Gaze3DFix: Detecting 3D fixations with an ellipsoidal bounding volume

Sascha Weber¹ · Rebekka S. Schubert¹ · Stefan Vogt¹ · Boris M. Velichkovsky^{1,2,3} · Sebastian Pannasch¹

Published online: 26 October 2017
© Psychonomic Society, Inc. 2017

Abstract Nowadays, the use of eyetracking to determine 2-D gaze positions is common practice, and several approaches to the detection of 2-D fixations exist, but ready-to-use algorithms to determine eye movements in three dimensions are still missing. Here we present a dispersion-based algorithm with an ellipsoidal bounding volume that estimates 3D fixations. Therefore, 3D gaze points are obtained using a vector-based approach and are further processed with our algorithm. To evaluate the accuracy of our method, we performed experimental studies with real and virtual stimuli. We obtained good congruence between stimulus position and both the 3D gaze points and the 3D fixation locations within the tested range of 200–600 mm. The mean deviation of the 3D fixations from the stimulus positions was 17 mm for the real as well as for the virtual stimuli, with larger variances at increasing stimulus distances. The described algorithms are

implemented in two dynamic linked libraries (Gaze3D.dll and Fixation3D.dll), and we provide a graphical user interface (Gaze3DFixGUI.exe) that is designed for importing 2-D binocular eyetracking data and calculating both 3D gaze points and 3D fixations using the libraries. The Gaze3DFix toolkit, including both libraries and the graphical user interface, is available as open-source software at <https://github.com/applied-cognition-research/Gaze3DFix>.

Keywords Binocular · 3D eye tracking · 3D gaze points · 3D fixations · Eye movement analysis · Methodology · Open-source software

✉ Rebekka S. Schubert
Rebekka.Schubert@tu-dresden.de

Sascha Weber
sascha.weber@tu-dresden.de

Stefan Vogt
stefan.vogt1@tu-dresden.de

Boris M. Velichkovsky
boris.velichkovsky@tu-dresden.de

Sebastian Pannasch
sebastian.pannasch@tu-dresden.de

¹ Faculty of Psychology, Technische Universität Dresden, 01069 Dresden, Germany

² Institute of Cognitive Studies, Kurchatov Research Center, Moscow, Russian Federation

³ Moscow Institute for Physics and Technology, Moscow, Russian Federation

Nowadays, eyetracking is widely used in basic and applied research on visual perception and cognition. However, the vast majority of eyetracking research has been done with two-dimensional stimuli. This might restrict the validity of some findings as the visual system evolved in a 3D environment, which makes experiments employing 3D scenes inevitable (Collewijn, Steinman, Erkelens, Pizlo, & van der Steen, 1992; Essig, Pomplun, & Ritter, 2006; Land, 2006; Lappi, 2016). Furthermore, applications of virtual reality containing stereoscopic presentations increase, thereby offering great possibilities for research, for example when investigating action and perception (Durgin & Li, 2010). In the field of eye tracking research it is common practice to determine gaze direction and/or the 2-D gaze position on a reference plane (mostly a monitor) (Duchowski, 2007; Hammoud, 2008; Hansen & Ji, 2010). Although several methods for the computation of gaze points in 3D have already been reported (see below), no implementation is immediately available.

The analysis of eye tracking measurements usually requires detecting fixations and saccades from the recorded sample data. Meanwhile, there are various approaches to estimating 2-D fixations and saccades from sample data. Several software packages

and tool boxes are available (see, e.g., Komogortsev, Gobert, Jayarathna, Koh, & Gowda, 2010). Furthermore, there have been several suggestions for detecting fixations in 3D scenes using monocular eyetrackers (Diaz, Cooper, Kit, & Hayhoe, 2013; Munn, Stefano, & Pelz, 2008; Pelz & Canosa, 2001; Reimer & Sodhi, 2006). However, those methods do not provide a position of the 3D fixation but only a start and end time. For the detection of 3D fixations and saccades only few approaches have been suggested (Duchowski et al., 2002; Duchowski, Medlin, Gramopadhye, Melloy, & Nair, 2001) and none of them can easily be used by researchers. The aim of the present article is to close this gap by introducing an approach for the detection of 3D fixations. The proposed algorithm was successfully tested, implemented, and is available open source, ready-to-use for everyone interested in 3D eye movement research.

We start with an overview of existing approaches for the computation of 3D gaze points and the detection of 2-D and 3D fixations, followed by the section in which our algorithms are introduced. In the third section, two empirical studies into the accuracy of our method are reported. We conclude the article by discussing the advantages and requirements of our method as well as its applicability.

Related work

First, we summarize various attempts that have been made so far to compute 3D gaze points and describe current approaches for the detection of 2-D and 3D fixations.

Computation of 3D gaze points

Our literature review revealed four main approaches to compute gaze depth and 3D gaze points: The extended gaze disparity approach, a neural network approach, the ray-based approach, and the vector-based approach.

The gaze disparity approach represents the simplest way to extract the gaze depth from binocular eye tracking data. Therefore, the difference of the horizontal pupil positions of both eyes needs to be calculated. This has often been done in perceptual research (e.g., Blythe, Holliman, Jainta, Tbaily, & Liversedge, 2012; Grosjean, Rinkenauer, & Jainta, 2012; Pobuda & Erkelens, 1993; Rambold, Neumann, Sander, & Helmchen, 2006; Semmlow, Hung, & Ciuffreda, 1986; Wismeijer, van Ee, & Erkelens, 2008). This measure is suitable for many research questions but does not provide a 3D gaze point.

Duchowski and colleagues (Daugherty, Duchowski, House, & Ramasamy, 2010; Duchowski, Pelfrey, House, & Wang, 2011; Wang, Pelfrey, Duchowski, & House, 2012, 2014) extended the gaze disparity approach using the following assumptions: (i) the viewing position is aligned with the screen center, (ii) the eyes are at the same height, and at equal distances from the screen, and (iii) a standard interpupillary

distance (a) of 6.3 cm is assumed. They measure the viewing distance (D) and compute the z -coordinate of the 3D gaze point from the gaze disparity (Δx) by using triangulation:

$$z = \frac{\Delta x D}{\Delta x - a}$$

With the eye locations known, the x - and y -coordinates of the 3D gaze point can be computed. To increase accuracy, online and offline filtering procedures as well as a 3D calibration routine were implemented (Duchowski et al., 2011; Wang et al., 2012, 2014).

