Dissertation

CHAIR OF HUMAN-MACHINE COMMUNICATION
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
TECHNICAL UNIVERSITY OF MUNICH

Computationally Modelling Human Visual Perception: Eye Movements and Saliency

Mikhail Starstev

Fakultät für Elektrotechnik und Informationstechnik

Lehrstuhl für Mensch-Maschine-Kommunikation

Computationally Modelling Human Visual Perception: Eye Movements and Saliency

Mikhail Startsev

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik
der Technischen Universität München zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs
genehmigten Dissertation.

Vorsitzende(r):	Prof. Dr. Sebastian Steinhorst
Prüfer der Dissertation:	
	1. TUM Junior Fellow Dr.-Ing. Michael Dorr
	2. Prof. Dr.-Ing. Klaus Diepold
	3. Prof. Dr.-Ing. Erhardt Barth

Die Dissertation wurde am 05.03.2020 bei der Technischen Universität München
eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik
am 21.09.2020 angenommen.

Abstract

With the goal of better analysing and understanding human visual perception, this work focused on two related areas – eye movement event classification and saliency prediction. For the former problem, we developed both supervised and unsupervised approaches, substantially improving upon the state of the art. We especially focused on an important but largely neglected eye movement type – smooth pursuit, improving its detection across the board. We also expanded the applicability of saliency modelling by (i) designing techniques to utilise 2D-image saliency models to produce high-quality predictions for 360° content; and (ii) proposing a novel type of video saliency analysis – separately modelling attention exhibited as different eye movements classes. We found that *e.g.* training to predict smooth pursuits yields more generalisable predictors of attention, thus improving on learning to predict fixations, which is the traditional saliency formulation.

Zusammenfassung

Mit dem Ziel eines besseren Verständnisses der menschlichen visuellen Wahrnehmung hat diese Dissertation sich zum einen auf die Klassifikation und zum anderen auf die Vorhersage von Augenbewegungen konzentriert. Für die erste Problemstellung haben wir mithilfe von sowohl überwachten als auch unüberwachten Methoden des maschinellen Lernens den Stand der Technik deutlich verbessert. Ein besonderer Schwerpunkt lag dabei auf den langsamen Augenfolgebewegungen, eine wichtige, aber bisher in der Literatur aufgrund von technischen Schwierigkeiten weitgehend vernachlässigte Art der Augenbewegungen. Für die zweite Problemstellung haben wir den Einsatzbereich und die Leistung der im Bereich des maschinellen Sehens sehr populären Saliency-Modelle erweitert, indem wir (i) neue Methoden entwickelten, um mit existierenden Algorithmen für 2D-Bilder hochwertige Prädiktionen auf 360-Grad-Material zu erzielen; und (ii) das Problem der Vorhersage informativer Bildbereiche für Videos durch Analyse der Blickrichtung neu formulierten: unterschiedliche Augenbewegungen repräsentieren unterschiedliche Komponenten der Aufmerksamkeit, die wir separat modellierten. Unter anderem gelang uns der Nachweis, dass unsere neuen, nur mit den Daten langsamer Augenfolgebewegungen trainierten Modelle bessere Generalisierungseigenschaften aufweisen als herkömmliche Modelle und diese dadurch sogar in der klassischen Problemstellung der Vorhersage von Blick-Fixationen übertreffen können.

Contents

List of Acronyms

AUC area under the curve

BLSTM bidirectional long short-term memory

CNN convolutional neural network

CTC connectionist temporal classification

IoU intersection-over-union ratio

LSTM long short-term memory

PSO post-saccadic oscillation

ROC receiver operating characteristic

SP smooth pursuit

VR virtual reality

1

Introduction

Human vision, as any part of a biological system, evolved under certain constraints – how much energy it can consume and how much bandwidth it can utilise to deliver the visual information to the brain. For example, [1] puts an estimate on the transmission capabilities of the human retina at ca. 8 megabit per second. Constantly perceiving our whole field of view at the highest level of detail available to the visual system would quickly exceed this limit. Our picture of the world is, therefore, much sparser at every single moment, and the task of inferring a consistent model of the whole scene around us is passed on to the brain. In reality, humans only see the finest details when those are projected onto a special area of the retina – the fovea, which covers only about one to two degrees of visual angle, a tiny part of the visual field [2, Section 2.5.1].

Due to this limitation of the visual system, the eyes have to be redirected from one area of the visual world to another in order to gather enough information for a detailed understanding of the scene. Since humans rely on visual information throughout their daily lives, the eye movements represent a very fundamental aspect of how we observe the world around us. The precise nature and properties of this observation process can reveal many seemingly unexpected details about the observer, from their level of interest in the scene [3, 4, 5] to the information about their neurological health [6, 7, 2*, 11†] or professional expertise [8, 9, 10].

The notion of sparsity, which is inherent to human vision, is also very important in developing computational methods for media analysis, where modelling human attention helps compress images or video streams while maintaining high perceived quality [11, 12, 13, 14], improve *e.g.* action recognition [15] or video summarisation [16] systems, reaching as well into human-robot interaction [17] and driver assistance [18, 19].

Deciphering the observer's intent from their eyes could be seen either as a glimpse of a dystopian future (hence the privacy considerations related to gaze data [20]), or as getting close to obtaining the holy grail of human-computer interaction – communicating one's intention to an automatic system merely through moving the eyes. This would not just immensely help those whose interaction capabilities are limited, by physical abilities or situation [21, 22], but also increase the productivity of the workflow for regular users [23, 24, 25, 26], potentially combined with other input modalities [27, 28, 29].

1.1 Eye Tracking

Eye tracking provides a vital tool for studying the attention and eye movement mechanisms, as it allows recording the patterns and directions of gaze. There are various possible hardware implementations of an eye tracking system (*cf*. [2, Section 2.1]), but they are essentially aiming to reveal where the observer (*i.e.* the person using the system) is looking at any given time, in some frame of reference (*e.g.* relative to a computer display, a virtual scene, the observer's head, or the footage of a head-mounted camera).

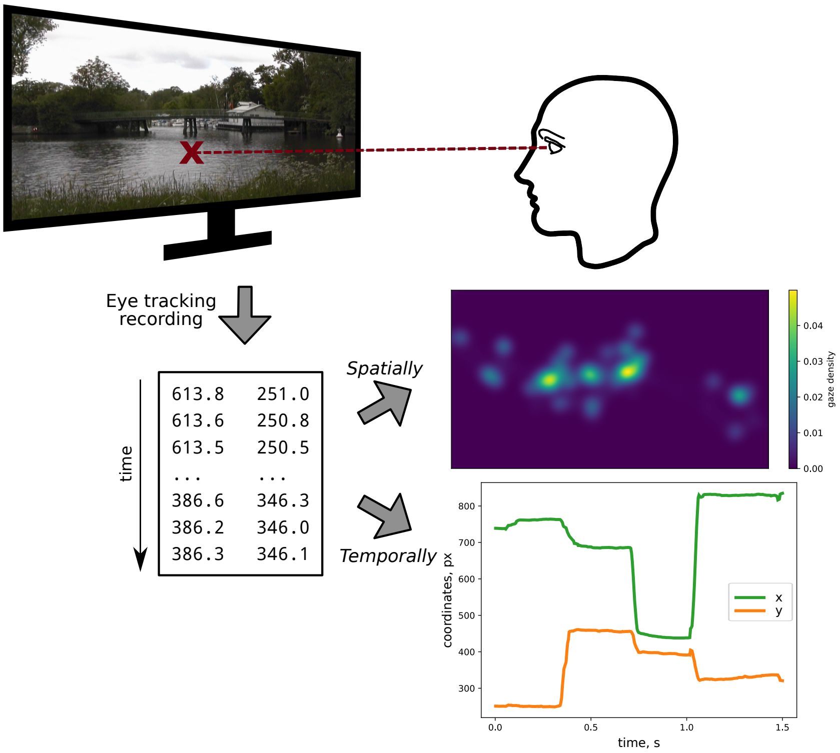
In the context of this work, we will refer to the outputs of an eye tracking system as *gaze locations* – a term we use interchangeably with "gaze points", "gaze coordinates", "gaze samples", or "points of regard". These will have different numerical representations depending on the coordinate system and use case. In the most common to date case of a computer monitor-based coordinate system, gaze locations could be represented by pairs of x and y coordinates. Depending on the frequency of a digital eye tracking system (which can range from 30 Hz in a wearable eye tracker to 2000 Hz in a state-of-the-art stationary system), these gaze locations are yielded once per a certain time interval (on average), resulting in gaze location sequences. The time stamp for each recorded gaze location can also be stored. The outputs of the gaze tracking system for an example of an observer viewing an image, as well as two of the alternative ways of their representation, can be seen in Figure 1.1. These representations highlight two aspects of the recorded gaze location sequences – spatial and temporal.

The spatial characterisation, achieved here by aggregating gaze locations across time (with added smoothing to account for the size of the fovea and the eye tracker measurement uncertainty, as well as for easier interpretation), results in a *saliency map* – an intensity matrix, with values corresponding to the density of gaze samples around each location of the stimulus (*e.g.* every pixel of an image). These saliency maps can be collected for either single individuals or populations, for both static images and videos. In the latter case, the aggregation across time happens within the time interval, during which each individual frame was displayed, resulting in a sequence of saliency maps, each corresponding to a video frame.

Plotting gaze coordinates over time reveals the temporal dynamics of gaze – the speed and size of the transitions between different locations on the scene, the amount of jitter in the signal, *etc*.

1.2 Eye Movements

On the visualisations of gaze coordinates over time (Figure 1.1 on the bottom), distinct patterns are visible that are following one another – periods of relatively static gaze and periods of rapid changes in gaze locations. This represents the basic intuition for eye movement analysis: The gaze signal contains "events" of different types, which can be differentiated by some criteria: *E.g.* the events in Figure 1.1 could be distinguished by the speed of gaze – *i.e.* how fast gaze moves from one location yielded by the eye tracking system to the next.



