Towards Mitigating the Eye Gaze Tracking Uncertainty in Virtual Reality*

Konstantin Ryabinin[1,2][0000−0002−8353−7641] and Svetlana Chuprina[2][0000−0002−2103−3771]

[1] Saint Petersburg State University, 7/9 Universitetskaya Emb., Saint Petersburg 199034, Russia
[2] Perm State University, 15 Bukireva Str., Perm 614068, Russia
kostya.ryabinin@gmail.com, chuprinas@inbox.ru

Abstract. We propose a novel algorithm to evaluate and mitigate the uncertainty of data reported by eye gaze tracking devices embedded in virtual reality head-mounted displays. Our algorithm first is calibrated by leveraging unit quaternions to encode angular differences between reported and ground-truth gaze directions, then interpolates these quaternions for each gaze sample, and finally corrects gaze directions by rotating them using interpolated quaternions. The real part of the interpolated quaternion is used as the certainty factor for the corresponding gaze direction sample. The proposed algorithm is implemented in the VRSciVi Workbench within the ontology-driven SciVi visual analytics platform and can be used to improve the eye gaze tracking quality in different virtual reality applications including the ones for Digital Humanities research. The tests of the proposed algorithm revealed its capability of increasing eye tracking accuracy by 25% and precision by 32% compared with the raw output of the Tobii tracker embedded in the Vive Pro Eye head-mounted display. In addition, the certainty factors calculated help to acknowledge the quality of reported gaze directions in the subsequent data analysis stages. Due to the ontology-driven software generation, the proposed approach enables high-level adaptation to the specifics of the experiments in virtual reality.

Keywords: Eye Tracking · Virtual Reality · Uncertainty Mitigation · Ontology-Driven Software Generation · Quaternion-Based Model.

1 Introduction

Modern Virtual Reality (VR) head-mounted displays (HMDs) enable not only the user's head orientation and position tracking, but also eye gaze tracking capabilities. These new capabilities were included in HMDs not long ago and have a lot of interesting applications, which make eye tracking a very promising extension for traditional VR technologies. First, eye tracking enables the so-called

* This study is supported by the research grant No. ID75288744 from Saint Petersburg State University.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

foveated rendering, which is an optimization technique that reduces rendering workload by lowering the image quality in the user's peripheral vision [19]. Second, eye tracking provides new ways of interaction with VR scene objects [15]. For example, a "hands-free" gaze-based selection of objects can be implemented, which may be useful for instantly getting the context information about the scene contents. Alternatively, eye tracking in multiplayer VR games/meetings allows making the users' avatars more vivid by animating their eyes in accordance with actual users' gaze. Last but not least, eye tracking has opened up novel ways to study the user's behavior. In simulators, eye tracking can be used for estimating the correctness of the user's behavior by checking if the user looks in the expected direction or not. For example, in the car driving simulator, the user can be alerted when looking off the road [20]. In special test stands, eye tracking can help to study the information perception mechanisms by providing data about the order of objects the user focuses on [14] because the direction of eye gaze provides a strong cue to the person's intentions and future actions [6].

While traditional eye tracking with remote stationary tracking devices is a well-defined methodology [26,9,14], VR brings some new challenges dealing with the accuracy and precision issues [27,30,17]. In eye tracking, accuracy is defined as "average angular error between the measured and the actual location of the intended fixation target", and precision "is the spread of the gaze positions when a user is fixating on a known location in space" [27]. Modern VR HMDs still have quite a low angular resolution, sampling rate, and peak signal-noise ratio of embedded eye tracking sensors compared with remote stationary eye trackers [30,17]. The tasks like foveated rendering require just an approximate gaze direction and their performance is not much affected by tracking inaccuracy. In contrast, for studying human behavior, eye tracking accuracy and precision are crucial. Inaccurate/noisy raw tracking data lead to an increase in the uncertainty of the analysis stage results and thereby can ruin the entire study.

At the same time, VR as a set of immersive visualization technologies provides new potential to organize advanced experiments for human behavior study within a field of Digital Humanities research [29,28,12]. Proper handling of the eye gaze tracking uncertainty is badly needed to make the results of those experiments reliable.

In this work, we propose a unified approach to evaluate and mitigate the eye gaze tracking data uncertainty in VR. The proposed approach is based on our experience of using Vive Pro Eye HMD with embedded Tobii eye tracking sensor as an immersion and measurement hardware within the Digital Humanities research. The uncertainty mitigation algorithm is an essential part of the so-called VRSciVi Workbench that is a set of tools within the SciVi visual analytics platform[3] aimed to automate the conducting of the experiments in VR and handling the results of these experiments by means of scientific visualization and visual analytics tools.

[3] https://scivi.tools/

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

2 Key Contributions

We propose ontology-driven tools to automatically generate the software for conducting eye-tracking-based experiments within the immersive VR environments mitigating the uncertainty of data reported by embedded eye trackers. The following hitpoints of the conducted research can be highlighted:

1. Novel quaternion-based model of the eye gaze tracking uncertainty.
2. Novel algorithm to mitigate the eye gaze tracking uncertainty at runtime that enables to increase the eye tracking accuracy and precision by ca. 30%.
3. Ontology-driven high-level software development tools to generate the interface for parametrization of the proposed algorithm and to adapt it for integration in a software to meet the specifics of different VR-based experiments.