Essig et al. (2006) employed an artificial neural network, specifically a parametrized self-organizing map (PSOM) for the calculation of 3D gaze points. Therefore, following a 2-D calibration of the eye tracker, a 3D calibration is done using calibration points in a $3 \times 3 \times 3$ calibration cube. The 3D coordinates of the calibration points (x, y, z) and the 2-D gaze positions measured by the eye tracker are used as input-output pairs to train the neural network. After the 3D calibration, the PSOM provides estimated 3D gaze points as a function of the 2-D binocular input. Empirical evaluation revealed estimates of higher accuracy for the PSOM method than for 3D gaze points derived from the vector-based approach with a fixed interpupillary distance (Essig et al., 2006; Pfeiffer, Latoschik, & Wachsmuth, 2009). However, as compared to the extended gaze disparity approach, PSOM is less accurate, is more difficult to implement, and requires a longer 3D calibration (Wang et al., 2014).

If the geometry of the presented stimuli is known, 3D gaze points can be inferred by allocating gaze points to stimulus positions as a ray-based approach (Duchowski et al., 2002; Duchowski et al., 2001; Mansouryar, Steil, Sugano, & Bulling, 2016). Therefore, a gaze vector needs to be estimated and the point of intersection with the reconstructed 3D scene represents the 3D gaze point. However, a gaze point can only be unambiguously determined if the gaze vector intersects exactly one point in the 3D scene. Otherwise, if none or several points in the 3D scene are hit, the selection is under- or overspecified, a problem common for all ray-based approaches (Pfeiffer et al., 2009). Furthermore, this approach requires a virtual reconstructing of the scene when real stimuli are presented.

We decided to use the vector-based approach, in which the gaze vectors for both eyes are calculated using their locations in space and the gaze positions on the measuring plane. Theoretically, the 3D gaze point would be the point at which both vectors intersect. Since this is seldom the case, Collewyn, Erkelens, and Steinman (1997) restricted all of their stimuli and the eye movements to a horizontal plane of regard, simplifying the computation to a two-dimensional triangle. To determine a real 3D gaze point, the shortest straight line between the two gaze vectors is computed, and the 3D gaze point is defined as the midpoint of this line (Epelboim et al., 1995; Hennessey & Lawrence, 2008; Pfeiffer et al., 2009; Wibirama & Hamamoto,

2012, 2014); see below for detailed formulas. In a study comparing the accuracy of 3D gaze points from vector intersection to 3D gaze points from gaze disparity, Duchowski et al. (2014) found that the depth estimates of both methods were highly correlated. However, the correlation decreased with increasing distance between the stimulus and observer. The authors concluded that vector-based gaze depths are more accurate, because depth estimates based on gaze disparity appeared to inflate the error magnitude. Furthermore, in contrast to the ray-based approach, a vector-based approach requires no knowledge of the geometry of the scene. In addition, no depth calibration is needed as in the extended gaze disparity approach, the PSOM, or other machine-learning approaches (although an additional depth calibration has been proposed by Wibirama & Hamamoto, 2014). This is beneficial because it allows for free head movements as long as the positions of the eyes are known, and it enables research into study convergence errors (when users converge slightly in front or behind the fixated object). Another advantage of machine-learning approaches is the possibility to adapt to the individual viewing characteristics of participants. Due to the above-mentioned advantages, we think that a vector-based approach is the most appropriate method for the computation of 3D gaze points.

Methods for 2-D fixation detection

Several methods to detect fixations and saccades on a two-dimensional reference plane have been described. Three approaches can be distinguished: Dispersion-based algorithms, velocity-based algorithms, and area-based algorithms (Salvucci & Goldberg, 2000; for a recent review, see Lappi, 2015).

Dispersion-based algorithms are built on the assumption that the distances between single gaze points are smaller in a fixation than in a saccade. A fixation is detected, when the spatial dispersion threshold and the temporal threshold is met. Measures for the spatial dispersion threshold can either be pairwise distance—which is computationally costly—maximum centroid distance, or bounding box size (Salvucci & Goldberg, 2000). Salvucci and Goldberg suggested a dispersion threshold between 0.5° and 1° of visual angle. The temporal threshold is defined by the minimum fixation duration. Since fixations are usually longer than 100 ms, most dispersion-based algorithms use a minimum duration of 100–200 ms (Salvucci & Goldberg, 2000).

Velocity-based algorithms use a threshold in the velocity. In the simplest form the point-to-point velocity is computed and the gaze point is considered as belonging to a fixation if the velocity is below the threshold and a defined minimum fixation duration is reached. More sophisticated velocity-based algorithms examine velocity peaks for the presence of a saccade and subsequently determine the beginning and ending of the saccade on the basis of the velocity profile (Lappi, 2015). Instead of using a fix velocity threshold, Nyström and Holmqvist (2010) suggest to calculate the threshold

algorithmically on the basis of the data. Since small saccades can be quite slow, velocity ranges of saccades can overlap with other types of eye movements such as smooth pursuit movements and the vestibulo-ocular reflex (Lappi, 2015). To circumvent this overlap, the use of acceleration as the criterion instead of velocity has been suggested (e.g., Behrens, MacKeben, & Schröder-Preikschat, 2010) and a combination of velocity and acceleration has been applied (e.g., SR Research, 2009).

Area-based algorithms require predefined regions in the visual field called areas of interest (AOIs). Gaze points inside an AOI are defined as belonging to a fixation once a minimal fixation duration is reached. Points outside the AOIs are discarded as well as points laying within an AOI but not meeting the temporal threshold. Since area-based algorithms do not allow identifying eye movements within an AOI they are no fixation detection algorithms per se; they rather permit to determine fixation groups of higher order (also called glances) belonging to visual targets (Salvucci & Goldberg, 2000).

Few attempts have been made to compare different fixation detection algorithms. For instance, Salvucci and Goldberg (2000) evaluated five exemplary algorithms. They concluded that incorporating a minimum fixation duration is advisable, velocity-based and dispersion-based algorithms achieve a similar performance, accuracy, and robustness of the fixation identification. Area-based algorithms were found to be less preferable in general, but suitable for certain types of aggregation. Komogortsev et al. (2010) compared three velocity-based with two dispersion-based algorithms. Although rather simple stimulus material was used, they reported pronounced differences between the algorithms. In a similar vein, Shic, Scassellati, and Chawarska (2008) showed that different algorithms result in variations with regard to the number and duration of fixations. Accordingly, this can also lead to different interpretations of the same data.

