The International Journal of Parallel, Emergent and Distributed Systems,
Vol. 21, No. 2, April 2006, 79–95

Taylor & Francis
Taylor & Francis Group

A neural network for 3D gaze recording with binocular eye trackers

KAI ESSIG†*, MARC POMPLUN‡ and HELGE RITTER†

†Neuroinformatics Group, Faculty of Technology, Bielefeld University, P.O. Box 10 01 31, 33501 Bielefeld, Germany
‡Department of Computer Science, University of Massachusetts, 100 Morrissey Boulevard, Boston MA 02125-3393, USA

Using eye tracking for the investigation of visual attention has become increasingly popular during the last few decades. Nevertheless, only a small number of eye tracking studies have employed 3D displays, although such displays would closely resemble our natural visual environment. Besides higher cost and effort for the experimental setup, the main reason for the avoidance of 3D displays is the problem of computing a subject's current 3D gaze position based on the measured binocular gaze angles. The geometrical approaches to this problem that have been studied so far involved substantial error in the measurement of 3D gaze trajectories. In order to tackle this problem, we developed an anaglyph-based 3D calibration procedure and used a well-suited type of artificial neural network—a parametrized self-organizing map (PSOM)—to estimate the 3D gaze point from a subject's binocular eye-position data. We report an experiment in which the accuracy of the PSOM gaze-point estimation is compared to a geometrical solution. The results show that the neural network approach produces more accurate results than the geometrical method, especially for the depth axis and for distant stimuli.

Keywords: Eye tracking; Neural network; 3D calibration; Anaglyphs

1. Introduction

In their endeavour to understand the mechanisms that underlie cognitive function, researchers recently made considerable progress, in particular with regard to the human visual system. A significant contribution to this success was made by the analysis of human eye movements (for a review see [1]). One of the most prominent experimental paradigms to investigate biological vision is visual search. In a typical visual search task, subjects have to decide whether a search display, i.e. a set of search items, contains a designated target. Recent visual search studies successfully employed eye-movement recording to gain a new level of insight into visual processing with regard to visual attention, object recognition, text comprehension, and visual working memory [2–9]. Despite the success of these studies, the validity of their findings may be restricted since they were based on two-dimensional stimuli. In order to investigate all capabilities of the visual system—which evolved in a 3D environment—and analyse them in their natural context, experiments employing 3D scenes

*Corresponding author. Email: kessig@techfak.uni-bielefeld.de

The International Journal of Parallel, Emergent and Distributed Systems
ISSN 1744-5760 print/ISSN 1744-5779 online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/17445760500354440

are inevitable. It is interesting to note that, despite the substantial scientific promise, to the authors' knowledge there has been only one prior study that measured and analysed eye movements during visual search in 3D stimuli [10].

The same situation is encountered in the development of more sophisticated gaze-controlled human-computer interfaces. The recent availability of gaze-controlled interfaces has significantly improved the lives of many physically challenged computer users [11–16]. One of the most popular types of such interfaces is called "typing by eye" [17]. The user is presented with a virtual keyboard on a computer screen and can type letters by looking at the corresponding key for some adjustable minimum duration. With the help of this system, users can type text of any length by just moving their eyes. If researchers implemented appropriate 3D gaze position tracking, physically challenged users could control with their eyes complex and realistic virtual 3D scenarios. This would allow them to further enhance interaction with their environment. Why are no such systems in use today and what are the reasons that discourage researchers from investigating eye movements in 3D scenes?

Besides higher cost and effort for the setup, the main reason for the avoidance of 3D displays is the problem of precisely computing a subject's current 3D gaze position. Previous studies attempting the measurement of gaze trajectories in 3D space used geometrical approaches [18]. The basic idea behind these techniques is to employ a binocular eye tracker, measure the vergence angle Φ, and use this information to determine the location of the 3D gaze point. The vergence angle is defined as half the angle between the viewing axes of the two eyes (see figure 1). Usually, the intersection of the two visual axes (more realistically, the midpoint of the shortest straight line connecting them) is geometrically computed and used as the 3D gaze-position estimate. Unfortunately, device specific systematic errors, noise in

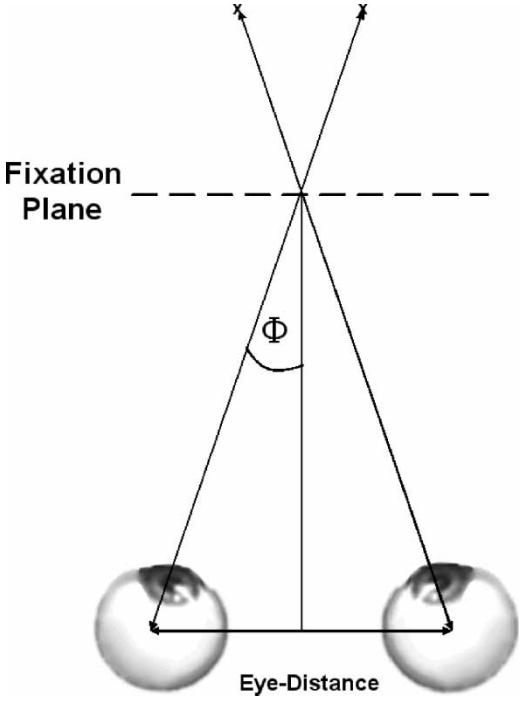

Figure 1. The definition of the vergence angle Φ.

gaze angle measurement, and necessarily imperfect modelling of the actual geometry typically induce large errors in the calculation of 3D gaze position.

The aim of the present study was to tackle the 3D eye tracking problem with a neural-network approach. This idea was motivated by the results of previous work on standard 2D eye tracking. It was shown that a specialised, individually calibrated artificial neural network could be used to significantly reduce the error in gaze-position measurement [19]. The network type applied in that study was a parametrized self-organizing map (PSOM), which is a rapidly learning variant of Kohonen's self-organizing maps (SOMs, see [20]). PSOMs are well-suited for the learning of smooth and highly non-linear functions [21]. Prior to an experimental session, a standard calibration procedure was administered in which the subject had to visually track a marker that appeared successively at different positions on the screen. The subject's gaze positions—as measured by the eye tracker—for each of the calibration markers were used to train the network. Through this training the network learned the eye tracker's error in measurement as a function of the subject's gaze position. In the subsequent experiment, the PSOM was used to estimate the actual gaze positions based on the gaze coordinates obtained by the system. In an empirical investigation, the network was shown to reduce the error in gaze-position measurement to less than 40% of its initial value.

