Utilize Speech and Gestures to Realize Natural Interaction in a Virtual Environment

Marc Erich Latoschik, Martin Fröhlich, Bernhard Jung, Ipke Wachsmuth
Technische Fakultät, AG Wissensbasierte Systeme
University of Bielefeld
P.O. Box 100131, 33501 Bielefeld
{marcl,martinf,jung,ipke}@techfak.uni-bielefeld.de

Abstract – **Virtual environments are a new means for human-computer interaction. Whereas techniques for visual presentation have reached a high level of maturity in recent years, many of the input devices and interaction techniques still tend to be awkward for this new media. Where the borders between real and artificial environments vanish, a more natural way of interaction is desirable. To this end, we investigate the benefits of integrated speech- and gesture-based interfaces for interacting with virtual environments. Our research results are applied within a virtual construction scenario, where 3D visualized mechanical objects can be spatially rearranged and assembled using speech- and gesture-based communication.**

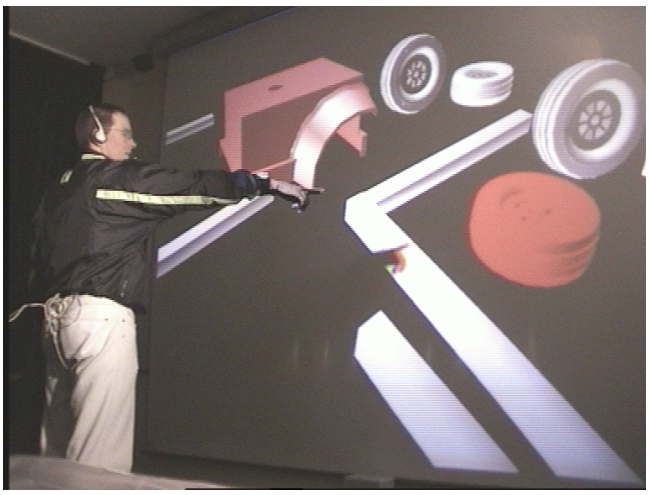

Fig. 1: A user partly immersed in a virtual environment; he is located in front of a large screen display and interacts with the system by selecting an object using a pointing gesture.

I. INTRODUCTION

Virtual environments are artificial surroundings created using virtual reality (VR) techniques. Whereas the latter term is often associated with it's enabling hardware, e.g. input devices like *cyber* gloves or head mounted displays (HMDs), or the generation of realistic real time computer graphics to visualize such an alternative environment, our research interest concentrates on a different, more abstract aspect: To experience the virtual space as a new media and achieving a new form for human-computer interaction. From the given facets of VR, the visualization and the simulation of environments are more likely to approach the essence of virtual reality, namely to give a user the feeling of a total immersion in a virtual scene; to accept the synthetic images as an extension of the real world where the user can act in a natural way. To enable interaction with such a system that embeds the user in an artificial surrounding, significant differences to the use of standard computer interfaces have to be taken into account. In VR, the scene is typically presented on either a stereoscopic head mounted device, or on large screen projection walls or caves [4]. In both cases all or most parts of the user's field of view are covered by the virtual scene. Also, in contrast to conventional desktop computer systems, the user is no longer "tied" to a relativley small area in front the system, but instead has the freedom to freely move around when inspecting the visualized scene.

But how can we operate these kinds of systems? Under the described circumstances mice and keyboards seem to be more a hindrance than a help. Obviously a different method for interaction is desirable. Our work aims at building speech-supported gesture interfaces for interacting with virtual environments. To demonstrate our approach, we develop a system that allows manipulation of a virtual construction scene projected on a large screen (the *wall*) with a user located in front of this display (Fig. 1). Natural language and gestures are used to communicate desired scene changes.

II. FUNCTIONAL ROLES OF GESTURES

A basic problem while dealing with "gesture" is the lack of a single generally accepted definition of the term. Therefore we present a functional classification of "open-hand" gestures, that is gestures of the upper limbs without tool usage.

