deformation. By this filter, local or global modification, which does not change the smoothness of the modified shapes, can be realized easily. In order to eliminate the illness state of the deformed shapes, an α blend is used between the initial and active normals of the hand surface. When α decreases from 1 to 0, the corresponding shape modification becomes more and more sensitive to the active normal of the hand surface. The shape modification is easy and intuitive in this way. Experimental results show the potential of the proposed method.

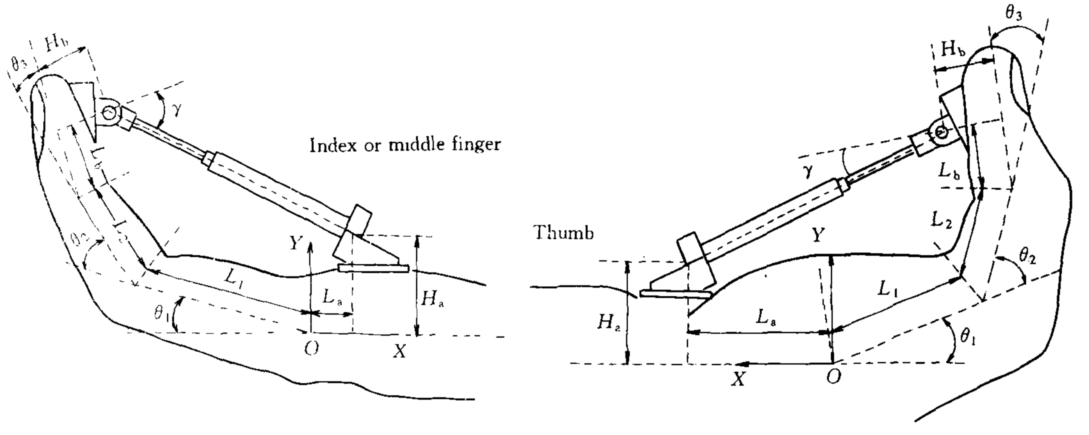

Fig. 1. Finger parameters of geometric shapes.

1 Input data with sensor gloves

The human fingers have flexion-extension and lateral abduction-adduction as illustrated in fig. 1. The process of transforming raw sensor data into finger joint angles is called "glove calibration". The calibration method differs between flexion sensors, and abduction-adduction ones. The sensor calibration method uses a least-square formula that was given by Tan and Hong[10].

$$\theta(r) = a + br + c\ln(r), \tag{1}$$

where r is raw sensor reading, $\theta(r)$ is flex angle of the finger, a, b, and c are calibration constants. DataGloves in their standard configuration do not measure the flexion angle θ_3 of the distal joints (except for the thumb). One way to determine θ_3 is to take into account the coupling that exists between θ_2 and θ_3 over the range of grasping motions[11].

A sensor glove has usually more than 22 sensors, e.g. CyberGlove[12] and DataGlove[7]. For convenience, the five fingers, thumb, index, middle, ring and pinkie are numbered fingers 1, 2, 3, 4 and 5, respectively. Let the joint points of the finger i be $Q_{i,1}$, $Q_{i,2}$, $Q_{i,3}$, $Q_{i,4}$ and $Q_{i,5}$, $i=1,2,3,4,5$. When the middle finger parameters L_a, L_1, L_2, L_b are given which depend on the user's hand characteristics, the joint points $Q_{3,1}$, $Q_{3,2}$, $Q_{3,3}$, $Q_{3,4}$ and $Q_{3,5}$ can be calculated according to the measured angles θ_1 and θ_2, and the local coordinate of the palm. The joint points of other fingers are determined similarly. According to the movements of figners and hand, the shape of curves and surfaces can be controlled in an intuitive way.

2 Construct the hand surface

To construct the hand surface which interpolates the measure joint points $Q_{i,j}$, $i, j = 1,2,3,4,5$, the bicubic B-spline surface form is used. The knot vectors, $U = \{u_0, u_1, \cdots, u_9, u_{10}\}$ and $V = \{v_0, v_1, \cdots, v_8, v_9, v_{10}\}$ in U and V directions respectively, are calculated with chord length. For convenience, the knot vectors are normalized such that they have Bezier type end knots. The domain of parameter (u, v) is thus a unit square region. Let the hand surface $H(u, v)$ be a bicubic B-spline surface

