One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms

Richard Andersson[1,2] · Linnea Larsson[3] · Kenneth Holmqvist[4] · Martin Stridh[3] · Marcus Nyström[4]

Abstract Almost all eye-movement researchers use algorithms to parse raw data and detect distinct types of eye movement events, such as fixations, saccades, and pursuit, and then base their results on these. Surprisingly, these algorithms are rarely evaluated. We evaluated the classifications of ten eye-movement event detection algorithms, on data from an SMI HiSpeed 1250 system, and compared them to manual ratings of two human experts. The evaluation focused on fixations, saccades, and post-saccadic oscillations. The evaluation used both event duration parameters, and sample-by-sample comparisons to rank the algorithms. The resulting event durations varied substantially as a function of what algorithm was used. This evaluation differed from previous evaluations by considering a relatively large set of algorithms, multiple events, and data from both static and dynamic stimuli. The main conclusion is that current detectors of only fixations and saccades work reasonably well for static stimuli, but barely better than chance for dynamic stimuli. Differing results across evaluation methods make it difficult to select one winner for fixation detection. For saccade detection, however, the algorithm by Larsson, Nyström and Stridh (IEEE Transaction on Biomedical Engineering, 60(9):2484–2493,2013)

outperforms all algorithms in data from both static and dynamic stimuli. The data also show how improperly selected algorithms applied to dynamic data misestimate fixation and saccade properties.

Keywords Eye-tracking · Inter-rater reliability · Parsing

Introduction

Eye movements are a common source of data in neurology, psychology, and many other fields. For example, many conditions and syndromes cause saccades that are "hypometric", i.e. undershooting the target (see Leigh & Zee, 2006, for many examples). It would thus be extremely unfortunate if the offset of the saccade was erroneously determined by the computer algorithm used to parse the data. Nonetheless, such algorithms are indeed frequently used when analysing data, and often without a conscious decision and evaluation of the algorithm candidates. There is little agreement on what combination of algorithms, thresholds, types of eye movements, sampling frequency, and stimuli that achieves sufficient classification accuracy for the researcher to be able to confidently draw conclusions about the parameters of the eye movements in question. This makes it very hard to confidently generalize research findings across experiments using different hardware, algorithms, thresholds, and stimuli. This paper compares eye movement parameters and similarities of ten different algorithms, along with two human expert coders, across five different eye movement events and using data from three types of stimuli. The overall goal is to evaluate the algorithms and to select a clear winner to recommend to the community.

✉ Richard Andersson
 Richard.Andersson@lucs.lu.se

[1] Eye Information Group, IT University of Copenhagen, Copenhagen, Denmark

[2] Department of Philosophy and Cognitive Science, Lund University, Box 192, 221 00 Lund, Sweden

[3] Department of Biomedical Engineering, Lund University, Lund, Sweden

[4] Humanities Laboratory, Lund University, Lund, Sweden

The reasons for parsing eye movements into distinct events

The act of classifying eye movements into distinct events is, on a general level, driven by a desire to isolate different intervals of the data stream strongly correlated with certain oculomotor or cognitive properties. For example, the visual intake of the eye is normally severely limited during a saccade, and this, along with a general need for data reduction, seems to have been motivating factors for the early fixation/saccade detectors (Kliegl & Olson, 1981; Matin, 1974; Rayner, 1998). Similarly, smooth pursuit movements are triggered by perceived motion, indicating visual intake (Rashbass, 1961), but these movements may stretch across many areas of interest (AOI), ruining standard AOI and fixation measures if the movements are treated as fixations. The eye blink is another form of movement, although not by the eyeball, but this is typically detected in order to exclude it from the data stream so it does not interfere with further eye movement classification, or used as training to remove eye-related artefacts in electro-oculographic data. The eye blink is also associated with a limited visual intake, even before the closing of the lid and after the re-opening (Volkmann, Riggs, & Moore, 1980). This is particularly important, as the raw data when the eyelid closes and opens may appear, in some eye-trackers, identical to saccades moving down and up again (see p.177, Holmqvist, Nyström, Andersson, Jarodzka & Van de Weijer, 2011).

Also, the wobble of the crystalline lens in the eye during a saccade is thought to produce deformations in the iris (and hence pupil) around the time of the saccade, producing what is known as *post-saccadic oscillations* (PSOs) in the eye-tracker signal (Hooge, Nyström, Cornelissen, & Holmqvist, 2015; Nyström, Andersson, Magnusson, Pansell, & Hooge, 2015a; Tabernero & Artal, 2014). PSOs are less commonly searched for, but are increasingly important as new studies and eye-trackers push the limits of temporal and spatial resolutions. The corresponding data samples of PSOs are not systematically grouped with either fixations or saccades (Nyström & Holmqvist, 2010). As there is still uncertainty about the precise nature of visual intake during such oscillations (e.g., intake but with distortions; Deubel & Bridgeman, 1995; Tabernero & Artal, 2014), how this event is classified is therefore crucial for a researcher using eye-movement classification algorithms for selecting periods of maximum visual intake.

It is up to the individual researcher to either manually identify these events or to use any of the commercial or freely available computer algorithms. Manual identification is of course best if you want a classification that best matches the researcher's conception of a fixation, saccade, or some other event. A human coder can also adaptively update his or her beliefs regarding what the typical event looks like, and can stop and discuss the problem when there are tricky borderline cases or incomplete event definitions. However, a human working manually on this problem is very slow compared to a computer, which is why computer-based algorithms today are the only practical solution. The most common practice is simply to use the event classifier provided with your system, and often leaving any parameter thresholds at their default settings. This practice is indeed fast, easily dividing up large amounts of raw samples into distinct events.

Current algorithms for eye-movement classification

There exists a large number of different algorithms today and it is impossible to evaluate them all, for several reasons. First, some algorithms are commercial closed-source solutions from the eye-tracker manufacturer and thus impossible to implement identically. Although it is possible to get both raw data and identified oculomotor events from a closed-source system, the original data are stored in a binary file particular to that commercial system, and the commercial event classifiers only accept their own binary files. Thus, we have to use algorithms that allow us to input data in an open format. Secondly, not all algorithms exist as actual ready-made implementations, so an evaluation means adapting or finishing these implementations, which in turn may add bugs and biases. So any valid algorithm needs to be an officially released implementation. Thirdly, not all algorithms work out-of-box on real-world data. For example, the algorithm by (Mould, Foster, Amano, & Oakley, 2012) failed when we evaluated it with a data file that had missing samples. Although this is a simple task to fix, it affects the original algorithm and is thus no longer objectively evaluated by us. Finally, for practical reasons, we limited ourselves to ten algorithms which furthermore should have no support for smooth-pursuit identification, which complicates matters further and is beyond the scope of this paper (but see Komogortsev & Karpov, 2012, for an evaluation of these algorithms). A search for a set of ten algorithms that fulfilled these criteria produced the algorithms that are described in the following paragraphs (see Table 1).

The *Fixation Dispersion Algorithm based on Covariance (CDT)* by Veneri et al. (2011) is an improvement of the fixation dispersion algorithm based on F-tests (FDT) previously developed by the same authors (Veneri et al., 2010). The improvement consists in complementing their previous, F-test-based, algorithm with co-variance calculations on the *x*- and *y*-coordinates of the gaze. The logic behind this is that the F-test is very sensitive to violations to the normality assumption of the data. This algorithm uses variance and co-variance thresholds as well as a duration threshold.

The algorithm used by Engbert and Mergenthaler (2006; henceforth *EM*) is a further development of the algorithm used earlier by Engbert and Kliegl (2003). This algorithm uses a

Table 1 Algorithm classifications

Algorithm	Events					
	Fixation	Saccade	PSO	Smooth pursuit	Blink	Undefined
(Humans)	√	√	√	√	√	√
CDT	√					
EM		√				
IDT	√	√				
IKF	√	√				
IMST	√	√				
IHMM	√	√				
IVT	√	√				
NH	√	√	√			
BIT	√					
LNS		√	√			

Note. The oculomotor events that are explicitly detected by each algorithm implementation are indicated with a check mark. The algorithms presented are, in order: human coders, Fixation Dispersion Algorithm based on Covariance (CDT), Engbert & Mergenthalser (EM), Identifiacation by Dispersion-Threshold (IDT), Identification by Kalman Filter (IKF), Identification by Minimal Spanning Tree (IMST), Identification by Hidden Markov Model (IHMM), Identification by Velocity Threshold (IVT), Nyström & Holmqvist (NH), Binocular-Individual Threshold (BIT), and Larsson, Nyström & Stridh (LNS)

velocity threshold to detect saccades, but it sets the threshold adaptively, in relation to the identified noise level of the data. Additionally, this algorithm enforces a minimal saccade duration to reduce the effects of noise. It should also be noted that this algorithm was primarily developed for detecting microsaccades, but that is also works for detecting voluntary (larger) saccades.

One of the most common algorithms for detecting fixations is the *Identification by Dispersion-Threshold (IDT)* algorithm. According to Salvucci and Goldberg (2000), it is based on the data reduction algorithm by Widdel (1984). The IDT algorithm works with *x*- and *y* data, and two fixed thresholds: the maximum fixation dispersion threshold and the minimum fixation duration threshold. To be a fixation, data samples constituting at least enough time to fulfill the duration threshold has to be within a spatial area not exceeding the dispersion threshold. The samples fulfilling these criteria are marked as belonging to a fixation. One particular detail of this specific implementation is that it is part of a package that also merges short nearby fixations, and also paired with a saccade detector (Komogortsev, Gobert, Jayarathna, Koh, & Gowda, 2010).

Real data is often noisy and may suffer from data loss. An algorithm that is designed to overcome this problem should be very promising. A Kalman filter is a recursive filter that provides an optimal, i.e. minimized error, combination of the current measurement and the predicted measurement given previous input. Strictly speaking, the Kalman filter in this

Identification by Kalman Filter (IKF) algorithm does not classify the eye-tracker signal into events, but in this particular implementation (Komogortsev & Khan, 2009) it is done by a χ^2-test. This test classifies all samples within a set window length as belonging to a fixation if the χ^2 value is below the set threshold and fulfils a minimum duration threshold, or as belonging to a saccade if this value is above the threshold. This particular implementation, as others by Komogortsev et al. (2010), uses the same post-processing as the other algorithms, and thus clusters nearby fixations.

Another approach to event detection is the *Identification by Minimal Spanning Tree (IMST)*. This algorithm creates a "tree" of the data, which branches out to the data samples. The algorithm strives to capture all the data with a minimum of branching so that samples from two different clusters are better captured by branches to two separate nodes (which are connected higher up in the tree) rather than forcing a very extensive branching to a single node at a lower level. By enforcing certain thresholds on the samples at the edges of a cluster, saccades are identified and excluded from the fixation detection. The implementation in question is created by Komogortsev et al. (2010).

Considering that the most common type of event distinction is that between the almost stationary fixation and the fast-moving saccade, an algorithm that classifies based on probabilistic switching between two states appears intuitive. Such an algorithm for *Identification by Hidded Markov Model (IHMM)* is described in Komogortsev et al. (2010), and is formed around a velocity-based algorithm and then wrapped by two additional algorithms. The first algorithm re-classifies fixations and saccades depending on probabilistic parameters (e.g., initial state, and state transition probability), and the second algorithm that updates these parameters.