Human gesture serves different functional roles [3]:

- The semiotic function of gesture is to communicate meaningful information. The structure of a semiotic gesture is conventional and commonly results from shared cultural experience.
- The ergotic function of gesture is associated with the notion of work. It corresponds to the capacity of humans to manipulate the real world, to create artifacts, or to change the state of the environment by "direct manipulation". Shaping pottery from clay, wiping dust, etc. result from ergotic gestures.
- The epistemic function of gesture allows humans to learn from the environment through tactile experience. By moving your hand over an object, you appreciate its structure, you may discover the material its made of, as well as other properties.

The ergotic and epistemic functions of gesture are not the primary concern of our work, but are promising to lead to future tools in virtual reality (VR) and human computer interaction (HCI) techniques. Furthermore, the semiotic function of gesture has been studied extensively, e.g. [5], [14], and [10], and, with special emphasis on HCI in [13], [11], [16]. MIT's Advanced Human Interface Group [19] describes a subdivision of the semiotic function which leads to our own operational model for machine gesture recognition that has described more thoroughly in [12].

For the purposes of this paper, we focus on semiotic, or "communicative" gestures and, among those, concentrate on deictic (i.e. pointing at an obejct or area) and mimetic gestures (i.e. using the upper limbs as place-markers for actions or behaviours of objects or states).

III. INTERACTING USING GESTURES

A closer look at the core work flow in virtual construction systems, e.g. in a 3D-visualization based CAD application, reveals two major tasks. Besides all the functionality related to work-organisation or visual presentation (like file I/O, data input or window-management), objects which a user wants to manipulate or assemble, have to be identified and distinguished from other parts in the virtual scene. This selection task precedes any further operation. Then the desired modification function has to be enabled and applied to the selected part(s). In a virtual construction environment, modification of the scene objects' position and orientation is of primary importance. Therefore the first functions we have to establish with the help of gestures shall allow us to manipulate the spatial representation of virtual objects. Simply spoken, we need to

1. select objects or locations
2. manipulate objects

and incorporate gesture-based input to accomplish this.

The next sections investigate both tasks more closely. The selection problem is referenced by exploiting (deictic) distant pointing gestures. For the manipulation task, we examine movements of a user's upper limbs that mimic the desired manipulations, hence the mimetic properties of gestures [12]. Also, we will distinguish two different characteristics of gestures for interaction, namely the qualitative and the quantitative aspects. Different approaches were developed to qualitatively recognise and identify gestures (e.g. from sign languages) and to apply the results to virtual reality [16] [1] [7]. We also evaluate the quantitative properties to use them for a manipulation task. For the technical detection task, we use specific hardware devices. The necessary datasets for the analyzation process are collected via electro-magnetic position and orientation sensors, and the actual hand shape data is gathered with a *cyber* glove. For the remaining sections we assume that a 6DOF (degree of freedom) sensor is mounted on the users hand-wrist. In collaboration with other research groups we work on a further enhancement of the input devices by incorporating camera-based position detection [15].

A. Pointing for Object Selection

The task of selecting objects can be seen as a specialization of a more general problem. How can we reference not only objects, but also exact places in space by pointing to a location? In [12] we presented a system that is capable of detecting a pointing gesture based on a hand shape, shape movement and shape acceleration. With this information we can determine the exact time when a pointing gesture reaches its climax, and we can approximate the spatial direction of the pointing gesture using the data from the wrist-mounted sensor.