Approaches from 2-D to 3D fixation detection

When moving from 2-D to 3D stimuli, other approaches are necessary. Diaz et al. (2013) suggest using a velocity-based algorithm to identify fixations in virtual environments presented in a head-mounted display. Further approaches for the identification of fixations in 3D scenes have been described by Munn et al. (2008), Pelz and Canosa (2001), or Reimer and Sodhi (2006). However, these approaches are intended for monocular eye trackers, and although they are suitable to identify fixations in 3D environments, they provide only a fixation direction, not a 3D fixation position.

Theoretically, the known 2-D detection algorithms could be used to first identify the start and end times of fixations on the basis of the binocular data. Afterward the fixation centers could be computed using the common 3D gaze point algorithms. However, using traditional algorithms for 2-D fixations on 3D eyetracking data leads to a component-wise analysis of movements of the left and the right eye, thereby effectively ignoring

depth (Duchowski et al., 2001). As a result, different fixation durations and variable start and end times of a fixation could be obtained for the right and the left eye. A similar approach is to use algorithms for the detection of 2-D fixations on some kind of a composite position of the eyes as, for example, on the mean gaze angle of left and right eye (Wismeijer et al., 2008) or on the conjugate eye movement signal ($[\text{left eye} + \text{right eye}]/2$; Grosjean et al., 2012). Along with Duchowski and colleagues (Duchowski et al., 2002; Duchowski et al., 2001), calculating gaze points in three-dimensional space before the identification of 3D fixations seems more appropriate to us because the 3D gaze point provides a composite representation of the user's gaze. We elaborate on this issue further in the supplement and provide a comparison between the two approaches: Identifying 2-D fixations using 2-D samples, before calculating the 3D fixation position versus calculating 3D gaze positions before identifying 3D fixations using 3D gaze points.

Duchowski et al. (2001) describe a simple velocity-based algorithm to detect 3D fixations in which the velocity is calculated using two successive three-dimensional gaze points. Their evaluation revealed that the algorithm underestimated fixation duration and number of fixations as compared to the theoretically expected values. After they improved their algorithm (Duchowski et al., 2002), the velocity calculation was based on five successive gaze points, and a velocity threshold of 130°/s was suggested. They also implemented an acceleration-based algorithm with an adaptive thresholding technique, which dynamically sets the threshold depending on the noise level. They furthermore added a minimum fixation duration, which is by default 150 ms. Their evaluation revealed that velocity-based algorithms are easier to implement, require less parameters but are sensitive to noise. Further, they inferred that acceleration-based algorithms can be more robust, but determining adequate parameters is difficult.

We decided to extend the dispersion-based approach to detect 3D fixations from 3D gaze points since it has been shown to be accurate and robust, requires few parameters and can also be used for eyetrackers with a low sampling rate.

Proposed algorithms to compute 3D gaze points and 3D fixations

In the following, we describe the calculation steps for the computation of 3D gaze points, the essential procedure for the use of a dispersion-based algorithm for the detection of 3D fixations as well as the implementation of those algorithms in the toolkit Gaze3DFix.

Computation of 3D gaze points

Using the vector-based approach, 3D gaze points are defined as the midpoint of the shortest straight line between the two

gaze vectors (see also Epelboim et al., 1995; Pfeiffer et al., 2009; Wibirama & Hamamoto, 2012, 2014).

The two gaze vectors $\vec{l}$ and $\vec{r}$ are either given by the eyetracking system or defined using the two position vectors of the eyes $\vec{e}_L$ and $\vec{e}_R$ and the two direction vectors located at the 2-D gaze points on the measuring plane $\vec{m}_L$ and $\vec{m}_R$.

$$\vec{l} = \vec{e}_L + \lambda_L * (\vec{m}_L - \vec{e}_L) \quad \text{with } \lambda_L \in \mathbb{R}, \quad (1a)$$

$$\vec{r} = \vec{e}_R + \lambda_R * (\vec{m}_R - \vec{e}_R) \quad \text{with } \lambda_R \in \mathbb{R}. \quad (1b)$$

Next, the point with the smallest distance to the two gaze vectors must be found. For this, the common perpendicular of both vectors is calculated.

Assumption: On the gaze vector of the left eye $\vec{l}$ there is a point P , and on the gaze vector of the right eye $\vec{r}$ there is a point Q . $\overrightarrow{PQ}$ is the shortest distance between the two vectors; it is assumed when

$$\overrightarrow{PQ} \perp \vec{l} \Rightarrow \overrightarrow{PQ} * (\vec{m}_L - \vec{e}_L) = 0 \quad (2a)$$

and

$$\overrightarrow{PQ} \perp \vec{r} \Rightarrow \overrightarrow{PQ} * (\vec{m}_R - \vec{e}_R) = 0. \quad (2b)$$

This means that $\overrightarrow{PQ}$ is perpendicular to both gaze vectors. $\overrightarrow{PQ}$ follows $\vec{P} - \vec{Q}$ and Eq. 1:

$$\overrightarrow{PQ} = (\vec{e}_L + \lambda_L * (\vec{m}_L - \vec{e}_L)) - (\vec{e}_R + \lambda_R * (\vec{m}_R - \vec{e}_R)). \quad (3)$$

Using Eqs. 2 and 3 results in

$$[(\vec{e}_L + \lambda_L * (\vec{m}_L - \vec{e}_L)) - (\vec{e}_R + \lambda_R * (\vec{m}_R - \vec{e}_R))] * (\vec{m}_L - \vec{e}_L) = 0. \quad (4)$$

and

$$[(\vec{e}_L + \lambda_L * (\vec{m}_L - \vec{e}_L)) - (\vec{e}_R + \lambda_R * (\vec{m}_R - \vec{e}_R))] * (\vec{m}_R - \vec{e}_R) = 0. \quad (5)$$

λ_L and λ_R can be calculated using Eqs. 4 and 5, and applying Eq. 1 results in the points P and Q .

