Real-time head-based deep-learning model for gaze probability regions in collaborative VR

Riccardo Bovo*
Imperial College London
London, United Kingdom
rb1619@ic.ac.uk

Daniele Giunchi*
University College London
London, United Kingdom
d.giunchi@ucl.ac.uk

Ludwig Sidenmark
Lancaster university
Lancaster, United Kingdom
l.sidenmark@lancaster.ac.uk

Enrico Costanza
University College London
London, United Kingdom
e.costanza@ucl.ac.uk

Hans Gellersen
Lancaster University
Lancaster, United Kingdom
h.gellersen@lancaster.ac.uk

Thomas Heinis
t.heinis@ic.ac.uk
Imperial College London
London, United Kingdom

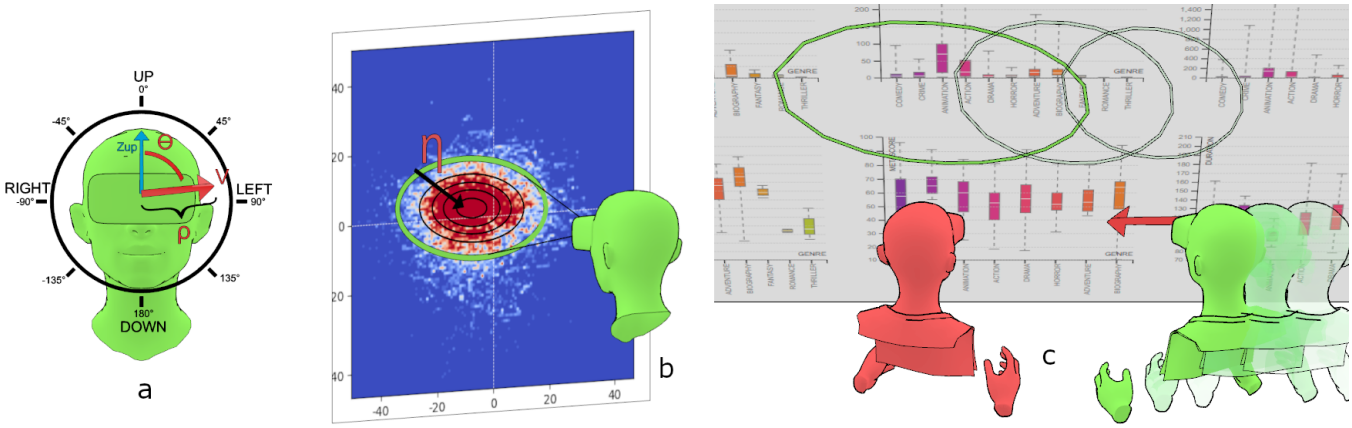


Figure 1: a) The head rotational velocity direction (θ) and magnitude (ρ) are extracted during a VR session. b) Probability density functions are extracted from eye-gaze distributions that correspond to the head rotational velocity and are converted into a series of percentile-based contours (η). c) Our real-time model uses the three parameters (θ,ρ,η) to provide a novel representation of visual attention for VR collaboration or interaction.

ABSTRACT

Eye behavior has gained much interest in the VR research community as an interactive input and support for collaboration. Researchers used head behavior and saliency to implement gaze inference models when eye-tracking is missing. However, these solutions are resource-demanding and thus unfit for untethered devices, and their angle accuracy is around 7°, which can be a problem in high-density informative areas. To address this issue, we propose a lightweight deep learning model that generates the probability density function of the gaze as a percentile contour. This solution allows us to introduce a visual attention representation based on a region rather than a point. In this way, we manage the trade-off between the ambiguity of a region and the error of a point. We tested our model in untethered devices with real-time performances; we evaluated its accuracy, outperforming our identified baselines (average fixation map and head direction).

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **Collaborative interaction**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

neural networks, gaze prediction, gaze inference, visual attention

ACM Reference Format:
Riccardo Bovo, Daniele Giunchi, Ludwig Sidenmark, Enrico Costanza, Hans Gellersen, and Thomas Heinis. 2022. Real-time head-based deep-learning model for gaze probability regions in collaborative VR. In *ETRA '22: ACM Symposium on Eye Tracking Research and Applications, June 08–11, 2022, Seattle, WA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3517031.3529642

*Both authors contributed equally to this research.

ETRA '22, June 03–05, 2022, Seattle, WA

ACM ISBN 978-1-4503-9252-5/22/06…$15.00
https://doi.org/10.1145/3517031.3529642

1 INTRODUCTION AND RELATED WORKS

In collaborative settings, gaze is seen as a vital cue for efficient communication and collaboration and can be used to predict collaborators' intention [Baron-Cohen et al. 1997], seek approval [Efran 1968], or understand the desire to communicate [Ho et al. 2015]. Gaze cues that show the current gaze position to collaborators are usually visualized through a cursor [Jing et al. 2021; Kim et al. 2020; Lee et al. 2017] or a ray [Bai et al. 2020; Jing et al. 2021; Li et al. 2019] in augmented and virtual reality (AR/VR) settings, and have been extensively studied to improve collaboration and communication. These works, together with results in desktop settings, have shown that gaze-based cues can be effective for establishing mutual orientation [D'Angelo and Gergle 2016; Hindmarsh et al. 1998; Jing et al. 2021; Li et al. 2016], enhancing the feeling of co-presence [Bai et al. 2020; Gupta et al. 2016], and improving the awareness of collaborators' attention and actions [Jing et al. 2021; Kuhn et al. 2009; Lee et al. 2017; Newn et al. 2017].

