Claymore: Augmented Direct Manipulation of Three-Dimensional Objects

Hideki Mitsunobu, Takashi Oshiba and Jiro Tanaka
University of Tsukuba,
1-1-1 Tennodai Tsukuba Ibaraki 305-0006 Japan
Phone:+81-298-53-5165, Fax:+81-298-53-5206
e-mail:{copel, ohshiba, jiro}@softlab.is.tsukuba.ac.jp

Abstract

This paper describes a method for manipulating a three-dimensional object without considering conventional three independent orthogonal views. We apply the direct manipulation to operate the three-dimensional object. We propose **"augmented manipulation"** *which is an enhanced direct manipulation technique by using additional information.*

We implement three-dimensional modeling tool "Claymore" by using the **"augmented manipulation"** *technique. Users can construct three-dimensional models in an intuitive manner.*

"Claymore" and the examples described in this paper are fully implemented and can be obtained via WWW.

Keywords — *3-D interface, virtual reality, design environment, visual programming, World Wide Web.*

1. Problems of three-independent orthogonal views

Conventional three-dimensional modeling tools use three independent orthogonal views [7]. The three independent orthogonal views are used as the method to operate a three-dimensional object. It is composed of three views. Each view is represented by two of the three orthogonal axes of x, y, z. When users operate the three-dimensional object by using the three independent orthogonal views, they often change the view. Therefore, they cannot operate the object intuitively.

2. Augmented Manipulation

Direct manipulation is the operation in which users' operation directly invokes the reaction of the system [1]. We apply the direct manipulation technique to operate a three-dimensional object. Some work on editing a three-dimensional object by using the direct manipulation technique have already been proposed [5, 6]. But they require a special input device. We think it is beneficial to make users be able to manipulate a three-dimensional object without using the special input device. In this paper, we use the mouse as our input device instead of using the special input device. We implement a system in which users can manipulate a three-dimensional object by using a mouse.

When users manipulate a three-dimensional object, they have the difficulty to specify the position of the object because the input device is two-dimensional. When they operate the three-dimensional object, there is a lack of information because the display of a computer is two-dimensional. To solve these problems, we propose **"augmented manipulation"** which is an enhanced direct manipulation technique by using additional information (see Figure 1). We use the following objects to represent the additional information.

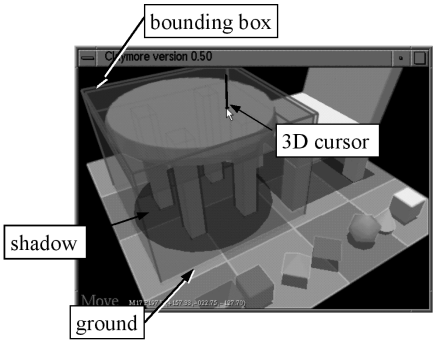

Figure 1. The additional information.

- Ground

  By showing the ground, users can grasp the spatial position of an object in a three-dimensional space. They can regard the ground as the standard position of the three-dimensional space.

- Shadow

By showing the shadow of an object on the ground, users can grasp the position of the object relative to other objects. They can understand the outline of the object by looking at the shape of the shadow.

- Three-dimensional cursor

  The position which is specified by users is represented as a three-dimensional cursor. Its shape is like an arrow.

- Bounding box

  A bounding box is a parallel hexahedron which encloses the specified object. Users often want to move an object along a plane which is used in the three independent orthogonal views. By using a plane which composes the bounding box, they can perform this operation without using the three independent orthogonal views.

We show all the additional information semi-transparently. When an object overlaps with additional information, showing the additional information semi-transparently is more effective [3].

3. The concept of modeler "Claymore"

We implement three-dimensional modeling tool "Claymore". We emphasize the following concepts:

- Augmented manipulation

  Users can operate a three-dimensional object directly by the additional information.

- Intuitive operation

  The object that is represented by the surface model is composed of a group of polygons. The inside of the object is empty. We show the object to users as if it is not empty.

3.1. The surface model

We use the surface model which is common to represent three-dimensional model. The surface model is represented by polygons. A polygon is defined by a group of vertices.

