A General Framework for Multimodal Interaction in Virtual Reality Systems: PrOSA

Marc Erich Latoschik (marcl@techfak.uni-bielefeld.de)
AI & VR Lab[6], Faculty of Technology
University of Bielefeld, Germany

Abstract

This article presents a modular approach to incorporate multimodal – gesture and speech – driven interaction into virtual reality systems. Based on existing techniques for modelling VR-applications, the overall task is separated into different problem categories: from sensor synchronisation to a high-level description of cross-modal temporal and semantic coherences, a set of solution concepts is presented that seamlessly fit into both the static (scenegraph-based) representation and into the dynamic (renderloop and immersion) aspects of a realtime application. The developed framework establishes a connecting layer between raw sensor data and a general functional description of multimodal and scene-context related evaluation procedures for VR-setups. As an example for the concepts, their implementation in a system for virtual construction is described.

1 Introduction

The development of new interaction techniques for virtual environments is a widely recognized goal. Multimodality is the keyword that suggests one solution for getting rid of the WIMP[1]-style point-and-click metaphors still found in VR-interfaces. The more realistic our artificial worlds become, the more seem our natural modalities gesture and speech to be the input methods of choice, in particular when we think in terms of *communication* and further the possible incorporation of lifelike characters as interaction mediators. Our goal is the utilisation of multimodal input in VR as an spatial represented environment. Considering the latter, Nespoulous and Lecour [4] proposed a gesture classification scheme that conveniently describes possible *coverbal* gesture functions when they specified *illustrative* gestures as:

- *Deictic*: Pointing to references that occur in speech by respective lexical units.
- *Spatiographic*: To sketch the spatial configuration of objects refered to in speech.
- *Kinemimic*: To picture an action associated with a lexical unit.
- *Pictomimic*: Describing the shape of an object refered to in speech.

We are exploiting these gesture types with slight adaptions for enabling basic multimodal interaction.

Work is done on both sides, on the developement of multimodal interpretation and integration (MMI) and on the enhancement of VR-technology and realism. Despite this fact, there are few approaches that deal with the systematic integration of the MMI-results under general VR-conditions. The latter justifies the foundation for the developement of the PrOSA (Patterns On Sequences of Attributes)[3] concepts as fundamental building blocks for multimodal interaction in VR as described in this paper.

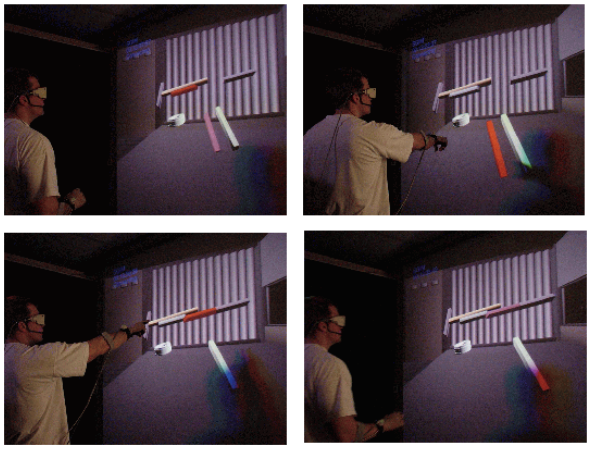

Figure 1: Multimodal interaction: a user speaks and gestures to achieve a desired interaction, in this case the connection of two virtual objects.

2 Gesture processing

Gesture detection heavily depends on sensor data which – in general – is neither synchronised nor represented with respect to a common base. There is no agreement about the bodydata representation which is suitable in gesture detection tasks:

[1] WIMP: Windows, Icons, Menu, Pointing

depending on the detection framework, it is often necessary to abstract from specific numeric sensordata (e.g., the position of one or several 6DOF sensors or the output of camera-based systems) and to consider relevant quantified movement information: static and dynamic atributes like *fingerstretching*, *hand-speed* , *hand-head-distance*, etc. , which PrOSA encapsulates in **attribute-sequences**, containers that establish a data flow network between a hierachy of different modular calculation components which are necessary for gesture analysis and detection. A schematic overview of the concepts and their cooperation in the network is shown in figure 2. The basic components are described in this section.