Based on this finding, it seemed promising to construct a PSOM for the computation of a subject's 3D gaze position from binocular gaze-direction data. The PSOM approach could be assumed to be flexible and adaptable enough to compensate for factors that are difficult to handle for purely geometrical algorithms. These factors include differences in the geometry of the eye-tracker setup, non-linear distortions of the camera images and even characteristics of individuals' vergence eye-movements. Therefore, our hypothesis was that using a PSOM for the computation of 3D gaze positions should lead to results that are accurate enough for useful eye-movement studies and human–computer interfaces in virtual or real 3D space. In the present article, we describe how to use the PSOM to compute the 3D gaze position from binocular gaze angle information. The precision of this method is demonstrated using one of the most common and least intrusive types of eye tracker, namely a binocular video-based device.

The remainder of the article is organised as follows: In Section 2, we report a 3D calibration procedure that needs to be performed prior to an experimental session. The data obtained through this procedure are used to train the PSOM, which is presented in complete mathematical detail in Section 3. In Section 4, an empirical study is carried out to evaluate the precision of the neural network approach and compare it to the geometrical solution. Section 5 concludes the article by discussing the results and the applicability of the presented approach to different problems.

2. From a 2D to a 3D calibration procedure

The most common apparatus used for eye-movement studies is a video-based eye tracking system combined with a computer screen for stimulus presentation. For eye tracking in such two-dimensional scenes, the device typically computes the point of intersection of one or both visual axes with the screen. Video-based systems achieve this with the help of cameras directed at the subject's eyes. Using image processing algorithms, these systems continuously determine the subject's pupil positions in the stream of camera images.

This information, in turn, is used to estimate the subject's horizontal and vertical gaze angles and the intersection of the visual axes with the screen.

However, there are physical differences across subjects, and the eye tracker setup (e.g. the position of the camera relative to the subject's head) varies considerably between experimental sessions. This requires the precise mapping of pupil positions to gaze positions on the screen to be determined prior to the experiment. Usually, this mapping is obtained through a calibration procedure during which the subject is shown a marker that appears at different positions on the screen, and the subject is asked to visually track this marker. Typically there are nine marker positions that are arranged in a 3×3 array. This procedure allows the system to obtain a set of samples indicating how pupil positions relate to gaze positions on the screen. An interpolation algorithm is applied to this data to approximate a pupil-to-gaze mapping for the entire stimulus presentation area. This interpolation is usually performed by fitting polynomials of degree two to the calibration data [22]. However, the same type of calibration procedure can also be used to obtain training data for a PSOM that performs such a mapping [19].

In the present study, an SMI EyeLink eye tracker was chosen for the implementation and evaluation of the proposed method. This binocular, video-based system measures a subject's gaze position on a computer screen with a frequency of 250 Hz and an accuracy between 0.5 and 1.0° of visual angle. The method for 3D stimulus presentation was decided to be anaglyph ("red–green") images. These images are perceived as 3D when viewed through glasses with a red filter for one eye and a green filter for the other eye. Although this technique has certain restrictions such as its incompatibility with colored stimuli, it could be integrated with the EyeLink system in a simple and straightforward way (see figure 2).

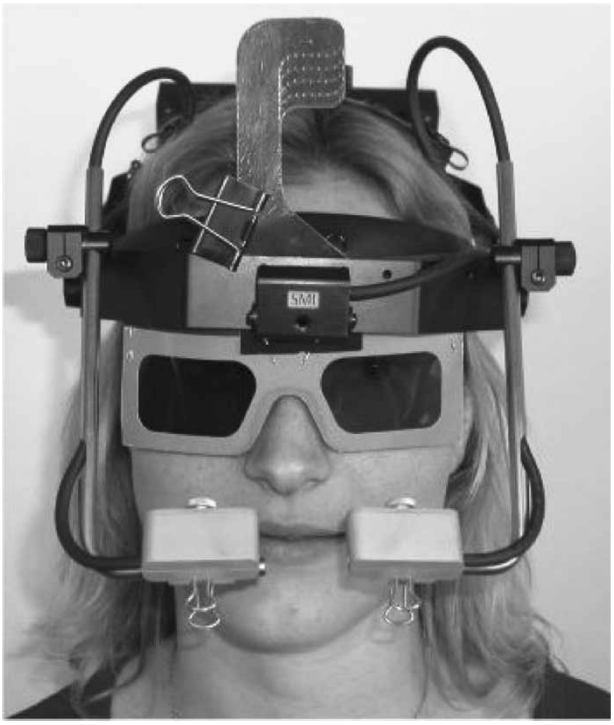

Figure 2. Eye tracker headset with 'red–green' glasses.

Previous experiments [23] demonstrated that vergence eye movements during the examination of anaglyph images were very similar to those in real-world 3D scenes, supporting the choice of the anaglyph method for the present study. However, it is important to note that the method of 3D gaze tracking presented here can also be applied to other techniques of 3D stimulus presentation such as shutter goggles.

Obviously, for the purpose of 3D gaze tracking a 3D calibration procedure needs to be developed. As a first step, let us consider the standard 2D calibration procedure of the EyeLink device. During this procedure, the subject has to visually track a dot that successively appears at nine different positions—arranged in a 3×3 array—on the screen. For these points the eye tracker system relates the pupil positions of the left (x_l, y_l) and the right eye (x_r, y_r) to the screen coordinates (x_b, y_b) of the corresponding calibration marker. These are the reference vectors of the calibration, which are used to estimate the gaze position that corresponds to a given pupil position. For those pupil positions that did not occur during the calibration of the system, the corresponding gaze positions are estimated by means of interpolation. The EyeLink system achieves this interpolation by fitting polynomials of degree two to the reference vectors.

In order to provide an adequate 3D calibration, we used anaglyphs to construct a virtual $3 \times 3 \times 3$ calibration cube. It consisted of three lattices that were parallel to the screen, each of them containing 3×3 cross markers. The horizontal and vertical distance between neighboring markers, as perceived by the subject, was 8 cm. The planes were 8 cm in front of the screen, on the screen, and 8 cm behind the screen, respectively. An anaglyph image of the cube is shown in figure 3.

