Gesture Recognition of the Upper Limbs – From Signal to Symbol

Martin Fröhlich* and Ipke Wachsmuth

Technische Fakultät, Universität Bielefeld,
Postfach 100 131, D-33501 Bielefeld, Germany
e-mail: {martinf, ipke}@TechFak.Uni-Bielefeld.DE

Abstract. To recognise gestures performed by people without disabilities during verbal communication – so-called coverbal gestures – a flexible system with task-oriented design is proposed. The issue of flexibility is addressed via different kinds of modules – grasped as agents –, which are grouped in different levels. They can be easily reconfigured or rewritten to suit another application. This system of layered agents uses an abstract body-model to transform the up-taken data from the six-degree-of-freedom-sensors, and the data gloves, to a first-level symbolic description of gesture features. In a first integration step the first-level symbols are integrated to second-level symbols describing a whole gesture. Second-level symbolic gesture descriptions are the entities which can be integrated with speech tokens to form multi-modal utterances.

1 Introduction

For the interaction with virtual worlds new metaphors are needed to cope with the user's demand for interfaces which are easy to use and easy to learn. In our working group we use a variation of the metaphor of a "desktop virtual reality application", in our case a large screen display with multi-modal input facilities, e.g. for speech and gesture. The user shall be enabled to communicate in a natural manner with the scenery on the large screen display which is filling the users field of vision. In our demonstrator, developed in the SGIM[1] project, utterances like "Make this bigger", together with a pointing gesture are to be detected and integrated to a command understandable by the scenery controller. The aim of our approach is to build a speech and gesture integration model which can be used to build task-oriented interfaces for virtual reality applications. We concentrate on the semiotic function of gesture as categorised by [3]. The function of such gestures is to communicate meaningful information. The ergodic (shaping) and epistemic (tactile sensing) functions of gesture are not addressed by our approach.

This paper concentrates on the question how to symbols could be derived from the signal sensors that register information from a more-or-less continuous

* Scholarship granted by "Graduiertenkolleg Aufgabenorientierte Kommunikation" of the Deutsche Forschungsgemeinschaft (DFG)

[1] Speech and Gesture Interfaces for Multimedia, see also [8]

stream of upper-limb gestures, and how to link them to "gesture units" that can form the basis of steering commands for a virtual reality application. The paper is organised as follows. In section 2, we describe a hierarchical approach that leads from the recognition of upper-limb movements to command symbols. In section 3 we describe the system structure of the software system we are developing and give insights into the technical processes involved.

2 Signal-to-Symbol Hierarchy

To cope with the task of signal-to-symbol transformation, we use a hierarchical symbol-based approach involving first-level and second-level symbols. First-level symbols describe gestural form features such as hand shape or hand orientation, and are derived from the signal via an abstract body model. Second-level symbols constitute application-specific semantic units, conceptualised in the light of the application, which are derived from the first-level symbols. This separation enables us to adapt the model to different applications by different interpretations of the symbols meaning, i.e. different second-level symbols. Thus, for most applications there will only be the need for changing the second-level symbols' interpretation.

2.1 Recognition of Movements

The movements of the wrists and the trunk, as well as the configuration (posture) of both hands are detected by sensor devices as six degree of freedom information, that is, three dimensions for translation and three dimensions for the rotation at the bottom level of the hierarchy. This information then is mapped to an abstract body model to cope with problems arising from temporarily unavailable data, miscalibration of the sensors, and – most importantly – to provide the recognisers with information which can only be derived from the input, e.g. the joint angles. This model can also enforce constraints to the joint angles, that is, it can detect nonsense sensor data (e.g. sensor has fallen down) without setting itself to a nonsense state. By this, the model always provides consistent data to the recognisers.

2.2 Symbolic Notation

Our approach builds on the idea of a set of connected recognisers which detect certain features of a gesture, which, in general, involves movements of several body parts.

For an internal representation, we have looked for a notation system providing, in the ideal case, notations for:

1. hand-shapes
2. hand orientation
3. hand/arm location
4. hand/arm movement

We found all these properties in the Hamburg Notation System (HamNoSys) [14]. In a task-oriented setting, we will not use all gestures possible, thus we can work with a subset of HamNoSys; for notation examples see Fig. 1 and Fig. 2. In the first approach, we concentrate on hand/arm location, hand orientation, and some hand-shapes.

