

Computer Vision based Interaction Techniques for mobile Augmented Reality

Christian Reimann

*Universität Paderborn, C-LAB
Fürstenallee 11, 33102 Paderborn
Tel. 05251 / 606118, Fax. 05251 / 606065
Christian.Reimann@c-lab.de*

Prof. Dr. rer. nat. Volker Paelke

*Juniorprofessor für 3D Geovisualisierung und Augmented Reality
Universität Hannover, Institut für Kartographie und Geoinformatik
Appelstraße 9a, 30167 Hannover
Tel. 0511/7622472, Fax. 0511/7622780
Volker.Paelke@ikg.uni-hannover.de*

Abstract

Computer vision is employed in the user interface of many VR and AR applications, especially for tracking. This paper reviews the domain of inside-out vision, in which a camera is manipulated to effect some interaction, and reports on initial experiments in the domain. Inside-out vision is of particular interest since cameras and camera equipped devices are already widely used in VR and AR systems, suggesting their use as interaction devices. Currently the selection of vision-based interaction techniques seems to be mostly ad-hoc, driven largely by the available toolkits and demo programs. Therefore, this paper analyzes the design space of inside-out vision in order to structure existing work, to identify possible alternative techniques and to inform future design choices of user interface designers. A system for the implementation of simple motion gestures that are applicable to a wide range of interactive tasks is presented as a first step towards generally applicable vision-based interfaces.

Keywords

computer vision, virtual and augmented reality, interaction techniques, motion gestures

1 Motivation

The widespread availability of camera equipped PDAs, smartphones and similar devices has led to their use as interaction devices in a number of virtual and augmented reality applications. Due to the formfactor of the devices, into which the camera is embedded, these are typically used in an inside-out setup. This means that the camera itself is manipulated in space to effect some interaction. The videostream captured by the camera is analyzed to derive high-level interaction events that control the application.

The additional input mechanism available on the device (e.g. buttons and sometimes the touch screen of a PDA) can be combined with the camera input to create more complex composite interaction techniques. So far, such interaction techniques have mostly been created on an ad-hoc basis by computer vision experts for use in technology demonstrators. Reuse has taken place largely based on availability, e.g. techniques used in publicly available demo programs have sometimes been reused in other programs based on implementational convenience, not on informed choices in the user interface design. Currently, little is known about the usability of inside-out vision (IOV) techniques, no libraries exist, and the exploration of IOV techniques and their application is still at an early stage. As an aid for future development we have structured the design space of IOV techniques. Such approaches have proven to be useful for the general study of interaction techniques in the past (e.g. [CMR91]).

In the following sections we identify the influences and constraints inherent in the concept of inside-out vision as a tool for developers of new interaction techniques as well as for user interface designers who want to apply IOV in their applications. We explain a number of simple interaction techniques that can be implemented in the IOV setup and introduce a system that allows to create composite motion gestures for more complex interaction tasks.

2 Influences and Constraints

The constraints that influence the design of interaction techniques based on inside-out vision can be separated into two categories: those that are due to the sensor and those that are due the human user and his environment.

Card's design space of input devices [CMR91] is based on the physical properties that are used by input devices (absolute and relative position, absolute and relative force, both in linear and rotary form) and composition operators (merge, layout, connection). Interaction techniques are constructed by combining several physical properties accessible to sensors through composition operators and mapping the resulting input domain to a logical parameter space suitable for applications. In

order to integrate IOV into this framework it is necessary to identify what properties can be sensed using a camera in the inside-out configuration. Differing from direct physical sensors the input properties must be extracted from noisy high-bandwidth image sequence. Table 1 shows what properties can be derived from image sequences. In practice, the requirement of interaction techniques to operate in real-time with minimal lag is often in conflict with the high processing requirements of computer vision techniques, especially if local processing on a mobile device is intended, so only a subset of these possibilities can be used.

	linear	rotary
Absolute position	possible if point of origin is provided	possible if point of origin is provided
Relative position (movement)	possible	possible
Absolute force	<i>not possible</i>	<i>not possible</i>
Relative force	<i>not possible</i>	<i>not possible</i>

TABLE 1: POSSIBLE INPUT PROPERTIES

Absolute position: Absolute positioning is only possible if a point of origin is provided that allows establishing a spatial relation between the environment and the image captured by the camera. A possible solution that allows for fast and relatively precise positioning is the use of markers/fiducials at known positions. Several software packages support 6DOF positioning using cameras and markers (e.g. ARToolkit [ART04]).

