Vision-Based Interaction within a Multimodal Framework

Vítor Sá [1,2] Cornelius Malerczyk [1] Michael Schnaider [1]

[1] Computer Graphics Center (ZGDV)
Rundeturmstraße 6
D-64283 Darmstadt

[2] University of Minho (UM)
Campus de Azurém
P-4800-058 Guimarães

{vitor.sa,cornelius.malerczyk,michael.schnaider}@zgdv.de

Abstract

Our contribution is to the field of video-based interaction techniques and is integrated in the home environment of the EMBASSI project. This project addresses innovative methods of man-machine interaction achieved through the development of intelligent assistance and anthropomorphic user interfaces. Within this project, multimodal techniques represent a basic requirement, especially considering those related to the integration of modalities. We are using a stereoscopic approach to allow the natural selection of devices via pointing gestures. The pointing hand is segmented from the video images and the 3D position and orientation of the forefinger is calculated. This modality has a subsequent integration with that of speech, in the context of a multimodal interaction infrastructure. In a first phase, we use semantic fusion with amodal input, considering the modalities in a so-called late fusion state.

Keywords

EMBASSI project, 3D deictic gestures, multimodal man-machine interaction, agent-based systems.

1. INTRODUCTION

EMBASSI is the name of a joint project sponsored by the German government (BMBF), which began in the summer of 1999, addressing innovative methods of *man-machine interaction* (MMI). In the broad area of information interfaces, the term MMI reminds us that computers are gradually infiltrating more and more of the machinery and equipment commonly used in our daily life.

In today's man-machine interaction, computer input and output are quite asymmetric. The amount of information or bandwidth that is communicated from computer to user is typically far larger than the bandwidth from user to computer [Jacob96]. Since this unbalance often influences both the intuitiveness and performance of user interaction, one of the EMBASSI benefits will be the enhancement of bandwidth from the user to the system.

While a computer *output* presentation over multiple channels has become familiar to us under the designation of *multimedia*, the input channels or sources, also called *input modes* or *modalities*, are the basis of those kinds of applications said to support *multimodal human-computer interaction*.

Our contribution is to the field of video-based interaction techniques and is integrated in the home environment of EMBASSI. In this paper, we are only considering the gesture modality, more concretely the gestural typology of deictic (pointing) gestures. A related study can be founded in [Kohler96].

The following Section 2 describes the EMBASSI project as the multimodal framework where we are applying our vision-based interaction techniques, specifically hand pointing gestures. Section 3 presents in some detail the necessary calibration, as well as the recognition and tracking system. Section 4 outlines the integration of the gesture modality with the others in the context of a multimodal interaction. In section 5, we present some conclusions.

2. FRAMEWORK

Our work is integrated in the EMBASSI framework - described in more depth in [Hildebrand00]. In EMBASSI, innovative interaction technology will be achieved through the development of intelligent assistance and anthropomorphic interfaces. Telecommunication and network infrastructure is used in order to provide an added value opponent to stand-alone systems with regard to usability and functionality.

2.1 Goals

The overall objective of EMBASSI is the support of humans during interaction with different kinds of technical infrastructures in everyday life. Although the foreseen achievements of EMBASSI can be applied to devices in industrial and office environments, the scope of EMBASSI is directed towards the private sector, including home, car and public terminal applications.

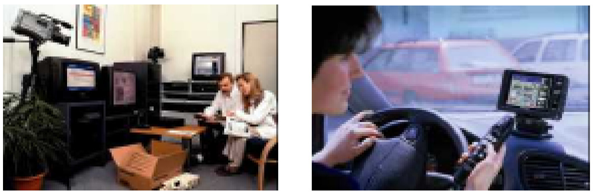


Figure 1 – EMBASSI application areas

The aim of the EMBASSI specification should lead to a new user paradigm in private application areas:

- Transition of the paradigm „device" to the paradigm „system", where user expectation in terms of environmental knowledge of the interaction is incorporated. The basic groundwork is provided by network technology, which allows the inquiry of certain system and device states (e.g. TCP/IP/IEEE802.3(11), IEEE1394/ HAVi).

