Pupil Labs fixation detector

Algorithm description - 3.6.24

Background

Humans visually sample the world by intermittently stabilizing gaze onto discrete targets in their environment. Such fixations are interrupted by saccades, during which gaze is rapidly redirected onto the next target in the visual scene. In dynamic contexts, i.e. one of the intended use-scenarios of our eye trackers, fixational gaze stabilization is achieved by compensatory eye movements counteracting head and body motion. In other words, even while fixating visual targets, our eyes are almost never completely at rest.

The above *functional* definition of a fixation [1] is well suited for the context of head-mounted eye tracking and was therefore adopted for the Pupil Labs fixation detector. Every time interval during which gaze is stabilized towards a visual target is considered a fixation, explicitly including compensatory eye movements in the presence of head or body motion.

In eye-tracking studies employing remote eye trackers with head-restrained subjects, gaze direction can often be expressed in a world-centered coordinate system, e.g. pixel coordinates of a stimulus screen. In such a setup, a standard approach to automated fixation detection is the I-VT (identification by velocity threshold) algorithm, which classifies all samples with gaze velocities below a given threshold value as belonging to a fixation.

For freely moving subjects wearing one of our head-mounted eye trackers, a naive application of the I-VT approach, however, would be prone to errors. Here, gaze direction is given in a subject-centered coordinate system, more specifically in the pixel space of the front-facing scene camera. When a subject moves their head during a fixation, the perspective view of the fixated visual target changes. This induces a concomitant shift of the measured gaze direction in scene camera space. Hence, fixations are not necessarily well characterized by low gaze velocities in this case.

With this in mind, we developed a method for the Pupil Labs fixation detector that works robustly and accurately under both dynamic and static conditions. For an in-depth discussion and evaluation of our approach see our peer-reviewed publication [2]. Our aim was to have a transparent processing pipeline that guarantees interpretability of internal operations and parameters. To achieve this, we extended the classic I-VT algorithm with three additional modules:

1. We use an optic-flow-based velocity-correction stage which effectively compensates for dynamic gaze stabilization during head and body turns.
2. We use an adaptive velocity threshold which adjusts the sensitivity of the algorithm according to the intensity of head motion.

3. We use a set of event-based post-processing filters.

In practice, we estimate global optic flow in scene-camera pixel space based on the output of the inertial measurement unit (IMU). As the IMU provides a direct quantification of head motion, this approach is equivalent to, but computationally more efficient than calculating global optic flow from the video stream. We fall back to the latter option whenever the IMU sensor stream is unavailable.

All parameters of the algorithm (see the next section) were optimized on device-specific in-house datasets, comprising hand-labeled fixations from both static and dynamic real-world scenarios. Note that the algorithm is optimized towards fixation detection performance, not saccade detection performance. While gaps between fixations often represent saccades, sometimes they also reflect periods of mere low signal-to-noise ratio.