In VR, it is also common to use the head orientation as an approximation of gaze, as eye tracking is not widely available in low-cost HMDs (e.g., Oculus Quest), and is prone to errors [Holmqvist et al. 2012]. As such, the head direction is commonly used as a replacement or proxy for gaze-based cues in collaborative AR, or VR [Atienza et al. 2016; Li et al. 2019]. Furthermore, due to the head's synergistic relationship with the eyes [Kollenberg et al. 2010; Pfeil et al. 2018; Sidenmark and Gellersen 2019a], head movement is a common feature in visual attention models that mimic gaze for egocentric videos [Li et al. 2013; Matsuo et al. 2014; Nakashima et al. 2015; Yamada et al. 2011], or VR [Hu et al. 2020, 2019]. However, head movements fail to approximate the fine-grained movements performed by the eyes. This means that users have to perform more head movement to align a head-based pointer with gaze during interaction [Sidenmark and Gellersen 2019b], and that visual attention models are not able to achieve accurate gaze point prediction based on head movements alone. Therefore, head-based collaboration cues commonly apply large frustum- or cone-shaped cursors to cover wider fields of view to ensure that the gaze position is within the cue [Bai et al. 2020; Piumsomboon et al. 2017]. Meanwhile, gaze prediction models based on head movement commonly include visual saliency as a feature to increase model accuracy [Hu et al. 2020, 2019; Li et al. 2013; Matsuo et al. 2014; Nakashima et al. 2015; Yamada et al. 2011]. However, while models can easily retrieve head movements from the HMD sensors in real-time, visual saliency is a resource-demanding process [Hu et al. 2020, 2019].

This work proposes a saliency-free deep-learning gaze-prediction model for visual attention cues in collaborative environments using only head movements for real-time use in low-cost HMDs (Figure 1a). We choose the multi-perceptron (MLP) neural network (NN) because it is widely used in regression problems [Murtagh 1991]. Our MLP architecture is simple, with a small amount of fully connected layers that learn the center and contours of probable fixation locations from a set of generated eye-gaze probability density functions (PDFs) (Figure 1b, Figure 2). These PDFs are pre-generated starting from gaze data filtered by head movement velocity (i.e., direction θ and magnitude ρ) and using a novel method. Our model does not aim to infer users' exact gaze position but rather the contour of probable gaze positions to address the accuracy limitation of head movements and minimize the size of head-based cue visualizations. Predicting the area of interest helps in a collaborative context where dense information can lead to erroneous interpretation if only gaze location coordinates are indicated. The model does not rely on visual saliency and can thus be used in a wide array of environments and contexts. We envisage the model to be effectively applied as an improved head pointer for interaction more in line with the gaze position, as a gaze-based attention indicator in co-located scenarios, or as a method to reconstruct gaze in offline analysis without knowledge of the gaze position or visual saliency.

We trained and evaluated our proposed model with a dataset of gaze and head data from 13 participants looking at 360 VR videos [Agtzidis et al. 2019]. We compared our novel PDF generator against the PDF generated by an improved 2D Gaussian fitting approach. To evaluate our deep-learning model, we compare it to a head-based baseline, the average fixation map (AFM) calculated as proposed by Radkowski [2015] and demonstrated to be a valid gaze prediction method by Tavakoli et al. [2019] (Figure 2f). We found that our novel PDF generator method outperform the Gaussian approach within certain range of velocities (θ =[-90 °,-45°],[45°, 80 °] and for ρ=[1°/s, 6.5 °/s]) and that our deep-learning model better reflects gaze during head movements than the head-baselines.

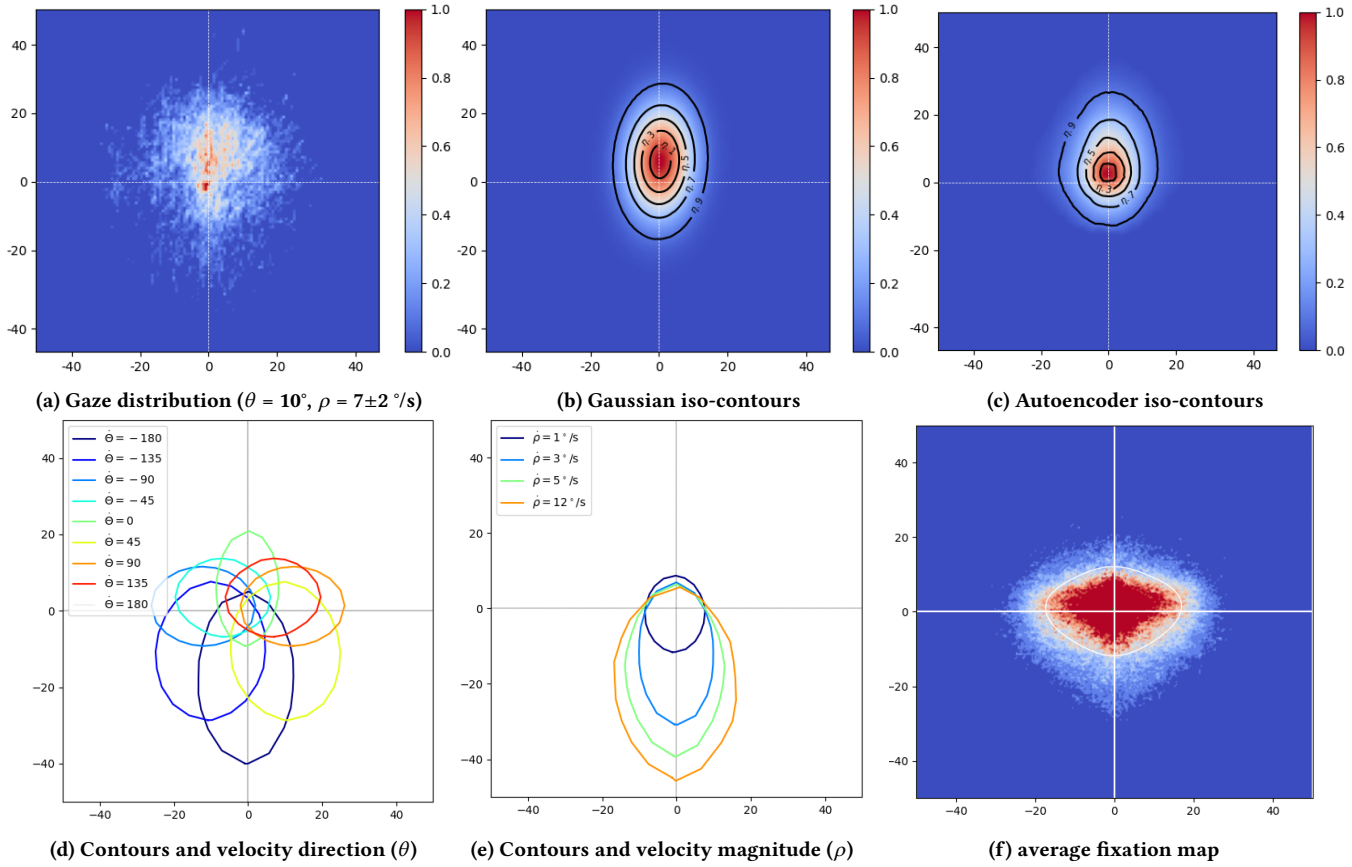
The contribution of this paper is three-fold. Firstly, we propose a novel method to generate PDFs from gaze maps based on a convolution auto-encoder which performs better than the multivariate Gaussian fitting (MGF) function according to specific head rotational velocity directions (θ) and magnitudes(ρ). Secondly, we introduce a lightweight visual attention model based on a MLP architecture tested on untethered devices. Thirdly, we evaluate our model's performance and comparison with the head-based model as baselines (i.e., head direction and AFM). We show that our model has better accuracy than the identified baseline.