- Transfer from unimodal to polymodal input and output. In addition to the use of speech and pointing gestures in private home applications, an anthropomorphic graphical output will be addressed.

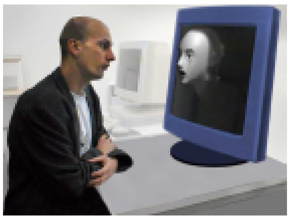


Figure 2 – Anthropomorphic user interface

The development of appropriate assistance technology is a primary objective of EMBASSI. This includes the elaboration of a uniform approach for the systematic development of user interfaces and assistance systems by the development of:

- *Modular building blocks* of interaction basic technology for a natural and intuitive man-machine interaction;

- A generic architectural framework, including an adequate semantic protocol for the realization of assistance systems based on interoperable components.

2.2 Architecture and protocol

The aim of the intended generic architecture is to provide the backbone of the different EMBASSI derivatives. Fundamental objectives of the architecture are:

- Homogenize the different application scenarios, especially with respect to the protocol and interfaces;

- Provide a common understanding of the interfaces and modules.

The complex interaction process of intelligent assistance in a multimodal manner, where the system consists of diverse technology components (from the recognizers and multimodal integrators, to the context and dialogue managers), results in very complex information processing.

One way to realize this information processing flow as an architecture is to pipeline the components via procedure calls -- or remote procedure calls -- in the case of a distributed but homogeneous system (in programming language). For distributed and heterogeneous software, this may prove difficult and the solution goes through *agent-based software engineering*. In essence, the several system components are "wrapped" by a layer of software that enables them to communicate via a standard language over TCP/IP. The communication is then processed directly based on some concepts of distributed systems, like asynchronous delivery, triggered responses and multi-casting, or, alternatively, by using a *facilitated* form. In EMBASSI, a facilitated ("hub-spoken") multi-agent architecture is being used. Only when unavoidable, due the possible bottleneck derived from high-volume multimedia data transfer, can this approach be "by-passed".

To integrate all the independent and heterogeneous system components, the widely used KQML (Knowledge Query and Management Language) was chosen as the agent communication language (ACL). This decision was based on the effectiveness of KQML regarding the communication between agent-based programs. It provides high-level access to information and can be used for low-level communication tasks, such as automatic error checking.

KQML is complementary to distributed computing approaches (e.g. OMG CORBA/IIOP), whose focus is on the transport level (how agents send and receive messages). The focus of KQML is the "language level" – the meaning of the individual messages. Furthermore, in order to successfully interoperate, the agent-based system must also agree at the "policy level" (how agents structure conversations) and at the "architecture level" (how to connect systems in accordance with constituent protocols) [Finin93].

KQML is a language for representing *communicative acts* - for programs to communicate attitudes about information. KQML has a base definition with the possibility of being extended, which allows agents to use so-called *performatives* (the name of KQML messages) that do not appear in the standard specification.

One of the most positive characteristics of this ACL is the separation from the language used to code its *contents*. This has been demonstrated to be a good practice in the context of agent communication. The EMBASSI project, not being restricted to KQML, follows this advised distinction and, therefore, is using XML (Extensible Markup Language) [W3C00] for syntax definition of the message content, and Description Logics (DL) [Domini97] for knowledge representation. The DL language is used to build sets of concepts and roles called ontology [Franconi99], the way the lower level modules of the architecture have to provide knowledge to the more universal ones.

To this end, we found several software tools to build agent-based systems. Some examples are OAA [Martin99], Jackal [Cost99] and JATLite [Jeon00]. The last one was made open-source software, available under the GNU general public license since December 1998.

2.3 Modalities

EMBASSI has an open and modular architecture with an amodal treatment of the different modalities at the level of the dialogue manager. This permits the inclusion of as many modalities as needed to perform the intended advanced interaction.

The speech modality itself is treated in two separate modules: one is called the *speech recognizer* and the other the *speech analyzer*. The first detects the different morphemes of the sentences (based on the possible allophones and the basic phonemes) and sends a *word hypotheses graph* to the next module to be analyzed. Consequently, the analyzer module, based on the received structure, determines the semantic information concerning the utterance in cause, always considering the EMBASSI-defined ontologies.