The following calibration procedure was administered: Prior to the 3D gaze tracking experiment, the eye tracker headset was fitted to the subject's head, the cameras were adjusted correctly, and the red–green glasses were fixed at a suitable position on the headset. Once the subject had taken a defined position in front of the screen, the calibration procedure began. At first, all nine cross markers of the middle plane were presented to the subject at once. One of them was randomly selected to be highlighted with a surrounding circle.

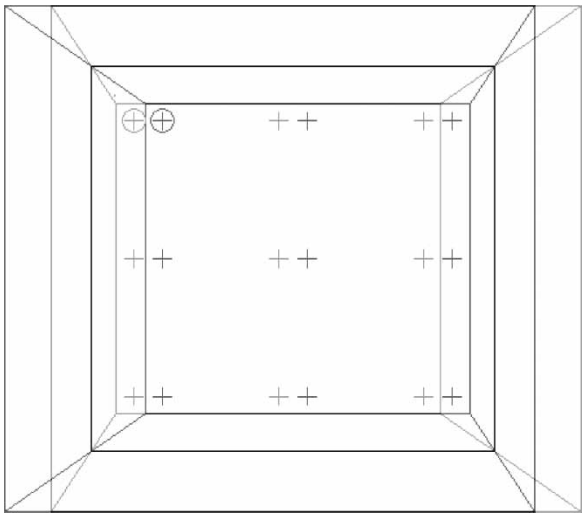

Figure 3. The calibration points on the last plane of the cube. The colors black, red, green, and yellow are represented by grey values of increasing brightness.

Subjects were asked to gaze exactly at this marker and then press a mouse button. After that another cross marker was randomly chosen to be highlighted, and so on, until every marker had been selected. Then the cross markers on the front plane and finally the cross markers of the back plane were shown in the same way.

This procedure was chosen for two reasons: First, showing all markers of one plane at the same time facilitates the subjects' perception of virtual depth and their ability to perform a precise eye movement towards the designated target. Second, showing only one plane at a time avoids visual interference between the planes. The data obtained through this method are used to train the PSOM, which is described in the following section.

3. Mathematical description of the PSOM

Some classes of neural networks are able to "learn" highly non-linear functions, which is a desired capability for the mapping of pupil to gaze positions. Among these networks are the so-called SOMs, developed by Kohonen [20]. It is indeed likely that SOMs could learn an accurate mapping from pupil to screen coordinates. However, SOMs have two properties that make them unsuitable for such purposes: First, they supply only the position of the most stimulated neuron in a "neuron lattice" rather than a continuous output. This means that for the present purpose every pixel of the screen would have to be represented by one neuron. Second, an SOM learning process usually requires thousands of training samples. For the required number of neurons, it would be impossible to collect sufficient calibration data within a reasonable amount of time.

Hence, we use a variant of the SOM, namely a PSOM [21]. This variant does not have the disadvantages of SOMs mentioned above. It provides the desired continuous output, and it does not require a training procedure with multiple iterations. Instead of such training, the PSOM receives only some selected input–output pairs as parameters, that is, the coordinates of the calibration points and the binocular gaze position data measured by the eye tracker.

The PSOM's task in the subsequent gaze-recording session is to approximate the 3D coordinates of the subjects' gaze point from the (x_l, y_l) and (x_r, y_r) coordinates on the screen as recorded by the eye tracker. Instead of using the raw pupil positions in the camera images as input to the PSOM, we use this gaze-position data to take advantage of certain pre-processing steps such as head-movement compensation performed by the EyeLink system. Moreover, while different eye trackers derive distinct data from the camera input, all of them compute the 2D gaze coordinates on the screen. Therefore, basing the neural computation on 2D gaze coordinates makes it more easily applicable to a variety of eye tracking systems.

The operation of the PSOM consists of two components: First it maps 3D gaze positions (x_b, y_b, z_b) through interpolation onto corresponding 2D (on-screen) gaze positions (x_l, y_l) and (x_r, y_r), based on the calibration data. Second, its recurrent component computes the inverse of this mapping, thereby providing the desired projection from 2D to 3D gaze coordinates.

Mathematically speaking, a PSOM is a recurrent neural network that realizes a multi-dimensional projection $\mathbf{f}$. For the present application we use a network with five input neurons, 27 inner neurons, and three output neurons, with bidirectional connections between the inner and output layers. The training of the PSOM consists in setting parameters for the inner neurons. Each neuron receives the spatial coordinates of one of the 27 calibration points

$k \in \mathbf{A}$, where

$$\mathbf{A} = \{k_{xyz} | k_{xyz} = x\hat{\mathbf{e}}_x + y\hat{\mathbf{e}}_y + z\hat{\mathbf{e}}_z; \quad x, y, z = 0, \ldots, 2\}. \tag{1}$$

Obviously, these 27 points are arranged in a $3 \times 3 \times 3$ grid with coordinates scaled from 0 to 2 in each dimension (see figure 4). All PSOM computations refer to this coordinate system.

For each point k, the corresponding inner neuron also receives information on the calibration data for k, that is, the gaze parameters measured while the subject was looking at k during the calibration procedure. These data are provided in the form of a vector $\vec{\mathbf{w}}_k$ given by $\vec{\mathbf{w}}_k = (x_{lk}, y_{lk}, x_{rk}, y_{rk}, x_{dk}) \in \mathbb{R}^5$, where x_{lk} is the x-coordinate for the left eye, and y_{lk} the corresponding y-coordinate for the left eye while fixating point k. For the right eye, the corresponding values are x_{rk} and y_{rk}. The fifth element of $\vec{\mathbf{w}}_k$ results from $x_{dk} := x_{rk} - x_{lk}$.

We introduce the divergence x_{dk} as the fifth dimension of $\vec{\mathbf{w}}_k$ because the z-coordinate of the 3D gaze-position mainly depends on this divergence. Since the differences in the divergence are smaller than those in the x- and y-directions, the divergence has to be weighted with a specific factor. This method leads to a faster termination of the PSOM calculations.