Alternative “marker-less” approaches (e.g. [NY04][SB02]) use a geometric model of the environment instead of markers. The main advantage is that no artificial markers in the environment are required, making them more appropriate for mobile and wearable systems. However, “marker-less” approaches are often more sensitive to environmental effects like changes in lighting, depend on the structure and “content” of the environment and the more complex image and model processing typically results in higher latency in the interaction. If no geometric model of the environment can be provided in advance, as is typically the case in mobile applications, it is necessary to construct the model on the fly, which is an active area of research ([SB02]).

These absolute positioning techniques can be used to determine the position and orientation of the IOV camera in all six degrees of freedom (6DOF), thus proving access to all three linear and three rotary degrees of freedom in Card’s design space. However, the precision of the information can vary significantly.

The detection of the presence/absence of objects also provides useful information that can be exploited in IOV. Because of its similarity to button-presses in conventional interfaces it is grouped under absolute positioning, although it does not require a point of origin. Again the detection of prepared objects like barcodes and

markers is simpler than that of generic real-world objects, but solution exists for both.

Relative position (motion): Motion can be sensed in three linear (x, y, z) and three rotary degrees of freedom by processing the incoming video stream. No point of origin is required for the detection of motion from image sequences, allowing the use in unprepared environments. However, in practice the precision that can be attained in unprepared environments is limited. While 2DOF motion detection is suitable for the limited processing power of current mobile devices (and for which special purpose hardware used in optical mice and video compression could eventually be adapted) 6 DOF motion tracking is much more difficult and computationally intensive. If the environment is specially prepared, e.g. by placing and tracking fiducials, processing on mobile devices becomes possible (e.g. [Wag03]), otherwise the processing often has to take place on more powerful hardware, using a client-server approach that can introduce problematic latencies.

Absolute and relative force: Information about force can not be extracted from image data without additional transducer hardware.

To identify the influences and constraints introduced by the human user and his environment the following questions must be considered when constructing an IOV interaction technique:

- Is the required positioning and motion of the camera possible for the user? This refers both to constraints on possible positions due to user anatomy, as well as to physical constraints imposed by the surroundings (e.g. use in an office vs. use on a plane).
- Is the required positioning and motion of the camera comfortable for the user? IOVs will only be used if users prefer them to alternative techniques, therefore criteria like fatigue, precision and speed must be considered.
- Are the required positioning and motion of the camera acceptable? For most applications IOVs will not be used if the required motions are embarrassing in public.
- Are the required input properties sensible with the available hardware? As discussed previously, only a subset of the theoretically available input properties can be used in practice. It has to be ensured that the required input properties can be provided with appropriate accuracy, speed and latency under the conditions of use.
- Is it possible to differentiate intentional inputs from unintentional camera movements? To avoid the “midas touch”-problem means to distinguish input from unintentional noise must be provided, e.g. by explicit input confirmation.

- Is the mapping from inputs to interaction events unambiguous?

3 The usage of Inside-Out Vision

The following discussion of (possible) uses of IOV is structured according to the interaction tasks select, position, quantify and gesture. It is based on the popular taxonomy of Foley et al. [FDFH96]. Due to the characteristics of IOV we have replaced the text task in the original taxonomy with a generic gesture recognition task. Interaction tasks specify what a users can try to achieve in an application on an abstract level - for the implementation in an actual user interface a concrete realization in the form of an interaction technique is required. The following section will present exemplary interaction techniques based on IOV for a specific interaction task.

3.1 Selection

The select task refers to symbolic selection from a set of options. Different approaches to symbolic selection are enabled by IOV: An interesting approach based on the tangible computing paradigm can be used if the set of options can be represented by associated physical objects. Then selection can be effected simply by placing the camera so that the object is in the camera's field of view. Examples for this include the use of barcodes which are easy to recognize even on performance limited hardware, the use of more complex markers (that also enable more complex tasks) or the use of geometry or image based object recognition.

While selection based on physical objects has interesting properties for some applications, it often can not be used either because the application has to operate in unprepared environments or because the set of options is too large or changes dynamically. In these cases approaches based on virtual representations of the set of options similar to menus in a desktop interface can be used. Figure 1 shows the use of "Kick-Up-Menus" ([PRS04]). Here simple motion detection is used on the image sequence provided by a camera facing downward from a PDA or Smartphone to detect "kicking" movements of the user's feet. When a collision between the user's "kick" and an interaction object shown on the screen of the mobile device is detected, a corresponding selection event for the application is generated. As Figure 1 shows "Kick-Up-Menus" can be structured hierarchically to enable access to large sets of options.