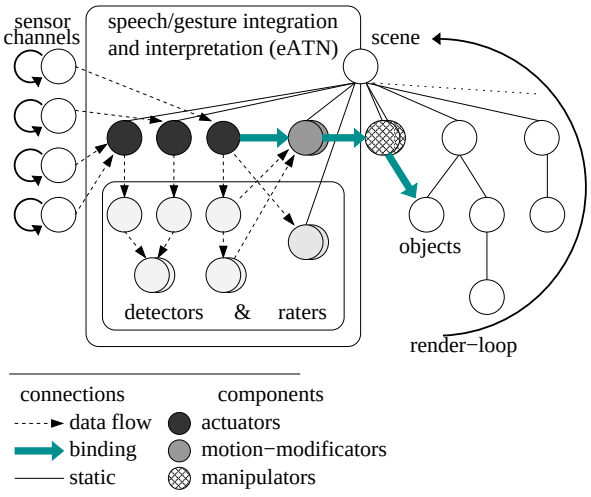

Figure 2: A schematic view of the main PrOSA concepts, their relations and the data flow between the different components.

2.1 Actuators

On the basic hierachy level, the attribute-sequences are anchored in so-called **actuators**. In our approach, actuators are entities that hide the sensor layer and provide reliable movement information even under unreliable frame-rate conditions. This is achieved by asynchronous sensor input and the prohibition of data extrapolation, which is particular important for trajectory interpretation. Actuators perform the following necessary steps to abstract from sensor data:

- Synchronization
- Representation to a common base
- Processing: transformation, combination, etc.
- Qualitative annotation

Actuators come in different flavours due to their output data format and the number of incoming sensor channels attached to them (s. fig. 2). Important examples are handform-actuators (providing information about finger bending and the angles between adjacent fingers) or single- and multichannel[2] *N*DOF-movement-actuators for significant body points (fingertips, wrists, head) and associated reference rays[3], line segments that represent deictic or iconic directional and orientational information (pointing direction, palm normal, etc.). Each actuator delivers a set of resulting synchronous movement samples for each frame and feed it into higher level processing units like **detectors** (and their subtypes) or into **motion-modificators** for an ongoing interaction processing.

2.2 Detectors

To classify the gesture movements, the incorporated gesture detection relies on template matching of eight spatio-temporal movement features:

1. Stop-and-Go
2. Leaving an associated *rest* position
3. Definite shape
4. Primitive movement profile
5. Repetition
6. Internal symmetry
7. External symmetry
8. External reference

The actuators deliver the preprocessed movement data to detector networks. Detectors of different kinds handle basic calculation tasks and operations for all necessary basic datatypes (real numbers, vectors, quaternions and 4x4 CGM's [4]), for example:

- Addition, subtraction or multiplication etc.
- Threshold tests and comparison operators
- Boolean operators
- Buffering of values over an interval

Each detector implements a simple function. Complex calculation networks can be constructed to detect the given spatio-temporal features using multiple detectors. Moreover, the calculation arithmetic is represented by the data flow network structure and is therefore easy to modify. Detectors can be added, exchanged, their parameters can be altered or they can be deleted at all. E.g., to detect definite shape of the hands, handform-actuators feed simple threshold detectors which themselves feed into boolean and/or detectors. A pointing posture can then be defined by the stretching of the index

[2] Depending on the number of sensor channels the actuator processes.

[3] A ray in the ideal sense. In practice, line segments are used.

[4] Computer Graphics Matrix

finger in addition to bending of the other fingers. In combination with a movement stop of the fingers and the whole hand this gives a high likeliness that a pointing gesture occured. A graphical example of a two layer network is again shown in figure 2.

2.2.1 Raters

Raters are a different kind of detectors. They are not concerned with gesture detection but with scene analysis and object rating (hence *raters*). To handle multimodal input for deictic utterances, e.g., " *Take* [pointing] *that blue wheel over there...*", it is necessary to process fuzzy input (in this case a pointing gesture) and combine it with speech interpretation. For this purpose, the actuators deliver line segments which represent the user's view and pointing direction – an example of reference ray usage. One type of rate-detectors will handle that input, estimate the difference between the segments, and sort the scene objects according to the resulting difference values. This information is one important basis for the multimodal interpretation.

