

Direct manipulation of FFD: efficient explicit solutions and decomposable multiple point constraints

Shi-Min Hu¹, Hui Zhang¹,
Chiew-Lan Tai², Jia-Guang Sun¹

¹ Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, P.R. China
E-mail: shimin@tsinghua.edu.cn

² Department of Computer Science, Hong Kong
University of Science and Technology, Hong Kong

Published online: 25 July 2001
© Springer-Verlag 2001

The ability to directly manipulate an embedded object in the free-form deformation (FFD) method improves controllability. However, the existing solution to this problem involves a pseudo-inverse matrix that requires complicated calculations. This paper solves the problem using a constrained optimization method. We derive the explicit solutions for deforming an object which is to pass through a given target point. For constraints with multiple target points, the proposed solution also involves simple calculations, only requiring solving a system of linear equations. We show that the direct manipulations exhibit the commutative group property, namely commutative, associative, and invertible properties, which further enhance the controllability of FFD. In addition, we show that multiple point constraints can be decomposed into separate manipulations of single point constraints, thus providing the user the freedom of specifying the constraints in any appropriate order.

Key words: Animation – Deformations – Geometric modeling

Corresponding author: S.-M. Hu

1 Introduction

The design of complex models is a key problem in computer-aided geometric design and computer graphics. Many efforts have been made towards developing more convenient techniques for designing and modifying solid or surface models (Barr 1984; Borrel and Bechmann 1991; Chang and Rockwood 1994; Celniker and Gossard 1991; Coquillart and Jancène 1991; Feng et al. 1996; Lazarus et al. 1994; Moccozet and Thalmann 1997; Sederberg and Parry 1986; Singh and Fiume 1998), and Bechmann (1994) presented a survey of these techniques. Barr (1984) introduced a range of geometric transformations that include bending, twisting, and tapering. Sederberg and Parry (1986) proposed the widely used free-form deformation (FFD) method. The idea of the FFD method is simple: by embedding the object to be deformed in a simple, yet flexible, solid, the deformation of the solid is propagated to the embedded object. In the original FFD method (Sederberg and Parry 1986), the embedding solid is a parallelepiped represented by a trivariate Bézier volume. This technique has been extended to include lattices of non-parallelepiped or arbitrary topology (Coquillart 1990; MacCracken and Joy 1996), and trivariate B-splines or NURBS volumes (Griessmair and Purgathofer 1989; Lamousin and Waggenspack 1994). A disadvantage of these FFD methods is in the indirect control of the deformation through adjusting control points or weights of the embedding volume. It is difficult to get the object to pass through desired points precisely. Moreover, a large number of control points in complex models makes it impractical to determine the exact number of control points to be changed and how they must be changed to produce a desired deformation. These deficiencies motivated Hsu et al. (1992) to investigate the direct manipulation of FFD. Given a user selection of source points on an object and the target points, their method automatically computes the necessary repositioning of the control points using a least-square formulation. With this method, designing complex objects using the FFD method becomes more intuitive; users no longer need to understand the tensor-product Bézier or NURBS volume to use the FFD as a modeling tool. Hsu et al.'s proposed solution to the FFD direct manipulation problem, however, uses a pseudo-inverse (generalized inverse) matrix that involves complicated calculations. This paper presents a new way of solving the FFD direct manipulation problem. The approach is based on a constrained optimization

method, which enables the derivation of simple explicit solutions in the case of single point constraints. We will prove that these direct manipulations exhibit the commutative group property, which indicates their high controllability. For multiple point constraints, we will derive solutions that only involve solving a system of linear equations. The decomposability of the multiple point constraints problem into separate manipulations involving single point constraints will also be proved.

2 FFD method using NURBS volume

We assume that the embedding volume is a NURBS parallelepiped. It is represented as a tensor-product NURBS volume $\mathbf{Q}(u, v, w)$ with uniformly distributed control points $\mathbf{P}_{i,j,k}$,

$$\mathbf{Q}(u, v, w) = \sum_{i,j,k=0}^{l,m,n} \mathbf{P}_{i,j,k} R_{i,j,k}(u, v, w), \quad 0 \leq u, v, w \leq 1 \quad (1)$$

where

$$R_{i,j,k}(u, v, w) = \frac{W_{i,j,k} B_{i,p}(u) B_{j,q}(v) B_{k,r}(w)}{\sum_{i,j,k=0}^{l,m,n} W_{i,j,k} B_{i,p}(u) B_{j,q}(v) B_{k,r}(w)} \quad (2)$$

and where $W_{i,j,k}$ are the corresponding weights of $\mathbf{P}_{i,j,k}$, and $B_{i,p}(u)$, $B_{j,q}(v)$, and $B_{k,r}(w)$ are the p th, q th, and r th order B-spline basis functions defined over the knot vectors $\mathbf{U} = \{u_0, u_1, \dots, u_p, \dots, u_l, \dots, u_{l+p}\}$, $\mathbf{V} = \{v_0, v_1, \dots, v_q, \dots, v_m, \dots, v_{m+q}\}$, and $\mathbf{W} = \{w_0, w_1, \dots, w_r, \dots, w_n, \dots, w_{n+r}\}$, respectively.