3.2. Specifying the position of a three-dimensional space

The following two pieces of information are used in this modeler:

- The mouse position.
- The shape of the object.

The system calculates the three-dimensional coordinate values of an object in the two-dimensional window. Users look at a two-dimensional image. They specify a point in the window by using a mouse. They can touch the surface of the three-dimensional object.

We have the following assumption to simplify the model of the perspective transformation: The viewing coordinate has the origin which exists in the center of the window (see Figure 2).

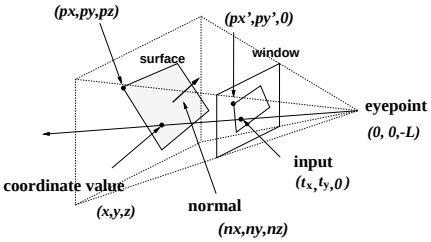

Figure 2. Specifying the coordinate values.

Let (p_x, p_y, p_z) be a point in the three-dimensional space. After the perspective transformation, (p_x, p_y, p_z) is transformed to (p'_x, p'_y) which is in the two-dimensional window:

$$p'_x = \frac{L}{p_z + L} p_x,$$
$$p'_y = \frac{L}{p_z + L} p_y.$$

Where L is the distance between the eye point and the window.

The system calculates the coordinate values of an intersection point between the following two elements (see Figure 2):

(1) A line which connects the eye point and the point specified by the mouse cursor in the two-dimensional window.

(2) A plane which composes the three-dimensional object that is specified by the mouse cursor in the window.

Let $(t_x, t_y, 0)$ be the position of the mouse cursor in the window. The equation of the line (1) which passes the eye point $(0, 0, -L)$ and the position of the mouse cursor $(t_x, t_y, 0)$ in the window is computed by:

$$\frac{x}{t_x} = \frac{y}{t_y} = \frac{z + L}{L}. \quad (1)$$

The equation of the plane (2) which composes the specified object is computed by:

$$n_x x + n_y y + n_z z + d = 0. \quad (2)$$

Where (n_x, n_y, n_z) is the normal vector of plane (2). The intersection point between the line (1) and the plane (2) is:

$$z = \frac{-(n_x t_x + n_y t_y + d)}{\frac{n_x t_x + n_y t_y}{L} + n_z},$$

$$x = \frac{t_x(z+L)}{L}, \quad y = \frac{t_y(z+L)}{L}.$$

(x, y, z) above is the three-dimensional point which is specified by users.

3.3. Moving an object

When users pick and move an object in the window, the object should be moved in accordance with the mouse movement.

Three parameters of x, y, z, are necessary for the movement of an object in a three-dimensional space. However, there are only two parameters because the mouse movement is two-dimensional. There is a lack of information. To solve this problem, we prepare two planes which restrict the movement of an object (see Figures 3 and 4). One is parallel to the ground. The other is vertical to the ground.

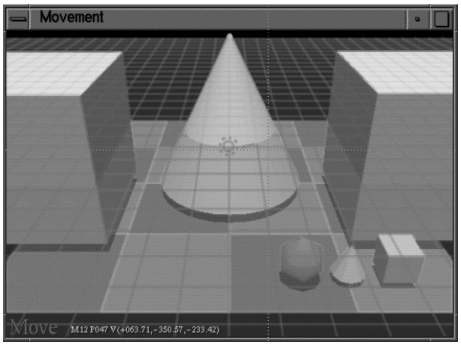

Figure 3. The plane that is parallel to the ground.

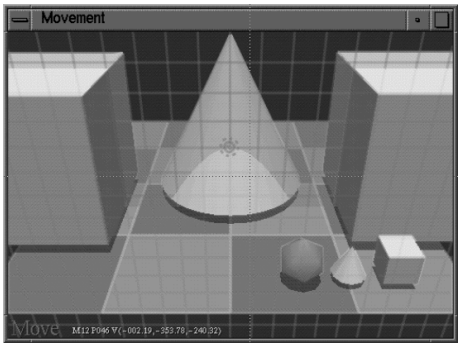

Figure 4. The plane that is vertical to the ground.