The video-based components are integrated in home, car, and public terminal scenarios. Different requirements concerning illumination, camera hardware, accuracy, and reliability are considered in various scenarios. Video-based input encompasses a large spectrum of modalities from *gestures*, *facial expressions* and *emotions*, to *lip-reading*, *eye tracking*, and *stick pointing*.

3. VISUAL INPUT

Milota and Blatner [Milota95] have defined "a taxonomy of gesture with accompanying voice". As mentioned at the beginning, we are only considering the deictic gestures typology.

The video-based interaction we are describing uses a stereoscopic approach and allows a natural selection of devices via a pointing gesture. This can be combined with a speech recognition component to enhance the independent unimodal inputs by an integrated multimodal approach. The pointing hand is segmented from the video images and the 3D position and orientation of the forefinger is calculated. Afterwards, the selected device is identified by using the known camera parameters and device positions.

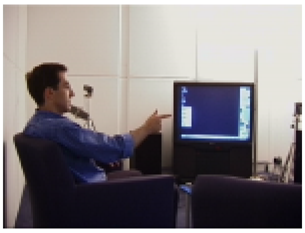


Figure 3 – Gesture-based interaction

The recognition of the pointing gesture is based on a template matching method. The gesture is described by predefined landmark points on the boundary of the object (see figure 4). A statistical description of the gesture in nature is calculated in combination with its modes of variation. This *Point Distribution Model* [Cootes00] of the hand is used for detecting and recognizing the gesture in gray level images. The system set up for gesture recognition contains modules for camera calibration of a stereoscopic camera system, data acquisition, image preprocessing, object recognition, object fitting, 3D pose estimation and communication. The modules and their relationship are described below.

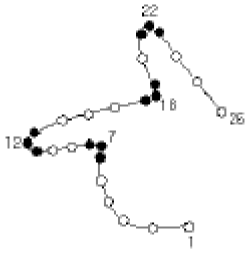


Figure 4 – Model of the pointing gesture

3.1 Image pre-processing

After the acquisition of two corresponding gray level images, image pre-processing is necessary. To suppress disturbing noise, averaging with a Gaussian convolution mask smoothes the images [Haberäcker95]. In the case of bad lighting conditions, performing histogram equalization enhances the image contrast. Furthermore, the edge information is extracted by applying various edge detection algorithms like the Sobel operator or the Canny edge detector [Sonka98] (see figure 5).

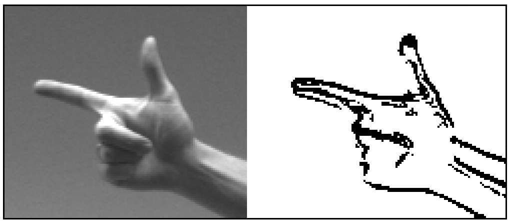


Figure 5 – Pointing gesture and binarized edge image

3.2 Object recognition

The pre-processed images are now used to detect and recognize eventually existing pointing gestures. The outcomes of this process are rough approximations of the contours of the gesture in the images. To detect these initial contours for a later fitting phase, the *Simulated Annealing* algorithm [Metropolis53, Pirlot96] was chosen. *Simulated Annealing* is a stochastic optimization algorithm. Its purpose is the minimization of an objective function *E(X)* where *X* is a multidimensional state vector of the objective function. An initial value x_0 is randomly changed over many iterations by updating the current solution by a solution randomly chosen in its neighborhood. The change of the variable from x_{i-1} to x_i may result in an increase or decrease of the function value. To avoid getting stuck in a local minimum, it is necessary not only to allow ameliorations but also suitable deteriorations. This is done by introducing a temperature parameter *T*, which is decreased every *n* iterations. $E(x_i)$ is accepted as the next current solution if $E(x_i) < E(x_{i-1})$. Otherwise, there are two possibilities for the state of *X*: Either x_i will be accepted with the probability $P(x_i)$ or rejected with $1- P(x_i)$. *P* is calculated according to the Boltzmann distribution

$$P(\Delta E) = \exp\left[\frac{-\Delta E}{k_B T}\right]$$

with $\Delta E = E(x_i) - E(x_{i-1})$ and k_B the Boltzmann's constant. At the end of the algorithm, when *T* is small enough, deteriorations will hardly be accepted and, most of the time, only downhill steps are accepted.