3 Related Work

Eye gaze tracking is a research methodology with more than a hundred years of history that is nowadays accessible and intensively used in a wide range of scientific domains [9]. The related hardware has progressed over several decades and its evolution is methodically outlined in the review by Shehu et al. [26]. The particular protocols of conducting experiments and algorithms for processing the collected data are elaborated and well-documented [9,14]. The corresponding software provides implementations of these algorithms along with the needed visualization and analytics techniques [25,3]. But the convergence of eye gaze tracking with VR brings new challenges in both of these fields demanding the development of new tools for handling gaze tracks in close relation to the VR scene objects [10] and immersion features [5,16].

Huge problems of eye tracking in VR are low resolution, accuracy, and precision of the trackers embedded in HMDs, which hinder the use of these trackers in special cases like medical or humanities research. The temporal resolution of embedded eye trackers is normally capped at the VR scene refresh rate, which is currently 90 Hz (corresponding to the refresh rate of the modern VR HMDs) [30,17]. In contrast, the modern stationary eye trackers operate on up to 2000 Hz [1]. As found by D. Lohr et al. on the example of Vive Pro Eye HMD, embedded trackers can have internal non-toggleable low-pass filters rejecting fast saccades even within the available sampling rate [17]. That means, it is fundamentally impossible to use such devices to study phenomena like ocular microtremor, etc.

A. Sipatchin et al. conducted a very elaborate case study of Vive Pro Eye's usability for clinical ophthalmology and concluded that although this device has "limitations of the eye-tracker capabilities as a perimetry assessment tool", it has a "good potential usage as a ready-to-go online assistance tool for visual loss" and the "upcoming VR headsets with embedded eye-tracking" can be introduced "into patients' homes for low-vision clinical usability" [27]. Another important contribution of this research group is the accuracy and precision measurement of the Vive Pro Eye's embedded Tobii eye tracker. They found out

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

that the accuracy deteriorates dramatically as the gaze location moves away from the display center and the spatial accuracy of the tracker is more than 3 times lower than the manufacturer claims. The same phenomenon was shown by K. Binaee et al. for "binocular eye tracker, built into the Oculus Rift DK2 HMD" [2]. This suggests that a lot of embedded eye trackers suffer from this problem. Despite the obvious limitations, Vive Pro Eye HMD is recognized as an apparatus with high ergonomic characteristics suitable for the eye-tracking-based study of human behavior and information perception [17,11,18]. According to [29,28,12,17,11,18], it can be considered as a consensus that VR research on improving the HMD user experience and embedded eye trackers quality is worth continuing because VR-based technologies provide great possibilities to conduct human-centered studies.

Along with the hardware improvements constantly provided by manufacturers, software improvements are possible as well by applying post-processing filters to the raw data of embedded eye trackers. For example, K. Binaee et al. propose a method for embedded eye tracker post-hoc calibration based on homography calculated via a random sample consensus and a dynamic singular value decomposition [2]. This method allows to increase the eye tracking accuracy up to 5 times compared with the built-in HMD calibration technique, but the disadvantage is the post-hoc nature of the method which means the calculations are off-line.

S. Tripathi et al. propose a self-calibrating eye gaze tracking scheme based on Gaussian Process Regression models [31]. This scheme avoids the explicit calibration of tracking devices while maintaining competitive accuracy. The disadvantage is that this scheme is only applicable for scenes with moving objects and cannot be used for static scenes.

A. Hassoumi et al. propose improving eye tracking accuracy by using a so-called symbolic regression, which seeks an optimal model of transforming the eye pupil coordinates to the gaze location within "a set of different types of predefined functions and their combinations" using a genetic algorithm [8]. This approach allows a 30% accuracy increase compared with the standard calibration procedures. This method was used for monocular eye tracking systems, but modern VR HMDs use binocular systems, so the method cannot be applied to VR directly.

H. Drewes et al. propose a calibration method based on smooth pursuit eye movement that is 9 times faster than the traditional calibration process (which uses 9 stationary points aligned by the regular grid as fiducial gaze targets with known coordinates), but the resulting accuracy is slightly lower [7].

A. Jogeshwar et al. designed the cone model "to acknowledge and incorporate the uncertainty" of eye gaze detected by embedded trackers in VR [12]. But the way to evaluate and mitigate this uncertainty is still an open question, especially for cases with small objects of interest. Our work contributes to solving this problem aiming to mitigate the eye gaze tracking uncertainty within Digital Humanities research.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

4 Facing the Eye Gaze Tracking Uncertainty in Virtual Reality

We faced the problem of uncertain eye gaze tracking data during the work on the research project "Text processing in L1 and L2: Experimental study with eye-tracking, visual analytics and virtual reality technologies"[4] (supported by the research grant No. ID75288744 from Saint Petersburg State University). One of the goals of this project is to study the reading process of humans within a VR environment using eye gaze tracking as a measurement technique.

The experiments are conducted using a specific VR setup including Vive Pro Eye HMD connected to the VR rendering station based on the AMD Ryzen 9 CPU and NVidia Titan RTX GPU. The rendering is performed by Unreal Engine 4. The eye tracking data are collected using the SRanipal SDK[5] plugin for Unreal Engine. The experiment control and the data analysis are performed by the VRSciVi Workbench within the SciVi visual analytics platform [22].