$$\boldsymbol{H}(u, v) = \sum_{i=0}^{6}\sum_{j=0}^{6} \boldsymbol{P}_{i,j} N_{i,3}(u) N_{j,3}(v), \tag{2}$$

where $N_{i,3}(u)$, $N_{j,3}(v)$ are normalized B-spline basis functions. Then from the interpolating conditions, we have $\boldsymbol{H}(u_i, v_j) = \boldsymbol{Q}_{i+1,j+1}$. Extra end conditions must be provided so that the solution is unique. According to the end condition, four rows of virtual data points $Q_{i,j}$, $i, j = 0,6$, should be computed such that the hand surface has the given end condition[13,14]. The following equations can be obtained:

$$\boldsymbol{Q} = \boldsymbol{APB}, \tag{3}$$

where $\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{A}$ and $\boldsymbol{B}$ are 7×7 matrices,

$$\boldsymbol{P} = (\boldsymbol{P}_{i,j}),\ \boldsymbol{Q} = (Q_{i,j}),\ \boldsymbol{A} = (N_{i,3}(u_j)),\ \boldsymbol{B} = (N_{j,3}(v_i)). \tag{4}$$

The solution of surface interpolating given data points and specific end conditions is obtained in a two-step process[13,14] (see figure 2).

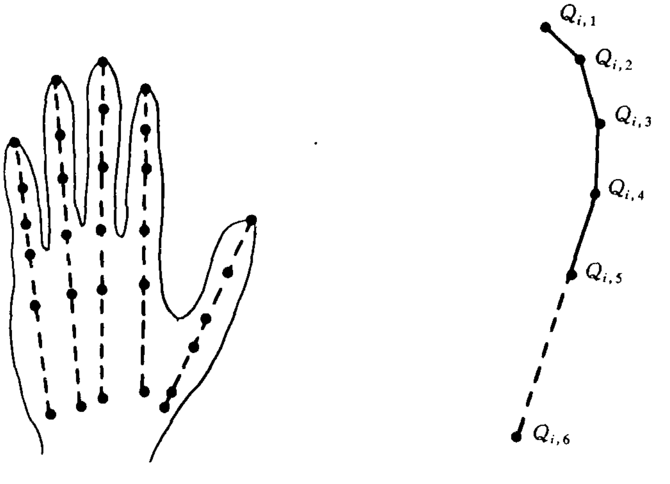

Fig. 2. The finger joints of the sensor glove and the control points of the hand surface.

In VR environment, the speed to obtain the solution is often more important than the quality. For fast computation, the hand surface is regarded as Bezier type B-spline surface and its control points are simply set equal to the data points. The knot vectors in U and V direction are also regarded as the chord length. However, one more row of data points are added in U direction so that the resulting hand surface approximates well the shape of sensor glove. The added data points are obtained by extending the last segments with a ratio β_i which is chosen according to the characteristic of user's hand (see figs. 1 and 2). Namely,

$$Q_{i,6} = Q_{i,5} + \beta_i(Q_{i,6} - Q_{i,5}).$$

Thus the hand surface can be obtained rapidly without much computation.

$$\begin{cases} H(u,v) = \sum_{i=0}^{4}\sum_{j=0}^{5} P_{i,j}N_{i,3}(u)N_{j,3}(v), \\ P_{i,j} = Q_{i+1,j+1},\ i = 0, \cdots, 4,\ j = 0, \cdots, 5. \end{cases} \tag{5}$$

3 Shape modification with hand surface

3.1 Setting up the corresponding mapping

Shape modification or deformation can be thought of as a self mapping onto itself. It is necessary to set up a well-defined corresponding mapping. Different from the free-form deformation (FFD) method proposed by Sederberg and Parry[6], a planar region called base surface, RH_0, is defined rather than a volume in the 3-D space. Objects to be deformed are embedded in the extended 3-D parametric space of RH_0. Note that RH_0 is the codomain of the hand surface in flat state whose control points are all in a plane π_H. The local coordinate of the hand surface is defined according to π_H and the 'palm center'. Let $P = (x_P, y_P, z_P)$ be a control point or a sampled point of the surfaces or objects. When P is projected onto π_H, its projection $P' = (x'_P, y'_P)$ with respect to π_H can be obtained (fig. 3). If the projection P' of P is within RH_0, then its parameter coordinate (u_P, u_P) lying on the base surface RH_0 can be calculated easily by the approximation method. An acceleration method can be designed similar to the computation of normal vectors (see sec. 4). When the projection P' of P is outside RH_0, P will not be modified.

