GESTURE CONTROLLED OBJECT INTERACTION : A VIRTUAL TABLE CASE-STUDY

Oliver Bimber

Fraunhofer Institute for Computer Graphics
Division Rostock
Joachim-Jungius-Strasse 11
18059 Rostock
Germany
obimber@egd.igd.fhg.de

ABSTRACT

The metaphor introduced in this paper describes a more natural, human-like way to interact with objects in virtual environments. The interaction is controlled by motion-based gestures, which are recognized using the method described in [Bimbe98].

In contrast to the application of artificial and unnatural tools, natural input devices (e.g. the human hand) and human-like expression methods (e.g. gestures) can be used to interact with the objects; but on cost of precision. Pre-selection and artificial assistance (like displayed handles, etc.) become unnecessary or might be used in addition. Possible application scenarios are imaginable wherever an exact alternation of the objects is secondary (e.g. presentation, brain-storming, tactical/strategic planning, etc.), or if more precise methods are used in combination.

Three problems have to be addressed : First, the definition of natural, human-like gestures; second, the recognition of these gestures; and, finally, the association with appropriate interactions.

We focus on finding human-like, natural gestures (classified as *act gestures* by [Humme97]) which, in contrast to predefined symbolic commands (e.g. *emblems*, used in GIVEN [Böhm92]), don't have to be learned by the user, and combine them with appropriate actions to interact with the objects. This augments ideas expressed by [Bolt80,Latos97].

Keywords : gesture recognition, object interaction, virtual environments, natural, human-like

INTRODUCTION

Virtual Tables [Krüge95] (Virtual Planes, Virtual Workbenches, etc.) are table-like 3D output-devices that can be used (in contrast to the existing vertical projection metaphor) in a horizontal manner. In combination with stereoscopic projection and the use of shutter glasses [Barfi95,Burde94,Durla95, Young96], three-dimensional, virtual objects seem to grow out of the table into the surrounding space. With the support of head-tracking [Barfi95, Burde94,Durla95,Young96], the user is able to walk around the table and inspect the objects from different points of view. A connected graphics workstation calculates the displayed stereoscopic images, depending on the viewer's position and orientation, thus the user gets the illusion of moving around real objects that appear to be placed on top of a real table.

Beside head-tracking and the stereoscopic projection, innovative interaction techniques with these table-like output devices are a major part of research. Different application specific approaches were made.

For example, a new 2D device was developed by [Schle98] that enables the user to interact with objects by pointing to them. Position and orientation of a pistol-like input device are used to intersect a calculated projector with the x/y plane of the table. The resulting two-dimensional table-coordinates make a more precise (but unnatural) interaction with the objects possible (after selecting them).

Translation, for instance, is implemented according to a drag-and-drop metaphor. To realize rotation and scaling of pre-selected objects, supporting handles are displayed or the distance between the object's center of gravity and the current 2D pointer position is used to trigger a timer-based alternation.

Based on earlier ideas (e.g. from MIT's *Put-that-there System* [Bolt80] or research done by [Latos97]), we focus on pointing-movements (classified by [Latos97] as *deictic*) because they are basic elements in gestural human machine interaction.

Different pointing-movements were classified by [Gibet97] :

- *Static* : motionless
- *Straight* : gesture describes a straight line
- *Arc* : gesture describes an arc
- *Circle* : gesture describes a circle
- *Complex* : gesture describes a complex shape

THE METAPHOR

In terms to understand the proposed metaphor, imagine standing in front of a real table with real objects placed on top of it, and explaining the actions that you want to perform (e.g. selection, translation, rotation and scaling) to a second person without words (i.e. using gestures). To express that you want to move object *1* that is located at position *A* (e.g. the far/left corner of the table) to location *B* (e.g. somewhere in the center), naturally, you would first look and point to object *1* (at position *A*) and than look and point to position *B*. By doing this, you expect the other person to understand that you want to move the object from the far/left corner to the center of the table.

