# Accuracy of Monocular Gaze Tracking on 3D Geometry

Xi Wang, David Lindlbauer, Christian Lessig and Marc Alexa

**Abstract** Many applications such as data visualization or object recognition benefit from accurate knowledge of where a person is looking at. We present a system for accurately tracking gaze positions on a three dimensional object using a monocular . We accomplish this by 1) using digital manufacturing to create stimuli whose geometry is know to high accuracy, 2) embedding fiducial markers into the manufactured objects to reliably estimate the rigid transformation of the object, and, 3) using a perspective model to relate pupil positions to 3D locations. This combination enables the efficient and accurate computation of gaze position on an object from measured pupil positions. We validate the accuracy of our system experimentally, achieving an angular resolution of 0.8° and a 1.5% depth error using a simple calibration procedure with 11 points.

## 1 Introduction

Understanding the viewing behavior of humans when they look at objects plays an important role in applications such as data visualization, scene analysis, object recognition, and image generation [33]. The viewing behavior can be analyzed by measuring fixations using eye tracking. In the past, such experiments, especially for object exploration tasks, were performed with flat 2D stimuli presented on a screen [12]. However, since the human visual attention mechanism has been developed in 3D environments, depth may have an important effect on viewing behavior [20]. To understand the role of depth information, some studies [15, 21, 8] recently combined eye tracking with stereoscopic displays. However, these displays fail to provide natural depth cues; for example they suffer from stereoscopic decoupling, the mismatch of accommodation and vergence for the displayed depth [13].

Xi Wang · David Lindlbauer · Christian Lessig · Marc Alexa
TU Berlin
email: xi.wang | david.lindlbauer | christian.lessig | marc.alexa@tu-berlin.de

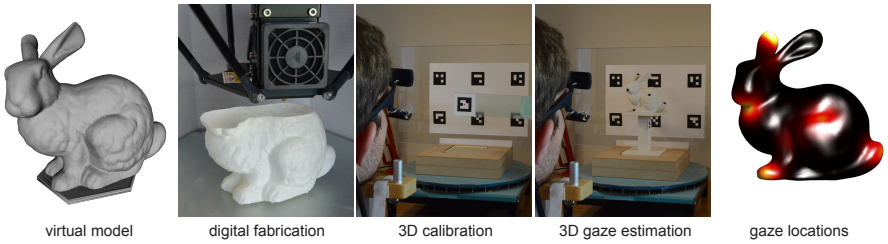


**Fig. 1** We accurately estimate 3D gaze positions by combining digital manufacturing, marker tracking and monocular eye tracking. With a simple procedure we attain an angular accuracy of 0.8°.

Since our research objective is to investigate the viewing behavior of humans for stimuli that are genuinely three-dimensional, we need to be able to track 3D gaze positions with high accuracy.

Standard eye tracking setups only determine human's viewing direction. The most common approach for determining viewing depth is to employ a binocular eye tracker and measure eye , that is the orientation difference between the left and the right eye that ensures both are focused on the same point in space. However, as exemplified in Fig. 2, experimentally determining depth from binocular vergence is inherently ill-conditioned. Even for an object at a modest distance the eyes and the object form a highly acute triangle so that the inevitable inaccuracies in measuring pupil positions [12] lead to large errors in the estimated depth values. Although non-linear mappings can be employed to reduce the error [7, 23, 1, 11, 19, 22, 26], these require complex calibration and expensive optimization of the mapping while still leading to relatively large inaccuracies.

We base our approach on a mapping between viewing directions gathered by an eye tracker and the physical world. This is done similar to EyeSee3D [27] by tracking fiducial markers in physical space with a camera mounted on the eye tracker. We extend their approach by not only acquiring establishing which object is looked at but also determining the exact 3D gaze position on the particular object. The main components to achieve such accurate tracking are:

1. are generated by digital manufacturing so that their geometry is known to high accuracy and also available in digital form without imposing restrictions on the geometry that is represented.
2. are integrated into the 3D stimuli in order to reliably and accurately estimate the stimuli's 3D position relative to the head.
3. A simple calibration procedure that allows for an accurate computation of the from 3D positions to monocular pupil positions.
4. An for the mapping enables the computation of plausible positions on the 3D stimulus.

Our results demonstrate that for typical geometries we are able to obtain 0.8° angular resolution and reliable depth values within 1.5% of the true value, including

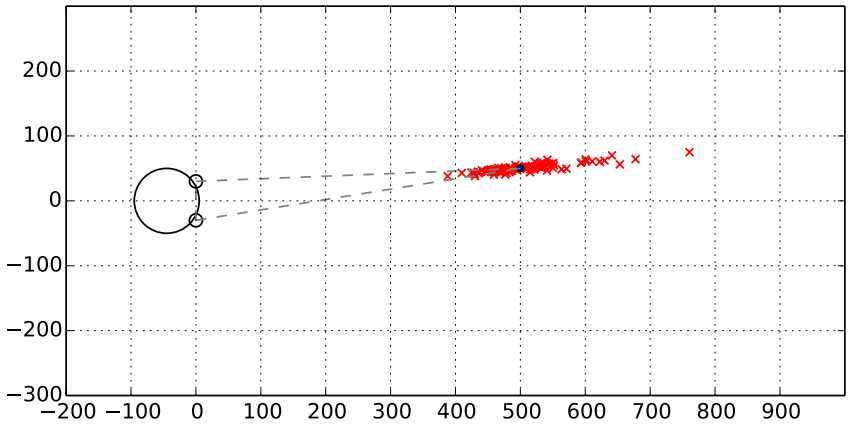


**Fig. 2** Inherent error of vergence-based depth estimation for an object at a distance of 500 mm away from the eyes. The red crosses mark estimated 3D positions for normally distributed gaze directions with mean equal to the correct angle for the object (black dot) and a variance of 0.5°. The highly acute triangle that leads to the ill-conditioning of the depth calculation is shown as dashed lines. The worst case relative error is almost 50%.

around silhouettes where the geometry has a large slope and depth estimation is hence particularly difficult. We accomplish this with only a and an 11-point calibration procedure.