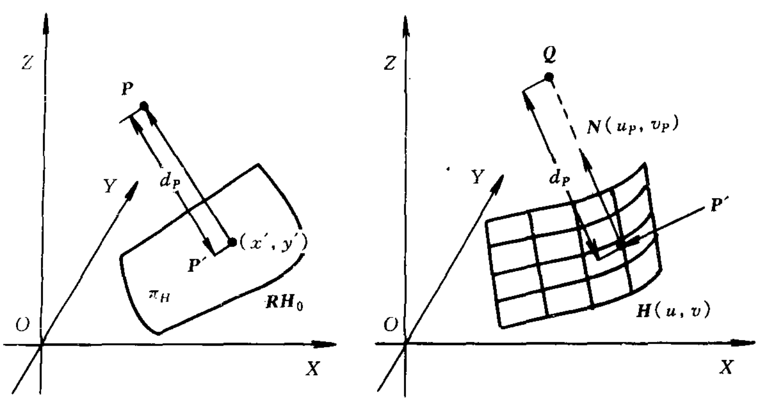

Fig. 3. The correspondence between P, P' and Q.

When the sensor glove changes its shape, the hand surface $H(u, v)$ changes accordingly. Then all surfaces or objects in the virtual space would be changed in a consistent way as if there were a force field formed by the hand surface. P is then transformed to a new position Q by the following mapping:

$$Q = H(u_P, v_P) + d_P N(u_P, v_P), \tag{6}$$

where $\boldsymbol{N}(u_P, v_P)$ is the unit normal vector of $\boldsymbol{H}(u, v)$ at (u_P, v_P) and d_P is the directional distance from $\boldsymbol{P}$ to the base surface $\boldsymbol{RH}_0$. When $\boldsymbol{P}$ is on the positive side of the hand surface, namely $(\boldsymbol{P}' - \boldsymbol{P}) \cdot \boldsymbol{N}_H^0 > 0$, $d_P > 0$, otherwise $d_P \leqslant 0$.

3.2 α blend used in the mapping

With mapping (6), virtual surfaces or objects are deformed by sensor glove. Notice that this mapping is very sensitive to the normal variation of the hand surface, especially, when the sensor glove is far away from the surfaces or objects to be modified. In some extreme cases, the resulting shapes are unsatisfactory, e.g. the modified objects may be self intersected(see Plate I). A modified mapping is introduced to solve this problem as follows. When the user initially extend the sensor glove in a flat state, the base surface $\boldsymbol{RH}_0$ and the local coordinate of the hand surface are set up. It is natural that the user pays more attention in $\boldsymbol{N}_H^0$ direction and modifies objects with respect to this direction. Thus an α blending is selected between the initial normal $\boldsymbol{N}_H^0$ and the normal of the hand surface at parameter (u_P, v_P). Mapping (6) is modified as follows:

$$\boldsymbol{Q} = \boldsymbol{H}(u_P, v_P) + d_P[(1 - \alpha)\boldsymbol{N}(u_P, v_P) + \alpha\boldsymbol{N}_H^0]_I. \quad (7)$$

If unit normal $\boldsymbol{N}(u_P, v_P)$ is occasionally in the opposite direction of $\boldsymbol{N}_H^0$ and their length has a ratio $\alpha : (1-\alpha)$, then vector $\boldsymbol{V} = (1-\alpha)\boldsymbol{N}(u_P, v_P) + \alpha\boldsymbol{N}_H^0 = 0$. $[\boldsymbol{V}]_I$, the unit normal vector of $\boldsymbol{V}$, is set equal to $\boldsymbol{N}_H^0$ in default and a warning signal is shown at this point. Other ways are to increase α a little or use the normal of a point neighboring $\boldsymbol{P}$. By selecting different values of blending parameter α, the user can attain different goals. When $\alpha = 0$, mapping (7) degenerates to mapping (6). When $\alpha = 1$, the mapping is similar to an FFD in which the deformation is a linear extrusion of the hand surface along $\boldsymbol{N}_H^0$. When α decreases from 1 to 0, the corresponding mapping becomes more and more sensitive to the active normal vector of the hand surface. The shape change of deformed objects also becomes less and less coherent to the shape of hand surface.