On the one hand it is clear that the target object is object *1*, because you pointed to it at the beginning. On the other hand it is definite that you want to move it to position *B* because of the rectilinear gesture, expressed by your hand (that points to position *B* at the end).

As in real life, the metaphor introduces a certain amount of inaccuracy. If, for example, another object (say object *2*) is located very close to object *1*, it becomes very difficult to identify whether object *1* or object *2* is meant. The user would have to move closer to the objects (i.e. the far/left corner), make use of a more precise metaphor or specify the object verbally to prevent ambiguity.

It is also difficult to tell the other person that you want to place object *1* exactly at the coordinates x/y, just by pointing. To achieve this in reality you would need the help of artificial tools, such as a ruler (i.e. a precise metaphor).

Inaccuracy is accepted for scenarios where precision is secondary and the user wants to interact with the objects on a quick, natural and human-like basis. Exact metaphors, however, can be used in combination and will not be substituted. Within the following sections, we will discuss examples how to implement the basic object interactions (i.e. translation, rotation, scaling, and selection) on Virtual Tables. Motion based gestures that are carried out are scanned and recognized by the method described in [Bimbe98].

OBJECT SELECTION

The most natural way to select an object that is located somewhere on the table plane is to look at it and point to it (*static pointing-movement*), or to encircle it (*circle pointing- movement*). To prevent inaccuracy that results of the noise received by the tracking device, we will discuss the second method (i.e. encirclement) as an example for object selection.

To select an object the user (who's eyes are located at position $\vec{P}_1$) looks at the object (who's center of gravity is located at position $\vec{P}_3$) and carries out a circle-like gesture that frames the object (e.g. with his tracked forefinger). To identify the object that is selected from the users perspective we have to find the object that intersects the projector which connects two the points : the user's eye position (i.e. $\vec{P}_1$) and the gesture's center of gravity ($\vec{P}_2$). If no intersection occurs, we have to find the object that is closest to the projector.

Figure 1 illustrates this.

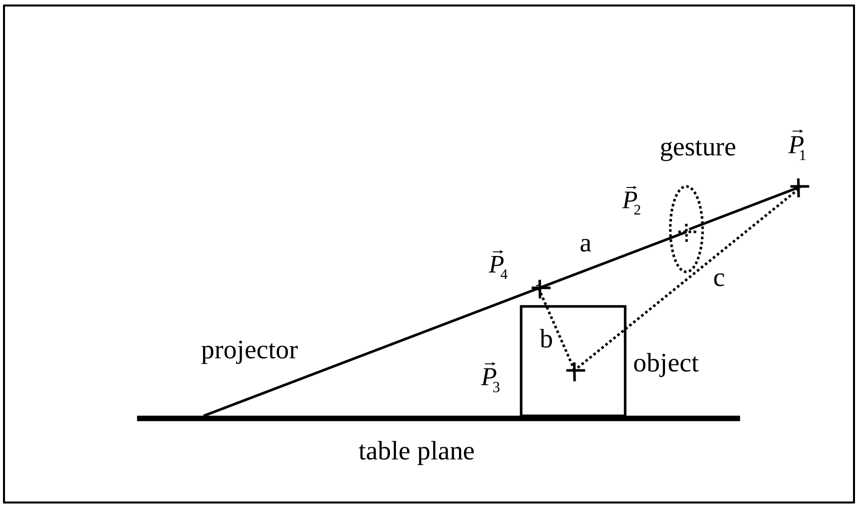

Figure 1 : Object Selection (Non-Intersection Case)