Considering the case of reading a relatively large poster ($40° \times 26°$ of vision area in size, see Fig. 1), we found out that the data retrieved from the embedded eye tracker is far from being suitable to trace the reading process on the level of individual letters/syllables or at least of individual words (the width of each letter is approx. 0.63°). Fig. 1 demonstrates the virtual scene (rendered by Unreal Engine 4) as viewed by the informant with the 183 words long text displayed on the wall poster. Fig. 2 shows the map of the gaze fixations measured by the Vive Pro Eye HMD embedded Tobii eye tracker. Circle size depicts fixation duration. Fixations are detected according to the method suggested in [16]; dwell time threshold is set to 250 ms, angular threshold is set to 1°.

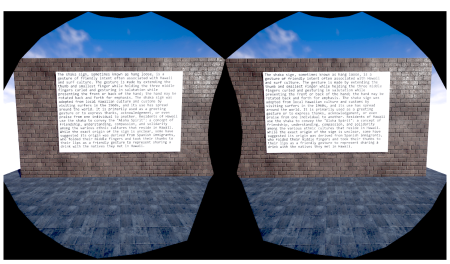

Fig. 1: The virtual scene as viewed by the informant

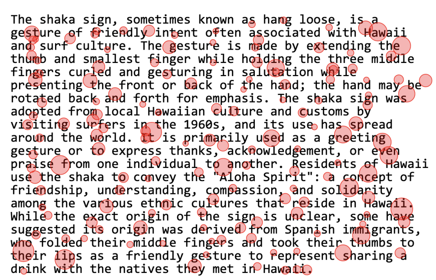

Fig. 2: The map of obtained gaze fixations

The fixations heatmap in Fig. 2 reveals obvious distortions: fixations miss the actual words and seem to be shifted. Such distortions hinder any subsequent analysis of the obtained data. As stated by A. Jogeshwar et al., to get reliable results of experiments the gaze uncertainty should be properly acknowledged [12].

[4] L1 and L2 stand for the native and foreign languages respectively

[5] https://developer-express.vive.com/resources/vive-sense/eye-and-facial-tracking-sdk/

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

As mentioned above, the VRSciVi Workbench is the central software element of our experimental setup. It controls the VR scene content, manages data extraction, transformation, and loading processes, and provides tools to visualize and analyze the collected data. This platform is driven by ontologies, which enable its extensibility and adaptability [23]. So, we have introduced the new components to SciVi for handling the eye gaze tracking uncertainty without changing the source code of any other components. These components are publicly available in the SciVi open-source repository and can be used in any SciVi-based VR project (see Section 6).

5 Evaluating and Mitigating the Uncertainty of the Vive Pro Eye Sensor Data

5.1 Evaluating the Uncertainty

To evaluate the eye gaze tracking uncertainty of the Vive Pro Eye sensor, we use a 5×5 points pattern similar to the one described in [27]. We display the pattern points on the billboard in the virtual scene (a white poster on the wall, see Fig. 1). Points are shown one by one, each for 2000 ms, and the informant is asked to stare at them. For each point, the first 500 ms of gaze data are discarded as suggested in [7] to trim the initial saccade. The gaze locations are measured and averaged during the subsequent 1500 ms. The result of the comparison of ground-truth and reported gaze locations based on 10 trials is demonstrated in Fig. 3. The blue points depict ground-truth gaze locations (the points of the pattern the persons were looking at) and the red ellipses bound the averaged gaze locations reported by the Vive Pro Eye sensor individually calibrated for each person using a built-in 5-points calibration procedure. It must be noted that the gaze locations are represented in the 2D texture space of the billboard the pattern is shown on. The reported gaze coordinates are calculated by obtaining the hit points of the gaze ray with this billboard and transforming these coordinates from the virtual scene coordinates to the billboard texture space. The gaze ray is reported by the SRanipal SDK plugin for Unreal Engine, which is the default (and the only official) way to interact with the embedded Tobii eye tracker of Vive Pro Eye HMD.

Fig. 3 clearly shows the non-uniform nature of eye gaze tracking uncertainty, which aligns with the results reported in [27,2]. We confirm the accuracy loss in the display periphery: the smallest angular error of 0.02° is located near the center and the biggest one of 2.4° is located near the border of the billboard; the average angular error is 0.77° and the standard deviation is 0.43°. The baseline accuracy in our case is about 1.7 times higher than reported in [27] because according to the setup of our experiment we are inspecting a narrower field of view. Moreover, the billboard is not tied to the user's head, so the user can slightly rotate their head reducing the angular distance between the target point and the vision area center. Still, this accuracy does not suit our needs related to the study of the reading process in VR because individual letters of the texts

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

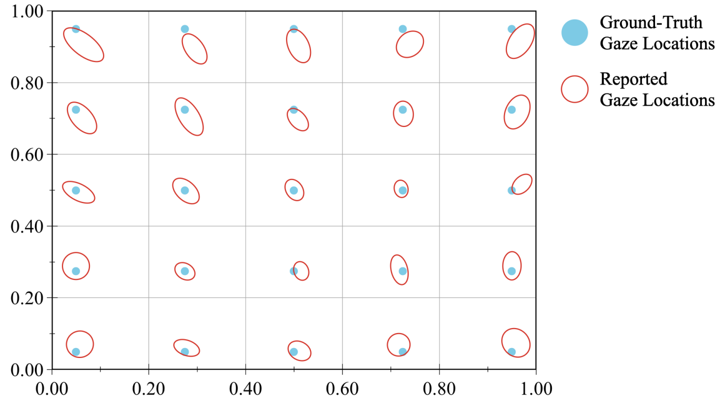

Fig. 3: Comparison of ground-truth and reported gaze locations

considered are ca. 0.63° in size. Therefore the accuracy should be improved by proper accounting of the gaze tracking uncertainty.