Having received the points k and associated reference vectors $\vec{\mathbf{w}}_k$, our PSOM "knows" the expected gaze parameters when the subject fixates one of the calibration points. In its feedforward step, for a given 3D gaze position, the PSOM interpolates between the vectors $\vec{\mathbf{w}}_k$ in order to estimate the subject's gaze parameters for any given 3D gaze position within or near the calibration cube. This interpolation function $\mathbf{f}(s)$ is constructed from the superposition of a suitable number of simpler basis functions $\mathbf{H}(.,.)$ as follows:

$$\mathbf{f}(s) = \sum_{k \in A} H(s, k)\vec{\mathbf{w}}_k \tag{2}$$

The values of the three basis functions are within the interval [0, 1] depending on different values s (see figure 5), so that the reference vector of a calibration point which is close to the

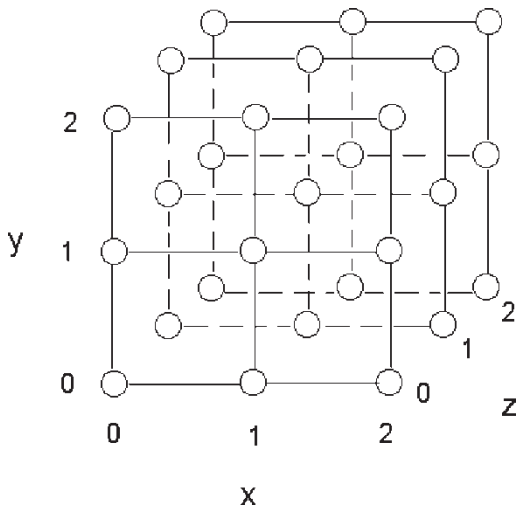

Figure 4. The calibration points in the virtual $3 \times 3 \times 3$ cube. The PSOM computation uses a 3D coordinate system that maps every calibration point to the value 0, 1, or 2 in each dimension.

current gaze position is weighted strongly (near 1) and points which are far away are weighted weakly (near 0). The basis functions $H : \mathbb{R}^3 \times A \rightarrow \mathbb{R}$ have to comply with the requirement

$$H(s,k) = \delta_{s,k} \quad \forall \quad s,k \in \mathbf{A} \tag{3}$$

where δ represents the Kronecker symbol. It is defined as:

$$\delta_{ij} = \begin{cases} 1 & : \quad i = j \\ 0 & : \quad i \neq j \end{cases} \quad \forall \quad i,j = 0,1,2 \text{ (in the special case)}$$

This ensures that

$$\mathbf{f}(s) = \vec{\mathbf{w}}_s \quad \forall \quad s \in \mathbf{A}$$

Thus, it is guaranteed that the interpolation function passes through the given points. But how can we choose suitable functions H that obey equation (3), that are smooth, and simple to handle?

One convenient solution is to make a product ansatz and to combine the suitable function from three 1D functions (one for each coordinate direction x, y and z):

$$H(s_x\hat{\mathbf{e}}_x + s_y\hat{\mathbf{e}}_y + s_z\hat{\mathbf{e}}_z, r_{xyz}) = H^{(1)}(s_x,x) \cdot H^{(1)}(s_y,y) \cdot H^{(1)}(s_z,z) \tag{4}$$

The new 1D functions must then have the property $H^{(1)} : \mathbb{R} \times \{0,1,2\} \rightarrow \mathbb{R}$:

$$H^{(1)}(q,n) = \delta_{q,n} \quad \forall \quad q \in \mathbb{R}, \quad n \in \{0,1,2\} \tag{5}$$

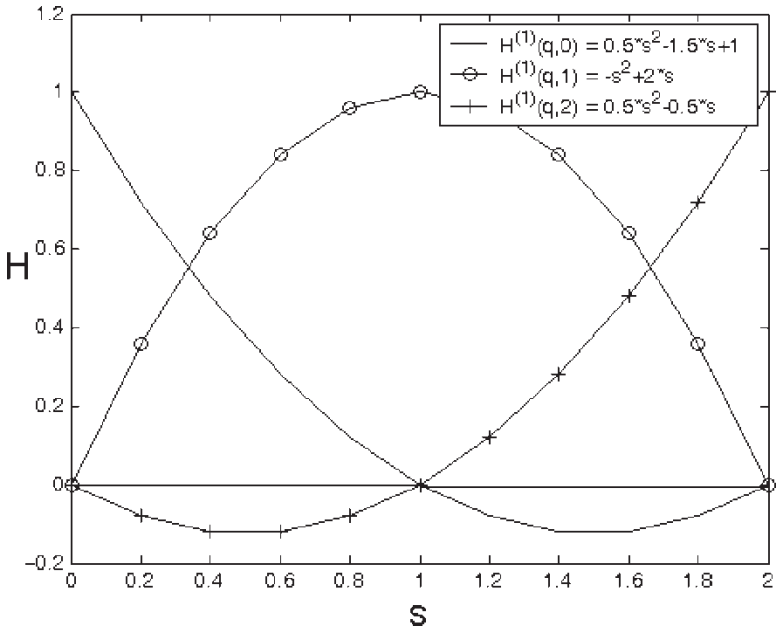

Figure 5. The three different basis functions.

Because n has only three possible values (0, 1 and 2), it is sufficient to choose a set of three basis functions $\mathbb{R} \rightarrow \mathbb{R}$. For this purpose, polynomials of second degree seem to be particularly suitable because they have no redundant degrees of freedom (see figure 5).

So far, we have seen how the PSOM, in its feedforward step, is able to project 3D coordinates onto corresponding 2D gaze positions for both eyes. However, our desired function should do the opposite, namely approximate the subject's 3D gaze-position from the system's 2D measurements. The feedback property of the PSOM accomplishes this by calculating the inverse function $\mathbf{f}^{-1}$ of $\mathbf{f}$. The network's recurrent connections implement an iterative minimisation of an error function. This error function is defined as:

$$E(s) = \frac{1}{2}(\mathbf{f}(s) - \mathbf{f}_{\text{et}})^2, \tag{6}$$

which is the deviation of the 2D gaze data $\mathbf{f}_{\text{et}}$ measured by the eye tracker from those 2D gaze data $\mathbf{f}(s)$ calculated by the PSOM for the current estimate s of the 3D gaze position. This estimate is stored in the activation pattern of the three output neurons and is updated in an iterative gradient-descent procedure until the results of equation (2) closely approximate the actual current 2D gaze-position data provided by the system:

$$s(t+1) = s(t) - \epsilon \frac{\delta E(s)}{\delta s}, \quad \text{with} \quad \epsilon > 0 \tag{7}$$

This means that the iteration process stops if $E(s(t))$ falls below a specified threshold value, which we should set accordingly to the screen resolution. The final value of s is taken as the network's output, indicating the subject's 3D gaze position as a function of the current 2D binocular input to the network.