A very common basis for separating samples belonging to fixations from samples belonging to saccades, is to identify the velocities of these samples. The *Identification by Velocity Threshold (IVT)* is a simple algorithm that functions in this way (Salvucci & Goldberg, 2000). It uses a fixed velocity threshold to identify fixation and saccades, where fixations are segments of samples with point-to-point velocities below the set velocity threshold, and saccades are segments of sample with velocities above this threshold. This basic velocity criterion is often the core of other algorithms. This particular implementation is from Komogortsev et al. (2010).

The algorithm presented by Nyström and Holmqvist (2010; henceforth *NH*) was the first algorithm to explicitly also identify post-saccadic oscillations (called "glissades" in the paper) along with fixations and saccades. It is an adaptive algorithm in the sense that it adjusts the velocity threshold based on the noise level in the data.

Detecting small saccades from noise is a challenge, and it makes sense to take advantage of the fact that the eyes are most often directed at the same object. So if the left eye moves

to a certain object, then the right eye should be doing so too. This makes it easier to determine whether a peak in velocity is due to a real movement or simply noise, as both eyes should show this peak in the velocity curve simultaneously. This idea is taken advantage of by the *Binocular-Individual Threshold (BIT)* algorithm, developed by van der Lans, Wedel and Pieters (2011). Like several other algorithms, this is an algorithm that adaptively sets thresholds.

The final algorithm that we consider in this paper is a recent development by Larsson, Nyström and Stridh (2013; henceforth *LNS*). This algorithm is the second algorithm that is able to detect post-saccadic oscillations, but it also detects saccades. The algorithm is adaptive, but what is novel is that it was designed with the aim to detect saccades and post-saccadic oscillations even in the presence of smooth pursuit movements of the eye. Smooth pursuit movements generate velocities that are inconsistently handled by algorithms that maintain standard velocity thresholds. Thus, there is generally no clear classification of these movements as either fixations or saccades, but they rather depend on the particular smooth pursuit movement and the current algorithm thresholds.

Evaluation of classification algorithms

The lack of a single standard algorithm used in all systems and that many algorithms addressing the same problem exist, suggests that eye movement classification is not a trivial problem, and that evaluating the performance of different algorithms may not be a trivial task. Crucially, determining a winner among several algorithms means that an appropriate evaluation method has to be devised. As this study is not the first one attempting to evaluate algorithm performance, it is fruitful to consider the benefits and drawback of already established methods.

The most basic evaluation technique, used for identifying events in early eye-movement research, was simply the manual inspection by the researcher (e.g., Javal, 1879). At this point in time, the purpose was to identify certain events (saccades) rather than evaluate algorithms. Modern evaluations, however, used this manual approach together with automated methods of evaluating classifications. For example, raw data samples cluster together in a fixation, and a fixation detection algorithm should detect all samples belonging to these clusters, and reject samples outside of the clusters (see for example Fig. 2 on p. 883 in Blignaut, 2009). Unfortunately, such manual parts of an evaluation are often mentioned in passing, e.g., that Vig, Dorr, and Barth (2009, p. 399) manually tweaked their parameters until it looked good, as referenced by Mould et al. (2012, p. 22).

This manual evaluation, however, can more rigorously be put to use if the human evaluators systematically code the same data using the same categories an algorithm would.

Then it is possible to directly compare the performance of algorithms relative to human experts. This is often also done when evaluating new algorithms, e.g., by Larsson, Nyström and Stridh (2013), and Mould et al. (2012). A Human – Algorithm comparison, however, often assumes that humans behave perfectly rationally and that, consequently, any deviation from perfect agreement is due to the mistakes of the algorithm. Thus, a question highly related to this approach is how reliable the human coders are. In many fields of research, this is analyzed using specific measures for inter-rater reliability, like Cohen's Kappa (K; Cohen, 1960), the ratio of non-agreeing samples, or calculating a correlation between coders (e.g., Munn, Stefano, & Pelz, 2008).

Another approach is to assume an optimal or rational relation between the stimuli and the viewing behavior of an individual. For example, Komogortsev et al. (2010) used an approach where an experiment participant is instructed to look at an animated single target that makes a series of jumps. Given a known number of jumps, known positions, known amplitudes, among other things, it is possible to calculate how the ideal eye movement behavior would look like. The gaze data parsed by the algorithms are then compared against this ideal gaze behavior, and the more similar, the better the algorithm. Although there may be some intuitive appeal in this approach, there are also some potential concerns. Primarily, participants are not always perfectly rational in their behavior, nor can they control their eye to the extent they want. In fact, they frequently undershoot their intended target and produce corrective saccades (see Kapoula & Robinson, 1986, for a nuanced view). A participant may be distracted and forget a target, or try to anticipate a future target, and in so doing, incurring a penalty to even a perfect eye-movement classification algorithm. This set up could also be biased towards simple tasks that are easy to follow for the participants, so the algorithms are never tested with eye movements that deviate from this norm. Additionally, Oculomotor events such as post-saccadic oscillations are also, by all current accounts, beyond the volition of the participant.

Finally, there are two aspects of the data that can be evaluated. The first aspect is the event that each sample gets assigned to, i.e. the label, regardless of the actual values of the particular data samples. The second is the actual data values contained in that sample which in turn determines the properties of the event it is part of. In the evaluations in this paper, we have decided to focus on the labeling process, i.e. the classification of samples as belonging to certain oculomotor events. One reason for this decision is that a pure classification task is rather straightforward as it is either hit or miss in assigning the correct label. The second reason is that these sample classifications in turn, "for free", provide three event parameters: the number of events, the durations (i.e. how many

consecutive samples) of these events, and the variance of these event durations. So, one evaluation task provides three possible quality values to evaluate. Furthermore, as saccades on average follow the main sequence, i.e., there is a systematic relationship between the duration and amplitude of a saccade, we get an indirect measure of the amplitude of the saccades as well (Bahill, Clark, & Stark, 1975).

The challenges for event detection algorithms

The detection of eye movement events is not a completely solved problem, for a number of reasons. First of all, and relating to the previous section, is that there is no consensus on how to evaluate the algorithms, which means that further refinement of the algorithms is hindered as we do not know whether differences are due to the algorithms or the evaluation process. Surprisingly little effort has gone into investigating the classifications of human coders, and what combination of knowledge, instructions, data, and visualizations drive the humans to code more (or less) similarly. Even measures tailored for estimating inter-rater reliability, such as Cohen's K, have flaws. Cohen's K estimates the reliability depending on the base rate of events, so it compensates for the fact that some events may be more common than others, but this base rate also assumes that humans would pick randomly across events if they are not certain, which may not be likely. Therefore, humans may achieve higher or lower reliability scores than warranted.

Secondly, we are not even always completely sure what we mean when we talk about an event. There is e.g., no *theoretically* motivated threshold for when the eye is moving sufficiently in a particular direction to be classified as a *saccade,* and anything below that threshold to be classified as something else. Often a *saccade* is detected with the motivation that our visual intake is severely limited during the movement, and the data from this event should be removed. However, visual intake is also limited even before (50 % detection at −20 ms) and after (50 % detection at +75 ms) a saccade onset (Volkmann, Schick, & Riggs, 1968), a fact which is not reflected in any event classification algorithms that we have seen. The classification algorithms seem to focus on a strict oculomotoric definition of fixations and saccades. Even from a purely oculomotor definition of eye movements, it is difficult to identify the point where a fixation ends and an extremely slow-moving smooth pursuit starts. This point is arbitrary and more or less determined by the precision of the system. Also, if human experts have a hard time agreeing on the same data, then of course computer algorithms designed by humans in the first place would also classify the same data stream differently. This is not helped by the fact that not all

algorithms can detect every event, which affects the few events it actually does detect. For example, an algorithm incapable of detecting a post-saccadic oscillation may see the oscillating movement as two saccades, divided by an implausibly short, e.g., 1 ms, fixation (p. 165, Holmqvist et al., 2011).

Thirdly, many algorithms have some form of settings that need to be set by the researcher, such as minimum fixation duration, saccade velocity threshold, etc. If there was a clear and theoretically driven threshold, then this would already be hard-coded into the algorithm. Now it is up to the researcher, which means that novel results deviating from previous results can be due to the selected algorithm, the selected thresholds, or both, or something completely else. It is common knowledge that different algorithms and different parameter values for these algorithms produce different classification results (e.g., Komogortsev et al., 2010; Salvucci & Goldberg, 2000; Shic, Chawarska & Scassellati, 2008).

Fourthly, there exist many algorithms, but not so many comparisons of the algorithms. Often, a modest evaluation is performed when presenting a novel algorithm, but this often considers only a few algorithms and is primarily oriented towards showcasing the new algorithm.

Fifthly, dynamic scenes are increasingly common as stimuli, but not all researchers are aware that it is improper to use standard fixation detection methods in the presence of smooth pursuit (the commercial software packages that we have seen do not prompt a warning about this), or dynamic stimuli are treated in research without mention of this challenge. As one example of several, a conference paper by Ali-Hasan, Harrington and Richman (2008) present "best practices for using eye-tracking to answer usability question for video and TV. Although they mention fixations, there was no mention at all of event detection algorithms in this inherently dynamic domain. Although the researchers may well be aware of this problem, their readers are not alerted to this potentially problematic issue.

Standard (non-smooth pursuit) algorithms should not be accurate for data from dynamic stimuli, but we also do not know the extent of the problem. We do not know whether the problem affects primarily fixations, or saccades, or both. This problem can also be viewed in relation to the fact that most algorithms do not detect post-saccadic oscillations. Is the problem of accurate post-saccadic oscillation identification a more pressing research area than the accurate identification of smooth pursuit?

Finally, despite fuzzy definitions, researchers talk about fixations, saccades, and other events at conferences with apparent ease. So there must be some intuitions between experts on what events occur in a given stream of data, although it is currently unclear around what events or types of data these intuitions are the strongest. Would they agree on the number of different events in the data, and just differ in where the

precise borders are, or would they even select completely different events for the same segment of data? So the human experts seem to have consistent intuitions that enable them to talk unhindered about these event with colleagues, which suggests that there is some ground truth against which the data can be evaluated, and ultimately find a winner among the evaluated algorithms. Nonetheless, an evaluation should not trust the human experts, but also include them on equal terms with the algorithms in the evaluation. If the human experts do not agree, than it would be unfair to hold the algorithms to that standard.

Aims of the reported evaluation

In a previous section, we pointed out several groups of problems with evaluating current eye movement classification algorithms. Naturally, it is beyond the scope of this paper to address all of them. Our focus is rather to evaluate ten event detection algorithms against two human experts in order to recommend one winner to researchers. However, we will also address a set of related questions:

- How does the number of detected events and their durations vary with the choice of an algorithm?
- How similar are the events detected by algorithms compared to events manually identified by human experts?
- How similar are algorithms and humans in a simple sample-by-sample comparison, which is the most human-like?
- How similar are our two human coders? Are they interchangeable, or will the event detection process depend on the human experts we use?
- Does the algorithm – human or human – human similarity depend on the stimuli used for eliciting the eye movement data, i.e. will it differ between static and dynamic stimuli?
- What are the consequences of trying to detect events in the presence of smooth pursuit using improper, i.e. not designed to handle smooth pursuit, algorithms? Is this a serious violation, or will such a use provide acceptable approximations of the properties of the true (human-identified) events?
- How congruent are different evaluation techniques, such as similarity based on event durations, compared to similarity based on Cohen's Kappa?
- Given that algorithms do not classify identically with human experts, what types of deviating classifications do the algorithms do? Are the deviations random, or do they indicate a clear bias in some direction? What area should developers focus on improving, to gain the highest marginal improvement (i.e., similarity to humans) of the algorithm?

