Automatic Analysis of 3D Gaze Coordinates on Scene Objects Using Data From Eye-Tracking and Motion-Capture Systems

Kai Essig
Daniel Dornbusch
Daniel Prinzhorn
Helge Ritter
Jonathan Maycock
Thomas Schack

Center of Excellence “Cognitive Interaction Technology (CITEC)”
Bielefeld University, PB 100131
33501 Bielefeld, Germany
kai.essig@uni-bielefeld.de

Abstract

We implemented a system, called the VICON-EyeTracking Visualizer, that combines mobile eye tracking data with motion capture data to calculate and visualize the 3D gaze vector within the motion capture coordinate system. To ensure that both devices were temporally synchronized we used previously developed software by us. By placing reflective markers on objects in the scene, their positions are known and by spatially synchronizing both the eye tracker and the motion capture system allows us to automatically compute how many times and where fixations occur, thus overcoming the time consuming and error-prone disadvantages of the traditional manual annotation process. We evaluated our approach by comparing its outcome for a simple looking task and a more complex grasping task against the average results produced by the manual annotation process. Preliminary data reveals that the program only differed from the average manual annotation results by approximately 3 percent in the looking task with regard to the number of fixations and cumulative fixation duration on each point in the scene. In case of the more complex grasping task the results depend on the object size: for larger objects there was good agreement (less than 16 percent (or 950ms)), but this degraded for smaller objects, where there are more saccades towards object boundaries. The advantages of our approach are easy user calibration, the ability to have unrestricted body movements (due to the mobile eye-tracking system), and that it can be used with any wearable eye tracker and marker based motion tracking system. Extending existing approaches, our system is also able to monitor fixations on moving objects. The automatic analysis of gaze and movement data in complex 3D scenes can be applied to a variety of research domains, i.e., human computer interaction, virtual reality or grasping and gesture research.

CR Categories: I.4.8 [Image Processing and Computer Vision]: Scene Analysis, Tracking—;

Keywords: eye tracking, motion tracking, gaze tracking, 3D gaze vector

1 Introduction

The calculation of subject’s gaze direction while allowing whole head and body movements while looking at and interacting with complex 3D scenes is very important for various research disciplines. The interpretation of gaze and grasping data allows us to investigate and gain a better understanding of human behavior while performing complex everyday tasks and as a by-product transfer these insights onto technical systems in order to make them more human-like.

The importance of attention in guiding manual actions has seen renewed interest of late [Rasolzadeh et al. 2010; Jonikaitis and Deubel 2011]. In order to achieve good eye-hand coordination, hands and eyes must work together in smooth and efficient patterns. Johansson et al. [Johansson et al. 2001] analyzed the coordination between gaze behavior, fingertip movements, and movements of a manipulated object. Obligatory gaze targets were those regions where contact occurred. They concluded that gaze supports hand movement planning by marking key positions to which the fingertips are subsequently directed.

We live in a dynamic environment and our eyes have to extract the relevant information very quickly from their surroundings in order to supply the brain with the input necessary to make correct decisions or to control our limbs [Land and Tatler 2009]. Whereas traditional eye tracking studies are mainly carried out under artificial conditions in the laboratory, with stimuli presented on a computer screen allowing a controllable setup ([Duchowski 2003; Essig 2008]), there has been a recent academic research shift towards the analysis of human perceptual and cognitive processes in more realistic environments ([Essig et al. 2011; Land and Tatler 2009; Henderson 2005; Zelinsky and Neider 2008; Rayner et al. 2009]). Mobile, head mounted eye-tracking systems were recently developed to meet these needs ([Land and Tatler 2009; Sensomotoric Instruments ]). These systems typically use a scene and an eye camera attached to a helmet or headset. The computer used to interface with the eye tracker and to store the data recording is placed in a backpack and is connected via a cable to the system. This allows subjects to move freely in an unlimited working area while their perceptual behavior is reliably recorded as they interact in natural environments. Whereas the hardware of the mobile systems is quite developed allowing the recording of eye movements under various environmental conditions, methods for the analysis of the recorded data are still in their infancy. The eye tracker records a scene video with overlaid gaze positions (marked by a circle) and the x- and y-positions of the gaze point (in the coordinate system of the scene video), along with the pupil size, timestamped to ensure temporal coherence. The recorded gaze positions are not related to the objects in the scene video, i.e., it is not evident how many fixations (including their time frames) are directed toward an object of interest over the whole video sequence. Additionally, the mobile system only provides the actual gaze position in the 2D coordinate system of the scene video.