The projector is defined by :

$$\vec{P} = \vec{P}_1 + \lambda\left(\vec{P}_2 - \vec{P}_1\right)$$

Let $\left(\vec{P}_2 - \vec{P}_1\right) = \left[d_x, d_y, d_z\right]$ be the projector's direction, then the smallest distance from the object's center of gravity and $\vec{P}$ is defined by :

$$A = \left[(x_3 - x_1)\cdot d_y - (y_3 - y_1)\cdot d_x\right] \ ,$$
$$B = \left[(y_3 - y_1)\cdot d_z - (z_3 - z_1)\cdot d_y\right] \ ,$$
$$C = \left[(z_3 - z_1)\cdot d_x - (x_3 - x_1)\cdot d_z\right] \ ,$$

$$b = \sqrt{\frac{A^2 + B^2 + C^2}{{d_x}^2 + {d_y}^2 + {d_z}^2}} \ ,$$

$$\text{with } \vec{P}_1 = \left[x_1, y_1, z_1\right], \ \vec{P}_3 = \left[x_3, y_3, z_3\right]$$

$\vec{P}_4$ is the position on the projector that is closest to the object's center of gravity (i.e. where $\left|\vec{P}\right| = a$). We can calculate :

$$c = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2}$$

and apply Pythagoras' Theorem to derive :

$$a = \sqrt{c^2 - b^2} = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 - b^2}$$

Thus, we obtain :

$$\vec{P}_4 = \vec{P}_1 + a \cdot \frac{\vec{P}_2 - \vec{P}_1}{\left|\vec{P}_2 - \vec{P}_1\right|}$$

It is important that we first check if any object is intersected by the projector. To simplify the intersection-test in our example we use the object's bounding sphere with the radius r to detect whether an object is intersected or not.

Only if no object is intersected we choose the one that is closest to the projector (e.g. the object with smallest $b - r$) to be selected. If several objects are intersect, we choose the one that is closest to the user's eye-position (i.e. the object with smallest a) to be selected. We can simplify that an object is intersected if $b - r \leq 0$, since we are not interested in the exact point of intersection.

Note, that since we only compare distances, we do not have to calculate square-roots for a and b. This will speed up the intersection-test.

We can also define (if no intersection occurred) that an object is only selected if its ($b - r$) is minimal and lays below some threshold to exclude the situation that no object was framed by the selecting gesture.

OBJECT TRANSLATION

As described above, three pieces of information have to be extracted out of the gesture to translate object *1* from position *A* to position *B*. First, we have to recognize that we want to translate an object. This is specified by any straight gesture (*straight pointing-movement*) that is parallel to the table plane. First, we look and point to object *1* to indicate that this is our target object. Then we point to the position to which object *1* has to be moved. The motion that is carried out while pointing to the target object (at position *A*) and moving to the target position (position *B*) describes a straight translation gesture and will be recognized by the gesture recognition process [Bimbe98].

The second information needed is the target object (i.e. that we want to translate object *1*). In contrast to the selection-method that is described above, we use the gesture's start position (instead of its center of gravity) to build a projector (together with the

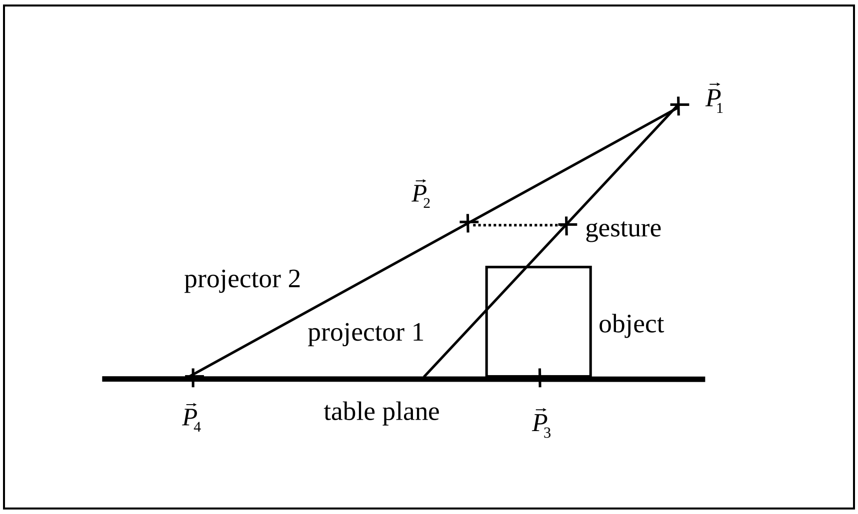

Figure 2 : Object Translation (Intersection Case)