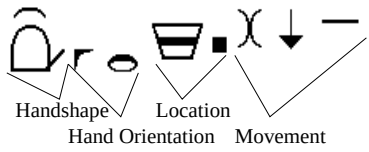

Fig. 1. HamNoSys: "mühsam" ("very tiring") performed in German sign language

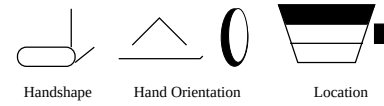

Fig. 2. HamNoSys example of a "straight forward" pointing gesture

In HamNoSys every class from the above list (i.e., hand-shapes, hand orientation, etc.) is represented in a disjunct class of symbols. At any given moment only one symbol of each class can be assigned. To assign a meaning to a certain string of such symbols, a defining "gesture lexicon" has to be used.

We would like to follow the views of Harling and Edwards in [6] that people produce a continuous stream of gestures, not naturally segmented into a flow of discrete gestures taken from a "gesture lexicon". But for the gesture recogniser's output we want a stream of distinct symbolic tokens for the fluidly connected gestures that appear in the input. Thus we need a solution for the gesture segmentation problem.

In our approach, this problem is addressed by using a time discrete model building on findings of Pöppel [13]. There is indication that human beings use a modular system to process up-taken stimuli and synchronize/integrate such modules in a multi-step process. For our approach we try to model these modules – which can be conceived as agents in Minsky's sense [10] – as different layers of symbolic tokens and processing stages. In the first stage symbols will be integrated in a time window smaller than the temporal order threshold for discontinuous information processing of 30 ms. Below that threshold human beings cannot distinguish the temporal order of two events. In the following stage, when gestural and speech information is to be integrated into meaningful units that can be transferred to the application system, another time window – here 3 s – is used. This time-span is the limit above which humans are no longer able to to create cognitive entities by integrating successive events. We consider these findings from cognitive science to be adequate in this technical context, because we want to detect gestures which could be addressed to a cooperating partner, who would have the ability to understand them, as well as to an application system.

These cognitive findings lead us to a data model which is visualised as a sliced, extruded triangle as shown in Fig. 3. The levels in the triangles on the left side indicate the different stages, the markers in the levels the symbols and the connections indicate the recognition and integration events. The slices on the right side of Fig. 3 indicate the atomic time units of 30 ms. The inclination of the slices represents the time delay between the actual time flow ("wall clock" time) and the system time.

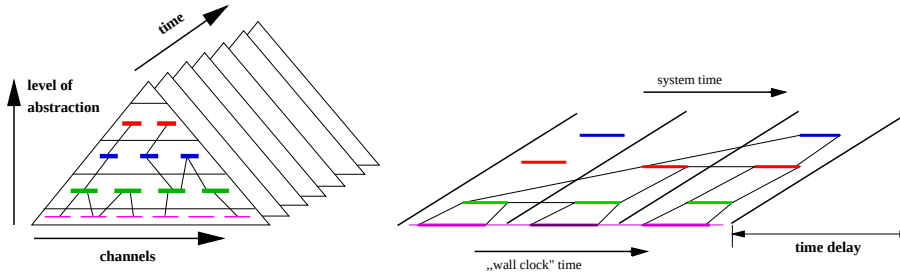

Fig. 3. Integration Hierarchy (further explanation see text)

3 System Structure

As mentioned in the introduction, our gesture recogniser is part of a virtual reality application. Fig. 4 shows the system architecture we are developing for detecting features from the sensor data and integrating them to meaningful utterances for the virtual reality environment. In more detail, this application is addressed by [8]. In the following sections, we describe the technical part that pertains to the focus of the paper, namely, how to obtain symbolic gesture data from more-or-less continuous and noisy sensor output.

3.1 Sensors

To acquire the data needed, we use the six-degree-of-freedom sensoring device called "Flock of Birds" by Ascension Technology Corporation. The device has an internal measurement cycle of up to 144 Hz and some internal filters for early signal processing. Further we use Virtual Technologies "CyberGlove" data gloves Model CG1801 for the detection of left and right hand configuration. Two Ascension sensors (called "Birds") are attached to the mounting point of each of the gloves. Another one is mounted on a light-weight helmet worn by the user. Two more Birds are available to be attached to the subjects upper arms to register the position and orientation of the humerus.