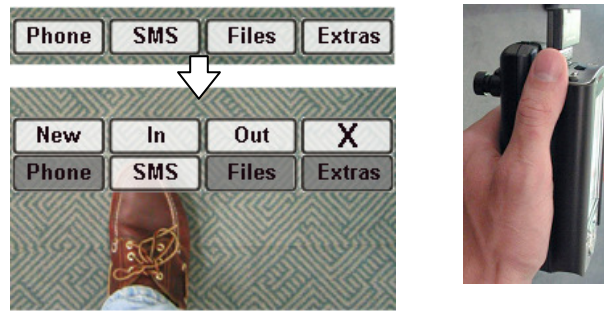


Figure 1: Kick-Up-Menus and PDA with IOV camera setup

A common selection task in 3D applications is spatial selection. While spatial selection of physical objects can be realized as described previously, spatial selection of virtual objects typically has to be constructed from one or more positioning tasks as described in the following subsection.

3.2 Position & Movement

Different from desktop environments, where positioning usually refers to xy-positioning using the mouse, VR and AR applications often require positioning with up to 6 degrees of freedom. As discussed in chapter 2 absolute positioning in 6DOF is possible using IOV if a point of origin is provided.



Figure 2: The Mozzies game on the SX1 smartphone with IOV camera

In these cases the 6DOF positioning data provided by the computer vision algorithm can be mapped (possibly through some transfer function) to the application domain. To provide positioning data with adequate precision and lag most existing applications use marker based approaches, e.g. ARToolkit [ART04] and Sony's Cybercode [RA00]. If not all 6DOF are required simpler, faster and more robust algorithms can be used that are suitable for mobile devices. Figure 2 shows the Mozzies game on the Siemens SX1 smartphone that uses simple 2D motion detection and a crosshair for 2D xy-positioning.

3.3 Quantification

The quantify interaction task is used to specify numeric values as input parameters to the application. In mouse-based interfaces potentiometer, slider and scrollbar widgets are often employed for this task. A similar approach is used in Spotcodes [Spot]. Spotcodes is a system based on circular markers from which rotation information can be derived. Interaction techniques are provided for the specification of rotation angles and values. Sometimes a direct mapping from the input to the application domain is possible without the need for widgets as an intermediary. In this way the pitch angle of the camera has been used to control scrolling (instead of a scrollbar widget). Figure 3 shows ARSoccer, a mobile soccer application [GPR04]. Here the direction and speed of a motion vector generated by a kicking foot are used to control a simple soccer game, resulting in an intuitive mapping between the input and application domains. The interaction techniques of AR-Soccer are now used in a commercial game implementation [KR].

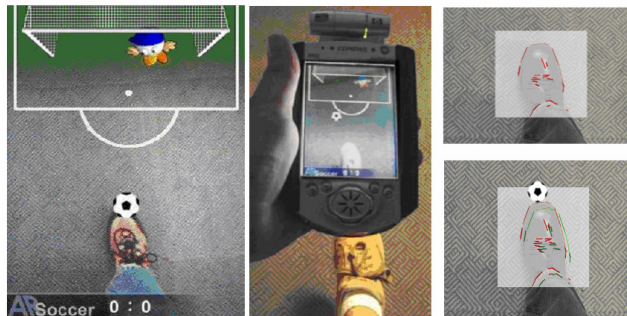


Figure 3: The AR-Soccer Application with simple edge tracking

3.4 Gesture

Gestures refer to the symbolic interpretation of camera motion. This can range from simple yes/no gestures over a simple gesture vocabulary (similar to mouse gestures in some applications) to complex sign languages. Here a careful tradeoff between the learning required of the user to become proficient with the gestures, the requirement for unambiguous gesture identification, the required processing power and the expressiveness of the gesture set is required. So far most applications use only simple gestures but techniques and gestures developed for the domain of head-gestures (that shares many properties with IOV (e.g. [K01])) could in principle be adapted to IOV. In the following section we introduce a system that allows to compose simple motion gestures for a variety of applications.

4 IOV – A Gesture Recognizer

Most existing gesture recognition systems (like [Da01] or [Mo04]) monitor the user, or the relevant part of the user, to observe the performed gestures and control the application. To achieve this the user's environment is typically equipped with sensors, e.g. cameras. While impressive results can be achieved indoors in controlled spaces with appropriate hardware such an approach is not possible in most outdoor and mobile applications (e.g. on factory floors), because an adequate instrumentation of the environment becomes impossible.