The smallest distance between the two gaze vectors is the midpoint of $\overrightarrow{PQ}$, and the 3D gaze point is calculated with

$$\vec{g} = \frac{\vec{p} + \vec{q}}{2}. \quad (6)$$

Detection of 3D fixations

Following the computation of 3D gaze points, these can be aggregated into fixations. As was described above, we

decided to use a dispersion-based approach. For the dispersion-threshold of 2-D fixations a circular area is usually assumed (Salvucci & Goldberg, 2000). Therefore, the logical transfer from 2-D would be to use a sphere for 3D fixations. This, however, seems not to be the optimal choice because the distribution of 3D gazes around a fixation point is not radially symmetrical. The fluctuation of values in the z -dimension is larger than in the dimension of x and y . The smaller the convergence angle the higher the impact to the z -coordinate.¹ Using a sphere would require a large spatial threshold to compensate for the fluctuating z values, thereby inflating the area on the x - and y -axes and thus reducing the spatial resolution of identifiable fixations (Fig. 1a). Otherwise, applying a smaller spatial threshold would neglect the 3D gaze points that most likely belong to the fixation (Fig. 1b).

Since a sphere seems not to be the optimal choice to describe the distribution of 3D gaze points, we decided to use an ellipsoid, with the size depending on the distance between the observer and the fixation point (see Fig. 2).

To detect 3D fixations, for each 3D gaze point the algorithm tests whether or not the point lies within the ellipsoidal bounding volume (see below). If yes, the 3D gaze point is assigned to the current fixation. The size and 3D position of the ellipsoid is determined by the relation of the current fixation center and the eyes of the observer (see Fig. 3).

The detection of 3D fixations requires a definition of the ellipsoidal bounding volume, which is based on the following calculation procedure:

1. The midpoint between the two eyes, called cyclopedian eye position, is determined:

$$x_C = (x_L + x_R)/2; \quad y_C = (y_L + y_R)/2; \quad z_C = (z_L + z_R)/2. \quad (7)$$

2. The distance between the cyclopedian eye and the current fixation center is calculated:

$$r = \sqrt{(x_C - x_F)^2 + (y_C - y_F)^2 + (z_C - z_F)^2}. \quad (8)$$

3. The parameters of the ellipsoid are defined depending on the distance to the current fixation center and the threshold of acceptable gaze point deviation δ . The spatial extensions are calculated as follows:

$$s_x = s_y = r * \delta, \quad (9)$$

$$s_z = s_x * \frac{r}{d}. \quad (10)$$

¹ The mathematical relation is described in greater detail in the supplement.

4. Since the ellipsoid is not radially symmetrical, it has to be oriented to the cyclopedian eye. The yaw angle ψ (rotation around the y -axis) and the pitch angle θ (rotation around the x -axis) between the cyclopedian eye position and the current fixation center result from

$$\psi = +\tan^{-1} * \left(\frac{x_F - x_C}{z_F - z_C} \right). \quad (11)$$

ψ is positive when the gaze is directed toward the left of the vertical plane of the cyclopedian eye and negative when the gaze is directed toward the right.

$$\theta = -\tan^{-1} * \left(\frac{y_F - y_C}{z_F - z_C} \right). \quad (12)$$

θ is positive when the gaze is directed upward from the horizontal plane of the cyclopedian eye and negative when the gaze is directed downward.

Calculation of the distances between the new 3D gaze point and the current fixation center is then accomplished, using these equations:

$$\Delta x = x_G - x_F; \quad \Delta y = y_G - y_F; \quad \Delta z = z_G - z_F. \quad (13)$$

5. The gaze point deviations are rotated according to the ellipsoidal bounding volume of the current fixation:

$$\Delta x' \cong \Delta x * \cos \psi + \Delta z * \sin \psi, \quad (14a)$$

$$\Delta y' \cong \Delta y * \cos \theta + \Delta z * \sin \theta, \quad (14b)$$

$$\Delta z' \cong \Delta z * \cos \theta * \cos \psi - \Delta x * \sin \psi - \Delta y * \sin \theta. \quad (14c)$$

6. The algorithm tests whether the new 3D gaze point lies within or outside of the ellipsoidal bounding volume of the current fixation:

If

$$\left[\left(\frac{\Delta x'}{r_{x'}} \right)^2 + \left(\frac{\Delta y'}{r_{y'}} \right)^2 + \left(\frac{\Delta z'}{r_{z'}} \right)^2 \right] < 1, \quad (15)$$

then

the new 3D gaze point lies within the ellipsoid bounding volume of the current fixation,

else

the new 3D gaze point lies outside the ellipsoid bounding volume of the current fixation.

7. If the new 3D gaze point lies within the ellipsoidal bounding volume, it is added to the current fixation, and the new fixation parameters (3D position of the center, fixation duration) are calculated.
8. With the next sample, the verification of the fixation hypothesis starts again with Step 1.

Fig. 1 Two examples of a spherical area (blue) for the detection of 3D fixations from 3D gaze points (orange), with a large spatial threshold (a) or a small spatial threshold (b)

Figure 4 illustrates, in an exemplary manner, the ellipsoidal bounding volume. Additionally, the minimum fixation duration needs to be specified as a temporal threshold.

The toolkit Gaze3DFix

We have implemented the described algorithms for the computation of 3D gaze points in the dynamic linked library Gaze3D.dll. The described calculation steps for the detection of 3D fixations by an ellipsoidal bounding volume could also be used in other dispersion-based fixation detection algorithms. Here, we are exemplarily building up on the algorithm of LC Technologies, Inc. (2014) in Fixations3D.dll. In both libraries the mathematical calculations are realized as

functions that can be called from different programming environments and are usable online during the recording of eye movements as well as offline after the recording process. To facilitate the application, we provide Gaze3DFixGUI.exe, a simple graphical user interface (GUI) for importing binocular eye tracking data and calculating the related 3D gaze points and 3D fixations. A more in depth description of the use of the libraries and Gaze3DFixGUI.exe can be found in the supplement of this article. The toolkit Gaze3DFix including the two dynamic linked libraries (Gaze3D.dll and Fixations3D.dll), and the GUI (Gaze3DFixGUI.exe) is available as open-source software. Thus, researchers can use and modify the libraries or the user interface for their work. They are available along with a manual at github under <https://github.com/applied-cognition-research/Gaze3DFix>.

