Human Interaction and Emerging Technologies (IHIET-FS 2026), Vol. 235, 2026, 110–120
https://doi.org/10.54941/ahfe1007266

BEHOLD: An Extensible Eye-Tracking Infrastructure Supporting Multimodal, Multi-Device Interaction Evaluation

Hugo Correia[1], Bernardo Marques[1], Liliana Vale Costa[2], and Samuel Silva[1]

[1]IEETA / DETI, University of Aveiro, Aveiro, Portugal
[2]Digimedia / DECA, University of Aveiro, Aveiro, Portugal

ABSTRACT

Interactive systems increasingly span multiple devices, modalities, and physical spaces, which makes interaction evaluation harder than in single-device settings. Traditional post-task methods (e.g., questionnaires and logs) often miss when and why problems occur. Eye tracking can complement these approaches by continuously capturing visual attention, helping reveal what users notice or overlook and how effort is distributed across interfaces. However, deploying eye tracking in such ecosystems raises challenges: different contexts require different trackers (wearables, environmental cameras, or display-mounted sensors); existing tools are often tied to specific hardware and may lack evaluation-oriented analysis; and synchronising gaze with other contextual and system-level data typically requires multiple disconnected components. This work presents the first stage of BEHOLD (Biometric Eye-tracking Hub for Observation, Logging, and Data), a plugin-based proof of concept that separates device-specific logic from a unified processing and analysis pipeline. Each eye tracker is supported via an individual plugin for parsing and validation, while the framework treats gaze streams consistently regardless of source. For storage, BEHOLD combines TimescaleDB for high-frequency gaze samples, PostgreSQL for session metadata, and MinIO for recordings and exports. We demonstrate the approach by integrating data from Tobii Pro Glasses, showing how the plugin architecture accommodates device-specific requirements while enabling a coherent workflow from acquisition to analysis.

Keywords: Eye tracking, Interactive ecosystems, Hardware-agnostic framework, Multimodal evaluation

INTRODUCTION

Interactive ecosystems, characterised by distributed touchpoints, heterogeneous devices, and multimodal interaction channels, represent a growing paradigm where users engage with interconnected systems spanning physical and digital boundaries (Streitz, 2018; Weiser, 1999). Evaluating user experience within these environments is challenging, as interaction unfolds across multiple devices and contexts rather than being confined to a single workstation (Shin, 2017). Traditional approaches relying on self-report questionnaires, think-aloud protocols, and interaction logging

Received April 9, 2026; Revised April 23, 2026; Accepted May 13, 2026; Available online June 22, 2026

capture important dimensions but struggle to reveal the moment-to-moment attentional dynamics driving behaviour in complex environments (Jacob & Karn, 2003; Nyström & Holmqvist, 2010).

Eye tracking offers a complementary evaluation lens by providing continuous, objective measures of where users direct their visual attention, how they allocate cognitive resources across competing information sources, and how their gaze patterns shift as they navigate multimodal interfaces (*Eye Tracking Methodology*, 2007; Just & Carpenter, 1976). Metrics such as fixation duration, saccadic amplitude, pupil dilation, and gaze transition entropy have been shown to correlate with cognitive load, task difficulty, and information processing strategies across diverse application domains, including web usability, assistive technology, industrial control, and virtual reality (Holmqvist & Andersson, 2017; Jacob & Karn, 2003; Krejtz et al., 2018).

Despite the analytical value of gaze data, integrating eye tracking into evaluation workflows for interactive ecosystems remains technically demanding. Three structural limitations in the current platform landscape hinder broader adoption. First, commercial platforms such as Tobii Pro Lab (*Find out More about the Latest Features in Tobii Pro Lab*) and SR Research Data Viewer are tightly coupled to their respective hardware ecosystems, creating vendor lock-in that constrains device selection and prevents cross-device comparison within a single study (Niehorster et al., 2025). Second, the separation between data collection and analysis forces practitioners to navigate fragmented toolchains where extensible frameworks like OpenIris (Sadeghi et al., 2024) offer custom hardware integration but lack built-in analytical capabilities, while comprehensive analysis platforms remain architecturally closed (Niehorster et al., 2025). Third, most existing tools optimise for controlled laboratory settings and do not adequately support the distributed, ecologically valid deployments increasingly required for interactive ecosystem evaluation (Abeysinghe et al., 2023; Lappi, 2015).

