REMoDNaV: Robust Eye Movement Detection for Natural Viewing

Asim H. Dar[*,1], Adina S. Wagner[*,2], Michael Hanke[2,3]

[*] Both authors have contributed equally

[1] Special Lab Non-Invasive Brain Imaging, Leibniz Institute for Neurobiology, Magdeburg, Germany

[2] Psychoinformatics lab, Institute of Neuroscience and Medicine (INM-7), Research Centre Jülich, Germany

[3] Institute of Systems Neuroscience, Medical Faculty, Heinrich Heine University Düsseldorf, Germany

January 16, 2020

Abstract

Tracking of eye movements is an established measurement for many types of experimental paradigms. More complex and lengthier visual stimuli have made algorithmic approaches to eye movement event classification the most pragmatic option. A recent analysis revealed that many current algorithms are lackluster when it comes to data from viewing dynamic stimuli such as video sequences. Here we present an event classification algorithm—built on an existing velocity-based approach—that is suitable for both static and dynamic stimulation, and is capable of detecting saccades, post-saccadic oscillations, fixations, and smooth pursuit events. We validated classification performance and robustness on three public datasets: 1) manually annotated, trial-based gaze trajectories for viewing static images, moving dots, and short video sequences, 2) lab-quality gaze recordings for a feature length movie, and 3) gaze recordings acquired under suboptimal lighting conditions inside the bore of a magnetic resonance imaging (MRI) scanner for the same full-length movie. We found that the proposed algorithm performs on par or better compared to state-of-the-art alternatives for static stimulation. Moreover, it yields eye movement events with biologically plausible characteristics on prolonged recordings without a trial structure. Lastly, algorithm performance is robust on data acquired under suboptimal conditions that exhibit a temporally varying noise level. These results indicate that the proposed algorithm is a robust tool with improved classification accuracy across a range of use cases. The algorithm is cross-platform compatible, implemented using the Python programming language, and readily available as free and open source software from public sources.

Introduction

A spreading theme in cognitive neuroscience is to use dynamic and naturalistic stimuli such as video clips or movies as opposed to static and isolated stimuli (Matusz et al., 2019). Using dynamic stimuli promises to observe the nuances of cognition in a more life-like environment (Maguire, 2012). Some interesting applications include the determination of neural response to changes in facial expression (Harris et al., 2014), understanding complex social interactions by using videos (Tikka et al., 2012) and more untouched themes such as the underlying processing of music (Toiviainen et al., 2014). In such studies, an unobtrusive behavioral measurement is required to quantify the relationship between stimulus and response. Tracking the focus of participants' gaze is a suitable, well established method that has been successfully employed in a variety of studies ranging from the understanding of visual attention (Liu and Heynderickx, 2011), memory (Hannula et al., 2010) and language comprehension (Gordon et al., 2006). Regardless of use case, the raw eye tracking data (gaze position coordinates) provided by eye tracking devices are rarely used "as is". Instead, in order to disentangle different cognitive, oculomotor, or perceptive states associated with different types of eye movements, most research relies on the classification of eye gaze data into distinct eye movement event categories (Schutz et al., 2011). The most feasible approach for doing this lies in the application of appropriate event classification algorithms.

However, a recent comparison of algorithms found that while many readily available algorithms for eye movement classification performed well on data from static stimulation or short trial-based acquisitions with simplified moving stimuli, none worked particularly well on data from complex naturalistic dynamic stimulation, such as video clips, when compared to human coders (Andersson et al., 2017). And indeed, when we evaluated an algorithm by Nyström and Holmqvist (2010), one of the winners in the aforementioned comparison, on data from prolonged stimulation (≈15 min) with a feature film, we found the average and median durations of labeled fixations to exceed literature reports (e.g., Holmqvist et al., 2011; Dorr et al., 2010) by up to a factor of two. Additionally, and in particular for increasing levels of noise in the data, the algorithm classified too few fixations, as also noted by Friedman et al. (2018), because it discarded potential fixation events that contained data artifacts such as signal-loss and distortion associated with blinks.

Therefore our objective was to improve upon the available eye movement detection and classification algorithms, and develop a tool that performs robustly on data from dynamic, naturalistic stimulation, without sacrificing classification accuracy for static and simplified stimulation. Importantly, we aimed for applicability to prolonged recordings that lack any kind of trial structure, and exhibit periods of signal-loss and non-stationary noise levels. Finally, one of our main objectives was to keep the algorithm as accessible and easily available as possible in order to ease the difficulties associated with closed-source software or non-publicly available source code of published algorithms.

Following the best practices proposed by Hessels et al. (2018), we define the different eye-movements that are supported by our algorithm on a functional and oculomotor dimension as follows: A *fixation* is a period of time during which a part of the visual stimulus is looked at and thereby projected to a relatively constant location on the retina. This type of eye movement is necessary for visual intake, and characterized by a relatively still gaze position in the eye-tracker signal. A fixation event therefore excludes periods of *smooth pursuit*. These are eye movements during which a moving part of the visual stimulus is looked at for visual intake, and thereby projected to a relatively constant location on the retina (Carl and Gellman, 1987). The event is characterized by steadily changing gaze position in the eye-tracker signal. If this type of eye movement is not properly detected and labeled, erroneous fixation and saccade events (which smooth pursuits would be classified into instead) are introduced (Andersson et al., 2017). Contemporary algorithms rarely provide this functionality (but see e.g., Larsson et al., 2015; Komogortsev and Karpov, 2013, for existing algorithms with smooth pursuit classification). *Saccades* on the other hand are also characterized by changing gaze positions, but their velocity trace is usually higher than that of pursuit movements. They serve to shift the position of the eye to a target region, and, unlike during pursuit or fixation events, visual intake is suppressed (Schutz et al., 2011). Lastly, *post-saccadic oscillations* are periods of ocular instability after a saccade (Nyström and Holmqvist, 2010).

Here we introduce REMoDNaV (robust eye movement detection for natural viewing), a new tool that aims to meet our objectives and classifies the eye movement events defined above. It is built on the aforementioned algorithm by Nyström and Holmqvist (2010) (subsequently labeled NH) that employs an adaptive approach to velocity based eye movement event detection and classification. REMoDNaV enhances NH with the use of robust statistics, and a compartmentalization of prolonged time series into short, more homogeneous segments with more uniform noise levels. Furthermore, it adds support for pursuit event classification. Just as the original algorithm, its frame of reference is world centered, i.e. the gaze coordinates have a reference to a stimulation set-up with a fixed position in the world such as x and y coordinates in pixel of a computer screen, and it is meant to be used with eye tracking data from participants viewing static (e.g. images) or dynamic (e.g. videos) stimuli, recorded with remote or tower-mounted eye trackers. Importantly, it is built and distributed as free, open source software, and can be easily obtained and executed with free tools. We evaluated REMoDNaV on three different datasets from conventional paradigms, and natural stimulation (high and lower quality), and relate its performance to the algorithm comparison by Andersson et al. (2017).

Methods

Like NH, REMoDNaV is a *velocity-based* event classification algorithm. Compared to *dispersion-based* algorithms, these types of algorithms are less susceptible to noise and spatio-temporal precision, and are thus applicable to a wide range of sampling frequencies. Furthermore, any influence of biologically implausible velocities and accelerations can be prevented with the use of appropriate filters and thresholds (Holmqvist et al., 2011).

The algorithm comprises two major steps: preprocessing and event classification. A general overview and pseudo-code are shown in Figure 1. The following sections detail individual analysis steps. For each step relevant algorithm parameters are given in parenthesis. Table 1 summarizes all parameters, and lists their default values. The computational definitions of the different eye movements (Hessels et al., 2018) are given within the event classification description. Note, however, that some of the computational definitions of eye movements can be adjusted to comply to alternative definitions by changing the algorithms' parameters.

Preprocessing

The goal of data preprocessing is to compute a time series of eye movement velocities on which the event classification algorithm can be executed, while jointly reducing non-movement-related noise in the data as much as possible.