Fig. 2 Schematic representation of the measurement accuracy of a binocular eyetracker, with measurement accuracy α (blue) for both gaze vectors and the resulting elliptical fluctuation range (red) for the 3D gaze point. The size of the ellipsoid defined between the intersection points I_N (near) and I_F (far) varies with the distance between the observer and the fixation point

Evaluation of the approach

Two experiments were conducted to test the accuracy of the suggested method for the computation of 3D gaze points and the detection of 3D fixations. In both experiments participants had to fixate predefined stimuli while their eye movements were tracked. In Experiment 1, real stimuli were shown at different locations, in Experiment 2, the same setup was replicated with virtual stimuli. Both experiments will be reported together due to their similarity.

Method

Participants

In Experiment 1, 12 participants (seven men, five women; 26 to 47 years of age, $M = 33$) took part, and in Experiment 2, 24 participants (12 men, 12 women; 25 to 49 years of age, $M = 32$) took part. All participants had normal or corrected-to-normal visual acuity. In Experiment 2, the stereopsis was also

Fig. 3 Ellipsoidal bounding volume for the detection of fixations

determined using the Titmus Fly Test. One participant failed the stereopsis test and was therefore excluded.

Apparatus and material

In both experiments, eye movements were binocularly tracked with the remote eyetracker Eyegaze EyeFollower™ 2 System (LC Technologies, Inc.) with 120 Hz in bright pupil mode. The EyeFollower™ uses four cameras, two for each eye; allows free head movements within an area of $96 \times 96 \times 50$ cm; and has an accuracy within 0.4 deg, as claimed by the manufacturer. For Experiment 1 the standard 2-D eyetracker calibration was done using a desktop monitor (24-in., $1,920 \times 1,080$ pixels, 60 Hz) that was positioned 700 mm in front of the participant. Participants were carefully positioned and a chinrest ensured a stable head location with the participants' eyes 50 mm below the upper monitor edge. Between the

participant and the monitor an acrylic glass panel on stilts was put up from which 15 steel spheres could be lowered in 45 positions. The spheres were positioned 200, 300, 400, 500, and 600 mm in front of the participant in three columns and three rows (Fig. 5a) gives a schematic overview of the possible stimulus positions). The stimulus positions formed a field of view of 37.56° horizontal and 24.63° vertical, which corresponds to the measurement range of the used eyetracker. The steel spheres had a diameter of 7 mm and were tagged with a fixation target (see Fig. 5b).

For Experiment 2, the standard 2-D eyetracker calibration (monoscopic) and stereoscopic stimulus presentation were done on a 3D monitor (ASUS VG236, 23-in., $1,920 \times 1,080$ pixels, 120 Hz). The monitor was synchronized with shutter glasses (NVIDIA® 3DVISION™) and positioned 600 mm in front of the participant. The shorter distance to the monitor and the eyetracker than in Experiment 1 was required in order to achieve sufficient intensity of infrared light as some amount of light was blocked by the shutter glasses. Replicating the distances of the real stimuli, spheres were presented in a distance of 200, 300, 400, and 500 mm in front of the participants in three columns and three rows (see Fig. 5c). Since replicating the exact positions of Experiment 1 led to stereoscopic images that did not fit on the monitor, we moved the stimuli closer together on the x - and y -axes, leading to a field of view of 24.60° horizontal and 13.79° vertical. The virtual spheres had a diameter of 7 mm, were marked with a fixation cross, and were rendered either dark in front of a light background or light in front of a dark background (see Fig. 5d). The stereoscopic stimuli were rendered with the framework Bildsprache

Fig. 4 Example of an ellipsoidal bounding volume (yellow) for the detection of 3D fixations from 3D gaze points (orange)

Fig. 5 Schematic representation of the setups for Experiment 1 (a) and Experiment 2 (c), with example stimuli from both Experiment 1 (b) and Experiment 2 (d)

Live Lab (BiLL; Wojdziak, Kammer, Franke, & Groh, 2011) individually for each participant's interpupillary distance.

Procedure

After a demographic questionnaire and the stereopsis screening test in Experiment 2, participants were positioned and received the instruction. Before the experiment, a 9-point calibration was performed. The standard 2-D calibration targets formed a field of view of 37.80° horizontal and 21.83° vertical in Experiment 1 and a field of view of 43.60° horizontal and 25.36° vertical in Experiment 2. If the average of the distances between target position and estimated gaze position was above 0.3° , calibration procedure was repeated. After a test trial and a short break to clarify possible questions the calibration procedure was repeated before the experimental session started with either 45 (Exp. 1) or 72 (Exp. 2) trials. In Experiment 2, calibration was done on a dark background, followed by a block of 36 trials with light target spheres on a dark background, and then calibration was done on a light background followed by a block of 36 trials with dark target spheres in front of a light background; the order of the blocks was counterbalanced across participants. This additional independent variable in Experiment 2 was implemented because of results from research on binocular coordination in reading

(Huckauf, Watrin, Yuras, & Koepsel, 2013; Liversedge, Rayner, White, Findlay, & McSorley, 2006; Nuthmann & Kliegl, 2009). Those results suggested that whether white-on-black or black-on-white stimuli are used might affect the convergence distance. A trial started either with the experimenter lowering one target sphere at the prescribed position (Exp. 1) or with one virtual target sphere moving from the monitor plane to the prescribed position (Exp. 2). The participant had to fixate the target stimulus in order to fuse the stereoscopic image. Participants pressed a button to terminate the trial in cases in which the stereoscopic image could not be fused. If participants were able to fixate the target stimulus appropriately, this was confirmed by pressing another button. After the button press, the participants had to maintain the fixation for a duration of 1,000 ms. A subsequent acoustic signal indicated that the end of the required time interval was reached—that is, the end of the trial. Between trials participants could look around freely and rest their eyes while the sphere was either removed by the experimenter (Exp. 1) or blanked out (Exp. 2).

Data analysis

From the eyetracking data, only samples between the confirmatory button press of the participant and the acoustic signal

were processed further. Samples with invalid values due to a blink or gaze locations outside the measuring range (frustum between the participants' eyes and eye tracker measuring plane) were filtered, as well as trials in which the participants were not able to fuse the stereoscopic images. In Experiment 1 we excluded 6.15%, and in Experiment 2 8.75%, of the data. For the remaining binocular samples, the 3D gaze point was computed using the library described above. From the 3D gaze points, 3D fixations were detected and their positions computed using the algorithm described above. We set the minimum duration to 100 ms and manipulated the dispersion threshold as an independent variable with the levels 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, and 2.0 deg of visual angle.