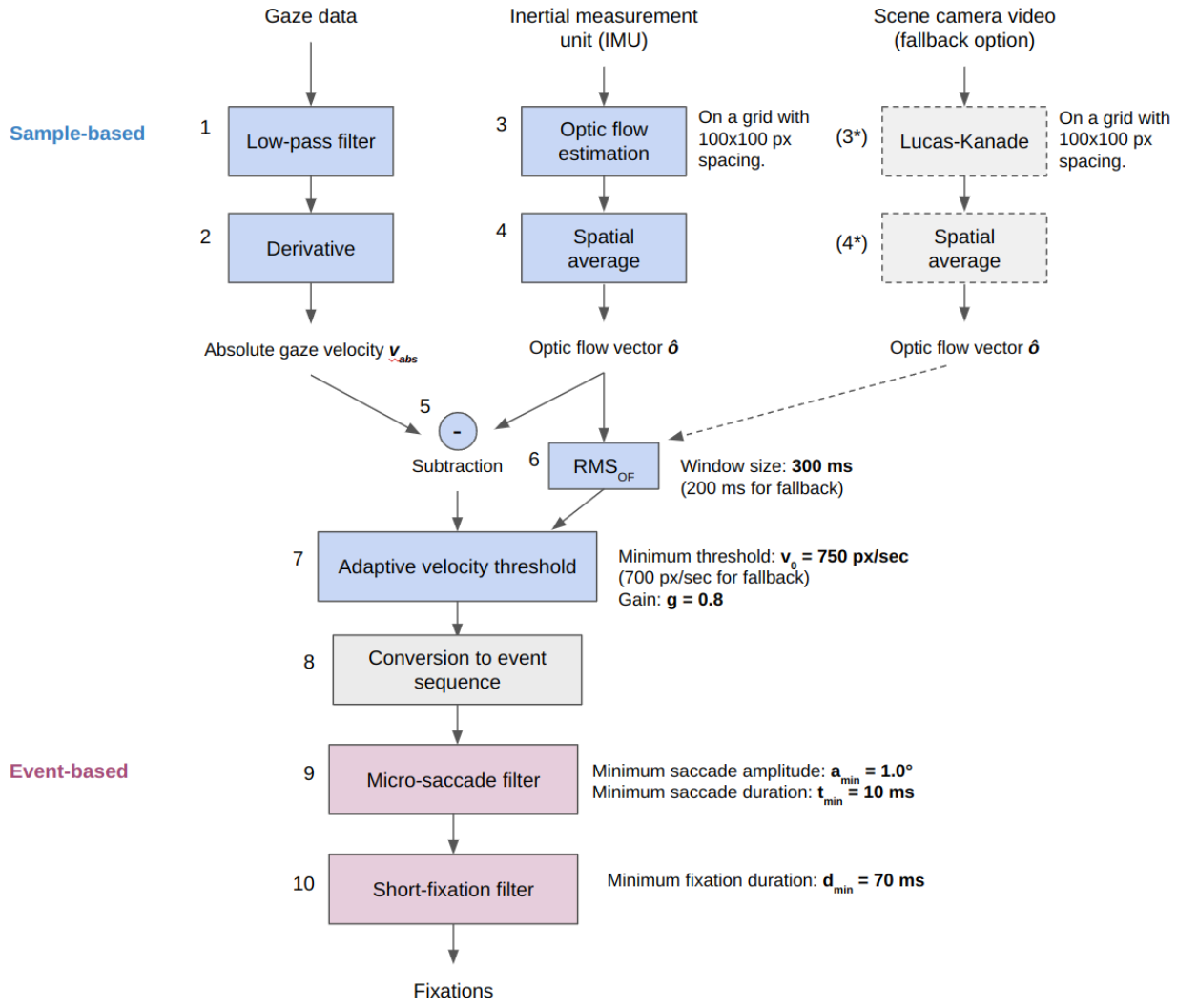

Figure 1: Pupil Neon Fixation Detector

Implementation details

In the following, we will describe the full algorithmic pipeline of the Pupil Labs fixation detector. Parameters given in the text are for **Neon**. Corresponding parameters for **Pupil Invisible** are provided in the last paragraph.

The Pupil Labs fixation detector (see **Figure 1**) takes three data streams as input: the time series of gaze data at 200Hz, the video from the front-facing scene camera at 30Hz, as well as the time series of IMU measurements at 110 Hz (when available).

(1) The gaze data is first low-pass filtered using a Savitzky-Golay filter with 55 ms window length and a polynomial of 3rd grade.

(2) Gaze velocity $\widehat{v}$ is computed from the filtered gaze data as the forward difference between consecutive samples in pixels/sec.

(3) Optic flow is estimated for each frame of the scene camera video by transforming the concurrent movement of the eye tracking headset as measured by the IMU into corresponding translation vectors of pixels in the visual field of the scene camera. Optic flow vectors are computed in pixels/sec on a regular grid with a spacing of 100 pixels. Whenever the IMU is not available for a recording, optic flow is calculated directly from consecutive video frames using the Lucas-Kanade method as a fallback (**3***).

(4) Then, for each frame a global optic flow vector $\widehat{o}$ is constructed by averaging all optic flow vectors over space. It is then upsampled to 200Hz via linear interpolation. This global optic flow vector represents an estimate of whole-field motion of the visual scene, as elicited mainly by head rotations and body turns (same for **4***).