613.8	251.0
613.6	250.8
613.5	250.5
...	...
386.6	346.3
386.2	346.0
386.3	346.1

Figure 1.1: Eye tracking data representations: The *spatial* aspect of the data can be examined by analysing the distribution of the gaze samples on the stimulus (*e.g.* on an image or a video frame), corresponding to saliency map analysis. The *temporal* aspect of the data can be visualised by plotting the gaze point coordinates over time, providing the intuition for different events in the gaze data.

While there is only a limited number of event types defined in the literature, not all of them are used by all groups, and their definitions (*i.e.* the differentiation criteria from above) differ even from field expert to field expert, and from one eye tracking experiment set-up to another [30, 31]. In this work, we mainly talk about eye movements in the context of video viewing on a computer monitor, and we focus on eye movements that have major contributions to perception, according to the literature. *Fixations* – maintaining a relatively stationary gaze location, thus stabilising the image of the gaze target on the retina – are the largest contributors to information extraction from our surroundings (and traditionally seen even as the only contributors). *Saccades* – rapid shifts of gaze direction that bring new objects or parts of objects onto the fovea – are the means of quickly moving the gaze in the scene and are indispensable for exploration. These two eye movements are often studied exclusively [32, 33, 34, 35], likely following the tradition in the field, since dynamic stimuli have been gaining traction only in recent years (for static stimuli, fixations and saccades do indeed account for almost all of the gaze behaviour). There also seems to be a certain degree of interplay between fixations and saccades [36], as we do not perceive the world around us as completely changed after a large saccade, even though the image on the retina undergoes drastic changes.

When dynamic stimuli are displayed to the participants, they can also perform *smooth pursuit (SP)* – a tracking eye movement, when a moving target is maintained foveated (*i.e.* its projection falls onto the fovea). This way, fine-grained visual information can be extracted from the target even while it is moving. Highlighting the importance of this eye movement type, recent research also examined and described its influence on our perception [37, 38]. One of the contributions of this work is demonstrating that even from a purely quantitative standpoint, the amount of time observers spend performing SP rivals the time spent performing saccades in natural scene free-viewing [4*].

The gaze samples that are not as closely related to perception – blinks, noise of the oculomotor or the eye tracking system – we label together under the name of *noise*. The definitions of the eye movements that we used in our works are laid out more precisely in Section 2.1.1.

When the recording set-up is less restrictive than viewing content on a computer monitor – *e.g.* watching 360-degree video on a head-mounted display, – the definitions become more complex. Our work in [1*] addressed the challenge of both systematically denoting and manually annotating the eye movement classes in this context as well (though it is not a part of this dissertation).

It is important to understand that the eye movement definitions are not universal and can be introduced even on a case-by-case basis, they just have to (i) be well and reproducibly described, and (ii) represent useful concepts in the scenario in which they are used. For human-computer interaction, for example, defining a "fixation" as maintaining gaze locations within an interactive element [39] is a useful definition, even though this can include both periods of stationary gaze and small-magnitude gaze shifts. The same definition would, of course, stop being useful when applied in the context of studying "miniature" eye movements [40].

1.3 Eye Movement Classification

Whichever definitions are applied, however, many uses of eye tracking require (or assume) a segmentation of the raw signal into discrete events as the first and essential step. Often, doing this algorithmically is critical – either due to the interactive nature of the application [41, 42], to make sure annotation is performed consistently in the same way for all recordings [30], or because of the large volumes of data that need to be analysed [6*], as experts can take almost 20 times longer than the duration of recordings to label these [30, 4*]. Manual labelling effort only increases when the eye tracking data originate from a more unconstrained scenario instead of standard monitor-based experiments, *e.g.* for omnidirectional video viewing data with free head rotation, annotating just one second of raw data could require about a minute of the expert's time [1*].

Whenever eye movement events in the gaze recording signal are labelled by an algorithm, we talk about eye movement classification (the same achieved via expert intervention would be *manual* eye movement classification). There is some debate in the field about the correct naming of this process – "eye movement detection" is often used instead. One argument for *classification* over *detection* is that "detection" somewhat implies that what is being detected is well-defined and only needs to be localised, whereas "classification" does not seem to make this assumption. As we have discussed above, the definitions of eye movement events should suit their intended usage, and are not set in stone. We will, therefore, prefer the term "classification", but "detection" and "segmentation" essentially refer to the same concept in this context.

It is important to note that not *all* gaze samples have to be assigned an eye movement label – for example, an algorithm may exclusively detect one type of eye movements, in which case we would be talking about *e.g.* fixation [43, 44, 45] or SP [7, 3†] classification. Additionally, eye movement labels do not have to be constrained to just one label per gaze sample – in [1*], for instance, we utilised a two-label system to better describe not just the movements of the eye, but some aspects of eye-head coordination as well.

1.3.1 Brief Taxonomy

To better understand this work, two ways of categorising eye movement classifiers could be useful: which eye movements are considered by the algorithm, and which methods are applied to achieve the desired classification. We will examine these in turn in the two following sections.

1.3.1.1 Detected Classes

When considering the eye movements that are classified by the algorithms, we note that comparatively early eye movement classifiers focused heavily on distinguishing between fixations and saccades [32, 46, 47], though very specialised algorithms for particular (rarer) eye movement classes existed in their respective fields for some time as well [48, 49, 50].

Recent years have seen a push for more granularity when distinguishing between the different states of the oculomotor system, with post-saccadic oscillation (PSO) – the physical oscillations in the eye after a quick deceleration when a saccade is finished – being treated separately from both the saccade preceding it and the ensuing eye movement [51, 52], or SP being more often included in the analysis [42, 53, 4*]. More general-purpose, "universal" eye movement classifiers, which would detect all the eye movements of interest, and not just one or two types, have also become more frequent in recent years [52, 54, 55, 3*]. Part of the motivation for this is the fact that models can struggle when an eye movement type unknown to them is present in the data, and the detection of the main target of their analysis can suffer in quality [47].

The methods presented in this work belong to the very few approaches in the literature that tackle the detection of SP – an eye movement type that is particular to the gaze behaviour in dynamic scenarios (which are only too common in the real world, but have been largely neglected in previous research). One of the methods [3*] is general-purpose by nature, and aims to classify every sample of each recording. Our main algorithmic contribution in developing another approach [4*] was targeting SP, but we built a framework around it that detects all eye movement types we were working with.

1.3.1.2 Computational Methods

Early classification models often relied on applying simple thresholds to certain statistics of the gaze movement – speed and dispersion [32, 46], values derived from principle component analysis [56], or statistics of gaze direction [57]. Both the features and the rules for classification in these algorithms were hand-crafted. Some very recent methods fall into this category as well [58, 59], with more sophisticated features and decision rules.

As the field progressed, the researchers started applying machine learning models to the hand-crafted features that were often already in use – for instance, k-nearest neighbours classification was used in [60], Bayesian classification – in [61], random forests – in [62], and support vector machines – in [42].

The latest works added deep learning to the list of models used in the eye tracking field [52, 63, 3*]. These essentially rely on a deep network in jointly extracting the relevant features and deriving a suitably optimised classification rule. The models in [52] operate with minimal pre-processing of the eye tracking data (shifts of gaze positions instead of their absolute values), while [63] used speed and acceleration features.

Our work in [3*] tested both the unprocessed gaze signal and simple features extracted from it used as input to the network. We found that hand-crafted feature extraction substantially aided our model, perhaps because of its relatively small size. This will undoubtedly be changing with more and larger eye tracking data sets becoming (publicly) available [9†], where meaningfully training larger models will become feasible. Some works aim at creating large synthetic data sets [52, 64], but the diversity of the "real" data still plays a role even in this context.

1.3.2 Performance Measures

Given the multitude of algorithms and approaches for eye movement classification, navigating this field requires objective metrics of their prediction quality. The detection of some eye movement types can be in principle somewhat assessed without expert annotations (*e.g.* saccades – via analysing the relationship between their magnitudes and corresponding gaze speeds [65], or SP – by validating that detected pursuits coincide with motion in the stimulus [58]). However, for a more robust and universal analysis, expert labels are required. A separate discussion can be had on the subject of whether these represent the "ground truth" [30, 31], but this is not essential for what is considered in this section. We will, therefore, use "ground truth" as a stand-in for "expert labels", even though experts themselves may disagree on the labelling.

To evaluate the algorithmic labels against the ground truth, two levels of analysis are widespread in the literature: comparing these on the level of individual gaze samples and on the level of whole events of each considered eye movement class. Sample-level analysis presents us with a classical problem setting, where two sets of labels for exactly the same sets of entities need to be compared, *i.e.* every gaze sample has a true and an assigned label (missing labels, if present, can be interpreted as a separate class). In this case, all evaluation measures used in other fields can be – and have been – applied directly: accuracy [47], Cohen's kappa [52, 58], precision, recall, F1 scores [66, 67, 3*, 4*], *etc.*

When it comes to evaluating whole detected events against those in the ground truth, the comparison is compounded by the absence (or even the impossibility) of a definitive one-to-one matching of the events in the ground truth and in the algorithmic labels. Multiple intuitively understandable types of labelling errors can be named: For instance, an algorithmically detected event might be shifted in time compared to the "true" event, its duration can be under- or overestimated; an event could be missed, falsely inserted, fragmented, or merged with a number of adjacent events, or simply the class label could be wrongly assigned.

Especially if we also consider the possible combination of the error types above, it becomes abundantly clear that no one metric could meaningfully quantify all of these possibilities directly. The multitude of existing event-level evaluation strategies [14†] broadly fall into two categories: (i) The ones used most frequently can be essentially summed up as imposing a set of criteria to declare each considered pair of eye movement events – one in the ground truth, one as detected by a model – a successful match. The other matches are considered unsuccessful, and these decisions help populate the confusion matrix (either in a binary sense – *i.e.* fixation *vs.* not a fixation – or for several classes at once). Any evaluation metric that can be computed based on a confusion matrix can, in this way, be applied to eye movement events. Examples in the literature include F1 scores [30, 3*] and Cohen's kappa [52, 7*] with various matching criteria. Levenshtein distance (also known as "edit distance") between the sequences of true and detected events [52] could also be assigned to this type of evaluations, as computing this measure effectively means finding the set of edit operations to transform one sequence into the other, which implicitly establishes the correspondence between the events in the two sequences (this matching is encoded in the algorithms calculating the distance).

