

¹ EyeIdentify3D: A Python package for gaze behavior classification of mobile eye-tracking data

³ **Eve Charbonneau**  ^{1,2}¶, **Thomas Romeas**  ^{1,2}, and **Maxime Trempe**  ³

⁴ 1 Université de Montréal, Canada 2 Institut national du sport du Québec, Canada 3 Bishop's University,
⁵ Canada ¶ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Open Journals](#) ↗

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).¹⁹²⁰

⁶ Summary

⁷ With the technological advances of mobile eye-tracking technologies, researchers can now place ⁸ participants in real-world settings (or in virtual environments that simulate the real world) and ⁹ measure their head and eye orientation to get the gaze orientation in 3D space. This raw ¹⁰ gaze orientation is hardly interpretable and must be post-processed to extract gaze behaviors ¹¹ (e.g., fixations, saccades, smooth pursuits, visual scanning). However, most open-source ¹² gaze classification algorithms were developed for screen-based eye-tracking, where data is ¹³ recorded on a 2D plane and the participant's head is kept still. These algorithms are ill-suited ¹⁴ for real-world mobile eye-tracking data (360°) as the gaze-vector origin and endpoint can ¹⁵ move substantially due to head rotations, a challenge also present in head-mounted display ¹⁶ systems used in virtual reality research. Additionally, eye-tracking researchers often rely on ¹⁷ study-specific analysis pipelines, which leads to methodological discrepancies that impede ¹⁸ cross-study comparison and the interpretation of results, ultimately limiting our understanding ¹⁹ of gaze behavior-related phenomena. To address this gap, we developed EyeIdentify3D, an ²⁰ automated and modulable pipeline for analyzing 360° eye-tracking data.

²¹ Statement of need

²² To address the need for automating the identification of gaze behaviors from eye-tracking ²³ data, a few open-source packages have been developed over the years. Most of them have ²⁴ focused primarily on the identification of fixations, either from fixed-screen eye-tracker data ²⁵ ([Krassanakis et al., 2014](#)) or mobile eye-tracker data [Munn & Pelz \(2009\)](#). Some have extended ²⁶ their identification capabilities to include other behaviors such as saccades, blinks, and micro- ²⁷ saccades, although these have remained limited to fixed-screen eye-trackers ([Ghose et al., 2020](#), ²⁸ [© berger:2012](#)). Notably, none of the existing packages have included the identification of ²⁹ gaze behaviors in dynamic environments that involve large eye and head movements, such as ³⁰ smooth pursuit and visual scanning.

³¹ EyeIdentify3D is a Python package for identifying multiple gaze behaviors (blinks, fixations, ³² saccades, smooth pursuits, visual scannings) from mobile eye-tracking data. It was designed ³³ to: 1. Interpret data from various mobile eye-tracking systems (e.g., Pupil Invisible), including ³⁴ those embedded in head-mounted displays (e.g., HTC Vive Pro, Pico Neo 3 Pro Eye). 2. ³⁵ Provide a simple user interface, where only a few lines of code are needed to identify the desired ³⁶ gaze behaviors and extract related metrics. 3. Enable visual inspection of the classification ³⁷ results.

³⁸ EyeIdentify3D was designed to be used in science and human performance analysis. It has ³⁹ already been used in sport psychology to analyze the gaze behavior of basketball players ⁴⁰ ([Trempe et al., 2025](#)), and was used in pilot studies trampolinists and boxers. Our objective ⁴¹ is to distribute the toolbox openly to help researchers more reliably identify and analyze

⁴² gaze behaviors in real-world scenarios, which involve movements of the head, and promote
⁴³ standardization in gaze analysis, thereby improving our understanding of visual strategies.

⁴⁴ Gaze behavior identification

⁴⁵ For each trial recorded during an experiment, the eyes and head rotations are extracted from
⁴⁶ the data collected by the eye-tracker and the inertial measurement unit, respectively. The gaze
⁴⁷ orientation (head and eye rotations combined) expressed over a 360° range is then analyzed
⁴⁸ frame-by-frame. For each frame, the pipeline applies a step-by-step classification based on the
⁴⁹ following criteria: 1. **Invalid**: The eye-tracker has declared having low confidence in the gaze
⁵⁰ orientation measurement and considers the data invalid. This often happens when the eyes are
⁵¹ closed (e.g., during a blink), the eye orientation is outside the eye-tracker's measurement range,
⁵² or if the eye-tracker was not positioned properly on the participant. 2. **Blink**: The eye openness
⁵³ is below the user defined threshold (Chen & Hou, 2021). 3. **Saccade**: Two criteria must be
⁵⁴ met to detect a saccade. 1) The eye movement must be faster than a dynamical threshold
⁵⁵ determined using a rolling median over a user defined window size. 2) The eye movement
⁵⁶ acceleration must exceed a user defined threshold for a user defined number of frames. This
⁵⁷ ensures that the eyes are moving rapidly between two targets, accelerating as they leave the
⁵⁸ first target and decelerating as they approach the second target (Van Opstal & Van Gisbergen,
⁵⁹ 1987). 4. **Visual scanning**: The gaze (head + eyes) velocity is larger than a user defined
⁶⁰ threshold (McGuckian et al., 2020). Visual scanning should be identified after saccades as
⁶¹ visual scanning behaviors could also present high eye velocity. 5. **Inter-saccadic interval**: Our
⁶² inter-saccadic interval classification was adapted from Larsson et al. (2015) implementation
⁶³ designed for screen-based eye-tracking data by replacing cartesian coordinates (2D plane) with
⁶⁴ spherical coordinates (360° range of motion). The frames that remained unidentified after
⁶⁵ the previous steps are grouped into intervals. The intervals lasting longer than a user defined
⁶⁶ duration threshold, are considered inter-saccadic intervals. These intervals are subdivided into
⁶⁷ windows of a user defined size. Each window is classified as either coherent or incoherent
⁶⁸ based on the gaze movement (moving in a consistent direction or not). Adjacent coherent
⁶⁹ and incoherent windows are merged together to form segments. Then, these segments are
⁷⁰ further classified as either 6. **fixation** or **smooth pursuit** behaviors based on the four criteria
⁷¹ described in Larsson et al. (2015): - Dispersion: $p_D < \eta_D$ - Consistent direction: $p_{CD} > \eta_{CD}$
⁷² - Positional displacement: $p_{PD} > \eta_{PD}$ - Spatial range: $p_R > \eta_{maxFix}$

