UnityEyes 2: Open source synthetic eye generation for camera-based eye tracking with machine learning

Alexander D. Smith
ads10@illinois.edu
University of Illinois at Urbana-Champaign, Computer Science
Urbana, Illinois, USA

Brijesh Muthumanickam
University of Illinois at Urbana-Champaign, Autonomy and Robotics
Urbana, USA

Yuanyi Feng
University of Illinois at Urbana-Champaign, Computer Science
Urbana, USA

Wenzhou Ding
University of Illinois at Urbana-Champaign, Mathematics
Urbana, USA

Kris Hauser
University of Illinois at Urbana-Champaign, Computer Science
Urbana, USA

ABSTRACT

This paper introduces UnityEyes 2, an open source and customizable synthetic image generator for training machine learning methods for eye tracking. Training models with synthesized images and ground truth is far more convenient than training from real images, which require manual annotation, but if viewing angles, distances, and lighting conditions do not match real conditions, model performance will degrade (the sim-to-real gap problem). UnityEyes 2 supports customization of distributions of eye pose, camera intrinsic and extrinsic parameters, multi-camera setups, and varied lighting conditions. A graphical user interface enables rapid prototyping of the data distribution and model evaluation. Experiments show that training convolutional neural network models from camera-specific synthetic datasets leads to better transfer to the real world compared to training from generic-viewpoint synthetic datasets.

KEYWORDS

Machine learning, pupil tracking, gaze vector estimation, simulation

ACM Reference Format:
Alexander D. Smith, Brijesh Muthumanickam, Yuanyi Feng, Wenzhou Ding, and Kris Hauser. 2025. UnityEyes 2: Open source synthetic eye generation for camera-based eye tracking with machine learning. In *2025 Symposium on Eye Tracking Research and Applications (ETRA '25), May 26–29, 2025, Tokyo, Japan.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3715669.3726838

1 INTRODUCTION

Despite significant progress in eye tracking, high-accuracy tracking from varying viewpoints, facial appearances, and lighting conditions remains a challenge. Although general-purpose pupil and gaze detection methods do exist, they are typically trained on forward, centered views of the eye. Hence, their performance suffers in applications such as AR glasses, teleconferencing cameras, and automated optometric devices. Capturing datasets of real images and performing manual annotation is one way to customize to a new hardware instantiation, but annotation is expensive and laborious. Moreover, experimenters must strive to sample representative distributions in subject appearance, head and eye movement, blinking, accessories like false eyelashes, and environmental conditions such as lighting and the background.

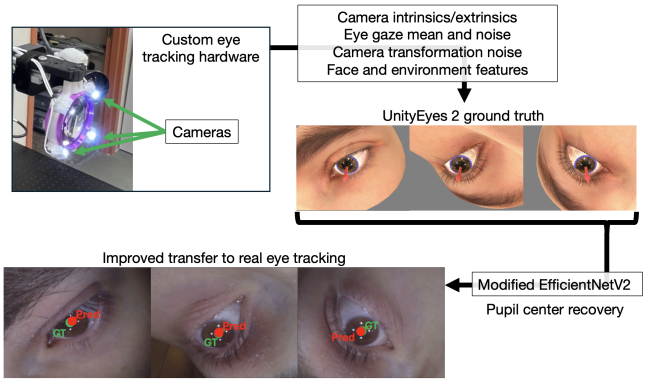

Figure 1: UnityEyes 2 example application, using a robotically-aligned camera array. A custom dataset for eyes must be generated to accurately resolve the 3D pupil center and align our robot robustly. Observe that pupil ground truth (GT) and predictions (Pred) are nearly coincident.

Simulated datasets are a promising approach for training because models can be trained on photorealistic images and precise ground truth annotations, with huge datasets generated at little cost, but doing so faces the so-called "sim-to-real gap" problem. Specifically, such models will degrade under mismatches in facial and ocular appearance, background objects, lighting, camera artifacts, and viewpoint changes. Fine-tuning on annotated real data is one approach to this problem, but we argue that the first line of attack should be better aligning simulators to reality.

UnityEyes 2 is a software package in development that supports greater customization in camera hardware choice and appearance

ETRA '25, May 26–29, 2025, Tokyo, Japan

ACM ISBN 979-8-4007-1487-0/2025/05.
https://doi.org/10.1145/3715669.3726838

variation than prior packages. Specifically, given the calibrated intrinsics and extrinsics of one or more (real world) cameras, it generates realistic images that closely match images generated by real cameras. Eye pose, lighting conditions, and subject appearance variations can also be specified. We explore an application of this method to a robotic eye alignment system, which uses a custom camera configuration to resolve the pupil position (Figure 1). Due to its varied position and oblique angles to the subject's eye, existing methods trained on frontal, centered views degrade in eye tracking performance. We show that using synthetic data from UnityEyes 2, we can train models from synthetic data with high accuracy in the real world.