Depending on the calibration and based on a 100Hz detection rate, for a single pointing gesture about 1 to 20 pointing events are triggered and analysed. This analysis involves two steps. First, for every pointing event we have to map the two coordinate systems, the real world one where the pointing takes place and the virtual one where we want a concrete reference to be resolved. Based on a sensor mounted on the pointing hands wrist, standard geometric transformations are applied to an imaginary ray sent out from this sensor, approximating the direction of the pointing hands index finger in the virtual space. Now we detect collisions between these rays and objects in the virtual space. Fig. 3 illustrates our technical realization of this procedure, the flow of hypotheses between distributed modules in an agent based system [12]. This selection operation, called *picking*, involves some more heuristics than to analyse a simple mouse click. When we use pointing gestures instead, we usually do not have a graphical representation like a mouse-pointer that informs us about the current interaction position. Without this help, our results show that when the pointing

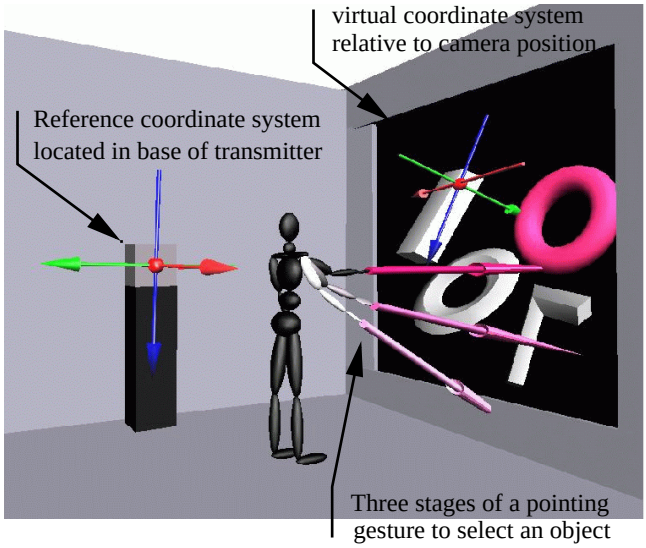

Fig. 2: A schematic view of the virtual construction scenario setup. It shows the different reference systems as well as three stages of a pointing gesture to point to a virtual wheel.

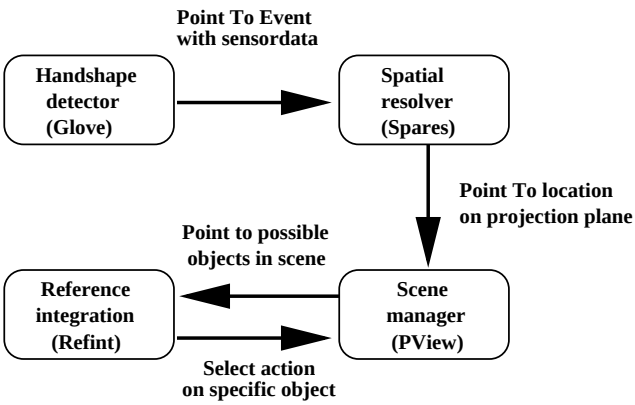

Fig. 3: The traversal of hypotheses between the different agents after a triggered pointing event.

is coarse, it leaves us with an unresolvable reference. One solution to this problem will be to gradually widen the radius of the picking rays. But obviously this will not be the best choice in every situation. What happens when a user did not reference an object but a location in space? In this case an answer can only be found by the support of information from other input sources, mainly the actual context of the operation and the users verbal expressions during the pointing.

In addition to the coarse-picking problem, one well-identified problem in gesture detection is to find the gesture's segmentation [8], in other words to determine the exact time when a gesture starts and ends. Our approach takes dynamics into account but does not try to find exactly one gesture. In contrast, we generate hypotheses according to the events of exceeding certain thresholds in the low-level detectors. This information is accumulated in integration modules where hypotheses from other detectors and sources (see section V.-A.) are evaluated concurrently [12].

IV. GESTURES FOR SPATIAL MANIPULATION OF OBJECTS

After an object has been selected – using pointing gestures alone as described above, or by a combination of speech and gestures, as described below –, the next step of the manipulation task involves some spatial transformation of the selected object. The basic manipulations we can apply to the visualized objects include geometric rotation, translation and scaling. Currently we concentrate on the first two operations and restrict manipulations to the relative position and orientation of the scene objects, but in contrast to other approaches, e.g. in the *Put that there System* [2], we allow a gradual manipulation of the objects and give a constant visual feedback during the manipulation process. The last one of the three operations, scaling, modifies the objects themselves. In this case different approaches seem to be useful, e.g. the detection of combined two-handed gestures might be appropriate, and we plan to address this in future work.