First, implausible spikes in the coordinate time series are removed with a heuristic spike filter (Stampe, 1993) (Figure 1A, 1). This filter is standard in many eye tracking toolboxes and often used for preprocessing (e.g., Nyström and Holmqvist, 2010). Data samples around signal loss (e.g., eye blinks) can be set to non-numeric values (NaN) in order to eliminate spurious movement signals without shortening the time series (`dilate_nan`, `min_blink_duration`; Figure 1A, 2). This is motivated by the fact that blinks can produce artifacts in the eye-tracking signal when the eyelid closes and re-opens (Choe et al., 2016). Coordinate time series are temporally filtered in two different ways (Figure 1A, 3). A relatively large median filter (`median_filter_length`) is used to emphasize long-distance saccades. This type of filtered data is later used for a coarse segmentation of a time series into shorter intervals between major saccades. Separately, data are also smoothed with a Savitzky-Golay filter (`savgol_`{`length,polyord`}). All event classification beyond the localization of major saccades for time series chunking is performed on this type of filtered data.

After spike-removal and temporal filtering, movement velocities are computed (Figure 1A, 4-5). To disregard biologically implausible measurements, a configurable maximum velocity (`max_vel`) is enforced—any samples exceeding this threshold are replaced by this set value.

Event classification

Saccade velocity threshold

Except for a few modifications, REMoDNaV employs the adaptive saccade detection algorithm proposed by Nyström and Holmqvist (2010), where saccades are initially located by thresholding the velocity time series by a critical

Table 1: Exhaustive list of algorithm parameters, their default values, and units.

Name	**Description**	**Value**
Preprocessing (in order of application during processing)		
`px2deg`	size of a single (square) pixel	no default [deg]
`sampling_rate`	temporal data sampling rate/frequency	no default [Hz]
`min_blink_duration`	missing data windows shorter than this duration will not be considered for `dilate_nan`	0.02 s
`dilate_nan`	duration for which to replace data by missing data markers on either side of a signal-loss window	0.01 s
`median_filter_length`	smoothing median-filter size (for initial data chunking only)	0.05 s
`savgol_length`	size of Savitzky-Golay filter for noise reduction	0.019 s
`savgol_polyord`	polynomial order of Savitzky-Golay filter for noise reduction	2
`max_vel`	maximum velocity threshold, will replace value with maximum, and issue warning if exceeded to inform about potentially inappropriate filter settings (default value based on Holmqvist et al., 2011)	1000 deg/s
Event classification		
`min_saccade_duration`	minimum duration of a saccade event candidate	0.01 s
`max_pso_duration`	maximum duration of a post-saccadic oscillation (glissade) candidate	0.04 s
`min_fixation_duration`	minimum duration of a fixation event candidate	0.04 s
`min_pursuit_duration`	minimum duration of a pursuit event candidate	0.04 s
`min_intersaccade_duration`	no saccade classification is performed in windows shorter than twice this value, plus minimum saccade and PSO duration	0.04 s
`noise_factor`	adaptive saccade onset threshold velocity is the median absolute deviation of velocities in the window of interest, times this factor (peak velocity threshold is twice the onset velocity); increase for noisy data to reduce false positives (Nyström and Holmqvist, 2010, equivalent: 3.0)	5
`velthresh_startvelocity`	start value for adaptive velocity threshold algorithm (Nyström and Holmqvist, 2010), should be larger than any conceivable minimum saccade velocity	300 deg/s
`max_initial_saccade_freq`	maximum saccade frequency for initial classification of major saccades, initial data chunking is stopped if this frequency is reached (should be smaller than an expected (natural) saccade frequency in a particular context), default based on literature reports of a natural, free-viewing saccade frequency of $\sim$1.7 $\pm$0.3 Hz during a movie stimulus (Amit et al., 2017)	2 Hz
`saccade_context_window_length`	size of a window centered on any velocity peak for adaptive determination of saccade velocity thresholds (for initial data chunking only)	1 s
`lowpass_cutoff_freq`	cut-off frequency of a Butterworth low-pass filter applied to determine drift velocities in a pursuit event candidate	4 Hz
`pursuit_velthresh`	fixed drift velocity threshold to distinguish periods of pursuit from periods of fixation; higher than natural ocular drift velocities during fixations (e.g., Goltz et al., 1997; Cherici et al., 2012)	2 deg/s

(A) Preprocessing

1) Apply heuristic spike filter (Stampe, 1993)

2) Mask signal loss (opt): **dilate_nan, min_blink_dir**

3) Noise reduction (opt):
- Median-filter (**median_filter_length**)
- Savitzky-Golay filter (**savgol_length, savgol_polyord**)

4) Calculate velocity
- if velocity > **max_vel**; trim sample to **max_vel**

5) Calculate acceleration

(B) Event detection

1) Calculate peak velocity threshold from entire data iteratively (adapted from NH, using robust statistics). Initial threshold: **velthresh_startvelocity**

2) Determine major saccades (SACC) from sorted velocity samples exceeding the final threshold
- if major saccade frequency > **max_initial_saccade_freq**; stop

3) Detect saccade onset/offset and LPSOs/HPSOs for all SACC (adapted from NH, with local velocity thresholds within **saccade_context_window_length**). Classify PSOs into HPSO or LPSO.

4) Segment data into saccade windows between major saccades

(C) Event detection (2)

5) Compute velocity thresholds within windows

6) Label saccade if
- peak velocity > **local velocity threshold**
- duration > **min_saccade_duration**
- distance to SACC > **min_intersaccade_duration**

7) Detect saccade onset/offset and PSOs as (B3)

8) If unlabeled window > **min_pursuit_duration** Butterworth low-pass filter (**lowpass_cutoff_freq**)

9) Detect smooth pursuit peaks similarly to (B1), using a **pursuit_velthresh**

10) Compute pursuit onset/offset as in (B3/7), using **pursuit_velthresh**

11) Label fixation if velocity < **pursuit_velthresh**

Figure 1: REMoDNaVworkflow in pseudocode. Optional steps and configurable parameters are in bold.

value. Starting from an initial velocity threshold (`velthresh_startvelocity`, termed PT_1 in NH), the critical value is determined adaptively by computing the variance of sub-threshold velocities (V), and placing the new velocity threshold at:

$$PT_n = \overline{V}_{n-1} + F \times \sqrt{\frac{\sum(V_{n-1} - \overline{V}_{n-1})^2}{N-1}} \quad (1)$$

where F determines how many standard deviations above the average velocity the new threshold is located. This procedure is repeated until it stabilizes on a threshold velocity.

$$|PT_n - PT_{n-1}| < 1^\circ/sec \quad (2)$$

REMoDNaV alters this algorithm by using robust statistics that are more suitable for the non-normal distribution of velocities (Friedman et al., 2018), such that the new threshold is computed by:

$$PT_n = median(V_{n-1}) + F \times MAD(V_{n-1}) \quad (3)$$

where MAD is the median absolute deviation, and F is a scalar parameter of the algorithm. This iterative process is illustrated in Figure 2.

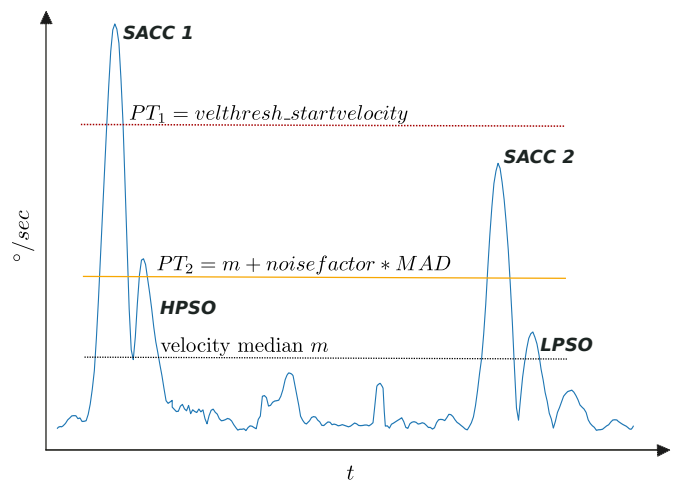

Figure 2: Iterative, global estimation of velocity thresholds for saccades (SACC), and high/low velocity post saccadic oscillations (HPSO/LPSO). The method is adapted from Nyström and Holmqvist (2010), but is modified to use robust statistics with median absolute deviation (MAD) as a measure of variability, more suitable for data with a non-normal distribution.

Time series chunking

As the algorithm aims to be applicable to prolonged recordings without an inherent trial structure and inhomogeneous noise levels, the time series needs to be split into shorter chunks to prevent the negative impact of sporadic noise flares on the aforementioned adaptive velocity thresholding procedure.