In order to calculate the 3D gaze vector, we have combined a mobile eye-tracking (SMI iViewX HED [Sensomotoric Instruments ]) with a motion-capture system (VICON [Vicon motion capture system ]).

In order to calculate the 3D gaze vector, we have combined a mobile eye-tracking (SMI iViewX HED [Sensomotoric Instruments ]) with a motion-capture system (VICON [Vicon motion capture system ]).
We previously implemented software (Multiple Start Synchronizer (MSS)) to ensure a temporal coherence of data streams from both systems [Maycock et al. 2010b]. The system developed for this paper, the VICON-EyeTracking Visualizer (VEV), combines mobile eye-tracking with body motion capture data which not only allows us to calculate and visualize the 3D gaze vector, but also allows for an automatic annotation process of the eye movements. Previous work on combined head-eye tracking systems require a relatively high degree of user calibration and only track head movements. In contrast, our system allows for whole body movements as it uses a mobile eye-tracking system.

In the next section we give a summary of previous work done on combined head-eye tracking systems. Then we describe our system setup (see Section 3) and the implementation details of the VEV program (see Section 4). In Section 5 we evaluate our system using a looking and a grasping study by comparing its results to those given by manual annotation. Finally, Section 6 presents a discussion and outlines future directions.

2 Previous Work

Although there are several approaches that combine head and eye tracking systems, they often have shortcomings or they are not easy to use [Huang et al. 2004]. For example, [Duchowski et al. 2001] describe a virtual reality simulator that features a binocular eye tracker built into the system’s Head Mounted Display, which allows the recording of the user’s dynamic gaze vector within the virtual environment. This system is difficult to specifically designed and suffers from the limitation of virtual reality Head Mounted Displays [Huang et al. 2004]. Allison et al. [Allison et al. 1996] use a magnetic head tracker, which allows only a relatively small range of head movements. Additionally, there is the need to manually measure the relative position between the eye and head tracker. A third approach by [Pérez et al. 2003] suffers from occlusions and a low update rate of gaze positions, making it ill-suited for large screen-based virtual reality systems. For these reasons [Huang et al. 2004] proposed a head-eye tracking system consisting of a video-based eye tracking and a magnetic-inertial head tracking system that can be used for desktop as well as immersive applications (e.g., VR-setups). Their system requires a complex three-stage calibration procedure in which the subject’s head is fixed by a chin rest. The authors illustrate the applicability of their approach using screen applications (2D gaze contingent display system and a 3D picking task) and sketch how the object of interest in an six side immersive virtual environment can be determined. However, their system is optimized for immersive virtual environments and the authors do not provide concrete evaluation data for their approach in the paper.

In [Herholz et al. 2008], the authors introduced an open-source library (called libGaze) to allow the computation of a viewer’s position and gaze direction in real time, using data from a head-mounted eye-tracking and a VICON MX MoCap system while people walk around freely, gesture and interact with large display screens. The system’s modular architecture allows for easy adaptation of the main components to alternative technologies. They evaluated the accuracy of the system with 6 subjects in a calibration procedure. First, the subjects performed the calibration procedure with their heads aligned to the screen. Then they reoriented their head to different positions. For each position they had to fixate points on a grid while holding their head still. The analysis of the median angular error across subjects between each fixation point and its calculated gaze-position on the screen was 1.12 degrees. Furthermore, they tested the real-time capability of their approach in a two-stage behavioral study where subjects stood 95cm in front of a large wall that they could fixate without the need for head movements. In the gaze shift task, 7 subjects had to look at two fixation crosses presented in succession. Then a task followed in which they had to rate the aesthetics of natural images presented on different positions on the wall on a 3-point scale or to count the number of animals depicted in a given image. These tasks capture the different demands required by free viewing and visual searching. The results revealed that subjects applied head movements even though the display was small enough to perceive all information just by eye movements. One disadvantage of this approach is that the head mounted eye-tracking system used does not allow free movements of the subjects. Furthermore, for the system calibration, a separated, quite complex three-stage calibration process has to be performed in order to calculate the relationships between the 3 coordinate systems, taken into account to calculate the 3D gaze position in real world coordinates, and to map the eye tracker coordinates to the 3D viewing vector in the eye coordinate system. The calibration procedure requires the manual adjustment of the orientation of a rectangular frame (representing the observer’s field of view (FOV)) until the subjects’ fixation point is in the FOV center and the top and bottom of the rectangular frame is perceived to be horizontal by the subject. The authors indicate two further shortcoming of their approach and propose suitable compensations: small but systematic errors in the gaze ray estimation result from the approximation of the origin of the eye coordinate system at the nasal bridge instead of the true location of the eye. Additionally, the calibration is quite sensitive to small head movements which have to be measured and compensated.