2 MAIN CONCEPTS

Previous works focused on predicting the exact gaze location as a 2D point. This approach can be error-prone in collaborative settings when the scene contains high-density areas of information since even a few degrees of prediction error can point towards entirely different pieces of information. To address this issue, we propose to capture the shape of gaze distribution instead of the most informative point. This approach exploits the trade-off between the ambiguity of a region's information and the error that a point can introduce. Our model's inputs consist of the head's rotational vector (θ, ρ) and contour percentile (η); the latter represents the trade-off between point/region and error/ambiguity (Figures 1b, 2b and 2c).

2.1 Visualization Dimensionality

PDFs are commonly visualized as a gradient to show the model's continuous probability output in space. Alternative visualizations of PDFs consist of percentile-based contour lines, or a closed line that divides the space into two regions, inside and outside, offering a binary classification. These visualizations only partially capture the PDF unless multiple lines are visualized (e.g., isohypse geological altitude representations or isobars in meteorology maps). As the model's purpose is to visualize where a user's visual attention



(a) Gaze distribution (θ = 10°, ρ = 7±2 °/s) **(b) Gaussian iso-contours** **(c) Autoencoder iso-contours**

(d) Contours and velocity direction (θ) **(e) Contours and velocity magnitude (ρ)** **(f) average fixation map**

Figure 2: a) A distribution of eye gaze samples generated from the two parameters of head angular velocity direction θ = 10°and magnitude ρ = 7°/s b) The PDF extracted via the MGF (Section 2.3.2), and the percentile-based contours extracted from the PDF. c) The PDF extracted via the Autoencoder (Section 2.3.1), and the percentile-based contours extracted from the PDF. d) A depiction of how contours change depending on the value of head angular velocity direction (θ) while the percentile ρ is fixed. e) A depiction of how contours change depending on the value of head rotational velocity magnitude (ρ) while the percentile parameter and θ are fixed. f) Average Fixation Map: we plot all the gaze data samples from the dataset [Agtzidis et al. 2019], and we generate an AFM calculated as proposed by Radkowski [2015] from the AFM. We generate the contour which represents 70% of the data. From this image is clear that the gaze distribution does not have a Gaussian shape.

is likely to be without occluding the virtual scene, a line type visualization is sufficient to convey the information and, unlike a gradient type visualization, does not occlude the scene. Moreover, a gradient in 2D space requires a bi-dimensional coordinate plus the value associated with them, while a line requires a bi-dimensional coordinate but no associated value.

2.2 Head Rotational Velocity Direction and Magnitude

We parametrize head rotational velocity as magnitude (ρ) and the angle with the z axes of the space (θ) (Figure 1). We chose such parameterization to be more understandable to humans by decoupling the velocity direction and magnitude parameters, unlike previous research where direction and magnitude are hidden among two Cartesian coordinates. This parameterization allows us to create a simple analysis tool to navigate the dataset distributions and inform our machine learning method.

2.3 Deep Learning Pipeline

We selected a dataset created by Agtzidis et al. [Agtzidis et al. 2019] containing both eye- and head-tracking in a virtual environment. They recruited 13 participants and played 15 360-panoramic videos for each participant for a total of 195 video sessions. We randomly split the sessions with a ratio of 2/3 (130 videos, 936000 data samples) for training and 1/3 (65 videos, 468000 data samples) for evaluation. We generated a series of gaze distributions from the dataset by filtering the data using different head rotational velocity directions (θ) and magnitudes (ρ). We then saved these distributions as gray-scaled images with a size of 128x128. Because we noticed that the gaze samples did not always have a Gaussian shape, we generated