REMoDNaV implements this chunking by determining a critical velocity on a median-filtered (`median_filter_length`) time series comprising the full duration of a recording (Figure 1D). Due to potentially elevated noise levels, the resulting threshold tends to overestimate an optimal threshold. Consequently, only periods of fastest eye movements will exceed this threshold. All such periods of consecutive above-threshold velocities are weighted by the sum of these velocities. Boundaries of time series chunks are determined by selecting such events sequentially (starting with the largest sums), until a maximum average frequency across the whole time series is reached (`max_initial_saccade_freq`). The resulting chunks represent data intervals between saccades of maximum magnitude in the respective data. Figure 3 exemplifies event detection within such an intersaccadic interval.

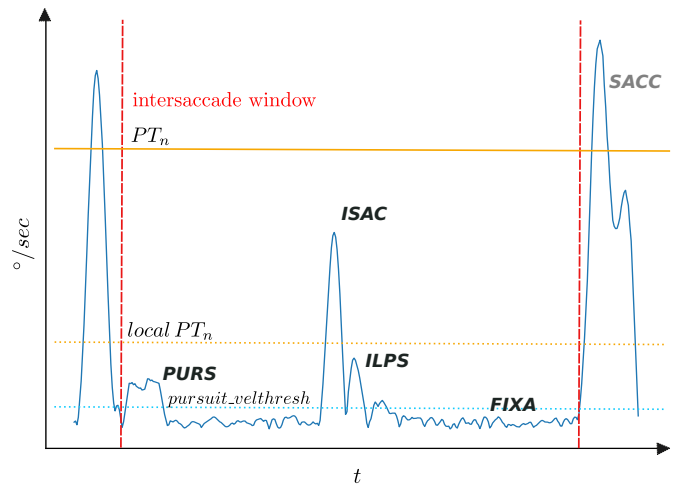

Figure 3: Iterative event detection between major saccades (SACC). The algorithm reports saccades within major saccade windows (ISAC), high/low velocity post saccadic oscillations after ISAC events (IHPS/ILPS), fixations (FIXA), and smooth pursuits (PURS).

Detection of saccades and post-saccadic oscillations

Detection of these event types is identical to the NH algorithm, only the data context and metrics for determining the velocity thresholds differ. For saccades that also represent time series chunk boundaries (event label `SACC`), a context of 1 s (`saccade_context_window_length`) centered on the peak velocity is used by default, for any other saccade (event label `ISAC`) the entire time series chunk represents that context (Figure 1E).

Peak velocity threshold and on/offset velocity threshold are then determined by equation 3 with F set to $2 \times$ `noise_factor` and `noise_factor`, respectively. Starting from a velocity peak, the immediately preceding and the following velocity minima that do not exceed the on/offset threshold are located and used as event boundaries. Qualifying events are rejected if they do not exceed a configurable minimum duration or violate the set saccade maximum proximity criterion (`min_saccade_duration`, `min_intersaccade_duration`).

As in NH, post-saccadic oscillations are events that immediately follow a saccade, where the velocity exceeds the saccade velocity threshold within a short time window (`max_pso_duration`). REMoDNaV distinguishes low-velocity (event label `LPSO` for chunk boundary event, `ILPS` otherwise) and high-velocity oscillations (event label `HPSO` or `IHPS`), where the velocity exceeds the saccade onset or peak velocity threshold, respectively.

Pursuit and fixation classification

For all remaining, unlabeled time series segments that are longer than a minimum duration (`min_fixation_duration`), velocities are low-pass filtered (Butterworth, `lowpass_cutoff_freq`). Any segments exceeding a minimum velocity threshold (`pursuit_velthresh`) are classified as pursuit (event label `PURS`). Pursuit on/offset classification uses the same approach as that for saccades: search for local minima preceding and following the above threshold velocities. Any remaining segment that does not qualify as a pursuit event is classified as a fixation (event label `FIXA`).

Operation

REMoDNaV is free and open-source software, written in the Python language and released under the terms of the MIT license. In addition to the Python standard library it requires the Python packages NumPy (Oliphant, 2006), Matplotlib (Hunter, 2007), statsmodels (Seabold and Perktold, 2010), and SciPy (Jones et al., 2001–) as software dependencies. Furthermore, DataLad (Halchenko et al., 2013–), and Pandas (McKinney et al., 2010) have to be available to run the test battery. REMoDNaV itself, and all software dependencies are available on

all major operating systems. There are no particular hardware requirements for running the software other than sufficient memory to load and process the data.

A typical program invocation looks like

```
remodnav <inputfile> <outputfile> \
    <px2deg> <samplingrate>
```

where `<inputfile>` is the name of a tab-separated-value (TSV) text file with one gaze coordinate sample per line. An input file can have any number of columns, only the first two columns are read and interpreted as X and Y coordinates. Note that this constrains input data to a dense data representation, i.e. either data from eye trackers with fixed sampling frequency throughout the recording, or sparse data that has been transformed into a dense representation beforehand. The second argument `<outputfile>` is the file name of a BIDS-compliant (Gorgolewski et al., 2016) TSV text file that will contain a report on one detected eye movement event per line, with onset and offset time, onset and offset coordinates, amplitude, peak velocity, median velocity and average velocity. The remaining arguments are the only two mandatory parameters: the conversion factor from pixels to visual degrees, i.e., the visual angle of a single (square) pixel (`<px2deg>` in deg), and the temporal sampling rate (`<sampling_rate>` in Hz). Any other supported parameter can be added to the program invocation to override the default values.

A complete list of supported parameters (sorted by algorithm step) with their description and default value, are listed in Table 1. While the required user input is kept minimal, the number of configurable parameters is purposefully large to facilitate optimal parameterization for data with specific properties. Besides the list of detected events, a visualization of the classification results, together with a time course of horizontal and vertical gaze position, and velocities is provided for illustration and initial quality assessment of algorithm performance on each input data file.

Validation analyses

The selection of datasets and analyses for validating algorithm performance was guided by three objectives: 1) compare to other existing solutions; 2) demonstrate plausible results on data from prolonged gaze coordinate recordings during natural viewing; and 3) illustrate result robustness on lower-quality data. The following three sections each introduce a dataset and present the validation results for these objectives. All analysis presented here are performed using default parameters (Table 1), with no dataset-specific tuning other than the built-in adaptive behavior.

Algorithm comparison

Presently, Andersson et al. (2017) represents the most comprehensive comparative study on eye movement classification algorithms. Moreover, the dataset employed in that study was made publicly available. Consequently, evaluating REMoDNaV performance on these data and using their metrics offers a straightforward approach to relate this new development to alternative solutions.

The dataset provided by Andersson et al. (2017)[1] consists of monocular eye gaze data produced from viewing stimuli from three distinct categories—images, moving dots and videos. The data release contains gaze coordinate time series (500 Hz sampling rate), and metadata on stimulus size and viewing distance. Importantly, each time point was manually classified by two expert human raters as one of six event categories: fixation, saccade, PSO, smooth pursuit, blink and undefined (a sample that did not fit any other category). A minor labeling mistake reported in Zemblys et al. (2018) was fixed prior to this validation analysis.

For each stimulus category, we computed the proportion of misclassifications per event type, comparing REMoDNaV to each of the human coders, and, as a baseline measure, the human coders against each other. A time point was counted as misclassified if the two compared classifications did not assign the same label. We limited this analysis to all time points that were labeled as fixation, saccade, PSO, or pursuit by any method, hence ignoring the rarely used NaN/blinks or "undefined" category. For a direct comparison with the results in Andersson et al. (2017), the analysis was repeated while also excluding samples labeled as pursuit. Table 2 shows the misclassification rates for all pairwise comparisons, in all stimulus types. In com-

[1] github.com/richardandersson/EyeMovementDetectorEvaluation

parison to the NH algorithm, after which the proposed work was modelled, REMoDNaVperformed consistently better (32/93/70% average misclassification for NH, vs. 6.5/10.8/ 9.1% worst misclassification for REMoDNaV in categories images, dots, and videos). Compared to all ten algorithms evaluated in Andersson et al. (2017), REMoDNaV exhibits the lowest misclassification rates across all stimulus categories. When taking smooth pursuit events into account, the misclassification rate naturally increases, but remains comparably low. Importantly, it still exceeds the performance of all algorithms tested in Andersson et al. (2017) in the dots and video category, and performs among the best in the images category. Additionally, both with and without smooth pursuit, REMoDNaVs performance exceeds also that of a recent deep neural network trained specifically on video clips (Startsev et al., 2018, compare Table 7: 34% misclassification versus 31.5% for REMoDNaV).

