A Natural Interface to a Virtual Environment through Computer Vision-estimated Pointing Gestures

Thomas B. Moeslund, Moritz Störring, and Erik Granum

Laboratory of Computer Vision and Media Technology
Aalborg University, Niels Jernes Vej 14
DK-9220 Aalborg East, Denmark
Email:{tbm,mst,eg}@cvmt.auc.dk

Abstract. This paper describes the development of a natural interface to a virtual environment. The interface is through a natural pointing gesture and replaces pointing devices which are normally used to interact with virtual environments. The pointing gesture is estimated in 3D using kinematic knowledge of the arm during pointing and monocular computer vision. The latter is used to extract the 2D position of the user's hand and map it into 3D. Off-line tests of the system show promising results with an average errors of 76*mm* when pointing at a screen 2*m* away. The implementation of a real time system is currently in progress and is expected to run with 25*Hz*.

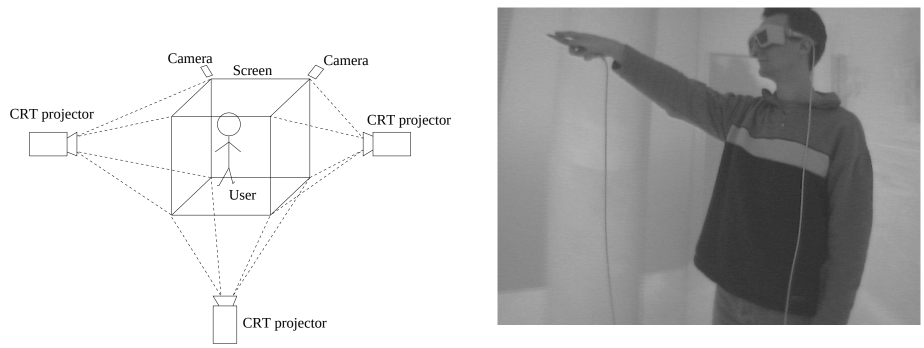
1 Introduction

In recent years the concept of a virtual environment has emerged. A virtual environment is a computer generated world wherein everything imaginable can appear. It has therefore become known as a virtual world or rather a virtual reality (VR). The 'visual entrance' to VR is a screen which acts as a window into the VR. Ideally one may feel immersed in the virtual world. For this to be believable a user is either to wear a head-mounted display or be located in front of a large screen, or even better, be completely surrounded by large screens.

The application areas of VR are numerous: training (e.g. doctors training simulated operations [13], flight simulators), collaborative work [9], entertainments (e.g. games, chat rooms, virtual museums [17]), product development and presentations (e.g. in architecture, construction of cars, urban planning [12]), data mining [3], research, and art. In most of these applications the user needs to interact with the environment, e.g. to pinpoint an object, indicate a direction, or select a menu point. A number of pointing devices and advanced 3D mouses (space mouses) have been developed to support these interactions. As many other technical devices we are surrounded with, these interfaces are based on the computer's terms which many times are not natural or intuitive to use. This is a general problem of Human Computer Interaction (HCI) and is an active research area. The trend is to develop interaction methods closer to those used in human-human interaction, i.e. the use of speech and body language (gestures) [15].

At the authors' department a virtual environment in the form of a six sided VR-CUBE[1], see figure 1, has been installed. A Stylus [19] is used as pointing device when interacting with the different applications in the VR-CUBE (figure 1 b). The 3D position and orientation of the Stylus is registered by a magnetic tracking system and used to generate a bright 3D line in the virtual world indicating the user's pointing direction, similar to a laser-pen.

In this paper we propose to replace pointing devices, such as the Stylus, with a computer vision system capable of recognising natural pointing gestures of the hand without the use of markers or other special assumptions. This will make the interaction less cumbersome and more intuitive. We choose to explore how well this may be achieved using just one camera. In this paper we will focus on interaction with only one of the sides in the VR-CUBE. This is sufficient for initial feasibility and usability studies and expendable to all sides by using more cameras.



a b

Fig. 1. VR-CUBE: a) Schematic view of the VR-CUBE. The size is 2.5 x 2.5 x 2.5*m*. Note that only three of the six projectors and two of the four cameras are shown. b) User inside the VR-CUBE interacting by pointing with a Stylus held in the right hand.