The helmet sensor is used as a reference point for the subject's height and as a coarse means to detect the subjects "nose vector" – the vector orthogonal to the

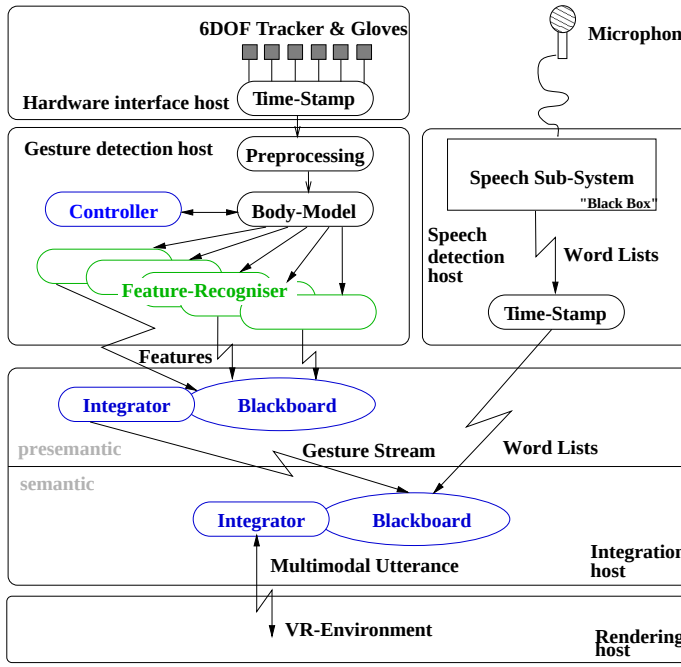

Fig. 4. The System Architecture

subject's face – to get an idea of the subject's area of interest. This information will be of use to reduce ambiguity whether the user intends to address the system (see section 4).

3.2 Distributing Data

The data of the "Flock of Birds"and the CyberGloves are delivered via serial lines, one for each 6DOF sensor and glove. The specific protocols for managing the measurement devices are handled by a single Sun SPARCstation-10 which time-stamps the data and distributes it to customer processes on other machines. For this purpose we use the connection-less user datagram protocol (UDP) multicast facility embedded in the TCP/IP protocol suite [16], [1]. The term "multicasting" refers to the fact that not all systems in the network or subnet are receiving the data, only a group of systems which have registered in the so-called multicast group will get copies of the data packets in question.

This method provides an as-fast-as-possible distribution of the data via network to as many clients as necessary. Every UDP packet is time-stamped and by this it is guaranteed to identify data loss or disordered packets as well as other discontinuities in the data stream.

3.3 Filtering Data

The data received from the sensors is loaded with noise that needs to be eliminated in early processing. In Fig. 5 a typical set of data is shown as produced by a stationary 6DOF sensor lying in approx. 1.5 m distance from the transmitter on a 73 cm high table. Fig. 6 shows a sketch of a regular 3D-array of reference points, seen through the mapping function given by the sensors. The lab floor consists of steel-armoured concrete, which explains the strange values at the bottom because the Flock uses an electro-magnetic field that is distorted by metallic objects. The transmitter is located near the concave edge of the graph. This edge is indented because of obstacles behind the transmitter. The canvas for the video-projector would form the background in Fig. 6.

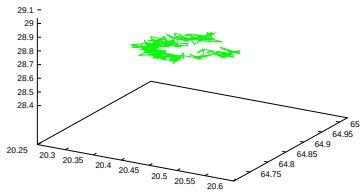

Fig. 5. Stationary 6DOF sensor (scales in inches).

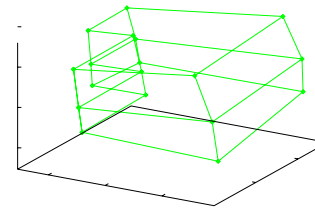

Fig. 6. Geometry distortion of sensor space

As a consequence the raw data consists of the net data and two components of noise: First linear noise, which can be easily eliminated via a median filter, secondly a circular drift, which is so small that in our case we can ignore it by rounding to a discrete sensor space with a distance of 1 cm between the grid lines.