Figure 4 shows confusion patterns for a comparison of algorithm classifications with human labeling. While REMoDNaV does not achieve a labeling similarity that reaches the human inter-rater agreement, it still performs well. In particular, the relative magnitude of agreement with each individual human coder for fixations, saccades, and PSOs, resembles the agreement between the human coders. Classification of smooth pursuits is consistent with human labels for the categories moving dots, and videos. However, there is a substantial confusion of fixation and pursuit for the static images. In a real-world application of REMoDNaV, pursuit classification could be disabled (by setting a high pursuit velocity threshold) for data from static images, if the occurrence of pursuit events can be ruled out a priori. For this evaluation, however, no such intervention was made.

In addition to the confusion analysis and again following Andersson et al. (2017), we computed Cohen's Kappa (Cohen, 1960) for an additional measure of similarity between human and algorithm performance. It quantifies the sample-by-sample agreement between two ratings following equation 4:

$$K = \frac{P_o - P_c}{1 - P_c} \tag{4}$$

where P_o is the observed proportion of agreement between the ratings, and P_c is the proportion of chance agreement. A value of $K = 1$ indicates per-

Table 2: Proportion of samples in each stimulus category classified in disagreement between human coders (MN, RA) and the REMoDNaV algorithm (AL). The MC (misclassification) column lists proportions considering all four event categories (fixation, saccade, PSO, pursuit), while the w/oP (without pursuit) column excludes pursuit events for a direct comparison with Andersson et al. (2017, Tables 8-10). The remaining columns show the percentage of labels assigned to incongruent time points by each rater (deviation of their sum from 100% is due to rounding).

Images

Comp	MC	w/oP	Coder	Fix	Sac	PSO	SP
MN-RA	6.1	3.0	MN	70	9	21	0
—	—	—	RA	13	15	20	53
MN-AL	23.1	6.5	MN	86	2	11	2
—	—	—	AL	5	13	6	75
RA-AL	22.8	6.4	RA	77	3	11	9
—	—	—	AL	13	13	6	68

Dots

Comp	MC	w/oP	Coder	Fix	Sac	PSO	SP
MN-RA	10.7	4.2	MN	11	10	9	71
—	—	—	RA	64	7	6	23
MN-AL	18.6	8.2	MN	9	5	8	78
—	—	—	AL	77	6	2	15
RA-AL	22.8	10.8	RA	28	4	6	61
—	—	—	AL	59	7	2	31

Videos

Comp	MC	w/oP	Coder	Fix	Sac	PSO	SP
MN-RA	18.5	4.0	MN	75	3	8	15
—	—	—	RA	16	4	3	77
MN-AL	31.5	7.9	MN	57	1	6	36
—	—	—	AL	36	5	3	55
RA-AL	28.5	9.1	RA	38	3	5	55
—	—	—	AL	53	6	5	35

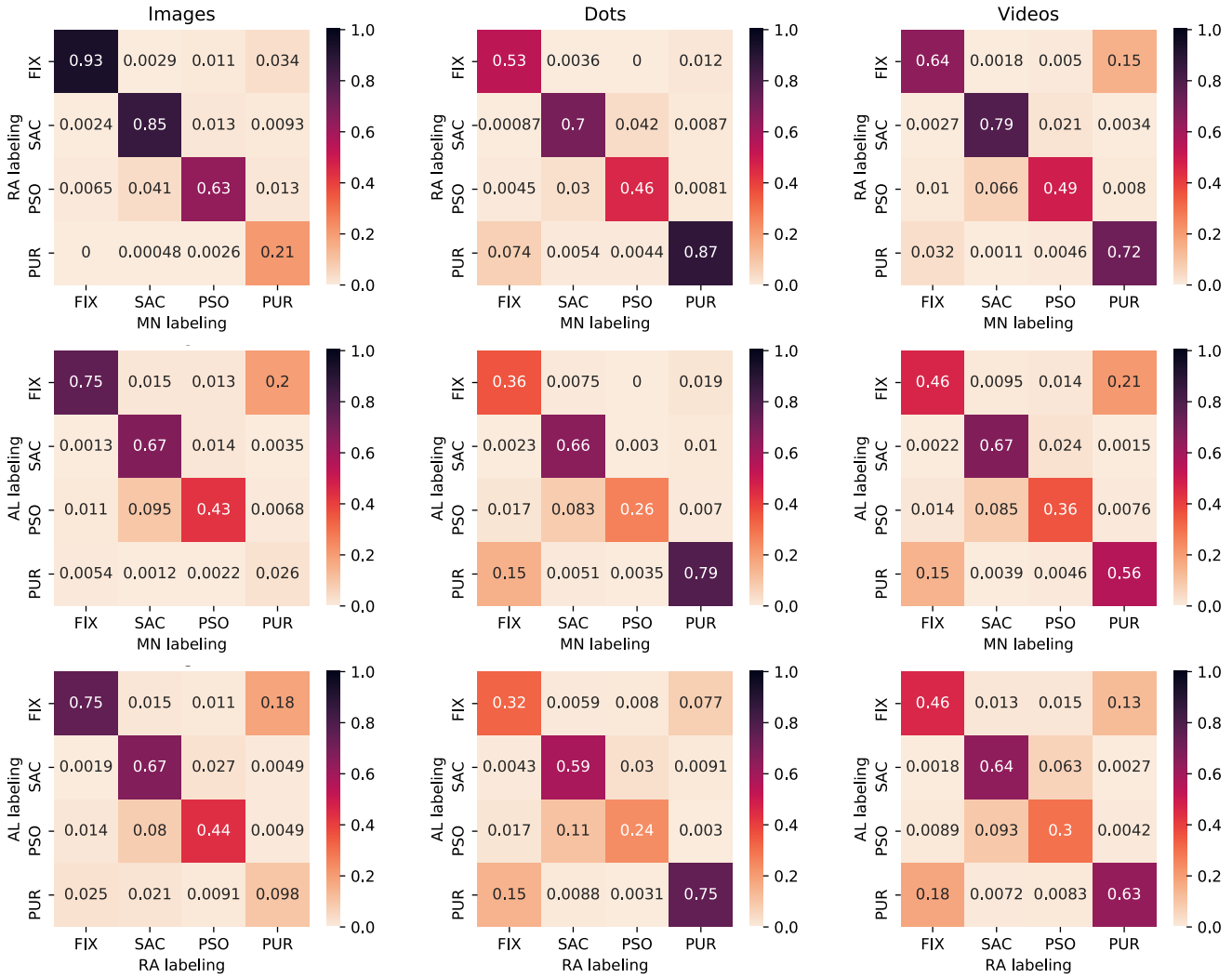

Figure 4: Confusion patterns for pairwise eye movement classification comparison of both human raters (MN and RA; Andersson et al., 2017) and the REMoDNaV algorithm (AL) for gaze recordings from stimulation with static images (left column), moving dots (middle column), and video clips (right column). All matrices present gaze sample based Jaccard indices (JI; Jaccard, 1901). Consequently, the diagonals depict the fraction of time points labeled congruently by both raters in relation to the number of timepoints assigned to a particular event category by any rater.

Table 3: Cohen's Kappa reliability between human coders (MN, RA), and REMoDNaV (AL) with each of the human coders.

Fixations			
Comparison	Images	Dots	Videos
MN versus RA	0.84	0.65	0.65
AL versus RA	0.55	0.37	0.44
AL versus MN	0.52	0.45	0.39
Saccades			
Comparison	Images	Dots	Videos
MN versus RA	0.91	0.81	0.87
AL versus RA	0.78	0.72	0.76
AL versus MN	0.78	0.78	0.79
PSOs			
Comparison	Images	Dots	Videos
MN versus RA	0.76	0.62	0.65
AL versus RA	0.59	0.38	0.45
AL versus MN	0.58	0.41	0.51

fect agreement, and $K = 0$ indicates chance level agreement. Table 3 displays the resulting values between the two human experts, and REMoDNaV with each of the experts, for each stimulus category and the three event types used in Andersson et al. (2017), namely fixations, saccades, and PSOs (compare to Andersson et al. (2017), table 7). For all event types and stimulus categories, REMoDNaV performs on par or better than the original NH algorithm, and in many cases on par or better than the best of all algorithms evaluated in Andersson et al. (2017) within an event or stimulus type.