In many Augmented Reality applications a video-see-through approach is used, in which the user is wearing a camera on his head (in combination with an HMD). Several AR applications have exploited the information provided by this camera to perform gesture recognition, using the camera view to monitor the user's hands and recognize hand or finger gestures like grasping and pointing. Moeslund et al. for example have shown a system that recognizes pointing and command gestures [Mo04]. The FingARtips system [Bu04] detects the users hand and two fingers by using special fiducial markers attached to a glove. With this approach it is possible to detect gestures like grasping or clicking on a virtual 2D WIMP interface. The system Vampire [Ba04] detects the user's finger, which can then be used to click on a WIMP UI or point to objects.

For all these approaches the user has to look at his fingers or hands, as otherwise the recognition would not work. In an IOV system the camera is attached to a mobile part that should be sensed, avoiding the necessity to look at something special. In a video-see-through AR application, with the camera attached to the users head, IOV allows to recognize simple head-gestures like nodding or shaking.

Another example where IOV can be used is a mobile phone equipped with a camera. Here the sensor is also directly attached to the device, which should be monitored.

In this section we present a gesture system using IOV, following the idea of simple light-weight recognizer for mobile applications. For the IOV Gesture Recognizer system the following goal for the system design were defined:

- **Fast, lightweight system:** As the recognizer should run on small mobile devices like cell phones in parallel to an application (leaving enough processor time for the application itself) it must be simple and neither CPU nor memory intense
- **Robust recognition:** Due to the low quality cameras in current mobile phones and PDAs the recognition should work even with low-resolution images and still allow for robust gesture detection.

- **Flexible definition for gestures:** To maintain high flexibility the gestures should be definable by the application.
- **Continuous recognition:** The gestures should be recognized out of a continuous flow of movements without telling the system when a new gesture starts or ends.

4.1 Architecture

To meet the design goals the system follows the architecture shown in figure 4. The system is split in four parts. In the first part “Motion-Detection” the image-stream provided by the camera is analysed and a two-dimensional movement vector is derived. The resulting 2d-vector is passed to the next part “Classification”, where it is discretised using a set of basic movements. This basic movement, together with predecessors is then compared with gesture templates from a template store.

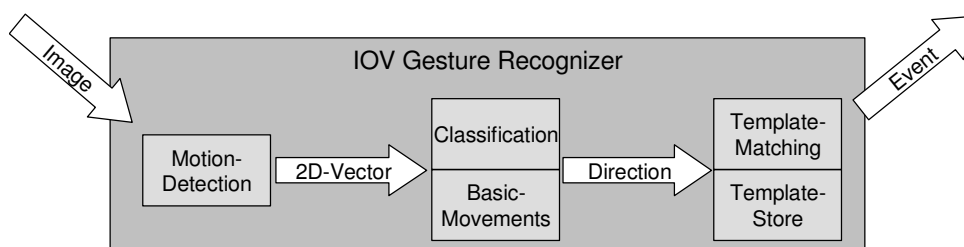


Figure 4: Architectural Overview

For the “Motion-Detection” any computer vision algorithm which results in a 2d-vector can be used, representing the overall movement of the image. In our case a simple optical flow algorithm was implemented. The simplification of a 6 DOF tracking to a 2d-movement significantly enlarges the robustness of the recognizer. For an AR application with computer vision based tracking it is possible to use in-between results from the tracking system, to avoid additional image analysis.

In the “Classification” part the 2d-vector is mapped to a basic movement. Here a set of eight movements was used, as shown in figure 5, mapping the vector to the characters a to h. For other experiments with very low accuracy images a set of just four basic movements was used.

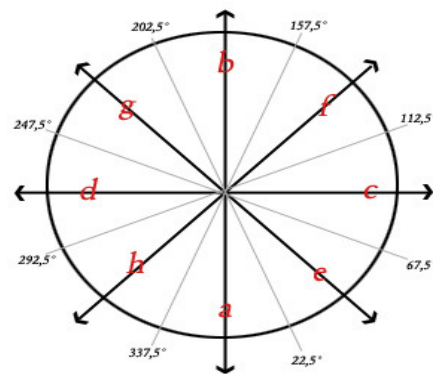


Figure 5: Classification & Basic-Movements

For the discretisation itself different variations were developed. Variation 0 simply takes the movement from image to image and approximates the direction by the classification. Variation 1 also takes the movements speed into account, by using multiple basic movements of the same type to express the vectors length. Variation 2 splits the movement into several sometimes different basic movements, while variation 3 additionally aims to minimize the overall error in taking the error from previous image-discretisation into account.