Method

Stimuli and data set

The data consisted of 34 data files of binocular data from 17 different students at Lund University. The used data files are a pseudo-random (picked blindly by a human) selection of a larger set of recorded data files. The data were recorded at 500 Hz and spanned a total of 103,872 samples. A sample refers to a time-stamped (x,y) coordinate pair. In order to increase the number of coded data files, only data from the right eye were used. The data came from three different types of stimuli (abbreviation later used are inside parentheses): photographies (img; 63,849 samples), linearly moving targets in the form of a dot (dot; 10,994 samples), and a real-world video of moving targets such as a rollercoaster and dolphins (vid; 29,029 samples). The instructions were to freely view the images, look at the moving objects in the videos, and to follow the moving dot targets.

The data were recorded using a tower-mounted Hi-Speed 1250 system, from SensoMotoric Instruments GmbH (Teltow, Germany), which minimizes head movements using a chin- and forehead rest. All data were recorded during one session in one of the experiment rooms of the Lund University Humanities Lab. Average gaze offset for the participants according to a 4-point validation procedure was 0.40° (note that offset is not important as neither algorithms nor human coders can see the stimuli, only the raw data). Precision was estimated by measuring the root-mean-square deviation (RMSD) of the x, y coordinates of samples identified by both human experts to be fixations. This resulting precision value was 0.03° RMSD. An overview of the participants, their contributing data, and there quality values can be seen in Table 11 in Appendix A.

No explicit filters were used, except the "bilateral filter," which is the default filter for this system, active at the recording stage. The filter "preserves the edges of large changes in the signal while averaging small changes caused by noise" (p. 310, Sensomotoric Instruments, 2009), and introduces no latency in the signal.

Human evaluation procedure and human "parameters"

The two human coders, 10 and 11 years of experience of working with eye movements, come from the same eye-tracking lab, but have slightly different backgrounds. Coder MN has a primarily worked with video compression and the evaluation thereof using eye-tracking, but he has also been involved in the design of the NH algorithm in this evaluation. Coder RA has a background primarily in psycholinguistics, and has not been involved in the design of any algorithm at the time of the data coding.

The two coders labeled the data samples manually and separately, using the same Matlab GUI. This GUI, shown in Fig. 1, contains several panels that together show, for a stretch of data in time, the current *x*- and *y*- positions (in pixels), the velocity (point-to-point, in degrees per second), and scatterplot representations of the data. Additionally, the GUI also showed a zoomed in portion of the sample positions, as well as a one-dimensional representation of the pupil size across time. Although the data was recorded binocularly, the human coding used only the right eye. Because the manual coding was very labor-intensive, we prioritized coding new material rather than a second eye for the same material.

The coders did not agree on any particular strategy for identifying the event, other than what events to identify and that they should have no information about the type of stimuli used for the particular data stream. This approach was intentional, and the rationale behind it was that each coder should be used as, to the extent it was possible, an independent expert, and have no more information available than the algorithms. Had we more strictly agreed on guidelines for how to identify events and code border-cases, then the agreement would be higher, but artificially so as the agreement would be less likely to generalize to a member outside of the raters.

Furthermore, a coding approach with two "naïve experts" is a one-time situation, that after the resulting classifications are shown and discussed, can never be recreated by the same coders. Together, this motivated an approach with little initial discussion about the details of the coding process.

The only thing the two coders agreed on was the events to identify in the data streams. These were: fixation, saccade, post-saccadic oscillation (PSO), smooth pursuit, blink, and undefined. The last event is a catch-all for any odd event that did not fit with the pre-defined events. In retrospect, this event was hardly ever used. These events were agreed on because they represent the most common events, except the PSO, which was detected because it is of special interest to our research group and has recently been the object of interest for two new algorithms (Larsson, Nyström & Stridh, 2013; Nyström & Holmqvist, 2010).

Furthermore, in order to get algorithm parameter values that were as equal as possible to the human decision criteria, or "human parameters," we extracted the empirical minimum fixation duration, maximum fixation dispersion threshold, and minimum saccade velocity from the files coded by the humans. These three parameters were selected as they are commonly understood by most eye-tracking researchers and

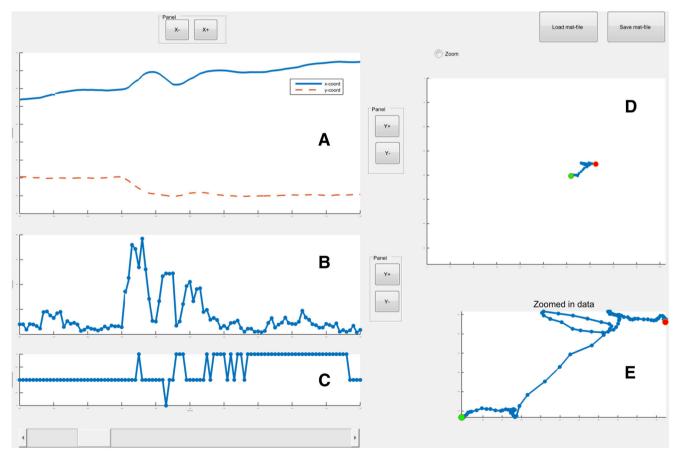**Fig. 1** The Matlab graphical user interface for hand-coding events on a sample-by-sample level. It provides curves of the (x,y) coordinates (**A**) and the gaze velocities (**B**), as well as drawing the data in the windows as a trace in a coordinates system matching the dimensions of the stimuli (**D**). Two windows also show the current segment of data zoomed in (**E**), and the vertical pupil diameter (**C**)

constitute the minimum parameters needed to be set for the different algorithms in this evaluation. As minimum and maximum values are sensitive to outliers and errors, we manually inspected these distributions of parameter values.

For fixation durations, we visually identified a small subset of fixations with durations between 8 and 54 ms, and the nearest following fixation duration was 82 ms. We therefore selected a minimum fixation duration of 55 ms to exclude this subset of very short fixations with durations outside the frequently defined ranges (Manor & Gordon, 2003). This value is also supported by previous studies (Inhoff & Radach, 1998; see also Fig. 5.6 in Holmqvist et al., 2011).

The maximum fixation dispersion for the human coders was, after removing one extreme outlier at 27.9°, found to be 5.57°. This was clearly more than expected, especially for data of this level of accuracy and precision, and the distribution had no obvious point that represented a qualitative shift in the type of fixations identified. However, after 2.7°, 93.95 % of the fixations values have been covered and the tail of the distribution is visually parallel to the x-axis. Thus, any selected value between 2.7 and 5.57 would be equally arbitrary. So, as even a dispersion threshold of 2.7 would be considered a generous threshold, we decided to not go beyond this and simply set the maximum dispersion at 2.7°. Do note that we used the original Salvucci and Goldberg (2000) definition of dispersion ($y_{max} - y_{min} + x_{max} - x_{min}$) which is around twice as large as most other dispersion calculations (see p. 886, Blignaut, 2009). The dispersion calculation for the humans was identical to the one implemented in the evaluated IDT algorithm.

For minimum saccadic velocity, we found no visually obvious subsets in the data. As the minimum peak saccadic angular velocity (unfiltered) in the distribution was 45.4°/s, a value in range with what to be expected from unfiltered values, we decided to keep this as is (Sen & Megaw, 1984).

To summarize, the parameter values extracted from the human expert coders are presented in Table 2.

Algorithm parameters

These algorithms have been used "as is" with only minimal changes in order to fit them into our testing framework. Thus, we have preferred implementations with some post-processing implementations over "bare" event detection

algorithms which would have required our tampering with the original code.

We did the modification expected of an intermediate user. That is, any person motivated enough to download a third-party implementation of an eye-movement classifier is also motivated to set obvious parameters that are relevant to this person's system. However, this person will only set the obvious parameters and not tweak every possible variable. That is, no algorithm was optimized with a full walk through the parameter space, but obvious mismatches in parameter selection should be avoided. One obvious drawback with optimizing the parameters of each algorithm is the risk of over-fitting the algorithms to this particular evaluation data. Although this could be mitigated by dividing our expert-coded data, such manually coded data is too precious for this approach. A second drawback is that we are giving an unfair advantage to non-adaptive algorithms, which then become "adaptive" by our tweaking. Thirdly, optimizing parameters would favor algorithms with a large number of parameters, which can then be perfectly fit to match our data, rather than data in general. An ideal algorithm would require no parameter tweaking from the user, but rather set thresholds automatically and then exhaustively classify all the samples of the data stream. Examples of parameters we did set are geometry parameters such as screen size, sampling frequency, or filter window size (if measured in samples or real time), *if* these are clearly indicated by comments in the code. Furthermore, we have set minimum fixation duration, maximum fixation dispersion, and minimum saccade velocity, according to our parameters derived from the human coders. By setting the parameters equal to the human parameters, we are giving the algorithms a fair chance to be similar to the human experts, without optimally tuning the algorithms. Also, these last three parameters are variables that we believe the average eye-tracking researcher is familiar with and so could set herself.

For reproducibility, we briefly report all used parameter values for each algorithm.

Fixation dispersion algorithm based on covariance (CDT)

We used the default .05 α significance level of the F-test, but changed the window size from six samples (for their 240 Hz system, which was equivalent to 25 ms) to 13 samples to match our 500 Hz system (equivalent to 26 ms).

Engbert and Mergenthaler, 2006 (EM)

The parameter specifying how separated the saccade velocity should be from the noise velocity, λ, was kept at the default value of 6. We used the type 2 velocity calculation recommended in the code. Minimum saccade duration (in samples) was also kept at the default value, which was three samples (equivalent to 6 ms), as both Engbert and Kliegl (2003) and

Table 2 Empirical human parameters

Parameter	Min	Max	Used
Minimum fixation duration (ms)	8	4428	55
Maximum fixation dispersion (°)	0.17	28.0	2.7
Minimum saccade velocity (°/s)	45.4	1096	45.4

Engbert and Mergenthaler ([2006](https://)) used three samples despite using data sampled at different rates (equivalent to 12 and 6 ms, respectively).

Identification by dispersion threshold (IDT)

We used the minimum fixation duration (55 ms) and maximum fixation dispersion (2.7°) as extracted from our human experts. The dispersion was calculated in exactly the same way in both cases. The original default values for this implementation were 100 ms minimum fixation duration and 1.35° maximum fixation dispersion.

Identification by Kalman filter (IKF)

The parameters for this algorithm was set using the GUI default, which gave us a chi-square threshold of 3.75, a window size of five samples, and a deviation value of 1000.

Identification by minimal spanning tree (IMST)

The saccade detection threshold was by default set to 0.6°, and the windows size parameter was set to 200 samples. There were the default values from their GUI.

Identification by hidden Markov model (IHMM)

For the IHMM, we set the saccade detection threshold to 45°/s, and used their GUI default for the two other parameters (Viterbi sample size = 200; Baum-Welch reiteration = 5).

Identification by velocity threshold (IVT)

This algorithm uses only one parameters, the velocity threshold for saccade detection, and this was set in congruence with humans and other algorithms, i.e. 45°/s.

Nyström and Holmqvist, 2010 (NH)

For this algorithm we used all the default values, except the minimum fixation duration, which was set using our human-extracted values (55 ms).

Binocular-individual threshold (BIT)

We changed the original three sample minimum fixation duration (equivalent to 60 ms at 50 Hz in the original study) to 28 samples (resulting in 56 ms, which is approximately equivalent to the human-derived threshold of 55 ms). All other parameters were left at default values.

Larsson, Nyström and Stridh, 2013 (LNS)

For this algorithm, we did not set any parameters at all and used all the default hard-coded values. The two most relevant values were the minimum time between two saccades and the minimum duration of a saccade, which were kept at 20 ms and 6 ms, respectively.