Ethics statement: This work describes a method to leverage synthetic data generation, which faces fewer ethical concerns than collecting data from humans. Synthesizing diverse appearances can also represent more diverse populations than small studies, which is a potential way to mitigate algorithmic bias.

2 RELATED WORK

Eye tracking has seen a growth in open-source image datasets that can be used to evaluate and train accurate models [Egawa and Shirayama 2012; Garbin et al. 2019; Krafka et al. 2016; Porta et al. 2019; Sugano et al. 2013, 2014; Swirski et al. 2012; Tonsen et al. 2016; Wood et al. 2015, 2016; Wu et al. 2020; Zhang et al. 2020, 2015a,b]. However, researchers developing specialized eye tracking hardware will face several forms of data distribution mismatch when applying these datasets to their hardware:

(1) **Relative positioning:** relative transformation (translation and rotation) of the eye relative to the camera.
(2) **Eye Pose Variation:** distribution of eye positions (pitch and yaw) and movements.
(3) **Camera intrinsics:** the intrinsic parameters of the camera, such as field of view, resolution, and frame rate.
(4) **Multiple cameras:** whether multiple views can be generated for the same scene for multi-camera tracking setups.
(5) **Facial Appearance:** the face characteristics, including facial prominences, blink state, scale of features, skin tone, scars, tattoos, accessories, and others.
(6) **Environment appearance:** the locations and parameters of the environment, including light sources, surrounding scene, and light spectrum (such as visible light or infrared).

Moreover, synthesizers should be able to generate datasets rapidly from a specified distribution of the parameters above, without manual intervention, a characteristic we call **automatic generation**.

Our work builds on the UnityEyes package, which enables generation of high-fidelity and easily configurable simulated eye gaze datasets [Wood et al. 2016]. UnityEyes provides a generation interface that enables users to rapidly create datasets on 18 different environments and 20 different face meshes by adjusting the pitch and yaw of one camera, and adjusting the pitch and yaw of the eye. All other scene configurations, such as pupil constriction, lighting, and distributions of noise are decisions made for the user. Our work enables the user to make decisions about these configurations, expanding the dataset generation capability of UnityEyes dramatically.

3 METHODS

UnityEyes 2 enables a user to automatically generate a dataset with a specified distribution over eye pose, camera parameters, and scene lighting. It also supports multi-camera setups, generating associated images, pupil, and gaze outputs for all cameras viewing a single scene. Features in development include the ability to customize the distributions of face and environment parameters. We will also implement user-specified custom faces and environments.

UnityEyes 2 is based on the original UnityEyes source [Wood et al. 2016]. Users specify a desired set of parameters and their distributions using a front-end user interface in the Unity application, or in a JSON interface. Camera extrinsics include a mean relative pose (translation and rotation) of the camera with respect to the eye center, as well as the distribution about the mean for which poses are sampled. Camera intrinsics include the image width w and height h, focal lengths f_x, f_y, and the image center c_x, c_y, all given in units of pixels. These parameters can be calibrated from hardware using standard toolboxes [Zhang 2000].

UnityEyes 2 is available as an open-source tool on GitHub. We also provide executable releases for Windows, Linux, and MacOS.

The outputs of our data generator include randomly generated eye images from a user's specification, ground-truth locations of the pupil center, a normalized vector representing the optical axis, the center of the globe of the eye, and the 2D parameters provided previously by UnityEyes, all with respect to the camera center. Our tests show that for an M3 Max MacBook Pro, UnityEyes 2 generates 85.7 images per second. UnityEyes generates 82.0 images per second, and U2Eyes generates 1.3 images per second.

4 IMPACT ON LEARNING

We use EfficientNetV2 [Tan and Le 2021] for pupil center and gaze vector estimation with our hardware shown in Figure 1, training on 10,000 samples from both the original UnityEyes data generator [Wood et al. 2016] and for each camera on a dataset generated by UnityEyes 2.

Input images are resized to 224×224, and the final fully connected layer of EfficientNetV2 is modified to output a 2D vector representing the pupil center in the resized image space. The model is trained by optimizing MSE loss with an Adam optimizer [Kingma and Ba 2017] using a learning rate of 1×10^{-3} and weight decay of 1×10^{-4} The models are trained for 50 epochs, with early stopping if no improvement in testing performance is observed for 10 consecutive epochs. The predicted pupil position is transformed back into the original image space for comparison.