To use *Simulated Annealing* for object recognition, it is necessary to adapt the algorithm by specifying the objective function *E(X)* and the change of the variable *X* from one state to another. Since the gesture is described by its boundary, the value of the objective function can be calculated as the sum of edge information values over all landmark points of the current shape. After placing an initial contour into the image, it is transformed from step to step by choosing random values for translations in x- and y- direction, a scaling and a rotation of the shape. Furthermore, suitable deformations of the current shape are allowed using the *Point Distribution Model* [Cootes00] that was calculated in an offline training phase.

For each step, the cost function is calculated as the sum of edge information values *g* over all landmark points p_i of the current shape:

$$E(X) = -\sum_{i=1}^{n} g(p_i)$$

Assuming that high values in the edge map indicate strong edge information, it is intuitively clear that a contour with no information at any landmarks generates a high function value and that a perfect matching contour produces the smallest possible function value of *E*.

3.3 Object fitting

The outcome of *Simulated Annealing* is a rough approximation of the true gesture. This approximation now has to be fitted to the real contour. Using an *Active Shape Model* [Cootes00], this fitting is done as the next step in the recognition process. Here, we iterate toward the best fit by examining an approximate fit, locating improved positions for all landmark points of the gesture, then recalculating a valid contour by using the underlying *Point Distribution Model*. The outcome of this process is the true boundary of the hand as seen in figure 6. We are now able to derive typical features of the pointing gesture like the position of the forefinger tip or the center of gravity of the hand to calculate a position and orientation of the gesture in three dimensions.

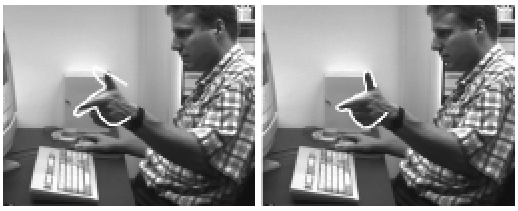


Figure 6 – Initial contour and fitted contour

3.4 3D Pose estimation

With a calibrated stereoscopic camera system, it is possible to reconstruct 3D coordinates of corresponding image points (e.g. finger tips or the center of the hand) in order to estimate a position and direction of the pointing gesture (see figure 7). For example, the center of the hand P_1 in the left camera image and the corresponding point P_2 in the right camera image are known and the optical axes are not parallel. Ideally, the rays through the camera center point $C_{M,1}$ and P_1 and through the camera center point $C_{M,2}$ and P_2 should intersect. Due to calibration errors and discretization, this does not normally occur. A good approximation is the midpoint of the shortest connection line between the two rays. Let s_1 be the direction of the ray through $C_{M,1}$ and P_1 and s_2 the direction of the ray through $C_{M,2}$ and P_2. Ray 1 is then given by:

$$C_{M,1} + t \cdot s_1 , \ t > 0$$

where $s_1 = P_1 - C_{M,1}$. Calculating the normal vector n_1 of the plane containing ray 1, the intersection of this plane with ray 2 is given by

$$n_1 \cdot (C_{M,2} + t \cdot s_2 - C_{M,1}) = 0.$$

A second point can be evaluated by intersecting the plane containing ray 2 with ray 1. The midpoint P_M of both intersection points then obtains the estimation of the reconstructed 3D coordinates of the center of the hand (see figure 7).

By reconstructing several landmark points lying on the upper boundary of the pointing forefinger, the direction of pointing is reconstructed applying a linear regression on these 3D points. The center of gravity of the gesture is used as the reconstruction of the 3D position. Position and orientation of the hand are used to start the tracking of the pointing hand.