They can select the plane by using the left or right button of the mouse when they move the object.

When users move the object while holding down the left button of the mouse, the object is moved along the plane that is parallel to the ground (see Figure 3). If they move the mouse cursor to the right (left), the object in the window is also moved to the right (left). If they move the mouse cursor up (down), the object in the window is moved to the front (back).

When users move the object while holding down the right button of the mouse, the object is moved along the plane that is vertical to the ground (see Figure 4).

The normal of the plane that restricts the movement of the object is computed by:

- $(0, 1, 0)$ — the plane is parallel to the ground.
- $(0, 0, 1)$ — the plane is vertical to the ground.

The new position of the object after moving it is the intersection point between the following two elements:

(1) The plane that restricts the movement of the object.

(2) The line which connects the eye point and the point specified by the mouse cursor in the two-dimensional window.

3.4. Rotating an object

The object rotates to the direction to which the mouse moves.

By using the conventional method of rotation [7, 9], the distance of the mouse movement is resolved into small distances, and the object is rotated little by little. However, the conventional method has a problem that if users return the mouse cursor to the beginning point of the rotation, the posture of the object may not return to the initial position (see Figure 5).

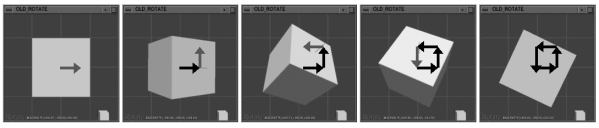

Figure 5. A conventional rotation technique: The final position is different from the initial one.

The direct manipulation technique requires “reversible manipulation” [1]. We adopt the reversible manipulation while rotating an object. Users can rotate the object with the corresponding direction of the mouse. If users move back the mouse cursor to the beginning point of the rotation, the posture of the object is returned to the initial position.

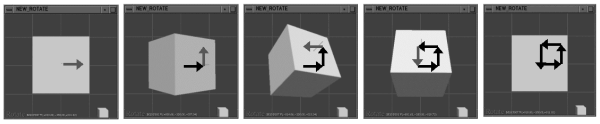

Figure 6. Our rotation technique: The final position and the initial one are the same.

When users rotate the object, the axis of rotation that is vertical to the direction of mouse movement is calculated. The object is rotated along this axis (see Figure 7).

A rotation angle is computed by the direction and the distance of the mouse movement. This rotation method is different from the conventional approaches because the rotation angle is independent of the path of the mouse movement.

Our rotation method is calculated by the following processes:

Let (t_x, t_y) be the coordinate values of the mouse cursor, M_{LW} be the transformation matrix which is used to transform the modeling coordinate into the world coordinate. (t_x, t_y) and M_{LW} are stored to be used as the initial condition of the rotation.

Let (m_x, m_y) be the current coordinate values of the mouse cursor. Let r be the distance between the position of the mouse click and the current position of the mouse. Let θ be the direction of the mouse movement (see Figure 7). r and θ can be computed by:

$$r = k\sqrt{(m_x - t_x)^2 + (m_y - t_y)^2},$$

$$\theta = atan\frac{m_y - t_y}{m_x - t_x}.$$

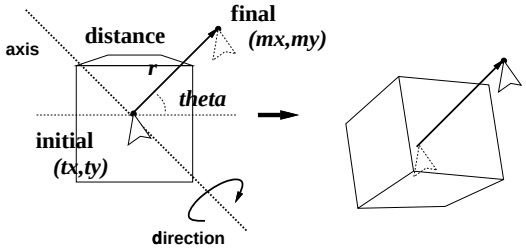

Figure 7. The rotation of the object.

The axis of rotation is vertical to the direction of the mouse movement. M_{LW} is transformed by rotating object along the axis. The new transformation matrix M'_{LW} is computed by:

$$M'_{LW} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos r & 0 & -\sin r \\ 0 & 1 & 0 \\ \sin r & 0 & \cos r \end{pmatrix}$$

$$\begin{pmatrix} \cos -\theta & \sin -\theta & 0 \\ -\sin -\theta & \cos -\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} M_{LW}.$$

By using M'_{LW} as the transformation matrix, users can rotate the object with the corresponding direction of mouse movement.