We denote the object to be deformed by \mathbf{O} , and embed it in a NURBS volume $\mathbf{Q}(u, v, w)$. Adjusting the control points, weights, or knot vectors of the NURBS volume will modify $\mathbf{Q}(u, v, w)$ and propagate the deformation to object \mathbf{O} . The process is as follows: calculate the parameters (u, v, w) of the points on \mathbf{O} with reference to $\mathbf{Q}(u, v, w)$, deform the NURBS volume, and compute the new positions of these points by substituting the computed parameters into the deformed NURBS volume. The uniform distribution of the control points enables the parameters (u, v, w) to be quickly derived through simple linear transformations.

3 Explicit solution for direct manipulation of FFD

Suppose that a source point \mathbf{S} on an object \mathbf{O} is to be moved to a target point \mathbf{T} , and that (u_s, v_s, w_s) are the parameters of \mathbf{S} with respect to the original NURBS volume. We compute the displacement $\delta_{i,j,k}$ of each control point $\mathbf{P}_{i,j,k}$ so that the following constraint is satisfied:

$$\begin{aligned} \mathbf{T} &= \sum_{i,j,k=0}^{l,m,n} (\mathbf{P}_{i,j,k} + \delta_{i,j,k}) R_{i,j,k}(u_s, v_s, w_s) \\ &= \mathbf{S} + \sum_{i,j,k=0}^{l,m,n} \delta_{i,j,k} R_{i,j,k}(u_s, v_s, w_s). \end{aligned} \quad (3)$$

The optimization objective is defined as

$$\text{Min} \sum_{i,j,k=0}^{l,m,n} \|\delta_{i,j,k}\|^2, \quad (4)$$

and therefore the Lagrange function can be written as

$$\begin{aligned} L &= \sum_{i,j,k=0}^{l,m,n} \|\delta_{i,j,k}\|^2 \\ &+ \lambda \left(\mathbf{T} - \mathbf{S} - \sum_{i,j,k=0}^{l,m,n} \delta_{i,j,k} R_{i,j,k}(u_s, v_s, w_s) \right) \end{aligned} \quad (5)$$

where $\lambda = (\lambda_1, \lambda_2, \lambda_3)^T$ is the Lagrange multiplier, and $\|\cdot\|$ is the Euclidean norm. By setting $\frac{\partial L}{\partial \lambda_1} = \frac{\partial L}{\partial \lambda_2} = \frac{\partial L}{\partial \lambda_3} = 0$, $\frac{\partial L}{\partial \delta_{i,j,k}^x} = \frac{\partial L}{\partial \delta_{i,j,k}^y} = \frac{\partial L}{\partial \delta_{i,j,k}^z} = 0$, and writing the derived formulas in vector form, we obtain the following equations:

$$\begin{cases} \mathbf{T} = \mathbf{S} + \sum_{i,j,k=0}^{l,m,n} \delta_{i,j,k} R_{i,j,k}(u_s, v_s, w_s) & (6a) \\ \delta_{i,j,k} = \frac{\lambda}{2} R_{i,j,k}(u_s, v_s, w_s), & (6b) \\ 0 \leq i \leq l, 0 \leq j \leq m, 0 \leq k \leq n. \end{cases}$$

Substituting (6b) into (6a), then equating it to the expression obtained from (6b), we obtain

$$\begin{aligned} \frac{\mathbf{T} - \mathbf{S}}{\sum_{i,j,k=0}^{l,m,n} R_{i,j,k}^2(u_s, v_s, w_s)} &= \frac{\lambda}{2} \\ &= \frac{\delta_{i,j,k}}{R_{i,j,k}(u_s, v_s, w_s)}. \end{aligned}$$

Solving for $\delta_{i,j,k}$ yields the explicit solution for the direct manipulation as follows:

$$\delta_{i,j,k} = \frac{R_{i,j,k}(u_s, v_s, w_s)}{\sum_{i,j,k=0}^{l,m,n} R_{i,j,k}^2(u_s, v_s, w_s)} (\mathbf{T} - \mathbf{S}). \quad (7)$$

That is, by setting each control point to its new position of $\mathbf{P}_{i,j,k} + \delta_{i,j,k}$, we will achieve the desired deformation of moving \mathbf{S} on the object \mathbf{O} to the target point \mathbf{T} .

Note that, due to the locality of the B-spline basis functions (de Boor 1978; Schumaker 1980), not all $\mathbf{P}_{i,j,k}$ are changed in this process. Suppose $u_s \in [u_h, u_{h+1}]$, where u_h and u_{h+1} are two consecutive knots in the knot vector \mathbf{U} , then the non-vanishing basis functions are $B_{i,p}(u_s)$, $h - p + 1 \leq i \leq h$, i.e., p non-vanishing ones. Thus, (7) has only $p \times q \times r$ non-vanishing terms.