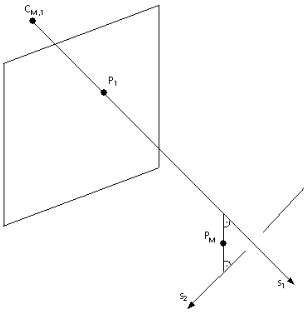
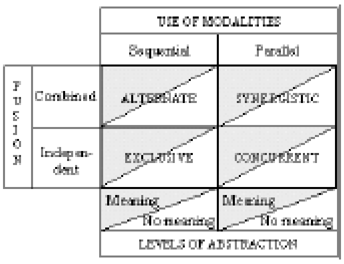


Figure 7 – 3D pose estimation

After being recognized and tracked, the pointing gesture must be integrated with the other modalities. This is what we describe in the next section, starting with some general consideration of the fusion of modalities.

4. MODALITY INTEGRATION

Following a design space in respect to the interaction process, parameters about temporal availability and the fusion possibility of the different modalities must be inferred. These values can have meaning or not, depending on the level of abstraction in which the data is being processed (the representation of speech input as a signal, as a sequence of phonemes or as a meaningful parsed sentence are examples of different abstraction levels). The next figure, taken from [Nigay93], classifies the different situations that should be considered. The shadowed zones are where a multimodal system would figure in.



		USE OF MODALITIES	
		Sequential	Parallel
FUSION	Combined	ALTERNATE	SYNERGISTIC
	Independent	EXCLUSIVE	CONCURRENT
		Meaning No meaning	Meaning No meaning
		LEVELS OF ABSTRACTION	

Figure 8 – Multi-feature system design space

There are two distinct classes of multimodal systems - one integrates signals at the *feature level* and the other at the semantic level. The first one is based in multiple hidden Markov models or temporal neural networks and is adequate for closely coupled and synchronized modalities (e.g. speech and lip movements). The other one is based on an *amodal input* and is appropriated when the modes differ substantially in the time scale characteristics of their features (e.g. speech and gesture input) [Wu99].

In the first phase of EMBASSI, we use semantic fusion with amodal input, considering the visual modalities in a so-called *late fusion* state. (The case of lip-reading, for instance, is considered visual, but is related to the perception of speech and requires an *early fusion* process).

Since the speech modality is out of the scope of this paper, we are giving emphasis to how the gesture modality will influence the global interaction. We will also include the gaze mode of interaction, considering that it has a similar, even if more restricted, purpose – the selection of devices.

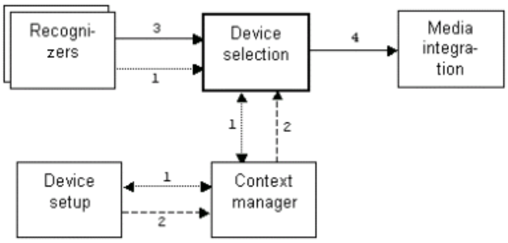
4.1 Device selection

Similar to the speech modality, just after being recognized, the visual modalities must be analyzed in order to send valid information to the integration component. Due to the restrictiveness of this analysis component, it was called the device selection module.

In our approach, the modality integration is done in a two-step process. The first step consists of a common analyzer for two recognized modes, gesture and gaze, with the advantage of a *mutual disambiguation* possibility. This procedure constitutes a kind of pre-fusion mechanism occurring just before the global fusion with the speech modality. After being processed, the results are references to devices, in a whole scenario of several devices ("living room scenario"). Together with the others, these modalities play an important role in the overall multimodal interaction process, yielding partial solutions to difficult problems of natural language anaphora.

The goal is to obtain a selected device, or a set of them in case of ambiguity, based on the vector received from the recognizer components. This is done through cooperation with other modules on several levels, one of them being the establishment of a *world coordinates system*.