In order to further rank the performance of the proposed algorithm with respect to the ten algorithms studied in Andersson et al. (2017), we followed their approach to compute root mean square deviations (RMSD) from human labels for event duration distribution characteristics (mean and standard deviation of durations, plus number of events) for each stimulus category (images, dots, videos) and event type (fixations, saccades, PSOs, pursuits). This measure represents a scalar distribution dissimilarity score that can be used as an additional comparison metric of algorithm performance that focuses on overall number and durations of detected events, instead of sample-by-sample misclassification. The RMSD measure has a lower bound of 0.0 (identical to the average of both human raters), with higher values indicating larger differences (for detail information on the calculation of this metric see Andersson et al., 2017).

Table 4 reproduces Andersson et al. (2017, Tables 3-6), and the RMSD calculation for the added rows on REMoDNaV is based on the scores for the human raters given in these original tables. As acknowledged by the authors, the absolute value of the RMSD scores is not informative due to scaling with respect to the respective maximum value of each characteristic. Therefore, we converted RMSDs for each algorithm and event type into zero-based ranks (lower is more human-like).

The LNS algorithm (Larsson et al., 2013) was found to have the most human-like performance for saccade and PSO classification in Andersson et al. (2017). REMoDNaV performs comparable to LNS for both event types (saccades: 2.0 vs. 3.3; PSOs: 2.3 vs. 2.0, mean rank across stimulus categories for LNS and REMoDNaV, respectively).

Depending on the stimulus type, different algorithms performed best for fixation classification. NH performed best for images and videos, but worst for moving dots. REMoDNaV outperforms all other algorithms in the dots category, and achieves rank 5 and 6 (middle range) for videos and images, respectively. Across all stimulus and event categories, REMoDNaV achieves a mean ranking of 2.9, and a mean ranking of 3.2 when not taking smooth pursuit into account.

Taken together, REMoDNaV yields classification results that are, on average, more human-like than any other algorithm tested on the dataset and metrics put forth by Andersson et al. (2017). In particular, its performance largely equals or exceeds that of the original NH algorithm. NH outperforms it only for fixation detection in the image and video category, but in these categories REMoDNaV also classifies comparatively well. These results are an indication that the changes to the NH algorithm proposed here to improve upon its robustness are not detrimental to its performance on data from conventional paradigms and stimuli.

Prolonged natural viewing

Given that REMoDNaV yielded plausible results for the "video" stimulus category data in the Andersson et al. (2017) dataset (Figure 4, and Table 4, right

Table 4: Comparison of event duration statistics (mean, standard deviation, and number of events) for image, dot, and video stimuli. This table reproduces Andersson et al. (2017, Tables 3-6), and root-mean-square-deviations (RMSD) from human raters are shown for fixations, saccades, PSOs, and pursuit as zero-based ranks (rank zero is closest to the average of the two human raters). Rows for REMoDNaV have been added.

Fixations

	Images				Dots				Videos			
Algorithm	Mean	SD	#	rank	Mean	SD	#	rank	Mean	SD	#	rank
MN	248	271	380	1	161	30	2	1	318	289	67	0
RA	242	273	369	0	131	99	13	0	240	189	67	1
CDT	397	559	251	10	60	127	165	9	213	297	211	7
EM	-	-	-	-	-	-	-	-	-	-	-	-
IDT	399	328	242	7	323	146	8	5	554	454	48	8
IKF	174	239	513	5	217	184	72	6	228	296	169	4
IMST	304	293	333	3	268	140	12	3	526	825	71	10
IHMM	133	216	701	8	214	286	67	8	234	319	194	6
IVT	114	204	827	9	203	282	71	7	202	306	227	9
NH	258	299	292	2	380	333	30	10	429	336	83	2
BIT	209	136	423	4	189	113	67	4	248	215	170	3
LNS	-	-	-	-	-	-	-	-	-	-	-	-
REMoDNaV	187	132	426	6	116	65	43	2	147	107	144	5

Saccades

	Images				Dots				Videos			
Algorithm	Mean	SD	#	rank	Mean	SD	#	rank	Mean	SD	#	rank
MN	30	17	376	0	23	10	47	0	26	13	116	0
RA	31	15	372	1	22	11	47	1	25	12	126	1
CDT	-	-	-	-	-	-	-	-	-	-	-	-
EM	25	22	787	9	17	14	93	8	20	16	252	6
IDT	35	15	258	3	32	14	10	7	24	53	41	9
IKF	62	37	353	10	60	26	29	10	55	20	107	8
IMST	17	10	335	6	13	5	18	6	18	10	76	4
IHMM	48	26	368	8	41	17	27	9	42	18	109	7
IVT	41	22	373	5	36	14	28	4	36	16	112	5
NH	50	20	344	7	43	16	42	5	44	18	1104	10
BIT	-	-	-	-	-	-	-	-	-	-	-	-
LNS	29	12	390	2	26	11	53	2	28	12	122	2
REMoDNaV	39	20	388	4	30	13	40	3	33	15	118	3

Post-saccadic oscillations

	Images				Dots				Videos			
Algorithm	Mean	SD	#	rank	Mean	SD	#	rank	Mean	SD	#	rank
MN	21	11	312	1	15	5	33	0	20	11	97	1
RA	21	9	309	0	15	8	28	1	17	8	89	2
NH	28	13	237	4	24	12	17	4	28	13	78	4
LNS	25	9	319	2	20	9	31	2	24	10	87	3
REMoDNaV	19	8	277	3	18	8	14	3	18	8	86	0

Pursuit

	Images				Dots				Videos			
Algorithm	Mean	SD	#	rank	Mean	SD	#	rank	Mean	SD	#	rank
MN	363	187	3	1	375	256	37	1	521	347	50	1
RA	305	184	16	0	378	364	33	0	472	319	68	0
REMoDNaV	197	73	118	2	440	385	34	2	314	229	97	2

columns), we determined whether it is capable of analyzing data from dynamic stimulation without a trial structure.

As a test dataset we used publicly available eye tracking data from the *studyforrest.org* project, where 15 participants were recorded watching a feature-length (≈2 h) movie in a laboratory setting (Hanke et al., 2016). Eye movements were measured by an Eyelink 1000 with a standard desktop mount (software version 4.51; SR Research Ltd., Mississauga, Ontario, Canada) and a sampling rate of 1000 Hz. The movie stimulus was presented on a 522 × 294 mm LCD monitor at a resolution of 1920 × 1280 px and a viewing distance of 85 cm. Participants watched the movie in eight approximately 15 min long segments, with measurement recalibration before every segment.

As no manual eye movement event labeling exists for these data, algorithm evaluation was limited to a comparison of marginal distributions and well-known properties, such as the log-log-linear relationship of saccade amplitude and saccade peak velocity (Bahill et al., 1975). Figure 5 (top row) depicts this main sequence relationship. Additionally, Figure 6 (top row) shows duration histograms for all four event types across all participants. Shapes and locations of these distributions match previous reports in the literature, such as a strong bias towards short (less than 500 ms) fixations for dynamic stimuli (Dorr et al., 2010, Fig. 3), peak number of PSOs with durations between 10-20 ms (Nyström and Holmqvist, 2010, Fig. 11), and a non-Gaussian saccade duration distribution located below 100 ms (Nyström and Holmqvist, 2010, Fig. 8, albeit for static scene perception). Overall, the presented summary statistics suggest that REMoDNaV is capable of detecting eye movements with plausible characteristics, in prolonged gaze recordings without a trial structure. A visualization of such a classification result is depicted in Figure 7 (top row).

Lower-quality data

An explicit goal for REMoDNaV development was robust performance on lower-quality data. While lack of quality cannot be arbitrarily compensated and will inevitably lead to misses in eye movement classification, it is beneficial for any further analysis if operation on noisy data does not introduce unexpected event property biases.

In order to investigate noise-robustness we ran REMoDNaV on another publicly available dataset from the *studyforrest.org* project, where 15 different participants watched the exact same movie stimulus, but this time while lying on their back in the bore of an MRI scanner (Hanke et al., 2016). These data were recorded with a different Eyelink 1000 (software version 4.594) equipped with an MR-compatible telephoto lens and illumination kit (SR Research Ltd., Mississauga, Ontario, Canada) at 1000 Hz during simultaneous fMRI acquisition. The movie was presented at a viewing distance of 63 cm on a 26 cm (1280 × 1024 px) LCD screen in 720p resolution at full width, yielding a substantially smaller stimulus size, compared to the previous stimulation setup. The eye tracking camera was mounted outside the scanner bore and recorded the participants' left eye at a distance of about 100 cm. Compared to the lab-setup, physical limitations of the scanner environment, and sub-optimal infrared illumination led to substantially noisier data, as evident from a generally higher amount of data loss and a larger spatial uncertainty (Hanke et al., 2016, Technical Validation): Six out of 15 subjects had one or more movie segments with data loss greater than 30%, compared to zero out of 15 subjects in the laboratory setting. The average spatial accuracy at the start of a recording, assessed with a 13-point calibration procedure, was 0.58 degrees of visual angle for the MRI sample and 0.45 degrees for the lab sample. An example of the amplified and variable noise pattern is shown in Figure 7 (bottom row, black lines). Except for the differences in stimulation setup, all other aspects of data acquisition, eye tracker calibration, and data processing were identical to the previous dataset.