Analogously, the above method can be extended to modify weights, instead of control points; for brevity, we omit the details in this paper.

4 Commutative group property of FFD direct manipulation

Suppose \mathbf{S} is a point on a deformable object. We can perform several consecutive manipulations that move \mathbf{S} to pass first through \mathbf{T}_1 , and then through \mathbf{T}_2 , \mathbf{T}_3 , etc. The manipulations are denoted by $\omega(\mathbf{S}, \boldsymbol{\rho}_i)$, where the displacements $\boldsymbol{\rho}_1 = \mathbf{T}_1 - \mathbf{S}$, and $\boldsymbol{\rho}_i = \mathbf{T}_i - \mathbf{T}_{i-1}$, $i = 2, 3, \dots$. We define a binary operator to represent the composition of two FFD direct manipulations: $\omega(\mathbf{S}, \boldsymbol{\rho}_i) \circ \omega(\mathbf{S}, \boldsymbol{\rho}_j)$. From (7), we obtain $\omega(\mathbf{S}, \boldsymbol{\rho}_i) \circ \omega(\mathbf{S}, \boldsymbol{\rho}_j) = \omega(\mathbf{S}, \boldsymbol{\rho}_i + \boldsymbol{\rho}_j)$, $i \neq j$. Since the composition entails a vector addition, clearly (Ω, \circ) is a commutative group, where Ω denotes the set of all direct manipulations; that is, Ω is commutative and associative under the operation \circ , it has the unit element $\omega(\mathbf{S}, \mathbf{0})$, and for each element $\omega(\mathbf{S}, \boldsymbol{\rho})$ in Ω , there exists an inverse element $\omega(\mathbf{S}, -\boldsymbol{\rho})$.

The direct manipulation provides a practical and explicit editing tool. The commutative group property further enhances the controllability of the technique in the following ways:

1. The result of a direct manipulation only depends on the starting point \mathbf{S} on the object to be deformed and on the final target point \mathbf{T} . Users can thus modify the shape of an object using any appropriate direct manipulations.

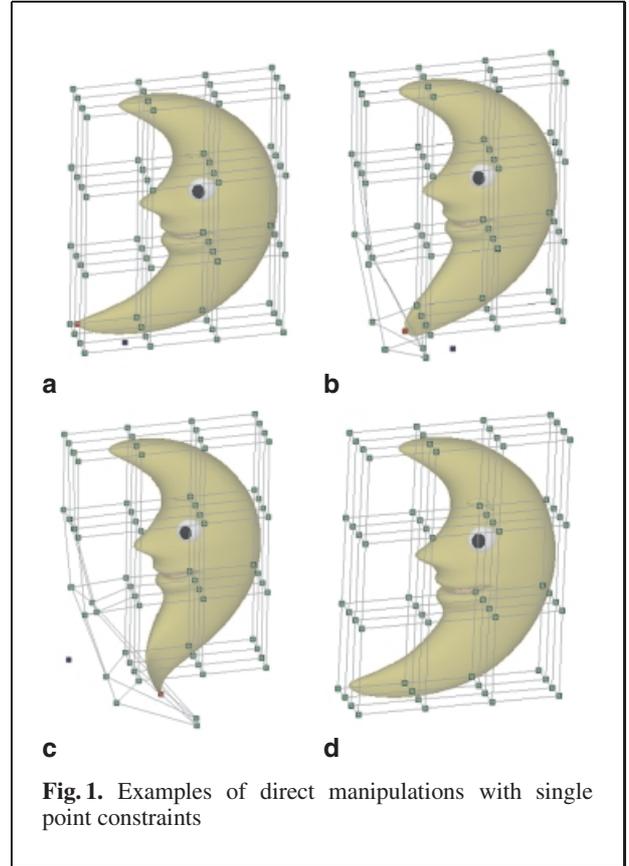


Fig. 1. Examples of direct manipulations with single point constraints

2. The existence of the inverse element enables the user to “undo” an undesired manipulation to revert to the original shape.

Figure 1 shows some examples of direct manipulations with single point constraints. The source point on the surface and its target point are shown in red and blue, respectively. The embedding lattice satisfying the specified constraints is automatically computed using the explicit solution. The first two manipulations are undone to revert to the original shape shown in Fig. 1d. Figure 2 illustrates the commutative group property; the displacements performed to the source point on the surface are $\boldsymbol{\rho}_1$ followed by $\boldsymbol{\rho}_2$ in Fig. 2a, and are $\boldsymbol{\rho}_2$ followed by $\boldsymbol{\rho}_1$ in Fig. 2b.

