In the Proceedings of the ECCV 2004 workshop on HCI (LNCS3058), pp. 176–186, Prague, May 2004
 http://www.springer.de/comp/lncs/index.html

An AR Human Computer Interface for Object Localization in a Cognitive Vision Framework

H. Siegl and G. Schweighofer and A. Pinz

Institute of Electrical Measurement and Measurement Signal Processing
Graz University of Technology, Austria
{siegl, gschweig, pinz}@emt.tugraz.at
http://www.emt.tugraz.at/~tracking

Abstract. In the European cognitive vision project VAMPIRE (IST-2001-34401), mobile AR-kits are used for interactive teaching of a visual active memory. This is achieved by 3D augmented pointing, which combines inside-out tracking for head pose recovery and 3D stereo HCI in an office environment. An artificial landmark is used to establish a global coordinate system, and a sparse reconstruction of the office provides natural landmarks (corners). This paper describes the basic idea of the 3D cursor. In addition to the mobile system, at least one camera is used to obtain different views of an object which could be employed to improve e.g. view based object recognition. Accuracy of the 3D cursor for pointing in a scene coordinate system is evaluated experimentally.

keywords: 3D interaction device, active cameras, real-time pose computation, augmented reality, mobile system, mobile AR

1 Augmented Reality

Augmented reality applications enrich perceived reality by giving additional information. This information is provided by representations ranging from text information and object highlighting to the projection of complex 3D objects. Therefore, this technique is perfectly suited as a visual aid for medical and military purposes, for entertainment, for assembly processes or for engineering design or for interactive teaching of visual active memory described in this paper.

Existing AR applications are too limited by restricted mobility and insufficient tracking (head-pose calculation) capabilities to be used in fully mobile, potentially outdoor applications.

The mobile augmented reality system (MARS) by Höllerer *et al.* [3] utilizes an inertial/magnetometer orientation tracker(Intersense) and a centimeter-level / real-time kinematic GPS position tracker which is dependent on a base station providing correction signals. The Tinmith system by Piekarski and Thomas [5] is based on GPS for position tracking and on a magnetometer for orientation tracking.

Our AR-kit has been designed for modular and flexible use in mobile, stationary, in- and outdoor situations. We suggest a wearable system which consists of

two independent subsystems, one for video augmentation and 3D visualization, the other one for real-time tracking fusing vision-based and inertial tracking components.

2 AR Components

An AR-kit usually consists of components providing information on the direction of view – i.e. (self-) localization and head pose recovery – such as vision-based tracking or inertial tracking devices and a possibility for the visualization of information – normally a head mounted display (HMD). Besides, a human computer interface is commonly used for the communication with the system providing the augmented information, for instance the PIP introduced in [8]. In figure 1 the sketch of the system designed for the EU Cognitive Vision project VAMPIRE is shown.

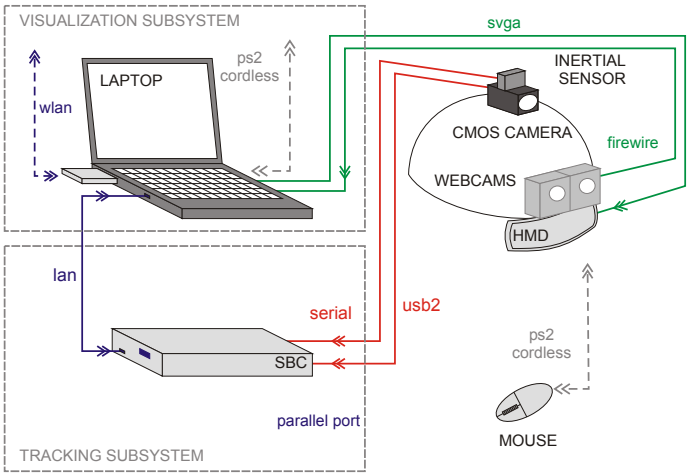
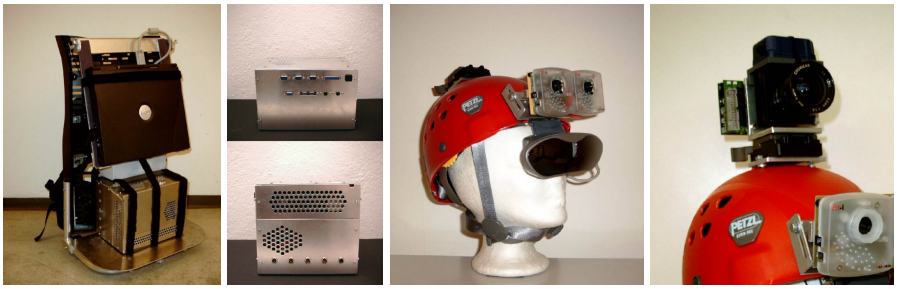