Evaluation procedure

The algorithms were evaluated based on their similarity to human experts in classifying samples as belonging to a certain oculomotor event. This was done in two ways: first, comparing the event duration parameters (mean, standard deviation, and number of events), and second, to the overall sample-by-sample reliability via the Cohen's Kappa metric. The first evaluation answers how the duration properties of the actual eye-movement events changes as a result of an algorithm. The second evaluation provides a statistic on how well the algorithms and humans agree in their classifications on the actual samples. Finally, a third analysis investigates what sample classification errors each algorithm produced compared to the humans. For example, whether a certain algorithm consistently under-detects fixations in favor of saccades, and thus in turn inflates saccade durations. Such systematic biases in a certain direction are important to highlight to both researchers and algorithm designers. This will also highlight what area each algorithm needs to improve in order to gain the greatest classification improvement.

Similarity by event duration parameters

The primary problem of event detection algorithms is that the event durations, such as fixation durations, vary depending on the choice of algorithm and settings. Thus, the output that is the most intuitive to compare are the distributions of the event durations. That is, how many events that are detected, the mean duration of these, and the standard deviation of the durations. Algorithms that work in an identical fashion should also produce identical results for these three parameters. However, an algorithm may achieve similar mean durations as a human expert, but detect a different number of events. Another algorithm could detect the identical number of events, but differ in the detected mean duration. To join these three parameters to a single similarity measure, we calculated the root-mean-squared deviations (RMSD) for all algorithms against the two human coders. A single similarity measure is needed if we are to rank the algorithms and provide some general claim that one is better than another.

First, the evaluation parameters (mean duration, standard deviation, and number of events) were rescaled to

the [0, 1] range according to Eq. 1. Here, M is the matrix consisting of the algorithms and humans, and their resulting distributions parameters, as rows (k) and columns (l) respectively.

$$(M_{kl}) \tag{1}$$

The normalized data is then separated into a matrix for the algorithms, A, and a matrix for the human experts, H. Then, the summed RMSD for the column of algorithms, a, was calculated as follows:

$$a^i_{RMSD} = \sum_{\forall j} \sqrt{\left(A_{ij} - \sum_m \frac{H_{mj}}{n}\right)^2} \tag{2}$$

where i is the algorithm index, j is the index for the event distribution parameter, m is the index for the human experts, and n is the number of human experts.

The algorithm with event parameters most similar to the humans experts, i.e., the "winner" (a_w), was then simply the algorithm having the minimum RMSD value produced by Eq. 2. It should be noted that this RMSD value is used to rank the similarity, but the absolute values do not guarantee some minimum level of similarity which warrant binary terms like "similar" or "different." Because each parameter property is scaled against its maximum value, an RMSD values from one set of comparisons are not transferable to another set of comparisons. Thus, the rank and distance to the next rank is of interest, and not the absolute value.

Similarity by Cohen's Kappa

The similarity between human and algorithm performance was also evaluated using Cohen's Kappa (Cohen, 1960). This measure is simply concerned with agreement on a sample-by-sample basis, ignoring effects of event durations (which can be seen as uninterrupted chains of samples with the same label). The metric is calculated as in Eq. 3, where P_o is the observed proportion of agreement of coders a and b for the samples n, and P_c is the proportion of chance agreement between the coders given their proportion of accepting (1) or rejecting (0) the label for the ith sample. The chance agreement reflects the level of agreement achieved if two coders selected events at random but followed their inherent bias for certain events, e.g., 90 % of event A and 10 % of event B. This metric ranges from -infinity to 1, where negative numbers indicate a situation where the chance agreement is higher than the observed agreement, i.e. the coders are worse than chance. A zero indicates the case where the observed and the chance agreement are identical. A perfect one (1) indicates a (theoretical) situation where the chance agreement is exactly zero (impossible to agree at all by chance) and the two coders are in perfect agreement of all events.

$$K = \frac{P_o - P_c}{1 - P_c} \tag{3}$$

Confusion analysis

To answer the question how algorithm designers should receive the best marginal improvement of their algorithm, we calculated confusion matrices for all algorithms against the human experts. A confusion matrix, C, can be described as a symmetrical matrix with sides equal to the size of the set of classification codes c. Two raters, a and b, then independently classify each sample in the data set (of size n) using codes i and j, respectively, and increment the value of the cell C_{ij} in the confusion matrix. A perfect agreement, i.e. no confusion, would result in a diagonal of 1 (if normalized), and 0 in every other cell.

The number of pair-wise comparisons is very large and not possible to exhaustively report in this article. Instead, we have collapsed the full confusion matrices into simpler matrices showing what events each algorithm, when they disagree with human experts, they over- or under-classify. For example, an algorithm that can detect saccades, but not fixations, will definitely under-classify samples as fixations, and so a major improvement could be achieved by adding fixation-detection capabilities. Also, any algorithm that detects events sequentially is likely to under-classify events that are detected later in this chain, unless the algorithm has the functionality to roll back previous classifications.

Results

Event durations per algorithm and stimuli type

The number of fixations and saccades vary dramatically with the algorithm and the type of stimuli used, as is evident from Tables 3 and 4, as well as visualized in Fig. 2.

According to the minimized root-mean-squared deviations against human experts (RMSD within parentheses, lower is better), the fixation detection algorithm most similar to human experts for image data was NH (0.36), with IMST (0.54) as runner-up. For moving dot stimuli, the winning fixation detector was the BIT (0.73) algorithm, with IMST (0.81) as runner-up. For video stimuli, IKF (0.68) was the most similar algorithm, then NH (0.78).

The most human-like saccade detection algorithm for image data was LNS (0.23), with IDT (0.49) in second place. For moving dots, LNS (0.23) was the winner, with IVT (0.97) as runner-up. With video data, LNS (0.28) was the winner and the runner-up was IDT (0.72).

Only two algorithms detected post-saccadic oscillations, and how they fared against human coders is shown in Table 5.

The algorithm most similar to humans experts when comparing post-saccadic oscillations was LNS (0.99, 1.12, 1.15),

Table 3 Fixation durations

Algorithm	Images				Moving dots				Videos			
	Mean	SD	No.	RMSD	Mean	SD	No.	RMSD	Mean	SD	No.	RMSD
CoderMN	248	271	380	<0.1	161	30	2	0.2	318	289	67	0.2
CoderRA	242	273	369	<0.1	131	99	13	0.2	240	189	67	0.2
CDT	397	559	251	2.3	60	127	165	1.6	213	297	211	1.0
EM	-	-	-	-	-	-	-	-	-	-	-	-
IDT	399	328	242	1.4	323	146	8	1.1	554	454	48	1.5
IKF	174	239	513	0.9	217	184	72	1.1	258	296	169	0.9
IMST	304	293	333	0.5	268	140	12	0.8	526	825	71	1.9
IHMM	133	216	701	1.6	214	286	67	1.5	234	319	194	0.9
IVT	114	204	827	2.1	203	282	71	1.4	202	306	227	1.1
NH	258	299	292	0.4	380	333	30	2.1	429	336	83	0.8
BIT	209	136	423	0.8	189	113	67	0.7	248	215	170	0.9
LNS	-	-	-	-	-	-	-	-	-	-	-	-

Note. Fixation durations for the different algorithms and stimuli types. Algorithms with dashes as values do not detect fixations. The algorithms presented are, in order: Coder MN, Coder RA, Fixation Dispersion Algorithm based on Covariance (CDT), Engbert & Mergenthalser (EM), Identifiacation by Dispersion-Threshold (IDT), Identification by Kalman Filter (IKF), Identification by Minimal Spanning Tree (IMST), Identification by Hidden Markov Model (IHMM), Identification by Velocity Threshold (IVT), Nyström & Holmqvist (NH), Binocular-Individual Threshold (BIT), and Larsson, Nyström, & Stridh (LNS)

with NH (2.24, 2.44, 2.16) in second (last) place. This order was the same for all three stimuli types.

For completeness, the events only detected by the human coders are listed in Table 6.

Algorithm – human sample-by-sample similarity

We compared the similarity of the algorithms to the human coders using Cohen's Kappa (higher is better). The results are presented in Table 7.

Starting with fixations, for the image data the IHMM, IVT, and BIT were the best ones, achieving very similar Kappa scores. For the moving dot data, no algorithm fared well, but the algorithm performing the least poorly was the CDT. For video data, the IKF and BIT were the best algorithms. For saccades, the LNS and the IVT were the two best algorithms for image data. For moving dots, the LNS and the EM algorithm were the best ones. For video data, LNS was best and IVT was the runner-up when detecting saccades. Considering post-saccadic oscillations, only two

Table 4 Saccade durations

Algorithm	Images				Moving dots				Videos			
	Mean	SD	No.	RMSD	Mean	SD	No.	RMSD	Mean	SD	No.	RMSD
CoderMN	30	17	376	<0.1	23	10	47	0.4	26	13	116	0.1
CoderRA	31	15	372	<0.1	22	11	47	0.4	25	12	126	0.1
CDT	-	-	-	-	-	-	-	-	-	-	-	-
EM	25	22	787	1.5	17	14	93	1.4	20	16	252	1.6
IDT	25	15	258	0.5	32	14	10	1.3	24	53	41	0.7
IKF	62	37	353	2.1	60	26	29	2.4	55	20	107	2.1
IMST	17	10	335	0.8	13	5	18	1.3	18	10	76	0.9
IHMM	48	26	368	1.0	41	17	27	1.3	42	18	109	1.4
IVT	41	22	373	0.6	36	14	28	1.0	36	16	112	0.9
NH	50	20	344	0.9	43	16	42	1.0	44	18	1104	1.5
BIT	-	-	-	-	-	-	-	-	-	-	-	-
LNS	29	12	390	0.2	26	11	53	0.2	28	12	122	0.3

Note Saccade durations for the different algorithms and stimuli types. Algorithms with dashes as values do not detect fixations
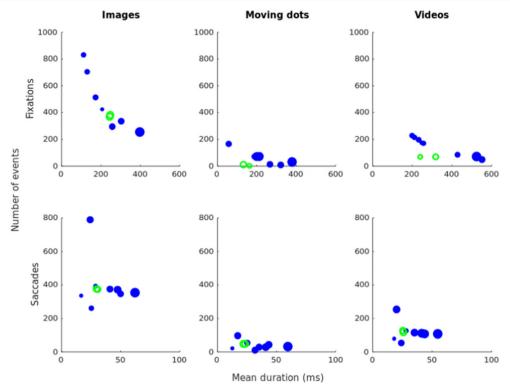
Fig. 2 Visualized similarity in events produced by the classifications from the different algorithms (blue, filled) and humans experts (green, unfilled). The ordinate shows the number of events, the abscissa the mean duration in milliseconds of these events, and the radius of each bubble shows the relative (within each panel) standard deviation of the durations. Note that not all algorithms are plotted for all events, as not all algorithms detection fixations or saccades (see Table 1)

algorithms can detect that event, and LNS outperforms NH decisively.

Across all stimuli types, a human expert was always better at matching the classifications of the other human than any algorithm was at matching the average of the two humans. Considering the detection across all events for image-viewing data, fixation detection algorithms achieved a high inter-rater reliability as the data contained mostly fixations and saccades, and post-saccadic oscillations were small compared to the other events. When presented with data from almost exclusively dynamic stimuli, these fixation detectors do not perform above chance. The video-viewing data, however, represent a more natural blend of dynamic and static stimuli, and here the algorithms were clearly not matching

the reliability of the humans, although they were better than chance.