From Fig. 6 we can see that the sensors imply a mapping function onto the measurement space. The data from Fig. 6 is derived via an equidistant measurement grid of 140 measuring points. The figure is simplified to the border of that grid. The sensor space has an internal curvature given by the mapping function if seen from the measurement space. This geometry distortion can easily be fixed by applying a compensation function $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ to the sensor data as shown in Fig. 7.

3.4 Geometric Body Model

Before the data received from the sensors and cleaned by filtering are analysed, they are mapped to an abstract body model of the upper limbs (incl. the head).

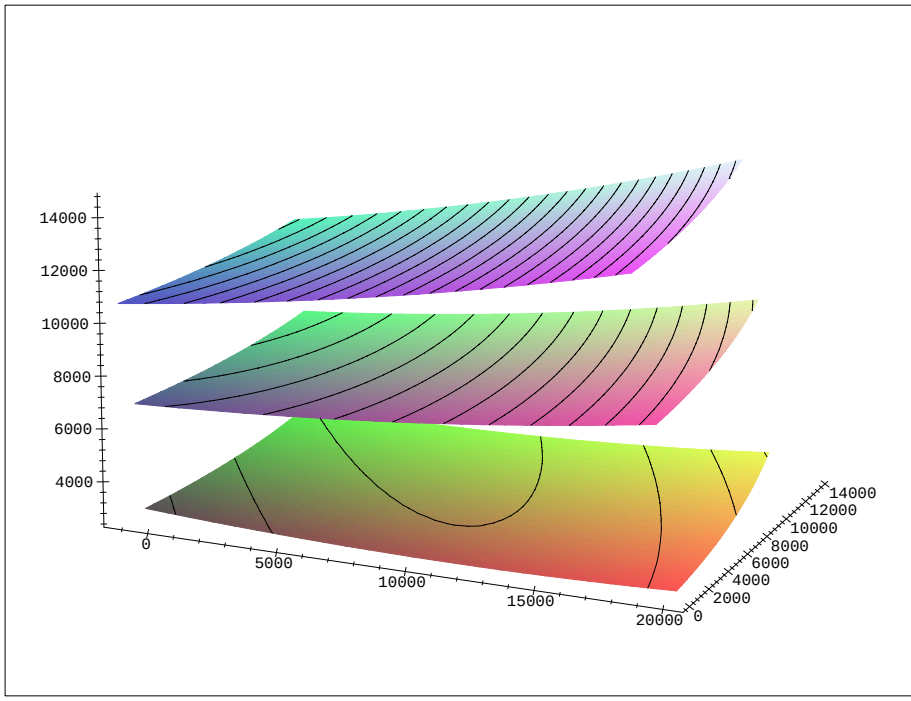

Fig. 7. Curvature compensation function $f(x, y, 3000), f(x, y, 8000), f(x, y, 12000)$

This enables us to use few calibration steps and the same detection device with subjects of different sizes. The recognition of gestures will take place on the data taken from the abstract model. The abstract body model is a crucial part in our system and thus is explained in detail below.

In anthropometry, orthopaedy, and physiology a variety of rather fine-grained models of the human body were developed. Some of them represent the shoulder joint as the sum of more than 50 single degree of freedom joints. This is a far too detailed view of the human body for our approach. Because of this, we searched for simplified models like the ones described by Ko in [7] and Lenarčič and Umek in [9].

In Fig. 8 a model of the upper limbs as proposed by H. Ko in his Ph.D. thesis is shown ([7] pp. 84-86). The graphic symbols (after [17]) are used to facilitate a compact drawing of the joints; cf. Fig. 9. The icon (C) is for a twist joint in which the links and rotation axis z are parallel. Icons (A) and (B) represent the same type of bending joint except for a difference in the direction of the rotation axis in the current view: Icon (A) is used when the rotation axis is left, right, up, or down; icon (B) is used when the rotation axis is either forward or backward. The model is based on the DH-notation from J. Denavit, R. S. Hartenberg [5],