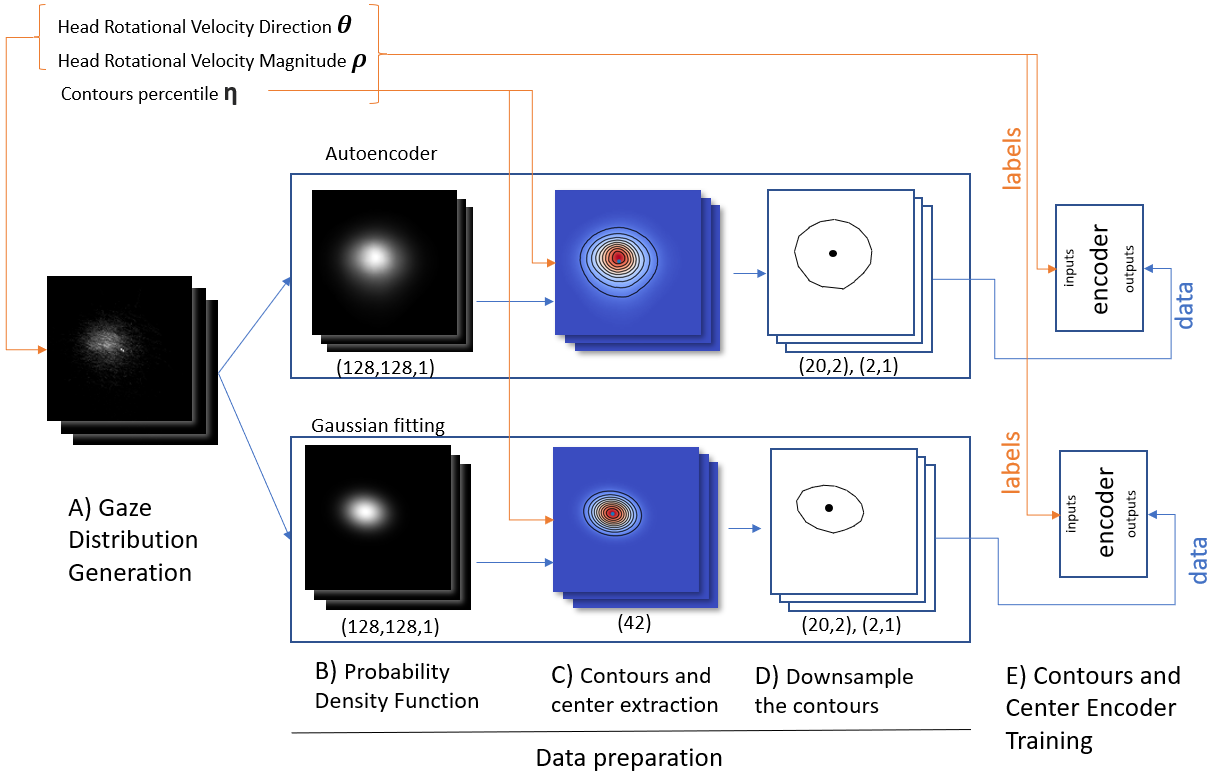


Figure 3: Our model's three-stage training pipeline involves two deep learning models and a contour extraction phase to process velocity angle and magnitude (θ,ρ) to generate the contour that contains the requested probability of gaze area. The contour encloses the visual interest area and includes the predicted gaze location. The first pipeline training stage consists of an autoencoder reconstructing the 2D PDF from gaze location distribution samples, or an MGF approach. The second stage is a multi-perceptron architecture that encodes contours from the angles and magnitudes of head-shift velocity.

PDFs with two different methods. Firstly, by using a trained autoencoder (Section 2.3.1) that captures the shape of the distribution and secondly, by fitting a multivariate Gaussian distribution (Section 2.3.2). Both models output gray-scaled images of the same size ($128x128$). We extracted the center and percentiles contours from each PDF paired with the different head angular velocity angles and magnitudes. We then trained a contour encoder to learn the various centers and contours from the PDFs grouped by head angular velocity (θ, ρ) and the contours percentile (η, Figure 1b). We used a shallow model, avoiding complex neural networks architectures such as the ones proposed by Hu et al. [2020, 2019].

2.3.1 PDF from Autoencoder. The dataset's eye gaze distribution does not have the Gaussian shape as shown in Figure 2f where a diamond shape is visible from the AFM. Therefore the autoencoder aims to provide an alternative approach to the standard Gaussian model. We developed the autoencoder with Keras/Tensorflow 2.7.0, and has a standard autoencoder architecture and is trained with a list of multivariate distributions such as Gaussian, Poisson, and Skewed Gaussian. We generate a training set with the distributions' PDFs as labels and data generated by applying uniform noises to PDFs. This method achieves a visually similar distribution to the real-world gaze-tracking data. We call this variant the *PDF autoencoder*, and accepts generated gray-scaled $128x128$ images as input. The output of the PDF autoencoder is the PDF given the heatmap of the gaze distributions obtained after filtering the data with the direction and amplitude of the head rotational velocity.

2.3.2 PDF from Multivariate Gaussian Fitting (MGF). Previous work used Gaussian fitting to describe a distribution of gaze locations Nakashima et al. [2015]; Yamada et al. [2011]. However, their implementations split the X and Y components of the distribution, losing their correlation and removing the Gaussian shape's tilting orientation. Instead, our proposed MGF is implemented so that the Gaussian shape can fit with a tilt value and better follow the distribution shape. We used Scipy 1.7.2 to calculate MGF. This training set and the PDF autoencoder set are created starting with the same samples using random θ and ρ.