A commercial solution for tracking point of gaze in any environment was recently developed using VICON Bonita cameras, Nexus Software and Ergoneers Eye Trackers [Eye Tracking Integration with Bonita ]. In contrast to their approach which is closed and requires specific hardware, our approach is more modular and in theory can be used with any mobile eye tracker and motion-tracking system. Additionally, we implemented stand-alone software for the visualization and analysis of eye movements and data export for further statistical processing. Their commercial solution uses a particular eye tracking system (Ergoneers Dikablis Eye Tracking Solution, [Ergoneers Dikablis Eye Tracking Solution]) and the VICON Nexus Software to track and calculate head position and 3D gaze vector to visualize the intersection between gaze and objects in the VICON workspace without providing functionality for eye movement analysis.

In this paper we describe a further approach for a combined head-eye tracking system using a flexible mobile eye-tracking and the VICON body-motion tracking system. We explain the procedure to calculate and visualize the 3D gaze vector by combining the data from both systems. A one-step calibration procedure, already necessary for the normal eye-tracking calibration setup, is used to ensure spatial coherence between the recorded raw data recorded by both systems and to find the relation between the head and the eye coordinate system. Then, by using the positional data of labeled markers on objects in the scene, the system can automatically analyze different eye tracking parameters with respect to objects in complex 3D scenes. In order to evaluate our method, we compare its results against those given by a manual annotation procedure. Before we describe our method in more detail, we first address the systems we use and how we synchronized the recording of both the eye-tracker and the motion capture system to ensure good temporal coherence between the recorded data streams.

3 System Setup

The Manual Intelligence Lab (MILAB) was purposefully built to capture the fine details of human manual interaction situations at a high level of spatial and temporal coherence. MILAB consists of fourteen VICON [Vicon motion capture system] MX+ cameras.
that can track reflective markers attached to both subjects and objects. Having so many cameras in such a small space allows us to track the human hand, which due to self occlusions and occlusions caused by objects is a difficult task. VICON has a marker tracking accuracy of below 1mm and this provides us with good ground truth data for our other vision modalities, in this case the mobile eye tracker. Therefore, in order to get insights into where humans look as they perform everyday tasks with their hands, we use a SMI iViewX (monocular) mobile eye-tracking system [Sensomotoric Instruments] for our experiments. It has a sampling rate of 200 Hertz, with a gaze position accuracy \(< 0.5^\circ - 1^\circ\). Each scene video has a resolution of 376 x 240 pixels at 25 fps. For a more detailed discussion of MILAB please see [Maycock et al. 2010b]. The interested reader is also directed towards recent work carried out in the lab involving recognition of manual actions [Martin et al. 2010] and the kinematics of grasping real versus virtual spherical objects [Maycock et al. 2010a].

The challenges of manual action capture are many and varied, but in most cases reduce to the problem of ensuring that there is a high degree of both spatial and temporal coherence amongst the different input channels. At the core of MILAB is a 14 camera VICON system providing us with high precision 3D positional data to which the other modalities need to be aligned. For modalities not directly supported by VICON, i.e., the eye tracker used in this experiment, it was necessary to develop a custom made solution in order to ensure spatial coherence (this is discussed in the next section). Equally important is the need to ensure temporal coherence across different modalities. The Multiple Start Synchronizer (MSS) was developed to ensure that recording of all data streams start at the same time, and run for a specific duration or until a stop command is sent. MSS first checks if all computer clocks are synchronized correctly using the Network Time Protocol. Pressing start sends all relevant data (i.e., start and stop time, experiment details) via Open Sound Control protocol [Open Sound Control ] to the listening clients. The client applications then wait for the starting time and begin capturing until the end time is reached or a stop signal from MSS is received.

4 Methodology: The VICON-Eyetracking Visualizer

The aim of our method, realized by software we call VICON-Eyetracking Visualizer (VEV), is the calculation of the 3D gaze vector from the data captured by the VICON and mobile eye-tracking systems. We attached markers to the helmet of the eye tracking system in order to measure subjects’ spatial head alignment. Together with the 2D gaze position data (recorded by the mobile eye-tracking system), and an extra marker attached to the nose bridge, we can calculate and visualize the 3D gaze vector. With this information, the program can count the number of fixations along with their cumulative fixation duration directed towards each marker of the scene. By clustering the data for the markers into particular objects, it is possible to calculate object specific perception duration. We now present the details of the method.