This paper presents BEHOLD (Biometric Eye-tracking Hub for Observation, Logging, and Data), a plugin-based framework that addresses these three gaps through a hardware-agnostic architecture for eye tracking data integration, processing, and analysis. The framework establishes abstract plugin interfaces that decouple the processing pipeline from specific eye tracking devices, and supports both post-hoc file analysis and real-time streaming through a unified pipeline (Kothe et al., 2025). Requirements for the framework were derived from consultation with domain experts involved in interactive system development, particularly in the context of smart home interactive technology evaluation involving partnerships with companies such as Maxiplás and Bosch.

Background and Context

Eye tracking for interaction evaluation rests on the eye-mind hypothesis (Just & Carpenter, 1976): gaze direction reflects ongoing cognitive processing, making it a continuous, non-intrusive window into user attention and effort. Event detection algorithms classify fixations and saccades from raw gaze streams, and derived metrics such as gaze transition entropy characterise scanning strategies and cognitive states (Krejtz et al., 2018). Visualisation techniques have progressed from heatmaps and scanpath overlays to real-time dashboards computing advanced metrics during ongoing experiments (Jayawardena, 2020; Jayawardena et al., 2024), and distributed systems such as A-DisETrac (Abeysinghe et al., 2023) extend these capabilities to multi-user gaze aggregation across networked participants.

Despite this maturity in individual techniques, the platform landscape remains fragmented. Commercial solutions such as Tobii Pro Lab provide well-validated workflows but are restricted to proprietary hardware (*Find out More about the Latest Features in Tobii Pro Lab*), while iMotions Lab offers broader device support through LSL integration (Kothe et al., 2025) at significant cost. On the open-source side, PyGaze (Dalmaijer et al., 2014) abstracts device differences for experiment scripting but delegates analysis to external tools, OpenIris (Sadeghi et al., 2024) provides a plugin architecture for custom hardware without integrated analytical capabilities, and PyTrack (Ghose et al., 2020) bridges manufacturer format differences for offline analysis, but lacks real-time support.

This landscape reveals three structural gaps, illustrated in Table 1, that compound when practitioners attempt to incorporate gaze data into interactive ecosystem evaluation. The first is hardware dependency: commercial platforms lock users into specific vendor ecosystems and open-source alternatives each cover limited device subsets, preventing cross-device comparison within unified workflows. The second is the separation between collection and analysis, where extensible acquisition frameworks lack analytical pipelines and comprehensive analysis platforms remain architecturally closed. The third is laboratory-centrism, as most tools optimise for controlled settings and inadequately support the distributed, ecologically valid deployments that interactive ecosystem evaluation increasingly requires (Lappi, 2015; Niehorster et al., 2025).

Table 1: Schematic view linking the three motivating gaps to the research questions they create.

Gap (What Breaks Today)	Motivating Question → BEHOLD Response
G1: Hardware dependency	*How can heterogeneous eye trackers be used in one workflow?* → Device-agnostic plugin interface that normalizes gaze streams while preserving device-specific fields.
G2: Collection–analysis separation	*How can ingestion, validation, and analysis be performed end-to-end without a fragmented toolchain?* → Unified four-stage pipeline (Extract, Validate, Analyse, Store) for both post-hoc and real-time modes.
G3: Laboratory-centrism	*How can gaze be evaluated in ecologically valid, distributed deployments with other modalities?* → LSL-based multimodal synchronization and architecture designed for distributed, multi-device settings.

These gaps create concrete barriers in practice. A researcher studying gaze behaviour in a smart home, for instance, may need to combine data from wearable glasses with screen-based trackers while synchronising gaze streams with environmental sensors and system logs. No current platform supports this workflow end-to-end without substantial custom integration. The question motivating this work is therefore how eye tracking data from heterogeneous devices can be integrated into a unified, extensible framework that supports both real-time monitoring and post-hoc analysis without imposing hardware vendor lock-in.