2.3.3 Contour extraction. The PDF images generated by the autoencoder and the MGF are processed to extract contours. Every percentile-based contour is associated with the two parameters of head velocity (θ,ρ) and the percentile value (η). The three parameters (θ,ρ,η) represent the training set for the next stage of the pipeline (Figure 3). The prediction stage does not require the autoencoder, and such knowledge is embedded directly into the next model (Contour encoder section 2.3.4). After generating the resulting PDF

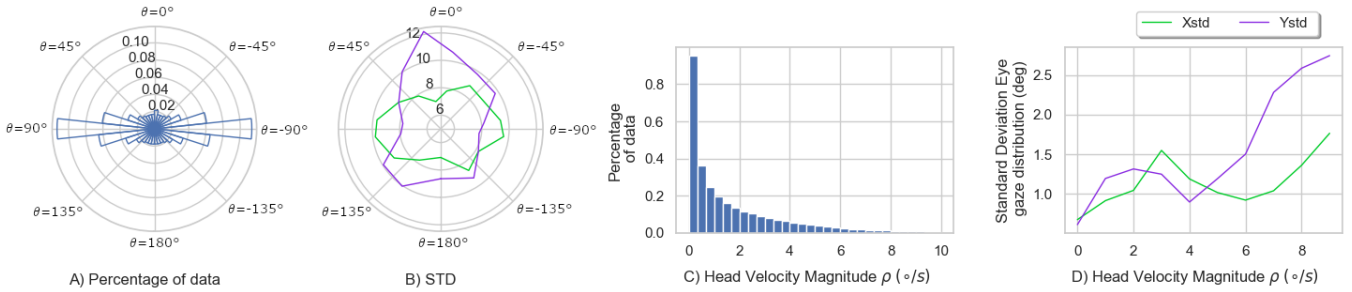


Figure 4: A) Percentage of data samples in the dataset for each head angular direction. B) X and Y Standard deviation of the eye-gaze samples' distribution for each head angular direction. C) Percentage of data present in the dataset for each head velocity magnitude (degrees/s). D) X and Y Standard deviation of the eye-gaze samples' distribution for each value of head velocity magnitude (degrees/s).

images from the autoencoder and the MGF, we perform a contour extraction based on the percentiles ($\eta_{.1}$,$\eta_{.2}$,$\eta_{.3}$,$\eta_{.4}$,$\eta_{.5}$,$\eta_{.7}$,$\eta_{.8}$,$\eta_{.9}$,η_1) of the samples of the gaze distributions. Contour extraction is performed via a python script and the library "shapely". After a geometric downsample, each contour is reduced to a sequence of 20 points with two coordinates. We also include the center of the contour as an additional dataset point to represent the estimated gaze location. After grouping numerical input data and such contours, we obtain the training dataset that is the input for the final training phase.

2.3.4 Contour Encoder. The second part of the training pipeline consisted of the Contour Encoder, which is used when predicting the contour and the gaze location. We visualize the PDF as an iso-contour adopting a lower dimensionality than a PDF gradient. Therefore, our neural network does not need 2D convolution (Section 2.1). Moreover, we choose not to use Recurrence Neural Networks (RNN) for two reasons. First, the resource consumption due to recurrence may not fulfill our real-time performance requirement. Second, the training of RNN can be affected by a vanishing gradient fostered by a high number of eye-fixations with a head rotational velocity of zero of the dataset (see the histogram in Figure 4). We want to keep the Neural Network (NN) architecture as simple as possible to achieve fast inference and portability to mobile headsets; therefore, we select MLP architecture. We design this NN as a simple fully connected model with four layers of 2048 hidden units. We train the model with the contours generated from the PDF autoencoder and MGF for comparison. The contour encoder inputs consist of three parameters (θ,ρ,η) and an output of 42 parameters: 20 2D points for the contour and one 2D point for the gaze.

2.3.5 VR-NN model implementation. We use TensorflowLite to port the model to an untethered VR headset by converting it to an Android-compatible version that can be loaded and executed via the Unity framework. In addition, we improve model stability when velocity is close to zero by linearly combining the contour estimation of the neural network and the AFM via Formula 1. Where V is the head angular velocity, V_{thres} is a velocity threshold over which no linear combination is performed. We use V_{thres}=0.05 upon visual estimation. t is the parameter with a range from 0.0 to 1.0. The AFM contour is displayed when t = 0.0, while the pure neural network contour is chosen when t = 1.0.

$$Contour_{V<V_{thres}} = (1-t) * Contour_{AFM} + t * Contour_{Neural} \quad (1)$$

3 RESULTS ANALYSIS

3.1 Data Exploration

We look at the overall dataset distribution of head angular velocity magnitude and direction to contextualize the evaluation of our autoencoder and Gaussian models,. The histogram of data samples to velocity magnitude shows that most of the data concentrate at low-velocity magnitude where the head is stationary (Figure 4C). Furthermore, the data also indicates that gaze samples relative to the head movement directions are most prevalent in horizontal directions (Figure 4A). The Standard deviation (STD) of the eye-gaze distributions measures show the significant sample spread. By measuring STD on the data filtered by head angular velocity direction and magnitude, we can sense how the PDF and contours change size and shape depending on head rotational velocity. When looking at STD changes in relation to the head direction (Figure 4B) we see that the X STD reaches the maximum when the head is rotating towards the left (θ = 90 °) or right (θ = -90 °) side while the Y STD reach the maximum when the head rotation is upward (θ = 0 °) or downward (θ = 180 or -180 °). Such results show a correlation of STD with the head rotation direction. When the head rotational direction is upward/downward, the eyes STD increases in the upward/downward axes (Y STD) and diminishes for the left/right (X STD). Likewise, when the head rotational direction is left/right, the eyes STD grows in the X and diminishes for the Y. Such change in STD is confirmed by the shape of the contours generated by the models (Figure 2d). The contours' size changes on the X and Y accordingly to X STD and Y STD. Furthermore, the STD increases as head velocity magnitude increases. Such result is consistent with the size of contours which increases with the increments in head velocity magnitude (ρ) as shown in Figure 2e.