2.3 Motion-modificators

Interaction is accomplished in two ways: *discrete* if the utterances form a complete interaction specification or *continuous* if information is missing and an ongoing gesture can be associated with the desired manipulation type. In the latter case motion-modificators abstract from unprecise user movements and map them continuously to precise changes of the virtual scene. The following *binding* will be established: an actuator routes data through a motion-modificator to an appropriate manipulator to map the movement to an attribute change (a *mimetic mapping*). This data flow is shown by the binding arrows in figure 2. A sequence of a resulting manipulation is presented in figure 3 :

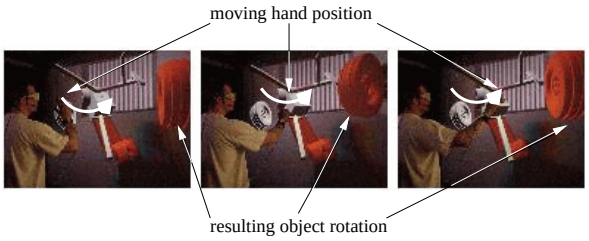

Figure 3: Continously rotating an object (a wheel). A kinemimic/mimetic gesture is used to achieve the desired object manipulation.

The duration of the binding is defined by the duration of the ongoing movement or the interception by external events. More precisely, a movement pattern consists of several constraints calculated by a detector network, e.g., a rotation of one hand is defined by:

- static hand form
- continous movement speed
- movement in one plane
- adajecent strokes with similar angles (less than 180°)

These informal descriptions[5] are translated into geometric and mathematical constraints based on actuator data to construct the resulting detector network. The motion-modificator receives the calculation results for each frame and keeps on working until the constraints are no longer satisfied. Another method to interrupt an ongoing manipulation is by external signals from the multimodal interpretation, e.g. when the user utters a "*stop*" or similar speech commands.

Motion-modificators map unprecise or coarse movements to precise object changes. To achieve this type of *filtering*, we need to monitor specific movement parameters – e.g., a rotation axis or a direction vector – and to compare them with a set of possible object parameters to modify. Therefore, when the binding is established, each motion-modificator receives a set of parameters that can be seen as *changegrid* members. For every simulation step, i.e. for every frame, they are compared to the actual movement parameters and the closest one (e.g., in the case of vectors the one with the minimal angular divergence) is chosen as the target parameter. This results in the desired filtering. To apply the parameter change to an object, a specific instance of basic **manipulators** receives frametime-adequate manipulation commands from the motion-modificator and changes the object parameter. Furthermore, by partitioning this operation using two different concepts, it is not only possible to establish a mimetic mapping: You could for example combine a rotation motion-modificator with a color or a sound manipulator. This would result in a kind of *metaphorical mapping*, an ongoing movement results in a color or sound change.

3 Multimodal interpretation

The interpretation process of gesture/speech-related utterances can handle temporal as well as *semantic* relations. An enhanced ATN[6] formalism has been developed to achieve the incorporation of temporal crossmodal constraints as well as to evaluate scene-related context information in real-time and to latch the interpretation into the driving render-loop (e.g. by using raters). The latter emphasizes the fact that the actual user's viewing perspective determines the reference semantics of all scene-related utterances dynamically. Their interpretation – the *deictic mapping*[3] – depends on both time-dependent dynamic as well as static

[5] A formal rule-based description can be found in [3]
[6] Augmented Transition Network

scene- and object-attributes like in: "*Take* [pointing] *that left blue big thing and turn it like* [rotating] *this*". Appropriate verbal (e.g., colors and positions) and gestural (e.g., view- and pointing direction) input is disambiguated during user movements through a direct connection to the scene representation (s. 2.2.1) and results are stored in so-called **spacemaps**. Fig. 4 shows the migration of one specific object (the black oval) in a spacemap during interaction and user movement. Every row represents the result for one simulation step. The first entry in each row holds additional data (time, segment, etc.).