Confusion analysis: Images

The confusion analysis reveals how each algorithm (or human expert) over- and under-classifies certain events in comparison to the human experts. Considering the human CoderRA, for example, this person under-classifies samples as fixations in comparison to the human CoderMN. As the samples that are under-classified as fixations must be classified as something else, we see that CoderRA tends to over-classify samples as smooth pursuit instead. However, the two humans agree to a

Table 5 Post-saccadic oscillation durations

Algorithm	Images				Moving dots				Videos			
	Mean	SD	#	RMSD	Mean	SD	#	RMSD	Mean	SD	#	RMSD
CoderMN	21	11	312	0.2	15	5	33	0.4	20	11	97	0.7
CoderRA	21	9	309	0.2	15	8	28	0.4	17	8	89	0.7
NH	28	13	237	2.2	24	12	17	2.5	28	13	78	2.2
LNS	25	9	319	1.0	20	9	31	1.1	24	10	87	1.2

Note. PSO durations for the different algorithms and stimuli types. Algorithms with dashes as values do not detect fixations

Table 6   Miscellaneous event durations

	Images				Moving dots				Videos			
Algorithm	Mean	SD	#	RMSD	Mean	SD	#	RMSD	Mean	SD	#	RMSD
MN pursuit	363	187	3	2.7	375	256	37	2.1	521	347	50	2.2
RA pursuit	305	184	16	2.7	378	364	33	2.1	472	319	68	2.2
MN blink	335	153	20	1.47	336	0	1	0.5	297	189	3	2.0
RA blink	392	237	19	1.47	212	0	1	0.5	187	31	3	2.0

Note PSO durations for the different algorithms and stimuli types. Algorithms with dashes as values do not detect fixations
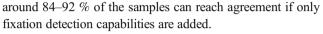
large extent, only disagreeing on 7 % of the data from image viewing, as indicated by the ratio column in Table 8.

Turning our attention to the algorithms and still considering the image-viewing data, the algorithms that detect fixations performed distinctly better than the two algorithms that do not (EM & LNS). A fixation-detecting algorithm disagreed on at most 24 % of the samples, compared to algorithms that do not detect fixations (at best 84 % disagreement). The fixation-detecting algorithms vary in their behavior. For example, the IDT over-classifies samples as fixations and rarely under-classifies the samples. However, the IHMM algorithm is much more balanced in its errors, over-classifying samples as fixations as much as it underclassifies them. In general, however, a large chunk of the samples where they disagree with humans are under-classified as fixations. Looking at the events that are not detected by the algorithms, it seems around 30 % of the disagreeing samples can reach agreement with the humans if classification support for PSO and smooth pursuit is implemented. For algorithms that only detect saccades,

around 84–92 % of the samples can reach agreement if only fixation detection capabilities are added.

To visualize how the different coders and algorithms classify an image-viewing trial, and what mistakes they make, the raw positional data along with the classifications as scarf plots are shown in Fig. 3.

Table 8   Confusion matrix for data from images

Algorithm	Ratio	Error	Fix	Sacc	PSO	SP	Blink	Other
coderMn	7 %	over	.68	.09	.18	<.01	.02	.03
		under	.13	.13	.17	.46	.11	<.01
coderRA	7 %	over	.13	.13	.17	.46	.11	<.01
		under	.68	.09	.18	<.01	.02	.03
CDT	23 %	over	.66	.00	.00	.00	.00	.34
		under	.04	.38	.22	.10	.25	<.01
EM	92 %	over	.00	.08	.00	.00	.00	.92
		under	.84	.01	.06	.03	.06	<.01
IDT	20 %	over	.80	.09	.00	.00	.00	.11
		under	.05	.27	.26	.12	.29	<.01
IKF	24 %	over	.18	.39	.00	.00	.00	.43
		under	.40	.03	.22	.10	.25	<.01
IMST	20 %	over	.78	.03	.00	.00	.00	.18
		under	.06	.26	.26	.12	.29	<.01
IHMM	20 %	over	.30	.29	.00	.00	.00	.41
		under	.27	.03	.27	.12	.30	<.01
IVT	19 %	over	.33	.20	.00	.00	.00	.47
		under	.26	.04	.27	.12	.30	<.01
NH	32 %	over	.08	.18	.12	.00	.00	.63
		under	.59	.03	.12	.08	.18	<.01
BIT	31 %	over	.12	.00	.00	.00	.00	.88
		under	.28	.29	.17	.08	.19	<.01
LNS	84 %	over	.00	.02	.03	.00	<.01	.95
		under	.92	.02	.02	.03	<.01	<.01

Note. Proportion of samples classified in disagreement with expert coders, for the image stimuli. The Ratio column indicates the proportions of samples, out of all classified samples, where the algorithm disagreed with the humans, or in the humans case how much one human disagreed with another. The columns for the different events show what proportion, out of the disagreeing samples, that can be explained as over- or under-classification of that particular event
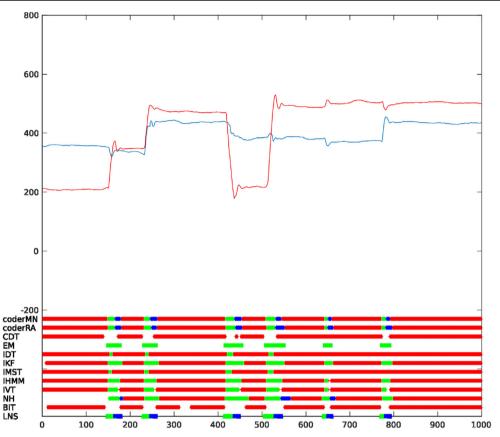
Table 7   Cohen's Kappa reliability between algorithms and human coders

	Fixations			Saccades			PSOs		
Algorithm	Img	Dots	Vid	Img	Dots	Vid	Img	Dots	Vid
CoderMN	.92	.81	.83	.95	.91	.94	.88	.82	.83
CoderRA	.92	.84	.82	.95	.91	.94	.88	.80	.81
CDT	.38	.06	.11	.00	.00	.00	.00	.00	.00
EM	.00	.00	.00	.64	.66	.67	.00	.00	.00
IDT	.36	.00	.03	.45	.26	.38	.00	.00	.00
IKF	.63	.03	.14	.58	.46	.59	.00	.00	.00
IMST	.38	.00	.03	.54	.30	.52	.00	.00	.00
IHMM	.67	.03	.13	.69	.60	.71	.00	.00	.00
IVT	.67	.03	.13	.75	.63	.76	.00	.00	.00
NH	.52	.00	.01	.67	.60	.68	.24	.20	.25
BIT	.67	.03	.14	.00	.00	.00	.00	.00	.00
LNS	.00	.00	.00	.81	.75	.81	.64	.59	.63

Note. Fixation, Saccade, and PSO agreement between algorithms and human coders, expressed in Cohen's Kappa. Negative values are set to zero. Higher is better

Fig. 3 Positional data from the first 1,000 samples of an image-viewing trial. The (x,y) coordinates are plotted as position over time in blue and red, respectively. The classification of the coders and the algorithms are plotted as scarf plots below, with fixations in red, saccades in green, and PSOs in blue. Absence of color (white) means the sample was not classified by the algorithm. The x-axis is in samples

Confusion analysis: Moving dots

For the data from the moving dot stimuli, we find that humans disagree on 11 % of the data, and the algorithms disagree on at least 84 % of the data. The SP column in Table 9 reveals that this is largely driven (81–92 %) by the inability of the considered algorithms to detect smooth pursuit.

For the algorithms that only detect fixations and saccades, smooth pursuit motion is primarily misclassified as fixation data. The net over-classification of fixations is likely from the smooth pursuit motions, and at least 52 % of the misclassified fixation samples can be recovered by adding smooth pursuit support to the algorithms. The over-classifications in the Other category are primarily samples that the algorithm could not classify at all, i.e. not meeting the criteria of any known events. It is preferable that the smooth pursuit samples end up here, rather than being falsely accepted as fixation samples.

Confusion analysis: Video

Finally, in Table 10, the data from the natural video stimuli shows that the humans now disagree on 19 % of the samples. Again, we see a distinction between the algorithms that can detect fixations and the ones that cannot. The prevalence of fixations in natural videos allows the fixation detectors to disagree on fewer samples than what they did during the moving dot stimuli. The non-fixation detectors are particularly punished by not detecting fixations.

Another obvious issue is that some algorithms do not classify samples that do not match a particular category, whereas others revert to some default category, as can be seen in the Other category in Table 10. For video stimuli, implementing smooth pursuit detection capabilities should be prioritized to recover the largest number of misclassified samples.

Discussion

The purpose of this study has been to find the best event detection algorithm to recommend to researchers. This was done by evaluating the performance of ten different event classifier algorithms for eye-movement data, and examining how they compare to human evaluators. Researchers use algorithms such as these, sometimes seemingly mindlessly as they are often tightly integrated in the eye-tracking software. The study controlled parameter differences between algorithms using a combination of sensible default values and reverse-engineered human-implicit parameters. Even though

Table 9 Confusion matrix for data from moving dots

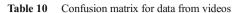| Algorithm	Ratio	Error	Fix	Sacc	PSO	SP	Blink	Other
coderMn	11 %	over	.11	.09	.08	.64	.05	.04
		under	.59	.06	.06	.27	.00	.01
coderRA	11 %	over	.59	.06	.06	.27	.00	.01
		under	.11	.09	.08	.64	.05	.04
CDT	89 %	over	.67	.00	.00	.00	.00	.33
		under	.02	.05	.02	.87	.01	.02
EM	96 %	over	.00	.03	.00	.00	.00	.97
		under	.14	.01	.02	.81	.01	.01
IDT	86 %	over	.98	.01	.00	.00	.00	.02
		under	<.01	.05	.02	.90	.01	.02
IKF	85 %	over	.81	.06	.00	.00	.00	.14
		under	.02	.02	.02	.91	.01	.02
IMST	86 %	over	.98	<.01	.00	.00	.00	.01
		under	<.01	.05	.02	.90	.01	.02
IHMM	84 %	over	.98	.02	.00	.00	.00	.09
		under	<.01	.02	.02	.92	.01	.02
IVT	84 %	over	.89	.02	.00	.00	.00	.09
		under	.01	.02	.02	.92	.01	.02
NH	93 %	over	.64	.05	.02	.00	.00	.30
		under	.12	.01	.02	.83	.01	.02
BIT	89 %	over	.72	.00	.00	.00	.00	.28
		under	.03	.05	.02	.87	.01	.02
LNS	93 %	over	.00	.02	.01	.00	.01	.95
		under	.14	.01	.01	.83	<.01	.01

Note. Proportion of samples classified in disagreement with expert coders, for the moving dot stimuli. The Ratio column indicates the proportions of samples, out of all classified samples, where the algorithm disagreed with the humans, or in the humans case how much one human disagreed with another. The columns for the different events show what proportion, out of the disagreeing samples, that can be explained as over- or under-classification of that particular event

Table 10 Confusion matrix for data from videos

Algorithm	Ratio	Error	Fix	Sacc	PSO	SP	Blink	Other
coderMn	19 %	over	.72	.03	.07	.15	.04	<.01
		under	.16	.04	.04	.76	.00	<.01
coderRA	19 %	over	.16	.04	.04	.76	.00	<.01
		under	.72	.03	.07	.15	.04	<.01
CDT	64 %	over	.80	.00	.00	.00	.00	.20
		under	.02	.08	.05	.82	.02	<.01
EM	95 %	over	.00	.04	.00	.00	.00	.96
		under	.40	<.01	.03	.55	.03	<.01
IDT	61 %	over	.98	.01	.00	.00	.00	.01
		under	<.01	.06	.05	.86	.03	<.01
IKF	62 %	over	.74	.09	.00	.00	.00	.18
		under	.07	.01	.05	.85	.03	<.01
IMST	61 %	over	.97	.01	.00	.00	.00	.03
		under	.01	.05	.05	.86	.03	<.01
IHMM	59 %	over	.83	.05	.00	.00	.00	.12
		under	.03	.01	.05	.88	.03	<.01
IVT	59 %	over	.84	.04	.00	.00	.00	.12
		under	.03	.01	.05	.88	.02	<.01
NH	70 %	over	.58	.05	.04	.00	.00	.33
		under	.18	.01	.03	.75	.03	<.01
BIT	67 %	over	.66	.00	.00	.00	.00	.35
		under	.07	.08	.04	.78	.03	<.01
LNS	92 %	over	.00	.01	.02	.00	<.01	.97
		under	.41	.01	.01	.57	<.01	<.01

Note. Proportion of samples classified in disagreement with expert coders, for the video stimuli. The Ratio column indicates the proportions of samples, out of all classified samples, where the algorithm disagreed with the humans, or in the humans case how much one human disagreed with another. The columns for the different events show what proportion, out of the disagreeing samples, that can be explained as over- or under-classification of that particular event

we expected some variance, we were completely surprised to find that the choice of an algorithm produced such dramatic variation in the event properties, even using identical and otherwise default settings.