REQUIREMENT ANALYSIS

Requirements for BEHOLD were derived through a combination of the literature gap analysis presented in Section Background and Context, consultation with domain experts involved in interactive system development at IEETA (particularly in the context of smart home and assistive technology evaluation), and a user-centred design approach grounding elicitation in representative evaluation scenarios from those domains.

Two personas capture the primary user profiles. P1 represents an HCI researcher conducting multimodal studies who needs device-agnostic data import, quality assessment, AOI-based analysis, and multimodal temporal alignment within a single workspace. P2 represents a product manager overseeing usability evaluation who needs real-time monitoring, cross-participant comparison, and exportable reports accessible to stakeholders without eye-tracking expertise. From these personas, nine user stories were elicited, spanning the full evaluation workflow, from data ingestion and quality reporting through analysis and visualisation to standardised export.

The user stories were translated into ten high-level functional requirements and four non-functional requirements, summarised in Table 2. The functional requirements cover the data lifecycle from ingestion (FR-01, FR-02) through validation (FR-03), analysis (FR-05 through FR-07), visualisation (FR-08), and export (FR-09), with multimodal synchronisation as a cross-cutting concern (FR-10). The non-functional requirements establish the key architectural constraints: extensibility without modifying core code (NFR-01) and hardware-independent processing logic (NFR-02).

Table 2: Summary of core functional and non-functional requirements.

ID	Description
FR-01	Support data ingestion from multiple eye tracking devices through a plugin interface
FR-02	Operate in both post-hoc (file import) and real-time (streaming) modes
FR-03	Validate incoming gaze data and generate quality metrics (signal loss, spatial accuracy)
FR-05	Detect gaze events (fixations, saccades, blinks) using configurable algorithms
FR-06	Compute standard metrics (fixation count/duration, saccadic amplitude) and advanced metrics (coefficient K, gaze transition entropy)
FR-07	Support AOI definition and compute AOI-based metrics (dwell time, entry count, transitions)

(Continued)

Table 2: (Continued)

ID	Description
FR-08	Generate visualizations (heatmaps, scanpaths, timelines) for both 2D and 3D contexts
FR-09	Export processed data and results in standardized formats
FR-10	Synchronize gaze data with external modalities using LSL timestamps
NFR-01	Extensibility: new device plugins can be added without modifying core framework code
NFR-02	Hardware independence: core processing is agnostic to the originating eye tracker
NFR-03	Usability: interface accessible to practitioners with varying technical backgrounds
NFR-05	Analytical accuracy: event detection and metric computation validated against established tools

SYSTEM ARCHITECTURE

BEHOLD's architecture is organised around a plugin-based device abstraction layer and a four-stage processing pipeline, supported by a polyglot storage strategy and real-time synchronisation infrastructure. Figure 1 provides an overview of the system, showing how the plugin layer, core platform, data layer, streaming bridge, and frontend components interact across both development phases. The following subsections describe each architectural component.

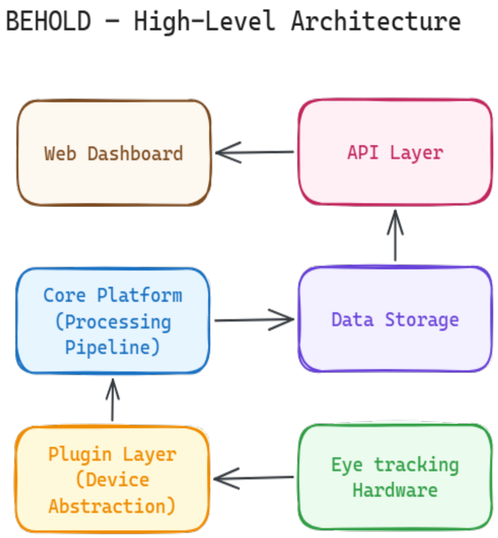

Figure 1: High-level architecture of BEHOLD. Eye-tracking hardware feeds device-specific data into the Plugin Layer, which abstracts vendor-specific formats into a standardised representation. The core platform processes this data through a four-stage pipeline (extract, validate, analyse, store), persisting results in the data storage layer. The API layer exposes framework capabilities to the web dashboard for interactive visualisation and to external tools for programmatic access.