5 Direct deformation with multiple point constraints

The user often wants to specify more than one constraint at a time. If these constraints are mutually independent, i.e. there are no common control

By solving for $\delta_{i,j,k}$ from the above system of linear equations, we derive the optimum solution of FFD direct manipulation under multiple point constraints. We now consider the existence of the solution of (10a) and (10b). To simplify the expressions, we let t_f denote the parameter values (u_f, v_f, w_f) of the point S_f . We write (10a) and (10b) in matrix form

$$\begin{cases} \mathbf{D} = \mathbf{R}^T \delta & (11a) \\ \delta = \mathbf{R} \lambda & (11b) \end{cases}$$

where $\mathbf{D} = [\mathbf{D}_1, \dots, \mathbf{D}_h]_{1 \times h}^T = [\mathbf{T}_1 - \mathbf{S}_1, \dots, \mathbf{T}_h - \mathbf{S}_h]_{1 \times h}^T$,

$$\delta = [\delta_{0,0,0}, \delta_{0,0,1}, \dots, \delta_{l,m,n}]_{1 \times [(l+1) \times (m+1) \times (n+1)]}^T,$$

$$\lambda = \frac{1}{2} [\lambda_1, \dots, \lambda_h]_{1 \times h}^T,$$

$\mathbf{R} =$

$$\begin{bmatrix} R_{0,0,0}(t_1) & \dots & R_{0,0,0}(t_h) \\ R_{0,0,1}(t_1) & \dots & R_{0,0,1}(t_h) \\ \vdots & \ddots & \vdots \\ R_{l,m,n}(t_1) & \dots & R_{l,m,n}(t_h) \end{bmatrix}_{[(l+1) \times (m+1) \times (n+1)] \times h}$$

Substituting (11b) into (11a), we obtain $\mathbf{D} = \mathbf{R}^T \mathbf{R} \lambda$, and thus $\lambda = (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{D}$. Solving for δ yields

$$\delta = \mathbf{R}(\mathbf{R}^T \mathbf{R})^{-1} \mathbf{D} = \mathbf{R}(\mathbf{R}^T \mathbf{R})^{-1} (\mathbf{T} - \mathbf{S}). \quad (12)$$

Hence, the existence of (10a) and (10b) depends on the existence of the inverse matrix of \mathbf{R} .

6 Decomposability of multiple point constraints

We now show that a direct manipulation with multiple point constraints can be decomposed into separate direct manipulations involving only single point constraints. To sculpt a complex model using the FFD method, it is often convenient to specify multiple constraints at a time. The decomposability property of the multiple point constraints further implies that the desired shape can also be obtained by interactively specifying only single point constraints.

Theorem. A direct manipulation of FFD with h point constraints can be decomposed into h manipulations with single point constraints.

Proof. Let S_1, S_2, \dots, S_h be source points on an object O , and T_1, T_2, \dots, T_h be the corresponding

target points. We assume the existence of the solution of this direct manipulation.

Suppose that the FFD direct manipulation with h point constraints can be decomposed into h steps of FFD direct manipulation with single point constraints. The first manipulation moves S_1 to $T_1^{(1)}$, which also causes S_f to move to some $T_f^{(1)}$, ($f = 2, \dots, h$). Denote the displacement of each source point in the first step as $D_f^{(1)}$, ($f = 1, \dots, h$) and the displacement of each control point in this step as $\delta_{i,j,k}^{(1)}$, ($0 \leq i \leq l, 0 \leq j \leq m, 0 \leq k \leq n$). The second step moves $T_2^{(1)}$ to $T_2^{(2)}$, which causes $T_f^{(1)}$ to move to some $T_f^{(2)}$, ($f = 1, \dots, h, f$ not equal 2), and so on. In general, the q th step moves $T_q^{(q-1)}$ to $T_q^{(q)}$, and causes $T_f^{(q-1)}$ to move to $T_f^{(q)}$, ($f = 1, \dots, h, f \neq q$). Denote the displacement of each source point in step q as $D_f^{(q)}$, ($f = 1, \dots, h$) and the displacement of each control point in this step as $\delta_{i,j,k}^{(q)}$, ($0 \leq i \leq l, 0 \leq j \leq m, 0 \leq k \leq n$). After h steps, points S_f are finally moved to T_f , ($f = 1, \dots, h$).

In the q th step, the direct manipulation is performed to the q th point, which yields the following equation:

$$\delta_{i,j,k}^{(q)} = \frac{R_{i,j,k}(t_q) D_q^{(q)}}{\sum_{i,j,k=0}^{l,m,n} R_{i,j,k}^2(t_q)}. \quad (13)$$

The displacements of all other points caused by the moving of the q th point are

$$D_p^{(q)} = \sum_{i,j,k=0}^{l,m,n} \delta_{i,j,k}^{(q)} R_{i,j,k}(t_p), \quad p = 1, \dots, h \quad (14)$$

Thus the total displacements of the p th point in all the h steps are

$$\begin{aligned} D_p &= T_p - S_p = \sum_{q=1}^h D_p^{(q)} & (15) \\ &= \sum_{q=1}^h \frac{\sum_{i,j,k=0}^{l,m,n} R_{i,j,k}(t_q) R_{i,j,k}(t_p)}{\sum_{i,j,k=0}^{l,m,n} R_{i,j,k}^2(t_q)} D_q^{(q)}, \\ & \quad p = 1, \dots, h \end{aligned}$$