When calculating the 3D gaze vector from the recorded data of the two different tracking systems, three different coordinate systems have to be taken into consideration (see Figure 1). In our paper, coordinate systems are denoted by superscripts:

- **World Coordinate System (World):** This is the coordinate system of the environment in which the 3D gaze vector resides (in our case: the VICON coordinate system). Each marker attached to the objects in the 3D scene has a representation in the world coordinate system, allowing us to calculate the distances between the gaze vector and the markers on the objects.

- **Head Coordinate System (Head):** Describes the orientation of subjects’ head and all marker locations relative to the head. The head coordinate system is determined by the four markers attached to the eye tracker helmet.

- **Eye Coordinate System (Eye):** This coordinate system originates at subjects’ right or left eyeball (depending on the tracked eye). This coordinate system results from shifting the head coordinate system to the nose bridge. From there, the origin is moved to the center of the left or right eyeball. A vector in the eye coordinate system represents the eyes’ viewing direction.

In order to calculate the 3D gaze vector from the data of the two tracking systems, some mathematical calculations and transformations need to be performed. The first step is to find the relationship between the head and the eye coordinate system. In order to get the relevant input data, we need five snapshots (reference frames) in which a calibration marker is fixated upon. This is done after the subject has been calibrated. This procedure is now explained.

![Figure 1: The three different coordinate systems and their locations. Red marks the world coordinate system (World), green the eye coordinate system (Eye) and yellow the head coordinate system (Head).](image1)

![Figure 2: The diode board with the five calibration markers.](image2)
4.1 VEV Calibration

During this procedure the subject looks in a specific sequence at five markers on the calibration board (see Figure 2). The first gaze is directed towards the center to direct the eye coordinate system on the central marker (calibration frame center). Then the subject looks sequentially at the other four markers on the board (i.e., upper left, upper right, lower right and lower left). The recorded gaze video and eye movement data are then used in the VEV as reference markers for the calculation of the 3D gaze vector (see below). The corresponding gaze positions of the four calibration markers are determined by selecting those four reference frames where the subject looks most closely at the markers, and pressing the calibration buttons in the VEV (see Figure 3). Before we use this information to calculate how the 3D gaze vector changes with subjects’ head movements, we first have to determine the head and eye coordinate systems.

We first calculate the z-vector of the head coordinate system in world coordinates \( z_{\text{head World}} \), with its origin in the head center and pointing towards the center of the field of view, by using the information of the three markers on the eye tracker helmet (i.e., left, right and front) (see Figure 1).:

\[
z_{\text{head World}} = H_{\text{left World}} + \frac{1}{2} (H_{\text{right World}} - H_{\text{left World}}),
\]

where \( H_{\text{left World}} \) and \( H_{\text{right World}} \) are the positions of the right and left marker on the eye tracker helmet in the world coordinate system. Then \( z_{\text{head World}} \) is normalized. In order to get \( x_{\text{head World}} \), we calculate the vector pointing from the left to the right marker on the helmet:

\[
x_{\text{head World}} = H_{\text{right World}} - H_{\text{left World}}.
\]

By applying cross multiplication followed by normalization we get \( y_{\text{head World}} \) which is orthogonal to \( x_{\text{head World}} \) and \( z_{\text{head World}} \) (see Figure 1). These three vectors form the head coordinate system (Head). The origin of the head coordinate system is then shifted to the eyeball position to form the origin of the eye coordinate system (Eye). This is done by the following calculations: based on the position of the marker on the nose bridge (see Figure 1), the origin of the head coordinate system is shifted towards the corresponding eyeball position using average anatomical data, e.g., in case of the right eye 4 cm on \( z_{\text{head World}} \) and \(-1\) cm on \( z_{\text{head World}} \), i.e., towards the inside of the head. In future we intend to measure this directly to ensure greater accuracy. Now we can calculate the gaze vector in world coordinates:

\[
V_{\text{gaze World}} = V_{\text{CM World}} - V_{\text{eye World}},
\]

where \( V_{\text{CM World}} \) is the position of the central marker on the calibration board in world coordinates, \( V_{\text{eye World}} \) is the origin of the eye coordinate system in world coordinates. Now we can transform the gaze vector \( V_{\text{gaze World}} \) to the eye coordinate system:

\[
V_{\text{gaze Eye}} = V_{\text{gaze World}} * O_{\text{Eye}}^{-1},
\]

where \( O_{\text{Eye}}^{-1} \) is the inverted orientation matrix of the eye coordinate system. For exact calibrations, i.e., the central marker of the diode board should be in the middle of the calibration video, then \( V_{\text{gaze Eye}} \) should result in the following vector:

\[
V_{\text{gaze Eye}} = [0, 0, D_{\text{CM}}],
\]