2 Pointing Gesture

The pointing gesture belongs to the class of gestures known as *deictic gestures* which MacNeill [16] describes as "gestures pointing to something or somebody either concrete or abstract". The use of the gesture depends on the context and the person using it [14]. However, it has mainly two usages: to indicate a direction or to pinpoint a certain object. A direction is mainly indicated by the orientation of the lower arm.

The direction when pinpointing an object depends on the user's distance to the object. If an object is close to the user the direction of the index finger is used. This idea is

[1] A VR-CUBE is a comparable installation to a CAVE™(CAVE Automatic Virtual Environment) [5] of the Electronic Visualization Laboratory, University of Illinois at Chicago.

used in [6] where an active contour is used to estimate the direction of the index finger. A stereo setup is used to identify the object the user is pointing to.

In the extreme case the user actually touches the object with the index finger. This is mainly used when the objects the user can point to are located on a 2D surface (e.g. a computer screen) very close to the user. In [20] the user points to text and images projected onto a desk. The tip of the index finger is found using an infra-red camera.

In [4] the desk pointed to is larger than the length of the user's arm and a pointer is therefore used instead of the index finger. The tip of the pointer is found using background subtraction.

When the object pointing to is more than approximately one meter away the pointing direction is indicated by the line spanned by the hand (index finger) and the visual focus (defined as the centre-point between the eyes). Experiments have shown that the direction is consistently (for individual users) placed just lateral to the hand-eye line [21]. Whether this is done to avoid occluding the object or as a result of the proprioception is unknown. Still, the hand-eye line is a rather good approximation. In [11] the top point on the head and the index finger are estimated as the most extreme points belonging to the silhouette of the user. Since no 3D information is available the object pointing toward is found by searching a triangular area in the image defined by the two extreme points.

In [10] a dense depth map of the scene wherein a user is pointing is used. After a depth-background subtraction the data are classified into points belonging to the arm and points belonging to the rest of the body. The index finger and top of the head are found as the two extreme points in the two classes.

In [7] two cameras are used to estimate the 3D position of the index finger which is found as the extreme point of the silhouette produced utilising IR-cameras. During an initialisation phase the user is asked to point at different marks (whose positions are known) on a screen. The visual focus point is estimated as the convergence point of lines spanned by the index-finger and the different marks. This means that the location of the visual focus is adapted to individual users and their pointing habit. However, it also means that the user is not allowed to change the body position (except for the arm, naturally) during pointing.

2.1 Context

In our scenario the distance between the user and the screen is approximately 1-2 meter. Objects can be displayed to appear both close to and far from the user, e.g. 0.1 or 10 meters away, thus both cases mentioned above might occur. However, pointing is mainly used when objects appear to be at least 2 meters away, hence the pointing direction is indicated by the line spanned by the hand and the visual focus.

The user in the VR-CUBE is wearing stereo-glasses, see figure 1 b). A magnetic tracker is mounted on these glasses. It measures the 3D position and orientation of the user's head which is used to update the images on the screen from the user's point of view. One could therefore simply use the position and orientation of the tracker as the pointing direction. However, conscious head movements for pointing has shown to be rather unnatural and will possibly transform the carpal-tunnel syndrome problem into the neck region [1]. Furthermore, due to the Midas Touch Problem [1] it is not as practical as it sounds. However, the 3D position of the tracker can be used to estimate the visual focus and therefore only the 3D position of the hand needs to be estimated in order to calculate the pointing direction. This could then be used to replace pointing devices with a natural and more intuitive action - the pointing gesture.

Estimating the exact 3D position of the hand from just one camera is a difficult task. However, the required precision can be reduced by making the user a 'part' of the system feedback loop. The user can see his pointing direction indicated by a 3D line starting at his hand and pointing in the direction the system 'thinks' he is pointing. Thus, the user can adjust the pointing direction on the fly.

2.2 Content of the Paper

The remaining part of this paper is structured as follows. In section three the method used to estimate the pointing gesture is presented. Section four presents the experiments carried out to test the proposed method. Finally the method and results are discussed in section five.