In the following sections, the proposed method is demonstrated with free-form surfaces. However, it is applicable to general objects. Let hand surface $\boldsymbol{H}(u, v)$ control free-form surface $\boldsymbol{R}(u, v)$. There are mainly two types of modification, local and global ones.

3.3 Global shape modification

All sample points or control points of the surface are modified. By scaling the projection region $\boldsymbol{RH}_0$ if necessary, all the projection points will lie in the interior of $\boldsymbol{RH}_0$. Namely, a virtual hand surface is constructed as if the users' hand were amplified.

3.4 Local shape modification

Part of a surface is modified. There are also two ways to attain this goal: (i) localizing the surface $\boldsymbol{R}(u, v)$ to be modified and, (ii) localizing the mapping or deformation (7). The first method is dependent on the representation of the surface to be modified while the second one is not.

3.4.1 Localizing the surface itself. An active region of the surface is specified in its parameter space or in the 3-D space. If a parameter region, $\Omega = [u_1, u_2; v_1, v_2]$, is specified, the surface $\mathbf{R}(u, v)$ is subdivided into sub-surfaces along parameter values, $u = u_1, u_2$; $v = v_1, v_2$. This process can be also achieved by knot insertion algorithm[13,15]. After the surface is localized, the sub-surface in the active region Ω is modified by the sensor glove.

3.4.2 Localizing the mapping with a smooth region filter function. The mapping (7) can be localized accordingly by introducing a region filter. The mapping then acts only on the selected region R_s and a fillet region $R_f \supset R_s$. In one-dimensional case, the region filter function can be represented by Bernstein-Bezier basis as follows (see figure 4).

$$F(u) = \begin{cases} 0, & u \notin [e, f], \\ 1, & u \in [a, b], \\ B_{2,3}\left(\frac{u-e}{a-e}\right) + B_{3,3}\left(\frac{u-e}{a-e}\right), & u \in [e, a], \\ B_{0,3}\left(\frac{u-b}{f-b}\right) + B_{1,3}\left(\frac{u-b}{f-b}\right), & u \in [b, f]. \end{cases}$$

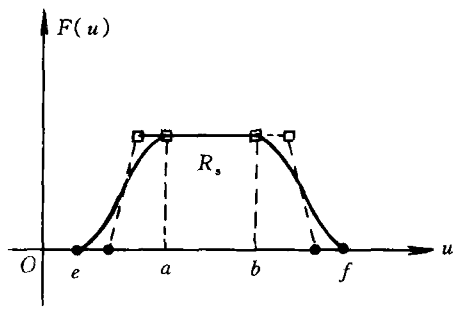

Fig. 4. The smooth region filter function $F(u)$ and its Bezier representation in $[a, e]$. ● denotes the Bezier coordinate at which $F(u) = 0$. □ denotes the Bezier coordinate at which $F(u) = 1$.

In fact, $F(u)$ is represented as a cubic Bezier curve by introducing related control points over $[a, e]$ (see figure 4).

In two-dimensional case, the region filter $W(u, v)$

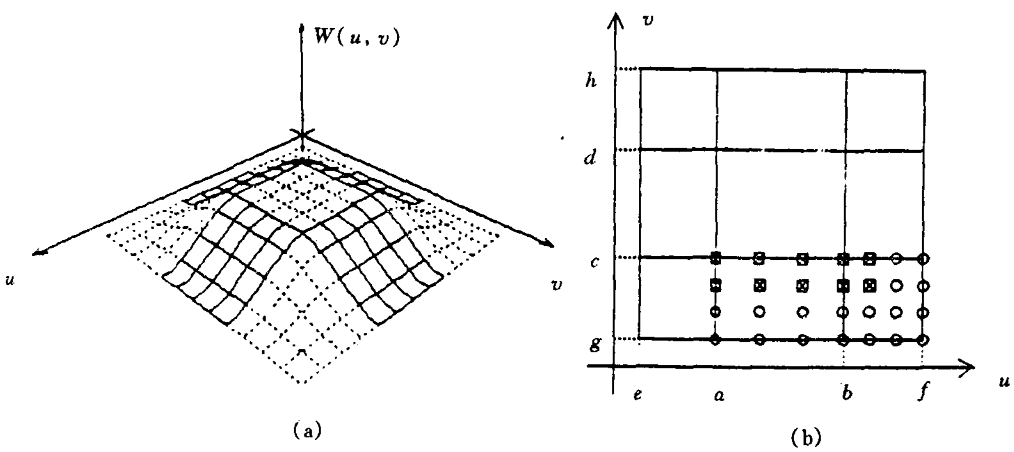

Fig. 5 The region filter function $W(u, v)$ and its Bezier control points. ○ denotes the Bezier coordinate of control point at which $W(u, v) = 0$. ⊠ denotes the Bezier coordinate of control point at which $W(u, v) = 1$.