where \( D_{\text{CM}} \) is the distance between subject’s eye and the central marker on the calibration board, which is 1 m. Now we calculate the differential angle between the x and z (\( \alpha_{xz} \)) and y and z axes (\( \alpha_{yz} \)) of the eye coordinate frame:

\[
\alpha_{xz} = \tan \left( \frac{V_{\text{gaze y}} - V_{\text{eye y}}}{V_{\text{gaze z}} - V_{\text{eye z}}} \right), \quad \alpha_{yz} = \tan \left( \frac{V_{\text{gaze x}} - V_{\text{eye x}}}{V_{\text{gaze z}} - V_{\text{eye z}}} \right),
\]

where \( \alpha_{xz} \) and \( \alpha_{yz} \) correspond to a rotation of the head to left or right, or up or down, respectively. Finally, we rotate the eye coordinate system around these angles. This assures that \( z_{\text{Eye}} \) points exactly towards the central marker of the calibration board (see Figure 1).

4.2 Relationship between Eye Movements and the 3D Gaze Vector

In order to calculate the 3D gaze vector for all head positions our system needs to calculate how the gaze ray changes when the eyes move. Here, we consider the data of the VEV calibration for the four outer markers on the board (see Figure 2). For each of these four reference frames we repeat the following procedure as described in the previous section, i.e.:

1. we calculate the head coordinate system from the markers on the helmet
2. we move the origin of the head coordinate system to the eye to get the eye coordinate system
3. we calculate \( \alpha_{xz} \) and \( \alpha_{yz} \) (see Equation 6) as described above and rotate the eye coordinate system around these angles
Additionally, we calculate the absolute gaze positions (eye\textsubscript{x} and eye\textsubscript{y}) for the four reference markers relative to the gaze coordinates of the central marker on the calibration board:

\[
\text{eye}_x = \text{eye}_\text{AM}_x - \text{eye}_\text{CM}_x, \quad \text{eye}_y = \text{eye}_\text{AM}_y - \text{eye}_\text{CM}_y, \quad (7)
\]

where \((\text{eye}_\text{AM}_x, \text{eye}_\text{AM}_y)\) and \((\text{eye}_\text{CM}_x, \text{eye}_\text{CM}_y)\) are the 2D gaze coordinates (measured by the eye tracking system) for the actual (i.e., for each of the four outer markers) and the central marker on the board, respectively (see Figure 2). Additionally, we calculate the range for the recorded eye tracking data in the horizontal (\(\delta_{\text{eye}_x}\)) and the vertical (\(\delta_{\text{eye}_y}\)) directions:

\[
\delta_{\text{eye}_x} = ((\text{eye}_\text{UR}_x - \text{eye}_\text{UL}_x) + (\text{eye}_\text{LR}_x - \text{eye}_\text{LL}_x))/2, \quad (8)
\]

\[
\delta_{\text{eye}_y} = ((\text{eye}_\text{LL}_y - \text{eye}_\text{UL}_y) + (\text{eye}_\text{LR}_y - \text{eye}_\text{UR}_y))/2. \quad (9)
\]

Now we calculated the range in the differential angle (\(\delta_{\text{angle}_x}\)) in the horizontal and the vertical directions (\(\delta_{\text{angle}_y}\)):

\[
\delta_{\text{angle}_x} = ((\alpha_{x_x} \text{UR}_x - \alpha_{x_x} \text{UL}_x) + (\alpha_{x_x} \text{LR}_x - \alpha_{x_x} \text{LL}_x))/2, \quad (10)
\]

\[
\delta_{\text{angle}_y} = ((\alpha_{y_y} \text{LL}_y - \alpha_{y_y} \text{UL}_y) + (\alpha_{y_y} \text{LR}_y - \alpha_{y_y} \text{UR}_y))/2. \quad (11)
\]

As the final step we calculate the course of the change in the angle relative to gaze shifts in horizontal and vertical directions:

\[
\text{CoC}_x = \frac{\delta_{\text{angle}_x}}{\delta_{\text{eye}_x}}, \quad \text{CoC}_y = \frac{\delta_{\text{angle}_y}}{\delta_{\text{eye}_y}}. \quad (12)
\]

\[4.3 \text{ Calculate the 3D Gaze Vector from the Actual Gaze Position}\]