Fig. 1. AR-kit components: A high end laptop and a custom stereo video see-through HMD are employed for visualization. An inertial sensor and a custom high speed CMOS camera are used for tracking.

A laptop is used for rendering information and serving the HMD with the video stream captured from a stereo pair consisting of two fire-wire cameras. A custom CMOS camera and an inertial tracker are used for hybrid tracking. A mouse (buttons only) is used as user interface.

Laptop and single board computer (SBC) are mounted on a backpack (see fig. 2.a and fig. 2.b) and are connected via LAN (direct link). HMD and tracking sensors are mounted on a helmet (see fig. 2.c and 2.d).



(a) (b) (c) (d)

Fig. 2. AR-kit: Our custom stereo video see-through set consisting of Fire-i firewire webcams and an I-visor 4400VPD HMD(a), hybrid tracking unit consisting of custom CMOS camera and an XSens MT9 (b), the backpack with laptop for visualization (c), snapshots of our custom SBC case (d)

2.1 Visualization Subsystem

The laptop applied for visualization has an OpenGL graphic chip (nVidia Quadro) which allows for hardware supplied stereo rendering of the graphics for the custom stereo optical see-through head mounted displays (HMD) consisting of low cost, off-the-shelf components such as two Fire-i firewire webcams and an I-visor 4400VPD HMD (see fig. 2.a). Table 1 lists components and their most important features.

Laptop	Dell Precision M50	Pentium 4, 1.8 Mhz, nVidia Quadro4 500 GoGL
HMD	I-visor 4400VPD	SVGA(stereo), 60, 70 and 75 Hz VESA, 31 degree diagonal field of view
Webcams	Fire-i	IEEE1394, 640 × 480, 30 fps (YUV411), 15 fps (RGB, monochrome)

Table 1. Components for video loop

2.2 Tracking Subsystem

A custom mobile PC system has been assembled for hybrid tracking, as laptops seemed to be not flexible enough to allow for experiments with various hardware for tracking (PCI extensions for e.g. frame grabbers). The system basically consists of a single board computer and a power supply (AC / DC) which also serves all the peripheral hardware of the mobile AR system such as HMD, webcams, CMOS camera and inertial sensor (see tab. 2 for details).

CPU	Intel PIII	1.2 GHz, Intel socket370
Single Board	Advantech PCI-9577FG	USBII, Gigabyte Ethernet
HD	IBM Microdrive	1 GB
CMOS camera	'i:nex'	1024×1024 pixels, 10 bit, USBII
Inertial sensor	Xsens MT9	6 degrees of freedom
Battery pack	custom	13500 mAh, $\approx$ 1 hour system uptime

Table 2. Components for hybrid tracking

This hardware is mainly used for self-localization or inside-out tracking. We implemented a custom Fuga 1000 based CMOS camera ('i:nex') [4] with USB2 interface to gain extremely fast access of small, arbitrarily positioned image portions (see tab. 3) typically used for tracking of e.g. corners or other local features with small support regions. In order to deal with fast movements of the head, vision-based tracking is fused with a commercially available inertial sensor by Kalman filtering.

window side length	**number of windows**	**requests/second**
8	5	2600
8	15	2000
8	25	1300
16	5	2000
16	15	1000
16	25	660

Table 3. Request rates vs. window sizes and number of windows: Request denotes a cycle consisting of window positioning and read-out.

3 VAMPIRE System Design

The project "Visual Active Memory Processes and Interactive REtrieval" (VAMPIRE) aims at the development of an active memory and retrieval system in the context of an Augmented Reality scenario. The AR gear provides the image data perceived from the user (stereo camera), the user's head pose (inside-out tracker) and basically the human computer interface (HCI) defining actions (query, learning, naming of objects). The VAM hierarchically maintains the acquired data (image data, head pose, object locations, gesture interpretation) from all the connected modules, tries to build contextual relations (cup – coffee – sugar) and thus provides data for connected modules (see fig. 3).