3.2 Performance

We tested our model's temporal performance on two untethered devices. We measured an average inference time on Oculus Quest of 8 ms, while for Oculus Quest 2 of 3 ms. Such results are compatible with the Oculus refresh rate showing that our model operates in

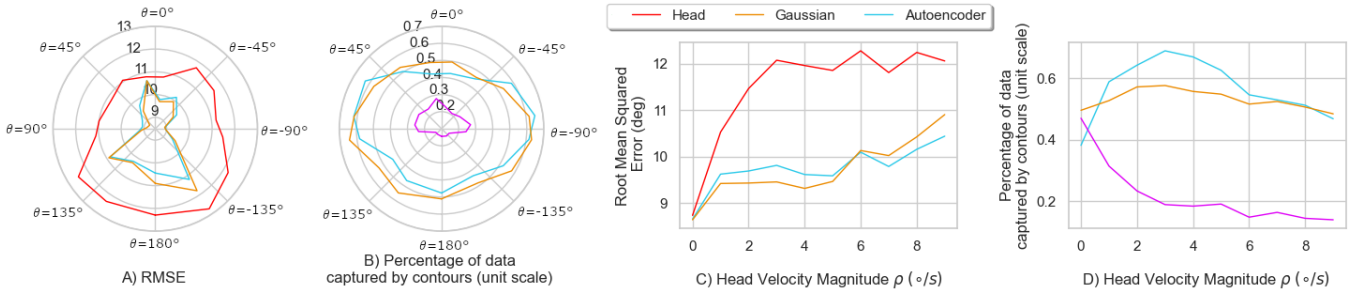


Figure 5: Results of the Autoencoder, Gaussian, and head direction baseline. A) RMSE of the predicted center relative to the distributions of eye-gaze samples for each head direction B) Percentage of data captured by the model contours for each head rotation direction C) RMSE of the models relative to Head Angular Velocity (degrees/s) D) The percentage of the data captured by the models' 70th percentile contours relative to the head's angular velocity magnitude (degrees/s).

real-time. To evaluate our model, we compared it to our identified baselines: the head direction and the average fixation map. We used two different metrics: firstly, the Root Mean Squared Error (RMSE) to identify how erroneous is the predicted center compared to the eye-gaze distributions. Secondly, we used the percentage of the data samples contained in the contours (in unit scale).

3.2.1 RMSE. We measured the performance of the models related to inferred gaze position by calculating RMSE between the inferred gaze and the real gaze (Figure 5A and Figure 5C). Comparing the models RMSE results we see that the autoencoder model performs better than the Gaussian for θ =[-90°, 90°] (Figure 5A), and for ρ=[0°/s, 6°/s] (Figure 5C). Both models perform consistently better than the baseline (i.e., head direction). Both models, as well as the baseline, tend to reach the lowest accuracy when downward head movements are performed (Figure 5A). Such a result is consistent with previous studies which describe how the eyes contribute more than the head during downward gaze shifts [Sidenmark and Gellersen 2019b]; because the eyes contribute more (when needed), the accuracy of both models and the baseline reduces during downward movements. We also can see that the model's minimum accuracy corresponds to the left/right head direction. Our models are consequently most efficient during left and right head movements. Such results agree with previous work from Hu et al. [2019] which explores the Pearson's Correlation Coefficient (PCC) of X and Y components of head rotational velocity and highlights that the X component correlates with the gaze location more than the Y. Model performance to the head velocity magnitude (ρ) (Figure 5C) show that when ρ is close to zero both models' accuracy are equivalent to the baseline. On the other hand, the models perform significantly better than the baseline across most of the range of ρ.

3.2.2 Percentage of data captured. We also measure the performance of the contours in capturing the eye-gaze samples. To do this analysis we use the 70% percentile contour (Figure 5B and Figure 5D). We use this metric to compare the models to each other and with our baseline, the AFM (Figure 2f). Results highlight how both models perform better than the AFM and how the best performance corresponds to the left/right direction, likewise for the MSE of the center. Comparison with the AFM also shows that when head rotational velocity is close to zero, the AFM performs equally well as the models (Figure 5D) likewise for the MSE center baseline. However, when the head is moving, results show significantly better performance for both models when compared to the baseline. We exploit this AFM characteristic to stabilize our model output when zero velocity magnitude happens. We compare the models percentage of data captured and we see how the autoencoder model performs better than the Gaussian for θ =[-90°,-45°],[45°, 80°](Figure 5B), and for ρ=[1 °/s, 6.5 °/s] (Figure 5D). Therefore, we can say that our approach to generate PDF based on autoencoder better captures the shape of gaze distribution in relevant ranges like the left/right head movements.

4 CONCLUSION AND FUTURE WORKS

During collaboration, visual attention cues can help clarify the shared visual context, allowing participants to see each other's focus. Mutual awareness of visual attention simplifies communication by reducing the necessity of verbally explicit references or hand-pointing gestures to identify/negotiate the current direction of the collaborative effort. This paper presented a lightweight deep learning model that can predict the PDF of gaze in VR based on head motion. The model is based on a novel method to generate PDFs. Our model facilitates interaction in collaborative VR by explicitly depicting the collaborator's likely area of visual focus in real-time. We developed and compared two models for inferring gaze and its PDF as a percentile contour. Results of RMSE of the predicted eye gaze show that both models perform better than the AFM baseline, and when compared to each other, they have similar results in terms of accuracy. The equal performance of both models shows that the Autoencoder can be effectively used as an alternative to the standard Gaussian 2D approach.

Furthermore, we demonstrated that our multi-perception model can be used in real-time applications on untethered devices. Our model and source code are available for the research and VR developers communities via GitHub[1]. Our implementation achieves a real-time performance on a low-cost Oculus Quest with an inference time of 7ms, demonstrating that our approach can be used to replace head orientation as an approximation of gaze in collaborative VR applications.