(5) Next, we correct the gaze velocity $\widehat{v}$ by subtracting the global optic flow $\widehat{o}$ and calculate the magnitude of the resulting vector, thus constructing a measure of *relative* gaze velocity $\widehat{v}_{rel}$. This step is based on the notion that in the case of gaze stabilization, image content needs to move in unison with the gaze point in scene camera coordinates. Any significant deviation from a parallel movement of gaze and image content corresponds to a shift in the specific gaze target, i.e. indicating the end of a fixation. From this follows that relative gaze velocity is suitable as an input to an I-VT algorithm, which assumes that gaze velocity must be low during fixations.

(6) Instead of using a standard fixed velocity threshold, however, we use an adaptive threshold which is modulated by the general level of intensity of optic flow during a time interval. This accounts for the fact that optic flow compensation might work less precisely during swift movements of the subject, e.g. head or body turns, due to motion-blur in the camera image or gaze prediction errors at such moments. The value of the adaptive threshold is set as

$$v_{thr} = v_0 + g \cdot RMS_{OF},$$

where v_0 is a minimum velocity threshold, g is a gain factor, and RMS_{OF} is an estimate of the magnitude of optic flow within a time window. The value of RMS_{OF} is calculated as

$$RMS_{OF} = \sqrt{\frac{1}{n}\sum o_x^2 + o_y^2}$$

where o_x and o_y are the x- and y-components of $\hat{o}$, respectively. The sum is taken over all n samples within a sliding window of 300 ms length (200 ms for the fallback option). For the other parameters, the Pupil Labs fixation detector for **Neon** uses $v_0 = 750$ px/s (700 px/s for the fallback option) and $g = 0.8$.

(7) The adaptive velocity threshold is then applied as a classification criterion, analogously to a standard I-VT algorithm: samples below the threshold are considered to be part of a fixation, other samples are classified as gap samples.

(8) After sample-wise classification, consecutive samples with the same label are grouped into events, defined by a type (fixation or gap) as well as a start and end time. The resulting event sequence is further processed in order to filter out events which are physiologically not plausible.

(8.1) First, a micro-saccade filter is applied which removes all gaps which have an amplitude below a minimum saccade amplitude $a_{min} = 1.0°$ and which are shorter than a minimum saccade duration $t_{min} = 10\,ms$. The amplitude is calculated as the angle between start and end point of the event. Removing a gap event leads to automatic merging of the two neighboring fixations.

(8.2) Second, all fixations which are shorter than a minimum fixation length $d_{min} = 70\,ms$ are removed, automatically merging the neighboring gap events.

The resulting event sequence is the output of the fixation detector. We verified that fixation detection performance in highly dynamic real-world scenarios benefits significantly from both the optic-flow compensation stage and the adaptive thresholding [2]. In effect, the Pupil Labs fixation detector regulates its sensitivity, allowing for fine-grained detection of fixations in static experimental settings, while maintaining robustness in more dynamic situations.

In case of recordings made with **Pupil Invisible**, the same algorithm is used, however, without IMU and with a fixed velocity threshold. Effectively, this means that the gain factor g is set to zero. The other parameters in this case are: $v_0 = 900\,px/s$, $a_{min} = 1.5°$, $t_{min} = 60\,ms$, and $d_{min} = 60\,ms$.

Parameter	Symbol	Neon	Pupil Invisible
Minimum velocity threshold	v_0	750 px/s	900 px/s
Gain	g	0.8	0
Minimum saccade amplitude	$a_{\min}$	1.0°	1.5°
Minimum saccade duration	$t_{\min}$	10 ms	60 ms
Minimum fixation duration	$d_{\min}$	70 ms	60 ms

References

[1] Hessels, R. S., Niehorster, D. C., Nyström, M., Andersson, R., & Hooge, I. T. (2018). I*s the eye-movement field confused about fixations and saccades? A survey among 124 researchers.* Royal Society Open Science, 5(8), 180502. https://doi.org/10.1098/rsos.180502

[2] Drews, M., Dierkes, K. (2024). *Strategies for enhancing automatic fixation detection in head-mounted eye tracking*. Behavior Research Methods. https://doi.org/10.3758/s13428-024-02360-0