The type (i) measures cannot directly quantify the quality (or other properties) of registered matches between the two sets of events, however. That is precisely the purpose of the evaluation strategies of category (ii): These focus on specific aspects of the correspondence between the detected and the true events. For instance, relative timing offset and deviation in [30] quantify the temporal shifts between the two event sets; intersection-over-union ratio in [68, 3*] quantifies the overlap between the true events and their algorithmically detected counterparts. Reporting average properties of the detected events of a certain class (*e.g.* duration, amplitude, *etc.*) is also popular [47, 59, 3*].

While type (i) evaluation strategies are targeting quantifying the *number* of matched events and are, by their nature, not tied to match *quality*, some hybrid strategies can exist. For example, in [3*] we proposed involving event match quality in the definition of valid matches between events, thus also indirectly quantifying the quality of the detected events (*i.e.* though the metric we used is F1 score, it would vary depending on the quality of matched events and the pre-defined cut-off threshold for said quality).

1.3.2.1 Choosing a Suitable Performance Measure

Navigating the field (and, to some extent, a minefield) of possible evaluation approaches for a single problem can be difficult, as not all of the proposed strategies are necessarily viable. In [7*], we implemented a handful of metrics used in the literature to date, and tested whether these could separate purpose-built eye movement classifiers from a set of *baselines* we proposed, which were not assigning labels based on the gaze movements in the considered recording. Before this work, no systematic way of examining a metric for eye movement classification existed, partly since the first metrics aiming at event-level evaluation beyond average event statistics started appearing in 2016 [66].

Our study [7*] revealed that not all evaluation metrics are made equal, and that some could possibly be wholly unsuitable for the purpose (depending on the set-up). We have, therefore, proposed a novel metric (based on the hybrid approach to classifier testing mentioned above), which we also tested to demonstrate its ability to avoid assigning deceptively high performance scores to the baseline methods.

1.4 Saliency Modelling

We now transition to another use case of the eye tracking technology. Not only does it allow us to analyse *how* the observers are looking at a scene, but also *where* they are allocating their attention over time. We note that this work deals with human attention in somewhat "naturalistic" conditions, *i.e.* where the presented stimuli resemble scenes from real life. Modelling the distribution of attention is referred to as saliency prediction. Saliency modelling determines, based on the image or video data alone, what regions of these are informative to a human observer. The degree of informativeness is essentially being quantified by how often the gaze would be directed to such regions. Usually, the observers, whose gaze locations are recorded, are assumed to be free-viewing the stimuli, *i.e.* are not given an explicit task [69, 70, 71, 72].

1.4.1 Stimuli Domains

Traditionally, images were presented during the gaze-based studies of task-free attention [73, 74, 75]. The field has produced an ever-increasing number of predictive models [76], established benchmarks [77], and large data collections [72, 78].

In recent years, modelling *video* saliency received growing interest [79, 80]. This problem setting is, on the one hand, more challenging, as temporal dependencies have to be modelled and taken into account. On the other hand, there are indications that, as motion is a very important factor in video viewing, it could explain a large part of human attention [81]. Several diverse large-scale data sets have been introduced in recent years as well [70, 82], allowing for model development and testing across a wider range of video types with their corresponding typical viewing behaviours. In cinematic material, for example, a narrative and accompanying camera motion tend to induce the viewers to focus on a certain character or object, leading *e.g.* to higher centre bias effect in such content – the tendency of observers' gaze to gravitate towards the centre of the stimulus [83]. By comparison, the more real life-like scenes without intentional gaze directing show lower biasing effect and correspondingly more dispersed gaze patterns [69].

Very recently, attention allocation in visual exploration of omnidirectional (360°) content has become a prominent part of saliency research [84, 85, 86, 87], covering both static and dynamic scenarios. The immersive nature of these stimuli deviates from what was typically modelled by saliency prediction approaches before: There is no "discontinuity" in the scene, compared to *e.g.* borders of the regular video or image stimuli. Also, the full scene cannot be perceived at once as it surrounds the observer, adding extra degrees of freedom (*i.e.* head rotation) to the observer's visual behaviour, which have to be accounted for in order to achieve high-quality realistic attention modelling [86]. Additionally, the saliency maps are much sparser compared to the 2D stimuli case, as they represent a much larger field of view (compare to *e.g.* 48×27° for video viewing in [69]). This leads to a much larger number of observers being required to construct reliable ground truth maps [88].

Part of this work was to enable a more effortless transition from regular image saliency to that in 360° images, while accounting for the differences between the domains [5*]. In particular, we were mostly dealing with image distortions in the omnidirectional content represented in a common projection, and the border effects that should not be present in continuous 360° stimuli. While other works in this domain deal with adapting the existing computational methods (*e.g.* convolutional operations) to the 360° content [89], the approach in [5*] operates on the level of adapting the input data themselves instead.

1.4.2 Influence of the Eye Movements

Traditionally, saliency prediction is associated with predicting the locations of human observers' fixations [77]. For static stimuli (*e.g.* images), this is mostly sufficient, as the eye movements there mostly consist of fixations and saccades. For animated or video stimuli, however, such coarse classification is not sufficient, but the vast majority of works on saliency prediction continue to utilise the fixation detectors that are built into the eye trackers [71, 82]. Those, however, do not have any standard algorithms that take smooth pursuit (SP) into account [2, Section 5.2], leading to the absence of a clear understanding of what label will be assigned to those samples. Depending on the algorithm and its parameters, SP samples can be attributed to both fixation and saccade classes, inconsistent with the implied eye movement definitions.

The situation gets even more compounded in the case of omnidirectional content, videos in particular – in this context, eye-head coordination comes into play, and the labelling system becomes significantly more complicated [1*]. While some data sets even in this context talk about fixation-based attention, even if fairly large gaze speed thresholds are utilised [87], a systematic way of formalising the concept of visual attention is clearly needed.

In [6*, 10†], we incorporated the information about eye movement types, classified by a dedicated algorithm [4*], directly into the saliency modelling pipeline. In these works, we trained same-architecture models for the tasks of predicting either fixations or SP in videos. Analysing the performance of these models revealed that training for the prediction of the typically ignored eye movement type – SP – helped the models' ability to generalise to other data sets. These results demonstrated the potential of incorporating the knowledge about the eye movement types into computer vision problems dealing with human visual behaviour.

1.4.3 Evaluation

To understand whether a saliency model is doing a good job at mimicking human attention, its output needs to be compared to the ground truth – the distribution of recorded fixations (or *e.g.* SPs) of a number of human observers (could be just one in case of the egocentric video saliency [90, 91] or up to a hundred in 360° video saliency data sets [88]). In order to numerically express the similarities or differences between the model output and the target of its prediction, a number of metrics have been employed in the literature [92]. We do not focus on the specific metrics for saliency prediction here, though there are some recent developments in that area as well (*e.g.* reconciling saliency model rankings produced according to different metrics [93]). Of primary interest to us is the pipeline for the evaluation, and how it should be adapted based on the stimulus domain and the considered eye movement types.

Specifically for videos, where SP constitutes a non-negligible part of the viewing behaviour, data imbalance may arise: Certain frames can have no gaze samples that are attributed to a certain eye movement class (usually SP due to its sparsity, but can also apply to fixations on parts of the videos that heavily induce target following by presenting one moving object of interest only). This means that frame-by-frame evaluation would not be appropriate for such type of data, leading to the necessity of full-video evaluation.

Even in this context, however, an issue of imbalance arises when several eye movement types are considered separately: Some videos could contain much less certain-type attention examples than others (for an illustration, see Figure 1.2; note the variation of SP percentage in the gaze behaviour during video viewing – from zero to almost 65%, depending on the clip). This makes the averaging of saliency evaluation metrics between different videos in a data set potentially unfair. We examined this issue in [6*], proposing to re-weight the contributions of the individual video clips to the overall score depending on the amount of each eye movement type in its corresponding recordings. This has proven to be a significantly better estimate of the overall score, compared to the traditional averaging employed in other works.

1.5 Contributions and Thesis Overview

Here we will summarise the main contributions of the thesis (the details can be found in Chapter 2 and respective appendices). The discussion of the results obtained in the context of this dissertation can be found in Chapter 3.

The contributions of this work can be broadly subdivided into two areas. For eye movement classification, we:

- systematically annotated a large-scale data set of gaze recordings during dynamic natural scene viewing [4*, 2†, 6†], at the same time for the first time characterising smooth pursuits occurring during free-viewing in this sizeable collection of eye tracking data [4*, 4†] (Sections 2.1.1 and 2.1.2, also Appendix B);
- developed a clustering-based method to detect smooth pursuit [3†] – the first approach that takes the gaze patterns of several different observers into account, – further

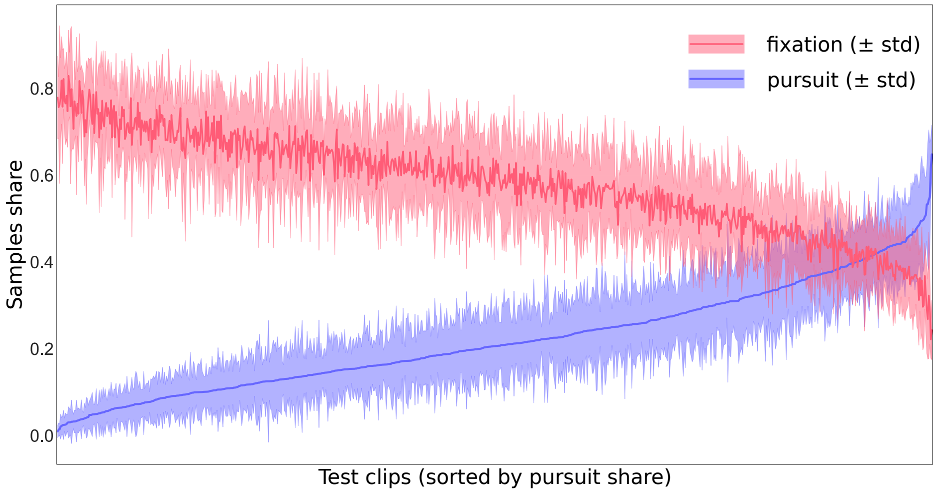


Figure 1.2: Fixation and pursuit shares (and their standard deviations) per video in the 844 test set clips of the Hollywood2 data set [94], as detected by our classifier [4*]. The remaining gaze samples are labelled as saccades or noise and are not represented here.

improving and analysing it in [4*, 12†, 13†] (Section 2.1.3, also Appendices A and B);

- introduced deep learning to the field of eye movement classification [3*, 7†, 8†], developing a state-of-the-art model for gaze event classification that leverages the temporal context by classifying the samples in a certain window of gaze samples at once (Section 2.1.4, also Appendix C);
- extended the evaluation procedures for the problem of eye movement classification, developing new event-level metrics that allow for more versatility [3*] or overcome the drawbacks of the previously used evaluation measures [7*] (Section 2.1.5, also Appendices C and D);
- introduced a general approach to testing the suitability of existing evaluation approaches – evaluating certain baseline methods that should by-design not be able to achieve good performance figures [7*] (Section 2.1.5.1, also Appendix D).