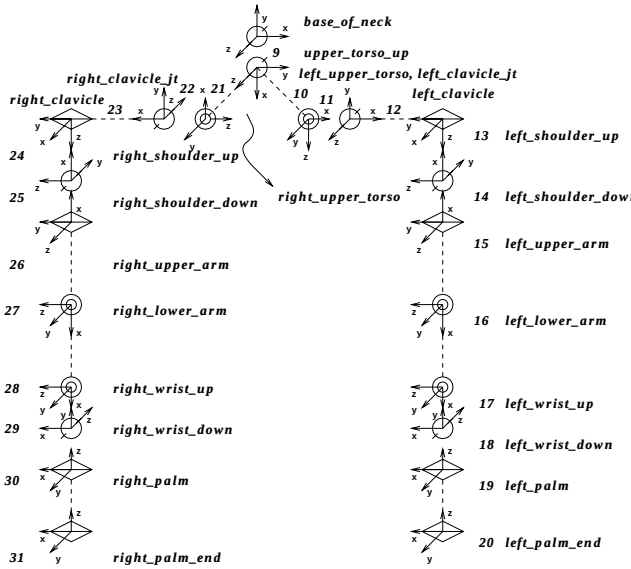

Fig. 8. Ko's model of the upper limbs (after [7])

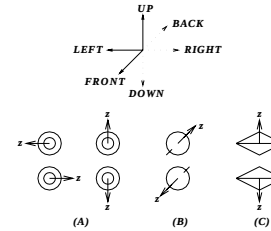

Fig. 9. Legend to Fig. 8

and Richard P. Paul [12]. Complex joints with multiple degrees of freedom are decomposed into several single-degree-of-freedom joints.

A still simpler model is proposed by Lenarčič and Umek [9]; cf. Fig. 10. This model had been designed to calculate the approximate human arm workspace related to the reachability of the selected reference point placed on the wrist.

To represent the human body geometrically we choose a model, shown in Fig. 11, which is almost as simple as Lenarčič's and Umek's but has added the other arm and the vertebra/thorax (1) and a very simple head/neck (2) representation. The graphic symbols are again in accordance with [17]. The dashed lines indicate joints with several degrees of freedom, which have been decomposed and simplified to single degree of freedom joints. The sternoclavicular and scapularthoracic joints are abstracted by structure (4), the shoulder including the acromioclavicular, the coracoclavicular, and the glenohumeral joints are abstracted by structure (3), and the elbow as well as the ulna-radius joint are represented by structure (5), including the humeroulnar, the humedoradial, and the superior radioulnar joint (cf. Fig. 11 throughout). The end effector's (hand/head) coordinate systems (6) are given by the flock-of-bird tracking devices. The trackers are mounted on the surface of the body via gloves and helmet, thus there coordinate system's origin is situated in some distance from the centre of the limb. Therefore a certain tracker offset has to be included, as to be found in the representation of the lower arm and the head. On the left side of Fig. 11 we include descriptions indicating the joint's measurable features, on the right side the links are named according to the major bones they represent.

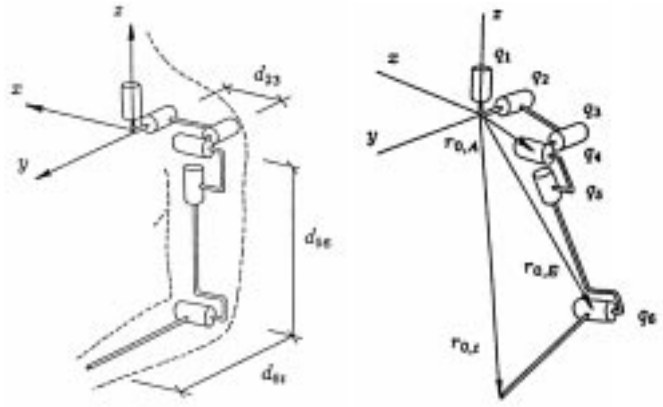

Fig. 10. Lenarčič's and Umek's Arm Model (from [9])

With this model we can describe the posture of the body but not the dynamic changes to the posture which we turn to below.

3.5 Dynamic Body Model

The geometric model is the basis for the dynamic model. The term "dynamic", here, refers to the temporal change of the model's posture, Together, the geometric and the dynamic body model form the so-called abstract body model. In biocybernetics robust models for walking machines were developed using the motion control mechanisms of insects [4], as well as by engineers and computer scientists in ergonomic research [9]. These models prove very useful for the design of a dynamic body model.