We performed the identical analysis as before, in order to compare performance between a high and lower-quality data acquisition. This approach differs from the common approach of adding increasing levels of artificial noise to data, but bears the important advantage of incorporating real lower-quality data characteristics instead of potentially inappropriate or unnatural noise. Figures 5-7 depict the results for the lab-quality dataset, and the MRI-scanner dataset in the top and bottom rows, respectively.

Overall, the classification results exhibit strong similarity, despite the potential behavioral impact of watching a movie while lying on their back and look-

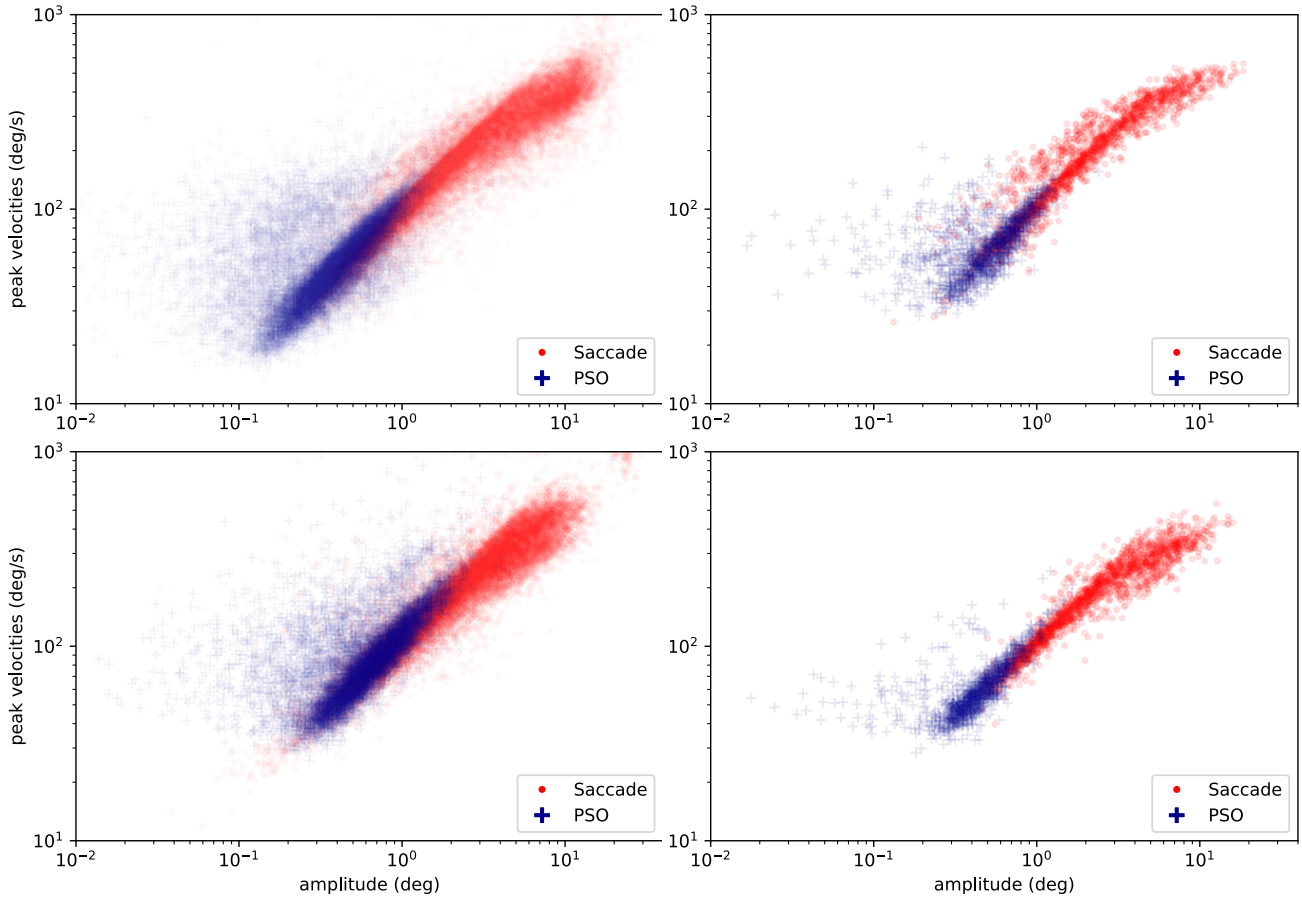

Figure 5: Main sequence of eye movement events during one 15 minute sequence of the movie (segment 2) for lab (top), and MRI participants (bottom). Data across all participants per dataset is shown on the left, and data for a single exemplary participant on the right.

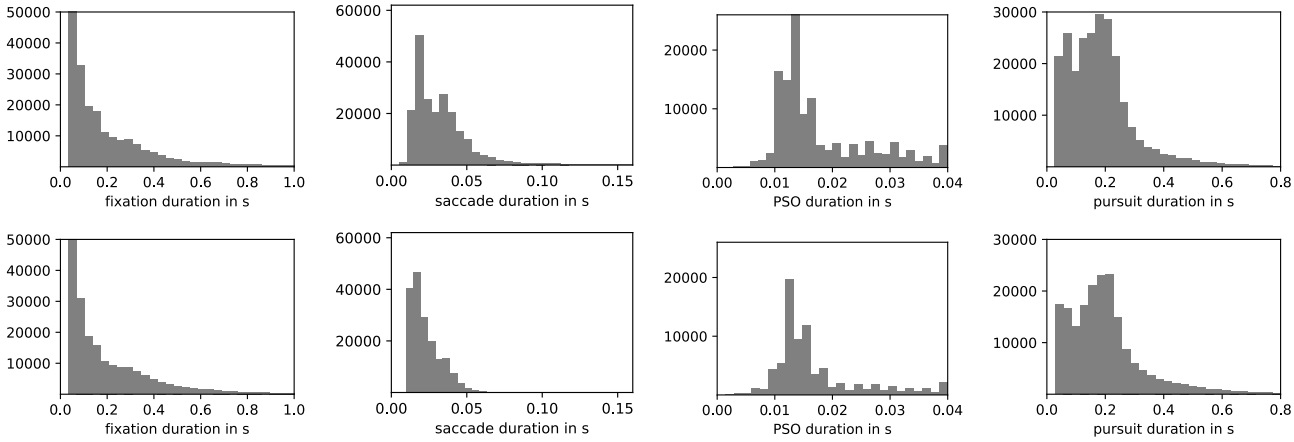

Figure 6: Comparison of eye movement event duration distributions for the high-quality lab sample (top row), and the lower quality MRI sample (bottom row) across all participants (each $N = 15$), and the entire duration of the same feature-length movie stimulus. All histograms depict absolute number of events. Visible differences are limited to an overall lower number of events, and fewer long saccades for the MRI sample. These are attributable to a higher noise level and more signal loss (compare Hanke et al., 2016, Fig. 4b) in the MRI sample, and to stimulus size differences (23.75 ° MRI vs. 34 ° lab).