A. Object Rotation

There are several possibilities to rotate an object in the virtual space. In mouse-based CAD systems, special menus could be enabled to inform the system about the following intended action. As a result of this operation, often special *handles* around the object are activated as a visual feedback, and to inform the user about the current system status. Another method is to enable special graphical tools, e.g. *sliders*, and allow object rotation by operating them. This type of interaction could be mirrored in the same way with only using menus projected into the virtual space and the pointing gesture as a 3D mouse. In our approach, we follow a different idea, that is, we take the mimetic aspect of human gesturing into account and try to detect if the user's upper limbs move in a way that can be evaluated as a rotation. Basically, a user can simulate a rotation with a combined hand/arm movement in two ways. The first is to describe a rotation by *drawing circles in the air* (Fig. 4). Whereas this type of gesturing can be performed continuously, the second type, twisting the hand and forearm around the forearm axis (like changing a light bulb), is different: Start at one position, twist until the maximum angle limit is reached, and then reset to the start position and repeat these steps until a desired position is reached. Both differ significantly from each other, but they both have the problem of determin the axis of rotation and the rotation speed in common. The difference lies only in detecting the initial direction of rotation. Besides the described combined arm/hand motions, mimetic rotation can also be done without involving the forearm, hence to move only the index finger or the hand in the base joint, leaving the forearm position unchanged. Due to our data-sensor position on the hand's wrist-base (aligned with the forearm), we start with detecting combined forearm/hand rotations of type one as

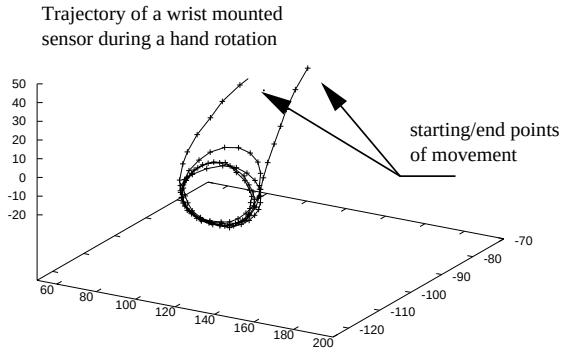

Fig. 4: A data plot of a hand rotation movement (the vertical axis is reversed to achieve a positive justification of the plot).

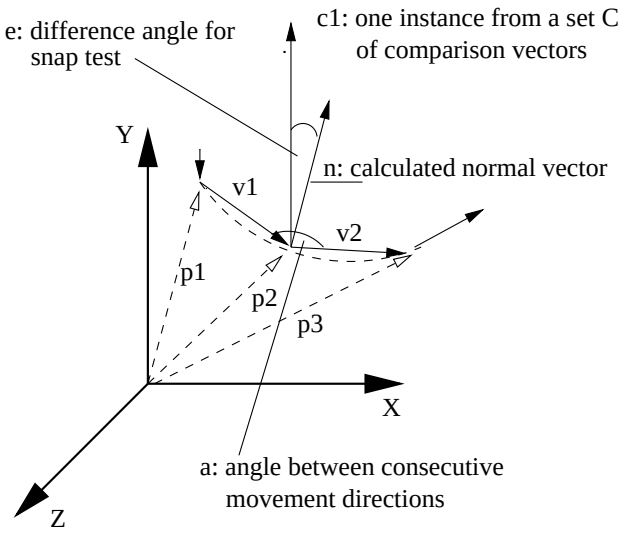

Fig. 5: Basic geometric calculations and comparison test for one data sample do determine the angle between the normal and the members of a set of comparison vectors.

described above. In this case the positional sensor data is of major interest, whereas for the twisting motions of type two, it is basically the rotational part.