Plugin Architecture and Device Abstraction

The central design principle is the separation of device-specific concerns from core processing logic through an abstract plugin interface. Each plugin transforms hardware-native data, regardless of manufacturer format, sampling rate, or coordinate system, into a standardised representation containing timestamps, 2D gaze coordinates, pupil diameter, and a validity indicator, enabling all downstream pipeline stages to operate identically regardless of the originating device (NFR-02). Device-specific extended fields remain accessible through an optional metadata dictionary, ensuring that hardware-unique capabilities such as 3D gaze vectors or scene camera frames do not pollute the core data model. New plugins can be registered without modifying any framework code (NFR-01), in accordance with the open/closed principle. The initial implementation targets Tobii Pro Glasses as a proof of concept, with the interface design informed by analysis of data formats from Tobii, HTC Vive Pro Eye (via the SRanipal SDK), Pupil Labs, and webcam-based systems to ensure the abstraction generalises across representative device categories.

Processing Pipeline

The processing pipeline, detailed in Figure 2, consists of four sequential stages, each implemented as a modular component that can be configured or replaced independently.

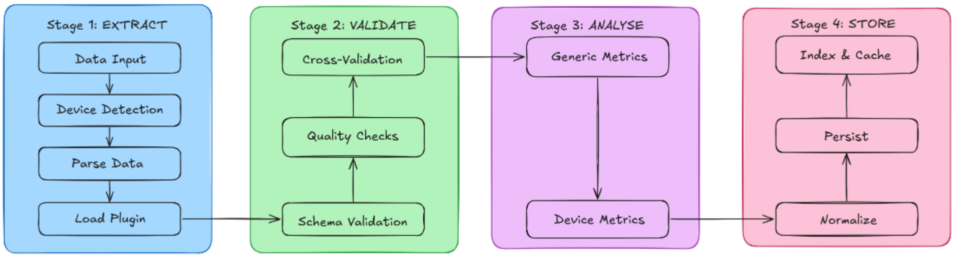

Figure 2: The four-stage processing pipeline. Each stage operates on the standardized output of the previous one, ensuring that device-specific concerns are fully resolved during extraction and all subsequent processing remains hardware-agnostic.

The *Extract* stage ingests raw gaze data from device plugins, either by parsing exported files (post-hoc mode) or consuming LSL inlet streams (real-time mode), normalising timestamps to a common clock reference and tagging each sample with its source device identifier. The *Validate* stage then assesses data quality through schema validation, completeness checks, and device-specific cross-validation rules, computing signal loss rates and temporal continuity metrics per segment (FR-03). In the *Analyse* stage, validated data passes through configurable analysis chains that compute both generic metrics (fixation and saccade detection via I-DT or I-VT, AOI dwell times and transition matrices, validity rates) and device-specific metrics (head pose for VR trackers, eye openness, 3D gaze vectors), addressing FR-05 through FR-07. Finally, the *Store* stage normalises results to a standard schema and persists them using a polyglot storage backend suited to the different characteristics of each data type: time-series gaze samples, session metadata, and large binary artefacts such as scene camera recordings.

Multimodal Synchronization

Temporal alignment of gaze data with other interaction modalities is achieved through Lab Streaming Layer (LSL) (Kothe et al., 2025), which provides software-based timestamping with sub-millisecond accuracy on local area networks. Each device plugin that supports real-time streaming publishes data as an LSL outlet, and the framework's ingestion layer consumes these streams alongside any other LSL-compatible data sources (EEG, GSR, event markers, system logs). The Extensible Data Format (XDF) serves as the archival format for synchronised multimodal recordings.

Visualisation and Interface

The framework provides a web-based dashboard implemented with Next.js, offering session management, real-time monitoring views with live heatmap and metric overlays, and post-hoc analysis workspaces for detailed exploration of recorded data. REST API endpoints expose all framework capabilities for programmatic access and integration with external analysis tools such as R or Python notebooks, supporting FR-08 and FR-09.