[1]https://github.com/Collaborative-Immersive-Visual-Toolkit/VR_Iso_Gaze_Contours

Moreover, the same technique could be used in post-process analysis scenarios as a fast method to estimate gaze from 3 DoF head data. In addition, this paper introduces a novel type of coordinates for head rotational speed with the characteristics of being human-readable and that can be used to analyze gaze distributions in function of the head rotational velocity. Further work could be carried out to compare our model with the AFM baseline via psychophysical experimentation and extend the model with different dataset types. For interactive applications, the model could be tested in a pointing task [Teather and Stuerzlinger 2011]. The same comparison but with the models' contours instead could be carried out for collaborative applications performing mixed methods studies such as [Piumsomboon et al. 2019; Prilla 2019] comprehensive of quantitative and qualitative measures.

ACKNOWLEDGMENTS

This work was supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant No. 101021229 GEMINI: Gaze and Eye Movement in Interaction, and Grant No. 739578 RISE).

REFERENCES

Ioannis Agtzidis, Mikhail Startsev, and Michael Dorr. 2019. 360-Degree Video Gaze Behaviour: A Ground-Truth Data Set and a Classification Algorithm for Eye Movements. In *Proceedings of the 27th ACM International Conference on Multimedia* (Nice, France) *(MM '19)*. Association for Computing Machinery, New York, NY, USA, 1007–1015. https://doi.org/10.1145/3343031.3350947

Rowel Atienza, Ryan Blonna, Maria Isabel Saludares, Joel Casimiro, and Vivencio Fuentes. 2016. Interaction techniques using head gaze for virtual reality. In *2016 IEEE Region 10 Symposium (TENSYMP)*. 110–114. https://doi.org/10.1109/TENCONSpring.2016.7519387

Huidong Bai, Prasanth Sasikumar, Jing Yang, and Mark Billinghurst. 2020. A User Study on Mixed Reality Remote Collaboration with Eye Gaze and Hand Gesture Sharing. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376550

Simon Baron-Cohen, Sally Wheelwright, , and Therese Jolliffe. 1997. Is There a "Language of the Eyes"? Evidence from Normal Adults, and Adults with Autism or Asperger Syndrome. *Visual Cognition* 4, 3 (1997), 311–331. https://doi.org/10.1080/713756761 arXiv:https://doi.org/10.1080/713756761

Sarah D'Angelo and Darren Gergle. 2016. Gazed and Confused: Understanding and Designing Shared Gaze for Remote Collaboration. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2492–2496. https://doi.org/10.1145/2858036.2858499

Jay S. Efran. 1968. Looking for approval: Effects on visual behavior of approbation from persons differing in importance. *Journal of Personality and Social Psychology* 10, 1 (1968), 21–25. https://doi.org/10.1037/h0026383

Kunal Gupta, Gun A. Lee, and Mark Billinghurst. 2016. Do you see what i see? the effect of gaze tracking on task space remote collaboration. *IEEE Transactions on Visualization and Computer Graphics* 22, 11 (11 2016), 2413–2422. https://doi.org/10.1109/TVCG.2016.2593778

Jon Hindmarsh, Mike Fraser, Christian Heath, Steve Benford, and Chris Greenhalgh. 1998. Fragmented Interaction: Establishing Mutual Orientation in Virtual Environments. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work* (Seattle, Washington, USA) *(CSCW '98)*. Association for Computing Machinery, New York, NY, USA, 217–226. https://doi.org/10.1145/289444.289496

Simon Ho, Tom Foulsham, and Alan Kingstone. 2015. Speaking and Listening with the Eyes: Gaze Signaling during Dyadic Interactions. *PLOS ONE* 10, 8 (08 2015), 1–18. https://doi.org/10.1371/journal.pone.0136905

Kenneth Holmqvist, Marcus Nyström, and Fiona Mulvey. 2012. Eye Tracker Data Quality: What It is and How to Measure It. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 45–52. https://doi.org/10.1145/2168556.2168563

Zhiming Hu, Sheng Li, Congyi Zhang, Kangrui Yi, Guoping Wang, and Dinesh Manocha. 2020. DGaze: CNN-Based Gaze Prediction in Dynamic Scenes. *IEEE Transactions on Visualization and Computer Graphics* 26, 5 (2020), 1902–1911. https://doi.org/10.1109/TVCG.2020.2973473

Zhiming Hu, Congyi Zhang, Sheng Li, Guoping Wang, and DInesh Manocha. 2019. SGaze: A Data-Driven Eye-Head Coordination Model for Realtime Gaze Prediction. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (2019), 2002–2010. https://doi.org/10.1109/TVCG.2019.2899187

Allison Jing, Kieran William May, Mahnoor Naeem, Gun Lee, and Mark Billinghurst. 2021. EyemR-Vis: Using Bi-Directional Gaze Behavioural Cues to Improve Mixed Reality Remote Collaboration. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 283, 7 pages. https://doi.org/10.1145/3411763.3451844

Seungwon Kim, Allison Jing, Hanhoon Park, Soo-hyung Kim, Gun Lee, and Mark Billinghurst. 2020. Use of Gaze and Hand Pointers in Mixed Reality Remote Collaboration. In *The 9th International Conference on Smart Media and Applications. SMA, Jeju, Republic of Korea*. 1–6.