The analyses were done using IBM Statistics SPSS 19, Mathworks MATLAB 7.11.0 and NYAN 2.0XT. To test accuracy, two measures for the deviation between the computed positions of the 3D gaze points and 3D fixations and the stimulus positions were determined. As a first measure, the 3D deviation as the absolute value of the vector from the stimulus position (S) to the computed 3D gaze position (G) was calculated using the Euclidean distance:

$$3D \text{ deviation} = |\vec{SG}| = \sqrt{(x_S - x_G)^2 + (y_S - y_G)^2 + (z_S - z_G)^2}. \quad (16)$$

As a second measure, the deviation along the z -axis was considered and calculated using the following formula:

$$z \text{ deviation} = z_S - z_G. \quad (17)$$

Medians were calculated for each participant and each condition. We used medians instead of means as the distributions for the 3D deviations and fixation durations were expected to be skewed to the right (see, e.g., Helo, Pannasch, Sirri, & Rama, 2014). For a right-skewed distribution the median represents a more reliable measure of central tendency than the mean. Repeated measures analyses of variance (ANOVAs) were applied to analyze deviations for the different stimulus

distances and the different dispersion thresholds, respectively. Greenhouse–Geisser correction (ϵ) was applied when the assumption of sphericity was violated. As an indication of effect size, eta-squared (η^2) is reported as recommended by Levine and Hullett (2002).

Results

First, the accuracy of the 3D gaze points depending on the stimulus distance was considered for both the real and the virtual stimuli. Second, the 3D fixation positions depending on the dispersion threshold used and the stimulus distance were compared.

3D gaze points

The 3D gaze points had mean 3D deviations from the stimulus positions of 16 mm for the real and 17 mm for the virtual stimuli. One-way repeated measure ANOVAs revealed a significant effect of stimulus distance on 3D deviations for real, $F(4, 44) = 58.494$, $p < .001$, $\epsilon = .509$, $\eta^2 = .842$, as well as for virtual, $F(3, 66) = 73.697$, $p < .001$, $\epsilon = .734$, $\eta^2 = .770$, stimuli, with greater deviations for larger fixation distances (see Fig. 6a). The mean z deviations of the 3D gaze points from the stimulus positions were 2 mm for the real and 5 mm for the virtual stimuli. One-way repeated measures ANOVAs showed that the effect of stimulus distance on the z deviation was not significant, for either real, $F < 1$, or virtual, $F(3, 66) = 1.104$, $p = .328$, $\epsilon = .506$, stimuli (see Fig. 6b).

3D fixations

Using the foveal resolution of about 2 deg of visual angle, we examined for the detection of 3D fixations, spatial thresholds of 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, and 2.0 deg of visual angle. As expected, the number

Fig. 6 3D deviations (a) and z deviations (b) of the 3D gaze points from the stimulus position as a function of stimulus distance, for both real and virtual stimuli. Error bars represent the 95% confidence intervals

Fig. 7 3D deviations of the positions of the 3D fixations from the stimulus position as a function of spatial threshold (a) and stimulus distance (b), for both real and virtual stimuli. Error bars represent the 95% confidence intervals

of detected fixations decreased with greater spatial thresholds, while the fixation duration consequently increased. The positions of the 3D fixations had a mean 3D deviation from the stimulus positions of 17 mm for real as well as for virtual stimuli. One-way repeated measures ANOVAs showed that the spatial threshold did not significantly influence the 3D deviations for real stimuli, $F < 1$, but it did for virtual stimuli, $F(10, 220) = 3.778$, $p = .017$, $\varepsilon = .283$, $\eta^2 = .147$ (see Fig. 7a). Again, one-way repeated measures ANOVAs revealed a significant effect of stimulus distance on the 3D deviations for real stimuli, $F(4, 40) = 46.890$, $p < .001$, $\varepsilon = .521$, $\eta^2 = .824$, and for virtual stimuli, $F(3, 66) = 30.690$, $p < .001$, $\varepsilon = .759$, $\eta^2 = .582$, with greater deviations for larger fixation distances (see Fig. 7b).

Discussion

To test the accuracy of our proposed method, real and virtual stimuli were presented, and the 3D gaze points and 3D fixations were calculated. For the tested distances of 200 to 600 mm, we found good congruence between the stimulus position and the 3D gaze points as well as the 3D fixation locations for both real and virtual stimuli. In fact, the results concerning real and virtual stimuli were highly similar. Looking at different stimulus distances, we found higher 3D gaze deviations (and thus higher 3D deviations of the 3D fixations) for increased fixation distances. However, the z deviations did not vary with distance. This means that there was no systematic distortion, like a constant under- or overestimation, but simply a higher variance. This was probably due to the smaller vergence angle for farther distances. The smaller the vergence angle, the more the computed 3D gaze point is affected by measurement errors of the eyetracking system. The spatial threshold setting for the detection of fixations did influence the number of detected fixations, and consequently the

fixation durations. The accuracy of the 3D fixation location, however, was not affected for real stimuli, and was only slightly affected for virtual stimuli. One reason for the obtained high accuracy might be that we used stimuli that were small and easy to fixate, leading to exact fixations by the participants. However, here we focused on accuracy; therefore, artificial fixations were generated by means of the instructions and the presentation of isolated stimuli. Hence, the present findings do not allow us to draw conclusions about the quality of the algorithms' output for the detection of 3D fixations in a free-viewing paradigm with natural sequences of fixations and saccades.

Since the measurement error increases with fixation distance, our method is suitable for stimuli within reaching distance, which is the case for all approaches that build on convergent eye movements. A further requirement is that the users' eye positions must be known, which means that the eye positions need to be measured and either kept stable or tracked continuously. This would, for example, be the case with the eyetracker we used in the present evaluation, since the EyeFollower™ provides the 3D positions of the eyes. However, in our experimental evaluation, head movements were restricted by using a chinrest, to avoid influences of head movements on the results, and we used a simple fixation task with stationary targets. Thus, the present results are limited, and we do not know how the method would cope with situations in which the targets or the observer moved, and thus in which slow eye movements, such as smooth pursuit movements or vestibulo-ocular reflex, were present. Further experiments will be needed to clarify in which situations our approach to detecting 3D fixations with an ellipsoidal bounding volume provides valid results, and in which it might not be sufficient.