5.2 VR HMD Frame of Reference

Fig. 4 shows the frame of reference used in our calculations by uncertainty evaluation and mitigation. The point O and the vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ build up the left-handed coordinate system $\mathcal{H}$ tied to the user's head. These elements are constructed by Unreal Engine based on the HMD positioning system and represented in the global coordinate system $\mathcal{G}$ of the virtual scene. The $\boldsymbol{r}$ vector represents gaze direction represented in $\mathcal{H}$ as reported by the SRanipal SDK plugin and the vector $\boldsymbol{g}$ represents true gaze direction. The angle ϕ determines how far the user looks to the side from their forward direction according to the reported by SRanipal SDK, and the angle ψ represents the angular difference between reported and true gaze directions.

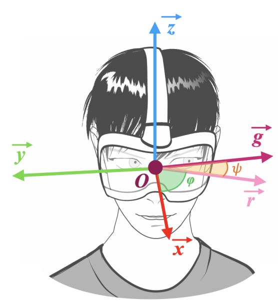

Fig. 4: VR HMD frame of reference

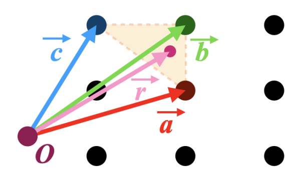

Fig. 5: Basis for reported gaze vector

5.3 Quaternion-Based Uncertainty Representation and Mitigation

To represent eye gaze tracking uncertainty we propose using a quaternion-based model [13]. Having a ground-truth gaze unit vector $\boldsymbol{g}$ and reported gaze unit vector $\boldsymbol{r}$, we can encode the tracking error as a quaternion q that represents the shortest arc rotation from $\boldsymbol{g}$ to $\boldsymbol{r}$:

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

$$q = \begin{cases} \{\boldsymbol{r}\cdot\boldsymbol{g}+1, \boldsymbol{r}\times\boldsymbol{g}\}, & \boldsymbol{r}\cdot\boldsymbol{g}\neq -1 \\ \{0, \quad \boldsymbol{u}\}, & \boldsymbol{u}\cdot\boldsymbol{r}=0,\ \boldsymbol{r}\cdot\boldsymbol{g}=-1 \end{cases}. \tag{1}$$

This quaternion should be normalized to be a versor (unit quaternion):

$$q_{\boldsymbol{r}\to\boldsymbol{g}} = \frac{q}{|q|}. \tag{2}$$

For this versor,

$$\boldsymbol{g} = q_{\boldsymbol{r}\to\boldsymbol{g}}\ \boldsymbol{r}\ q^*_{\boldsymbol{r}\to\boldsymbol{g}}, \tag{3}$$

where $q^*_{\boldsymbol{r}\to\boldsymbol{g}}$ is a complex conjugate of $q_{\boldsymbol{r}\to\boldsymbol{g}}$, $q^*_{\boldsymbol{r}\to\boldsymbol{g}} = \{\mathrm{Re}(q_{\boldsymbol{r}\to\boldsymbol{g}}), -\mathrm{Im}(q_{\boldsymbol{r}\to\boldsymbol{g}})\}$.

The real part of $q_{\boldsymbol{r}\to\boldsymbol{g}}$ represents a cosine of half angle between $\boldsymbol{g}$ and $\boldsymbol{r}$, which means $\mathrm{Re}(q_{\boldsymbol{r}\to\boldsymbol{g}}) = 1$ if $\boldsymbol{r}=\boldsymbol{g}$ and $\mathrm{Re}(q_{\boldsymbol{r}\to\boldsymbol{g}}) = 0$ if $\boldsymbol{r}=-\boldsymbol{g}$. In this regard, we propose to interpret the real part of $q_{\boldsymbol{r}\to\boldsymbol{g}}$ as a certainty factor (CF) of reported gaze direction:

$$\mathrm{CF}(\boldsymbol{r}) = \mathrm{Re}(q_{\boldsymbol{r}\to\boldsymbol{g}}). \tag{4}$$

The predicted angular error $\psi_{\boldsymbol{r}}$ of $\boldsymbol{r}$ can be extracted from CF as

$$\psi_{\boldsymbol{r}} = 2\arccos\left(\mathrm{CF}(\boldsymbol{r})\right). \tag{5}$$

Consequently, if we could find a corresponding versor $q_{\boldsymbol{r}\to\boldsymbol{g}}$ for an arbitrary reported gaze vector $\boldsymbol{r}$, we can then evaluate, how certain is the reported gaze direction using the formula (4), predict its angular error using the formula (5) and even correct the uncertain direction to the true one using the formula (3).