In the next section, we discuss related work on . Subsequently, in Sec. 3, we detail our setup and explain how 3D positions can be related to pupil coordinates. This is followed by a discussion of how 3D viewing positions can be obtained from pupil positions in Sec. 4. Experimental results verifying the accuracy of our approach are presented in Sec. 5. We conclude the paper in Sec. 6 with a discussion and possible directions for future work.

## 2 Related Work

The viewing behavior of humans is typically analyzed using eye tracking by measuring a subject's fixations. However, usually only flat 2D stimuli on a screen are employed, e.g. [3, 16, 24, 28], even when one is interested in 3D objects. Only recently the first studies considering the effect of depth were performed. Lang et al. [21] collected a large eye fixation database for still images with depth information presented on a stereoscopic display. Their results show that depth can have a significant influence on a subject's fixations. Jansen et al. [15] also employed a stereoscopic display to analyze the effect of depth, demonstrating that depth information leads to an overall increase in spatial distribution of gaze positions for visual exploration tasks. Both Lang et al. [21] and Jansen et al. [15] report that visual

attention shifts over time from objects closer to the viewer to those farther away. Differences in fixations between 2D and 3D stimuli were recently also investigated for stereoscopic video [8, 9, 14, 29]. Discrepancies were mainly observed for scenes that lack on obvious (high-level) center of attention, with fixations having a larger spatial distribution when depth information is present.

Existing work investigating the role of depth information on fixation locations hence demonstrates that, at least under certain circumstances, depth has a significant effect on a subject's viewing behavior. However, so far only stereoscopic displays were employed, which do not provide all depth cues and suffer from stereoscopic decoupling [13]. Moreover, Duchowski et al. [6] showed that for stereoscopic displays the gaze depth of subjects does not fully correspond to the presented depth. Therefore, we believe that to understand viewing behavior for 3D objects, one should study stimuli that are genuinely three-dimensional. This provides the principal motivation for our work.

With 3D stimuli, also the depth values of fixation points have to be determined. The most common approach for obtaining fixation depth is to measure the vergence using a binocular eye tracker. However, computing depth values from binocular vergence is ill-conditioned since already for modest distances minuscule measurement errors in the pupil positions lead to large depth errors, cf. Fig. 2. To improve the accuracy, Essig et al. [7] trained a neural network that maps from eye vergence to depth values. Maggia et al. [23] proposed a somewhat simpler but also nonlinear model for the mapping from measured disparity to depth. Building on these works, current techniques [1, 11, 19, 22, 26] that employ binocular vergence to determine fixation depth obtain an error that is within 10% of the correct depth value.

Our work was inspired by existing approaches relating view directions to *known* geometry, e. g. in applications of virtual reality [32, 5]. Pfeiffer and Renner used fiducial markers to align the physical world to camera space [27]. By measuring eye vergence, they achieved an angular accuracy of 2.25 degrees, which gives correctly classified fixation targets on the scale of whole objects. However, for investigating human viewing behavior on the surface of 3D objects, more accurate gaze tracking is required. Consequently, we create a setup with the goal of tracking visual attention on 3D objects.

## 3 From 3D positions to pupil coordinates

In this section we describe our setup and how it enables to accurately determine gaze positions on an object. We use a monocular head mounted eye tracking device with a front facing world camera capturing the environment and an eye facing camera capturing the pupil movement.

The world camera yields the position and orientation of fiducial markers, for example fixed to objects, relative to the subject's head relative to its reference frame. A projective mapping is then relates these 3D coordinates to pupil positions relative

to the camera tracking the eye. This establishes a mapping between points in 3D space and pupil coordinates (this basic idea is illustrated in Figure 3).

The mapping is calibrated by having a subject focus on markers at different locations, including varying depths. Once the mapping is established, 2D pupil positions can be turned into rays corresponding to gaze directions in 3D space. The gaze directions then determine the 3D positions on the object a subject is looking at, by intersecting the rays with the known 3D geometry.

In the following we will describe these steps in more detail.

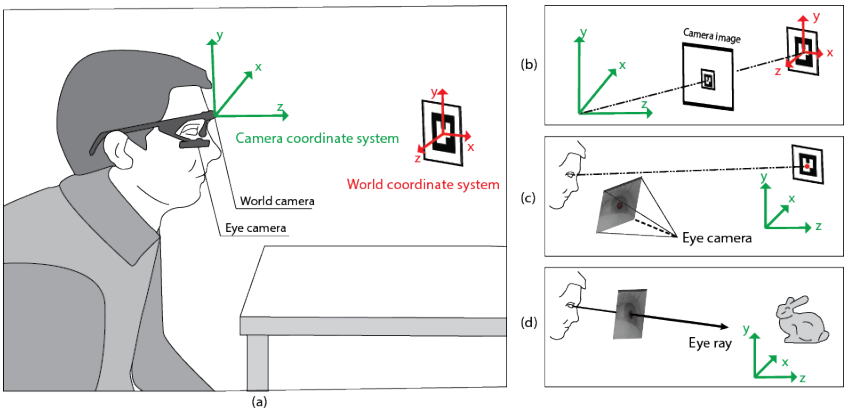


**Fig. 3** The main idea of our approach is to establish a mapping between points in 3D space (i.e. world coordinate system) and pupil coordinates in the image coordinate system of the eye camera. We consider all 3D positions relative to the coordinate system of the world camera (i.e. camera coordinate system). (b) A point in world coordinate system is first transformed into the camera coordinate system. (c) We model the mapping between pupil position in the eye camera image and a location in world camera space as projection. (d) From the estimated projective transformation, we can estimate a corresponding eye ray for each pupil position.