Steinkühler and Cruse provide such a dynamic model with their MMC (Mean of Multiple Computations) network [4]. It can solve the problem of conversion between world and joint coordinates (and vice versa) providing a continuous path, including the underconstrained case. The MMC is a recurrent network which relaxes to adopt a stable state corresponding to a geometrically correct solution, even when the input does not fully constrain the solution. The MMC can easily cope with limitations by the joint space or the workspace and is robust against singularities. The disadvantage is that the convergence of the MMC is only proven for the linear case that cannot keep the segment length, see [4]), but experimental results ([15]) show that after typically 30 iterations a usable solution is presented. Even if the net is queried at an intermediate state, the result will represent a (nonlinear) interpolation between the start and target point.

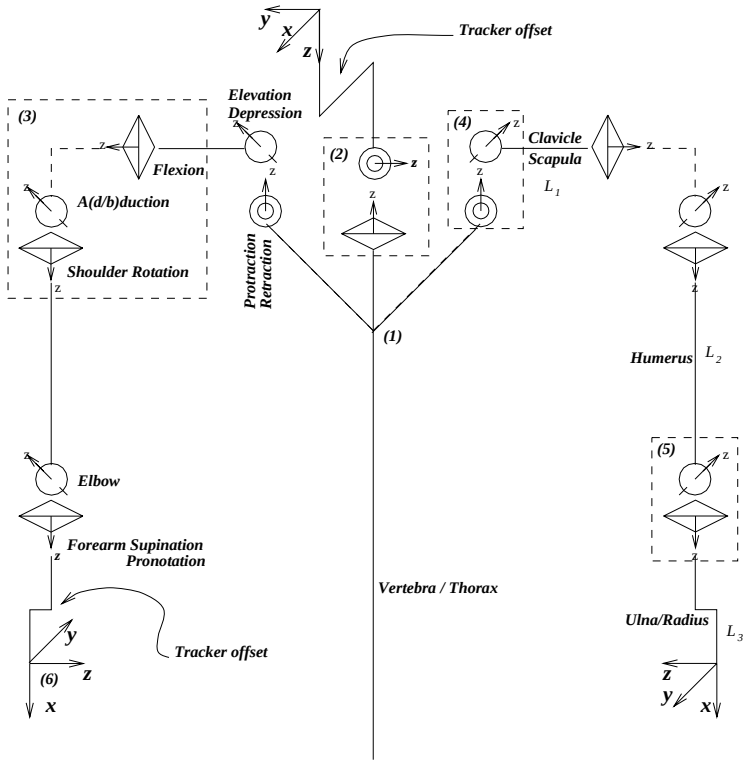

Fig. 11. The Geometric Body Model chosen in our approach

4 Feature Recognition and Integration

Building on the data available by the earlier integration stages described in the previous sections, we have started to develop the feature recognisers and the integrators. An autonomous agent-type component observes the geometrical constraints and the raw data, detecting inconsistencies. Additional autonomous components can be added to compute further momentums from the coordinates, like speed and acceleration.

For our goal VR application we need recognisers concerning *hand orientation*, *hand location* and some of those concerning *movement*. We are currently developing a separate recogniser for the detection of the *hand shapes*, as well as a recogniser for detection of the *orientation of the head*. The main purpose of the head orientation information is to grasp whether or not the probationer is addressing the system. This enables us to sort out irrelevant limb movements.

The feature symbols (first-level, cf. section 2) are gathered within a tight measurement cycle, determined by the measurement cycles given by the Flock and the Gloves. This measurement cycle reoccurs with approx. 3-10 Hz and by

this well within the temporal order threshold for discontinuous information processing of 30 ms as discussed in section 2.2. Each symbol is labelled with the period of time in which it has been detected, a normalised confidence value from the detector, and the most significant raw data. This additional information will be used after the integration phase, e.g. to identify the position of a selected object.

The basic idea is to incrementally integrate a buffer storing the symbols which have occurred during the respective time frame (30 ms or 3 s), delivering one or more symbols describing the events in that period of time.

The semantic integration of second-level symbols is planned to be constructed under the same principles as the presemantic one of first-level (presemantic) symbols.