The components involved are the gesture and gaze *recognizers*, an *embassi sensor* responsible for setting-up the system, and a *context manager*. The following picture depicts the flux of information. We are considering the use of a context manager to store the spatial characteristics of the devices.



```
1 = World Coordinates System (WCS)
2 = Spatial information of the devices
3 = Gesture and gaze vectors
4 = Selected device(s)
```

Figure 9 – Cooperation for device selection

After being "agentified" by the EMBASSI KQML-based infrastructure, the components communicate with each other by using appropriated performatives. The following illustration represents an example of communication between the recognition and analyzer modules: - with a KQML message and after calculating the position and direction, the gesture (or gaze) recognizer informs the gesture analyzer with spatial and temporal information.

```
(tell :sender GestureRec
      :receiver DeviceSelection
      :reply_with Ge-Rec_Msg1.0
      :ontology spatialOntology
      :language XML
      :content (<event time="23:59:59:321">
                  <vector x="1" y="2" z="2"
                          dx="0.4" dy="0.75"
                          dz="0.3"
                   actor="hand" />
            </event> ) )
```

Figure 10 – Agents communication example

The core operation of the device selection component is based on the ray interception step, detecting the intersections between the line that represents the hand pointing gesture and the devices' surfaces. The same procedure could be applied to the gaze direction.

Due to precision restrictions, we are considering several lines within a probability space. That consists of a *space of ambiguity*, which can be visually represented by a cone in space, formed from the user position (P_1) to the *device zone*.

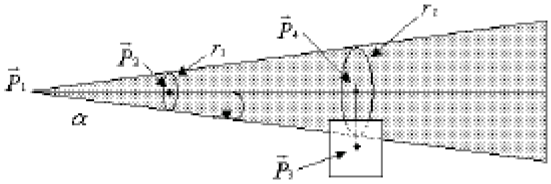


Figure 11 – Space of ambiguity

Here, an angle a is introduced to form a cone, within which the intersections are calculated. The further the object is from the user's position, the greater the imprecision that can occur. In the picture, we can see two positions, (P_2 or P_4) with different coefficients of ambiguities represented by the rays r_1 and r_2, respectively. The objects in P_3 can be detected due to the great ambiguity in position P_4.

Therefore, it is often possible to have more than one selected device. The first way to solve this is through fusion with the gaze. The fusion mechanism is done by "time-proximity", which is feasible due the basic system characteristic of event time-stamping: supposing that P_t represents a pointing act in an instant t, and that it occurs during some ?t. The relevant points in time for this act are the beginning (t) and the end (t+?t) where P_t = $P_{t+?t}$. It is in this interval of time (?t) that the fusion is performed.

In case of ambiguity persistence, it will be solved with the help of speech. The input of the media integrator in Fig. 9 is a graph of device probabilities.

4.2 Actions

Currently, we are using only the deictic gesture. It can be seen like a pen in the 3D space, pointing and performing linear movements, particularly used to interact with elements in a display like a big TV set.

Due to tracking and segmentation difficulties in natural environments, and in order to build a robust system, we have restricted the vocabulary, namely in that which concerns the rapidity of movements. Optimistically speaking, this is a good characteristic due the fact that the vocabulary set must be small to be accepted and learnable by the user. The next table summarizes the possible actions we are implementing.

Motion	Meaning	Action examples
Fix	- select	- with "turn this on"
Fast change	- origin/ destination	- with "put that there"
slowly left slowly right	- horizontal scrolling	- video forward - TV menu items
slowly up slowly down	- vertical scrolling	- volume - TV menu items
forward	- activate	- turn on
backward	- deactivate	- turn off

Table 1 – Gesture vocabulary

Our vocabulary set contemplates the "on" and "off" actions, the left, right, up, down movements (to select a set of options vertically or horizontally distributed), and another more aleatory "from-to" pointing.

5. CONCLUSIONS

We have briefly presented an ambitious project addressing innovative methods of man-machine interaction in non-professional environments of everyday life, such as at home and in the car. Most of the prototypes already developed will undergo improvements in the second phase of the project that is now underway.

Our main focus in this paper was the home environment, concerning the input with gestures in order to complement, e.g. infrared or speech remote controls. We are demonstrating that by using only one feature, the deictic gesture, we can tremendously reduce the recognition task and still have a functional dialogue system.

This natural method of interaction, without the need for markers attached to the user's body, for example, remains a very difficult task due to precision problems. Two ways to minimize these problems are to use (visual or auditory) feedback and the benefits of multimodality, which enjoys a high level of preference among users, as many research studies (e.g. [Chu97]) report.