DEVELOPMENT

To validate the architectural decisions described in the previous section, an initial implementation targeting Tobii Pro Glasses 2 was developed. This section describes the device plugin, the processing results, and the web-based dashboard that together demonstrate the framework's feasibility across the full data lifecycle.

Tobii Plugin Implementation

The Tobii plugin ingests native recording files, extracting binocular gaze coordinates, pupil diameter, and inertial data, normalising per-eye representations into combined gaze points and mapping manufacturer-specific validity codes to the framework's standardised confidence scale. For event detection, the pipeline adopts the I-VT (Velocity-Threshold Identification) algorithm, consistent with Tobii Pro Lab's own validated fixation filter (Olsen & Matos, 2012), ensuring that results produced through BEHOLD remain directly comparable with those from the manufacturer's established toolchain. A study comprising 29 participants and 29 recording sessions was processed through the pipeline, yielding 347,752 total gaze samples, of which 321,921 (92.2%) were classified as valid, a quality level the framework labels as "Excellent." The average session duration was 4.0 minutes at a mean sampling rate of 50 Hz, with per-session validity rates of 86% to 97%.

Web Dashboard

The framework provides a web-based dashboard implemented in Next.js, offering coordinated views for study management and gaze analysis. As shown in Figure 3, the interface supports the full post-hoc evaluation workflow: practitioners can monitor study-level metrics and data quality (Figure 3a), inspect individual recording sessions (Figure 3b), compare attention patterns across participants (Figure 3c), and explore gaze distribution through interactive heatmap visualisations overlaid on scene camera footage (Figure 3d). These views collectively address the visualisation and export requirements (FR-08, FR-09) while remaining accessible to domain specialists without eye-tracking expertise (US-07).

Additional analysis features were integrated into the dashboard, including a temporal gaze patterns view (Figure 4) that plots normalised X and Y coordinates over time, enabling practitioners to inspect raw gaze dynamics and identify fixation periods, saccadic transitions, or signal loss before aggregate analysis.

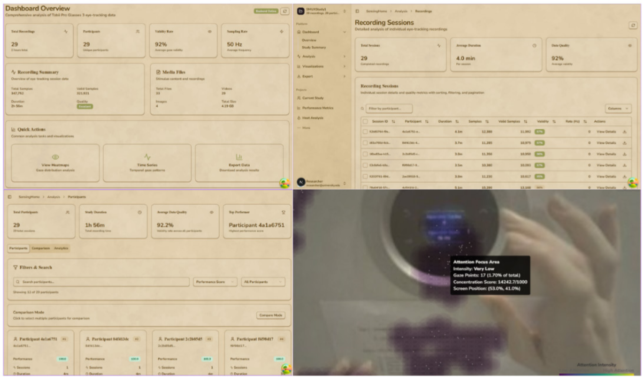

Figure 3: BEHOLD dashboard interface showing: (a) the main overview with key study metrics and quick access to analysis tools; (b) the recording sessions view with sorting, filtering, and session quality indicators; (c) participant analysis enabling cross-participant comparison and performance scoring; and (d) a visual attention heatmap overlaid on scene camera footage with interactive gaze metrics.

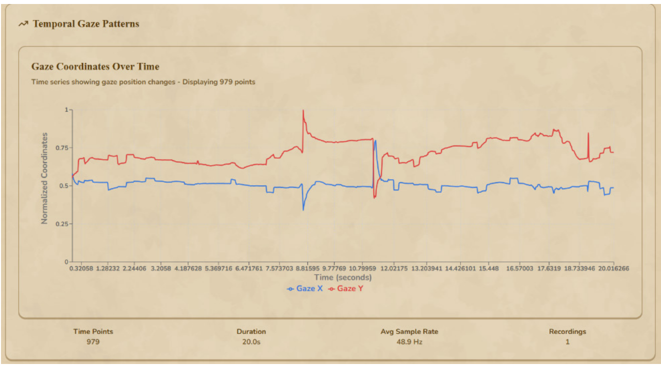

Figure 4: Temporal gaze patterns view displaying normalised gaze X and Y coordinates over a 20-second segment of a single recording. The plot reveals moment-to-moment gaze dynamics, including stable fixation periods and rapid position changes indicative of saccades.