4. An empirical investigation of the accuracy of measurement

The aim of this experiment was to measure how precisely the PSOM calculates a subject's 3D gaze position. To have a baseline measure for comparison, we also determined the accuracy of the geometrical solution which was based on the 2D calibration procedure. The subjects had to gaze at cross markers in space, created through anaglyph images. The deviation of the 3D gaze coordinates calculated with each method to the exact positions of the markers were measured, analysed and compared.

4.1 Method

4.1.1 Subjects. Eight students of the University of Bielefeld participated in this experiment (1 female, 7 male). They had already taken part in eye tracking experiments and had experience with anaglyph images. They had normal or corrected visual acuity and, according to their own statements, after a short time of practice no problems to perceive the 3D structure of the anaglyph images presented to them.

4.1.2 Stimuli. The only stimuli were cross markers with a diameter of about one degree of visual angle. Individual markers were highlighted by drawing a circle around them. The circle served as an indication to the subjects to direct their fixation on that particular marker.

Through the anaglyph technique these markers could be shown at virtual positions in front of, behind, or on the screen.

4.1.3 Apparatus. We used an SMI Eyelink eye tracker for the experiments. This system enables binocular eye-movement recording at a sampling rate of 250 Hz. The resolution of its gaze-position measurement is 0.005° with an average gaze position error between 0.5 and 1.0°. For stimulus presentation we used a screen (Sony Multiscan 20 *sfII*) with a diagonal size of 20 in. (36.5 × 28.0 cm), where the resolution of the image was 640 × 480 pixels.

4.1.4 Procedure. After setting up the eye tracker hardware, the standard 2D calibration procedure was performed, followed by the 3D calibration routine as described above. Afterwards, a similar procedure using the same cross marker symbol was carried out. Sixteen markers on each of four equidistant planes of a cube were presented to the subject in order to investigate the accuracy of the 3D gaze-point calculation. None of these 64 positions was identical to any of the 27 positions in the calibration procedure. The virtual planes were located at 11 and 3.67 cm in front of and 3.67 and 11 cm behind the screen. In the first experimental condition, stimuli were shown plane by plane. The temporal sequence of planes was third, fourth, first, and second (as seen from the subject's perspective, that is, the first plane being closest to the subject). The subject's task was the same as during the 3D calibration. In the second experimental condition, only one out of 64 cross markers was shown to the subject at a time and the temporal order of the planes was chosen randomly.

4.1.5 Data analysis. As a baseline for assessing the performance of the neural network, we determine the precision of a purely geometrical solution. This solution is based on the standard 2D calibration. In a first step, the equations for the left and right visual axes are calculated on the basis of both the measured binocular gaze positions on the screen and the subject's head position. The coordinates of the left and the right eye were set to (values in cm) (figure 6):

$$\vec{\mathbf{a}}_{\mathrm{l}} = \begin{pmatrix} -3.3 \\ 0.0 \\ 50 \end{pmatrix} \quad \vec{\mathbf{a}}_{\mathrm{r}} = \begin{pmatrix} 3.3 \\ 0.0 \\ 50 \end{pmatrix} \tag{8}$$

Thus, we ensured that the subjects sat 50 cm in front of the screen and assumed a distance of 6.6 cm between the centers of the eyeballs. The linear equations for the visual axes can be written as:

$$\vec{\mathbf{g}}_{\mathrm{l}} = \vec{\mathbf{a}}_{\mathrm{l}} + \mu\vec{\mathbf{w}}_{\mathrm{l}} \quad \vec{\mathbf{g}}_{\mathrm{r}} = \vec{\mathbf{a}}_{\mathrm{r}} + \eta\vec{\mathbf{w}}_{\mathrm{r}}, \tag{9}$$

where $\vec{\mathbf{w}}_{\mathrm{l}}$ and $\vec{\mathbf{w}}_{\mathrm{r}}$ are the direction vectors for the left and right visual axis, respectively, pointing to the screen plane. The estimated 3D gaze position is the midpoint of the shortest straight line connecting the two visual axes:

$$|\vec{\mathbf{a}}_{\mathrm{l}} + \mu(\vec{\mathbf{s}}_{\mathrm{l}} - \vec{\mathbf{a}}_{\mathrm{l}}) - \vec{\mathbf{a}}_{\mathrm{r}} + \eta(\vec{\mathbf{s}}_{\mathrm{r}} - \vec{\mathbf{a}}_{\mathrm{r}})| := \min \tag{10}$$

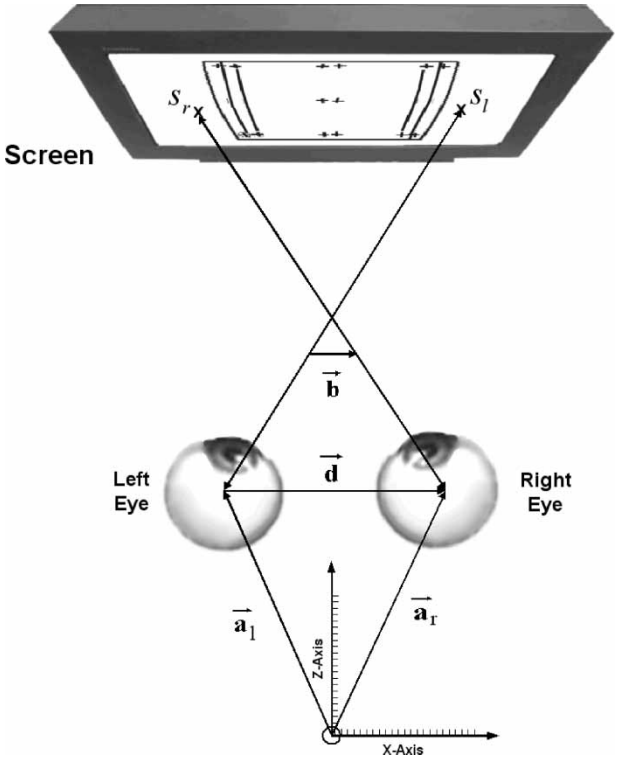

Figure 6. The geometrical solution.

