

# Claymore: Three-Dimensional Modeling Tool

**Jiro Tanaka**

University of Tsukuba  
1-1-1 Tennodai  
Tsukuba Ibaraki  
305-0006 Japan  
+81-298-53-5165

jiro@softlab.is.tsukuba.ac.jp

**Hideki Mitsunobu**

University of Tsukuba  
1-1-1 Tennodai  
Tsukuba Ibaraki  
305-0006 Japan  
+81-298-53-5165

copel@softlab.is.tsukuba.ac.jp

**Takashi Oshiba**

University of Tsukuba  
1-1-1 Tennodai  
Tsukuba Ibaraki  
305-0006 Japan  
+81-298-53-5165

ohshiba@softlab.is.tsukuba.ac.jp

## ABSTRACT

We describe a method for manipulating a three-dimensional object without considering conventional three independent orthogonal views. We apply the direct manipulation method to operate the three-dimensional object. We propose “**augmented manipulation**” method 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.

## KEYWORDS

3-D modeling, augmented manipulation, direct manipulation, mouse, interactive graphics, World Wide Web.

## 1 Problems of three-independent orthogonal views

Conventional three-dimensional modeling tools use three independent orthogonal views [3]. The three independent orthogonal views are used the method to operate a three-dimensional object. It is composed of three views. Each view is represented by two of 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. They cannot operate the object intuitively because of this reason.

## 2 Augmented Manipulation

Direct manipulation is the operation in which users’ operation directly invokes the reaction of the system [10]. We apply the direct manipulation technique to operate a three-dimensional object. Some works on editing a three-dimensional object by using the direct manipulation technique have already been proposed [11, 4].

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. We use the mouse as our input device instead of by 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 a 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.

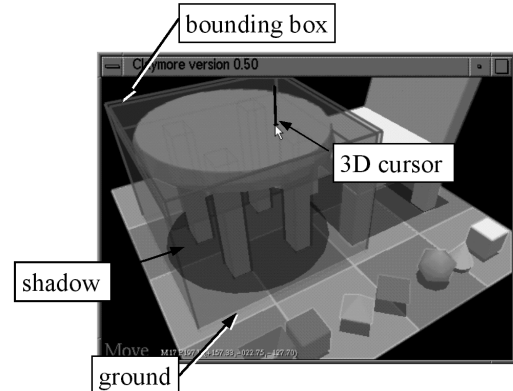


Figure 1: The additional information.

- 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 effective [5].

### 3 The concept of modeler “Claymore”

We implement three-dimensional modeling tool “Claymore” [6]. 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 the object is not empty.

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.

### 4 The modeling tool “Claymore”

“Claymore” is implemented with OpenGL [8, 7, 1]. It runs on Unix and Microsoft Windows.

We implement “augmented manipulation” for this modeler.

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 2).

We have implemented the following functions. They can select the buttons that are placed at the top of the window (see Figure 3). They can change a operation mode by using the buttons. These buttons show the following

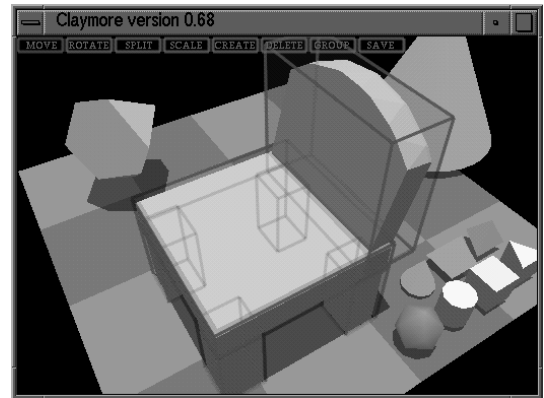


Figure 2: Screen shot of “Claymore”.

functions: Move, Rotate, Cut, Deform, Create, Delete, Group, and Save to VRML format.



Figure 3: Selecting buttons.

#### 4.1 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 4).

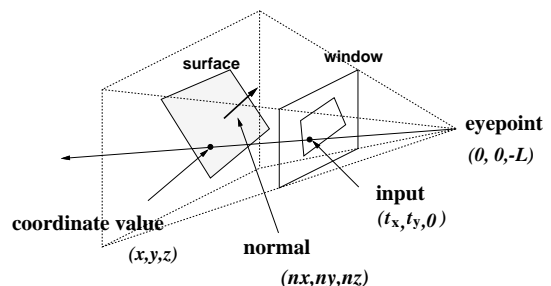


Figure 4: Specifying the coordinate values.

The system calculates the coordinate values of an intersection point between the following two elements (see Figure 4):

- (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. Let  $(0, 0, -L)$  be the position of the eye point. Let  $(n_x, n_y, n_z)$  be the normal vector of plane (2). The intersection point between the line (1) and the plane (2) is computed by:

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

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

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

## 5 Modeling processes

Users can construct a model by using the following processes.