## *3.1 From local 3D positions to world-camera coordinates*

We employ fiducial markers to determine the 3D coordinates of locations in space in the world camera coordinate system. The mapping of a position $\mathbf{x} \in \mathbb{R}^3$, for example a point on a marker, to its projection $\mathbf{m} \in \mathbb{R}^2$ in the world camera image is given by

$$\begin{pmatrix} \mathbf{m} \\ 1 \end{pmatrix} = \mathbf{K}(\mathbf{R}\mathbf{x} + \mathbf{t}), \quad \mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{I} \tag{1}$$

where $\mathbf{K} : \mathbb{R}^3 \to \mathbb{R}^2$ is the intrinsic world camera matrix, modelling the perspective mapping, and $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation of the camera, forming the

rigid transformation. The mapping of $\mathbf{x}$ to its representation $\mathbf{w} \in \mathbb{R}^3$ in the world camera coordinate system is hence

$$\mathbf{w} = \mathbf{R}\mathbf{x} + \mathbf{t}. \tag{2}$$

We determine the intrinsic world camera matrix $\mathbf{K}$, which includes both radial and tangential distortion, in a preprocessing step using the approach of Heng et al. [10]. To determine the rigid transformation given by $\mathbf{R}$ and $\mathbf{t}$ we exploit that detected marker corner points $\mathbf{m}_i \in \mathbb{R}^2$ in the camera image have known 3D locations $\mathbf{x}_i \in \mathbb{R}^3$ in the marker's local coordinate system. Given at least three such points $\mathbf{m}_i$ in the camera image, we can determine $\mathbf{R}$ and $\mathbf{t}$ by minimizing the reprojection error.

Once $\mathbf{R}$ and $\mathbf{t}$ have been estimated, we can employ Eq. 2 to determine the position of the center of the marker in the world camera coordinate system, as required for calibration, or to map an object with a fixed relative position to a marker into the space, as is needed to determine gaze positions.

### *3.2 From world camera coordinates to pupil positions*

Given positions $\mathbf{w} \in \mathbb{R}^3$ in the world camera coordinate system, obtained as described in the last section, we have to relate these to a person's gaze direction, described by pupil positions $\mathbf{p}$ in the eye camera image. We model the mapping as a projective transformation, because the cameras and the system of the eye (i.e. the head) are in fixed relative orientation and position. In homogeneous coordinates the transformation is given by

$$s \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix} = \mathbf{Q} \begin{pmatrix} \mathbf{w} \\ 1 \end{pmatrix} \tag{3}$$

where $\mathbf{Q} \in \mathbb{R}^{3\times 4}$ is a projection matrix that is unique up to scale. Given a set of correspondences $\{(\mathbf{w}_i, \mathbf{p}_i)\}$ between 3D points $\mathbf{w}_i$ in the world camera coordinate system and pupil positions $\mathbf{p}_i$ describing the gaze direction towards $\mathbf{w}_i$, we can determine $\mathbf{Q}$ by minimizing

$$E(\mathbf{Q}) = \sum_i \left\| s_i \begin{pmatrix} \mathbf{p}_i \\ 1 \end{pmatrix} - \mathbf{Q} \begin{pmatrix} \mathbf{w}_i \\ 1 \end{pmatrix} \right\|_2^2. \tag{4}$$

Fixing one coefficient of $\mathbf{Q}$ to eliminate the freedom on scale (we choose $\mathbf{Q}_{3,4} = 1$), this is a standard linear least squares problem. In practice, we solve this problem using correspondences $\{(\mathbf{w}_i, \mathbf{p}_i)\}$ obtained during calibration, as described in Sec. 5.

Since $\mathbf{Q}$ is a projective transformation we can factor it into an upper triangular intrinsic camera matrix $\mathbf{A}_Q$ and a rigid transformation matrix $\mathbf{T}_Q = (\mathbf{R}_Q, \mathbf{t}_Q)$. The factorization is given by

$$\mathbf{Q} = \mathbf{A}_Q \mathbf{T}_Q = (\mathbf{A}_Q \mathbf{R}_Q, \mathbf{A}_Q \mathbf{t}_Q) \tag{5}$$

and hence can be determined from the $RQ$ decomposition of the left $3 \times 3$ block $\mathbf{A}_Q\mathbf{R}_Q$ of $\mathbf{Q}$. It can be computed using the $QR$ decomposition as

$$\mathbf{J}(\mathbf{A}_Q\mathbf{R}_Q)^T\mathbf{J} = (\mathbf{J}\mathbf{A}_Q^T\mathbf{J})(\mathbf{J}\mathbf{R}_Q^T\mathbf{J}) \tag{6}$$

where $\mathbf{J}$ is the exchange matrix, which in our case is the column inversed version of the identity matrix.

### *3.3 From pupil positions to*

So far we have related 3D locations to pupil positions. To determine a gaze point on an object we also have to relate pupil positions to a cone of positions in space. This also corresponds to the angular accuracy of our setup.