Conclusion

In this article we have presented a new approach for the detection of 3D fixations using an ellipsoidal bounding volume.

The proposed algorithm, as well as an algorithm to calculate the necessary 3D gaze points using a vector-based approach, are implemented as dynamic linked libraries and can be assessed through GUI-based software.

The proposed approach has the advantage of not being hardware- or application-specific, which allows for high flexibility in its use. Furthermore, no 3D calibration is required, since the normal 2-D calibration of the eyetracker is sufficient. With the use of the dynamic linked libraries, the algorithms are applicable online during the recording of eye movements. Offline analyses of binocular eye movements can be done using Gaze3DFixGUI.exe, which has a GUI, thus making our algorithms available for researchers not well versed with programming.

A requirement of our method is that for both the computation of 3D gaze points and the detection of 3D fixations, the users' eye positions must be known. This means that the eye positions need to be measured and either kept stable or tracked continuously. The present results are limited to a simple and artificial environment; future testing will need to examine the quality of the detection of 3D fixations in free-viewing paradigms.

The Gaze3DFix toolkit is available open-source at github, <https://github.com/applied-cognition-research/Gaze3DFix>. We hope that the availability of ready-to-use algorithms for analyses of 3D eye movements will boost research in this field. Furthermore, using the same algorithms would allow for better comparisons between the results of future experiments.

Author note S.W. was co-financed by the European Social Fund and the Free State of Saxony, Germany (SAB 100080340). R.S.S. was supported by a grant from the German Research Council (Deutsche Forschungsgemeinschaft, DFG; VE 192/17-1). The authors thank Dixon Cleveland (LC Technologies, Inc.) for technical support, and Dietrich Kammer (Chair of Media Design, TU-Dresden) for programming the stereoscopic visualization.

References

- Behrens, F., MacKeben, M., & Schröder-Preikschat, W. (2010). An improved algorithm for automatic detection of saccades in eye movement data and for calculating saccade parameters. *Behavior Research Methods*, 42, 701–708. <https://doi.org/10.3758/BRM.42.3.701>
- Blythe, H. I., Holliman, N. S., Jainta, S., Tbaily, L. W., & Liversedge, S. P. (2012). Binocular coordination in response to two-dimensional, three-dimensional and stereoscopic visual stimuli. *Ophthalmic and Physiological Optics*, 32, 397–411. <https://doi.org/10.1111/j.1475-1313.2012.00926.x>
- Collewijn, H., Erkelens, C. J., & Steinman, R. M. (1997). Trajectories of the human binocular fixation point during conjugate and non-conjugate gaze-shifts. *Vision Research*, 37, 1049–1069.
- Collewijn, H., Steinman, R. M., Erkelens, C. J., Pizlo, Z., & van der Steen, J. (1992). Effect of freeing the head on eye movement characteristics during three-dimensional shifts of gaze and tracking. In A. Berthoz, W. Graf, & P. P. Vidal (Eds.), *The head-neck sensory motor system* (pp. 412–418). Oxford, UK: Oxford University Press.
- Daugherty, B. C., Duchowski, A. T., House, D. H., & Ramasamy, C. (2010). Measuring vergence over stereoscopic video with a remote eye tracker. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (pp. 97–100). New York, NY, USA: ACM Press.
- Diaz, G., Cooper, J., Kit, D., & Hayhoe, M. (2013). Real-time recording and classification of eye movements in an immersive virtual environment. *Journal of Vision*, 13(12), 5. <https://doi.org/10.1167/13.12.5>
- Duchowski, A. T. (2007). *Eye tracking methodology: Theory and practice* (Vol. 373). New York, NY: Springer Science & Business Media.
- Duchowski, A. T., House, D. H., Gestring, J., Congdon, R., Świrski, L., Dodgson, N. A., ... Krejtz, I. (2014). Comparing estimated gaze depth in virtual and physical environments. In P. Qvarfordt & D. Witzner Hansen (Eds.), *Proceedings of the Symposium on Eye Tracking Research and Applications* (pp. 103–110). New York, NY, USA: ACM Press.
- Duchowski, A. T., Medlin, E., Cournia, N., Murphy, H., Gramopadhye, A., Nair, S., ... Melloy, B. (2002). 3D eye movement analysis. *Behavior Research Methods, Instruments, & Computers*, 34, 573–591. <https://doi.org/10.3758/BF03195486>
- Duchowski, A. T., Medlin, E., Gramopadhye, A., Melloy, B., & Nair, S. (2001). Binocular eye tracking in VR for visual inspection training. In C. Shaw, W. Wang, & M. Green (Eds.), *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (pp. 1–8). New York, NY, USA: ACM Press.
- Duchowski, A. T., Pelfrey, B., House, D. H., & Wang, R. I. (2011). Measuring gaze depth with an eye tracker during stereoscopic display. In S. N. Spencer (Ed.), *Proceedings of the Symposium on Applied Perception in Graphics and Visualization* (pp. 15–22). New York, NY, USA: ACM Press.
- Durgin, F. H., & Li, Z. (2010). Controlled interaction: Strategies for using virtual reality to study perception. *Behavior Research Methods*, 42, 414–420. <https://doi.org/10.3758/BRM.42.2.414>
- Epelboim, J., Steinman, R. M., Kowler, E., Edwards, M., Pizlo, Z., Erkelens, C. J., & Collewijn, H. (1995). The function of visual search and memory in sequential looking tasks. *Vision Research*, 35, 3401–3422.
- Essig, K., Pomplun, M., & Ritter, H. (2006). A neural network for 3D gaze recording with binocular eye trackers. *International Journal of Parallel, Emergent and Distributed Systems*, 21, 79–95. <https://doi.org/10.1080/17445760500354440>
- Grosjean, M., Rinkenauer, G., & Jainta, S. (2012). Where do the eyes really go in the hollow-face illusion? *PLoS ONE*, 7, e44706. <https://doi.org/10.1371/journal.pone.0044706>
- Hammoud, R. I. (2008). *Passive eye monitoring: Algorithms, applications and experiments*. New York, NY: Springer Science & Business Media.
- Hansen, D. W., & Ji, Q. (2010). In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32, 478–500.
- Helo, A., Pannasch, S., Sirri, L., & Rama, P. (2014). The maturation of eye movement behavior: Scene viewing characteristics in children and adults. *Vision Research*, 103C, 83–91. <https://doi.org/10.1016/j.visres.2014.08.006>
- Hennessey, C., & Lawrence, P. (2008). 3D point-of-gaze estimation on a volumetric display. In K.-J. Rähkä & A.T. Duchowski (Eds.), *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (p. 59). New York, NY: ACM Press.