3 Method

Since we focus on the interaction with only one side we assume that the user's torso is fronto-parallel with respect to the screen. That allows for an estimation of the position of the shoulder based on the position of the head (glasses). The vector between the glasses and the shoulder is called displacement vector in the following. This is discussed further in section 4.2. The pointing direction is estimated as the line spanned by the hand and the visual focus. In order to estimate the position of the hand from a single camera we exploit the fact that the distance between the shoulder and the hand (denoted R), when pointing, is rather independent of the pointing direction. This implies that the hand, when pointing, will be located on the surface of a sphere with radius R and centre in the user's shoulder (X_S, Y_S, Z_S):

$$(X - X_S)^2 + (Y - Y_S)^2 + (Z - Z_S)^2 = R^2 \tag{1}$$

These coordinates originate from the cave-coordinate system which has its origin in the centre of the floor (in the cave) and axes parallel to the sides of the cave. Throughout the rest of this paper the cave coordinate system is used.

The camera used in our system is calibrated[2] to the cave coordinate system. The calibration enables us to map an image point (pixel) to a 3D line in the cave coordinate system. By estimating the position of the hand in the image we obtain an equation of a straight line in 3D:

$$\boldsymbol{P}(t) = \mathbf{P}_0 + t \cdot \mathbf{D} \;\Rightarrow\; \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + t \cdot \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} \tag{2}$$

[2] We use Tsai's calibration method [22] with full optimisation

where $\mathbf{P}_0$ is the optical centre of the camera and $\mathbf{D}$ is the direction unit vector of the line.

The 3D position of the hand is found as the point where the line intersects the sphere. This is obtained by inserting the three rows of equation 2 into equation 1 resulting in a second order equation in t. Complex solutions indicate no intersection and are therefore ignored. If only one real solution exist we have a unique solution, otherwise we have to eliminate one of the two solutions.

A solution which is not within the field-of-view with respect to the orientation of the tracker is eliminated. If further elimination is required we use prediction, i.e. to choose the most likely position according to previous positions. This is done through a simple first order predictor. The pointing direction is hereafter found as the line spanned by the non-eliminated intersection point and the visual focus point. The line is expressed as a line in space similar to the one in equation 2. For a pointing direction to be valid the position of the tracker and the hand need to be constant for a certain amount of time.

3.1 Estimating the 2D Position of the Hand in the Image

The VR-CUBE at the authors' department is equipped with four miniature s-video cameras which are placed in its four upper corners. They may be used for usability studies and for computer vision based user interfaces. The only illumination sources during image capture are the CRT-projectors[3], which are back-projecting images with 120*Hz* on the six sides of the VR-CUBE, see figure 1. This gives a diffuse ambient illumination inside the VR-CUBE which changes its colour depending on the displayed images. The brightness inside the VR-CUBE is determined by the displayed images as well. The average brightness in a 'normal' application is 25 Lux, which is rather little for colour machine vision. The auto gain of the cameras is therefore set to maximum sensitivity, the shutter is switched off, and the maximum opening is used, which results in noisy images with little colour variations.

Hirose *et al.* [9] recently proposed a system to segment the user in a VR-CUBE from the background in order to generate a video avatar. They used infrared cameras to cope with the poor light conditions and simulate a reference background image which is then subtracted from the infrared image containing the user. They get satisfying results.

The simulation of the background also gives information about the illumination the user is exposed to. This could be used, e.g. to estimate an intensity threshold for segmenting the user. However, due to the orientation of the cameras in the VR-CUBE this would be calculation intensive because the cameras' field of view covers parts of three sides, that means a background image has to be synthesised. Furthermore, the image processing is taking place on another computer, thus a lot of data would have to be transfered.

In this project we are using one of the s-video cameras and *a priori* knowledge about the scenario in the camera's field of view:

– Only one user at a time is present in the VR-CUBE

[3] Cathode Ray Tube projector. Each projector consists of three CRTs. One for red, green, and blue, respectively. The VR-CUBE is equipped with ELECTRICHOME MARQUEE® projectors

– The 3D position and orientation of the user's head is known by a magnetic tracker
– The background is brighter than the user, because an image is back-projected on each side and the sides have, especially at the shorter wavelengths, a higher reflectance than human skin
– Skin has a good reflectance for long wavelengths

Figure 2 shows the algorithm to segment the user's hand and estimate its 2D position in the image. Firstly the image areas where the user's hand could appear when pointing are estimated using the 3D position and orientation of the user's head (from the magnetic tracker), a model of the human motor system and the kinematic constraints related to it, and the camera parameters (calculating the field of view). Furthermore, a first order predictor [2] is used to estimate the position of the hand from the position in the previous image frame. In the following we will, however, describe our algorithm on the entire image for illustrative purposes.