Furthermore we develop a controlling agent for the MMC net. The agent observing the geometrical constraints and the raw data, adapts the constraints according to anthropometric data. The length of individual links can be estimated by using easily available calibration data, for instance the subject's height (helmet sensor) or diameter of human arms workspace (calibration procedure) using tables from a variety of sources describing anthropometric features as available from [2], [11] and from the German Institute of Standardisation (DIN). In our model we have parameterised all constraints, links, and joints and a set of rules for initialising the parameters according to the calibration data. We use the 50th percentile for adult German men as a starting point which then is adapted to the actual user by matching the measured data against features derived from the inner constraints of the standardised statistical anthropometric data. This component can be grasped as a control agent for the MMC, as is has sensor (raw data), knowledge (anthropometric tables), reacts to changes in the environment (data mismatch), acts (adjusts the constraints of the MMC), and has the general goal to keep the model consistent with the observed data.

5 Conclusion

In this paper, we described an approach how to derive symbolic tokens from a stream of upper-limb gestures uttered by users of a virtual reality application system. An agent-controlled abstract body model is developed which shall enable the recognition of body gestures from data delivered by position tracking devices. To adapt the model to different user parameters, anthropometric knowledge is used to reassign constraints to the dynamical model. Combined with a time-discrete model of temporal integration, our approach provides a foundation to derive presemantic representations of upper-limbs movement, denoted in a subset of the Hamburg Notation System (HamNoSys). So far deictic and emblematic gestures have been considered. In our next work, we will also include mimetic gestures (i.e., gestures that mimic interaction with an object) and deal with the question how to interpret tokens from gesture and speech.

References

1. Douglas E. Comer and David L. Stevens. *Internetworking with TCP/IP: Client - Server Programming and Applications; BSD socket version*, volume III. Prentice-Hall International, Englewood Cliffs, NJ, 1993.
2. Renato Contini. Body segment parameters, part ii. *Artificial Limbs*, 16(1):1–19, Spring 1972.
3. James L. Crowley and Joëlle Coutaz. Vision for man machine interaction. In *Proceedings of Engineering Human Computer Interaction* (EHCI'95 Grand Targhee, August 1995), pages 28–45, Chapman and Hall, London, August 1995.
4. H. Cruse and U. Steinkühler. Solutions of the direct and inverse kinematic problems by a common algorithm based on the mean of multiple computations. *Biological Cybernetics*, 69:345–351, 1993.
5. J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanics*, 77:215–221, 1955.
6. Philip A. Harling and Alistair D. N. Edwards. Hand tension as a gesture segmentation cue. In Philip A. Harling and Alistair D. N. Edwards, editors, *Progress in Gestural Interaction: Proceedings of Gesture Workshop '96*, pages 75–87, Springer, Berlin et al., 1997.
7. Hyeongseok Ko. *Kinematic and Dynamic Techniques for Analysing, Predicting, and Animating Human Locomotion*. PhD thesis, University of Pennsylvania, 1994.
8. Marc Erich Latoschik and Ipke Wachsmuth. Exploiting distant pointing gestures for object selection in a virtual environment. same volume.
9. Jardan Lenarčič and Andreja Umek. Simple model of human arm reachable workspace. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(8):1239–1246, August 1994.
10. Marvin Lee Minsky. *The Society of Mind*. Simon and Schuster, New York, NY, 1985.
11. NASA: U.S. National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, Houston, TX *Man-System Integration Standards (MSIS): NASA-STD-3000 I-III Revision B*, August 1994.
12. Richard P. Paul. *Robot Manipulators*. MIT Press, Cambridge, MA, 1981.
13. Ernst Pöppel. A hierarchical model of temporal perception. *Trends in Cognitive Sciences*, 1(2):56–61, May 1997.
14. Siegmund Prillwitz, Regina Leven, Heiko Zienert, Thomas Hamke, and Jan Henning. *HamNoSys Version 2.0: Hamburg Notation System for Sign Languages: An Introductory Guide*, volume 5 of *International Studies on Sign Language and Communication of the Deaf*. Signum Press, Hamburg, 1989.
15. Ulrich Steinkühler. *MMC- Modelle zur Lösung kinematischer Aufgabenstellungen eines redundanten Manipulators*. Dissertation, Universität Bielefeld, Fakultät für Mathematik, 1994.
16. W. Richard Stevens. *The Protocols*, volume 1 of *TCP/IP Illustrated*. Addison-Wesley, Reading, MA, 1994.
17. Tsuneo Yoshikawa. *Foundations of Robotics: analysis and control*. MIT Press, Cambridge, MA, 1990.