Let

$$\mathbf{M} = \left[\frac{D_1^{(1)}}{\sum_{i,j,k=0}^{l,m,n} R_{i,j,k}^2(t_1)} \quad \dots \quad \frac{D_h^{(h)}}{\sum_{i,j,k=0}^{l,m,n} R_{i,j,k}^2(t_h)} \right]_{1 \times h}^T,$$

we obtain $\mathbf{R}^T \mathbf{R} \mathbf{M} = \mathbf{D}$; thus,

$$\mathbf{M} = (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{D} \quad (16)$$

and the total displacements of each control point is

$$\delta_{i,j,k} = \sum_{q=1}^h \delta_{i,j,k}^{(q)} = \sum_{q=1}^h \frac{R_{i,j,k}(\mathbf{t}_q) \mathbf{D}_q^{(q)}}{\sum_{i,j,k=0}^{l,m,n} R_{i,j,k}^2(\mathbf{t}_q)}, \quad (17)$$

$$0 \leq i \leq l, 0 \leq j \leq m, 0 \leq k \leq n.$$

Writing (17) in matrix form, we obtain

$$\delta = \mathbf{R} \mathbf{M} = \mathbf{R} (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{D} \quad (18)$$

which is the same as (12). Thus, the effect produced by these h separate direct FFD with single point constraints is the same as the effect of one direct manipulation with h multiple point constraints. This completes the proof of the decomposability of multiple point constraints. •

From the above result, not only can we obtain the displacement $\delta_{i,j,k}^{(q)}$, $q = 1, \dots, h$ of each control point in each step, but we can also obtain the explicit solution of the displacement $\mathbf{D}_q^{(q)}$, $q = 1, \dots, h$ of the q th point to be moved in the q th step.

$$\mathbf{D}_q^{(q)} = \frac{\delta_{i,j,k}^{(q)} \sum_{i,j,k=0}^{l,m,n} R_{i,j,k}^2(\mathbf{t}_q)}{R_{i,j,k}(\mathbf{t}_q)}, \quad q = 1, \dots, h. \quad (19)$$

7 Modeling examples

We have modeled several examples using direct manipulations with multiple point constraints. Cubic NURBS volumes with uniform knot vectors are used in all the examples. The bottle in Fig. 3 is obtained by specifying four point constraints. In Fig. 4, a hollow cylinder is transformed into a cup using multiple point constraints; the target points are not shown in the figure due to their large number. Figure 5 shows the results of specifying four point constraints and using lattices of different resolutions in Fig. 5b–d. Two constraints are specified near the mouth of the fish and another two constraints near its tail. It is observed that with lattices of higher resolutions, the deformation is confined to regions closer to the specified source points. Figure 6 shows an example of local deformation, achieved by specifying two constraints at the top of the alien's