Tobit Kollenberg, Alexander Neumann, Dorothe Schneider, Tessa-Karina Tews, Thomas Hermann, Helge Ritter, Angelika Dierker, and Hendrik Koesling. 2010. Visual Search in the (Un)Real World: How Head-Mounted Displays Affect Eye Movements, Head Movements and Target Detection. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (Austin, Texas) *(ETRA '10)*. Association for Computing Machinery, New York, NY, USA, 121–124. https://doi.org/10.1145/1743666.1743696

Gustav Kuhn, Benjamin W. Tatler, and Geoff G. Cole. 2009. You look where I look! Effect of gaze cues on overt and covert attention in misdirection. *Visual Cognition* 17, 6-7 (2009), 925–944. https://doi.org/10.1080/13506280902826775 arXiv:https://doi.org/10.1080/13506280902826775

Gun A. Lee, Seungwon Kim, Youngho Lee, Arindam Dey, Thammathip Piumsomboon, Mitchell Norman, and Mark Billinghurst. 2017. Improving Collaboration in Augmented Video Conference using Mutually Shared Gaze. In *ICAT-EGVE 2017 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*, Robert W. Lindeman, Gerd Bruder, and Daisuke Iwai (Eds.). The Eurographics Association. https://doi.org/10.2312/egve.20171359

Jerry Li, Mia Manavalan, Sarah D'Angelo, and Darren Gergle. 2016. Designing Shared Gaze Awareness for Remote Collaboration. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion* (San Francisco, California, USA) *(CSCW '16 Companion)*. Association for Computing Machinery, New York, NY, USA, 325–328. https://doi.org/10.1145/2818052.2869097

Yin Li, Alireza Fathi, and James M. Rehg. 2013. Learning to Predict Gaze in Egocentric Video. In *2013 IEEE International Conference on Computer Vision*. 3216–3223. https://doi.org/10.1109/ICCV.2013.399

Yuan Li, Feiyu Lu, Wallace S Lages, and Doug Bowman. 2019. Gaze Direction Visualization Techniques for Collaborative Wide-Area Model-Free Augmented Reality. In *Symposium on Spatial User Interaction* (New Orleans, LA, USA) *(SUI '19)*. Association for Computing Machinery, New York, NY, USA, Article 11, 11 pages. https://doi.org/10.1145/3357251.3357583

Kenji Matsuo, Kentaro Yamada, Satoshi Ueno, and Sei Naito. 2014. An Attention-Based Activity Recognition for Egocentric Video. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 565–570. https://doi.org/10.1109/CVPRW.2014.87

Fionn Murtagh. 1991. Multilayer perceptrons for classification and regression. *Neurocomputing* 2, 5-6 (1991), 183–197. https://doi.org/10.1016/0925-2312(91)90023-5

Ryoichi Nakashima, Yu Fang, Yasuhiro Hatori, Akinori Hiratani, Kazumichi Matsumiya, Ichiro Kuriki, and Satoshi Shioiri. 2015. Saliency-based gaze prediction based on head direction. *Vision Research* 117 (2015), 59–66. https://doi.org/10.1016/j.visres.2015.10.001

Joshua Newn, Eduardo Velloso, Fraser Allison, Yomna Abdelrahman, and Frank Vetere. 2017. Evaluating Real-Time Gaze Representations to Infer Intentions in Competitive Turn-Based Strategy Games. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (Amsterdam, The Netherlands) *(CHI PLAY '17)*. Association for Computing Machinery, New York, NY, USA, 541–552. https://doi.org/10.1145/3116595.3116624

Kevin Pfeil, Eugene M. Taranta, Arun Kulshreshth, Pamela Wisniewski, and Joseph J. LaViola. 2018. A Comparison of Eye-Head Coordination between Virtual and Physical Realities. In *Proceedings of the 15th ACM Symposium on Applied Perception* (Vancouver, British Columbia, Canada) *(SAP '18)*. Association for Computing Machinery, New York, NY, USA, Article 18, 7 pages. https://doi.org/10.1145/3225153.3225157

Thammathip Piumsomboon, Arindam Dey, Barrett Ens, Gun Lee, and Mark Billinghurst. 2017. [POSTER] CoVAR: Mixed-Platform Remote Collaborative Augmented and Virtual Realities System with Shared Collaboration Cues. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. 218–219. https://doi.org/10.1109/ISMAR-Adjunct.2017.72

Thammathip Piumsomboon, Arindam Dey, Barrett Ens, Gun Lee, and Mark Billinghurst. 2019. The Effects of Sharing Awareness Cues in Collaborative Mixed Reality. *Frontiers in Robotics and AI* 6 (2019). https://doi.org/10.3389/frobt.2019.00005

Michael Prilla. 2019. "I simply watched where she was looking at": Coordination in short-term synchronous cooperative mixed reality. *Proceedings of the ACM on Human-Computer Interaction* 3, GROUP (2019). https://doi.org/10.1145/3361127

Rafael Radkowski. 2015. Investigation of visual features for augmented reality assembly assistance. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 9179. Springer,

Cham, 488–498. https://doi.org/10.1007/978-3-319-21067-4{_}50

Ludwig Sidenmark and Hans Gellersen. 2019a. Eye, Head and Torso Coordination During Gaze Shifts in Virtual Reality. *ACM Trans. Comput.-Hum. Interact.* 27, 1, Article 4 (dec 2019), 40 pages. https://doi.org/10.1145/3361218

Ludwig Sidenmark and Hans Gellersen. 2019b. Eye&Head: Synergetic Eye and Head Movement for Gaze Pointing and Selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1161–1174. https://doi.org/10.1145/3332165.3347921

Hamed R. Tavakoli, Esa Rahtu, Juho Kannala, and Ali Borji. 2019. Digging deeper into egocentric gaze prediction. In *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*. Institute of Electrical and Electronics Engineers Inc., 273–282. https://doi.org/10.1109/WACV.2019.00035

Robert J. Teather and Wolfgang Stuerzlinger. 2011. Pointing at 3D targets in a stereo head-tracked virtual environment. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. 87–94. https://doi.org/10.1109/3DUI.2011.5759222

Kentaro Yamada, Yusuke Sugano, Takahiro Okabe, Yoichi Sato, Akihiro Sugimoto, and Kazuo Hiraki. 2011. Attention Prediction in Egocentric Video Using Motion and Visual Saliency. In *Proceedings of the 5th Pacific Rim Conference on Advances in Image and Video Technology - Volume Part I* (Gwangju, South Korea) *(PSIVT'11)*. Springer-Verlag, Berlin, Heidelberg, 277–288. https://doi.org/10.1007/978-3-642-25367-6_25