The following calculations (Equations 13 - 16) are computed every 5ms for each data sample. A prerequisite to these are that the 3 step procedure described at the beginning of Section 4.2 to calculate the head and eye coordinate system and the eye rotation angles are also computed every 5ms. Now we have all the information we need to calculate the rotation factors to adjust the eye coordinate system to the actual gaze position:

\[
\beta_x = \vec{dx} \ast \text{CoC}_x, \quad \beta_y = \vec{dy} \ast \text{CoC}_y, \quad (13)
\]

where \(\vec{dx}\) and \(\vec{dy}\) are the actual 2D gaze positions relative to the central marker position, i.e.:

\[
\vec{dx} = \text{eye}_x - \text{eye}_\text{CM}_x, \quad \vec{dy} = \text{eye}_y - \text{eye}_\text{CM}_y. \quad (14)
\]

We then rotate the eye coordinate system by \(\beta_x\) and \(\beta_y\) so that it is adjusted to the actual gaze position. Now the 3D gaze vector (with a raylength \(rl\) set to a max. of 2m in our studies) can be calculated in coordinates of the world coordinate system according to:

\[
V_{\vec{GazeSpot\ World}} = [0, 0, rl] \ast O_{\text{Eye}}. \quad (15)
\]

Finally the 3D gaze vector is draw from the eye to actual gaze position by:

\[
V_{\vec{Gaze\ Vector\ World}} = V_{\vec{Gaze\ Spot\ World}} - V_{\vec{eye\ World}}. \quad (16)
\]

We then compute the distance from each marker to the 3D gaze vector. The VEV system determines the fixated object by choosing the one which marker has the smallest Euclidean distance to the calculated 3D gaze vector.

\[5 \text{ Experiments and Some Initial Results}\]

To test our implementation we designed a simple experiment in which the task was to pick up different objects (i.e., cup, stapler and sphere) from a left position on a table and place them down on the right (see Figure 4). Five subjects participated in the experiments. First, markers were attached to the subjects’ hand. The human hand has up to 27 degrees of freedom and therefore to capture its complexity and allow modeling of it requires a large number of markers to be used (in our case 26). Then the motion-capture system was calibrated. After that the subject stood in front of a glass table, and the eye tracker was set-up and calibrated using the diode board. For the eye tracker calibration, subjects had to successively fixate on five calibration markers on the diode board (see Figure 2). Then the eye-tracking data for the VEV calibration (described in Section 4.1) was recorded. After that, the diode board was removed from the table and the actual experiment started. Two markers were placed on the table 40cm from the subject. The one on the left hand side indicated the initial position of the objects. The marker on the right hand side marked the position where the subjects had to place the objects down. Three trials were performed with each object. In order to know the positions of the objects in the VICON coordinate...
frame of reference, which is necessary for the automatic annotation, we attached markers to them according to the following two rules: 1) we put enough markers on the objects to be able to get sufficient eye tracking data; 2) the position and number of markers should not interfere with the interaction with the objects. The subject’s hand and eye movements, plus the objects used, were all tracked. All subjects had normal or corrected-to-normal vision and had no known impairments related to arm or hand movements. All gave written informed consent to be part of the study and the experiment was carried out according to the principles laid out in the 1964 Declaration of Helsinki.

We first analyzed the data for the calibration procedure. Subjects had no known impairments related to arm or hand movements. All subjects had normal or corrected-to-normal vision and had no known impairments related to arm or hand movements. All gave written informed consent to be part of the study and the experiment was carried out according to the principles laid out in the 1964 Declaration of Helsinki.

Extracting eye tracking data from such an experiment usually requires the scene video to be manually annotated, which is a very time-consuming and error-prone process. A professional tool that supports the creation of complex annotations on video and audio resources is ELAN (EUDICO Linguistic Annotator) [ELAN Annotation Tool], developed by the Max Planck Institute for Psycholinguistics in Nijmegen, Netherlands. ELAN provides multiple, hierarchically interconnected layers to create various annotation units (tiers) (see Figure 5). The object of interest and the corresponding gaze data (marked by a red line in the fixation tier) can then be observed by the user. Compared to the frame-by-frame annotation, ELAN depicts the relevant fixation data in contiguous blocks, on which the user can concentrate during the annotation process. There were three manual annotators in our study: one of the authors, who was completely inexperienced with regard to the program and the study. They were instructed to analyze the data according to the number of fixations and cumulative fixation duration for the objects, hand and background. Data for the background comprises all the fixations that neither fall on the objects nor on the hand. If a fixation is close to the object or hand, but not on it, the annotators had to decide if they should count it (as either a fixation on the object or hand, respectively) or map it to the background based on their own decisions. The manual annotation took on average around 11.25 minutes for each trial (trial duration 30 sec). Additionally, we loaded the data of both tracking systems in the VEV. After performing the VEV calibration, the data was analyzed automatically using the VEV software on a Intel Core i3 M350 computer with 2x 2.26 GHz and 4 GByte DDR3-RAM under a Windows 7 (32 bit) operating system. The automatic analysis for the 30 sec trials took approximately one second.