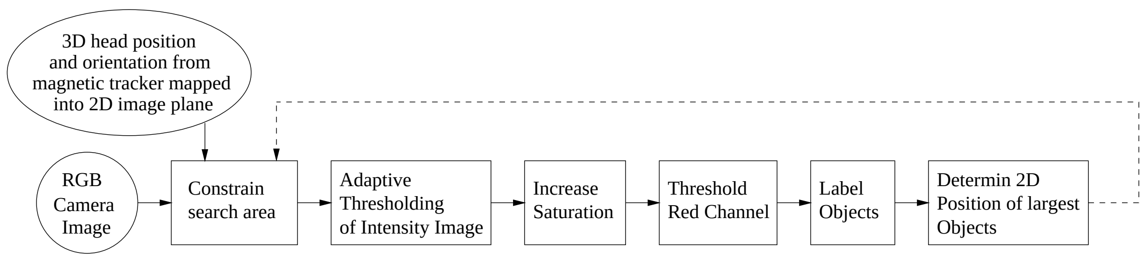
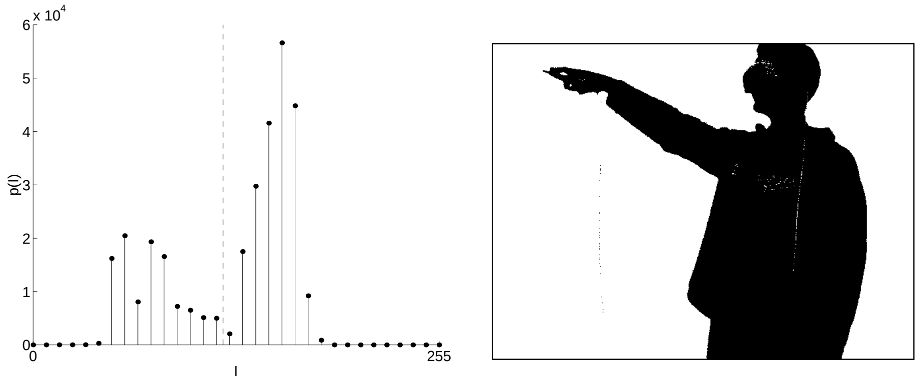


Fig. 2. Segmentation algorithm for the 2D position estimation of the hand in the camera image.

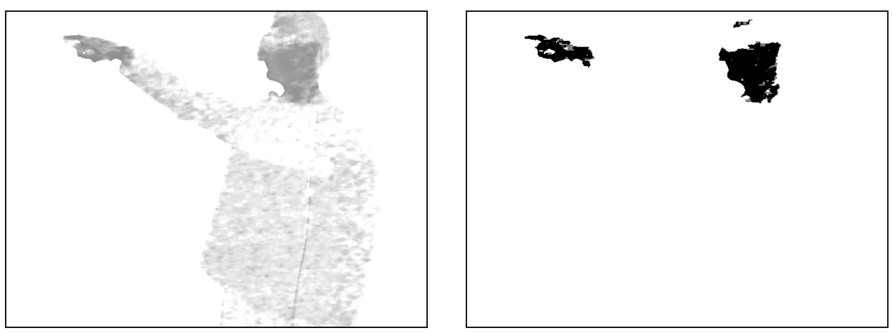
The histogram of the intensity image has a bimodal distribution, the brighter pixels originate from the background whereas the darker originate from the user, figure 3 a). This is used to segment the user from the background. The optimal threshold between the two distributions can be found by minimising the weighted sum of group variances [18]. The estimated threshold is indicated by the dashed line. Figure 3 b) is the resulting binary image after applying this threshold.

The colour variations in the camera image are poor. All colours are close the the gray vector. Therefore the saturation of the image colours is increased by an empirical factor. The red channel of the segmented pixels has maxima in the skin areas (figure 4 a) as long as the user is not wearing clothes with a high reflectance in the long (red) wavelengths. The histogram of the red channel is bimodal, hence it is also thresholded by minimising the weighted sum of group variances. After thresholding a labelling [8] is applied. Figure 4 b) shows the segmentation result of the three largest object. As the position of the head is known the associable skin areas are excluded. The remaining object is the user's hand. Its position in the image is calculated by the first central moments (centre of mass) [8].



a b

Fig. 3. Segmentation of the user. a) Histogram of the intensity image. The dashed line is the threshold found by minimisation of the weighted sum of group variance. b) Thresholded image.



a b

Fig. 4. a) Red channel of the pre-segmented camera image. b) Thresholded red channel after labelling the three largest objects. The gray values of the images are inverted for representation purpose.