We propose tackling this problem by spatial interpolation of a discrete set of versors obtained during the custom calibration process. The general description of the calibration algorithm is as follows:

1. Place the white billboard in the VR scene, tied to the user's head. The size of this billboard is a matter of experimenting, for now we end up with $56° \times 37°$ This billboard will be a canvas for displaying target points. Since it is tied to the user's head, the user has to only rotate the eyes to look at the points, and not the head, which allows accounting the eye gaze direction only.
2. Choose the calibration pattern. This is still a matter of experimenting, but to start with, we use a traditional 3×3 regular grid of points, which center matches the center of the billboard.
3. For each point $P_i = \{x_i, y_i, z_i\}$ (coordinates are represented in the $\mathcal{H}$ reference frame mentioned in Section 5.2) from the pattern, $i = \overline{1,n}$, $n = 9$:
   3.1. Display P_i as a filled circle of radius $0.38°$ related to the vision area.
   3.2. During 1000 ms:
      3.2.1. Decrease the circle radius down to $0.18°$. As stated in [21], the size reduction helps the user to concentrate on the point's center.
      3.2.2. Discard the first 500 ms of gaze data to trim the initial saccade [7].
      3.2.3. Use the subsequent data to calculate the $\boldsymbol{r}_i$ vector averaging the reported gaze vectors.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

3.3. Calculate the ground-truth gaze vector as $\boldsymbol{g}_i = -P_i/|P_i|$. This formula is valid because P_i is represented in $\mathcal{H}$.
3.4. Calculate the versor $q_{\boldsymbol{r}_i \to \boldsymbol{g}_i}$ using the formulas (1) and (2). Let us denote this versor as q_i for brevity.
3.5. Store the calibration tuple $\langle \boldsymbol{r}_i, q_i \rangle$ for future reference.

After this custom calibration procedure, each time the gaze direction $\boldsymbol{r}$ is reported and its uncertainty is evaluated and mitigated using the following algorithm:

1. For the vector $\boldsymbol{r}$, obtain the following 3 reference vectors (see Fig. 5):
   $\boldsymbol{a} = \boldsymbol{r}_{n/2+1}$,
   $\boldsymbol{b} = \boldsymbol{r}_j | \boldsymbol{r} \cdot \boldsymbol{r_j} \to \max,\ j = \overline{1,n},\ j \neq n/2+1$,
   $\boldsymbol{c} = \boldsymbol{r}_k | \boldsymbol{r} \cdot \boldsymbol{r}_k \to \max,\ k = \overline{1,n},\ k \neq n/2+1,\ k \neq j$.
2. Let the matrix M be composed from the coordinates of $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ represented in the basis of $\mathcal{H}$:
   $M = \begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ a_z & b_z & c_z \end{pmatrix}$.
3. If $|M| = 0$ (that means, $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are linearly dependent), take the versor associated with the nearest vector among the calibration tuples:
   $q_{\boldsymbol{r} \to \boldsymbol{g}} = q_l | \boldsymbol{r} \cdot \boldsymbol{r}_l \to \max,\ l = \overline{1,n}$.
4. If $|M| \neq 0$,
   4.1. Find the coordinates $\{t_a, t_b, t_c\}$ of $\boldsymbol{r}$ in the basis $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$:
   $(t_a\ t_b\ t_c)^\top = M^{-1}(r_x\ r_y\ r_z)^\top$.
   4.2. Use these coordinates to calculate the desired versor interpolating the versors from calibration tuples, which correspond to the vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$: $q_{\boldsymbol{r} \to \boldsymbol{g}} = t_a q_a + t_b q_b + t_c q_c = t_a q_{n/2+1} + t_b q_j + t_c q_k$.
5. Calculate the gaze CF by the formula (4) and the corrected gaze vector $\boldsymbol{g}$ by the formula (3).

The results of applying the described algorithms are discussed in Section 7.

6 Implementation of Uncertainty Mitigation Algorithm in VRSciVi

VRSciVi Workbench within the SciVi platform is suited for controlling immersive VR environments. VRSciVi has the client-server architecture. VRSciVi Server is responsible for displaying the VR scene using Unreal Engine 4 as a graphics rendering system. VRSciVi Client is a set of SciVi platform plugins to build and control virtual scenes, as well as to collect and analyze related data about human activities within VR. The communication between Client and Server relies on the WebSocket protocol, which is detailed in [22].

The algorithm of uncertainty mitigation is implemented within VRSciVi Server and controlled by VRSciVi Client. The main plugin provided by VRSciVi Client is currently a so-called VRBoard, which enables placing different visual

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

stimuli on the billboard located in the virtual world (see. Fig. 1). VRBoard provides needed settings for the uncertainty mitigation algorithm and a command to start the custom calibration procedure. Like any other SciVi plugin, VRBoard is specified by a light-weight application ontology that is used for four main purposes. First, it documents the plugin. Second, it drives the automatic generation of a graphical user interface code for the plugin to allow the users to set up the needed parameters. Third, it drives the automatic generation of execution code to properly run the plugin in the given software environment. Fourth, it allows SciVi to organize proper communication of this plugin with the others, taking into account the computing nodes the plugins run on. Thanks to underlying ontology and built-in reasoning mechanism, all of these goals are achieved in a uniform way. The fragment of this ontology is demonstrated in Fig. 6.