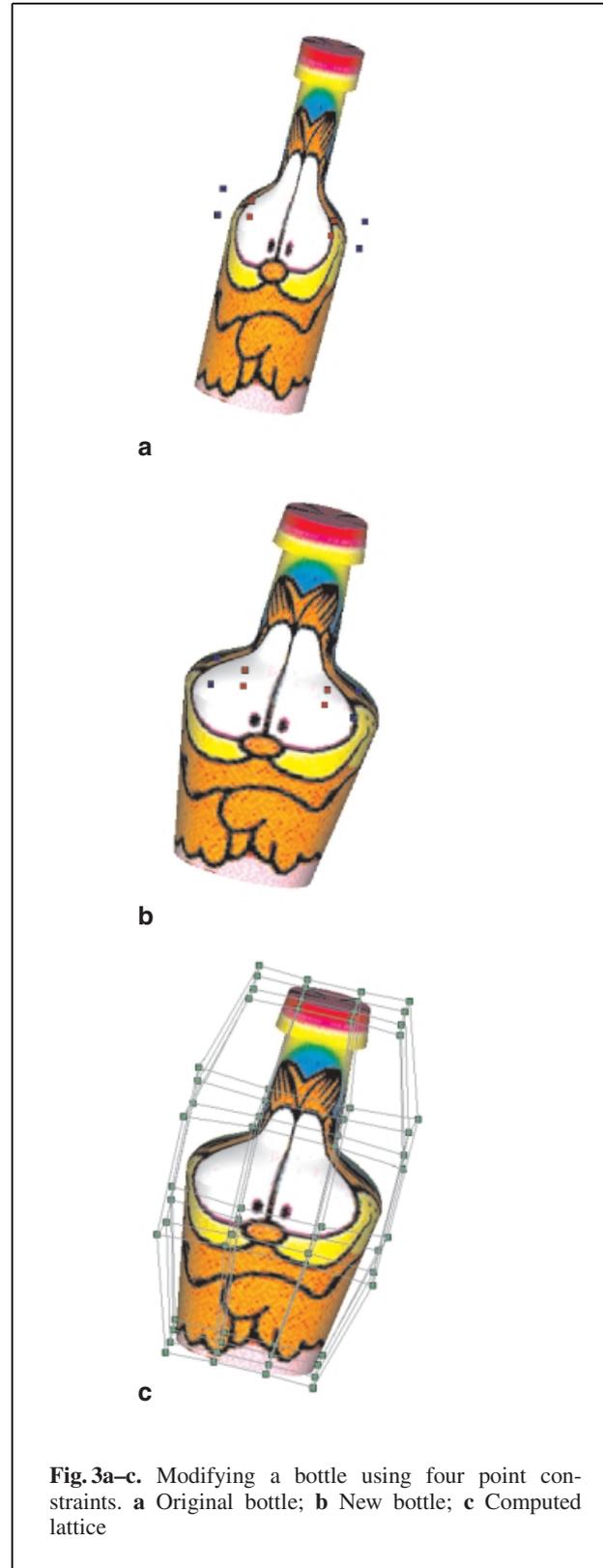


Fig. 3a–c. Modifying a bottle using four point constraints. **a** Original bottle; **b** New bottle; **c** Computed lattice

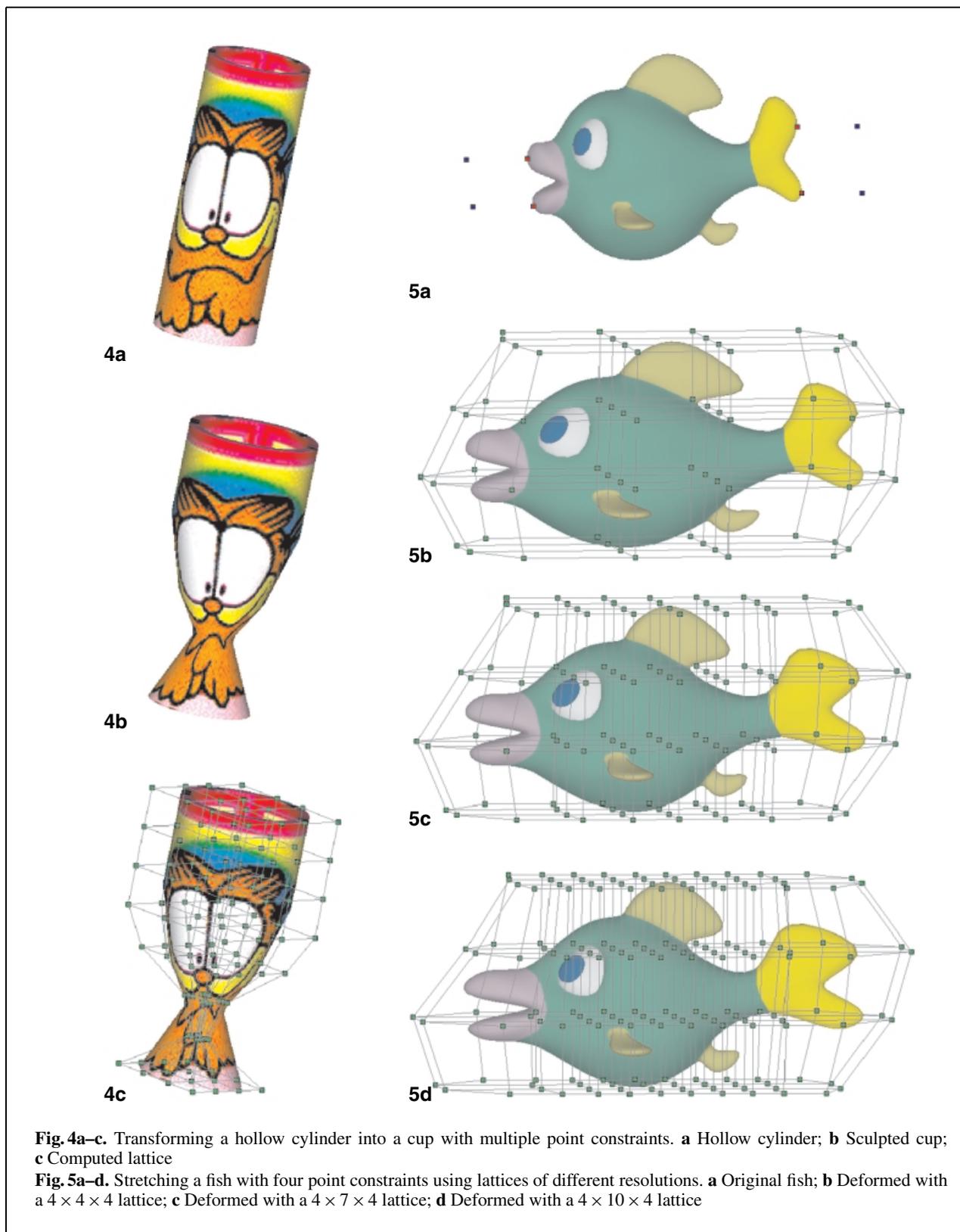


Fig. 4a–c. Transforming a hollow cylinder into a cup with multiple point constraints. **a** Hollow cylinder; **b** Sculpted cup; **c** Computed lattice