With $\vec{\mathbf{d}} = \vec{\mathbf{a}}_\mathrm{l} - \vec{\mathbf{a}}_\mathrm{r}$, equation (10) can be written as:

$$(\vec{\mathbf{d}} + \mu\vec{\mathbf{w}}_\mathrm{l} + \eta\vec{\mathbf{w}}_\mathrm{r})^2 := \min \tag{11}$$

Resolving equation (11) to μ and η results in:

$$j + \eta h + \mu k = 0 \tag{12}$$

$$g + \mu h + \eta i = 0, \tag{13}$$

where $g = \vec{\mathbf{j}}\vec{\mathbf{w}}_\mathrm{r}$, $h = \vec{\mathbf{w}}_\mathrm{r}\vec{\mathbf{w}}_\mathrm{l}$, $i = \vec{\mathbf{w}}_r^2$, $j = \vec{\mathbf{d}}\vec{\mathbf{w}}_\mathrm{l}$, $k = \vec{\mathbf{w}}_l^2$. From equations (12) and (13), we can derive the values of μ and η that accomplish the distance minimization:

$$\mu = \frac{ij - gh}{h^2 - ik} \tag{14}$$

$$\eta = \frac{hj - gk}{h^2 - ik} \tag{15}$$

By combining equations (9), (14) and (15) the estimated 3D gaze position can then be calculated as:

$$\vec{\mathbf{b}} = \frac{\vec{\mathbf{b}}_\mathrm{l} + \vec{\mathbf{b}}_\mathrm{r}}{2}, \tag{16}$$

with $\vec{\mathbf{b}}_\mathrm{l} = \vec{\mathbf{a}}_\mathrm{l} + \mu\vec{\mathbf{w}}_\mathrm{l}$ and $\vec{\mathbf{b}}_\mathrm{r} = \vec{\mathbf{a}}_\mathrm{r} + \eta\vec{\mathbf{w}}_\mathrm{r}$.

4.2 Results and discussion

We compare the precision of the 3D gaze-position by the neural network using a three-factorial analysis of variance. The three independent variables are: the sequence of looking at the cross markers (individual or by the plane), the method of position calculation (PSOM or geometrical solution), and the depth plane (0–3, where 0 signifies the plane with the longest distance from the subject). The measurement error, defined as the Euclidean distance from the measured to the real gaze-position, is the only dependent variable.

For the first independent variable (sequence of cross markers) the statistical analysis of the experiment data shows that the difference between the individual or the plane-by-plane sequence of viewing is not significant. The average total error of the neural net and the geometrical method in x-, y- and z-direction is 4.41 cm for the plane-by-plane sequence and 4.55 cm for the individual sequence of viewing. This result suggests that the vergence movements that are executed between the fixation of crosses with different distances from the subject are fast and exact. Since the influence of this variable is not significant it will be disregarded in further analyses.

For the other two independent variables the ANOVA shows significant results. The methods for the gaze-position calculation (neural network vs. geometrical calculation) provide results with a different precision ($F(1,9) = 19.42$; $p < 0.005$). The average total error in x-, y- and z-direction for the geometrical solution is 6.18 cm, whereas the neural net provides a value of 2.78 cm. Hence, the neural net calculates 3D gaze positions from the eye tracker data with an error of only 45% in comparison to the geometrical solution.

Furthermore, for both methods, the precision of the calculation of the exact gaze-point coordinates increases from the back to the front plane ($F(3,27) = 44.44$; $p < 0.001$). Table 1 shows the significant differences of the average total error in x-, y- and z-direction between the individual planes. In the following, 0 signifies the plane with the longest and 3 the one with the shortest distance to the subject. The values for the geometrical method and the neural network indicate the average values and their standard error.

Obviously, the measurement error decreases from the back to the front plane for both methods of computation. The standard error decreases similarly (see figure 7). There is an obvious explanation for this effect: The greater a subject's vergence angle (i.e. the closer the 3D gaze position to the subject), the more it changes during a shift of the 3D gaze point. Hence, the resolution and the precision of the spatial measurement of the eye tracker is higher for near than for distant gaze points.

The statistical analysis of the experiment data furthermore shows that the interaction between the method of calculation and the depth plane is significant ($F(3,27) = 11.99$;

Table 1. Average total error for individual planes.

	Both methods	*Geometrical method*	*Neural net*
Plane 1	*Average total error (cm)*	*Average total error (cm)*	*Average total error (cm)*
0	5.89	8.27 ± 0.81	3.51 ± 0.22
1	5.08	7.20 ± 0.64	2.96 ± 0.20
2	3.95	5.37 ± 0.43	2.53 ± 0.16
3	3.00	3.87 ± 0.35	2.14 ± 0.17

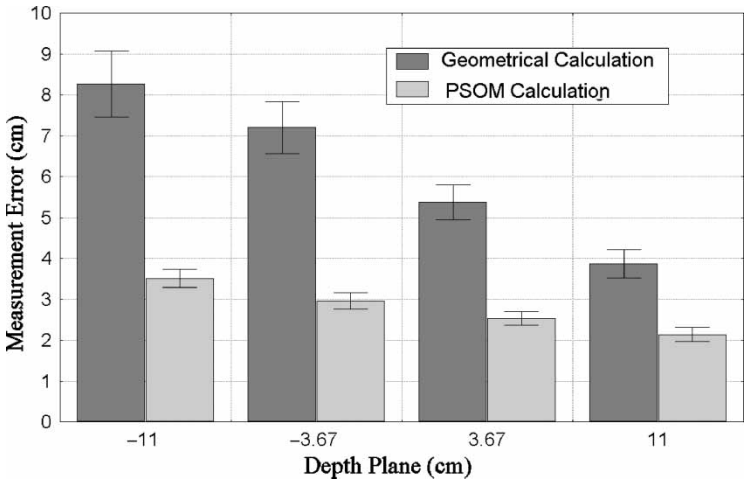

Figure 7. The measurement errors for the gaze-point calculation at different depth planes. The values on the *x*-axis indicate how far before (positive values) or behind (negative values) the corresponding depth plane is located relative to the screen plane. Error bars show standard error.

$p < 0.0001$). The interaction means, as shown in figure 7, that the measurement error for the geometrical method depends much more strongly on the depth plane as it is the case for the PSOM.

We calculated a second analysis of variance with a fourth independent variable, the "dimension of the measurement error". The result of the ANOVA shows that the distribution of the measurement error varies significantly across the *x*-, *y*- and *z*-coordinates ($F(2,18) = 54.26$; $p < 0.0001$).

Table 2 shows the average total error for all subjects, separated for the individual coordinates.

Figure 8 shows the results in a bar-chart diagram. It is obvious that the *z*-error is always higher than the *x*- and *y*-errors, because the *z*-coordinate is much more sensitive to small changes in the binocular gaze positions than are the *x*- and *y*-coordinates. Table 3 shows the measurement errors for the individual planes.