3.5. Deforming an object

The Free-Form Deformation (FFD) technique is the deformation method of an object [8].

An object can be transformed smoothly by using FFD. An object is surrounded with a bounding box which has several control points. Users can transform the object by moving the control points. We use simplified FFD.

We prepared eight control points for the vertices of the bounding box. Users can move four control points simultaneously by moving the plane that composes the bounding box (see Figure 8).

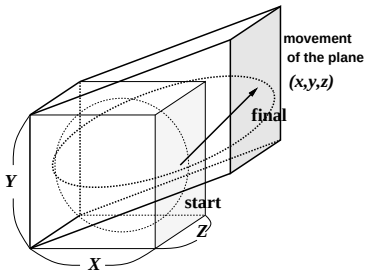

Figure 8. Simplified FFD.

Let (X, Y, Z) be the size of the bounding box (see Figure 8). Let (x, y, z) be the amount of movement of the plane of the bounding box. The transformation matrix of the deformation is computed by the following cases:

- the specified plane is vertical to the axis X:

$$\begin{pmatrix} 1+\frac{x}{X} & \frac{y}{X} & \frac{z}{X} & \frac{x}{2} \\ 0 & 1 & 0 & \frac{y}{2} \\ 0 & 0 & 1 & \frac{z}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

- the specified plane is vertical to the axis Y:

$$\begin{pmatrix} 1 & 0 & 0 & \frac{x}{2} \\ \frac{x}{X} & 1+\frac{y}{X} & \frac{z}{X} & \frac{y}{2} \\ 0 & 0 & 1 & \frac{z}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

- the specified plane is vertical to the axis Z:

$$\begin{pmatrix} 1 & 0 & 0 & \frac{x}{2} \\ 0 & 1 & 0 & \frac{y}{2} \\ \frac{x}{X} & \frac{y}{X} & 1+\frac{z}{X} & \frac{z}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

All the vertices of the object are transformed by using this matrix.

They can transform the object in the bounding box smoothly. The moving method of the plane that composes the bounding box is the same as the method which is used in **3.3**.

3.6. Cutting and grouping an object

Users can cut an object by using a cutter plane. When the cutter plane cuts the object, the object is split into two parts.

They can move the cutter plane by using the mouse movement. The normal of the cutter plane depends on the point where they click (see Figure 9). There are three possibilities of the direction:

a — a surface.

b — an edge of the object.

c — a vertex of the object.

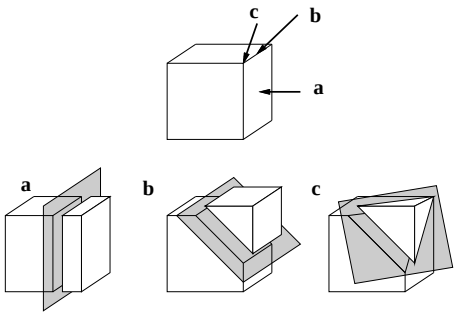

Figure 9. Three kinds of the cutter plane.

They can move the cutter plane along the direction of its normal.

When they split an object, the system substitutes all the coordinate values of the vertices of the object for the equation of the cutter plane ($ax + by + cz + d = 0$). The system divides the object into two parts by using the result of the substitution. One is the right side of the plane ($ax + by + cz + d \geq 0$). The other is the reverse side of the plane ($ax + by + cz + d < 0$).

After cutting, the system creates new two surfaces automatically, at the place of the cutter plane. Because of the automatic creation of the two surfaces, the users feel as if the objects are not empty.

Users can group selected objects. When they select an object by clicking the mouse button, the bounding box of the object appears. More than one object can be selected to form a group. The bounding box for the group of selected objects is constructed by combining the bounding boxes of each of the objects in the group.