Fig. 5a–d. Stretching a fish with four point constraints using lattices of different resolutions. **a** Original fish; **b** Deformed with a $4 \times 4 \times 4$ lattice; **c** Deformed with a $4 \times 7 \times 4$ lattice; **d** Deformed with a $4 \times 10 \times 4$ lattice

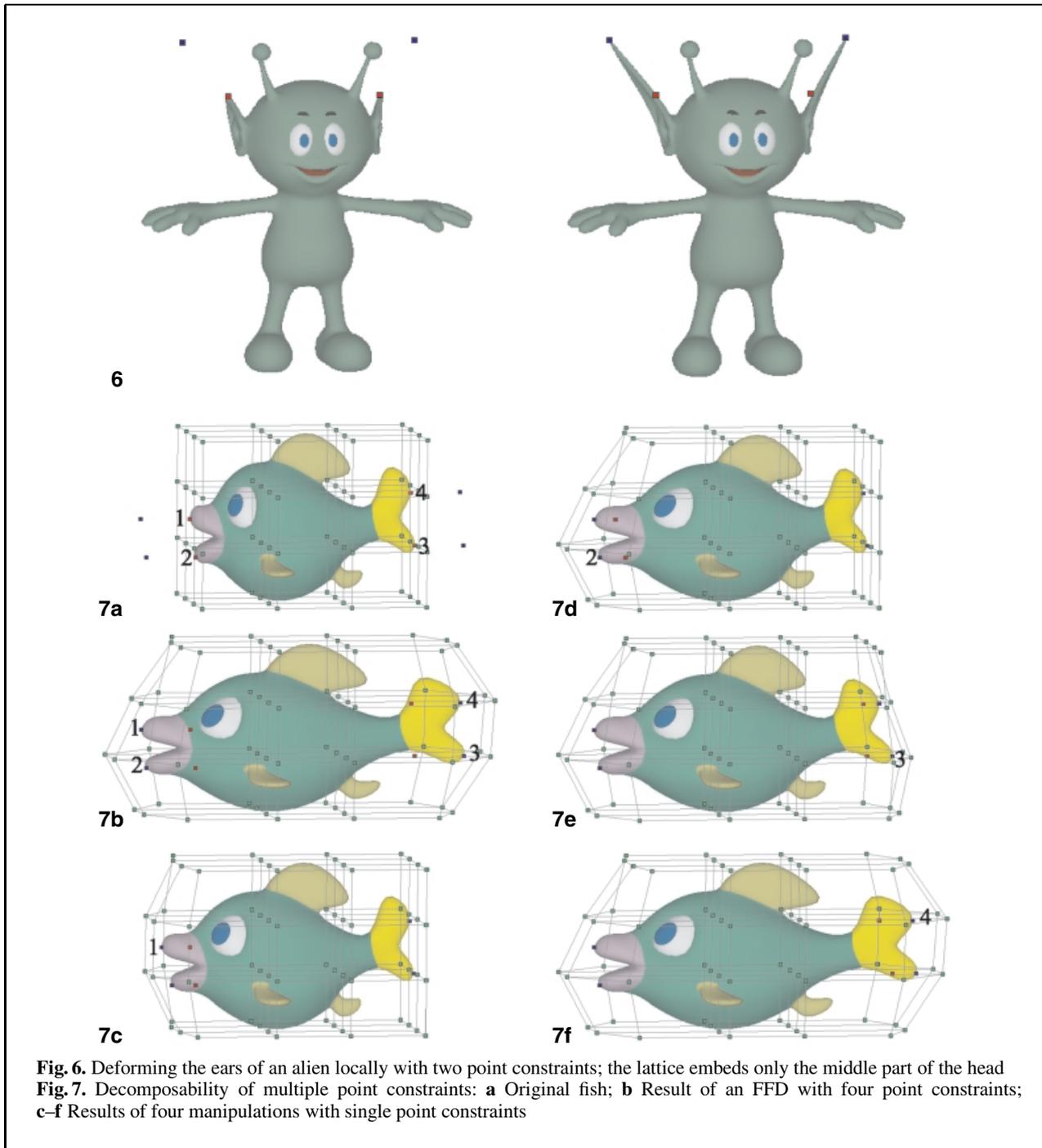


Fig. 6. Deforming the ears of an alien locally with two point constraints; the lattice embeds only the middle part of the head
Fig. 7. Decomposability of multiple point constraints: **a** Original fish; **b** Result of an FFD with four point constraints; **c-f** Results of four manipulations with single point constraints

ears and embedding only the middle part of the head in a lattice. Figure 7 illustrates the decomposability property of multiple point constraints. Figure 7a shows the original fish with four points to be deformed; the result of FFD with four point

constraints is given in Fig. 7b. The decomposition process of this manipulation with four constraints to four separate manipulations with the i th point constrained in the i th step ($i = 1, \dots, 4$) is shown in Fig. 7c-f.