ACKNOWLEDGEMENTS

The work of the first author is partially funded by the Portuguese Ministry of Science and Technology (MCT), through the Information Society Operational Program (reference PRAXIS XXI/BD/20095/99).

The EMBASSI project is supported by the German Ministry of Science and Education (BMBF) as a focus project on Man-Machine Interaction (signature 01IL904).

REFERENCES

[Cootes00] Cootes T., Taylor C.: "Statistical Models of Appearance for Computer Vision", University of Manchester, http://www.wiau.man.ac.uk, 2000.

[Cost99] Cost R. et al.: *"An Agent-Based Infrastructure for Enterprise Integration"*, 1st International Symposium on Agent Systems and Applications / 3rd International Symposium on Mobile Agents, IEEE Computing Society, 1999.

[Chu97] Chu C., Dani T., Gadh R.: *"Multimodal Interface for a Virtual Reality Based Computer Aided Design System"*, Proceedings of IEEE International Conference on Robotics and Automation, 2:1329-1334, April 1997.

[Domini97] Domini F. M. et al.: "Reasoning in Description Logics", CSLI Publications, 1997.

[Finin93] Finin T. et al.: "Specification of the KQML Agent-Communication Language", University of Maryland, 1993.

[Franconi99] Franconi E.: *"Ontology!"*, http://www.cs.man.ac.uk/~franconi/ontology.html, 1999.

[Haberäcker95] Haberäcker P.: "Praxis der Digitalen Bildverarbeitung und Mustererkennung", Carl Hansen Verlag, 1995.

[Hildebrand00] Hildebrand A., Sá V.: *"EMBASSI: Electronic Multimedia and Service Assistance"*, Intelligent Interactive Assistance and Mobile Multimedia Computing – IMC2000, Rostock-Warnemünde, Germany, November 9-10, 2000.

[Jacob96] Jacob, R. J.: "The Future of Input Devices", ACM Computing Surveys 28A(4), December 1996.

[Jeon00] Jeon H., Petrie C., Cutkosky M.: *"JATLite: A Java Agent Infrastructure with Message Routing"*, IEEE Internet Computing, March-April 2000.

[Kohler96] Kohler M.: "Vision Based Remote Control in Intelligent Home Environments", 3D Image Analysis and Synthesis '96, Erlangen, 18-19 November, 1996.

[Martin99] Martin D., Cheyer A., Moran D.: *"The Open Agent Architecture: A Framework for Building Distributed Software Systems"*, Applied Artificial Intelligence: An International Journal", 13(1-2): 91-128, January-March 1999.

[Metropolis53] Metropolis N. et al.: *"Equation of state calculation by fast computing machines"*, Journal of Chemical Physics, 21:187-1092, 1953.

[Milota95] Milota A., Blattner M.: *"Multimodal interfaces with voice and gesture input"*, IEEE International Conference on Systems, Man and Cybernetics, 3:2760-2765, 1995.

[Nigay93] Nigay L., Coutaz J.: *"A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion"*, Proceedings of InterCHI'93, Conference on Human Factors in Computing Systems, ACM, 1993.

[Oviatt97] Oviatt S., DeAngeli A., Kuhn K.: *"Integration and Synchronization of Input Modes during Multimodal Human-Computer Interaction"*, Proceedings of Conference on Human Factors in Computing Systems: CHI '97, 415-422, ACM Press, New York, 1997.

[Pirlot96] Pirlot M.: "General local search methods", Elsevier Science B.V, 1996.

[Sonka98] Sonka M., Hlavac V., Boyle R.: "Image Processing, Analysis and Machine Vision", PWS Publishing, 1998.

[Wu99] Wu L., Oviatt S., Cohen P.: *"Multimodal Integration – A Statistical View"*, IEEE Transactions on Multimedia, 1(4):334-341, 1999.

[W3C00] W3C – World Wide Web Consortium: "*Extensible Markup Language*", http://www.w3.org/XML, 1997-2000.