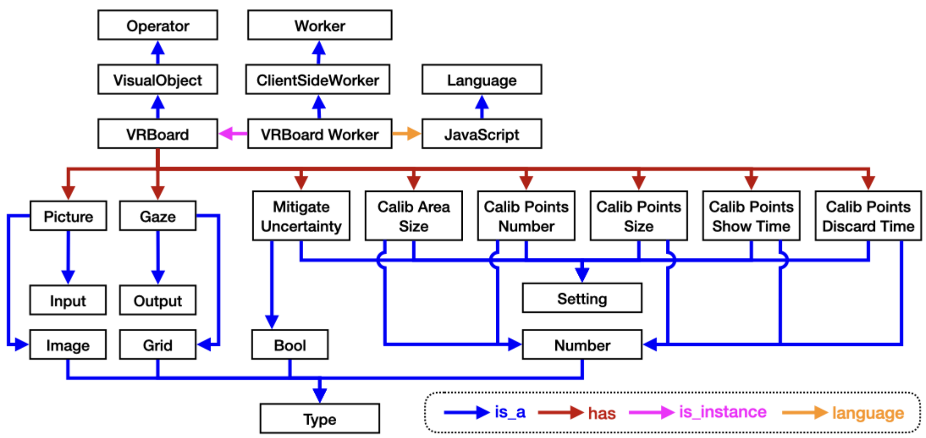

Fig. 6: Fragment of VRBoard ontology used in VRSciVi

Ontologies are designed within the ONTOLIS visual editor and have a proprietary JSON-based format .ONT [4]. As a rule, we use only a restricted set of basic relationship types to maintain the unified ontology model to specify different application ontologies for solving different tasks and reusing the built-in SciVi reasoning mechanism. This basic relationship set enables to improve the efficiency of code generation [24].

As can be seen in Fig. 6, VRBoard has the input “Picture” of the “Image” type that denotes a visual stimulus to be placed in VR and provides “Gaze” of the “Grid” type as an output. Also, there are several numerical settings related to the calibration (“Calib Area Size” – the angular size of the calibration grid, “Calib Points Number” – the number of target points in the calibration pattern, “Calib Point Size” – the angular size of the target point, “Calib Point Show Time” – the time to show each point, “Calib Point Discard Time” – the time to discard for trimming the initial saccade). The settings mentioned above help us in experimenting to find the optimal calibration strategy.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

The “Mitigate Uncertainty” Boolean flag toggles the error correction. If it is set to True, reported gaze vectors are rotated by the calculated versors, otherwise just CF values are calculated and transmitted along with unchanged reported gaze vectors. “VRBoard Worker” provides a client-side JavaScript-implementation of VRBoard. If needed, this implementation makes the VRBoard plugin along with the uncertainty mitigation algorithm available in any software generated by SciVi.

Like the entire SciVi platform, VRSciVi Workbench is an open-source project publicly available on the Web: `https://scivi.tools/vrscivi`.

7 Discussion

To estimate the quality of uncertainty mitigation, we run the accuracy assessment described in Section 5.1 considering not only the reported gaze vectors, but also the corrected ones. Fig. 7 sums up the results of 10 trials. In each trial, first, the built-in calibration was performed, then the custom calibration, and after that, the gaze data of sequential looking at 25 target points (aligned by 5 grid) were collected. Red ellipses bound reported gaze targets (just like in Fig. 3) and green ellipses bound corrected gaze targets. Blue points denote ground-truth target locations. Each sample's background highlights whether the correction algorithm improved both accuracy and precision (green), improved either accuracy or precision (yellow), impaired accuracy and precision (red).

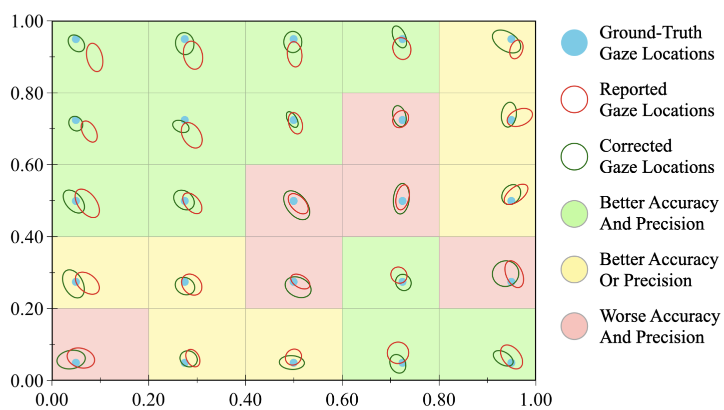

Fig. 7: Comparison of ground-truth, reported, and corrected gaze locations

The comparison of accuracy and precision of reported and corrected gaze data collected in this experiment is given in Table 1.

Table 1: Comparison of reported and corrected gaze data accuracy and precision

	Min. error	Max. error	Av. error	Std. dev. of error
Reported gaze data	0.03°	2.15°	0.71°	0.41°
Corrected gaze data	0°	1.59°	0.53°	0.28°
Uncertainty Mitigation	100%	26%	25%	32%

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

As shown in the table, the proposed uncertainty mitigation algorithm increases the eye tracking accuracy by 25% and precision by 32%. The maximal registered error (1.59°), average error (0.53°) and standard deviation (0.28°) are still too large to reliably distinguish individual letters in the texts during the reading study that requires the error to be no more than 0.63°. Nevertheless, the minimal error of 0° proves the potential of the proposed approach.