With the intrinsic eye camera matrix $\mathbf{A}_Q$, as determined in the last section, we can relate a homogeneous pupil position $\hat{\mathbf{p}} = (\mathbf{p}, 1)^T$ to an associated ray $\mathbf{r}$ in 3D world camera space:

$$\hat{\mathbf{p}} = \mathbf{A}_Q\mathbf{r}. \tag{7}$$

The depth along $\mathbf{r}$ is indeterminate since $\mathbf{A}_Q$ is a projection matrix. The angle between two rays $\mathbf{r}_i, \mathbf{r}_j$, represented by pupil coordinates $\mathbf{p}_i, \mathbf{p}_j$, is hence given by

$$\cos\eta_{ij} = \frac{\mathbf{r}_i^T\mathbf{r}_j}{\|\mathbf{r}_i\|\|\mathbf{r}_j\|} = \frac{\hat{\mathbf{p}}_i^\mathsf{T}\mathbf{A}_Q^{-\mathsf{T}}\mathbf{A}_Q^{-1}\hat{\mathbf{p}}_j}{\|\mathbf{A}_Q^{-1}\hat{\mathbf{p}}_i\|\|\mathbf{A}_Q^{-1}\hat{\mathbf{p}}_j\|}. \tag{8}$$

This suggests to interpret the matrix $\mathbf{A}_Q^{-\mathsf{T}}\mathbf{A}_Q^{-1}$ as an induced inner product $\mathbf{M}_Q = (\mathbf{A}_Q\mathbf{A}_Q^\mathsf{T})^{-1}$ on homogeneous pupil coordinates. The angle $\eta_{ij}$ then becomes

$$\cos\eta_{ij} = \frac{\hat{\mathbf{p}}_i^\mathsf{T}\mathbf{M}_Q\hat{\mathbf{p}}_j}{(\hat{\mathbf{p}}_i^\mathsf{T}\mathbf{M}_Q\hat{\mathbf{p}}_i)^{1/2}(\hat{\mathbf{p}}_j^\mathsf{T}\mathbf{M}_Q\hat{\mathbf{p}}_j)^{1/2}}. \tag{9}$$

For multiple pairs $\mathbf{p}_i$, $\mathbf{p}_j$, Eq. 9 can be solved efficiently when the involved matrices are precomputed.

## 4 From pupil coordinates to locations on an object

Our objective is to determine a gaze position $\bar{\mathbf{w}} \in \mathbb{R}^3$ in space from a pupil position $\bar{\mathbf{p}}$ describing a gaze direction. Central to our approach for determining $\bar{\mathbf{w}}$ is that the geometry of the observed object is known to high accuracy. This is ensured by 3D printing the object $\mathscr{M}$ from its digital representation as a triangulated surface $\mathbf{M}$.

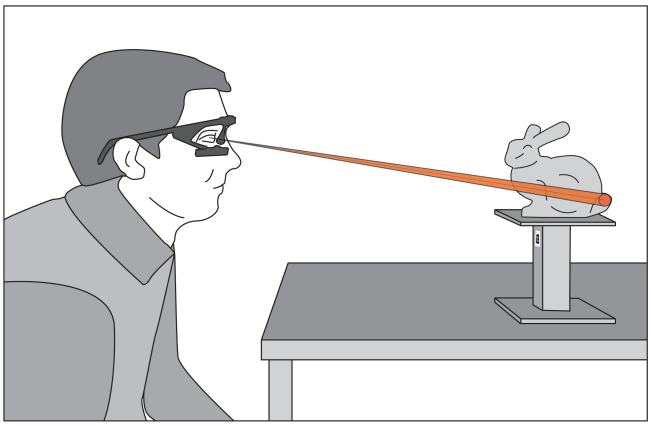

**Fig. 4** Pupil positions provide by the eye tracker correspond to cones in 3D space. The fiducial marker on the 3D printed marker allows tracking the geometry in 3D space. Intersecting the cone against the geometry yields gaze points on the object.

The printed object also includes a fiducial marker, which allows us to determine the rigid transformation of the object in space as described in Sec. 3.1.

As explained before, in view of inaccuracies, the pupil position $\bar{\mathbf{p}}$ describes a cone in 3D space. Consequently, we wish to identify the vertices on the object that intersect the cone and are visible. We could then potentially identify the vertex closest to the center of the cone as the desired gaze location. The approach is illustrated in Figure 4.

Let

$$\hat{\mathbf{p}}_i = \mathbf{Q}(\mathbf{R}\mathbf{v}_i + \mathbf{t}) \tag{10}$$

be the homogeneous pupil position $\mathbf{p}_i = (p_{i_1}, p_{i_2}, p_{i_3})^\mathsf{T}$ corresponding to vertex $\mathbf{v}_i$. Then we find the set of vertices

$$\Gamma_c(\bar{\mathbf{p}}) = \left\{ \mathbf{v}_i \in M \;\middle|\; \frac{\hat{\bar{\mathbf{p}}}^\mathsf{T}\mathbf{M}_Q\hat{\mathbf{p}}_i}{(\hat{\bar{\mathbf{p}}}^\mathsf{T}\mathbf{M}_Q\hat{\bar{\mathbf{p}}})^{1/2}(\hat{\mathbf{p}}_i^\mathsf{T}\mathbf{M}_Q\hat{\mathbf{p}}_i)^{1/2}} > \cos c \right\}; \tag{11}$$

that is, we are determining which vertices $\mathbf{v}_i$ on the object lie within the cone of angular size $c$ centered around the eye ray corresponding to $\bar{\mathbf{p}}$. From these vertices, we consider the one closest to the eye as the intersection point. This point can be determined efficiently solely using $p_{i_3}$. Note that since the metric $\mathbf{M}_Q$ has a natural relation to eye ray angle, we can choose $c$ based on the accuracy of our measurements.

### *4.1 Spatial partitioning tree*

For finely tessellated meshes, testing all vertices based on Eq. 11 above results in high computational costs. Spatial partitioning can be used to speed up the computation, by avoiding to test vertices that are far away from the cone. Through experimentation we have found sphere trees to outperform other common choices of spatial data structures (such as *k*d-trees, which appear as a natural choice) for the necessary intersection against cones.