Obviously, the changes in the measurement error from plane to plane in the *z*-coordinate are considerably greater than those for the *x*- and *y*-coordinates. It can also clearly be seen that the measurement errors decrease from the back to the front plane. This suggests that the neural network compensates the errors in the back plane better than the geometrical method. Figure 9 summarizes the results for the geometrical method in a bar-chart diagram, while figure 10 shows the corresponding results for the neural network.

Table 2. Average total error for individual coordinates.

	Both methods	*Geometrical method*	*Neural net*
Coordinate	*Average total error (cm)*	*Average total error (cm)*	*Average total error (cm)*
X	0.97	1.41 ± 0.04	0.52 ± 0.05
Y	1.03	1.24 ± 0.05	0.82 ± 0.05
Z	4.16	5.79 ± 0.36	2.53 ± 0.11

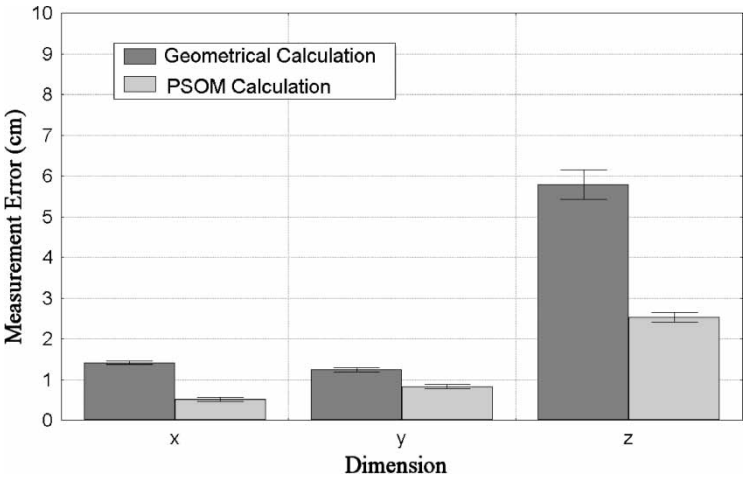

Figure 8. The measurement errors for the gaze-point calculation relating to the *x*-, *y*- and *z*-coordinates. Error bars show standard error.

5. Conclusions

The results confirm that a subject's gaze position can be calculated considerably more accurately with the neural network than with a geometrical approach. One of the critical improvements attained by the neural net is the superior error compensation for both the *z*-coordinate and the back plane. Hence, the proposed integration of the neural net with an eye tracking environment is an effective, fast and economical method to accurately compute an individual's gaze position from the 2D binocular coordinates. Furthermore, the new neural network calibration overcomes the limitations of purely geometrical solutions by adapting to each user's gaze characteristics and compensating for camera-related image distortions.

The ability to precisely calculate 3D gaze position has many positive ramifications for human-machine interfaces. With an accurately calculated 3D gaze position, the system can more precisely react to user inputs, which makes visually mediated navigation more

Table 3. Measurement error for individual coordinates and planes.

Plane	*Coordinate*	*Geometrical method (cm)*	*Neural net (cm)*
0	*x*	1.63 ± 0.09	0.79 ± 0.14
	y	1.37 ± 0.11	0.95 ± 0.11
	z	7.89 ± 0.85	3.17 ± 0.24
1	*x*	1.52 ± 0.08	0.49 ± 0.08
	y	1.34 ± 0.10	0.83 ± 0.06
	z	6.84 ± 0.67	2.76 ± 0.20
2	*x*	1.34 ± 0.07	0.45 ± 0.07
	y	1.19 ± 0.11	0.81 ± 0.11
	z	4.97 ± 0.46	2.28 ± 0.16
3	*x*	1.15 ± 0.06	0.36 ± 0.05
	y	1.05 ± 0.09	0.71 ± 0.12
	z	3.44 ± 0.38	1.92 ± 0.15

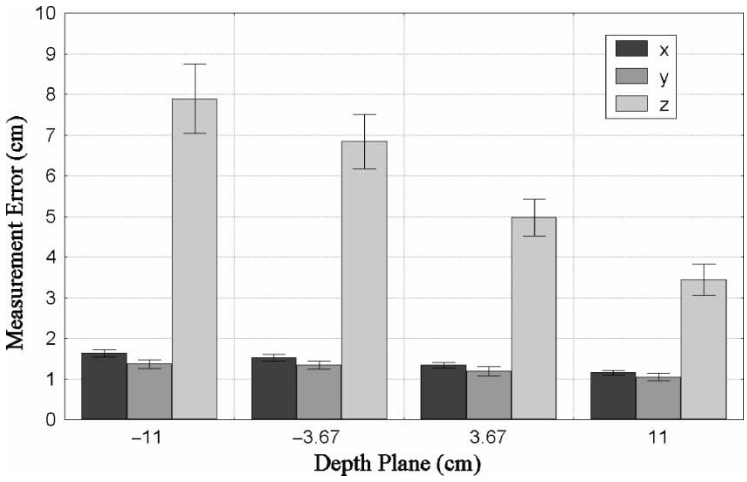

Figure 9. The measurement errors in the gaze-point calculation using the geometrical method. The values on the *x*-axis indicate how far before (positive values) or behind (negative values) the corresponding depth plane is located relative to the screen plane. Error bars show standard error.

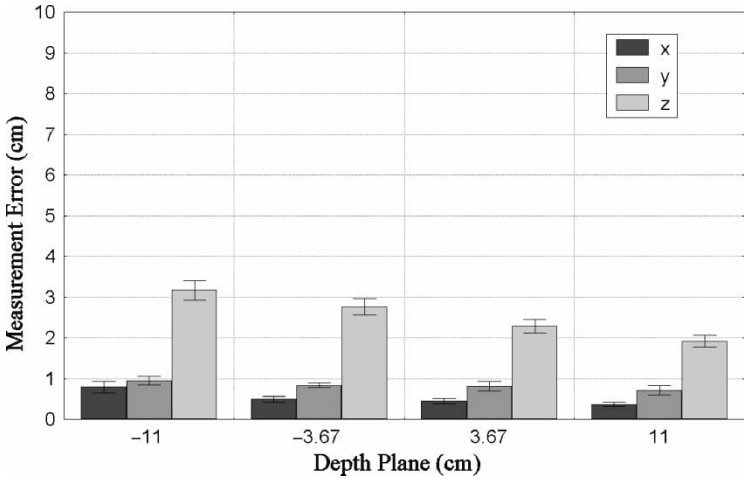

Figure 10. The measurement errors of the gaze point calculation using the neural network. The values on the *x*-axis indicate how far before (positive values) or behind (negative values) the corresponding depth plane is located relative to the screen plane. Error bars show standard error.

powerful than in existing systems. Individuals could navigate through a variety of VR-scenarios, such as a virtual museum, and gaze at the desired points or directions of interest.