To further improve the accuracy, other calibration patterns should be examined [8] along with more sophisticated versors' interpolation strategies. At the same time, it is important to keep the calibration process as simple as possible so as not to exhaust the user [8,31]. Currently, the built-in and custom calibration routines take ca. 30 s and 15 s respectively, which is fairly fast.

8 Conclusion

We propose a quaternion-based model suitable to evaluate and mitigate eye gaze tracking uncertainty in VR applications. This model requires a single-pass custom calibration of an eye tracker performed right after a built-in calibration procedure. During the custom calibration, the differences between reported and ground-truth gaze directions are encoded as unit quaternions (versors), which are afterward interpolated over the vision area. For each gaze direction sampled from the eye tracking device, the corresponding interpolated versor is used for correction, whereby the real part of this versor serves as a certainty factor.

We implemented the proposed approach in VRSciVi Workbench to use with Vive Pro Eye VR HMD. This allowed us to increase the eye tracking accuracy by 25% and precision by 32%. While the resulting accuracy and precision are still not high enough to reliably study such processes as reading large texts (more than 100 words long) in VR, the proposed approach can be considered as having potential for future refinement. In addition, this approach allows evaluating the uncertainty for each eye gaze tracking sample to consider it during the upcoming analysis stage. Ontology-driven software development tools of SciVi provide high-level means to generate the interface for parametrization of the proposed algorithm and to adapt it for integration in a software to meet the specifics of different VR-based experiments.

For future work, we plan to try overcoming the limitations of the currently implemented algorithm by using machine learning methods for versor interpolation, as well as to experiment with more sophisticated custom calibration strategies.

References

1. Andersson, R., Nyström, M., Holmqvist, K.: Sampling Frequency and Eye-Tracking Measures: How Speed Affects Durations, Latencies, and More. Journal of Eye Movement Research **3**(3) (2010). https://doi.org/10.16910/jemr.3.3.6
2. Binaee, K., Diaz, G., Pelz, J., Phillips, F.: Binocular Eye Tracking Calibration during a Virtual Ball Catching Task Using Head Mounted Display. In: Proceedings of the ACM Symposium on Applied Perception. pp. 15–18 (2016). https://doi.org/10.1145/2931002.2931020

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

3. Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., Ertl, T.: State-of-the-Art of Visualization for Eye Tracking Data. In: EuroVis – STARs. pp. 63–82 (2014). https://doi.org/10.2312/eurovisstar.20141173
4. Chuprina, S., Nasraoui, O.: Using Ontology-Based Adaptable Scientific Visualization and Cognitive Graphics Tools to Transform Traditional Information Systems into Intelligent Systems. Scientific Visualization **8**(1), 23–44 (2016)
5. Clay, V., König, P., König, S.U.: Eye Tracking in Virtual Reality. Journal of Eye Movement Research **12**(1) (2019). https://doi.org/10.16910/jemr.12.1.3
6. Clifford, C.W.G., Palmer, C.J.: Adaptation to the Direction of Others' Gaze: A Review. Frontiers in Psychology **9** (2018). https://doi.org/10.3389/fpsyg.2018.02165
7. Drewes, H., Pfeuffer, K., Alt, F.: Time- and Space-Efficient Eye Tracker Calibration. In: Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications. pp. 1–8 (2019). https://doi.org/10.1145/3314111.3319818
8. Hassoumi, A., Peysakhovich, V., Hurter, C.: Improving Eye-Tracking Calibration Accuracy Using Symbolic Regression. PLoS ONE **14**(3) (2019). https://doi.org/10.1371/journal.pone.0213675
9. Holmqvist, K., Andersson, R.: Eye-tracking: A Comprehensive Guide to Methods, Paradigms and Measures. Lund Eye-Tracking Research Institute, Lund (2017)
10. Iacobi, J.C.E.: Software for Analyzing User Experiences in Virtual Reality Using Eye Tracking. KTH Royal Institute of Technology (2018), https://kth.diva-portal.org/smash/get/diva2:1231972/FULLTEXT01.pdf, visited on 16 Feb 2022
11. Imaoka, Y., Flury, A., de Bruin, E.D.: Assessing Saccadic Eye Movements With Head-Mounted Display Virtual Reality Technology. Frontiers in Psychiatry **11** (2020). https://doi.org/10.3389/fpsyt.2020.572938
12. Jogeshwar, A., Diaz, G., Farnand, S., Pelz, J.: The Cone Model: Recognizing Gaze Uncertainty in Virtual Environments. In: IS&T International Symposium on Electronic Imaging (2020). https://doi.org/10.2352/ISSN.2470-1173.2020.9.IQSP-288
13. Kuipers, J.B.: Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality. Princeton University Press, Princeton, New Jersey (1999)
14. Kuperman, V., Siegelman, N., Schroeder, S., Acartürk, C., Alexeeva, S., Amenta, S., Bertram, R., Bonandrini, R., Brysbaert, M., Chernova, D., Da Fonseca, S.M., Dirix, N., Duyck, W., Fella, A., Frost, R., Gattei, C.A., Kalaitzi, A., Lõo, K., Marelli, M., Nisbet, K., Papadopoulos, T.C., Protopapas, A., Savo, S., Shalom, D.E., Slioussar, N., Stein, R., Sui, L., Taboh, A., Tønnesen, V., Usal, K.A.: Text Reading in English as a Second Language: Evidence from the Multilingual Eye-Movements Corpus (MECO). Studies in Second Language Acquisition (2020)
15. Lang, B.: Eye-Tracking is a Game Changer for VR That Goes Far Beyond Foveated Rendering. https://www.roadtovr.com/ (2013), https://www.roadtovr.com/why-eye-tracking-is-a-game-changer-for-vr-headsets-virtual-reality/, visited on 16 Feb 2022
16. Llanes-Jurado, J., Marín-Morales, J., Guixeres, J., Alcañiz, M.: Development and Calibration of an Eye-Tracking Fixation Identification Algorithm for Immersive Virtual Reality. Sensors **20**(17) (2020). https://doi.org/10.3390/s20174956
17. Lohr, D.J., Friedman, L., Komogortsev, O.V.: Evaluating the Data Quality of Eye Tracking Signals from a Virtual Reality System: Case Study using SMI's Eye-Tracking HTC Vive. arXiv (2019), https://arxiv.org/abs/1912.02083, visited on 16 Feb 2022