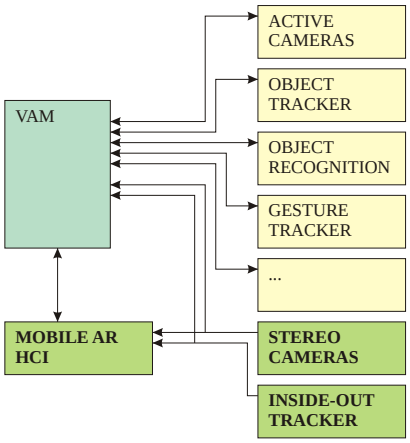
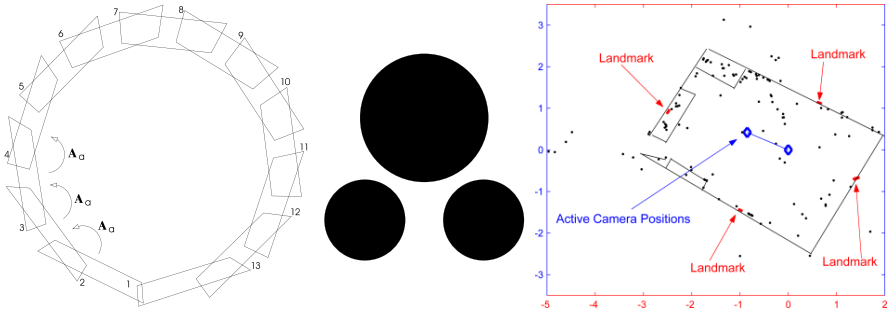


Fig. 3. Vampire System (functional sketch): The visual active memory (VAM) manages and interprets data collected from various modules. The highlighted modules (mobile AR HCI, stereo cameras and inside-out tracker) are for technical reasons physically and functionally closer connected to the VAM than the other modules.

4 VAMPIRE Application Scenario

Within the VAMPIRE project, we aim at mobile indoor applications in unprepared rooms. During an off-line initialization phase the scene is analyzed by recording two image panoramas with a camera (Sony DFW-VL500) mounted on a pan tilt unit (Directed Perception PTU-46-17.5) and extracting a set of artificial landmarks consisting of three disks (see fig. 4.b) and prominent natural corners. This is followed by a sparse reconstruction (see fig. 4.c) of the scene in terms of these landmarks and their scene coordinates using the 'stereo' information provided by multiple recordings [6].

At the moment an artificial target providing corner features (see fig. 5.a) is applied for initialization of the vision-based tracking and the alignment of the coordinate systems (this target defines the origin of the scene coordinate system). Afterwards, the landmarks found during the reconstruction stage are used for online real-time tracking of camera / head pose. Then, the user receives visual feedback using the stereo head-mounted-display (HMD), so that the real scene can be augmented by virtual content. In order to teach the VAM as well as to receive interpretations of the scene and recognition results, several modalities of user-system interaction are required. Pointing at objects in 3D plays an essential role. We found that a 3D cursor [7] would be adequate for most of our applications when an HCI is required for teaching or query, especially in cluttered scenes. In the subsequent sections the concept of the 3D cursor is explained and an application with an active camera is outlined.



(a) (b) (c)

Fig. 4. This figure shows a sketch of the office panoramic view (maximum horizontal rotation of the PTU is 317°) recorded with the camera mounted on the PTU (a). The distance of two different positions of the camera is used as baseline for stereo reconstruction with these panoramas. In order to match these sets of views, a set of targets was applied which do not infer with the natural features (corners) founding the reconstruction of the room. One of these targets is depicted in (b). To the right, an example of a sparse reconstruction of an office scene is shown (c).

5 Pointing at Objects Using a 3D Cursor

The implemented 3D cursor is basically operated by mouse wheel and buttons. It exploits disparity and object size to generate the perception of distance which allows – together with the head pose obtained from the tracking subsystem – to compute an estimate of an objects size and its position in the room (see fig. 5).