For saliency modelling, we:

- achieved state-of-the-art saliency prediction for omnidirectional images by developing a set of techniques to transfer saliency predictors from conventional two-dimensional images to the omnidirectional image domain [5*] (Section 2.2.1, also Appendix E); this approach won the corresponding 2017 IEEE ICME Grand Challenge [95];
- introduced the problem of smooth pursuit-based saliency prediction – *supersaliency* – and tested our proposed models and literature approaches on this problem and traditional saliency problem formulation [6*, 10†]; we showed that training for supersaliency leads to models that generalise better, even for the traditional saliency task (Section 2.2.2, also Appendix F);
- amended the saliency evaluation pipeline used in the literature by introducing video-wise re-weighting of the scores, and augmented it by a new metric quantifying how well a given saliency model differentiates between the salient regions corresponding to fixations and smooth pursuits [6*] (Section 2.2.2, also Appendix F).

2

Methods

This chapter will cover the methodology of the approaches developed in the context of this dissertation. Just as the contributions of this work, it is subdivided into two parts: eye movement classification (Section 2.1) and saliency modelling (Section 2.2).

2.1 Eye Movement Classification

2.1.1 Definitions and Manual Annotation

First of all, we address the definitions for the eye movements that we mostly used in this work, as these are not universal and should be stated for a given context [30, 31]. In these definitions, as for most of our experiments, we limited ourselves to monitor-based gaze tracking set-ups, with the participant's head being stationary (usually fixed by a chin bar). In this context, we defined the following eye movements:

- **Fixations** are periods of slow or absent gaze point movement (*i.e.* relatively stationary gaze), not following any moving object in the video. We note that absolute movement on the screen is meant here: an object that is stationary in the real world could be moving on the screen due to camera motion, and a moving object followed by the camera could be rendered stationary on the video surface.
- **Saccades** are rapid changes in the gaze point position, with the end of a saccade being defined as the time when gaze position becomes stable again (without setting a specific upper or lower limit on saccade amplitude).
- **Smooth pursuits (SPs)** are periods of time, when the gaze point is moving (slow motion is acceptable), as long as it is following a moving object in the video (*i.e.* gaze point moving at a similar speed and direction as the corresponding object).
- **Noise**, in our definition, refers to periods of lost or impaired tracking, which includes *e.g.* blinks, gaze outside the stimulus area, or physiologically impossible gaze movement signals.

These definitions were used for obtaining the "ground truth" for the eye movement classification problem by the means of manual annotation in [4*] (the annotators used the graphical interface developed in [2†]).

To differentiate between individual gaze samples with a certain label and uninterrupted sequences of same-class samples, we refer to the latter as *events* or *episodes*.

2.1.2 Annotated Data Sets and Analysis

To give context to the descriptions of the pipelines below, we here describe the properties of the data sets we used throughout this work. Example stimuli frames are provided in Figure 2.1 to illustrate the typical video content. Additionally, the basic statistics of these data sets in terms of the annotated eye movements are listed in Table 2.1.

All the data sets contain time series of on-screen gaze coordinates (relative to the top left corner of the displayed video), and our algorithms operate in the same coordinate system. The coordinates themselves are in pixel units, but these can be converted to degrees of visual angle using what we call meta-information of the recordings (stimulus resolution and physical size, plus the distance from the observer's eyes to the monitor) – the latter units are much more frequently used in the field of gaze event classification as they lend a meaning to the eye movement characteristics that is independent of the experiment set-up.

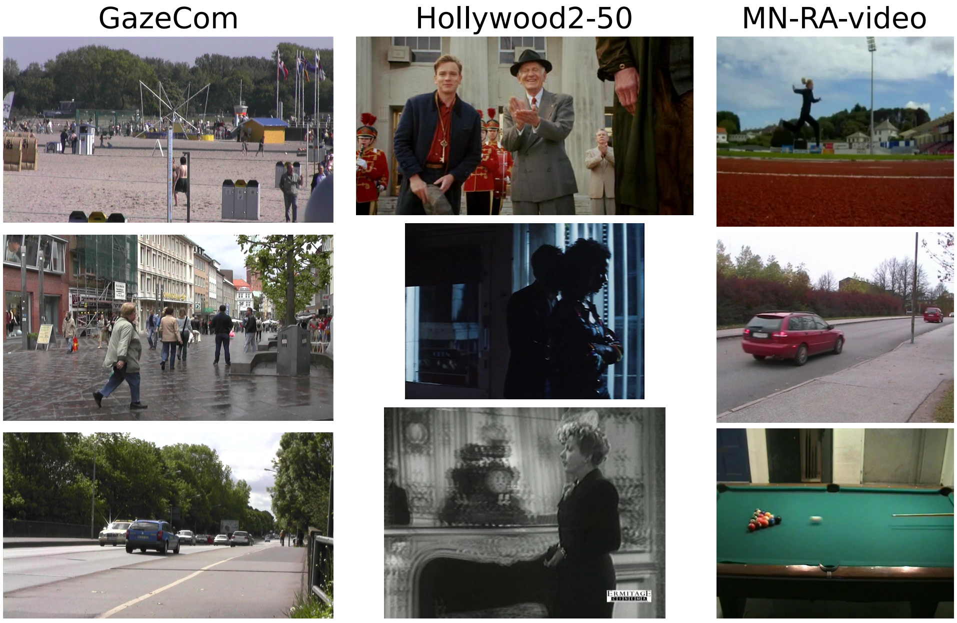


Figure 2.1: Representative examples of data set stimuli frames.

2.1.2.1 GazeCom

Most of our experiments were performed on this data set, as it is the largest available annotated set of eye movement recordings, especially when it comes to smooth pursuit. Overall, it contains over 4.5 h of gaze data, recorded for 18 short video clips (ca. 20 s each). The data set was introduced in [69], and we collected the manual expert annotations of eye movements for it in [4*]. This data set contains the eye tracking recordings of

Property	GazeCom	Hollywood2-50	MN-RA-video*
Total duration	4.8 h	2 h	0.03 h
Number of clips	18	50	3
Average observers per clip	46.9	12.8	6
Sampling frequency	250 Hz	500 Hz	500 Hz
Number of samples	4,318,056	3,669,529	58,058
Fixation share	72.5%	62.4%	37.7%
Fixation events	38,629	14,643	163
Fixation samples	3,132,536	2,295,233	21,888
Saccade share	10.5%	9.1%	5.3%
Saccade events	39,217	15,082	244
Saccade samples	454,787	335,190	3098
SP share	11%	24.2%	52.2%
SP events	4631	5649	121
SP samples	475,817	878,779	30,306

Table 2.1: Summary of the eye movement data sets used in this work. Marked with * – each recording in MN-RA-video was labelled by two equivalent raters and is contained in the data set twice, so the actual amount of unique data points and observers is overestimated by a factor of two in this table.

participants free-viewing a set of natural dynamic outdoor scenes (see example frames in Figure 2.1 on the left). These contain such moving targets as pedestrians, cars, cyclists, and animals, portrayed at a variety of distances to the camera.

While there are comparatively few video clips, each was viewed by 52 observers (though some recordings were discarded by the data set authors due to data loss [69]). This is particularly helpful for our clustering-driven method for SP identification (in the following Section 2.1.3), as it allowed us to thoroughly analyse the relationship between the algorithm's performance and the number of observers, whose data are analysed simultaneously.

2.1.2.2 Hollywood2-50

In this work, we also analysed an annotated subset [9†] of the data set Hollywood2, originally introduced in 2012 by Mathe *et al.* for the purpose of studying saliency [94]. The original data set (Hollywood2) contains over 22,000 gaze recordings for 1707 clips, totalling over 70 h. We annotated a random 50-clip (2 h) subset of its test set for the eye movement types we defined in Section 2.1.1. The full data set was used to train our saliency models in Section 2.2.2, however, in combination with algorithmically produced eye movement labels.

The video content of this data set is substantially different from GazeCom (see examples in Figure 2.1): Unlike in that data set, the videos contain professionally shot and edited material with camera movement and scene cuts. Additionally, only four out of the 16 observers were recorded in the free-viewing paradigm, while the other 12 were given the task of recognising the actions appearing in the scene (from a limited set of labels). Together, these factors influenced gaze behaviour, which is evident even from the rudimentary statistics in Table 2.1 – at least based on the amounts of different gaze events. Nonetheless, no specific gaze behaviour strategies were prescribed during the data collection.

2.1.2.3 MN-RA-video

In contrast to Hollywood2, the subjects, whose eye movements were recorded for this data set [51], were explicitly instructed to follow the moving objects in the videos. The data set itself was collected in [51], and annotated by two experts in [51] and [47]. We name it after the initials of the two annotators (MN-RA), and particularly focus on the video-viewing subset (the full set also contains image viewing and following a moving dot with the eyes). Examples of representative video frames can be seen in Figure 2.1 on the right. This data set is much smaller than the other two, and we mostly used it for validating our algorithm.