4 Experimental Evaluation

This section presents the experimental evaluation of the different parts of the system. First the accuracy of pointing as described in section 3 is tested. Secondly the segmentation of the hand (section 3.1) is tested. The implementation of a real time system is currently in progress, thus test with visual feedback for the user are not yet available.

4.1 Segmentation of the Hand in the Camera Image

Several image sequences of users (Caucasian race) pointing inside the VR-CUBE were taken under different applications, hence different backgrounds and illumination conditions. The 2D position estimation of the hand has been tested off-line on these sequences (figure 4). Only qualitative results are available until now. The 2D position estimation works robustly if a mixture of colours is displayed, which is the case in the majority of the applications. The skin segmentation fails if the displayed images are too dark or if one colour is predominant, e.g. if the red CRT-projector is not used at all for display the measurements of the red channel of the camera become too noisy.

The implementation of a real time system is currently in progress. The calculation intensive part is the 2D estimation of the hand position which is working in a first non-optimised version on entire images (without reducing to regions of interest) with 10*Hz* on 320x240 pixels images on a 450*MHz* Pentium III™. We expect to get 25*Hz* after introducing the reduced search area and optimising the code.

4.2 Pointing Experiments without Visual Feedback

This subsection describes pointing experiments and their results, which were done to evaluate the accuracy of the pointing direction estimation described in section 3. A user was asked to point to 16 different points displayed on the screen as shown in figure 5. No visual feedback was given during these experiments, hence the user should be unbiased and show a natural pointing gesture. Experiments with five different user were done. An image of each pointing gesture was taken together with the data of the magnetic head tracker. The displacement vector between the head tracker and the shoulder was measured for each user.

During the evaluation of the data it turned out that the uncertainty of the position estimate of the head from the magnetic tracker was up to 15*cm* in each direction. It is at the moment not possible to calibrate the device in order to achieve a higher accuracy. This error is too large to be used as head position information in the method described in the previous section. In order to get a more accurate 3D position of the users' heads the visual focus point was segmented in the image data and together with the X position of the tracker, the 3D position of the visual focus point was estimated. This position was then used to estimate the position of the shoulder by the displacement vector as described in section 3. Figure 6 a) shows the results of a representative pointing experiment. The circles ($\circ$) are the real positions displayed on the screen and the asterisks ($*$) connected by the dashed line are the respective estimated positions where the user is pointing to. The error in figure 6 a) is up to 0.7*m*. There are no estimates for the column

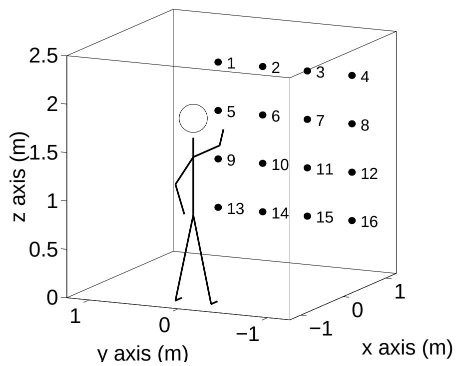


Fig. 5. Experimental setup for pointing experiments without visual feedback in the VR-CUBE. The user has a distance of approximately 2*m* from the screen where 16 points in a 0.5*m* raster are displayed.