- 1 Creating a fundamental object.
- 2 Constructing parts by cutting and deforming the object.
- 3 Arranging the parts by moving and rotating.
- 4 Combining the parts.
- 5 Saving the model.

### 5.1 Creating an object

We prepare the following objects as fundamental models: Sphere, Cylinder, Cone, Cube, Square pyramid, Triangular prism, Tetrahedron.

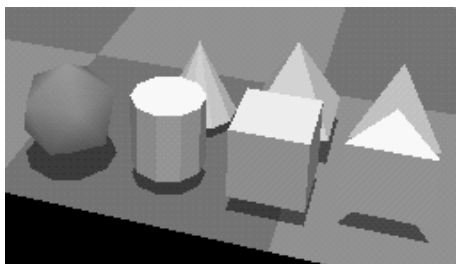


Figure 5: The fundamental models.

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.

### 5.2 Cutting 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 6). There are the three possibilities of the direction:

a — a surface.

b — an edge of the object.

c — a vertex of the object.

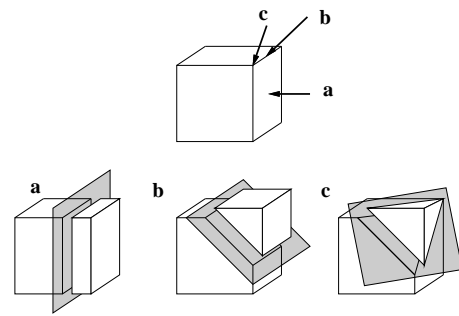


Figure 6: Three kinds of the cutter plane.

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

After the 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.

### 5.3 Deforming an object

The Free-Form Deformation (FFD) technique is the deformation method of an object [9].

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 7).

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.

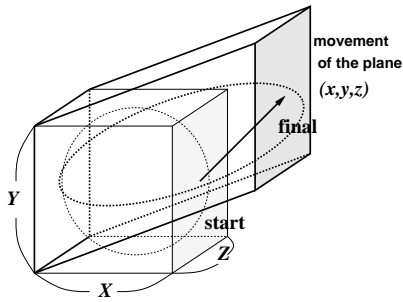


Figure 7: Simplified FFD.

#### 5.4 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 8 and 9). One is parallel to the ground. The other is vertical to the ground.

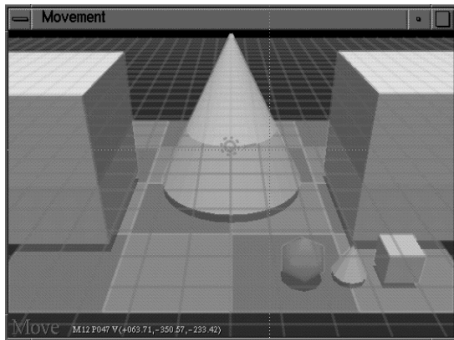


Figure 8: The plane that is parallel to the ground.

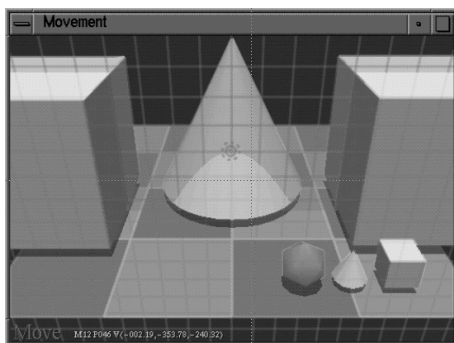


Figure 9: The plane that is vertical to the ground.

They can select the plane by using the left or right but-

ton 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 8). 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 9).

#### 5.5 Rotating an object

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

By using the conventional method of rotation [3, 2], 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 10).



Figure 10: A conventional rotation technique: The final position is different from the initial one.

The direct manipulation technique requires “reversible manipulation” [10]. 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 11: 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 12).

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  $\theta$  be the direction of the mouse movement (see Figure 12).  $r$  and  $\theta$  can be computed by:

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

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

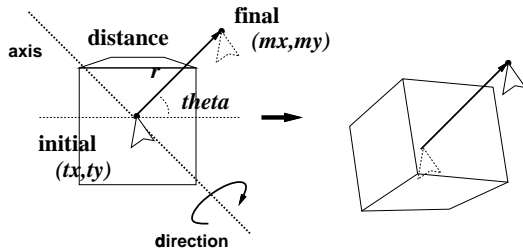


Figure 12: The rotation of the object.

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.

## 5.6 Grouping objects

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

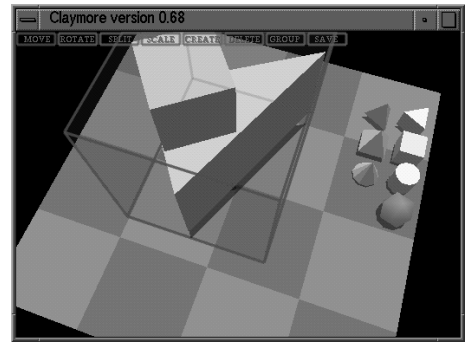


Figure 13: Grouping objects.

group of selected objects is constructed by combining the bounding boxes of each of the objects in the group.