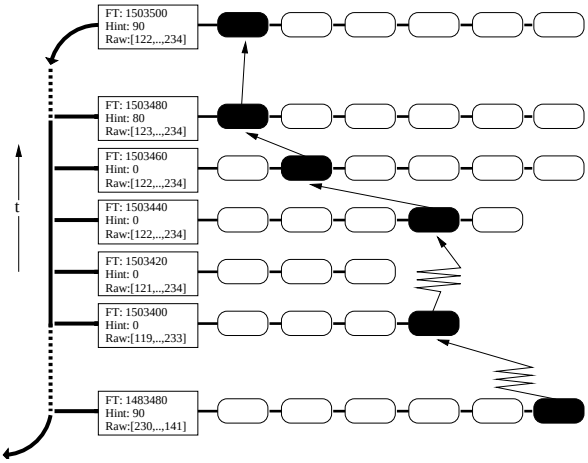

Figure 4: A spacemap as a temporary memory. The relative position of an object to a line segment is represented for every simulation step.

This preprocessing allows the handling of varying multimodal temporal relationships (e.g., a look-back) without the necessity for buffering all scene descriptions of past frames. In addition, the enhanced ATN allows to express application logic in the same representation as the multimodal integration scheme. This results in a convenient way to adapt multimodal interfaces to different applications.

4 Implementation and application

Figures 1 and 3 show a sequence during user interaction with the virtual construction application[1][7]. In addition to pure speech commands (for triggering actions like opening of doors etc.), basic interactions are enabled using gesture and speech. Objects can be instantiated and connected as well as referenced and moved around with distant *communicative* interaction as well as with direct manipulation (if desired). Actuators, detectors and motion-modificators enable to trigger and evaluate deixis (view, pointing), kinemimic/mimetic gestures (rotating of the hands), grasping and several more symbolic gestures. Work is on the way to add pictomimic/spatiographic (iconic) gestures and to incorporate an articulated figure [2] as well as to test an unification based speech/gesture integration using the existing framework. The goal is to develop a toolkit set of basic PrOSA-concepts to establish detection networks for frquently needed standard interactions in virtual environments. The speech recognition system is a research prototype and works speaker-independent. The current PrOSA-concept implementation makes use of, but is not limited to the AVANGO-toolkit[5]. The data flow has been established using *field* connections (a concept similar to the one found in VRML97). All components can be constructed and all connections can be established by the AVANGO-internal scripting language (which is Scheme). This allows on-the-fly changes and a rapid prototyping approach for new projects. Particular design efforts have been made to achieve portability by an explicit formal definition of all concepts [3] by taking general VR-conditions into account.

[7]This work is partially supported by the *Virtuelle Wissensfabrik* of the federal state North-Rhine Westfalia and the Collaborative Research Center SFB360 at the University of Bielefeld.

References

[1] Bernhard Jung, Stefan Kopp, Marc Latoschik, Timo Sowa, and Ipke Wachsmuth. Virtuelles Konstruieren mit Gestik und Sprache. *Künstliche Intelligenz*, 2/00:5–11, 2000.

[2] Stefan Kopp and Ipke Wachsmuth. A knowledge-based approach for lifelike gesture animation. In W. Horn, editor, *ECAI 2000 - Proceedings of the 14th European Conference on Artificial Intelligence*, pages 663–667, Amsterdam, 2000.

[3] Marc Erich Latoschik. *Multimodale Interaktion in Virtueller Realität am Beispiel der virtuellen Konstruktion.* PhD thesis, Technische Fakultät, Universität Bielefeld, 2001.

[4] J.-L. Nespoulous and A.R. Lecours. Gestures: Nature and function. In J.-L. Nespoulous, P. Rerron, and A.R. Lecours, editors, *The Biological Foundations of Gestures: Motor and Semiotic Aspects.* Lawrence Erlbaum Associates, Hillsday N.J., 1986.

[5] Henrik Tramberend. A distributed virtual reality framework. In *Virtual Reality*, 1999.

[6] http://www.techfak.uni-bielefeld.de/techfak/ags/wbski/wbski_engl.html.