Discussion

The proof-of-concept implementation demonstrates that the plugin architecture can accommodate the full data lifecycle, from ingestion of device-native formats through validation, event detection, and interactive visualisation, without requiring any modification to the core framework. Processing 29 participant sessions with a 92.2% average validity rate confirms that the pipeline produces research-usable output, while the adoption of Tobii Pro Lab's own I-VT parameterisation ensures that results remain directly comparable with those from the manufacturer's established toolchain. These outcomes provide initial evidence that BEHOLD addresses each of the three gaps identified in Section Background and Context: the plugin layer eliminates hardware vendor lock-in (G1), the unified pipeline bridges the collection–analysis separation (G2), and support for both post-hoc and real-time modes, combined with LSL-based synchronisation, enables distributed deployments beyond laboratory configurations (G3). However, this validation currently rests on a single device family, and broader generality will require additional plugins and cross-device comparison. BEHOLD occupies a distinct intersection of capabilities that no single existing tool currently covers: unlike PyGaze (Dalmaijer et al., 2014), which abstracts device differences but delegates analysis to external tools, or OpenIris (Sadeghi et al., 2024), which shares the plugin-based philosophy but lacks an integrated analysis platform, BEHOLD unifies acquisition, processing, and visualisation in a single extensible pipeline. It complements rather than replaces existing tools; practitioners can export synchronised data through the REST API for specialised analyses alongside packages such as PyTrack (Ghose et al., 2020) or R-based statistical tools.

CONCLUSION

This paper presented BEHOLD, a plugin-based framework for device-agnostic eye tracking integration in interactive ecosystem evaluation. The architecture addresses hardware vendor lock-in through an extensible plugin abstraction layer, bridges the collection-analysis separation through a unified four-stage processing pipeline, and supports distributed deployments through LSL-based multimodal synchronisation. A proof-of-concept with Tobii Pro Glasses 2, processing 29 sessions at 92.2% average gaze validity, demonstrates the viability of this approach across the full data lifecycle.

Future work will add plugins for HTC Vive Pro Eye and webcam-based systems to test the abstraction across devices with different accuracy characteristics, and will integrate real-time LSL streaming under varying network conditions. A three-phase validation strategy will accompany this development: technical validation against established tools such as Tobii Pro Lab, cross-device validation comparing metric outputs across plugins, and field validation through pilot studies in smart home and assistive technology contexts at IEETA, leveraging ongoing partnerships with Maxiplás and Bosch.

ACKNOWLEDGEMENT

This work was funder by project SENSINGHOME - “Agenda ILLIANCE” [C644919832-00000035 | Project n.º46], financed by PRR – Plano de Recuperação e Resiliência under the Next Generation EU from the European Union and supported by FCT under unit 00127-IEETA.

REFERENCE

Abeysinghe, Y., Mahanama, B., Jayawardena, G., Sunkara, M., Ashok, V., & Jayarathna, S. (2023). Gaze Analytics Dashboard for Distributed Eye Tracking. *2023 IEEE 24th International Conference on Information Reuse and Integration for Data Science (IRI)*, 140–145. https://doi.org/10.1109/IRI58017.2023.00031

Dalmaijer, E. S., Mathôt, S., & Van der Stigchel, S. (2014). PyGaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments. *Behavior Research Methods*, *46*(4), 913–921. https://doi.org/10.3758/s13428-013-0422-2

Eye Tracking Methodology. (2007). Springer. https://doi.org/10.1007/978-1-84628-609-4

Find out more about the latest features in Tobii Pro Lab. (sem data). Obtido 13 de dezembro de 2025, de https://www.tobii.com/products/software/behavior-research-software/tobii-pro-lab/features

Ghose, U., Srinivasan, A. A., Boyce, W. P., Xu, H., & Chng, E. S. (2020). PyTrack: An end-to-end analysis toolkit for eye tracking. *Behavior Research Methods*, *52*(6), 2588–2603. https://doi.org/10.3758/s13428-020-01392-6

Holmqvist, K., & Andersson, R. (2017). *Eye-tracking: A comprehensive guide to methods, paradigms and measures*.