In order to make sense of the results, let us revisit our aims from the beginning.

Which is the best algorithm?

Interestingly, it is not quite as simple as that. Considering fixation detection in static stimuli, perhaps the most common form of event detection, the NH algorithm was the most similar to the human experts in terms of matching the number of events and their durations. However, when we considered sample-by-sample comparisons, the IHMM, IVT, and BIT algorithms performs the best and with similar scores. If you have access to binocular data, then BIT could possibly perform even better. Interestingly, when NH ranks well, IHMM

ranks poorly, and vice versa, depending on the evaluation approach. We will discuss this in more detail shortly. For detecting fixation in dynamic stimuli, the algorithms are not near the human experts, leading us to conclude that there are no winners in these contexts.

Concerning saccades and post-saccadic oscillations, the LNS algorithm, however, was consistently the most suitable choice, no matter the underlying stimuli. This was supported by both the event duration distributions, and the sample-by-sample comparison.

The answer to this research question is further elaborated on in the following sections.

How does the number and duration of events change depending on the algorithm?

Considering detecting fixations in data from image-viewing, a very common type of event-detection, the average fixation

duration varied with a factor of four due to the algorithm choice. Assuming the "average human"' classification was true, then for the same data, the fixation duration estimates deviated from the true value by up to a factor of three. For dynamic stimuli, the fixation durations differed with a factor of nearly eight and a factor of nearly three from the human experts.

For saccades, the corresponding algorithm differences were a factor of four for both static and dynamic stimuli. For differences against the humans experts, static stimuli produced a factor of two, and dynamic stimuli a factor of three.

How similar are algorithm-detected events to human-detected events?

We estimated the similarity of the events detected by algorithms and humans using the root-square-mean deviations of the unweighted combination of the number of events, the mean duration (i.e., how many samples were included in the event) and the standard deviation of these events. The algorithm that minimized the deviation was considered the most similar to the human experts. This value varies theoretically from zero (no deviation at all) to three (maximal deviation for all three event properties).

We found that for detecting fixations in data from static stimuli, NH was the most similar with a deviation of 0.36, and the closest alternative was at 0.54. For the dynamic stimuli types, the deviation increased to 0.81 and 0.98, respectively, showing that the algorithms have a harder time properly detecting fixation when the data contain smooth pursuit.

For saccades, the LNS was clearly more similar to the humans than the second closest algorithm, even across different stimuli types (RMSD: 0.23, 0.23, 0.28, vs. 0.49, 0.97, 0.72, for image, dot, and video, respectively).

How similar are algorithms to humans in a simple sample-by-sample comparison?

The sample-by-sample comparison in the form of Cohen's K evaluates the classification of each sample independently, and it also adjusts for the base rate of each event type. The evaluation is thus a matter of correctly classifying each sample, but also doing so better than a random guess given knowledge of the proportion of samples from each event.

For detecting fixations from static stimuli, the algorithms with fixation detection capabilities fared reasonably well (K values between .36 and .67 compared to the humans .92). For the stimuli with moving dots, they were almost indistinguishable from chance (.00 to .14). For natural videos they performed slightly better but not well (.01 to .14).

Detecting saccade samples appeared easier than detecting fixations. The average K scores for image, dot, and video data, respectively, were in the ranges .45–.81, .26–.75, and .38–.81,

compared to human experts (.95, .91, .94). Thus saccade detection even in data from dynamic stimuli appeared to be not completely improper, at least for sample-by-sample evaluation.

Are the human experts interchangeable?

Evaluations against manual classifications such as these typically only have a single human coder, as the coding process is very laborious. This raises the question whether the results are mainly determined by the human, rather than the algorithms. To explore this concern, we looked at the results against each coder in isolation, and against each other. If the results were consistent, then human biases would have been negligible.

Humans were the most similar to each other, if we evaluated them in the same way as the algorithms, i.e. against a hypothetical "average coder." This is perhaps completely obvious, as each human has contributed half of the data for this average coder. If we compare the human coders and algorithms to only one human coder at a time, then the same pattern with the humans remain: they are more similar to each other than any algorithm. The same pattern for the algorithm rankings remain when considering only one human coder at a time. For fixation-detection in data from static stimuli, the NH algorithm is the most similar candidate, with IMST as a runner-up. The same was also true for saccade detection: the LNS algorithm was consistently the most similar to any human, regardless of what underlying stimuli elicited the eye movement data.

The patterns also largely remained for the individual coders when we considered the sample-by-sample evaluation using Cohen's Kappa. Both coders were the most similar to each other, the BIT algorithm was the best fixation-sample classificator for static stimuli for coder RA. For coder MN the BIT was among the top three algorithms (IHMM, IVT, and BIT) with very similar scores. For both coders the LNS algorithm was the best saccade- and PSO-sample classificator for sample-by-sample classification.

We had an initial concern that coder MN had been involved in the development of two of the algorithms (which performed well), and this coder's particular view on event detection biased both the design of the algorithm and the coding process, leading to inflated agreement levels. However, since both coders shared the same top-ranking algorithms, it was clear that this design–coder contamination could not undermine our algorithm ranking results.

Are the algorithm–human similarities dependent on the underlying stimuli that elicited the data?

It was clear from our results that the stimuli that elicited the data determined the evaluation scores. Consistently, for both evaluation methods, we saw that data from static stimuli were
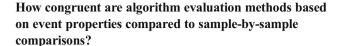
easier than dynamic stimuli (moving dots and natural videos). This was hardly surprising, as most algorithms only detected fixations and saccades, which are the predominant events of static stimuli viewing.

Naturally, humans are at an advantage, for several reasons. They can form a model of the underlying stimuli, and then make use this model in the coding process whenever the un-certainty is high. Humans can also code the full set of events, and change strategies if needed (e.g., to some backward-elimination process if forward positive identification appears difficult).

The evaluation accuracy was also a question of the type of event that was to be identified (see Table 7). Fixation-detec-tion, e.g., was almost indistinguishable from random chance for moving dots, but better than random chance for videos. Saccade-detection was more consistent across stimuli type, although this varied much between algorithms.

Are there consequences of using algorithms designed for static stimuli on data from dynamic stimuli?

If we ignore the question of evaluating similarity between humans and algorithms, and are simply interested in detecting fixations and saccades for our research, then what are the consequences? The answer is not straight-forward. Considering the RMSD, algorithms could completely change their ranking when going from one type of stimulus to the other. For example, NH (rank 3, just after the two humans) fell to rank 9 when going from image stimuli to moving dot stimuli, but back up again to rank 3 when proceeding to the video stimuli. One reason for this dramatic change is that the moving dot stimuli almost entirely consists of motion, which distorts the detected fixations if the algorithm does not expect smooth pursuit in the data. This fixation output would be nonsensical to use. The fixations detected by NH from the video-elicited data *look* like they are decent, but it is not ob-vious that they can be trusted. The deviation (RMSD) from the coders is clearly larger than for image-elicited data (0.36 vs. 0.81). When considering the scores from the sample-by-sample evaluation, the performance drops dramatically for NH from .52 (images) to .01 (videos). In other words, even if the event duration distributions for video-elicited data ap-pear roughly similar to human-classified events, the algo-rithms are not adding much value to the detection process, and the result would have been roughly similar had a simple proportional guess been made. Such a decent proportional guess can possibly be made by an algorithm with thresholds and criteria that coincide with the expected event measures. This would superficially look appealing, but there is no guar-antee that the samples are classified well, leading to improper onsets and offsets of the events. In other words, using an algorithm design for static stimuli data on dynamic stimuli data may result in almost nonsensical event classifications.

How congruent are algorithm evaluation methods based on event properties compared to sample-by-sample comparisons?

The previous discussion led us to the question of how much one evaluation method could say about another evaluation method. The short answer is that they answer different aspects. The durations, and their distributions, of different events rep-resent the level most close to researchers, i.e. measures such as fixation durations. Given the right data, it is possible to have algorithms performing at 100 % accuracy. However, this could then be more driven by the nature of the data rather than the quality of the algorithms. That is why a method such as Cohen's K, which adjusts for the base-rate of the events, is motivated.

Also, because the two methods focus on different aspects, it is possible for an algorithm to perform poorly in one aspect, but perform well in another one. For example, an event could be identified as a very long fixation by the human, but could have some noisy sample in the middle of the event which the human disregards as noise. The algorithm, however, interprets that noisy sample as some non-fixation sample, and terminates the fixation. This results in two, shorter, fixations. So the number of fixations and durations deviate decidedly, but in terms of a sample-by-sample comparison all the samples but that one noisy sample are coded in agreement with the human. This is what we found for the NH and BIT algorithms that we looked more closely at. BIT gets disrupted in the fixation detection, and produces chains of shorter fixations. Although it detects each sample well, and scores high on a sample-by-sample comparison, it performs poorly in matching the human-detected events in terms of duration, and number.

To conclude this section, the researcher must decide herself what aspect is relevant, and select the appropriate algorithm accordingly. Is the priority to get unbiased durations of fixa-tions or saccades, or is it to classify the largest amounts of sample correctly?

What are the most pressing areas in which to improve the algorithms?

The evaluated algorithms are not similar, so it is difficult to give general advice. However, we can distinguish between the saccade *or* fixation algorithms, and the algorithms that seem to have an ambition to detect all events. The largest improvement can be found when adding fixation-detection capabilities to a a saccade-only algorithm. Fixations are a very common and relatively long type of event, which consequently carries much weight in the data file (cf. Fig. 3). Thus, it is no surprise that these algorithms, although excellent at what they do (e.g., the LNS algorithm), get a poor overall score when considering the coding of the individual samples (see confusion matrices in Tables 8, 9, and 10). The lack of this capability explains

around 47–90 % of the disagreements, if we consider image and video stimuli where fixation events are common. The BIT algorithm is a fixation-only algorithm, and should benefit from having saccade-detection capabilities. We should note that we may have also underestimated the performance of this algorithm, as it is capable to taking binocular data into account. No other algorithm does this.