This concept is outlined in figure 6. A horizontal displacement of the cursor from the center of the image planes emulates distance. Hence, point correspondence for stereo vision is established manually by manipulating the x-coordinates of a pair of corresponding points with the mouse wheel.

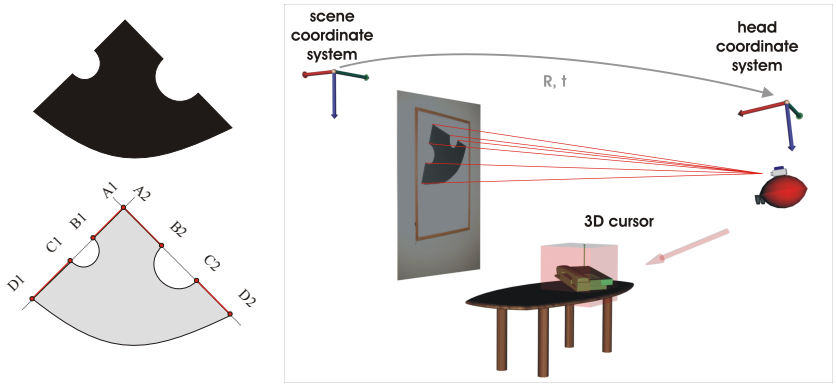
As the cursor is rather placed in the center of the image than in the border regions where the image is stronger affected by lens distortion, it was tried to calibrate the 3D cursor directly (uncalibrated cameras).

For this purpose, the stereo camera was directed towards objects in 3D with known distance (3 times for each distance) and then the cursor was placed on the object in both images provided by the cameras (see fig. 7).

The mean disparity (see fig. 8) obtained from this localization procedure was used to approximate the relation between distance and disparity by a hyperbola $t_z(d)$

$$t_z(d) = \frac{a}{d+b} \tag{1}$$

where a and b are constants determined by a LSE fit ($a = 28.8868$, $b = 1.9464$). In figure 8.b this approximation is compared to the results obtained from



(a) (b)

Fig. 5. This figure shows an artificial corner target [1] which is used as an intermediate step on the way to natural features and to initialize vision-based tracking using corners as features, respectively. The target is identified by the perspectively invariant cross ratio (CR) of the segments on the two intersecting lines. The pose can be calculated by the positions of the corners (a). To the right, the 3D cursor application is depicted. The tracking system processes the corner features of the CR target for self-localization. The selection of the phone with a 3D cursor allows to estimate the position of this object in scene coordinates (b).

the standard stereo vision procedure [2] for the reconstruction of depth (internal and external camera calibration, relative orientation, 2D point correspondence, etc). Due to the manual calibration technique (see fig. 7), the difference between the more precise stereo reconstruction method and the direct method increases with the measured distance (decrease of disparities), as the cursor cannot by placed manually that accurately.

Then, the obtained depth or translation along the z-axis of the camera (perpendicular to the image plane) can be written as

$$\mathrm{t}_{disp} = \begin{pmatrix} 0 \\ 0 \\ t_z \end{pmatrix}. \tag{2}$$

Applying the obtained function, it is possible to compute the approximate position of the object in 3D by

$$\mathrm{t}_{obj} = \mathrm{t}_{h2w} - \mathbf{R}_{h2w}\mathrm{t}_{c2h} + \mathbf{R}_{h2w}\mathbf{R}_{c2h}\mathrm{t}_{disp} \tag{3}$$

where t_{obj} approximates the position of the object, $\mathbf{R}_{h2w}$ and t_{h2w} denote the pose received from inside-out tracking and $\mathbf{R}_{c2h}$ and t_{c2h} the relative pose (obtained from extrinsic calibration of all three cameras) of the Fire-i cameras for the video loop, respectively (see fig. 9).