shown in fig. 5 is a generalized tensor product function of $F(u)$ and $F^*(v)$, which can be represented as

$$W(u,v)=\begin{cases}0, & u\notin[e,f;\ g,h],\\ 1, & u\in[a,b;\ c,d],\\ B_{2,3}\left(\frac{u-e}{a-e}\right)+B_{3,3}\left(\frac{u-e}{a-e}\right), & (u,v)\in[e,a;\ c,d],\\ B_{0,3}\left(\frac{u-b}{f-b}\right)+B_{1,3}\left(\frac{u-b}{f-b}\right), & (u,v)\in[b,f;\ c,d],\\ B_{2,3}\left(\frac{v-g}{c-g}\right)+B_{3,3}\left(\frac{v-g}{c-g}\right), & (u,v)\in[a,b;\ g,c],\\ B_{0,3}\left(\frac{v-d}{h-d}\right)+B_{1,3}\left(\frac{v-d}{h-d}\right), & (u,v)\in[a,b;\ d,h],\\ \sum_{i=2}^{3}\sum_{j=2}^{3}\left[B_{i,3}\left(\frac{u-e}{a-e}\right)+B_{j,3}\left(\frac{v-g}{c-g}\right)\right], & (u,v)\in[e,a;\ g,c],\\ \sum_{i=0}^{1}\sum_{j=2}^{3}\left[B_{i,3}\left(\frac{u-b}{f-b}\right)+B_{j,3}\left(\frac{v-g}{c-g}\right)\right], & (u,v)\in[b,f;\ g,c],\\ \sum_{i=2}^{3}\sum_{j=0}^{1}\left[B_{i,3}\left(\frac{u-e}{a-e}\right)+B_{j,3}\left(\frac{v-d}{h-d}\right)\right], & (u,v)\in[e,a;\ d,h],\\ \sum_{i=0}^{1}\sum_{j=0}^{1}\left[B_{i,3}\left(\frac{u-b}{f-b}\right)+B_{j,3}\left(\frac{v-d}{h-d}\right)\right], & (u,v)\in[b,f;\ d,h].\end{cases} \tag{8}$$

Now the local mapping from sample $\boldsymbol{P}$ to the modified point $\boldsymbol{Q}^L(\boldsymbol{P})$ can be described as

$$\boldsymbol{Q}^L = W(u_P,\ v_P)\boldsymbol{Q} + (1 - W(u_P,\ v_P))\boldsymbol{P}, \tag{9}$$

where $\boldsymbol{Q}$ is obtained from mapping (7). Of course, parameter value $(u_P,\ v_P)$ is calculated at first, $\boldsymbol{Q}^L$ is simply equal to $\boldsymbol{P}$ if the parameter is outside the fillet region.

4 Accelerating the computation

In this section, some techniques are presented for accelerating the proposed algorithm.

4.1 Fast computation of normal vectors

In the applications, the hand surface $\boldsymbol{H}(u,v)$ is subdivided adaptively according to its local curvature and then approximated by planar triangles. The normal vector of hand surface at parameter $(u_P,\ v_P)$ can then be approximated by weighted average of normals at nearby vertices of a corresponding triangle. Assume that $\boldsymbol{H}(u_P,\ v_P)$ is within triangle $\boldsymbol{T}_1\boldsymbol{T}_2\boldsymbol{T}_3$, and $\boldsymbol{T}_i=\boldsymbol{H}(u_i,\ v_i)$, $i=1,2,3$. Then the normal vector $\boldsymbol{N}_P$ at $(u_P,\ v_P)$ is set as follows (see figure 6).

$$N_p = [w_1 N_1 + w_2 N_2 + w_3 N_3]_I, \tag{10}$$

where N_i is the normal vector of $H(u, v)$ at point $T_i (i = 1, 2, 3)$ and (w_1, w_2, w_3) is the barycentric coordinate of (u_P, v_P) with respect to triangle $P_1 P_2 P_3$ where $P_i = (u_i, v_i)$, $i = 1, 2, 3$.