⁷³ All behaviors are mutually exclusive (except for invalid and blink that can happen simultane-
⁷⁴ ously). For example, a frame cannot be classified as both a visual scanning and a smooth
⁷⁵ pursuit. Thus, the order of the identification is important as the first behavior identified will
⁷⁶ take precedence over the others.

⁷⁷ More details on the definition of events and how they are identified can be found in the
⁷⁸ documentation.

⁷⁹ Finally, EyeDentify3D enables visualisation of the classified gaze data and extraction/export
⁸⁰ of metrics related to the behaviors (e.g., mean duration, time ratio spent in each behavior,
⁸¹ number of occurrences, saccade amplitude, smooth pursuit trajectory length, etc.).

⁸² Note on the implementation

⁸³ We believe that the choices made in EyeDentify3D are the most suitable for the analysis
⁸⁴ of gaze behavior in 3D space (especially in sporting context). However, we are very open
⁸⁵ to implement other identification methods that might be more suitable in other application
⁸⁶ contexts.

87 Acknowledgements

88 This project was supported by a Research and Creative Activity grant from Bishop's University.

89 Conflict of interest

90 The authors declare no conflict of interest.

91 Declaration of generative AI

92 During the preparation of this work the developer used ChatGPT, Claude, and Copilot to speed
93 up development and enhance code clarity. Aider and Claude were also used to write tests.
94 After using these tools/services, the developer reviewed and edited the content as needed and
95 takes full responsibility for the content of the repository.

96 References

- 97 Chen, X., & Hou, W. (2021). Visual fatigue assessment model based on eye-related data in
98 virtual reality. *2021 IEEE 7th International Conference on Virtual Reality (ICVR)*, 262–268.
99 <https://doi.org/10.1109/ICVR51878.2021.9483841>
- 100 Ghose, U., Srinivasan, A. A., Boyce, W. P., Xu, H., & Chng, E. S. (2020). PyTrack: An
101 end-to-end analysis toolkit for eye tracking. *Behavior Research Methods*, 52(6), 2588–2603.
102 <https://doi.org/10.3758/s13428-020-01392-6>
- 103 Krassanakis, V., Filippakopoulou, V., & Nakos, B. (2014). EyeMMV toolbox: An eye movement
104 post-analysis tool based on a two-step spatial dispersion threshold for fixation identification.
105 *Journal of Eye Movement Research*, 7(1). <https://doi.org/10.16910/jemr.7.1.1>
- 106 Larsson, L., Nyström, M., Andersson, R., & Stridh, M. (2015). Detection of fixations and
107 smooth pursuit movements in high-speed eye-tracking data. *Biomedical Signal Processing
and Control*, 18, 145–152. <https://doi.org/10.1016/j.bspc.2014.12.008>
- 109 McGuckian, T. B., Beavan, A., Mayer, J., Chalkley, D., & Pepping, G. (2020). The association
110 between visual exploration and passing performance in high-level U13 and U23 football
111 players. *Science and Medicine in Football*, 4(4), 278–284. <https://doi.org/10.1080/24733938.2020.1769174>
- 113 Munn, S. M., & Pelz, J. B. (2009). FixTag: An algorithm for identifying and tagging fixations
114 to simplify the analysis of data collected by portable eye trackers. *ACM Transactions on
115 Applied Perception (TAP)*, 6(3), 1–25. <https://doi.org/10.1145/1577755.1577759>
- 116 Trempe, M., Charbonneau, E., Lévesque, V., Ba, A., & Romeas, T. (2025). Expanding
117 the view: Differences in gaze behavior between immersive 360° videos and traditional
118 fixed-screen videos. *Under Review*.
- 119 Van Opstal, A. J., & Van Gisbergen, J. A. M. (1987). Skewness of saccadic velocity profiles:
120 A unifying parameter for normal and slow saccades. *Vision Research*, 27(5), 731–745.
121 [https://doi.org/10.1016/0042-6989\(87\)90071-X](https://doi.org/10.1016/0042-6989(87)90071-X)
- 122 West, J. M., Haake, A. R., Rozanski, E. P., & Karn, K. S. (2006). eyePatterns: Software for
123 identifying patterns and similarities across fixation sequences. *Proceedings of the 2006
124 Symposium on Eye Tracking Research & Applications*, 149–154. <https://doi.org/10.1145/1117309.1117360>