In addition to the three main classes we consider (fixations, saccades, SPs), the experts labelled post-saccadic oscillations (PSOs) and blinks, with some samples not attributed to any class as well. We only used the annotations of the three main classes in our pipeline (merging PSOs and saccades to match the definitions we used).

2.1.3 Clustering-based Smooth Pursuit Detection

It has been known in the literature on eye movements that motion in video attracts human attention [96] and, specifically, corresponds to the dense regions in the distribution of the gaze points [69, 81]. Additionally, our own work [4*] directly quantified the spatio-temporal consistency of attention in GazeCom specifically for the smooth pursuit eye movement, demonstrating that it was higher than for any other eye movement type.

These observations served as the chief motivation for our clustering-based approach to detecting smooth pursuit [4*, 3†] – the first work in the literature to fuse the gaze signals of several observers to perform eye movement classification. The pipeline can be briefly described as follows: (i) First, saccades and blinks are detected with an approach developed in conjunction with the GazeCom data set (see Section 2.1.2.1) [69]. (ii) Fixations are then detected with a gaze speed- and dispersion-based approach, together with some of the noise samples. (iii) All the remaining gaze samples are declared pursuit *candidates*. Those are aggregated for all observers that have viewed a given stimulus, and clustered in the three-dimensional space comprised of time and two spatial axes defining the video frame surface (x and y). The samples that form dense clusters are then classified as SP, with the rest labelled as noise. The paragraphs below detail each of these steps.

The parameters of our fixation and pursuit detection methods were jointly optimised via a random grid-search [4*, 13†], and these optimised values are presented below.

2.1.3.1 Saccade and Blink Detection

We detected saccades via a dual-threshold algorithm proposed in [69], where each saccade is required to contain at least one point with a sample-to-sample gaze speed no lower than the “fast” threshold (138°/s), and its on- and offset are then determined by the requirement that all samples in a saccade correspond to sample-to-sample speeds no lower than the “slow” threshold (17°/s). Additionally, thus detected saccades that are too short ($\leq$15 ms) or too long ($\geq$160 ms) are discarded as noise, as well as samples with unexpectedly high speeds ($\geq$ 1030°/s).

Blinks were identified as periods of missing eye tracking data, extended to potentially include the saccades surrounding them, provided that those are within 25 ms from the lost tracking samples (these “saccades” are the likely artefacts of the eyelid partially occluding the eye in the process of a blink, which are common to video-based eye tracking [2, Section 5.7]).

All thresholds listed above are the same as in the original paper [69]. We also note that while the sample-to-sample speeds were used for GazeCom, for the data sets with higher eye tracker sampling frequency (*e.g.* Hollywood2-50 or MS-RA-video, *cf*. Table 2.1), an equivalent speed integration window was used (4 ms, equivalent to two sampling periods of a 500 Hz system).

2.1.3.2 Fixation Detection

For detecting fixations, we designed a relatively simple thresholding-based algorithm that is similar in principle to other modern fixation detection methods [32, 46, 56]: Fixations are defined by the gaze being nearly stationary, which can be quantified in terms of gaze speed (*i.e.* how far are the consecutive gaze samples from another, maybe with some smoothing) or dispersion (*i.e.* how far are the samples in a certain time window from one another).

In our approach, we considered inter-saccadic intervals (*i.e.* periods of time between already detected saccades), and processed them independently. We immediately marked those intervals with dispersion below 1.4° as fixations. For the rest of the intervals, a sliding window of 100 ms was applied (with a step of one sample), in which the speed was computed (as displacement between the start and the end of the window, divided by time). When the speed fell below 2°/s, a fixation onset was marked. When it rose above the same threshold, a fixation offset was marked.

All intervals not labelled as either saccade, blink, fixation, or noise up to this point, and lasting $\geq$140 ms, were kept as “pursuit candidates”.

2.1.3.3 Smooth Pursuit Detection

Pursuit candidates were pooled from all the observers that have viewed a particular video, and henceforth processed together. For detecting dense clusters in this set of gaze coordinates, we adapted DBSCAN (the abbreviation stands for “density-based spatial clustering of applications with noise”) [97].

2.1.3.3.1 Base Algorithm Choice We decided to build our approach on this algorithm because of several of its properties. First, as the name suggests, it was designed for the data that contain noise. In our scenario, we expected pursuit candidates to be subdivided into either dense regions of gaze samples that correspond to the observers' eyes following the same objects, or sparsely distributes gaze points that are likely recording noise (since those would have been left unmarked by the saccade and fixation detectors in previous steps).

Second, DBSCAN does not assume that the clusters in the data can be described as centroids. This is very important for SP detection, as their trajectories can be very stretched and also depend on the motion in the stimulus, which does not have to be linear (see example clusters in Figure 2.2).

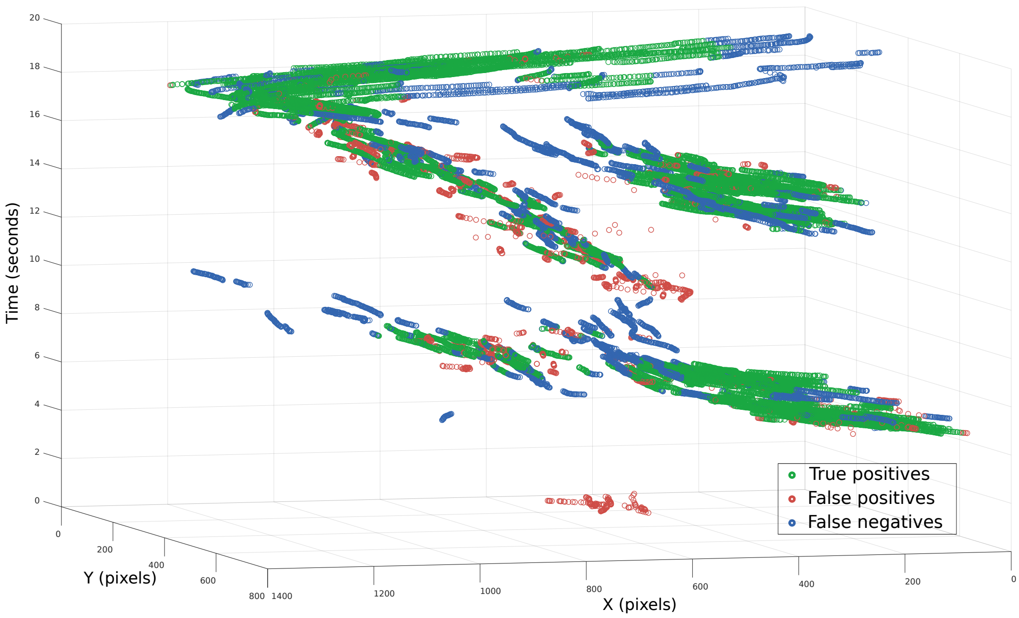


Figure 2.2: Clustering output example of our algorithm (for the "koenigstrasse" video of the GazeCom data set [69]), with true positives, false positives (*i.e.* false alarms), and false negatives (*i.e.* missed pursuit samples) marked in different colours (green, red, and blue, respectively).

Lastly, this algorithm does not require knowing the number of clusters in advance. As the number of SP targets in a video is unknown a priori, this a crucial point for processing an arbitrary collection of pursuit candidates.

These properties set it aside from traditionally used clustering approaches such as k-means [98] or mixture of Gaussians.

2.1.3.3.2 Original and Modified DBSCAN To separate densely clustered points from noise, DBSCAN relies on the concept of point neighbourhood, in which the density is estimated as the number of other data points. In classical DBSCAN, the neighbourhood is defined by Euclidean distance. For a given data point p in $\mathbb{R}^N$, any point $q \in \mathbb{R}^N$ is said to belong to the neighbourhood of p if the following inequality is satisfied:

$$\rho(p,q) = \sqrt{\sum_{n=1}^{N} (p_n - q_n)^2} \le \epsilon, \tag{2.1}$$

where ϵ is a parameter of the algorithm, defining the size of the neighbourhood.

Based on the point's neighbourhood, it is attributed to one of three categories: (i) p is declared *a core point* in a set of points P if its neighbourhood consists of at least $minPts$ points belonging to P ($minPts$ is another parameter of DBSCAN). (ii) p is *a border point* if it is not a core point itself, but its neighbourhood contains at least one core point. (iii) Otherwise, p is *an outlier point*. Core and border points can be considered as belonging to some cluster (with the neighbourhood relationship defining the cluster identities), and outliers represent noise.

In the context of SP detection, data points are represented by triplets $(time, x, y)$. The detected outliers were labelled as noise in our approach, while the gaze points belonging to clusters received an SP label.

As the feature space, in which we operate, consists of coordinates with different units of measurement (seconds and pixels or degrees of visual angle), utilising Euclidean distance would necessitate scaling the temporal and spatial coordinates to ensure their balanced contribution to the distance measure. As there is no universally meaningful constant to perform this scaling operation, we introduced two distance thresholds instead of one (ϵ_{time} and ϵ_{xy} instead of ϵ), effectively redefining the neighbourhood condition expressed by the inequality (2.1) in the following way:

$$\begin{cases} \rho_{time}(p,q) & = & \sqrt{(p_{time} - q_{time})^2} \le \epsilon_{time} \\ \rho_{xy}(p,q) & = & \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \le \epsilon_{xy}. \end{cases} \tag{2.2}$$

This can be seen as a case of a more general DBSCAN modification, where the set of coordinates $1 \ldots N$ is subdivided into K groups $G_k = \{g_i^k \mid 1 \le i \le L_k, g_i^k \in 1 \ldots N\}$, $k \in 1 \ldots K$ such that $\forall j \ne k : G_j \cap G_k = \emptyset$ and $\cup_{k=1}^{K} G_k = \{1 \ldots N\}$. In that case, the condition for point q belonging to the neighbourhood of p could be written out as

$$\forall k \in 1 \ldots K : \rho_{G_k}(p,q) = \sqrt{\sum_{i \in G_k} (p_i - q_i)^2} \le \epsilon_k, \tag{2.3}$$

where $\epsilon_k, k \in 1 \ldots K$ would be the parameters of the modified algorithm. In principle, other distance metrics can be used as well, instead of the Euclidean distance.