One area to which this neural network method can be successfully applied is assistive technology. Physically challenged users could be able to not only steer their way through virtual environments but also through real environments. By simply moving their eyes, they could adeptly control a computer or other machines. In a typical construction scenario, a subject points to individual parts that a robot manipulator should sequentially grasp and assemble. With the average accuracy of 2.78 cm provided by the neural net method, it would be possible to determine at which object the subject is looking and to translate that gaze position into an instruction to the robot manipulator to maneuver towards that object.

With the described techniques, it is possible to generate more complex experiments to investigate the capabilities of the visual system by measuring eye movements in a more naturalistic setting. For example, given the spatial accuracy of the new approach, 3D visual search experiments could be conducted in which the subject's sequence of attended search items could be analysed. The results obtained from these environments can yield deeper insight into the mechanisms underlying natural 3D vision. The combination of eye tracking with arising technologies opens up a wide range of possibilities not only for the scientific investigation of 3D vision but also for the development of assistive tools for the physically challenged.

Acknowledgements

This research was funded by the Deutsche Forschungsgemeinschaft (Sonderforschungsbereich 360, Situierte Künstliche Kommunikatoren*).*

References

[1] Rayner, K., 1998, Eye Movements in reading and information processing: 20 years of research, *Psychological Bulletin*, **124**, 372–422.
[2] Bertera, J.H. and Rayner, K., 2000, Eye movements and the span of the effective visual stimulus in visual search, *Perception and Psychophysics*, **62**, 576–585.
[3] Findlay, J.M. and Gilchrist, I.D., 1998, Eye guidance and visual search. In: G. Underwood (Ed.) *Eye Guidance in Reading, Driving and Scene Perception* (Oxford: Elservier), pp. 295–312.
[4] Jacobs, A.M., 1987, Toward a model of eye movement control in visual search. In: J.K. O'Regan and A. Lévy-Schoen (Eds.) *Eye Movements: From Physiology to Cognition* (North Holland: Elsevier Science Publishers), pp. 275–284.
[5] Motter, B.C. and Belky, E.J., 1998, The guidance of eye movements during active visual search, *Vision Research*, **38**, 1805–1815.
[6] Rayner, K. and Fisher, D.L., 1987, Eye movements and the perceptual span during visual search. In: J.K. O'Regan and A. Lévy-Schoen (Eds.) *Eye Movements: From Physiology to Cognition* (North Holland: Elsevier), pp. 293–302.
[7] Williams, D.E. and Reingold, E.M., Preattentive guidance of eye movements during triple conjunction search tasks, (in press), Paper submitted to *Psychnomic Bulletin and Review.*
[8] Williams, D.E., Reingold, E.M., Moscovitch, M. and Behrmann, M., 1997, Patterns of eye movements during parallel and serial visual search tasks, *Canadian Journal of Experimental Psychology*, **51**, 151–164.
[9] Zelinsky, G.J. and Sheinberg, D.L., 1997, Eye movements during parallel-serial visual search, *Journal of Experimental Psychology: Human Perception and Performance*, **23**, 244–262.
[10] McSorley, E. and Findlay, J.M., 2001, Visual search in depth, *Vision Research*, **41**, 3487–3496.
[11] Jacob, R.J.K., 1991, The use of eye movements in human-computer interaction techniques: What you look at is what you get, *ACM Transactions on Information Systems*, **9**(3), 152–169.
[12] Frey, L.A., White, P.K. and Hutchinson, T.E., 1990, Eye-gaze word processing, *IEEE Transactions on Systems, Man, and Cybernetics*, **20**(4), 944–950.
[13] Levine, J.L., 1981, An eye-controlled computer. Res. Rep. RC-8857, Yorktown Heights: IBM Thomas J. Watson Research Center.
[14] Parker, J.L. and Mercer, R.B., 1987, The ocular communicator: a device to enable persons with severe physical disabilities to communicate using eye movements, *Exceptional Child*, **34**(3), 221–226.
[15] Spaepen, A.J. and Wouters, M., 1989, Using an eye-mark recorder for alternative communication. In: A.M. Tjoa, H. Reiterer and R. Wagner (Eds.) *Computers for Handicapped Persons* (Vienna: R. Oldenbourg), pp. 475–478.
[16] Pomplun, M., Ivanovic, N., Reingold, E.M. and Shen, J., 2001, Empirical evaluation of a novel gaze-controlled zooming interface. In: M.J. Smith, G. Salvendy, D. Harris and R.J. Koubek (Eds.) *Usability Evaluation and Design: Cognitive Engineering, Intelligent Agents and Virtual Reality*, Proceedings of the 9th International Conference on Human-Computer Interaction 2001, New Orleans, USA.
[17] Stampe, D.M. and Reingold, E.M., 1995, Selection by looking: A novel computer interface and its application to psychological research. In: J.M. Findlay, R. Walker and R.W. Kentridge (Eds.) *Eye Movement Research: Mechanisms, Processes, and Applications* (Amsterdam: Elsevier), pp. 467–478.
[18] Duchowski, A.T., 2003, *Eye Tracking Methodology: Theory and Practice* (London: Springer).

[19] Pomplun, M., Velichkovsky, B. and Ritter, H., 1994, An artificial neural network for high precision eye movement tracking. In: B. Nebel and L. Dreschler-Fischer (Eds.) *Lecture Notes in Artificial Intelligence: AI-94 Proceedings* (Berlin: Springer Verlag), pp. 63–69.
[20] Kohonen, T., 1990, The self-organizing map, *Proceedings of IEEE*, **78**, 1464–1480.
[21] Ritter, H., 1993, Parametrized self-organizing maps. *ICANN93-Proceedings* (Berlin: Springer), pp. 568–577.
[22] Stampe, D.M., 1993, Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems, *Behavior Research Methods, Instruments & Computers*, **25**(2), 137–142.
[23] Essig, K., 1998, Messung von Binokularen Augenbewegungen in realen und virtuellen 3D-Szenarien. Diploma thesis, Faculty of Technology, Bielefeld University, Germany.