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51

18. Mirault, J., Guerre-Genton, A., Dufau, S., Grainger, J.: Using Virtual Reality to Study Reading: An Eye-Tracking Investigation of Transposed-Word Effects. Methods in Psychology **3** (2020). https://doi.org/10.1016/j.metip.2020.100029
19. Mohanto, B., Islam, A.B.M.T., Gobbetti, E., Staadt, O.: An Integrative View of Foveated Rendering. Computers & Graphics (2021). https://doi.org/10.1016/j.cag.2021.10.010
20. Popov, A., Shakhuro, V., Konushin, A.: Operator's Gaze Direction Recognition. In: Proceedings of the 31st International Conference on Computer Graphics and Vision (GraphiCon 2021). pp. 475–487 (2021). https://doi.org/10.20948/graphicon-2021-3027-475-487
21. Qian, K., Arichi, T., Price, A., Dall'Orso, S., Eden, J., Noh, Y., Rhode, K., Burdet, E., Neil, M., Edwards, A.D., Hajnal, J.V.: An Eye Tracking Based Virtual Reality System for Use Inside Magnetic Resonance Imaging Systems. Scientific Reports **11** (2021)
22. Ryabinin, K., Belousov, K.: Visual Analytics of Gaze Tracks in Virtual Reality Environment. Scientific Visualization **13**(2), 50–66 (2021). https://doi.org/10.26583/sv.13.2.04
23. Ryabinin, K., Chuprina, S.: Development of Ontology-Based Multiplatform Adaptive Scientific Visualization System. Journal of Computational Science **10**, 370–381 (2015). https://doi.org/10.1016/j.jocs.2015.03.003
24. Ryabinin, K., Chuprina, S.: Ontology-Driven Edge Computing. Lecture Notes in Computer Science **12143**, 312–325 (2020). https://doi.org/10.1007/978-3-030-50436-6_23
25. Sharafi, Z., Sharif, B., Guéhéneuc, Y.G., Begel, A., Bednarik, R., Crosby, M.: A Practical Guide on Conducting Eye Tracking Studies in Software Engineering. Empirical Software Engineering **25**, 3128–3174 (2020)
26. Shehu, I.S., Wang, Y., Athuman, A.M., Fu, X.: Remote Eye Gaze Tracking Research: A Comparative Evaluation on Past and Recent Progress. Electronics **10**(24) (2021). https://doi.org/10.3390/electronics10243165
27. Sipatchin, A., Wahl, S., Rifai, K.: Eye-Tracking for Clinical Ophthalmology with Virtual Reality (VR): A Case Study of the HTC Vive Pro Eye's Usability. Healthcare **9**(2) (2021). https://doi.org/10.3390/healthcare9020180
28. Skulmowski, A., Bunge, A., Kaspar, K., Pipa, G.: Forced-Choice Decision-Making in Modified Trolley Dilemma Situations: A Virtual Reality and Eye Tracking Study. Frontiers in Behavioral Neuroscience **8** (2014). https://doi.org/10.3389/fnbeh.2014.00426
29. Sonntag, D., Orlosky, J., Weber, M., Gu, Y., Sosnovsky, S., Toyama, T., Toosi, E.N.: Cognitive Monitoring via Eye Tracking in Virtual Reality Pedestrian Environments. In: Proceedings of the 4th International Symposium on Pervasive Displays. pp. 269–270 (2015). https://doi.org/10.1145/2757710.2776816
30. Stein, N., Niehorster, D.C., Watson, T., Steinicke, F., Rifai, K., Wahl, S., Lappe, M.: A Comparison of Eye Tracking Latencies Among Several Commercial Head-Mounted Displays. i-Perception **12**(1), 1–16 (2021). https://doi.org/10.1177/2041669520983338
31. Tripathi, S., Guenter, B.: A Statistical Approach to Continuous Self-Calibrating Eye Gaze Tracking for Head-Mounted Virtual Reality Systems. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 862–870 (2017). https://doi.org/10.1109/WACV.2017.101

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_51