The “Template-Matching” searches the incoming stream of basic movements for completed gestures in the template store. Depending on the intended gesture and the needed flexibility there are different possibilities to define a gesture. The “Simple-Gesture” is just a sequence of basic movements, which have to be exactly recognized for the gesture to be completed. The “Loop-Gesture” is very similar to the “Simple-Gesture” but does not have to start at the beginning of the sequence, but at any position. The “Grammar-Gesture” is the most flexible way to define a gesture and allows for the most robust detection. In addition to the sequence of basic movements the quantity of each basic movement can be defined with a lower and an upper boundary. Also tolerant equality can be defined, e.g. the sequence should have (approximately) the same number of up- and down-movements in it. For example a simple nod-gesture could be:

- (a,b) for the basic movements
- (2, 2) and (10,10) for the boundaries
- (x, x) as a and b should occur nearly equal times

For the “Grammar-Gesture” it is also possible to set a flag for looping, to state whether the gesture can start at any position of the sequence, or has to start at the beginning.

4.2 Results

The IOV Gesture Recognizer was evaluated by different sets of movements in a laboratory setting, where first the movement was recorded, and then fed into the recognizer. Although this does not fully substitute a “real-world” test, it ensures the reproducibility of the test and allows comparing the different variations. Figure 6 show the results of the test. On the y-axis the amount of recognized gestures is placed. The recognized gestures are on the left bar, the middle bar shows the amount of error recognitions (a gesture was recognized, which was not made) and the right bar shows the amount of not recognized gestures. On the x-axis are the different variations of the recognition system. At first the variations 0, 1, 2, 3, then 0, 1, 2 with a modification in the input modality and last the variations with a modified base vector length.

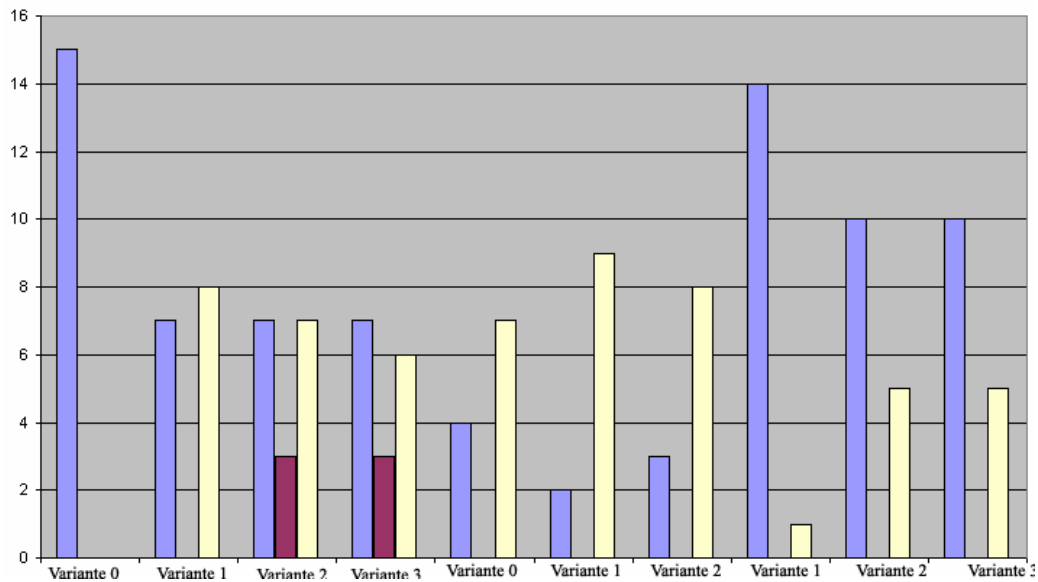


Figure 6: Test Results

The results show that the variations with the lowest mathematical error in the discretisation (variations 2 & 3) perform significantly worse than the simple versions 0 and 1. Surprisingly the best results were achieved using variation 0. Here the recognition was very robust and had the fewest errors.

Informal application tests with a mobile phone were conducted using only 4 basic movements, as the movement detection with the available cameras was too inaccurate for eighth movement directions. But even with the reduced set of movements the recognition was also very robust and easy to use. The performance (only the recognition rate) was also very promising: We achieved approx. 15 analysed images per second on a Nokia 6630 phone, with the camera speed as the limiting factor.