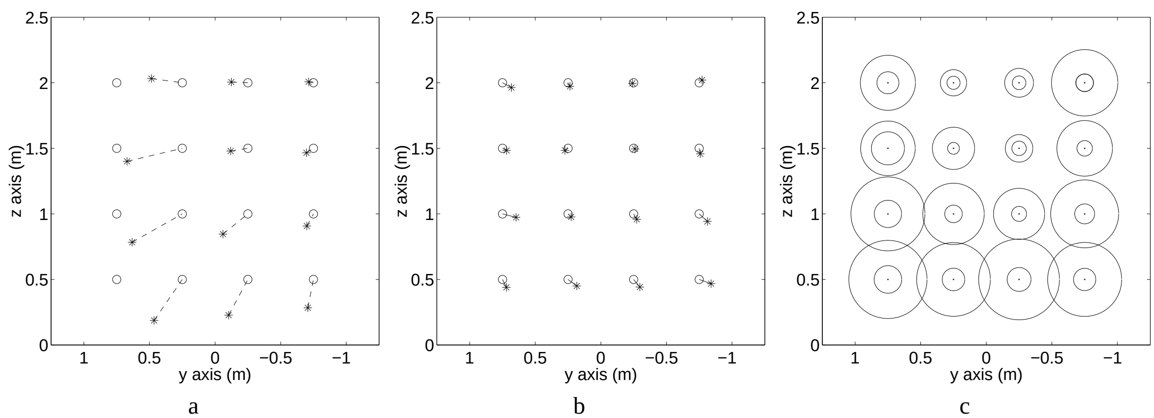


Fig. 6. Results from pointing experiments. The circles in the two first figures are the real positions on the screen. The asterisks are the estimated pointing directions from the system. a) The results of a representative user, using a constant displacement vector. b) The results of a representative user, using a LUT for the displacement vector. c) The inner circle shows the average error of all experiments. The outer circle shows the maximum error of all experiments.

to the left because there is no intersection between the sphere in equation 1 and the line spanned by the camera and the hand of the user.

The error is increasing the more the user points to the left. This is mainly due to the incorrect assumption (made in section 3) that the displacement vector is constant. The direction and magnitude of the displacement vector between the tracker and shoulder is varying. This is illustrated in figure 7.

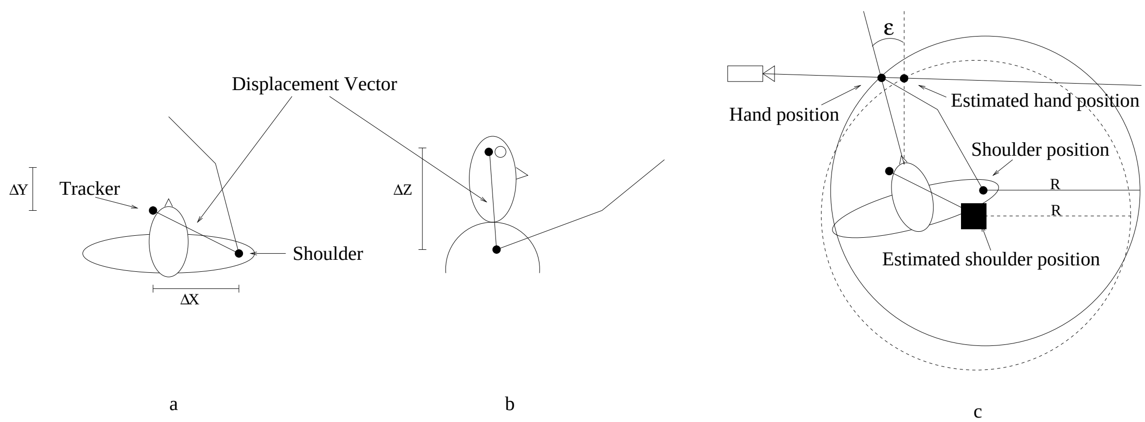


Fig. 7. a+b) The user and the displacement vector between the tracker and shoulder seen from above (a) and from the the right side (b). c) An illustration of the error introduced by assuming the torso to be fronto-parallel.

Figure 7.a and 7.b illustrate the direction and magnitude of the displacement vector between the tracker and shoulder when the user's head is looking straight ahead. As the head is rotated to the left the shoulder is also rotated as illustrated in figure 7.c. This results in a wrong centre of the sphere and therefore a wrong estimation of the 3D hand position. The error is illustrated as the angle ε. Beside the rotation the shoulder is also squeezed which makes the relation between the tracker (head) rotation and the displacement vector non-linear.

Figure 8 shows the components of the displacement vector for the 16 test-points in figure 5 (for a representative user) estimated from the shoulder position in the image data and the tracker data. For each user a lookup table (LUT) of displacement vectors as a function of the head rotation was build. Figure 6 b) shows the result of a representative pointing experiment (same as used in figure 6 a) using a LUT of displacement vectors to estimate the 3D position of the shoulder. Notice that after the position of the shoulder has been correction estimates for the left column is available.