We first analyzed the data for the calibration procedure. Subjects had to move their eyes in a determined order, like in a regular calibration procedure, without moving their head: They successively gazed at the central, then at upper left, upper right, lower right and lower left marker. The average results of the manual annotation for the number of fixations and cumulative fixation duration for the single markers can be found in Table 1 (left two columns). Note that we compare at this point the results of our software to a subjective and imprecise manual annotation - we do not address which result resembles more closely the exact number of fixations on the markers or objects in the videos. In order to answer this question, we would need further manual annotations from more people. The results of the VEV are also depicted in Table 1 (right two columns). As can be seen the results of the automatic analysis come very close to those of the manual annotation. The deviation between the results of the manual and automatic annotation is at most 3 percent in case of the cumulative fixation duration, ranging from 14ms (upper left) to 177ms (center) in absolute values. With regard to the number of fixations there are less fixations in the results of the VEV. This is due to different analysis techniques for the recorded sample data: The eye tracker clusters all subsequent samples to one fixation when they are located close to each other (i.e., their spatial distance is below a pre-specified threshold).

The VEV, on the other side, clusters all samples that are within a pre-specified radius around the marker. This radius is larger than the one used by the eye tracker software and can be specified for each marker by the user. For example, there are many markers on the hands. Thus, the radius should be small (around 2 cm). On the sphere, markers are distributed coarsely. Thus, a larger radius (around 5cm) is chosen corresponding to the diameter of the sphere. For the analysis of the data recorded in the calibration procedure, the radius is set to 5 cm for each marker and the threshold for the minimum fixation duration was set to 100ms in the VEV program.

The results for the grasping task with the three objects are depicted in Table 2. The results of the VEV are close to those of the manual annotation. The differences in the data between both methods depend on the object size. For the largest object (i.e., the sphere) which is not occluded when grasping it with the hand, the differences in the number of fixations is very small and the cumulative fixaton duration deviates less than 16 percent (or 950ms). For the smaller objects (i.e., the cup and the stapler) that are occluded by the hand, where subjects’ did not exactly point their gaze in the 3D scene and where saccades between objects and the background occur, the differences are bigger. With regard to the number of fixations, these differences are between 0 to 3 for the objects and 0-2 for the hand. For one reason manual annotation becomes inevitably more subjective: on the other hand both methods analyze the 3D scene from a slightly different point of view (i.e., the center of the eyeball for the automatic and from the scene camera attached to the eye-tracker helmet for the manual annotation). The largest differences in the average number of fixations and cumulative fixation durations between the manual annotation and the VEV-results occur for cup3 and stapler2 (see Table 2). The reason is that the subject’s gaze is located quite close to the hand or to the object, but does not directly points towards them. In those cases, the human annotator may incorrectly assign the fixation towards the object, hand or background. These differences become even bigger in case of fixation durations. Although, for the other trials, the VEV results come close to those of a manual annotation (the average differences are 984ms and 399ms for the object and hand, respectively) and reflect the fixation distribution between the objects and the hands.

6 Discussion and outlook

We have presented a solution for the synchronized recording of a mobile eye-tracking and a motion-tracking system. We introduced a novel method for the off-line visualization and object specific analysis of 3D gaze data. The system is easy to use and the existing
The results of the VEV program for the calibration procedure closely resemble those of the manual annotation procedure. Because this task reflects a regular calibration procedure, the subjects fixate exactly on the markers of the diode board (see Figure 2). Thus, it is also quite obvious for the human annotator on which marker the subject fixates. In contrast to this, for the grasping trials the gaze is often located close to the object or the hand. Then, the human annotator classifies the gaze according to his or her subjective decisions, which may lead to misclassifications. Additionally, there are also technical reasons for the differences between the two methods: the VEV uses the gaze vector pointing from the eyeball center towards the 3D scene, whereas the 2D gaze video is recorded from the scene camera which is displaced from the eyeball (see Figure 1). Thus, the human annotator sees the gaze videos from a slightly different perspective. Fixations close to the objects in the scene video may be directed towards the object when seen directly from the eyeball position (as it is the case for the VEV). We wanted to first validate the program with an everyday task: picking up an object, moving it, and placing it down again. Because the VEV calculates the 3D gaze vector for each sample pair of data from both tracking systems (i.e., every 5 ms), we intend to test our approach in the future on moving objects, for example when the subject traces a ball moving from the left to the right, to demonstrate a smooth pursuit trace. For our test stimuli, where the hand is moving over the objects and when the hand markers are closer to the gaze location than markers on the objects, the VEV may cluster the fixation to the wrong marker. This occurs less often when the object is larger in size (i.e., it is less occluded by the hand) or when the object is moved from the left to the right marker on the table. In order to overcome these limitations and to calculate better results, the VEV allows for adjustments to be made to the radius for each individual marker (depending on markers’ density on the object) and threshold durations (i.e., ignoring very short fixations).