Each fixation on the object is the intersection of the eye ray cone with the object surface, which is represented by a triangulated surface $\mathbf{M}$. Therefore, in the first step we perform an in-cone search to find all intersected vertices. This intersection result contains both front side and back side vertices. We are, however, only interested in visible vertices that are unoccluded with respect to the eye.

Popular space-partitioning structure for organizing 3D data are $K$-d trees, which divide space using splitting hyperplanes, and octrees, where each cell is divided into eight children of equal size. For our application, such axis-aligned space partitionings would require a cone-plane or cone-box intersection, which potentially incurs considerable computational costs. In order to avoid this, we build a space-partitioning data structure based on a sphere tree.

**Sphere tree construction** Our sphere tree is a binary tree whose construction proceeds top-down, recursively dividing the current sphere node into two child nodes. To determine the children of a node, we first apply principle component analysis and use the first principle vector, which corresponds to the largest eigen value of the covariance matrix, as the splitting direction. A partitioning hyperplane orthogonal to the splitting direction is then generated so that the elements in the node are subdivided into two sets of equal cardinality. Triangle faces intersecting with the splitting hyperplane are assigned to both sets. The child nodes are finally formed as the bounding spheres of the two sets and computed as proposed in [31].

We calculate the sphere-cone intersection following the method proposed in [30]. The problem is equivalent to checking whether the sphere center is inside an extended region, which is obtained by offsetting the cone boundary by the sphere radius. Note that the extended region differs from the extend cone, and its bottom is a sector of the sphere. For each intersected leaf node, we perform the following in-cone test to find the intersected vertices.

**In-cone test** A view cone is defined by an eye point $\mathbf{a}$ (i. e. the virtual eye position), a unit length view direction $\mathbf{r}$ , and opening angle $\delta$. The in-cone test allows us to determine if a given point $\mathbf{v}_i$ lies inside this cone. Given the matrix $\mathbf{M} \in \mathbb{R}^{4\times4}$

$$\mathbf{M} = \begin{pmatrix} \mathbf{S}, & -\mathbf{Sa} \\ -\mathbf{a}^T\mathbf{S}, & \mathbf{a}^T\mathbf{Sa} \end{pmatrix}, \tag{12}$$

where $\mathbf{S} = \mathbf{r}\mathbf{r}^T - \mathbf{d}^2\mathbf{I}$ with $\mathbf{d} = cos\delta$, the point $\mathbf{v}_i$ lies inside the cone only when $\hat{\mathbf{v}}^T\mathbf{M}\hat{\mathbf{v}} > 0$ where

$$\hat{\mathbf{v}} = \hat{\mathbf{v}}_i - \hat{\mathbf{a}} = \begin{pmatrix} \mathbf{v}_i \\ 1 \end{pmatrix} - \begin{pmatrix} \mathbf{a} \\ 1 \end{pmatrix}. \tag{13}$$

**Visibility test** The visibility of each intersected vertex is computed by intersecting the ray from eye point to the vertex with the triangle mesh. The vertex is visible if no other intersection is closer to the eye point. We use the Möller-Trumbore ray-triangle intersection algorithm [18] for triangles in intersected bounding spheres. In our implementation, the maximum tree depth is set to 11, which allows for fast traversal and real-time performance.

### *4.2 Implementation*

Our software implementation uses OpenCV [25], which was in particular employed to solve for the rigid transformations **R**,**t** as described in Sec. 3.1. We determine **Q** using Eq. 4 with the Ceres Solver [4]. The optimization is sensitive to the initial estimate, which can result in the optimization converging to a local minimum, yielding unsatisfactory results. To overcome this problem, we use a RANSAC approach for the initial estimate, with the error being calculated following Eq. 14 and 1000 iterations. The result of this procedure serves as input for the later optimization using the Ceres solver.

## 5 Experiments

In the following, we will report on preliminary experimental results that validate the accuracy of our setup for tracking 3D gaze points and that demonstrate that a small number of correspondences suffices for calibration. These results were obtained using two exploratory experiments with a small number of subjects ($n = 6$).

**Participants and apparatus** We recruited 6 unpaid participants (all male), all of which were students or staff from a university. Their age ranged from 26 to 39 years and all had normal or corrected-to-normal vision, based on self-reports. Four of them had previous experience with eye tracking.

The physical setup of our experiment is shown in Fig. 5. For measuring fixations we employed the Pupil eye tracker [17] and the software pipeline described in the previous sections.

### *5.1 Accuracy of calibration and gaze direction estimation*

In Sec. 3.2 we explained how the projective mapping **Q** from world camera coordinates to pupil positions can be determined by solving a linear least squares problem.

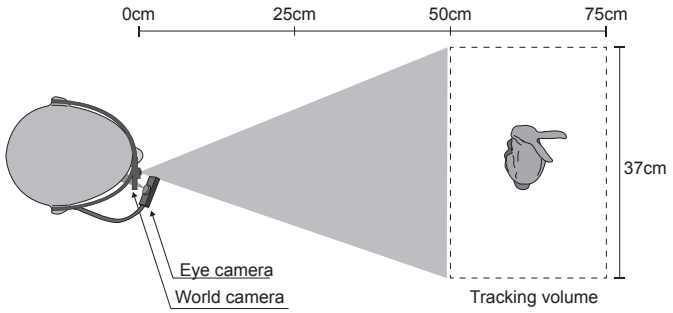


**Fig. 5** Physical setup used in our experiments.

As input to the problem one requires correspondences $\{(\mathbf{w}_i, \mathbf{p}_i)\}$ between world camera coordinates $\mathbf{w}_i$ and pupil positions $\mathbf{p}_i$. The correspondences have to be determined experimentally, and hence will be noisy. The accuracy with which $\mathbf{Q}$ is determined therefore depends on the number of correspondences that is used. In our first experiment we investigated how many correspondences are needed to obtain a robust estimate for $\mathbf{Q}$. The same data also allows us to determine the angular error of our setup.