2.1.4 Deep Learning-based Classifier

While our clustering-based algorithm for SP detection (Section 2.1.3) focused on one eye movement type, using traditional algorithms to detect the rest, here we introduce a deep learning-based approach that simultaneously labels all eye movement types in the classified recording [3*]. This effectively incorporates the idea that designing classification algorithms without accounting for other possibly occurring eye movement types can harm their performance [47].

The input to our model is a fixed-length window of gaze samples' features (either simply their xy coordinates, or a combination of speed, direction, and acceleration of gaze movement at several temporal scales). The expected output is a corresponding window of labels (per-class probabilities, to be precise, with the most likely class label chosen for every sample in order to obtain the final classification result). Accordingly, we used categorical cross entropy as a loss function.

The architecture of our model consists of two parts: First, a stack of four convolutional layers is used for feature extraction. We employed one-dimensional convolutions over the time axis, as our data are sequential. Following these, we included a stack of two bidirectional long short-term memory (BLSTM) structures. Long short-term memory (LSTM) was introduced in [99], with the bidirectional aspects of temporal processing explored in *e.g.* [100, 101]. In our architecture, the BLSTM layers are followed by a temporally-distributed (*i.e.* the weights are shared across all time points) densely connected layer, yielding the output of the system.

We note that the slightly different network originally introduced in [3*] used an additional fully-connected layer (also time-distributed) between the convolutional layers and the BLSTM, but our subsequent experiments have shown its redundancy, and led us to increase the number of the convolutional and BLSTM layers [8†], leading to the network structure described here. In the same work, we also quantitatively validated the usefulness of combining the forward and backward LSTM states in a bidirectional manner for the eye movement classification.

2.1.4.1 Cross-Validation

An important part of the model testing pipeline is the way in which the available data are used for training, validation, and testing. We trained our model on GazeCom (see Section 2.1.2.1) – the largest available data set with smooth pursuit annotation. However, while we additionally tested [3*, 9†] our model on *e.g.* Hollywood2-50 and the much smaller MN-RA-video data set (Sections 2.1.2.2 and 2.1.2.3), the amount of recordings in GazeCom also makes it a preferred testing ground for any classification algorithm. To utilise it for both purposes, and yet not unfairly inflate our model's performance, we used cross-validation – a commonly employed model testing technique. An overview of typical validation schemes is given *e.g.* in [102].

In its basic form, all samples of the data set (in this case "samples" would constitute windows of gaze data) are randomly split into K folds (in which case we talk about K-fold cross-validation). This works, because data set samples are usually treated as independent (which is perfectly adequate for *e.g.* image classification with typically independent images, *etc.*). Some inter-dependency between the samples on a class level is considered in stratified cross-validation, where class balance is preserved when partitioning the data. The work in [102] also ensured that very similar same-class samples are placed in different folds, adding another aspect of sample inter-dependency. This measure was aimed at balancing the intraclass feature distributions between the validation folds, especially for the data sets with few samples.

In the context of eye movement classification, however, a different problem needs to be addressed: Typical data sets are recorded for a fixed set of observers and stimuli, meaning that even the individual recordings are related – different observers viewing the same video clip show similar eye movement patterns [69, 7*], and the recordings of the same observer can share certain properties across stimuli (*e.g.* eye movement characteristics can be used in biometrics [103, 104]).

In our work, we compared two ways of partitioning the available data into validation folds: based on the observer identity or on the stimulus content. The former strategy ensures that no observer's recordings are contained in more than one fold. The latter, on the other hand, partitions the data in such a way that all recordings for the same video are contained in the same validation fold.

We directly empirically compared the two techniques in [3*] under similar conditions (same number of folds and an identical training set-up) to determine whether they differ in terms of leading to the overestimation of the model's performance. We also argued for the theoretical advantages of the video-based validation split, specifically for the detection of stimulus-dependent SP.

2.1.5 Evaluation Metrics

Algorithmically producing eye movement labels could be regarded as a sample-wise classification problem, with traditional corresponding quality measures (*e.g.* precision, recall, F1 score, or Cohen's kappa). However, the sequential aspect of the data leads to another possibility: considering not individual gaze samples, but whole gaze events. The evaluation in this domain is more in line with the intuitive understanding of the experts who usually think in terms of episodes rather than individual samples. Despite the intuitiveness of the concept, its formalisation may be ambiguous, especially for determining which events in the ground truth and in the algorithmic labels should be matched for evaluation.

Several approaches for matching the automatically annotated and the ground truth event sequences have been proposed in the literature. *E.g.* in [66], the authors only consider the ground truth event boundaries, and limit themselves to using individual samples from the algorithm's output in order to "vote" for the event label within the boundaries. In [30], the earliest event in the algorithm's labels to overlap with the ground truth event is chosen, while in [52] the event with the largest overlap is chosen.

In our work, we proposed several improvements to the evaluation scheme for eye movement classifiers. First, since the events of different classes inherently have different durations (*e.g.* according to [2, Table 2.3] saccades typically last between 30 and 80 ms, while typical fixations are 200–300 ms), the same amount of overlap for two possible matches could reflect a different *degree* of overlap. For instance, if a ground truth saccade of 50 ms overlaps by 25 ms with both an algorithmically detected saccade of 30 ms and a fixation of 200 ms, the intuitive degree of overlap is higher for the pair of saccades. We formalise this by making sure the event pair with the highest intersection-over-union ratio (IoU) is chosen [3*], instead of the highest overlap (*i.e.* intersection) that was used in [52]. For two events A and B, IoU is defined simply as $\frac{A \cap B}{A \cup B}$, both intersection and union expressed in time units (IoU itself being, therefore, unitless).

Another technique we proposed was to only allow for the two events to constitute a match if their IoU is above a certain threshold (we used 0.5 in [3*], where we explained the theoretical and practical benefits of this value). This set a "match quality threshold", meaning that the evaluation would only register as successful detections those event pairs that have a higher IoU, thus avoiding weak matches.

We also extended this approach by testing models at a set of such IoU thresholds, allowing to compare models' performance over a range of match quality strictness levels. This is similar to receiver operating characteristic (ROC) analysis, where models can be compared *e.g.* in terms of their true positive rates at a variety of false positive rate values. Similarly, when the IoU thresholds are systematically varied, models can be compared at the respective detection quality cut-off levels.

Lastly, in [7*] we proposed a novel evaluation metric, which we based on event-level Cohen's kappa in [52], combined with our IoU-thresholded matching scheme in [3*]. We experimentally justified a higher quality threshold, and modified the existing procedure [52] for computing event-level Cohen's kappa in order for the metric to represent the same intuitive concept as on the level of samples – *i.e.* how good a model is at correctly placing the labels it assigns, compared to randomly shuffling those. We reformulated this intuition and proposed a way to estimate this value for eye movement episodes instead of individual samples.

2.1.5.1 Metric Testing via Classification Baselines

To navigate the field of eye movement metrics, we proposed a systematic approach to testing these, by evaluating a set of baselines for eye movement classification [7*], which we designed in such a way that their performance should not be comparable to designated algorithms for eye movement detection.

All of the baselines are based on the premise that they cannot take into account the eye tracking data, for which they are producing the labels. Within this paradigm, we considered the following approaches: (i) randomly assigning event sequences, according to average prior and transition probabilities between different classes, and drawing event durations from corresponding realistic distributions; (ii) assigning to each gaze sample the most frequent eye movement class label for the samples of all observers around the corresponding time point; (iii) based on video features, predicting the most frequent eye movement label of all observers for each stimulus video frame; and (iv) using the ground truth labels of *another* observer to produce the eye movement labels for every recording.

Intuitively, these baselines should be inferior to a dedicated algorithm that is basing its decisions on the gaze dynamics in the recording that is being processed (*i.e.* the data which are ignored by the baselines). Surprisingly, this was not a trend we observed for many metrics used in the literature, for event-level evaluation in particular. This way, the baseline approaches we introduced allowed us to test whether a certain metric is suitable to capture the differences between gaze-independent and gaze-dependent eye movement classifiers. In [7*], we analysed eight metrics from the literature and our proposed metric, discovering that only two of those separated the baselines from the dedicated models, and pointing out a metric that favoured our baselines over most of the algorithms.