A closer look at the position-data output produced by one wrist-sensor during a rotation illustrates the basic rotation features and the contiguities between different sample records. In Fig. 4 we can see that the sensor (and therefore the wrist) describes more or less perfect circles around that y-axis, except during the start/end phases. To gather quantitative information for one time step we use two consecutive datasets. Given three points $\vec{p1}$, $\vec{p2}$ and $\vec{p3}$ or their link vectors $\vec{v1}$ and $\vec{v2}$ we can calculate the resulting plane and normal $\vec{n}$. Fig. 5 illustrates the necessary operations. Between every three points we calculate the resulting normal vector, hence the possible axis of a rotation, and the angle velocity using $\vec{v1}$ and $\vec{v2}$. The inspection of the raw data plot reveals that it is very unlikely that two consecutive normals are parallel to each other due to raw data noise and erratic user motion. To enable a smooth system operation and to restrict the possible generated hypotheses we compare all resulting normals to a set C of comparison vectors (Fig. 5) that predetermines the possible rotations. These rotation-axes are computed from different reference frames in virtual construction:

1. rotation-axes defined through user (camera) view
2. object-intrinsic rotation-axes
3. rotation-axes defined through part mating constraints

In the current system we take all of the above reference systems, hence their three main orthogonal axes, into account. To achieve even higher manipulation freedom, we plan to use a more granular partition of the possible rotation space, e.g. to divide each octant with its diagonal. Finally, when the test against one of the reference vectors indicates that the specific angle is smaller than a defined threshold, this concrete reference vector becomes the resulting rotation axis.

B. Object Translation

Object translation can be achieved with a similar approach to the rotation detection technique. In fact the raw data exhibits that in contrast to the rotation test, we can use just the link vectors $\vec{v1}$ or $\vec{v2}$ (Fig. 5) and use them for testing against the reference systems. To avoid a possible translation for each emitted line vector, we test if consecutive vectors $(\vec{v1}, \vec{v2}, ..., \vec{vN})$ are almost parallel. Only then we assume that the user describes a translation with his hand. To gather information about the desired speed of the operation, in contrast to calculate the angle velocity during rotation, here the ratio (vector length)/(sample rate) is taken into account.

V. COMBINING SPEECH AND GESTURE

Enhancing gesture-based interfaces with parallel speech input makes human-computer-interaction more natural and powerful. For example, some scene manipulations might be easily communicated using language, but awkward or just impossible using gestures. Another reason to combine speech and gesture is that – if speech and gesture recognition are performed in an integrated fashion – information from the speech channel can be used to make gesture recognition more robust and efficient. The gesture detection examples from the preceding sections should clarify this point: Using the described methods, a big amount of hypotheses about detected gestures is generated, e.g. in the case of pointing detection about 1 to 20 hypotheses

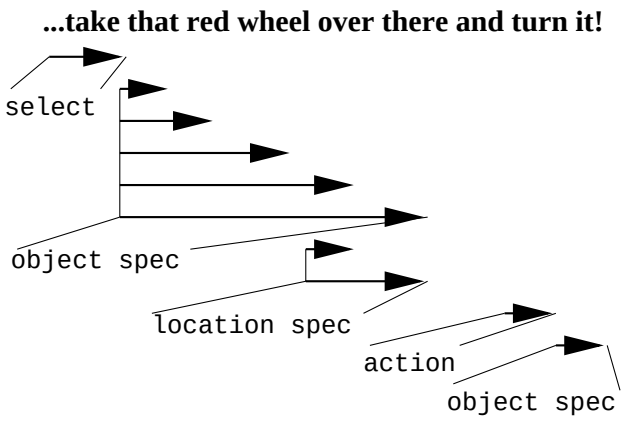

Fig. 6: Simple word spotting. After words are recognized, hypotheses based on their possible lexical categories are generated.

are proposed (see section III.-A.). Most of these hypotheses do not correspond to the intended, "communicative" gesture of the user but emerge during the "non-communicative" natural body movement. Many of the latter hypotheses are readily discarded, if information from the speech channel is accounted for in gesture recognition. For example, pointing gestures will typically occur just before, or (approximately) in parallel to the verbal utterance of certain key words such as *"this"*, *"that"*, or *"the"*. Similarly, other verbal phrases, e.g. *"red wheel"*, further narrow down the set of candidates for object selection.