Jacob, R., & Karn, K. (2003). Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises. Em *Mind; a Quarterly Review of Psychology and Philosophy* (Vol. 2, pp. 573–605). https://doi.org/10.1016/B978-044451020-4/50031-1

Jayawardena, G. (2020). RAEMAP: Real-Time Advanced Eye Movements Analysis Pipeline. *ACM Symposium on Eye Tracking Research and Applications, ETRA ’20 Adjunct*, 1–4. https://doi.org/10.1145/3379157.3391992

Jayawardena, G., Ashok, V., & Jayarathna, S. (2024). Advanced Gaze Analytics Dashboard. *2024 IEEE International Conference on Information Reuse and Integration for Data Science (IRI)*, 114–119. https://doi.org/10.1109/IRI62200.2024.00034

Just, M. A., & Carpenter, P. A. (1976). Eye fixations and cognitive processes. *Cognitive Psychology*, *8*(4), 441–480. https://doi.org/10.1016/0010-0285(76)90015-3

Kothe, C., Shirazi, S. Y., Stenner, T., Medine, D., Boulay, C., Grivich, M. I., Artoni, F., Mullen, T., Delorme, A., & Makeig, S. (2025). The lab streaming layer for synchronized multimodal recording. *Imaging Neuroscience*, *3*, IMAG.a.136. https://doi.org/10.1162/IMAG.a.136

Krejtz, K., Duchowski, A. T., Niedzielska, A., Biele, C., & Krejtz, I. (2018). Eye tracking cognitive load using pupil diameter and microsaccades with fixed gaze. *PLOS ONE*, *13*(9), e0203629. https://doi.org/10.1371/journal.pone.0203629

Lappi, O. (2015). Eye Tracking in the Wild: The Good, the Bad and the Ugly. *Journal of Eye Movement Research*, *8*(5), 1–21. https://doi.org/10.16910/jemr.8.5.1

Niehorster, D. C., & Nyström, M. (2025). TittaLSL: A toolbox for creating networked eye-tracking experiments in Python and MATLAB with Tobii eye trackers. *Behavior Research Methods*, *57*(7), 190. https://doi.org/10.3758/s13428-025-02714-2

Niehorster, D. C., Nyström, M., Hessels, R. S., Andersson, R., Benjamins, J. S., Hansen, D. W., & Hooge, I. T. C. (2025). The fundamentals of eye tracking part 4: Tools for conducting an eye tracking study. *Behavior Research Methods*, *57*(1), 46. https://doi.org/10.3758/s13428-024-02529-7

Nyström, M., & Holmqvist, K. (2010). An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, *42*(1), 188–204. https://doi.org/10.3758/BRM.42.1.188

Olsen, A., & Matos, R. (2012). Identifying parameter values for an I-VT fixation filter suitable for handling data sampled with various sampling frequencies. *Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA '12*, 317–320. https://doi.org/10.1145/2168556.2168625

Sadeghi, R., Ressmeyer, R., Yates, J., & Otero-Millan, J. (2024). Open Iris—An Open Source Framework for Video-Based Eye-Tracking Research and Development. *bioRxiv: The Preprint Server for Biology*, 2024.02.27.582401. https://doi.org/10.1101/2024.02.27.582401

Shin, D.-H. (2017). Conceptualizing and measuring quality of experience of the internet of things: Exploring how quality is perceived by users. *Information & Management*, *54*(8), 998–1011. https://doi.org/10.1016/j.im.2017.02.006

Streitz, N. (2018). *Beyond 'smart-only' cities: Redefining the 'smart-everything' paradigm | Journal of Ambient Intelligence and Humanized Computing | Springer Nature Link*. Obtido 17 de março de 2026, de https://link.springer.com/article/10.1007/s12652-018-0824-1#Bib1

Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, *3*(3), 3–11. https://doi.org/10.1145/329124.329126