**Procedure** We obtained correspondences $\{(\mathbf{w}_i, \mathbf{p}_i)\}$ by asking a subject to focus on the center of a single fiducial marker (size 4 cm $\times$ 4 cm) while it is presented at various locations in the desired view volume (see Fig. 1, third image). We have augmented the center of the marker with a red dot to make this task as unambiguous as possible. At each position of the marker, we estimate a single correspondence $(\mathbf{w}_i, \mathbf{p}_i)$ based on the estimation of the rigid transformation for the marker, cf. Sec. 3.1. For each participant, we recorded 100 correspondences $\{(\mathbf{w}_i, \mathbf{p}_i)\}$ for two different conditions, resulting in a total of 200 measurements per participant. In the first condition the head was fixed on a chin rest while in the second condition participants were only asked to keep facing towards the marker. For both conditions the marker was moved in a volume of 0.37 m (width) $\times$ 0.4 m (height) $\times$ 0.25 m (depth) at a distance of 0.75 m from the subject (see Fig. 5).

**Data processing** For each dataset we perform 10 trials of 2-fold cross validation and estimate the projection matrix using 7 to 49 point pairs. In each trial, the 100 correspondences are randomly divide into 2 bins of 50 point pairs each. One bin is used as training set and the other as testing set. Point pair correspondences from the training set are used to compute the projection matrix $\mathbf{Q}$ which is then employed to compute the error between the gaze direction given by the pupil position $\mathbf{p}_i$ and the true direction given by the marker center $\mathbf{w}_i$ for the points in the test data set. From Eq. 9 this error can be calculated as

$$\eta_i = \cos^{-1} \frac{\mathbf{p}_i^\top \mathbf{M}_Q \mathbf{Q} \mathbf{w}_i}{(\mathbf{p}_i^\top \mathbf{M}_Q \mathbf{p}_i)^{1/2} (\mathbf{w}_i^\top \mathbf{Q}^\top \mathbf{M}_Q \mathbf{Q} \mathbf{w}_i)^{1/2}}. \tag{14}$$

**Analysis and results** In order to analyze the influence of the number of calibration points as well as the usage of the chin rest on the estimation accuracy, we performed a repeated measures ANOVA ($\alpha = .05$) on the independent variable *Chin rest* with 2 levels (with, without) and *Calibration* with 43 levels (the corresponding number of calibration points, i.e., 7 to 49). The dependent variable was the calculated angular error in degree. We used 10 rounds of cross validation for our repeated measures, with each data point being the average angular error per round. This resulted in an overall of 860 data points per participant (2 *Chin reset* $\times$ 43 *Calibration* $\times$ 10 cross validation).

Results showed a main effect for *Calibration* ($F_{42,210} = 19.296$, $p < .001$). The difference between 20 points ($M = 0.75$, $SE = 0.02$) and 42, 44, 45, 46, 47 and 48 points (all $M = 0.71$, $SE = 0.02$) was significantly different, as well as 22 points ($M = 0.74$, $SE = 0.02$) compared to 45 points (all $p < .05$). No other combinations were statistically significantly different, arguably due to high standard deviation for lower number of calibration points. Mean values and standard errors are depicted in Figure 6.

When using 11 to 49 calibrations points, the angular error averages at around $0.73^\circ$ ($SD = 0.02$), which is within the range of human visual accuracy and goes in line with the specifications of the pupil eye tracker for 2D gaze estimation [17, 2]. The results furthermore demonstrate that even for a relatively low number of calibration points, comparable to the 9 points typically used for calibration for 2D gaze estimation [12, 17], our method is sufficiently accurate.

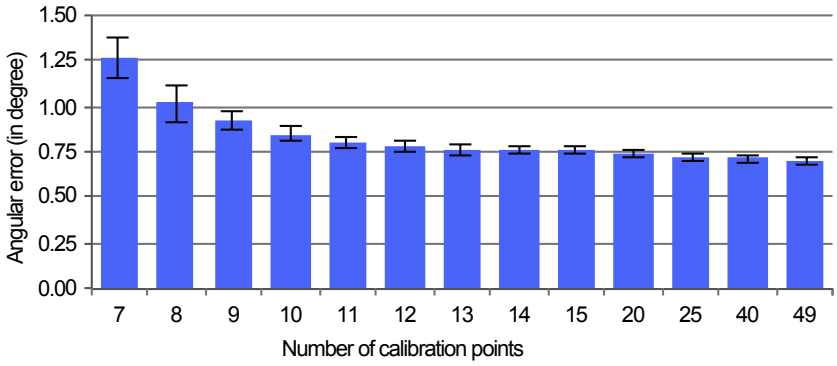


**Fig. 6** Mean values and standard errors for angular error as a function of the number of calibration points ranging from 7 to 49. No significant changes in angular error occur when using 11 or more calibration points.

No significant effect for *Chin rest* ($F_{1,5} = 0.408$, $p = .551$; with chin rest $M = 0.73$, $SE = 0.05$; without chin rest $M = 0.78$, $SE = 0.04$) was present, suggesting that the usage of a chin rest has negligible influence on the angular accuracy and our method is hence insensitive to minor head motion. This goes in line with the observation that light head motion has no effect on the relative orientation and

position of eye, eye camera, and world camera. It should be noted, however, that participants, although not explicitly instructed, were mostly trying to keep their head steady, most likely due to the general setup of the experiment. Giving participants the ability to move their head freely is an important feature for exploring objects in a natural, unconstrained manner. However, quantifying the effect of large scale motion on accuracy should be subject to further investigations.