2.2 Saliency Modelling

2.2.1 Omnidirectional Saliency

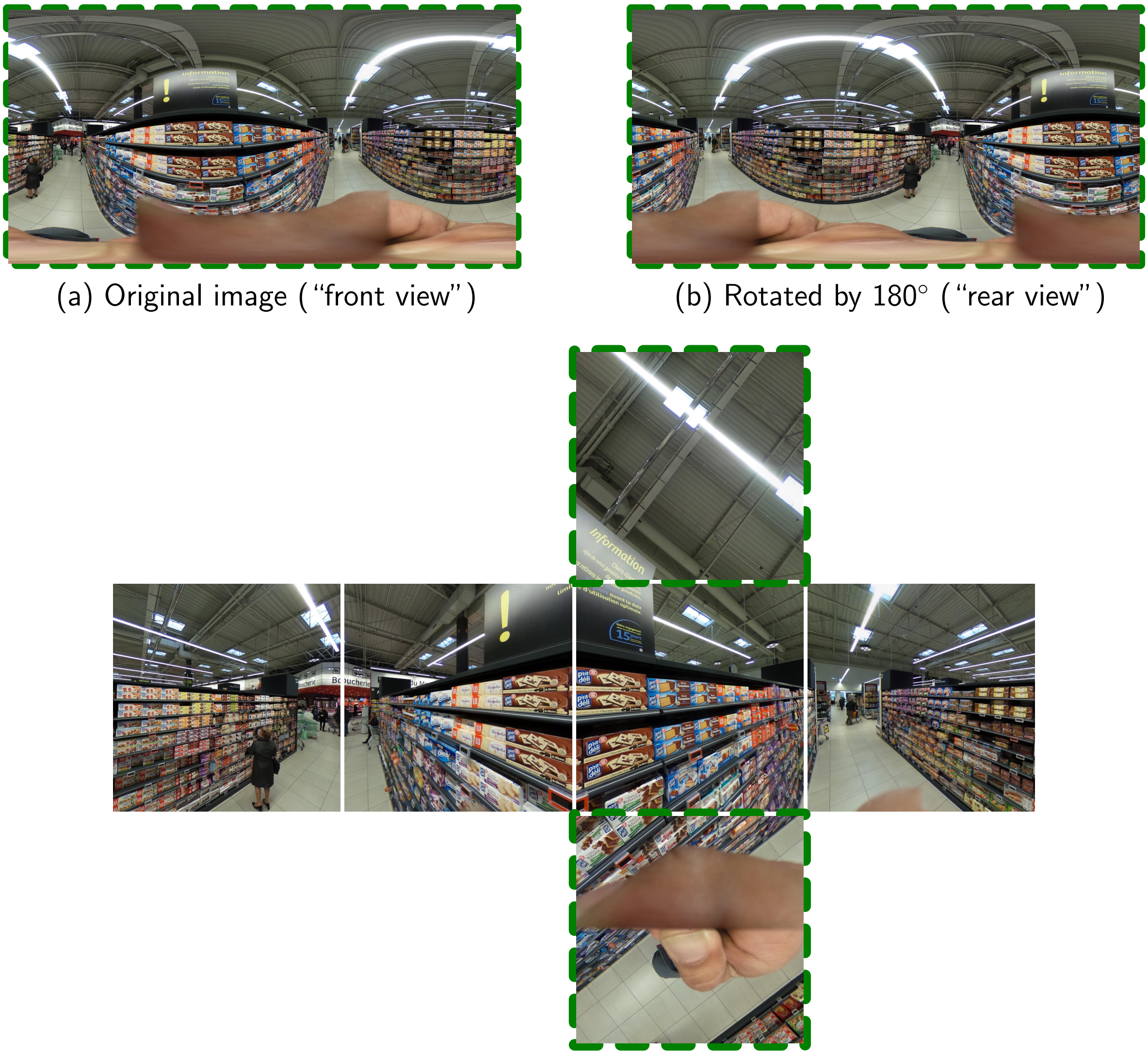
One important step toward ubiquitous human attention modelling could be formulated as transitioning from predicting the viewing patterns in monitor-based experiments to more immersive and less restrained set-ups. While modelling human behaviour entirely "in the wild" – *i.e.* moving around in the real world, including different modes of locomotion and interaction – would be the perfect solution, this problem setting is difficult both to accurately represent numerically and to model. Virtual reality (VR) can be seen as an intermediate step towards full real-life attention while maintaining tractability.

In this domain, 360-degree content is an important part of the experience. Such omnidirectional images and videos are gaining popularity both with the end users and the researchers. In order to facilitate the transition from the vast number of traditional two-dimensional saliency predictors to the 360° domain, which is often represented by scenes in equirectangular projection [86] (see example in Figure 2.3a; note the distortions towards top and bottom of the image), we proposed a set of techniques to allow the application of any traditional saliency model without changing the model in any way [5*].

To this end, we manipulated the input data instead. While a saliency model can be directly applied to an image in an equirectangular projection, this leads to a few artefacts in the resulting prediction. We aim to nivellate these with the following manipulations: (i) To avoid vertical border effects (at the borders of the equirectangular image; no such borders are observed in the ground truth maps as head rotation is allowed in all directions), we run the saliency model to predict two saliency maps – one for the original equirectangular representation, and one for the representation rotated by 180° around the vertical axis (see Figure 2.3b). (ii) We also represent the 360° scene as a set of cube map faces (Figure 2.3c) and compute the saliency maps for these separately, thus avoiding the distortions in the images. (iii) Finally, combining the two previous manipulations, we employed the cube map-based technique for the regions of the equirectangular image that are most distorted (the top and the bottom faces of the cube), also computing the two saliency maps as in (i). The respective input manipulations used for this combined approach are highlighted in Figure 2.3.

For all the techniques (i)–(iii) above, several saliency maps are produced. All of these were then re-projected onto the original coordinate space to match the input image. For the rotated saliency map produced by (i), this simply meant the inverse rotation. For the cube map faces, the inverse projection to spherical coordinates was performed.

After several saliency maps were produced in this way (by an unmodified saliency model applied to the modified inputs), we combined these via a pixel-wise maximum operation. Compared to averaging the values, this does not produce lowered saliency scores at the locations close to the image borders on at least one of the manipulated inputs.



(a) Original image (“front view”)

(b) Rotated by 180° (“rear view”)

(c) Cube faces

Figure 2.3: Example of an original omnidirectional image (2.3a) and its manipulated versions used in our work – rotated around the vertical axis (2.3b) and projected onto the faces of a cube (2.3c). Green frames indicate the images used in the final model, combining both approaches.

2.2.2 Smooth Pursuit-based Attention

In the domain of video stimuli, we experimented with the underlying problem setting of saliency prediction. While “fixation prediction” is traditionally used as an equivalent of “attention prediction”, most works consider saliency as a purely computational problem (producing an output that is similar to the ground truth based on the input), without going into detail as to what is represented by these “fixations” for dynamic video content.

Conversely, we specifically examined the different ways to formalise the concept of predicting human attention in video viewing, based on the eye movement category that expresses this attention. We separate the problems of predicting the distribution of fixations and of smooth pursuits (SPs) [10†], additionally considering posing the problem of predicting the combined distribution of the gaze samples for both classes in [6*].

To create the data set for our problem statement, we used our eye movement classification approach [4*] from Section 2.1.3. The fixation and SP gaze samples detected by this algorithm on the full training set of the Hollywood2 data set [94] (on which we based the Hollywood2-50 subset in Section 2.1.2.2) were then converted to respective attention density maps by a spatio-temporal Gaussian filter.

We trained a slicing convolutional neural network (CNN) model [105] on these data (a model with two-dimensional convolutional filters applied in different planes in order to process the three-dimensional data of a video sequence), and compared the models trained in the same way but on differing ground truth saliency data sources – fixation and SP attention. We additionally trained an end-to-end deep model (the architecture based on deep densely connected networks [106] and convolutional long short-term memory (LSTM)-based encoder-decoder model in [107]). We then evaluated the performance of the trained models on unseen data, including the algorithmically annotated [4*] Hollywood2-50 data set (Section 2.1.2.2) and the fixation and pursuit saliency maps for GazeCom corresponding to the manual expert annotations of these eye movements (Section 2.1.2.1), as well as CITIUS-R [71], an additional data set of video viewing.

For Hollywood2-50 and GazeCom, we tested our and literature models (including two recent deep learning models – ACLNet [70] and DeepVS [82]) against both fixation and SP ground truth. Meanwhile, CITIUS-R is a traditional saliency data set, with only fixation onset data (detected by a standard algorithm) provided by the authors. We also tested the models in a similar traditional saliency set-up on Hollywood2-50 and GazeCom in order to test a scenario directly comparable to the pipelines in the literature (though the results were very similar to testing against fixation samples detected by our more complex algorithm that accounts for SP).

We used a variety of existing metrics to test the models [92], proposing, however, a different averaging scheme for aggregating the scores between the videos in a data set – weighting the scores proportionally to the number of corresponding ground truth attention samples – *i.e.* fixation- or SP-labelled gaze points. We quantitatively demonstrated [6*] that *e.g.* for several receiver operating characteristic (ROC)-based metrics this is a significantly closer approximation of the same measure computed over all the samples in the data set, compared to simply averaging the scores for all videos.

We also directly evaluated the ability of different models to separate between video regions likely to induce fixations and pursuits. To this end, we computed an area under the curve (AUC) of an ROC, where the class samples are drawn from the respective sets of fixation and SP locations in the video. The saliency scores yielded by each model were considered as scores for choosing one of the classes over the other (*e.g.* SP over fixation). The resulting AUC score would reflect how much the models “favour” one eye movement class over the other (*i.e.* assign its corresponding locations higher saliency values). We found that while most of the models favoured SP on average, and in principle the classes were well-separable, the scores yielded by the tested models were relatively low.

3

Discussion and Conclusions

3.1 Eye Movement Classification

The first area to which this dissertation contributed is the detection of various eye movement classes. We were largely motivated by the identification of smooth pursuit (SP) episodes, which were, prior to this work, (and often still are) neglected in the eye tracking data processing pipelines.

To validate and optimise any algorithmic classification pipeline, some form of ground truth data or expert knowledge are needed. Previous data sets of eye movement annotations were modest in size [51, 61] and rarely accounted for pursuit. Larger-scale eye tracking data sets were chiefly collected for the problem of saliency prediction, and were accordingly automatically pre-processed into scanpaths that lack the temporal resolution necessary for high-quality eye movement analysis [70, 86, 87]. In contrast to these, we created a large-scale annotated set of eye movement recordings, where we labelled SP as well as fixations and saccades [4*]. This aligns with the general trend in the eye movement research to explore more dynamic and naturalistic set-ups [38, 44, 68, 108, 109, 1*, 9†], where traditionally defined fixations and saccades do not constitute the only eye movements of interest.

As to automatic eye movement detectors themselves, approaches developed in the literature before the algorithms introduced here were, first, most frequently ignoring SP entirely [32, 46, 47], and also relatively simple computationally – typically based on thresholding or simple, mostly hand-crafted, statistics [56, 57, 61, 110]. This work introduced two eye movement detection pipelines, both accounting for SP and both furthering the state of the art in their own way. We will discuss these in turn below.

The first approach (described in Section 2.1.3) uses traditional methods to detect fixations and saccades, while applying a clustering method to the gaze samples of several different observers in order to jointly identify the likely smooth pursuits in these. To the best of our knowledge, this is the only eye movement detector to date that utilises multiple eye tracking recordings for the same stimulus in its classification pipeline. This work has thus demonstrated that leveraging this multi-observer information can yield a useful signal for eye movement classification. The particular algorithm we chose is, however, close in nature to the hand-crafted approaches: We designed it to find dense groups of potential SP gaze samples according to a fixed set of rules. Future works could combine the benefits of multi-observer data with modern machine learning methods [52, 67, 3*] that can learn features directly from data.