The second family of algorithms are the fixation- and saccade-detectors, which detect the two most common events, and achieve an OK overall detection accuracy. However, adding support for detecting smooth pursuit can provide great improvements. More than 83 % of the disagreeing samples can be explained by this, at least for stimuli with a large prevalence of smooth pursuit (see Table 9). For a more moderate prevalence, as in our natural videos, around 73 % seem to be the proportion of disagreeing samples that can be recovered. This improvement would not only have the benefit of detecting smooth pursuit, but would also improve the performance of fixation and saccade detection, as there is a clear treatment of this "no man's land" of the velocity curve, which causes some algorithms to assign slow pursuit movements to the fixation category. It remains to be seen, however, if adding smooth pursuit detection makes the algorithms worse off in their fixation and saccade detection. Our human coders, with no knowledge of the underlying stimuli, indicated smooth pursuit motion in data from static stimuli. If the humans make this error, then it is likely that algorithms will also make such errors. It should also be noted, as is evident from Fig. 3, that even saccade detection, even though it is a distinct movement, has room for improvement. Failure to detect a saccade often means (as is the case in Fig. 3) that two separate fixations are instead identified as a single large fixation. This will of course have consequences for the estimated fixation durations, for example.

Post-saccadic oscillations, however, are fairly small events occurring at the end of saccades. Considering the data from the confusion matrices, the proportion of misclassification driven by these samples normally falls below the misclassification proportion from fixations and saccades. Thus, it is better for a designer to improve the existing fixation and saccade detection than to add support for this new event. However, as can be seen in Fig. 3, the amplitude of the post-saccadic oscillation is more likely to trigger some algorithms than others, forcing a premature termination of the saccade. In this figure, the IDT and IMST algorithms have notably shorter saccade amplitudes than other algorithms. Also, it is visible that the CDT algorithm identifies a segment at the peak of the first oscillation as a small fixation. This happens for the IHMM and IVT algorithms as well, but is less clearly visible in the scarf plot (after the fifth identified saccade).

Considering the distribution of average event durations in Fig. 2, it seems a number of algorithms can improve simply by tweaking the settings, or adding automatic threshold adjustments. At least for the fixation durations from image viewing, it becomes apparent that the average fixation duration follows a negative exponential curve, where the duration of the fixations go up as the number of detected fixations go down. This is most likely due to smaller separate fixations merge as settings, like a dispersion threshold, become more inclusive. When the algorithm is inclusive, it captures more sampling and the resulting fixation has longer durations.

Are humans a gold standard?

On the one hand, humans seem to, across the board, agree with each other very well. For data from the different stimuli types (image, dot, and video), the resulting RMSD and the proportion of disagreeing samples, it is clear that, on average, the human experts are the most similar to each other. On the other hand, it is difficult to draw any far-reaching conclusions based on only two raters.

Currently, the humans are most similar to each other, but humans also make mistakes. However, once an algorithm reaches a level of accuracy on par with the humans, it becomes challenging to say whether the errors are driven primarily by the inaccurate algorithm, or the inaccurate humans. At the moment there seems to be no experimental data on what factors influence a human expert's coding behavior, but it is reasonable that not all coding instructions are equal. For example, in our case there could be a difference between a coder actively trying to figure out the underlying stimulus of the data, and not trying to do this. If sufficient evidence is accumulated for the data being from a static stimulus, then any implicit thresholds for detecting smooth pursuits are likely to be raised.

Similarly, there is a difference between coding for the presence of a single or several events. A sample classified as belonging to one event cannot, in the current coding scheme, be a part of another event. More event categories would mean increasing the coding difficulty, which also means that misclassified samples are less likely to be correct by chance, further increasing the difficulty for the algorithm. Additionally, the raw data was visualized in a particular way, in a particular GUI for this study. It is likely that the way the data are presented will alter the classification thresholds and biases of the humans.

A part of the underlying problem of the current classification approach is that the current oculomotor events are fuzzy, human-defined labels. If the definitions were algorithmic, there would by definition always be a winning standard algorithm, barring the potential influence of noise. So the solution for the problem should perhaps be sought in the intuitions of the researchers using these events. Or, the evaluation standard should be switched from accurately classifying artificial

labels, to a more grounded phenomenon, such as the level of visual intake during a certain state of the eye. Usually the events are detected for a particular reason and not for an interest in the labels *per se*, and these motivations could also be used to form the new measurement sticks for algorithm performance.

In summary, we have a classification problem without a solid gold standard against which we can verify oculomotor event classifications, of which the events lack clear definitions. Strictly, this would mean that this is apparently an insolvable problem. Human coders are not perfect and there are indeed difficult classification cases, but the general sentiment is that, in the simple case, what is a fixation and what is a saccade is something we can agree on. The point of agreement may be arbitrary, but humans are still in some agreement, much like agreeing on meaning in language. We also find that for detecting fixation durations from viewing static stimuli, the same algorithm is the best match for both of the two coders. The same is true for saccades and post-saccadic oscillations, across all stimuli types. Thus, it appears some intuitions of the coders are shared and not completely arbitrary.

Generalizability of the results

Having answered the questions we had at the start of this study, we wondered whether these results would generalize. First of all, the data are recorded using a particular system. In this case it is a video-oculographic system, the SMI HiSpeed 1250. Although producing good data compared to other VOG systems, it produces data with the typical characteristics of a VOG system, such as distinct overshoots during saccades and a higher noise-level, compared to e.g., scleral coil systems (see for example Hooge, Nyström, Cornelissen & Holmqvist, 2015). Although all systems have an overshoot component to some degree, the eye-tracker signal does not look the same across systems. This emphasizes that we should not expect event-detection to look the same either. The consequences of system type for the data signal, and consequently for the event detection has been, at least for microsaccades, discussed in Nyström, Hansen, Andersson and Hooge (2015b). There, the system type together with a common event-detection algorithm created an interaction that produced an artificially large microsaccade amplitude. Additionally, not all VOG systems are the same. An EyeLink 1000 system (SR Research, 2014) can track the pupil using an ellipse-fitting procedure, or a center-of-mass procedure, which determines the noise-level and tracking robustness. The tracking algorithm for the SMI HiSpeed, however, is not obvious from the manual (Sensomotoric Instruments, 2009).

Furthermore, VOG systems are affected by the natural changes of the pupil size, which does not change uniformly in all directions, and thus introduces a bias in the gaze estimation (see Drewes, Masson, & Montagnini, 2012). This may be one part of the explanation why one of our coders classified some segments of samples as belonging to smooth pursuits, rather than fixations, in data from static image viewing.

This hints at the larger question of how data quality affects event detection, which is discussed, e.g., by Holmqvist, Nyström and Mulvey (2012). It is expected that as the signal becomes noisier, it becomes increasingly harder to reliably detect the events, and especially so for the smaller and less distinct events. Likely, the PSO would become impossible to detect. Smaller saccades, as well as short and slow pursuit movements would become difficult to separate from a regular (noisy) fixation. With higher noise levels it is critical that the algorithms can adequately filter the signal, and perhaps adaptively so. This becomes especially important if the goals is to have an algorithm that can be used "out of the box" with none or few user-controlled parameters. Evaluations such as this, that use humans as some form of standard or reference, may also produce increasing higher human – algorithm deviations as the noise level increases. The expert knowledge and strategic flexibility in human coders suggests that the humans would not be disrupted to the same extent that algorithms would be. In this light, this would indicate that this study, using data with low noise, actually over-estimates the ability of the algorithms, compared to data from noisier systems.

Going beyond the question of system type and data quality, this evaluation was conducted by two humans at the same lab. Working in the same lab means attending the same seminars and the same discussions, often aligning the views of the people. Therefore, even when explicitly trying not to discuss the coding process too much beforehand, this may likely have led to similar decisions in cases of uncertainty. A future study could address this by having members from different labs, yet somehow coming together for this coding task despite minimal previous exchanges. To compensate for biases towards algorithms that were developed at our lab, for our eye-trackers, we did explore the possibility to tune some of the other algorithms parameters. The results are presented in Appendix B, and although we can see improvements in several cases, it does not change the main findings in this evaluation.

It should also be noted that perhaps this evaluation has actually overestimated the classification quality of the algorithms. In order to get around the problem of what default values to use for the algorithm parameters, we reverse-engineered the human coders. This also means that the

algorithms were somewhat tuned to the coders. Although this was done to ensure a fair competition between algorithms, this solution may have caused a slight overestimation of the performance of the algorithms.

We should not forget that there are also researchers that use whatever algorithm is provided by their (commercial) system. Should these results matter to them? Although we did not evaluate commercial algorithms, the variation in results between algorithm in this study should be a sign of warning. The commercial algorithms are at their core developed around known algorithms, such as a velocity-threshold algorithm (Tobii Technology, 2012; Sensomotoric Instruments, 2010) or a velocity-and-acceleration-threshold algorithm (SR Research, 2014), and their closed-source nature makes it more difficult to evaluate them. There is no reason to believe that these algorithms are immune to the challenges raised in this article. Their primary advantage, however, is their proliferation, which means there are plenty of other publications with the same algorithm which the researcher can compare the results against. This provides an indication of the reliability of the algorithms, but not necessarily the validity of these algorithms.

To summarize, there are a number of factors that were not systematically explored in this evaluation, and thus we cannot confidently generalize across factors such as eye-tracker model, noise levels, or coders from different labs. However, this study could indicate what expectations to hold for event-detection performance, given the limited number of evaluation studies at the moment.

Future directions of algorithm design

During this work, and having manually reviewed much raw eye-tracker data, we have noted some ideas of event detection algorithm design that seems to have been left unexplored.

The first idea is that events are assumed to be mutually exclusive. Whereas this makes sense for fixations and saccades, it does not make sense for, e.g., post-saccadic oscillations. We have seen data where saccades to a moving target results in an eye-movement that is tracking the moving target (smooth pursuit), but also oscillating from the end of the saccade (post-saccadic oscillation). What is the correct classification for such an event? Using current algorithms, no classification will be fully correct. It seems a new framework is needed, where different properties of the eye (movement) can overlap.

The second idea is whether thresholds separating two or more events should be placed in an unbiased manner, e.g., balancing the false-positive and false-negative rates between these events. From what we can see, e.g., by studying Table 8, algorithms are not balanced, and this

issue is not discussed at all. However, it would make sense to explicitly address this for both the algorithm designers, and the researchers using them. In one context, an unbiased trade-off between events may be desired, but for another context, more biased thresholds are important. For example, if the researcher has a strong desire to extract segments of data that are near-guaranteed to contain no smooth pursuit movements, then smooth-pursuits must first be detected in order to be discarded. Then, perhaps an algorithm which detects smooth pursuits, at the expense of a higher false-positive rate for smooth pursuits and higher false-negative rates for other events, is desired. Although this can be achieved by setting the thresholds for the different events accordingly, this could be made more explicit by allowing the use other types of thresholds. For example, an algorithm that also tries to set the certainty level of a particular classification, would in turn allow a researcher to select, e.g., only data from fixations with a certainty above 90 %. Or, if there is a probability for every event type for a given sample, then a desired certainty delta could be set, rejecting the sample if there is a too high chance of it belonging to a particular competing event. Related to this, it is evident from Fig. 3 that some of our algorithms refrain from classifying a sample if it does not meet the critera, whereas others operate with a more exhaustive strategy. Thus, modern algorithms are not consistent in whether they should only classify when they are "certain" or if they should be forced to make an informed guess about the sample. This is likely an option that the researcher would like to make for the particular study.