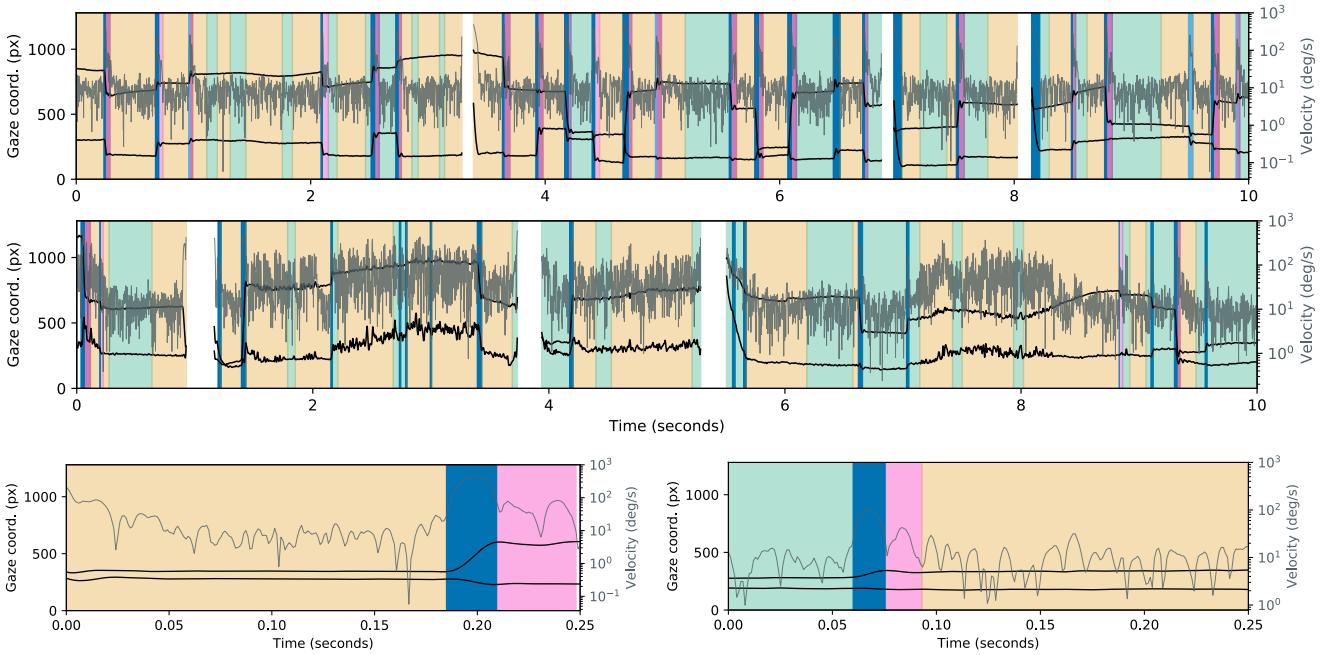

Figure 7: Exemplary eye movement classification results for the same 10 s excerpt of a movie stimulus for a single participant in the high quality lab sample (top), and in the lower quality MRI sample (middle). The bottom panel shows 250 ms of lab (left) and mri (right) sample in greater detail. The plots show filtered gaze coordinates (black), computed velocity time series (gray) overlayed on the eye movement event segmentation with periods of fixation (green), pursuit (beige), saccades (blue), and high/low-velocity post-saccadic oscillations (dark/light purple). The variable noise level, and prolonged signal loss (white) visible in the MRI sample represent a challenge for algorithms. REMoDNaV uses an adaptive approach that determines major saccade events first, and subsequently tunes the velocity threshold to short time windows between these events. Figures like this accompany the program output to facilitate quality control and discovery of inappropriate preprocessing and classification parameterization.

ing upwards on the participants, or the well known effect of increasing fatigue (Tagliazucchi and Laufs, 2014) during a two-hour session in an MRI-scanner. In particular, saccade amplitude and peak velocity exhibit a clear main-sequence relationship that resembles that found for the lab acquisition (Figure 5). Duration distributions for fixations, PSOs, and pursuits are strikingly similar between the two datasets (Figure 6), except for a generally lower number of detected events for the MRI experiment, which could be explained by the higher noise level and fraction of signal loss. There is a notable difference regarding the saccade duration distributions, with a bias towards shorter saccades in the MRI dataset. This effect may be attributable to the differences in stimulus size (30% smaller in the MRI environment).

Conclusion

Based on the adaptive, velocity-based algorithm for fixation, saccade, and PSO classification by Nyström and Holmqvist (2010), we have developed an improved algorithm that, in contrast to the original, performs robustly on prolonged recordings with dynamic stimulation, without a trial structure and variable noise levels, and also supports the classification of smooth pursuit events. Through a series of validation analyses we have shown that its performance is comparable to or better than ten other contemporary detection algorithms, and that plausible classification results are achieved on high and lower quality data. These aspects of algorithm capabilities and performance suggest that REMoDNaV is a state-of-the-art tool for eye movement classification with particular relevance for emerging complex, naturalistic data collections paradigms, such as the combination of eye tracking and functional MRI in simultaneous measurements.

The proposed algorithm is rule-based, hence can be applied to data without prior training, apart from the adaptive estimation of velocity thresholds. This aspect distinguishes it from other recent developments based on deep neural networks (Startsev et al., 2018), and machine-learning in general (Zemblys et al., 2018). Some statistical learning algorithms require (labeled) training data, which can be a limitation in the context of a research study. However, in its present form REMoDNaV cannot be used for real-time data analysis, as its approach for time series chunking is based on an initial sorting of major saccade events across the entire time series. The proposed algorithm presently does not support the classification of eye blinks as a category distinct from periods of general signal loss. While such a feature could potentially be added, the current default preprocessing aims at removing blink-related signal. The algorithm maintains a distinction between high- and low-velocity PSOs first introduced by Nyström and Holmqvist (2010), although, to our knowledge, the present literature does not make use of such a distinction. Algorithm users are encouraged to decide on a case-by-case basis whether to lump these event categories together into a general PSO category, as done in our own validation analyses. Lastly, the evaluation results presented here are based on data with a relatively high temporal resolution (0.5 and 1 kHz). While the algorithm does not impose any hard constraints on data acquisition parameters, its performance on data from low-end, consumer grade hardware (e.g., 50 Hz sampling rate) has not been tested.

Just as Andersson et al. (2017), we considered human raters as a gold standard reference for event classification when evaluating algorithms. The implications of the results presented herein are hence only valid if this assumption is warranted. Some authors voice concerns (e.g., Komogortsev et al., 2010), regarding potential biases that may limit generalizability. Nevertheless, human-made event labels are a critical component of algorithm validation, as pointed out by Hooge et al. (2018).

The validation analyses presented here are based on three different datasets: a manually annoted dataset (Andersson et al., 2017), and two datasets with prolonged recordings using movie stimuli (Hanke et al., 2016). Beyond our own validation, a recent evaluation of nine different smooth pursuit algorithms by Startsev, Agtzidis and Dorr as part of their recent paper (Startsev et al., 2018) also provides metrics for REMoDNaV. In their analysis, algorithm performance was evaluated against a partially hand-labelled eye movement annotation of the Hollywood2 dataset (Mathe and Sminchisescu, 2012). We refrain from restating their methodology or interpreting their results here, but encourage readers to consult this independent report[2].

REMoDNaV aims to be a readily usable tool,

[2]https://www.michaeldorr.de/smoothpursuit/

available as cross platform compatible, free and open source software, with a simple command line interface and carefully chosen default settings. However, as evident from numerous algorithm evaluations (e.g., Andersson et al., 2017; Larsson et al., 2013; Zemblys et al., 2018; Komogortsev et al., 2010) different underlying stimulation, and data characteristics can make certain algorithms or parameterizations more suitable than others for particular applications. The provided implementation of the REMoDNaV algorithm (Hanke et al., 2019) acknowledges this fact by exposing a range of parameters through its user interface that can be altered in order to tune the classification for a particular use case.

The latest version of REMoDNaV can be installed from PyPi[3] via `pip install remodnav`. The source code of the software can be found on Github[4]. All reports on defects and enhancement can be submitted there. The analysis code underlying all results and figures presented in this paper, as well as the LaTeX sources, are located in another Github repository[5]. All required input data, from Andersson et al. (2017) and the *studyforrest.org* project, are referenced in this repository at precise versions as DataLad[6] subdatasets, and can be obtained on demand. The repository constitutes an automatically reproducible research object, and readers interested in verifying the results and claims of our paper can recompute and plot all results with a single command after cloning the repository.

Author contributions

AD, MH conceived and implemented the algorithm. AD, AW, MH validated algorithm performance. AD, AW, MH wrote the manuscript.

Competing interests

No competing interests were disclosed.