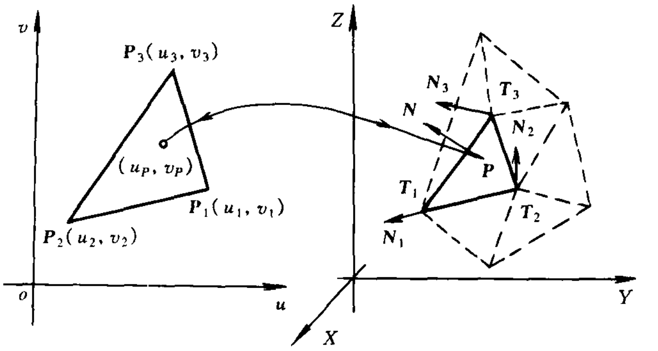

Fig. 6. The approximation of normal vector at (u_P, v_P).

4.2 Incremental method for computing the interpolated hand surface

There are two main steps in the proposed method, namely the construction of the hand surface $H(u, v)$ and the modification of the underlying surfaces or objects. To accelerate the computation, incremental method can be employed in both steps. Since finger parameters are constants in the process, matrices A and B are constants in eq. (2) too. Their inverse matrices can thus be pre-computed. Therefore, when only a few data points, say, $P_{i,j}$ are changed with a quantity $d_{i,j}$, the updated hand surface can be calculated very fast. Let D be a matrix whose elements are all zero except the element $d_{i,j}$ at (i, j) location. Then the coefficient matrix P, namely the control point set, will change as follows:

$$P' = P + A^{-1} D B^{-1}. \tag{11}$$

Since D has only one nonzero element, the above equation has only $n + n^2$ multiplications of scalar with vector which are much less then $n^3 + n^2$ ones in general cases, here $n = 7$ is the matrix order

4.3 Getting deformed point P with incremental method

If the shape of hand surface just changes a little, e.g. one control point or one finger joint moves, then the deformed point will be changed accordingly from $Q(P)$ to $Q'(P)$. Q' can be evaluated by a simple approximation formula. This technique is very useful when the control points of the hand surface are equal to the data points obtained from the sensor glove as in eq. (5). Since there are often a great number of sample points of objects in VR, the following technique apparently accelerates the related algorithm. Suppose that control point $P_{i0,j0}$ is changed with a small quantity of $d_{i0,j0}$. Let H_u, H_v, H'_u, H'_v denote the partial derivatives of the hand surface $H(u, v)$ and the updated one $H'(u, v)$ respectively at parameter (u_P, v_P). Let

$$G(u, v) \equiv N_{i0,3}(u) N_{j0,3}(v).$$

G_u, G_v are partial derivatives of G. Then the following results can be derived:

$$\begin{cases} \boldsymbol{H}'(u_p, v_P) = \boldsymbol{H}(u_p, v_p) + \boldsymbol{d}_{i0,j0}G(u_p, v_p), \\ \boldsymbol{H}'_u \times \boldsymbol{H}'_v = \boldsymbol{H}_u \times \boldsymbol{H}_v + (\boldsymbol{H}_u G_v + G_u \boldsymbol{H}_v) \times \boldsymbol{d}_{i0,j0}, \\ \boldsymbol{Q}' \approx \boldsymbol{H} + \boldsymbol{d}_{i0,j0}G(u_p, v_p) + d_p[(1-\alpha)\boldsymbol{H}'_u \times \boldsymbol{H}'_v + \alpha \boldsymbol{N}^0_H]_I. \end{cases} \tag{12}$$

When the hand surface is approximated with triangles by subdivision method, a look-up table can be set up. The look-up table stores the pre-computed quantities (u_i, v_i), $\boldsymbol{H}(u_i, v_i)$, $\boldsymbol{H}_u(u_i, v_i)$, $\boldsymbol{H}_v(u_i, v_i)$, $\boldsymbol{N}(u_i, v_i)$, $(i=1,2,\cdots, num)$ at vertices of triangles where num denotes the number of vertices. All the above quantities at arbitrary parameter (u_P, v_P) are the weighted averages of the corresponding quantities at three vertices of respective triangle containing (u_P, v_P) (see fig. 6 and eq. (10)). The detailed formulas are omitted here. It is clear that $\boldsymbol{H}'$, $\boldsymbol{H}'_u$, $\boldsymbol{H}'_v$ can be obtained from $\boldsymbol{H}$, $\boldsymbol{H}_u$, $\boldsymbol{H}_v$ with little computation. Thus eq. (12) only involves a little more operation for computing the new deformed sample points or control points when the hand surface is changing slowly.