## *5.2 Accuracy of 3D gaze position*

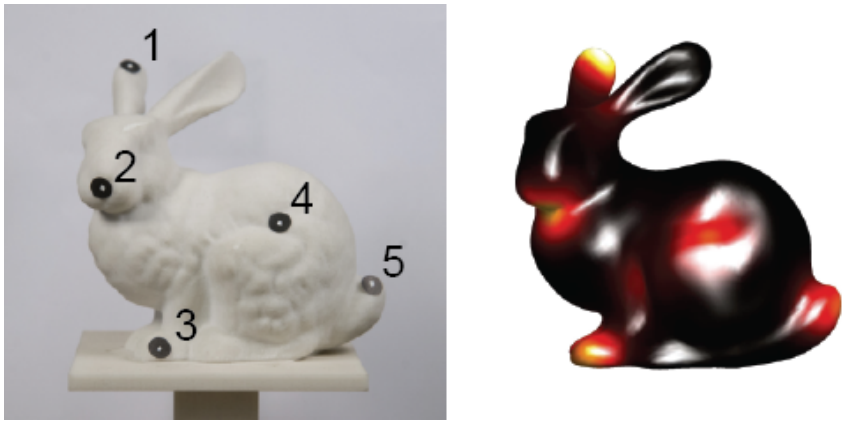


**Fig. 7** *Left*: physical bunny model with target markers (numbers indicate order); *right*: heat map of obtained gaze directions.

In our second experiment we explored the accuracy of our approach when viewing 3D stimuli. As model we employed the Stanford bunny and marked a set of pre-defined target points on the 3D printed bunny as shown in Fig. 7, left. After a calibration with 11 correspondences, as described in the last section, the test subjects were asked to focus on the individual targets (between 1 and 2 seconds). A heat map of the obtained gaze positions is shown in Fig. 7, right. Fixations are calculated based on Eq. 11 where the angular size $c$ is set to be $0.6^\circ$. Table 1 shows the angular error of each target in degrees as well as the depth error in mm.

Angular error depends mostly on the tracking setup. However, since the intersection computation with eye ray cones is restricted to points on the surface (vertices in our case), we get smaller angular errors on silhouettes.

Depth accuracy, on the other hand, depends on the slope of the geometry. In particular, at grazing angles, that is when the normal of the geometry is orthogonal or almost orthogonal to the viewing direction, it could become arbitrarily large. For the situations of interest to us where we have some control over the model, the normal is orthogonal or almost orthogonal to the viewing direction mainly only around the

silhouettes. Since we determine the point on the object that best corresponds to the gaze direction, we obtain accurate results also around silhouettes. This is reflected in the preliminary experimental results where we obtain an average depth error of 7.71 mm at a distance of 553.97 mm, which corresponds to a relative error of less than 2%, despite three of five targets being very close to a silhouette.

**Table 1** Errors of individual markers on bunny.

| Marker index | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Angular error (deg.) | 0.578 | 1.128 | 0.763 | 0.846 | 0.729 |
| Depth error (mm) | 7.998 | 8.441 | 10.686 | 3.036 | 8.381 |

## 6 Discussion

The proposed method for estimating fixations on 3D objects is simple yet accurate. It is enabled by:

- generating stimuli using digital manufacturing to obtain precisely known 3D geometry without restricting its shape;
- utilizing fiducial markers in a known relative position to the geometry to reliably determine its position relative to a subject's head;
- using a projective mapping to relate 3D positions to 2D pupil coordinates.

We experimentally verified our approach using two explorative user studies. The results demonstrate that 11 correspondences suffice to reliably calibrate the mapping from pupil coordinates to 3D gaze locations with an angular accuracy of 0.8 degree. This matches the accuracy of 2D gaze tracking. We achieve a depth accuracy of 7.7 mm at a distance of 550 mm, corresponding to a relative error of less than 1.5%.

With the popularization of 3D printing, our approach can be easily applied to a large variety of stimuli, and thus usage scenarios. At the same time, it is not restricted to 3D printed artifacts and can be employed as long as the geometry of an object is known, for example when manual measurement or 3D scanning has been performed. Our approach also generalizes to simultaneously tracking gaze with multiple objects, as long as the objects' position and orientation are unambiguously identified, e. g. by including fiducial markers. The tracking accuracy in such situations will be subject to future investigation.

We developed our approach for 3D gaze tracking to analyze viewing behavior for genuine 3D stimuli, and to explore what differences to 2D stimuli exist. Our approach in particular enables researchers to study visual saliency on physical objects without scarifying accuracy. Given the substantial amount of work on saliency and related questions that employed 2D stimuli for studying 3D objects, we believe this to be a worthwhile research question that deserves further attention.

We believe 3D gaze tracking will be a valuable tool for research in computer science, cognitive science, and other disciplines. The fast and simple calibration procedure (comparable to typical 2D calibration) that is provided by our approach enables researcher to extend their data collection without significantly changing their current workflow.

**Acknowledgements** This work has been partially supported by the ERC through grant ERC-2010-StG 259550 (XSHAPE). We thank Felix Haase for his valuable support in performing the experiments and Marianne Maertens for discussions on the experimental setup.