user's eye position). As above, we use the projector as a reference to select an object. Once the gesture is expressed, we use the gesture's end position (again together with the user's current eye position) to build a second projector that is intersected with the table. The resulting x/y position on the table is the projected target location.
Figure 2 illustrates this.
$\vec{P}_4$ is the two-dimensional intersection of projector 2 on the table plane and can be calculated as follows :

$$\vec{P}_4 = \left[x_1 - \frac{z_1}{z_2 - z_1} \cdot (x_2 - x_1), y_1 - \frac{z_1}{z_2 - z_1} \cdot (y_2 - y_1) \right],$$

where

$\vec{P}_1 = [x_1, y_1, z_1]$ is the user's eye position,
$\vec{P}_2 = [x_2, y_2, z_2]$ the gesture's end position and
$\vec{P}_3 = [x_3, y_3]$ is the object's center of gravity (projected onto the table plane).

We can express the object's translation from $\vec{P}_3$ to $\vec{P}_4$ as triggered linear interpolation on the connecting straight line :

$$\vec{P} = (1-\lambda) \cdot \vec{P}_3 + \lambda \cdot \vec{P}_4, \text{ with } 0 \le \lambda \le 1$$

Note, that we can make λ dependent to the gesture's speed by defining a ratio between the gesture's total length and the number of sample positions that we are able to receive from the tracking device while the gesture is carried out. This will allow us to control the translation speed with the gesture's dynamics.

OBJECT SCALING

To express gesturally that an object has to be increased or decreased in size could be done as follows : Adjust the object with the eyes and the forefinger and then pull or push the object into (or out of) the table plane. This carries out a straight gesture (*straight pointing-movement*) that is perpendicular to the table plane and naturally represents scaling. The identification of the object to be scaled, is done exactly as for the translation case (using the gesture's start position as reference point to build the selection-projector). Once the gesture is carried out, the difference between the start and end point's z-components (Δz) indicates the scaling offset.
Figure 3 illustrates this.
$\vec{P}_2 = [x_2, y_2, z_2]$ is the gesture's start position and $\vec{P}_3 = [x_3, y_3, z_3]$ the gesture's end position. The scaling offset can be calculated with : $\Delta z = z_3 - z_2$.
We can again express the object's scaling from the old size (s_1) to the new size ($s_2 = s_1 + \frac{\Delta z}{2}$) as triggered linear interpolation :

$$s = (1-\lambda) \cdot s_1 + \lambda \cdot s_2, \text{ with } 0 \le \lambda \le 1$$

Note, that we have to update the object's z-position to make it stick to the ground.