4.4 Experimental results

The proposed algorithm is encoded in C programming language and implemented at the State Key Laboratory of CAD&CG, Zhejiang University. Plate I shows some of the experimental results. Plate I(a) shows the rendered picture of the original object and the planar hand surface (top) and, the deformed object with the hand surface when the thumb moves up (bottom); Plate I(b) illustrates the deformed object and the hand surface where all fingers move up(top) and the case where two fingers ring and pinkie are in flat state while the others move up (bottom); in Plate I(c), the deformed objects are shown respectively when local modification is used(bottom) or not(top), where all fingers of the hand surface move up as in Plate I(a)(top). In Plate I(a)—(c), $\alpha=0$, the shape of the deformed cup is very sensitive to the active normal of the hand surface and it is self intersected along the rim of the cup. In Plate I(a′)—(c′), $\alpha=0.5$, the shape of the cup becomes less sensitive to the active normal of the hand surface. When $\alpha=0$, the deformed shape of the cup is independent of the active normal of the hand surface. Of course, it still depends on the shape of hand surface. In Plate I(c) and (c′), local deformation is used which employs the region filter.

5 Conclusions

Experimental results demonstrate the potential of the proposed method in the following aspects:

(i) Our new method is more intuitive and convenient compared with the existing ways, e.g. Pigel's method and FFD method.

(ii) The hand surface is used as a control tool rather than a tri-cubic Bernstein Bezier tensor volume. Thus this procedure involves much less computation and runs fast, especially when the acceleration techniques are incorporated.

(iii) Local or global modification can be realized easily without changing smoothness. It is independent of the representation of the surfaces of objects.

Acknowledgement We are very grateful to the staff of the State Key Laboratory of CAD&CG and research group of the Department of Computing, Hong Kong Polytechnic University for their help in preparing the paper.

References

1 Fog, G., B-spline surface system for ship hull design, *Computer Applications in the Automation of Shipyard Operation and Ship Design*, North-Holland, 1985, 359—366.

2 Piegl, L., *Modifying the Shape of Rational B-splines*, Part 1: Curves, CAD, 1989a, 21(8): 509.

3 Piegl, L., *Modifying the Shape of Rational B-splines*, Part 2: Surfaces, CAD, 1989b, 21(9): 538.

4 Barr, A. H., Global and local deformation of solid primitives, SIGGRAPH'84, *ACM Computer Graphics*, 1984, 18(3): 21

5 Feng, J. Q., Ma, L Z., Peng, Q. S., A new free-form deformation through the control of parametric surfaces, *Computer & Graphics*, 1996, 20(4): 531.

6 Sederberg, T. W., Parry, R., Free-form deformation of solid geometric models' SIGGRAPH'86, *ACM Computer Graphics*, 1986, 20(4): 151.

7 VPL Research Inc, *DataGlove Model 2 User's Manual*, Redwood City, CA, 1987.

8 Stone, R., *Virtual Reality Tutorial*, *MICAD Conference*, Paris, France, 1992.

9 Marcus, B., Churchill, P., Human hand sensing for robotics and teleoperators, *Arthur D. Little Technical Report*, November, 1989.

10 Hong, J., Tan, X, Teleoperating the Utah/MIT Hand with a VPL DataGlove I, DataGlove Calibration, in *Proceedings of IEEE*, 1988, 1752—1757.

11 Burdea, G., Zhuang, J. C., Roskos, E. et al., A portable dextrous master with force feedback, *Presence: Teleoperators & Virtual Reality*, 1992, 1(1): 18.

12 Virtex, Co., *Company Brochure*, Standford, CA, October 1, 1992.

13 Farin, G, *Curves and Surfaces in Computer Aided Geometric Design*, 3rd ed., Academic Press, INC, Harcourt Brace Jovanovich, Publishers, 1993.

14 Ma, L Z, Peng, Q. S., Feng, J. Q., Explicit formulas for bicubic B-spline surface interpolation, *CAD/Graphics'95, SPIE*, 1995

15 Hoschek, J., Lasser, D. (translated by Schumaker, L. L.), *Fundamentals of Computer Aided Geometric Design*, A. K. Peters, Wellesley, Massachusetts, 1993.