Table 1 shows the average errors and the maximum errors of the five pointing experiments in *mm* for the respective points on the screen. These errors are also illustrated in figure 6 c) where the inner circle indicates the average errors and the outer circle the maximum errors. The average error of all points in all experiments is 76*mm*.

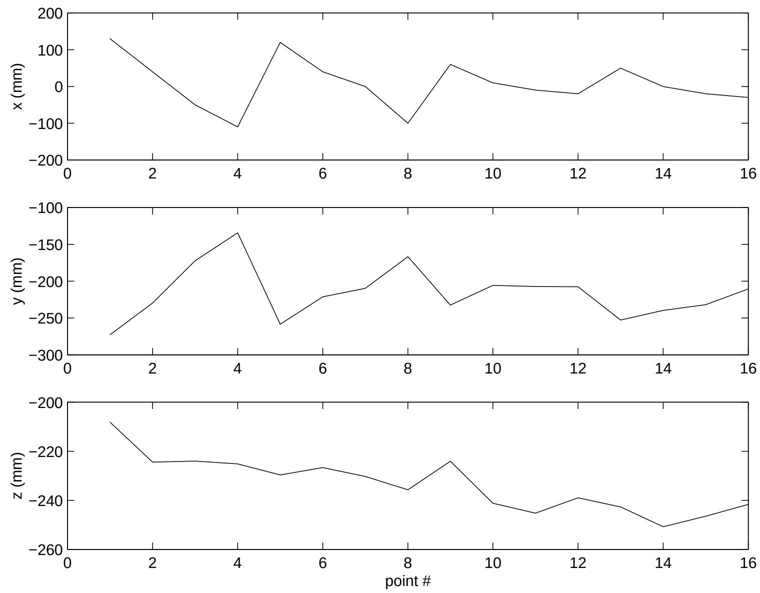


Fig. 8. Components (x,y,z) of the displacement vector as a function of the test-points in figure 5.

Table 1. Average errors and (maximum errors) in *mm* for the respective points on the screen.

	y axis			
z axis	750	250	-250	-750
2000	84 (210)	50 (100)	52 (110)	67 (253)
1500	126 (208)	45 (161)	55 (105)	59 (212)
1000	104 (282)	67 (234)	57 (195)	76 (259)
500	105 (298)	86 (281)	91 (308)	85 (282)

5 Discussion

In this paper we have demonstrated that technical interface devices can be replaced by a natural gesture, namely finger pointing. The pointing gesture is estimated as the line spanned by the 3D position of the hand and the visual focus, defined as the centre point between the eyes. The visual focus point is at the moment estimated from the image data and a X measure. In the future this should be given from the position and orientation of the electro magnetic tracker mounted on the stereo glasses worn by the user. The 3D position of the hand is estimated as the intersection between a 3D line spanned by the hand and camera, and a sphere with centre in the shoulder of the user and radius equal to the length of the user's arm when pointing, R. Pointing experiments with five different user were done. Each user was asked to point to 16 points at a screen in 2*m* distance. Due to , especially, movements of the shoulder during pointing errors up to 700*mm* between the estimated and the real position on the screen was observed. To reduce the errors a LUT was used to correct the position of the shoulder. This reduced the average error to 76*mm* and the maximum error to 308*mm*. This we find to be a rather accurate result given the user is standing two meters away. However, whether this error is too large depends on the application.

In the final system the estimated pointing direction will be indicated by a bright 3D line seen through the stereo glasses starting at the finger of the user and ending at the object pointed to. Thus, the error is less critical since the user is part of the system loop and can correct on the fly. In other words, if the effect of the error do not hinder the user in accurate pointing (using the feedback of the 3D line), then they may be acceptable. However, if they do or if the system is to be used in applications where no feedback is present, e.g. in a non-virtual world, then we need to know the effect of the different sources of errors and how to compensate for them.

The error originates from five different sources: the tracker, the image processing, the definition of the pointing direction, the assumption of the torso being fronto-parallel with respect to the screen, and the assumption that R is constant.

Currently we are deriving explicit expressions for the error sources presented above and setting up test scenarios to measure the effect of these errors. Further experiments will be done in the VR-CUBE to characterise the accuracy and usability as soon as the real time implementation is finished. The experiments will show whether the method allows us to replace the traditional pointing devices as is suggested by our off-line tests.