A. Keyword Spotting for Fast Speech Analysis

Particularly in VR, smooth system operation and seamless human-computer-interaction are important factors for the user's acceptance of, and immersion into, the virtual scene. Hence we need all clues about the user's intention from the speech channel as soon as possible. Thus, instead of performing a time-consuming deep syntactic analysis of spoken input, we employ a more simple, but faster key word spotting approach. Right after the speech recogniser [6] delivers a word hypothesis, it is further classified according to its possible functions in the multimodal (speech and gesture) utterance. For example, a word may be part of phrases specifying an object, a location, or an action. As a result of the word classification process, further hypotheses are generated in a format similar to gesture hypotheses. Further more, the contextual information about word order is exploited to strengthen or weaken word hypotheses: As soon as a word is classified in any way, it is compared to the lastly emitted hypothesis. If both belong to the same class, often the new word specifies the last guess in a concrete manner. As a result the new information (e.g. the object's color) is added to the last estimation and amplifies it to build a reinforced hypothesis. Fig. 6 illustrates how successive hypotheses (represented by the horizontal arrows) are produced.

VI. VIRTUAL CONSTRUCTION

Our approach of combining gesture and speech is applied for interacting with a virtual assembly environment developed in our group, the CODY Virtual Constructor [18, 9]. The Virtual Constructor enables the knowledge-based simulation of various assembly-related operations over 3D-visualized parts, such as assembly, dissassembly, and modification of aggregates along transformational degrees of freedom of object connections. Whereas in the desk-top version of the Virtual Constructor, typed verbal input and direct manipulation with the mouse are used as means for human-computer-interaction, the speech/gesture-based interface described above is employed for user interaction in front of a large-screen display (Fig. 1).

In the virtual assembly application, typical multimodal utterances look like the following:

- *"attach <pointing gesture> this wheel to the cart!"*
- *"now unmount it and place it over <pointing gesture> there"*
- *"turn the stearing gear <rotation gesture> a little bit to the right!"*

As these examples show, user instructions refer to task-specific object manipulations such as assembly or dis-assembly but no more to purely geometric transformations, such as translation and rotation of objects. The Virtual Constructor is equipped with assembly-task knowledge that can be exploited when processing speech- and gesture-based instructions. For example, there might be only one way of mounting a particular wheel on an axle. Since the system has knowledge about these objects' only mating possibility, the user must not specify the exact target location of that wheel in his instruction. Or, more generally, in many cases user instruction must only specify which objects are to be selected, whereas system knowledge is exploited for an automatic manipulation of objects. Similarly, in other instructions where gestures are used as (partial) specification of object manipulations, e.g. rotation of parts w.r.t. other parts of an assembly, assembly constraints about legal relative movement of parts can be exploited for further interpretation of the instructions.

VII. CONCLUSION

In this paper, we introduced an approach to utilize a speech supported gesture interface for driving a knowledge-based virtual assembly application projected onto a large-size display. Two kinds of communicative gestures are so far considered in our approach: deictic (pointing) gestures for selection of objects and locations, and mimetic gestures, such as hand or finger rotations, that contribute to the specification of object manipulations. The current implementation comprises recognizers for both kinds of gestures. We have

further implemented a keyword-spotting approach for fast analysis of speech input. Current work includes the integration of all these subsystems into a single demonstrator system.

VIII. ACKNOWLEDGMENT

This research was partly supported by the Ministry of Science and Research (MWF) of the Federal State North Rhine-Westfalia in the framework of the collaborative effort "Virtual Knowledge Factory", and the Graduate College "Task Oriented Communication" of the German National Science Foundation (DFG).