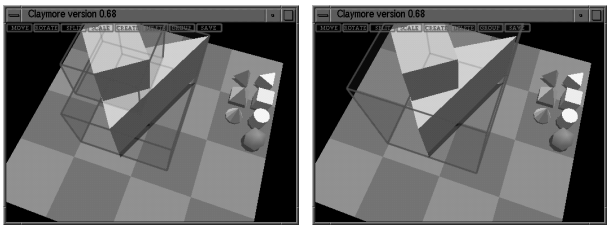

Figure 10. Grouping objects.

The group of objects can be manipulated as one combined object. Users can make a complex model by using the cutting and grouping operations.

3.7. Creating an object

We implement two kinds of methods for the creation of an object.

3.7.1 Creating a fundamental model

We prepare the following objects as fundamental models:

- Sphere,
- Cylinder,
- Cone,
- Cube,
- Square pyramid,
- Triangular prism,
- Tetrahedron.

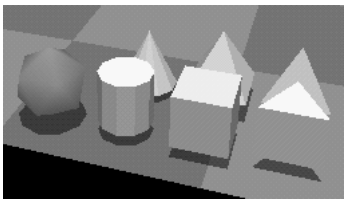

Figure 11. The fundamental models.

The miniatures of these models are placed as icons. When users click one of the miniatures, the model appears in the center of the window. Its shape corresponds to its original model.

3.7.2 Creating a model on the surface of the other object

Users can create an object on the surface of the other object. They can control the size of the object by using the mouse movement. When the object is created, the bounding box of the object is used (see Figure 12).

The base of the bounding box is decided as follows:

Position — The center point of the base of the bounding box is decided based on the beginning point of the mouse movement.

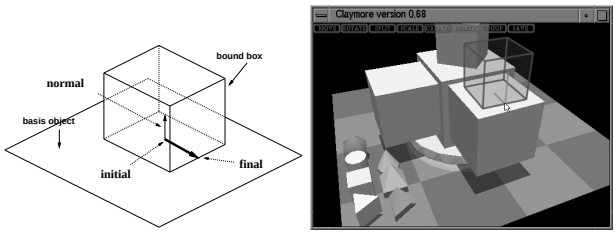

Figure 12. Creating an object.

Posture — The normal of the base is the normal of the surface of the other object.

Size — The base length of the base is twice the amount of the mouse movement.

They can create an object onto the other object that is previously created by this method. They can make a model by piling up these objects like a building block.

4. The modeling tool "Claymore"

"Claymore" is implemented with OpenGL [11, 12, 13]. It runs on Unix and Microsoft Windows.

We implement **"augmented manipulation"** for this modeler.

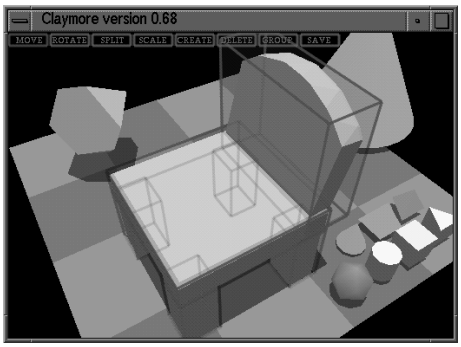

Figure 13. Screen shot of "Claymore".

Users can manipulate an object overlapping the additional information by using the mouse, while looking at the perspective figure at the same time (see Figure 13).

We have implemented the following functions. The users can select the buttons that are placed at the top of the window (see Figure 14). They can change an operation mode by using the buttons. These buttons show the following functions: Move, Rotate, Cut, Deform, Create, Delete, Group, and Save to VRML format. Users can store the modeling data to VRML format or alternately Wavefront OBJ format.

The model that is made by using "Claymore" can be exported to other three dimensional tools. They can make use

Figure 14. Selecting buttons.