- Huckauf, A., Watrin, L., Yuras, G., & Koepsel, A. (2013). Brightness and contrast effects on binocular coordination. Paper presented at the 55th Tagung experimentell arbeitender Psychologen, Vienna, Austria.
- Komogortsev, O. V., Gobert, D. V., Jayarathna, S., Koh, D. H., & Gowda, S. M. (2010). Standardization of automated analyses of oculomotor fixation and saccadic behaviors. *IEEE Transactions on Biomedical Engineering*, 57, 2635–2645. <https://doi.org/10.1109/TBME.2010.2057429>
- Land, M. F. (2006). Eye movements and the control of actions in everyday life. *Progress in Retinal and Eye Research*, 25, 296–324.
- Lappi, O. (2015). Eye tracking in the wild: The good, the bad and the ugly. *Journal of Eye Movement Research*, 8, 1–21. [10.16910/jemr.8.5.1](https://doi.org/10.16910/jemr.8.5.1)
- Lappi, O. (2016). Eye movements in the wild: Oculomotor control, gaze behavior and frames of reference. *Neuroscience & Biobehavioral Reviews*, 69, 49–68.
- LC Technologies. (2014). *Eyegaze edge analysis system: Programmer's manual*. Fairfax, VA, USA: LC Technologies, Inc.
- Levine, T. R., & Hullett, C. R. (2002). Eta squared, partial eta squared, and misreporting of effect size in communication research. *Human Communication Research*, 28, 612–625. <https://doi.org/10.1111/j.1468-2958.2002.tb00828.x>
- Liversedge, S. P., Rayner, K., White, S. J., Findlay, J. M., & McSorley, E. (2006). Binocular coordination of the eyes during reading. *Current Biology*, 16, 1726–1729. <https://doi.org/10.1016/j.cub.2006.07.051>
- Mansouryar, M., Steil, J., Sugano, Y., & Bulling, A. (2016). 3D gaze estimation from 2D pupil positions on monocular head-mounted eye trackers. *Proceedings of the International Symposium on Eye Tracking Research and Applications (ETRA)*. 197–200
- Munn, S. M., Stefano, L., & Pelz, J. B. (2008). Fixation-identification in dynamic scenes: Comparing an automated algorithm to manual coding. In S. Creem-Regehr & K. Myszkowski (Eds.), *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization* (pp. 33–42). New York, NY: ACM Press.
- Nuthmann, A., & Kliegl, R. (2009). An examination of binocular reading fixations based on sentence corpus data. *Journal of Vision*, 9(5), 31. <https://doi.org/10.1167/9.5.31>
- Nyström, M., & Holmqvist, K. (2010). An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, 42, 188–204. <https://doi.org/10.3758/brm.42.1.188>
- Pelz, J. B., & Canosa, R. (2001). Oculomotor behavior and perceptual strategies in complex tasks. *Vision Research*, 41, 3587–3596.
- Pfeiffer, T., Latoschik, M. E., & Wachsmuth, I. (2009). Evaluation of binocular eye trackers and algorithms for 3D gaze interaction in virtual reality environments. *Journal of Virtual Reality and Broadcasting*, 5, 1660.
- Pobuda, M., & Erkelens, C. J. (1993). The relationship between absolute disparity and ocular vergence. *Biological Cybernetics*, 68, 221–228. <https://doi.org/10.1007/BF00224855>
- Rambold, H., Neumann, G., Sander, T., & Helmchen, C. (2006). Age-related changes of vergence under natural viewing conditions. *Neurobiology of Aging*, 27, 163–172. <https://doi.org/10.1016/j.neurobiolaging.2005.01.002>
- Reimer, B., & Sodhi, M. (2006). Detecting eye movements in dynamic environments. *Behavior Research Methods*, 38, 667–682.
- Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. *Proceedings of the 2000 Symposium on Eye Tracking Research and Applications* (pp. 71–78). New York, NY, USA: ACM Press.
- Semmlow, J. L., Hung, G. K., & Ciuffreda, K. J. (1986). Quantitative assessment of disparity vergence components. *Investigative Ophthalmology and Visual Science*, 27, 558–564.
- Shic, F., Scassellati, B., & Chawarska, K. (2008). The incomplete fixation measure. *Proceedings of the 2008 Symposium on Eye Tracking Research and Applications* (pp. 111–114). New York, NY, USA: ACM.
- SR Research LTD. (2009). *EyeLink® 1000 user manual—Version 1.5.0*. Mississauga, Ontario, Canada.
- Wang, R. I., Pelfrey, B., Duchowski, A. T., & House, D. H. (2012). Online gaze disparity via binocular eye tracking on stereoscopic displays. In *Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission* (pp. 184–191). Washington, DC, USA: IEEE Computer Society.
- Wang, R. I., Pelfrey, B., Duchowski, A. T., & House, D. H. (2014). Online 3D gaze localization on stereoscopic displays. *ACM Transactions on Applied Perception*, 11, 1–21. <https://doi.org/10.1145/2593689>
- Wibrama, S., & Hamamoto, K. (2012). A geometric model for measuring depth perception in immersive virtual environment. In *Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction* (pp. 325–330). New York, NY: ACM Press.
- Wibrama, S., & Hamamoto, K. (2014). 3D gaze tracking on stereoscopic display using optimized geometric method. *IEEE Transactions on Electronics, Information and Systems*, 134, 345–352.
- Wismeijer, D., van Ee, R., & Erkelens, C. J. (2008). Depth cues, rather than perceived depth, govern vergence. *Experimental Brain Research*, 184, 61–70. <https://doi.org/10.1007/s00221-007-1081-2>
- Wojdziak, J., Kammer, D., Franke, I. S., & Groh, R. (2011). BiLL: An interactive computer system for visual analytics. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (p. 264). New York, NY: ACM Press.