Grant information

Michael Hanke was supported by funds from the German federal state of Saxony-Anhalt and the European Regional Development Fund (ERDF), Project: Center for Behavioral Brain Sciences (CBBS). Adina Wagner was supported by the German Academic Foundation.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

[3] https://pypi.org/project/remodnav

[4] https://github.com/psychoinformatics-de/remodnav

[5] https://github.com/psychoinformatics-de/paper-remodnav/

[6] `http://datalad.org`

References

Amit R, Abeles D, Bar-Gad I, Yuval-Greenberg S (2017) Temporal dynamics of saccades explained by a self-paced process. Scientific reports 7(1):886, DOI 10.1038/s41598-017-00881-7

Andersson R, Larsson L, Holmqvist K, Stridh M, Nyström M (2017) One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms. Behavior Research Methods 49(2):616–637, DOI 10.3758/s13428-016-0738-9

Bahill AT, Clark MR, Stark L (1975) The main sequence, a tool for studying human eye movements. Mathematical Biosciences 24(3-4):191–204, URL `https://doi.org/10.1016/0025-5564(75)90075-9`

Carl JR, Gellman RS (1987) Human smooth pursuit: stimulus-dependent responses. Journal of Neurophysiology 57(5):1446–1463, DOI 10.1152/jn.1987.57.5.1446, pMID: 3585475

Cherici C, Kuang X, Poletti M, Rucci M (2012) Precision of sustained fixation in trained and untrained observers. Journal of Vision 12(6):31–31, DOI 10.1167/12.6.31

Choe KW, Blake R, Lee SH (2016) Pupil size dynamics during fixation impact the accuracy and precision of video-based gaze estimation. Vision research 118:48–59

Cohen J (1960) A coefficient of agreement for nominal scales. Educational and psychological measurement 20(1):37–46

Dorr M, Martinetz T, Gegenfurtner KR, Barth E (2010) Variability of eye movements when viewing dynamic natural scenes. Journal of vision 10(10):28–28, DOI 10.1167/10.10.28

Friedman L, Rigas I, Abdulin E, Komogortsev OV (2018) A novel evaluation of two related and two independent algorithms for eye movement classification during reading. Behavior Research Methods 50(4):1374–1397, DOI 10.3758/s13428-018-1050-7

Goltz H, Irving E, Steinbach M, EizenamnIZENMAN M (1997) Vertical eye position control in darkness: Orbital position and body orientation interact to modulate drift velocity. Vision Research 37(6):789 – 798, DOI https://doi.org/10.1016/S0042-6989(96)00217-9

Gordon PC, Hendrick R, Johnson M, Lee Y (2006) Similarity-based interference during language comprehension: Evidence from eye tracking during reading. Journal of Experimental Psychology: Learning, Memory, and Cognition 32(6):1304–1321, DOI 10.1037/0278-7393.32.6.1304

Gorgolewski KJ, Auer T, Calhoun VD, Craddock RC, Das S, Duff EP, Flandin G, Ghosh SS, Glatard T, Halchenko YO, et al. (2016) The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. Scientific Data 3:160044, DOI 10.1038/sdata.2016.44

Halchenko YO, Hanke M, et al. (2013–) DataLad: perpetual decentralized management of digital objects. DOI 10.5281/zenodo.1470735, URL `http://datalad.org`

Hanke M, Adelhöfer N, Kottke D, Iacovella V, Sengupta A, Kaule FR, Nigbur R, Waite AQ, Baumgartner F, Stadler J (2016) A studyforrest extension, simultaneous fMRI and eye gaze recordings during prolonged natural stimulation. Scientific Data 3:160092, DOI 10.1038/sdata.2016.92

Hanke M, Dar AH, Wagner A (2019) psychoinformatics-de/remodnav: Submission time. DOI 10.5281/zenodo.2651042

Hannula DE, Althoff RR, Warren DE, Riggs L, Cohen NJ, Ryan JD (2010) Worth a glance: using eye movements to investigate the cognitive neuroscience of memory. Frontiers in Human Neuroscience 4:166, DOI 10.3389/fnhum.2010.00166

Harris RJ, Young AW, Andrews TJ (2014) Dynamic stimuli demonstrate a categorical representation of facial expression in the amygdala. Neuropsychologia 56(100):47–52, DOI 10.1016/j.neuropsychologia.2014.01.005

Hessels RS, Niehorster DC, Nyström M, Andersson R, Hooge IT (2018) Is the eye-movement field confused about fixations and saccades? a survey among 124 researchers. Royal Society open science 5(8):180502

Holmqvist K, Nyström M, Andersson R, Dewhurst R, Jarodzka H, Van de Weijer J (2011) Eye tracking: A comprehensive guide to methods and measures. OUP Oxford

Hooge ITC, Niehorster DC, Nyström M, Andersson R, Hessels RS (2018) Is human classification by experienced untrained observers a gold standard in fixation detection? Behavior Research Methods 50(5):1864–1881, DOI 10.3758/s13428-017-0955-x

Hunter JD (2007) Matplotlib: A 2d graphics environment. Computing in science & engineering 9(3):90–95, DOI 10.1109/MCSE.2007.55

Jaccard P (1901) Étude comparative de la distribution florale dans une portion des alpes et des jura. Bull Soc Vaudoise Sci Nat 37:547–579

Jones E, Oliphant T, Peterson P, et al. (2001–) SciPy: Open source scientific tools for Python. URL `http://www.scipy.org`

Komogortsev OV, Karpov A (2013) Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. Behavior Research Methods 45(1):203–215, DOI 10.3758/s13428-012-0234-9

Komogortsev OV, Gobert DV, Jayarathna S, Koh DH, Gowda SM (2010) Standardization of automated analyses of oculomotor fixation and saccadic behaviors. IEEE Transactions on Biomedical Engineering 57(11):2635–2645, DOI 10.1109/TBME.2010.2057429

Larsson L, Nyström M, Stridh M (2013) Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit. IEEE Transactions on Biomedical Engineering 60(9):2484–2493, DOI 10.1109/TBME.2013.2258918

Larsson L, Nyström M, Andersson R, Stridh M (2015) Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. Biomedical Signal Processing and Control 18:145 – 152, DOI https://doi.org/10.1016/j.bspc.2014.12.008

Liu H, Heynderickx I (2011) Visual attention in objective image quality assessment: Based on eye-tracking data. IEEE Transactions on Circuits and Systems for Video Technology 21(7):971–982, DOI 10.1109/TCSVT.2011.2133770

Maguire EA (2012) Studying the freely-behaving brain with fmri. Neuroimage 62(2):1170–1176

Mathe S, Sminchisescu C (2012) Dynamic eye movement datasets and learnt saliency models for visual action recognition. In: Proceedings, Part II, of the 12th European Conference on Computer Vision — ECCV 2012 - Volume 7573, Springer-Verlag, Berlin, Heidelberg, p 842–856

Matusz PJ, Dikker S, Huth AG, Perrodin C (2019) Are we ready for real-world neuroscience? Journal of Cognitive Neuroscience 31(3):327–338, DOI 10.1162/jocn_e_01276, pMID: 29916793

McKinney W, et al. (2010) Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference, Austin, TX, vol 445, pp 51–56

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

Oliphant TE (2006) A guide to NumPy, vol 1. Trelgol Publishing USA

Schutz AC, Braun DI, Gegenfurtner KR (2011) Eye movements and perception: A selective review. Journal of Vision 11(5):9–9, DOI 10.1167/11.5.9

Seabold S, Perktold J (2010) Statsmodels: Econometric and statistical modeling with python. In: 9th Python in Science Conference

Stampe DM (1993) Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. Behavior Research Methods, Instruments, & Computers 25(2):137–142, DOI 10.3758/BF03204486

Startsev M, Agtzidis I, Dorr M (2018) 1d cnn with blstm for automated classification of fixations, saccades, and smooth pursuits. Behavior Research Methods DOI 10.3758/s13428-018-1144-2

Tagliazucchi E, Laufs H (2014) Decoding wakefulness levels from typical fmri resting-state data reveals reliable drifts between wakefulness and sleep. Neuron 82(3):695 – 708, DOI https://doi.org/10.1016/j.neuron.2014.03.020, URL `http://www.sciencedirect.com/science/article/pii/S0896627314002505`

Tikka P, Väljamäe A, de Borst AW, Pugliese R, Ravaja N, Kaipainen M, Takala T (2012) Enactive cinema paves way for understanding complex real-time social interaction in neuroimaging experiments. Frontiers in Human Neuroscience 6:298, DOI 10.3389/fnhum.2012.00298

Toiviainen P, Alluri V, Brattico E, Wallentin M, Vuust P (2014) Capturing the musical brain with Lasso: Dynamic decoding of musical features from fMRI data. NeuroImage 88:170–180, DOI 10.1016/J.NEUROIMAGE.2013.11.017

Zemblys R, Niehorster DC, Holmqvist K (2018) gazenet: End-to-end eye-movement event detection with deep neural networks. Behavior Research Methods DOI 10.3758/s13428-018-1133-5