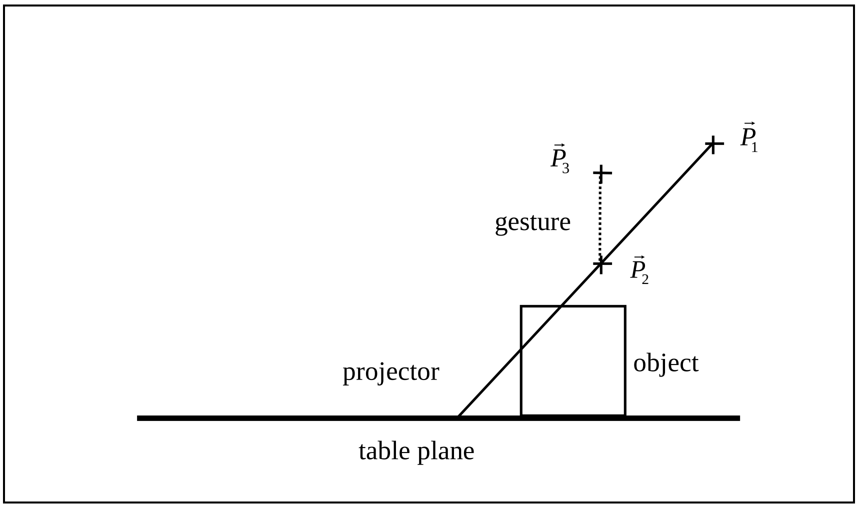

Figure 3 : Object Scaling

OBJECT ROTATION

To rotate an object around the z-axis we can adjust it with the eyes and the forefinger again, and then express a gesture that describes a semicircle (*arc pointing-movement*) in the direction we want to rotate it (clockwise or counterclockwise). We use the gesture's center of gravity and the user's eye position again to build the reference-projector that helps to identify the selected object. We can then intersect two new projectors that are given by the user's eye position and the gesture's start and end position with the table plane. The resulting two intersections ($\vec{P}_1, \vec{P}_2$), together with the object's projected center of gravity ($\vec{P}_3$), build the rotation-offset angle (α). Figure 4 illustrates this.

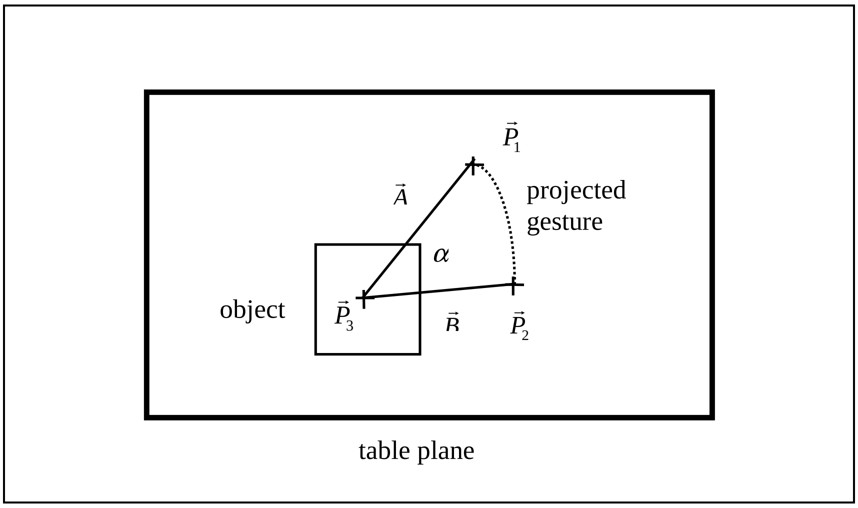

Figure 4 : Object Scaling

We can calculate the rotation-offset with :

$$\alpha = a\cos\left(\frac{\vec{A}\cdot\vec{B}}{|\vec{A}|\cdot|\vec{B}|}\right),$$

where $\vec{A} = \vec{P}_1 - \vec{P}_3$ and $\vec{B} = \vec{P}_2 - \vec{P}_3$.
Similar to scaling and translation we express the rotation as a triggered linear interpolation between the object's original rotation (r_1) and its new rotation ($r_2 = r_1 + \alpha$) :

$$r = (1-\lambda)\cdot r_1 + \lambda \cdot r_2, \text{ with } 0 \le \lambda \le 1$$

Note, that r has to be normalized to a range of $0 \le r \le 2\cdot\pi$.

AN EXPERIMENTAL SCENARIO

We have chosen a scenario for tactical and strategic planning to demonstrate the applicability of the proposed metaphor. Major aspects for selecting this kind of application were that it focuses on the exchange of strategic and tactical information and that a precise alternation of used objects is secondary.

The selected example implements a virtual basketball field that supports basketball-coaches in planning and revising game courses and moves etc.

Beside the virtual court, ten virtual players (divided into two teams) were implemented. These players are controlled by the user's (e.g. the coach's) gestural input only.

Five gestures support the interaction with the players.

A straight pointing-movement that is parallel to the table plane indicates that a player has to run from position *A* to position *B*. As described above (see *Object Translation*), we select a player by looking and pointing to him (i.e. to position *A*). Then we look and point to the target position (position B). The animation of a player who is running from location *A* to location *B* is achieved by a triggered linear interpolation (see above).