VIII. REFERENCES

[1] K. Böhm, W. Broll, and M. Sokolewicz. Dynamic gesture recognition using neural networks; a fundament for advanced interaction construction. In S. Fisher, J. Merrit, and M. Bolan, editors, *Stereoscopic Displays and Virtual Reality Systems, SPIE Conference Electronic Imaging Science & Technology*, volume 2177, San Jose, USA, 1994.

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

[3] J. L. Crowley and J. Coutaz. Vision for man machine interaction. In *Proceedings of Engeneering Human Computer Interaction*, pages 28–45, Grand Targhee, USA, Aug. 1995. EHCI'95, Chapman and Hall, Ltd.London, UK. WWW-page: `http://pandora.imag.fr/ECVNet/IRS95/article.html`.

[4] C. Cruz-Neira, D. Sandin, and T. DeFanti. Surround-screen projection based virtual reality: The design and implementation of the cave. In *Computer Graphics Proceedings, Annual Conference Series 1993*, pages 135–142. ACM SIGGRAPH, 1993.

[5] D. Efron. *Gesture and Environments.* King's Crown Press, Morningside Hights, New York, 1941.

[6] G. A. Fink, C. Schillo, F. Kummert, and G. Sagerer. Incremental speech recognition for multimodal interfaces. In *This Volume*. 1998.

[7] M. Fröhlich and I. Wachsmuth. Gesture recognition of the uppler limbs: From signal to symbol. In Wachsmuth and Fröhlich [17], pages 173–184.

[8] P. A. Harling and A. D. N. Edwards. Hand tension as a gesture segmentation cue. In P. A. Harling and A. D. N. Edwards, editors, *Process in Gestural Interaction: Proceedings of Gesture Workshop '96*, pages 75–87, Berlin Heidelberg New York, 1997. Dep. of Computer Science, University of York, Springer-Verlag.

[9] B. Jung, M. Latoschik, and I. Wachsmuth. Knowledge-based assembly simulation for virtual prototype modeling. In *This Volume*, 1998.

[10] A. Kendon. Current issues in the study of gestures. In J.-L. Nespopulous, P. Rerron, and A. Lecours, editors, *The Biological Foundations of Gestures: Motor and Semiotic Aspects.* Lawrence Erlbaum Associates, Hillsday N.J., 1986.

[11] G. Kurtenbach and E. A. Hulteen. *Gestures in Human-Computer Communication*, chapter Technique and Technology, pages 309–317. Addison-Wesley, Reading, MA, USA, Jan. 1990.

[12] M. E. Latoschik and I. Wachsmuth. Exploiting distant pointing gestures for object selection in a virtual environment. In Wachsmuth and Fröhlich [17], pages 185–196.

[13] P. Maes. Alive: An artificial live interactive video environment. In T. E. Linehan, editor, *Computer Graphics Visual Proceedings*, Annual Conference Series, page 189, New York, NY 10036, USA, 1993. ACM Press, ACM SIGGRAPH.

[14] D. McNeill. *Hand and Mind: What Gestures Reveal about Thought.* University of Chicago Press, Chicago, 1992.

[15] C. Nölker and H. Ritter. Illumination independent recognition of deictic arm postures. In *This Volume*. 1998.

[16] K. Väänänen and K. Böhm. Gesture driven interaction as a human factor in virtual environments - an approach with neural networks. In *Virtual Reality Systems, conference proceedings.* British Computer Society, Academic Press, 1992.

[17] I. Wachsmuth and M. Fröhlich, editors. *Gesture and Sign-Language in Human-Computer Interaction: Proceedings of Bielefeld Gesture Workshop 1997*, number 1371 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg New York, 1998. Springer-Verlag.

[18] I. Wachsmuth and B. Jung. Dynamic conceptualization in a mechanical-object assembly environment. *Artificial Intelligence Review*, 10(3-4):345–368, 1996.

[19] A. D. Wexelblat. An approach to natural gesture in virtual environments. *acm Transactions on Computer-Human Interaction*, 2(3):179–200, 1995.