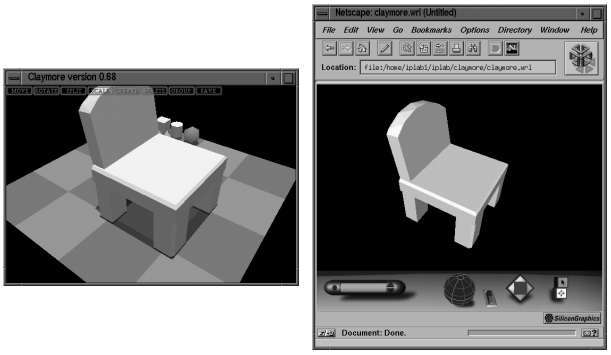

Figure 15. Saving to VRML format.

of the modeling data while creating VRML web pages (see Figure 15).

5. Related work

Michael Chen et al. [9] evaluated the method of rotation in a three-dimensional space by using a mouse. They proposed the method of rotation by using a virtual sphere controller. In this paper, a rotation angle is computed by the direction and the distance of the mouse movement. This rotation method is different from their approaches because the rotation angle is independent of the path of the mouse movement.

Robert C. Zeleznik et al. [10] proposed a direct manipulation technique for operating a three-dimensional object. Their technique was implemented in their three-dimensional modeler "SKETCH" which used a pen as a two-dimensional input device. Users did not look at an object in perspective because "SKETCH" projected a three-dimensional object to a window by using the orthogonal transformation. In this paper, our modeler "Claymore" adopts a mouse as the input device. The object is in perspective because "Claymore" uses perspective transformation.

6. Summary

We apply the direct manipulation to operate a three-dimensional object. We propose **"augmented manipulation"** which is an enhanced direct manipulation technique by using additional information. We implement the three-dimensional modeling tool "Claymore" by using this technique.

This research is supported in part by Grant-in-Aid for Scientific Research #09878076 and Parallel and Distributed processing Consortium (PDC). “Claymore” and the examples described in this paper are fully implemented and can be obtained via WWW from the following URL.

http://www.softlab.is.tsukuba.ac.jp/iplab/claymore/

References

[1] Ben Shneiderman: The principle of direct manipulation, The method to design a user interface, Nikkei BP company, pp.126-161 *(in Japanese)*.

[2] Masayuki Nakajima: The three-dimensional computer graphics, Ohm company *(in Japanese)*.

[3] Hideki Mitsunobu, Jiro Tanaka: The idea supporting system with three-dimensional view, 13th Conference Proceedings Japan Society for Software Science and Technology, pp.357-360, September, 1996 *(in Japanese)*.

[4] Hideki Mitsunobu, Jiro Tanaka: “Claymore”: The three-dimensional modeling tool by using direct manipulation, Workshop on Interactive Systems and Software pp.212, December, 1997 *(in Japanese)*.

[5] Juli Yamashita, Yukio Fukui, Hiroshi Yokoi, Makoto Shimojyou: 3D-DDM: The direct deformation technique for free form curved surfaces by using three-dimensional B-Spline, IEEJ. pp.253-260, February, 1995 *(in Japanese)*.

[6] Kiyoshi Kiyokawa: VLEGO: The virtual modeler by using both hands, The master paper of NAIST, NAIST-IS-MT9451033, February, 1996 *(in Japanese)*.

[7] James M. Hebert: LIGHTWAVE 3D USER GUIDE , NewTek company *(in Japanese)*.

[8] Thomas W. Sederberg, Scott R. Parry: Free-Form Deformation of Solid Geometric Models, ACM SIGGRAPH Computer Graphics, pp.151-160, August, 1986.

[9] Michael Chen, S.Joy Mountford, Abigail Sellen: A Study in Interactive 3-D Rotation Using 2-D Control Devices, ACM SIGGRAPH Computer Graphics, pp.121-129, August, 1988.

[10] Robert C. Zeleznik, Kenneth P. Herndon, John F. Hughes: SKETCH: An Interface for Sketching 3D Scenes, ACM SIGGRAPH Computer Graphics, pp.163-170, August, 1996.

[11] Addison Wesley: OpenGL Programming Guide, Seiun company, 1995 *(in Japanese)*.

[12] Kenjiro Miura: Open GL 3D graphics primer, Asakura Books, 1995 *(in Japanese)*.

[13] Yasuhiro Aikawa: OpenGL Programming Guide Book, Katoh Bunmei company, 1995 *(in Japanese)*.