An arc pointing-movement is used to make a player turn in the specific direction (clockwise or counterclockwise). This was realized as discussed above (see *Object Rotation*).

To realize a pass from player *A* to player *B* we use the circle pointing-movement that was discussed above (see *Object Selection*). If we select player *B* by encircling him, player *A* will pass the ball to player *B*. Player *B* will be rendered semitransparent to indicate that he carries the ball until he passes it to another player.

Beside running, turning and passing, a player is able to jump. This is implemented using a complex gesture that is neither trivial straight nor trivial circular. A natural gesture that can represent jumping is a combination of two straight motions. The first is carried out away from the table plane, followed by the second straight motion with opposite direction (i.e. to the table plane). This symbolizes the vertical motion-path that the player follows during the jump. The jump itself is realized as triggered interpolation over the reverse square-function :

$$h = h_g + \left(1 - \lambda^2\right) \cdot \Delta h \text{, with } -1 \le \lambda \le 1 \text{,}$$

where h is the current height during the jump, h_g is the ground level and Δh the height of the jump.

Another complex gesture was defined to stop all actions that are currently performed by the players (e.g. some players can run, turn and jump, some can pass, etc. at the same time). A gesture that describes an '*X*' will freeze all actions. Note, that this is the only 'unnatural' gesture (*predefined symbolic command* [Humme97]) used for this example.

Figure 5 illustrates the experimental basketball scenario.

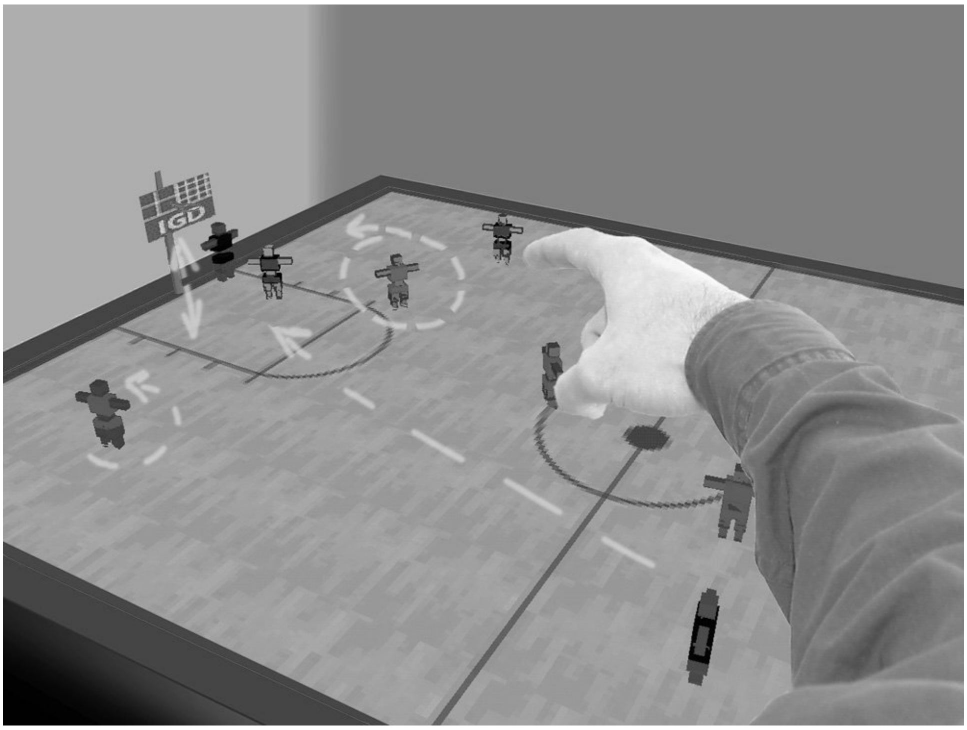

Figure 5 : The Virtual Basketball Field from the Coach's Perspective

Supported by these gestures, a coach is able to interact with several players simultaneously on a natural and human-like basis. This cannot be achieved by regular interaction metaphors.
On the other hand it is not possible, for example, to let a player run to a precisely specified location on the table (e.g. within a range of $\pm$ 1mm) or to turn him around by (for example) exactly 37.3°. This, however, is not necessary for this kind of application where precision is secondary.
The basketball experiment is just an example to demonstrate that a user-friendly, gesture-based interaction with objects is possible for specific scenarios. Many other applications are imaginable.