5 Outlook

Work on IOV based interaction techniques is still at an early stage. We have provided an overview of the available design space and illustrated it with examples. Several areas are of interest for future work: On the theoretical side the combination of IOV with other input modalities is an interesting domain to explore. PDAs and smartphones typically provide a number of buttons or even a touch screen. Using Card's design space the resulting possibilities can be explored systematically. The construction of specialised IOV input devices consisting of a camera and extra sensors could also be interesting. For example, pressure sensors could be added to make the properties of relative/absolute force accessible to cover the complete design space.

On the practical side the viability and usability of IOV based interaction techniques is best explored by experiment. However, computer vision is a hard problem even with existing libraries (e.g. [Spot]). A problem with many existing computer vision algorithms is that they were designed for other purposes and that "intermediate results" that can often be exploited in IOV based interaction techniques, are not accessible to the user. The adaption of computer vision techniques to the requirements of designing IOV interaction techniques is therefore necessary. Possible hardware support for these computer vision techniques is another interesting research problem. A general approach and standardized description for motion gestures would also help developers in the creation of new IOV techniques.

Acknowledgments

We would like to thank all colleagues and students that have contributed to the hardware and software developed for the various prototypes. Also the participants of the 2005 IEEE workshop on New Directions in 3D User Interfaces who contributed to the discussion on a general framework for IOV interfaces.

References

- [ART04] ARToolkit: http://hitl.washington.edu/research/shared_space, accessed 28. Jan. 2004
- [Ba04] Ingo Bax, Holger Bekel, Gunther Heidemann, Helge Ritter. Visual Learning of Motion Behaviours and Classification of Spatiotemporal Events; Specialized Model for Hand Context. Report. Bielefeld, Juli 2004
- [Bu04] Volker Buchmann, Stephen Violich, Mark Billinghurst, Andy Cockburn. FingARTip - Gesture Based Direct Manipulation in Augmented Reality. GRAPHITE 2004, S. 212- 221, Singapur 2004

- [CMR91] Card, S. K.; Mackinlay, J.D. and Robertson, G.G.: A Morphological Analysis of the Design Space of Input Devices, *ACM Transactions on Information Systems*, Vol. 9 , No. 2, April 1991, pp. 99 – 122
- [Da01] James Davis, Serge Vaks. A Perceptual User Interface for Recognizing Head Gesture Acknowledgements. *PUI Workshop*. Orlando 2001
- [FDFH96] Foley, J. D. ; van Dam, A.; Feiner, S.K. and Hughes, J. F.: *Computer Graphics - Principles and Practice*, Second Edition in C, Addison Wesley, 1996
- [GPR04] Geiger, C.; Paelke, V. and Reimann, C. :*Mobile Entertainment Computing*, *Lecture Notes in Computer Science*, Vol. 3105 / 2004, Springer Verlag 2004, pp. 142 – 147
- [KR] KickReal: <http://www.kickreal.de/>, accessed 28. Jan. 2006
- [K01] Kjeldsen, R.: Head Gestures for Computer Control, *Proc. IEEE RATFG-RTS Workshop on Recognition And Tracking of Face and Gesture*, Van-couver, Canada, July 2001, pp. 61-67
- [Mo04] Thomas Moeslud, Moritz Störring, Erik Granum. Pointing and Command Gestures for Augmented Reality. In *ICPR workshop on visual Observation of Deictic Gestures (Pointing '04)*, Cambridge, August 2004
- [NY04] Neumann, U. and You, S.: Natural Feature Tracking for Augmented-Reality, *IEEE Transactions on Multimedia*, 2004
- [PRS04] Paelke, V.; Reimann, C. and Stichling, D.: Kick-Up-Menus, in: *Extended abstracts of ACM CHI 2004*, Vienna, 2004
- [RA00] Rekimoto, J. and Ayatsuka, Y.: *CyberCode: Designing Augmented Reality Environments with Visual Tags*, *Proc. Designing Augmented Reality Environments DARE 2000*, Elsinore, Denmark, April 2000
- [SB02] Simon, G. and Berger, M-O.: Reconstructing while registering: A novel approach for markerless augmented reality, in: *Proc. IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp.285-294, 2002
- [Spot] Spotcode: <http://www.highenergymagic.com>, accessed 28. Jan. 2006
- [Wag03] Wagner, D.: Porting the Core ARToolKit library onto the PocketPC Platform, *Proc. 2nd IEEE International Augmented Reality Toolkit Workshop*, October 2003, Tokyo, Japan