A third idea, which has previously been mentioned, is to actually make use of information in the stimuli. By knowing where animated objects are located in the visual fields, it should become easier for an algorithm to distinguish between a fixation and a smooth pursuit movement. To our knowledge, all current algorithm are stimuli blind.

A fourth idea is to make use of the pupil size signal, as well as the common x and y signals. As pupil size changes cause shifts in the position of the pupil center (Drewes, Masson, & Montagnini, 2012), drift movements may occur that may be very difficult to separate from smooth pursuit movements, and is likely what caused our coders to identify smooth pursuit in data from static stimuli. As far as we can tell, this information is not used in any algorithm.

A final idea is perhaps to forgo the process that revolves around (ill-) defined labels of oculomotor events, and develop a new ground truth against which the algorithm can be compared more straight-forwardly and in line with the aims of the researchers using the algorithms. One such approach, which has already been mentioned, would be to optimize algorithms against actual visual intake, which

may be easier to empirically ground compared to the label intuitions of researchers.

To conclude this section, there is much underused information in the eye and the output from most eye-tracking hardware, which can inform and improve, algorithms for eye movement event detection in the future.

Appendix A

Table 11 Participant and stimuli information

Participant no.	Stimulus name	Stimulus type	Offset (deg)	RMS (deg)
20	trial1	Moving dot	0.59	0.03
20	konijntjes	Image	0.59	0.05
21	Rome	Image	0.48	0.02
21	trial17	Moving dot	0.48	0.02
21	trial1	Moving dot	0.48	0.01
21	BergoDalbana	Video	0.48	0.02
22	trial17	Moving dot	0.32	0.03
23	Europe	Image	0.42	0.03
23	triple_jump	Video	0.42	0.03
24	trial17	Moving dot	0.27	N/A
25	trial1	Moving dot	0.44	0.02
27	vy	Image	0.29	0.03
27	trial17	Moving dot	0.29	0.03
27	triple_jump	Video	0.29	0.03
28	konijntjes	Image	0.35	0.07
29	Europe	Image	0.46	0.03
29	dolphin	Video	0.46	0.03
30	triple_jump	Video	0.23	0.06
31	konijntjes	Image	0.94	0.03
31	trial1	Moving dot	0.94	N/A
31	triple_jump	Video	0.94	0.03
33	vy	Image	0.34	0.02
33	trial17	Moving dot	0.34	0.03
34	Europe	Image	0.28	0.03
34	vy	Image	0.28	0.03
34	BergoDalbana	Video	0.28	0.02
38	trial1	Moving dot	0.15	N/A
38	dolphin	Video	0.15	0.03
39	konijntjes	Image	0.45	0.08
39	trial1	Moving dot	0.45	0.02
43	Rome	Image	0.48	0.04
47	Europe	Image	0.28	0.03
47	BergoDalbana	Video	0.28	0.03
47	konijntjes	Image	0.28	0.03

Appendix B

Parameter tuning

To explore whether the algorithms for which we had used the default parameters could be improved, we performed a simple parameter walk to see which combination of parameters that yielded the best (lowest) RMSD. This was performed with one parameter combination across all stimulus types, but separately for fixations and saccades, respectively.

Fixations
NH: no change at 0.67 (best)
CDT: no improvement
IMST: improvement from 1.58 to 0.75 (SaccDetThres 0.7; window_size 175)
IHMM: no improvement
BIT: improvement from 1.13 to 1.08 (n_lost 2; perc_control 0.96)

Saccades
EK: improvement from 1.46 to 0.61 (vfac 7, mindur 12)
IKF: no improvement
IMST: improvement from 1.09 to 0.21 (shared best with LNS) (saccDetThres 0.8; winsize 300)
IHMM: improvement from 1.09 to 1.00 (Viterbi 100; Baum Welch 7)
NH: no improvement

Conclusion
Some algorithms did improve. The only finding that upsets our previous results is that IMST can be made as accurate as the LNS algorithm. However, do note that the IMST is tuned separately for fixations and saccades, and the optimal parameter combination is not the same for the two events. In other words, the saccade detection will improve at the cost of the fixation detection, and vice versa.

References

Ali-Hasan, N. F., Harrington, E.J., & Richman, J.B. (2008). Best practices for eye tracking of television and video user experiences, *Proceedings of the 1st international conference on Designing interactive user experiences for TV and video*.Silicon Valley, California, USA. doi:10.1145/1453805.1453808

Bahill, A. T., Clark, M. R., & Stark, L. (1975). The main sequence, a tool for studying human eye movements. *Mathematical Biosciences, 24*(3), 191–204.

Blignaut, P. (2009). Fixation identification: The optimum threshold for a dispersion algorithm. *Attention, Perception, & Psychophysics, 71*(4), 881–895.

Cohen, J. (1960). A coecient of agreement for nominal scales. *Educational and Psychological Measurement, 20*(1), 37–46.

Deubel, H., & Bridgeman, B. (1995). Perceptual consequences of ocular lens overshoot during saccadic eye movements. *Vision Research, 35*(20), 2897–2902.

Drewes, J., Masson, G. S., & Montagnini, A. (2012). Shifts in reported gaze position due to changes in pupil size: Ground truth and compensation. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (pp. 209–212). ACM.

Engbert, R., & Kliegl, R. (2003). Microsaccades uncover the orientation of covert attention. *Vision Research, 43*(9), 1035–1045.

Engbert, R., & Mergenthaler, K. (2006). Microsaccades are triggered by low retinal image slip. *Proceedings of the National Academy of Sciences, 103*(18), 7192–7197.

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

Holmqvist, K., Nyström, M., & Mulvey, F. (2012). Eye tracker data quality: What it is and how to measure it. In *Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA '12* (pp. 45–52). New York, NY, USA: ACM.

Hooge, I. T. H., Nyström, M., Cornelissen, T., & Holmqvist, K. (2015). The art of braking: Post saccadic oscillations in the eye tracker signal decrease with increasing saccade size. *Vision Research, 112,* 55–67.

Inhoff, A. W., & Radach, R. (1998). Definition and computation of oculomotor measures in the study of cognitive processes. In G. Underwood (Ed.), *Eye guidance in reading and scene perception* (pp. 29–53). Oxford, England: Elsevier Science Ltd.

Javal, L. (1879). Essai sur la physiologie de la lecture. *Annales d'Oculistique, 82,* 242–253.

Kapoula, Z., & Robinson, D. (1986). Saccadic undershoot is not inevitable: Saccades can be accurate. *Vision Research, 26*(5), 735–743.

Kliegl, R., & Olson, R. K. (1981). Reduction and calibration of eye monitor data. *Behavior Research Methods & Instrumentation, 13*(2), 107–111.

Komogortsev, O. V., & Karpov, A. (2012). Automated classification and scoring of smooth pursuit eye movements in presence of fixations and saccades. *Behavioral Research Methods, 45*(1), 203–215.

Komogortsev, O. V., & Khan, J. I. (2009). Eye movement prediction by oculomotor plant kalman filter with brainstem control. *Journal of Control Theory and Applications, 7*(1), 14–22.

Komogortsev, O. V., Gobert, D. V., Jayarathna, S., Koh, D. H., & Gowda, S. M. (2010). Standardization of automated analyses of oculomotor fixation and saccadic behaviors. *IEEE Transactions on Biomedical Engineering, 57*(11), 2635–2645.

Larsson, L., Nystrom, M., & Stridh, M. (2013). Detection of saccades and post-saccadic oscillations in the presence of smooth pursuit. *IEEE Transaction on Biomedical Engineering, 60*(9), 2484–2493.

Leigh, R. J., & Zee, D. S. (2006). *The neurology of eye movements*. New York: Oxford University Press.

Manor, B. R., & Gordon, E. (2003). Defining the temporal threshold for ocular fixation in free-viewing visuocognitive tasks. *Journal of Neuroscience Methods, 128*(1), 85–93.

Matin, E. (1974). Saccadic suppression: A review and an analysis. *Psychological Bulletin, 81*(12), 899–917.

Mould, M. S., Foster, D. H., Amano, K., & Oakley, J. P. (2012). A simple non-parametric method for classifying eye fixations. *Vision Research, 57,* 18–25.

Munn, S. M., Stefano, L., & Pelz, J. B. (2008). Fixation-identification in dynamic scenes: Comparing an automated algorithm to manual coding. In *Proceedings of the 5th symposium on Applied perception in graphics and visualization* (pp. 33–42). ACM.

Nyström, M., & Holmqvist, K. (2010). An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods, 42*(1), 188–204.

Nyström, M., Andersson, R., Magnusson, M., Pansell, T. & Hooge, I. (2015a). The influence of crystalline lens accommodation on post-saccadic oscillations in pupil-based eye trackers. *Vision Research, 107,* 1–14. Elsevier. http://dx.doi.org/10.1016/j.visres.2014.10.037.

Nyström, M., Hansen, D. W., Andersson, R., & Hooge, I. (2015b). *Why have microsaccades become larger? investigating eye deformations and detection algorithms*. Vision research. In press.

Rashbass, C. (1961). The relationship between saccadic and smooth tracking eye movements. *The Journal of Physiology, 159*(2), 326.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin, 124*(3), 372–422.

SR Research (2014). *EyeLink 1000 Plus User Manual 1.0.5*. 2014.

Salvucci, D., & Goldberg, J. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications* (pp. 71–78). New York: ACM.

Sen, T., & Megaw, T. (1984). The eects of task variables and prolonged performance on saccadic eye movement parameters. In A. Gale, & F. Johnson (Ed.), *Theoretical and Applied Aspects of Eye Movement Research*. Elsevier Science Publishers.

Sensomotoric Instruments (2009). *iView X Manual*, ivx-2.4-0912.

Sensomotoric Instruments (2010). *BeGaze 2.4 Manual*.

Shic, F., Chawarska, K., & Scassellati, B. (2008). The amorphous fixation measure revisited: With applications to autism. In *30th Annual Meeting of the Cognitive Science Society*. Washington, D.C.

Tabernero, J., & Artal, P. (2014). Lens oscillations in the human eye. Implications for post-saccadic suppression of vision. *PloS one, 9*(4).

Tobii Technology (2012). *Tobii I-VT Fixation Filter – Algorithm Description*.

Van der Lans, R., Wedel, M., & Pieters, R. (2011). Defining eye-fixation sequences across individuals and tasks: The binocular-individual threshold (bit) algorithm. *Behavior Research Methods, 43*(1), 239–257.

Veneri, G., Piu, P., Federighi, P., Rosini, F., Federico, A., & Rufa, A. (2010). Eye fixations identification based on statistical analysis – case study. In *Proceedings of 2nd International Workshop on Cognitive Information Processing* (pp. 446–451). IEEE.

Veneri, G., Piu, P., Rosini, F., Federighi, P., Federico, A., & Rufa, A. (2011). Automatic eye fixations identification based on analysis of variance and covariance. *Pattern Recognition Letters, 32*(13), 1588–1593.

Vig, E., Dorr, M., & Barth, E. (2009). Ecient visual coding and the predictability of eye movements on natural movies. *Spatial Vision, 22*(5), 397–408.

Volkmann, F. C., Schick, A., & Riggs, L. A. (1968). Time course of visual inhibition during voluntary saccades. *Journal of the Optical Society of America, 58*(4), 562–569.

Volkmann, F. C., Riggs, L. A., & Moore, R. K. (1980). Eyeblinks and visual suppression. *Science, 207*(4433), 900–902.

Widdel, H. (1984). Operational problems in analysing eye movements. In A. G. Gale & F. Johnson (Eds.), *Theoretical and applied aspects of eye movement research* (pp. 21–29). New York: Elsevier.