The group of objects can be manipulated as one combined object.

## 5.7 Exporting to VRML

Users can store the modeling data to VRML format or alternately Wavefront OBJ format.

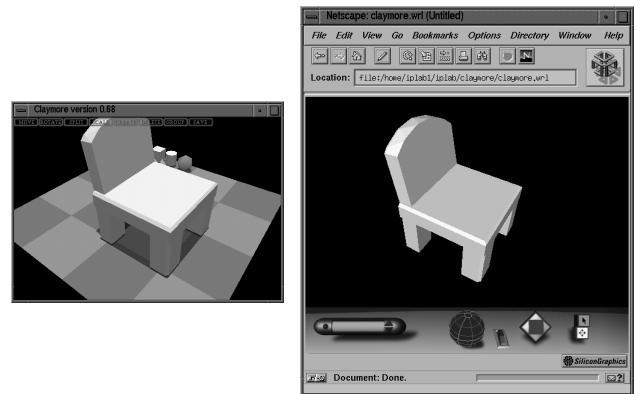


Figure 14: Saving to VRML format.

The model that is made by using “Claymore” can be exported to other three dimensional tools. They can make use of the modeling data while creating VRML web pages (see Figure 14).

## 6 Objects connected by joints

We developed a three-dimensional editor “Juggler” which extends the functions of “Claymore”.

“Juggler” provides *joints* through which two objects can be connected (see Figure 15). Once connected, the objects can be moved while the connected end of the objects remain fixed. Users can rotate the object along the *joint*. The motion is like the motion of a leg around the knee or the motion of an arm around the elbow.

Users can operate the objects by using “augmented manipulation”. We apply the rotation method of “Clay-

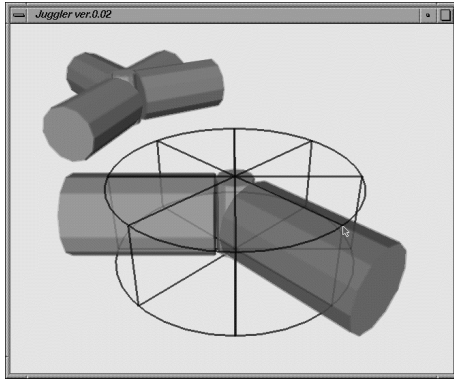


Figure 15: Screen shot of “Juggler”.

more” to rotate the objects connected by *joint*.

The method which manipulates the objects connected by *joint* can be used in “Claymore”.

## 7 Related work

Michael Chen et al. [2] 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 our approach, 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. [12] 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. Our modeler “Claymore” adopts a mouse as the input device. The object is in perspective because “Claymore” uses perspective transformation.

## 8 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 #09245203.

“Claymore” and the examples described here are fully implemented and can be obtained via WWW from the following URL.

<http://www.softlab.is.tsukuba.ac.jp/ip/lab/claymore/>

## REFERENCES

- [1] Yasuhiro Aikawa. *OpenGL Programming Guide Book (in Japanese)*. Katoh Bunmei company, 1995.
- [2] Michael Chen, S.Joy Mountford, and Abigail Sellen. A study in interactive 3-d rotation using 2-d control devices. In *ACM SIGGRAPH Computer Graphics*, pages 121–129, August 1988.
- [3] James M. Hebert. *LIGHTWAVE 3D USER GUIDE (in Japanese)*, 1996.
- [4] Kiyoshi Kiyokawa. VLEGO: The virtual modeler by using both hands (*in japanese*), February 1996.
- [5] Hideki Mitsunobu and Jiro Tanaka. The idea supporting system with three-dimensional view (*in japanese*). In *13th Conference Proceedings Japan Society for Software Science and Technology*, pages 357–360, September 1996.
- [6] Hideki Mitsunobu and Jiro Tanaka. “Claymore”: The three-dimensional modeling tool by using direct manipulation (*in japanese*). In *Workshop on Interactive Systems and Software*, page 212, December 1997.
- [7] Kenjiro Miura. *Open GL 3D graphics primer (in Japanese)*. Asakura Books, 1995.
- [8] Jackie Neide, Tom Davis, and Mason Woo. *OpenGL Programming Guide (in Japanese)*. Seiun company, 1995.
- [9] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. In *ACM SIGGRAPH Computer Graphics*, pages 151–160, August 1986.
- [10] Ben Shneiderman. *The principle of direct manipulation, The method to design a user interface (in Japanese)*, pages 126–161. Nikkei BP company, 1993.
- [11] Juli Yamashita, Yukio Fukui, Hiroshi Yokoi, and Makoto Shimojyou. 3d-ddm: The direct deformation technique for free form curved surfaces by using three-dimensional b-spline (*in japanese*). In *IEEJ.*, pages 253–260, February 1995.
- [12] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. Sketch: An interface for sketching 3d scenes. In *ACM SIGGRAPH Computer Graphics*, pages 163–170, August 1996.