The main reason for the discrepancies between the automatic and manual annotation results is because currently our approach does not consider fixation data from the eye tracker. In the future we will add a fixation detection algorithm to the VEV program. The information on eye rotation and sampling frequency is already available to more accurately determine fixations within gaze clusters. An overview about different algorithms to calculate fixations, saccades, smooth pursuit, and other events directly from the raw data samples, as well as their advantages and disadvantages can be found in [Holmqvist et al. 2011]. These algorithms separate gaze by small saccades, and should improve the comparability of the manual and automatic labeling results. It also lessens the problem of choosing an adequate radius around the markers in the VEV program.

Another problem concerns the estimation of the fixated object when several objects can be aligned within the scope of the calculated 3D gaze vector. This is the case, when several objects are placed behind each other. Our current implementation does not decide about the length of the vector (i.e., depth of an object) which may result in incorrect assignments of fixated objects. Since we have several markers attached to each object, we are able to calculate their visual mass centers and spatial extensions. Considering this data may be the first step to overcome these allocation problems.

Due to its modular design, our system can be adapted to a wide range of different eye-tracking and motion-tracking systems. We will use the program on the data recorded with moving objects to control the results and to find further hints for program improvement. In the future we will extend it to a real-time system, allowing 3D gaze location information to be sent to a HCI or VR system in order to create interactive sessions, such as produce changes in the stimuli when the subjects moves his head or to invoke certain actions depending on subject’s actual 3D gaze position. Our system allows for the automatic analysis of eye-hand span for various grasping and interaction studies and could indeed be used in HMI and VR applications too.

Acknowledgements

This research was supported by the DFG CoE 277: Cognitive Interaction Technology (CITEC). We thank Jonathan Helbach for his help with the analysis, Tobias Roehlig for his help with the experiments and all of our subjects for their time.

Table 1: Results for the average number (over all manual annotations) of fixations (numFix) and fixation duration (fixDur in ms) in case of the calibration for the manual (ELAN) and automatic (VEV) annotation.

<table>
<thead>
<tr>
<th>Calibration Board marker position</th>
<th>Manual Annotation numFix</th>
<th>Hand numFix</th>
<th>VEV numFix</th>
<th>Hand numFix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>7777</td>
<td>7950</td>
<td>5</td>
<td>7950</td>
</tr>
<tr>
<td>Upper Left</td>
<td>6296</td>
<td>6310</td>
<td>1</td>
<td>6310</td>
</tr>
<tr>
<td>Upper Right</td>
<td>6035</td>
<td>6060</td>
<td>1</td>
<td>6060</td>
</tr>
<tr>
<td>Bottom Right</td>
<td>4878</td>
<td>4980</td>
<td>3</td>
<td>4980</td>
</tr>
<tr>
<td>Bottom Left</td>
<td>2088</td>
<td>2130</td>
<td>2</td>
<td>2130</td>
</tr>
</tbody>
</table>

Table 2: Results for the average number (over all manual annotations) of fixations (numFix) and fixation duration (fixDur in ms) in case of the single objects for the manual (ELAN) and automatic (VEV) annotation.

<table>
<thead>
<tr>
<th>Object</th>
<th>Manual Annotation</th>
<th>VEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>cup1</td>
<td>numFix fixDur</td>
<td></td>
</tr>
<tr>
<td>cup2</td>
<td>numFix fixDur</td>
<td></td>
</tr>
<tr>
<td>cup3</td>
<td>numFix fixDur</td>
<td></td>
</tr>
<tr>
<td>sphere1</td>
<td>numFix fixDur</td>
<td></td>
</tr>
<tr>
<td>sphere2</td>
<td>numFix fixDur</td>
<td></td>
</tr>
<tr>
<td>sphere3</td>
<td>numFix fixDur</td>
<td></td>
</tr>
<tr>
<td>stapler1</td>
<td>numFix fixDur</td>
<td></td>
</tr>
<tr>
<td>stapler2</td>
<td>numFix fixDur</td>
<td></td>
</tr>
<tr>
<td>stapler3</td>
<td>numFix fixDur</td>
<td></td>
</tr>
</tbody>
</table>
References