8 Conclusions

The direct manipulation method of FFD proposed by Hsu et al. (1992) is based on computing a pseudo-inverse matrix that involves complicated calculations. We solve the direct manipulation problem using a constrained optimization method and obtain the explicit solutions involving only simple calculations. The new technique excels in its simplicity and geometric explicitness in comparison with the previous approach, and it is easy to implement. We have also proved the commutative, associative, and invertible properties of the direct manipulations, as well as the decomposability property of the multiple point constraints problem. With these properties, the efficiency and controllability of the direct manipulation of FFD are greatly enhanced.

Acknowledgements. We thank Mr. Hou Soon Ming for providing the models of moon, fish, and alien and Yangzhou Du for his helpful suggestion in simplifying the proof of the theorem.

This work was supported by the Natural Science Foundation of China (project number 69902004), and the Hong Kong Research Grant Council (HKUST6215/99E).

References

1. Barr AH (1984) Global and local deformation of solid primitives. *Comput Graph* 18(3): 21–30
2. de Boor C (1978) *A practical guide to Splines*. Springer, New York
3. Bechmann D (1994) Space deformation models survey. *Comput Graph* 18(4): 571–586
4. Borrel P, Bechmann D (1991) Deformation of N-dimensional objects. *Int J Comput Geom Appl* 1(4): 137–155
5. Chang YK, Rockwood AP (1994) A generalized de Casteljau approach to 3D free-form deformation 28(4): 257–260
6. Celniker G, Gossard D (1991) Deformable curve and surface finite-elements for free-form shape design. *Comput Graph* 25: 257–266
7. Coquillart S (1990) Extended free-form deformation: a sculpturing tool for 3D geometric modeling. *Comput Graph* 24(4): 187–193
8. Coquillart S, Jancène P (1991) Animated free-form deformation: an interactive animation technique. *Comput Graph* 25(4): 23–26
9. Feng JQ, Ma LZ, Peng QS (1996) New free-form deformation through the control of parametric surfaces. *Comput Graph* 20(4): 531–539
10. Griessmair J, Purgathofer W (1989) Deformation of solids with trivariate B-spline. In: Hansmann W, Hopgood FRA, Strasser W (eds) *Proc. Eurographics '89*. Elsevier, North-Holland, pp 137–148
11. Hsu WM, Hughes JF, Kaufman H (1992) Direct manipulation of free-form deformation. *Comput Graph* 26(2): 177–184
12. Lamousin HJ, Waggenspack WN (1994) NURBS-based free-form deformations. *IEEE Comput Graph Appl* 14(6): 59–65
13. Lazarus F, Coquillart S, Jancène P (1994) Axial deformation: an intuitive technique. *Comput Aided Des* 26(8): 607–613
14. MacCracken R, Joy K (1996) Free-form deformation with lattices of arbitrary topology. *Comput Graph* 30(4): 181–189
15. Moccozet L, Thalmann NM (1997) Dirichlet free-form deformations and their application to hand simulation. In: Thalmann NM, Thalmann D (eds) *Proc. of the Computer Animation '97*, IEEE Computer Society Press, pp 93–102
16. Schumaker LL (1980) *Spline functions: basic theory*. Wiley, New York
17. Sederberg TW, Parry SR (1986) Free-form fundamentals of solid geometry. *Comput Graph* 20(4): 151–160
18. Singh K, Fiume E (1998) Wires: a geometric deformation technique. *Comput Graph* 32(4): 405–414

Photographs of the authors and their biographies are given on the next page.



SHI-MIN HU is an associate professor in the Department of Computer Science and Technology at Tsinghua University. He received a BSc in mathematics from Jilin University of China in 1990, an MSc and a PhD in CAGD and computer graphics from Zhejiang University of China in 1993 and 1996, respectively. He finished his postdoctoral research at Tsinghua University in 1998. His current research interests are in computer-aided geometric design and computer graphics. A list of recent publications can be found at <http://ncc.cs.tsinghua.edu.cn/~shimin>.



HUI ZHANG is a PhD student in the Department of Computer Science and Technology, Tsinghua University, Beijing, where she received a BSc in 1997. Her research interests are computer-aided design and computer graphics.



CHIEW-LAN TAI received her BSc and MSc in mathematics from the University of Malaya, and her MSc in computer and information sciences from the National University of Singapore. She pursued her PhD from the University of Tokyo on a fellowship awarded by the Japan Society for the Promotion of Science. She joined the Hong Kong University of Science and Technology, Department of Computer Science, as an Assistant Professor in August 1997. Her research interests include geometric modeling, computer graphics, and scientific visualization.



JIA-GUANG SUN is a professor in the Department of Computer Science and Technology at Tsinghua University. He is also the Director of the National CAD Engineering Center at Tsinghua University and Academician of the Chinese Academy of Engineering. He received a BSc in computer science from Tsinghua University in 1972. Between 1985 and 1986, he was a visiting scholar at UCLA. His current research interests are in computer-aided geometric design, computer graphics, and product data management.