Another issue which we intend to investigate is the Midas Touch Problem - how to inform the system that a pointing gesture is present. In a simple test scenario with only one gesture - pointing, it is relatively easy to determine when it is performed. As mentioned above (see also [10]) the gesture is recognised when the position of the hand is constant for a number of frames. However, in more realistic scenarios where multiple gestures can appear, the problem is more difficult. One type of solution is presented in [7] where the thumb is used as a mouse bottom. Another, and more natural, is to acommandate the gesture with a spoken input [4], e.g. ”select that (point) object”. Which path we will follow is yet to be decided.

References

1. L. Bakman, M. Blidegn, and M. Wittrup. Improving Human-Computer Interaction by adding Speech, Gaze Tracking, and Agents to a WIMP-based Environment. Master's thesis, Aalborg University, 1998.
2. Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, INC., 1988.
3. M. Böhlen, E. Granum, S. Lauritzen, and P. Mylov. 3d visual data mining. http://www.cs.auc.dk/3DVDM/.
4. T. Brøndsted, L. Larsen, M. Manthey, P. McKevitt, T. Moeslund, and K. Olesen. The Intellimedia WorkBench - an environment for building multimodal systems. In *second international Conference on Cooperative Multimodal Communication*, 1998.
5. D. Browning, C. Cruz-Neira, D. Sandin, and T. A. DeFanti. Virtual reality: The design and implementation of the cave. In *SIGGRAPH'93 Computer Graphics Conference*, pages 135–142. ACM SIGGRAPH, Aug. 1993.
6. R. Cipolla, P. Hadfield, and N. Hollinghurst. Uncalibrated Stereo Vision with Pointing for a Man-Machine Interface. In *IAPR Workshop on Machine Vision Applications*, Yokohama, Japan, December 1994.
7. M. Fukumoto, Y. Suenaga, and K. Mase. "Finger-Pointer": Pointing Interface By Image Processing. *Computer & Graphics*, 18(5), 1994.
8. R. C. Gonzalez and P. Wintz. *Digital Image Processing.* ADDISON-WESLEY PUBLISHING COMPANY, 1987.
9. M. Hirose, T. Ogi, and T. Yamada. Integrating live video for immersive environments. *IEEE MultiMedia*, 6(3):14–22, July 1999.
10. N. Jojic, B. Brumitt, B. Meyers, S. Harris, and T. Huang. Detection and Estimation of Pointing Gestures in Dense Disparity Maps. In *The fourth International Conference on Automatic Face- and Gesture-Recognition*, Grenoble, France, March 2000.
11. R. Kahn and M. Swain. Understanding People Pointing: The Perseus System. In *International Symposium on Computer Vision*, Coral Gables, Florida, November 1995.
12. E. Kjems. Creating 3d-models for the purpose of planning. In *6th international conference on computers in urban planning & urban management*, Venice, Italy, Sept. 1999.
13. O. Larsen, J. Haase, L. Østergaard, K. Hansen, T. Søndergaard, and H. Nielsen. The Virtual Brain Project - Development of a Neurosurgical Simulator. Accepted for The 9th Annual Medicine Meets Virtual Reality, January 2001.
14. E. Littmann, A. Drees, and H. Ritter. Neural Recognition of Human Pointing Gestures in Real Images. *Neural Processing Letters*, 3:61–71, 1996.
15. B. MacIntyre and S. Feiner. Future multimedia user interfaces. *Multimedia Systems*, 4:250–268, 1996.
16. D. MacNeill. *Hand and mind: what gestures reveal about thought.* University of Chicago Press, 1992.
17. K. Mase and R. Kadobayashi. Gesture Interface for a Virtual Walk-through. In *Workshop on Perceptual User Interface*, 1997.
18. N. Otsu. A thresholding selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:62–66, 1979.
19. Polhemus. Stylus magnetic tracker. http://www.polhemus.com/stylusds.htm.
20. Y. Sato, Y. Kobayashi, and H. Koike. Fast Tracking of Hands and Fingertips in Infrared Images for Augmented Desk Interface. In *The fourth International Conference on Automatic Face- and Gesture-Recognition*, Grenoble, France, March 2000.
21. J. Taylor and D. McCloskey. Pointing. *Behavioural Brain Research*, 29:1–5, 1988.

22. R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.