The results are shown in Table 1. Using mean squared pixel error, the models are compared to a hand-labeled 263 frame dataset with three 640×480 camera views. These results indicate that configuring a custom dataset specific to hardware increases accuracy of transfer to real eyes.

5 CONCLUSIONS AND FUTURE WORK

With UnityEyes 2, we show that a simulated data generator for eye-feature prediction can be directly applied to eye tracking on novel hardware configurations. We believe that our comprehensive, fully-configurable eye tracking and gaze estimation data generator will be broadly beneficial to the eye tracking research community. Our

Table 1: Mean squared pixel error from ground truth comparing UnityEyes and UnityEyes 2 on custom hardware

Model	Cam1	Cam2	Cam3	Overall
UnityEyes [Wood et al. 2016]	62.542± 65.569	10.129± 4.882	19.603± 9.464	30.758± 38.352
UnityEyes 2 (Ours)	10.699± 7.124	5.976 ± 3.110	5.501 ± 3.625	7.392 ± 4.952

* all intervals reported as mean ± one standard deviation in pixels

model-based approach is a simple demonstration that UnityEyes 2 is valuable for real-world applications, and we leave optimal strategies for resolving gaze vectors from UnityEyes 2 datasets to future exploration.

Future work will explore environment augmentations, improved human appearance, a Python API, predictive model improvements, and application of this work to 3D gaze vector prediction and tracking in robotic eye examinations, VR and AR headsets, and video conferencing cameras.

ACKNOWLEDGMENTS

This work is supported by the NIH National Robotics Initiative under award number NIH R01 EY035106 A. We thank Erroll Wood for sharing his work and insights on the original UnityEyes project.

REFERENCES

Akira Egawa and Susumu Shirayama. 2012. A method to construct an importance map of an image using the saliency map model and eye movement analysis. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 21–28. https://doi.org/10.1145/2168556.2168559

Stephan J. Garbin, Yiru Shen, Immo Schuetz, Robert Cavin, Gregory Hughes, and Sachin S. Talathi. 2019. OpenEDS: Open Eye Dataset. *CoRR* abs/1905.03702 (2019). arXiv:1905.03702 http://arxiv.org/abs/1905.03702

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG] https://arxiv.org/abs/1412.6980

Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye Tracking for Everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2176–2184. https://doi.org/10.1109/CVPR.2016.239

Sonia Porta, Benoît Bossavit, Rafael Cabeza, Andoni Larumbe-Bergera, Gonzalo Garde, and Arantxa Villanueva. 2019. U2Eyes: A Binocular Dataset for Eye Tracking and Gaze Estimation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 3660–3664. https://doi.org/10.1109/ICCVW.2019.00451 ISSN: 2473-9944.

Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. 2013. Graph-based joint clustering of fixations and visual entities. *ACM Trans. Appl. Percept.* 10, 2, Article 10 (June 2013), 16 pages. https://doi.org/10.1145/2465780.2465784

Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. 2014. Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 1821–1828. https://doi.org/10.1109/CVPR.2014.235

Lech Swirski, Andreas Bulling, and Neil Dodgson. 2012. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 173–176. https://doi.org/10.1145/2168556.2168585

Mingxing Tan and Quoc V. Le. 2021. EfficientNetV2: Smaller Models and Faster Training. arXiv:2104.00298 [cs.CV] https://arxiv.org/abs/2104.00298

Marc Tonsen, Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 2016. Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research and; Applications (ETRA '16)*. ACM. https://doi.org/10.1145/2857491.2857520

Erroll Wood, Tadas Baltruaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. 2015. Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 3756–3764. https://doi.org/10.1109/ICCV.2015.428

Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. 2016. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. Association for Computing Machinery, New York, NY, USA, 131–138. https://doi.org/10.1145/2857491.2857492

Zhengyang Wu, Srivignesh Rajendran, Tarrence van As, Joelle Zimmermann, Vijay Badrinarayanan, and Andrew Rabinovich. 2020. MagicEyes: A Large Scale Eye Gaze Estimation Dataset for Mixed Reality. *CoRR* abs/2003.08806 (2020). arXiv:2003.08806 https://arxiv.org/abs/2003.08806

Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. 2020. ETH-XGaze: A Large Scale Dataset for Gaze Estimation under Extreme Head Pose and Gaze Variation. *CoRR* abs/2007.15837 (2020). arXiv:2007.15837 https://arxiv.org/abs/2007.15837

Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2015a. Appearance-based Gaze Estimation in the Wild. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4511–4520.

Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2015b. Appearance-Based Gaze Estimation in the Wild. *CoRR* abs/1504.02863 (2015). arXiv:1504.02863 http://arxiv.org/abs/1504.02863

Z. Zhang. 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 11 (Nov. 2000), 1330–1334. https://doi.org/10.1109/34.888718 Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.