CONCLUSION

Even if exact object interaction is difficult, in combination with precise interaction tools, the proposed metaphor simplifies the work within virtual environments. If objects have to be transformed quickly and precision is secondary, the metaphor offers a more natural and human-like way of interaction than usual methods. If necessary, an exact alternation can be done in addition to guarantee accuracy.
The application of transformation rules (e.g. part *A* can only be attached to part *B* at a specific, known position and orientation of part *B*), or the combination with speech recognition (multi-modal [McKen97]) restricts the object's transformation possibilities and increases the metaphor's reliability and effectiveness.
The *virtual basketball experiment* has clearly demonstrated that a more user-friendly and simultaneous object interaction is possible, using the proposed metaphor.
Note, that the method described in this paper to perform the intersection-test (bounding sphere) is just an approximation and was chosen to simplify the introduction of the new metaphor. For real applications better methods should be used (e.g. bounding box or surface intersection).
Examples for suitable applications are presentation, tactical/strategic-planning, brainstorming, etc., and other scenarios that combine the metaphor with precise transformation methods or/and transformation rules.

REFERENCES

Barfield, W., Furness, T.A. 1995 *Virtual Environments And Advanced Interface Design.* Oxford University Press New York.

Bimber, 0. 1998 *Continuous 3D Gesture Recognition: A Fuzzy Logic Approach.* Fraunhofer Institute for Computer Graphics. Technical Report Log: 98i013-FEGD.

Böhm, K., Hübner, W., Väähänen, K. 1992 *GIVEN: Gesture Driven Interactions in Virtual Environments – A Toolkit Approach to 3D Interactions*. Interface to Real and Virtual Worlds-Conference, Montpellier, France.

Bolt, R.A. 1980 *Put-that-there: Voice and gesture at the graphics interface*. ACM SIGGRAPH-Computer Graphics, ACM Press New York.

Burdea, G., Coiffet,P. 1994 *Virtual Reality Technologies*. John Wiley and Sons New York.

Durlach, N.I., Mavor, A.S. 1995 *Virtual Reality Scientific and Technological Challenges.* National Academy Press Washington D.C.

Hummels, C., Smets, G., Overbeeke, K. 1997 *An Intuitive Two-Handed Gestural Interface for Computer Supported Product Design.* International Gesture Workshop, Bielefeld, Germany.

Krüger, W., Bohn, C.A., Fröhlich, B., Schütte, H., Strauss, W., Welsch, G. 1995 *The responsive Workbench: A Virtual Work Environment*. IEEE Computer Society, IEEE Computer, Vol. 28, No. 7.

Latoschik, M.E., Wachsmuth, I.. 1997 *Exploiting Distant Pointing Gestures for Object Selection in Virtual Environments*. International Gesture Workshop, Bielefeld, Germany.

Gibet, S., Richardson, J., Lebourque, T., Braffort A. 1997 *Corpus of 3D Natural Movements and Sign Language Primitives of Movements*. International Gesture Workshop, Bielefeld, Germany.

McKenzie, K., Mills, Alty, J.L. 1997 *Investigating the Role of Redundancy in Multimodal Input Systems*. International Gesture Workshop, Bielefeld, Germany.

Schlempp, A. 1998 *ViComp – Urban Design Using Virtual Modeling and Composition on the Virtual Table*. CG Topics 3/98.

Youngblut, C., Johnson, R.E., Nash, S.H., Wienclaw, R.A., Will, C.A. 1996 *Review of Virtual Environment Interface Technology*. Institute of Defense. Technical Report IDA Paper P-3186, Log: H96-001239.