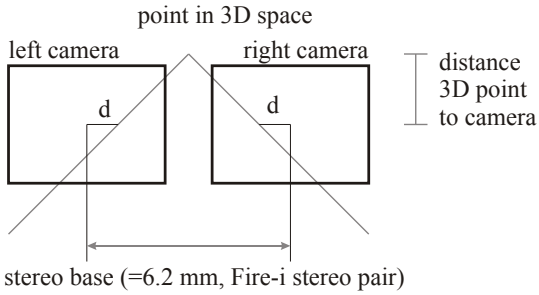


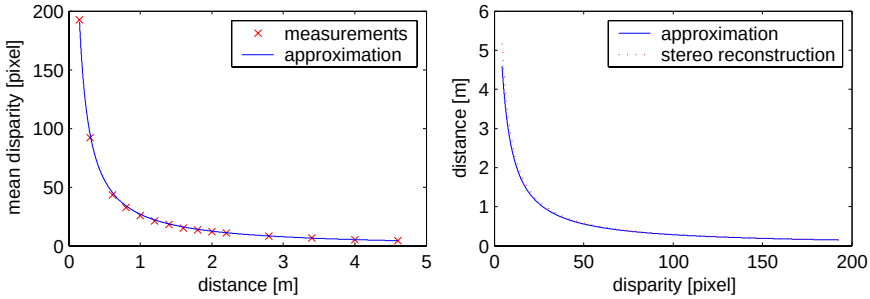
Fig. 6. 3D cursor: A horizontal displacement of the cursor from the center of the image planes emulates distance.

Fig. 7. 3D cursor: An object is focused by the user. The object is perceived in the center of the image displayed by the HMD. In fact, there is a displacement in the images provided by the two cameras. For calibration, both images are displayed next to each other and the cursor is moved from the center parallel to the horizontal axis (same disparity for both cameras and images, respectively) until in both images the same position in the scene is covered. This eliminates the deviations caused by users of the stereo HMD.

Figure 10 shows an experimental verification of our approach. Three users without any experience with our 3D cursor and one well trained user placed the 3D cursor 2 times on the surface of an object (distance=1,2,..,5 m). It can be seen that the achieved accuracy depends on training and distance to the object.

6 A 3D Cursor Application

In our lab we implemented the following setup for the 3D cursor (see fig. 11): In the sparsely reconstructed office, the user points at an object in the room using the 3D cursor. His direction of view determined by inside-out tracking and the distance of the object along this direction are used to estimate the absolute position of the object in the room. This information is sent to the



(a) (b)

Fig. 8. Calibrating the 3D cursor: Approximation with hyperbola (a) approximation vs. standard stereo vision procedure (b)

computer controlling the active camera which was used to create the sparse reconstruction of the room and uses the same coordinate system as the inside-out tracker because of the CR-target. Therefore, it is possible to change the direction of view of the active camera so that an independent second view is obtained (in fact, an arbitrary number of cameras could be used). There is a number of applications to this scenario, for instance:

- acquire a different view of a remote object for view based object recognition
- display enlarged images of remote objects in the HMD, e.g. the title of a book on a high bookshelf.

7 Conclusion

We presented the mobile AR gear which is employed as human computer interface for the cognitive vision project VAMPIRE which tries to model human memory processes in an office environment. Besides, we discussed an AR 3D cursor for pointing and presented a 3D cursor application where the object position determined by head pose and the estimated distance via 3D cursor are used in combination with active cameras. In the future, the integration of pointing gestures will yield a more natural feel for simple scenes than the employment of a mouse as interaction device. Besides, various experiments are performed to find the most suitable shape of the cursor.

Acknowledgement

This research was funded by *VAMPIRE* Visual Active Memory Processes and Interactive REtrieval (EU-IST Programme IST-2001-34401), and by the Austrian Science Fund (FWF project S9103-N04).

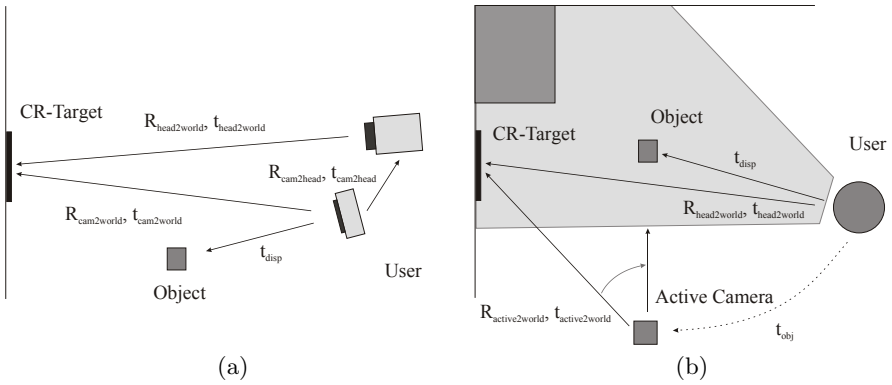


Fig. 9. 3D Cursor in combination with an active camera: Sketch of the required coordinate transforms to compute the external calibration of stereo cameras and tracking camera and to obtain the position of an object in 3D; side view depicting cameras mounted on the mobile Ar kit (a), top view including the active camera and the field of view of the user (b).