## References

1. Abbott, W. W., Faisal, A. A.: Ultra-low-cost 3D gaze estimation: an intuitive high information throughput compliment to direct brain-machine interfaces. Journal of Neural Engineering, **9**, 1–11 (2012)
2. Barz M., Bulling A., Daiber F.: Computational Modelling and Prediction of Gaze Estimation Error for Head-mounted Eye Trackers. German Research Center for Artificial Intelligence (DFKI) Research Reports, pp. 10 (2015) `https://perceptual.mpi-inf.mpg.de/files/2015/01/gazequality.pdf` Cited 21 Dec 2014
3. Bruce, N., Tsotsos, J.: Saliency Based on Information Maximization. Advances in Neural Information Processing Systems, 155–162 (2006)
4. Agarwal S., Mierle K., others: Ceres Solver. `http://ceres-solver.org` Cited 21 Dec 2015
5. Cournia, N. Smith, J. D., Duchowski, A. T.: Gaze-vs. hand-based pointing in virtual environments. CHI'03 extended abstracts on Human factors in computing systems, 772–773. ACM (2003)
6. Duchowski, A. T., Pelfrey, B., House, D. H., Wang, R.: Measuring gaze depth with an eye tracker during stereoscopic display. Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization, pp. 15. ACM (2011)
7. Essig, K., Pomplun, M., Ritter, H.: A neural network for 3D gaze recording with binocular eye trackers. International Journal of Parallel, Emergent and Distributed Systems, **21**, 79–95 (2006)
8. Häkkinen, J., Kawai, T., Takatalo, J., Mitsuya, R., Nyman, G.: What do people look at when they watch stereoscopic movies? In: Woods, A. J., Holliman, N. S., Dodgson, N. A. (eds.) Stereoscopic Displays and Applications XXI (2010)
9. Hanhart, P., Ebrahimi, T.: EYEC3D: 3D video eye tracking dataset. 2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX), pp. 55–56. IEEE (2014)
10. Heng, L., Li, B., Pollefeys, M.: Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on, pp. 1793–1800. IEEE (2013)
11. Hennessey, C., Lawrence, P.: Noncontact Binocular Eye-Gaze Tracking for Point-of-Gaze Estimation in Three Dimensions. IEEE Transactions on Biomedical Engineering, **56**, 790–799 (2009)
12. Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., van de Weijer, J.: Eye Tracking: A Comprehensive Guide to Methods and Measures. Oxford University Press (2011)
13. Howard, I. P.: Preceiving in Depth. Oxford University Press (2012)
14. Huynh-Thu, Q., Schiatti, L.: Examination of 3D visual attention in stereoscopic video content. In: Rogowitz, B. E., Pappas, T. N. (eds.) IS&T/SPIE Electronic Imaging, pp. 78650J–78650J. International Society for Optics and Photonics (2011)

15. Jansen, L., Onat, S., König, P.: Influence of disparity on fixation and saccades in free viewing of natural scenes. Journal of Vision, **9**, 1–19 (2009)
16. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. International Conference on Computer Vision, pp. 2106–2113. IEEE (2009)
17. Kassner, M., Patera, W., Bulling, A.: Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct., pp. 1151–1160. ACM (2014)
18. Kensler A., Shirley P.: Optimizing ray-triangle intersection via automated search. Interactive Ray Tracing, IEEE Symposium on, pp. 33–38. IEEE (2006)
19. Ki, J., Kwon, YM.: 3D Gaze Estimation and Interaction. 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, pp. 373–376, IEEE (2008)
20. Koenderink, J. J.: Pictorial relief. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, **356**, 1071–1086 (1998)
21. Lang, C., Nguyen, T. V., Katti, H., Yadati, K., Kankanhalli, M., Yan, S.: Depth Matters: Influence of Depth Cues on Visual Saliency. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Computer Vision ECCV 2012, pp. 101–115. Springer (2012)
22. Lee, J. W., Cho, C. W., Shin, K. Y., Lee, E. C., Park, K. R.: 3D gaze tracking method using Purkinje images on eye optical model and pupil. Optics and Lasers in Engineering, **50**, 736–751 (2012)
23. Maggia, C., Guyader, N., Guérin-Dugué, A.: Using natural versus artificial stimuli to perform calibration for 3D gaze tracking. In: Rogowitz, B. E., Pappas, T. N., de Ridder, H. (eds.) Human Vision and Electronic Imaging XVIII (2013)
24. Mathe, S., Sminchisescu, C.: Dynamic Eye Movement Datasets and Learnt Saliency Models for Visual Action Recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C (eds.) Computer Vision ECCV 2012, pp. 842–856. Springer (2012)
25. Bradski, G.: OpenCV. Dr. Dobb's Journal of Software Tools (2000)
26. Pfeiffer, T., Latoschik, M. E., Wachsmuth, I.: Evaluation of Binocular Eye Trackers and Algorithms for 3D Gaze Interaction in Virtual Reality Environments. Journal of Virtual Reality and Broadcasting, **5** (2008)
27. Pfeiffer, T., Renner, P.: Eyesee3d: A low-cost approach for analyzing mobile 3d eye tracking data using computer vision and augmented reality technology. Proceedings of the Symposium on Eye Tracking Research and Applications, pp. 369-376. ACM (2014)
28. Ramanathan, S., Katti, H., Sebe, N., Kankanhalli, M., Chua, T.-S.: An Eye Fixation Database for Saliency Detection in Images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision ECCV 2010, pp. 30–43. Springer (2010)
29. Ramasamy, C., House, D. H., Duchowski, A. T., Daugherty, B.: Using eye tracking to analyze stereoscopic filmmaking. Posters on SIGGRAPH '09, pp. 1. ACM (2009)
30. Schneider P.J., Eberly, D.: Geometric Tools for Computer Graphics. Elsevier Science Inc., New York, USA (2002)
31. Ritter, J.: An Efficient Bounding Sphere. In: Glassner, A.S. (eds.) Graphics Gems, pp. 301-303. Academic Press, Boston, MA (1990)
32. Stellmach, S., Nacke, L., Dachselt, R.: 3d attentional maps: aggregated gaze visualizations in three-dimensional virtual environments. Proceedings of the international conference on advanced visual interfaces, pp. 345–348. ACM (2010)
33. Toet, A.: Computational versus psychophysical bottom-up image saliency: a comparative evaluation study. IEEE transactions on pattern analysis and machine intelligence, **33**, 2131–46 (2011)

# Index