References

1. M. K. Chandraker, C. Stock, and A. Pinz, *Real-time camera pose in a room*, 3rd Intern. Conference on Computer Vision Systems, April 2003, pp. 98–110.
2. O. D. Faugeras, *Three-dimensional computer vision : a geometric viewpoint*, MIT Press, 1993.
3. T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallawa, *Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system*, Computers and Graphics **23** (1999), no. 6, 779–785.
4. U. Mühlmann, M. Ribo, P. Lang, and A. Pinz, *A new high speed CMOS camera for real-time tracking applications*, Proc ICRA 2004, New Orleans.
5. W. Piekarski and B. Thomas, *Augmented reality with wearable computers running linux*, 2^{nd} Australian Linux Conference (Sydney), January 2001, pp. 1–14.
6. M. Ribo, G. Schweighofer, and A. Pinz, *Sparse 3d reconstruction of a room*, submitted to: ICPR'04, Cambridge, 2004.
7. H. Siegl and A. Pinz, *A mobile AR kit as a human computer interface for cognitive vision*, Proc WIAMIS'04, Lissabon, 2004.
8. Z. Szalavari and M. Gervautz, *The personal interaction panel - a two-handed interface for augmented reality*, Computer Graphics Forum **16** (1997), no. 3, 335–346.

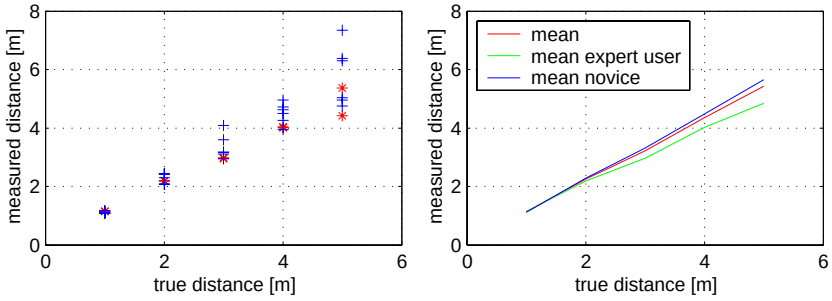


Fig. 10. Experimental verification: Three users (+) without any experience and one well trained user (*) placed the 3D cursor 2 times on the surface of an object (distance=1,2,..,5 m). It can be seen that the achieved accuracy depends on training and distance to the object.

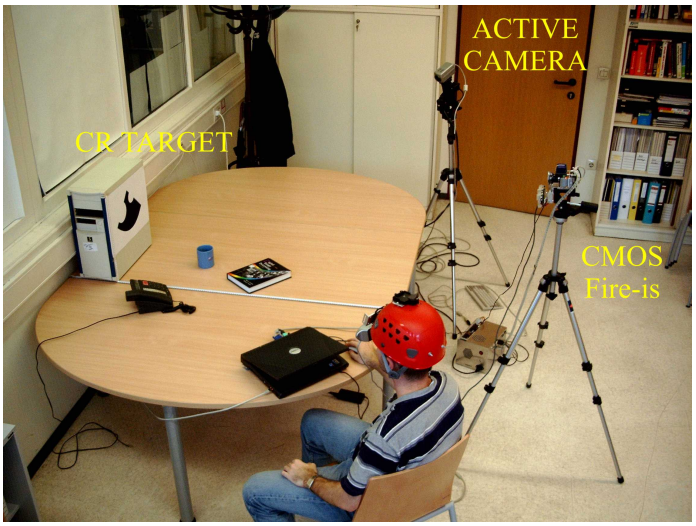


Fig. 11. This figure shows a setup for the verification of our approach. The cameras are mounted on a tripod to eliminate a possible influence of the movements of the user who wears the AR helmet and places the 3D marker next to various objects on the table (error in depth (t_z) < 10 % for a trained user, if $t_z < 4$ m).