There are several things to note about our SP detection algorithm itself as well, however. First of all, as it uses an unsupervised learning technique (clustering), it has a different relationship with data, compared to other models. Thresholding-based approaches (we apply this name to all models with pre-set parameters, either by an expert or some statistics, but *unchanging*) have no dependency on data volumes, as all their parameters are fixed in advance, and no new data would allow for altering those. Data-driven methods employ some form of machine learning to perform the classification. These algorithms require ground truth data to optimise the values of their parameters, *e.g.* [52, 62, 3*]. These models, especially those with many parameters, will be typically improved as more ground truth data become available – training such a system on a larger and more diverse data set usually leads to improved generalisation. Our experiments showed, however, that the performance of our clustering-based approach (despite its fixed parameters, which would render it a thresholding-based algorithm in this classification) also improves when the recordings of more observers are analysed at the same time. An important distinction here is that our model does *not* require the expert labels (the "ground truth"), but merely the recordings themselves. Not needing manual annotation to improve performance saves a lot of time, as annotation typically requires at least an order of magnitude more time than the corresponding recording session lasts [30, 4*].

We also note that our approach is not the first to use clustering in eye movement classification [50, 111, 112], not even for SP detection [58]. The crucial distinction is that while these works all used clustering, it was used *within a single recording* [111], and often within a small temporal window (*e.g.* 200 ms in [112] or the duration of each respective inter-saccadic interval in [58]). The approach in [50] operates on the whole recording, clustering microsaccade candidate events into no fewer than two groups, from which one is later selected based on a fixed criterion. This method could, for instance, benefit from analysing *longer* recordings (thus accumulating more examples for a more robust clustering), but not from analysing *more* recordings (unlike our clustering approach). The former has to do with the initial experiment design, while the latter allows for obtaining more eye tracking data post-hoc in order to improve classification.

The second algorithm for eye movement classification described in this dissertation (see Section 2.1.4) relies on a deep learning architecture both to extract features more complex than the hand-crafted filters and to perform simultaneous classification of all considered eye movement types. Unlike other machine learning algorithms for eye movement classification [60, 62, 63, 66], our model directly operates on windows of gaze data, instead of classifying gaze samples individually. Thus, our architecture outputs a sequences of labels. The authors of [67] recently employed a similar approach, achieving temporal aggregation by the combination of convolutional and pooling operations, compared to the long short-term memory (LSTM) layers we used. While they performed other interesting analyses, no influence of classification window size on model performance was reported. In our experiments, we demonstrated [3*] that increasing the size of these windows at the training stage improves the resulting model performance (especially for

SP detection), meaning that this property of working on larger temporal contexts can be immediately beneficial for eye movement classification.

A further argument for a model to produce windows of class probabilities has to do with the potential for applying more complex loss functions, such as *e.g.* connectionist temporal classification (CTC) [113], which accounts for the likelihood of the ground truth sequence of *events*, given the class probabilities of the individual samples (opposed to the typically employed loss functions that only evaluate individual sample classification). This possibility is especially relevant given the recent increase in the interest for event-level evaluation of the classifiers [30, 52, 68, 3*, 7*]. In a proof-of-concept study, we used CTC loss together with traditional categorical cross entropy to demonstrate the potential of this combination to improve both sample- and event-level classification [14†].

The last point we raise here concerns the data that serve as the basis for eye movement classification. The importance of this issue becomes clear in connection to the eye movement definitions and their annotation procedure. Experts often rely on stimulus data when assigning the labels [44, 68, 1*, 2†], while the vast majority of the algorithms do not utilise any information about the stimulus [46, 47]. This discrepancy will likely make correct classification unattainable for such automatic classifiers.

In our eye movement definitions and annotations (cf. Section 2.1.1), for example, we used the correspondence of gaze movement to the moving targets in the video to deal with potential ambiguities between noisy fixations and pursuits: Slow gaze motion that does not correspond to a potential target movement in the scene would be labelled as a fixation (likely affected by drift or other recording artefacts [114]), even if a pursuit-like trend is observed in the gaze coordinates. Indeed, several cases like this (with *e.g.* gaze motion perpendicular to the trajectories of all moving targets in the video) were present in the recordings of the GazeCom data set [69], and had to be labelled accordingly.

The few algorithms that do incorporate stimulus information in their decisions include [16] and [44], neither of which accounting for the gaze dynamics *as well as* the stimulus content around the gaze point, thus neglecting to explicitly consider the actual motion of the eye. The authors of [43] combined gaze movement features with *static* video frame features at the gaze location, *i.e.* forgoing the possibility to verify whether gaze and video motion are aligned.

To rectify this, dynamic gaze features and dynamic video content features should be employed together. One of the gaze classification baselines that we proposed in [7*] (video-based baseline; also see Section 2.1.5.1) used whole-frame movement features to determine whether most of the observers would pursue some target on this frame. Similar features at the gaze location could be combined with gaze features either in a rule-based [59] or machine learning [43] classification pipeline. The authors of [58] used object tracking to find moving shapes in video, comparing their movement to gaze. They only used this method for the purpose of testing an SP detector without any manual annotation and not as a stand-alone classifier, however.

Nevertheless, joint video and gaze data processing does seem promising, especially when de facto "standard" pre-trained video-processing deep networks emerge: A similar development for image data [115] has allowed *e.g.* for the gaze event classifiers in [43, 44].

3.2 Saliency Modelling

The second contribution area of this work is attention modelling. In particular, we developed one of the early approaches for 360° saliency prediction, as well as proposed a novel problem formulation for video attention, combining eye movement modelling with saliency prediction, thus furthering a much more developed field of video saliency.

In [5*] (also see Section 2.2.1) we introduced several techniques to allow for predicting omnidirectional saliency using conventional saliency models. Our approach accounted for the artefacts of 360° data representation, which necessarily arise if traditional models and pipelines are applied (even if dedicated training is performed [116]). Our work preceded such specialised computational tools as *e.g.* spherical domain-adapted convolutions [89, 117] becoming efficient tools for saliency prediction [118] and other tasks [119]. Nevertheless, our approach provides an easy entry point for 360° saliency prediction, being able to leverage any existing two-dimensional model. Due to its performance [95] and ease of usage, our model has been often used in recent works as a reference approach [120, 121, 122, 123] or as a step in a data processing pipeline [124].

It also addressed several general challenges of working with omnidirectional content. In fact, many new deep learning-based approaches are dealing with the same issues in a similar way: *E.g.* [85, 125, 126] use cube map projections to avoid distortions. The equirectangular image manipulations proposed in our work can also be used as a data preparation stage for a specially trained predictor network.

The second saliency-related contribution of this dissertation is combining eye movement class information with attention modelling (in the context of video viewing). Prior to this work, the eye tracking data for video saliency prediction were treated in much the same way as for image saliency – with the help of standard fixation detection algorithms [70, 71], which were mostly developed for gaze movements in response to static stimuli [32]. Our work laid the foundation for systematically utilising eye movement classes in saliency prediction: We proposed a novel problem setting – smooth pursuit-based saliency prediction, presented the factors differentiating it from traditional fixation-based saliency (as well as the necessary evaluation pipeline modifications), and demonstrated the practical benefits of training models for SP prediction. We have shown that the models that are trained to predict the dynamic pursuit behaviour generalise to unseen data sets better, compared to the same architectures trained for fixation prediction. Our pursuit-predicting model showed state-of-the-art performance for saliency prediction, on average across the data sets we considered. We hypothesise that its success as well as the generalisation effects have to do with the sparsity of SP data combined with its higher spatio-temporal density [4*]. Compared to fixations, object tracking via pursuit needs to be initiated and maintained, while fixations cannot be suppressed even in the absence of stimuli or attention.

While there are other methods that can be attempted to filter out inattentive viewing (*e.g.* approaches based on pupillometry [127], EEG [128], or even fMRI [129]), analysing eye movements allowed us to separate the data into segments of gaze to keep or to discard directly, without requiring an additional data modality.

The pursuit-based attention concept can also be expanded to the omnidirectional video domain by *e.g.* modelling the attention related to following targets with some combination of eye and head movement. To automatically detect different patterns in eye movements and eye-head coordination, we proposed a basic speed thresholding-based algorithm in [1*], which could be extended by verifying that the same object is being looked at continually [44]. This type of eye tracking data processing could pave the way to more insightful attention modelling in 360° or virtual reality stimuli, also aiding the recent effort to connect brain activity, attention, and eye movements [108, 109].

3.3 Implications and Applications

While the contributions of this dissertation advanced the respective areas of human visual perception modelling, they also lead to new possibilities in related research areas. A major focus point of this work is eye movement classification, and smooth pursuit detection in particular – both the classification systems themselves [3*, 4*, 2†, 12†], the data sets needed to develop these [1*, 4*, 2†, 9†], and the evaluation methods [3*, 7*]. While valuable in their own right, eye movement classifiers and their improved performance contribute to better data processing in other areas, enabling new research questions and problem settings. In principle, even a poor but somewhat systematic classifier could yield enough information to perform a higher-level task based on the characteristics of its imperfect outputs. In [62], however, the authors demonstrated *e.g.* that a higher-quality eye movement classifier leads to a better-functioning biometric system. As noted in [47], or as we have demonstrated in this work, the standard fixation and saccade detectors [32] often perform poorly in the set-ups with dynamic stimuli. Therefore, an eye movement classifier designed for a dynamic set-up is a prerequisite for a good-quality analysis.

For instance, many works dealing with neurological disorder classification based on eye tracking (*e.g.* to support the diagnosis or for large-scale screening purposes) rely on eye movement characteristics [130, 131, 132, 133, 134, 135, 136] or the attention allocation strategies in relation to certain areas or objects in the stimulus [6, 137, 138, 139, 140]. Combined with this diagnostic capacity of eye movement data, bringing reliable automated analysis to various dynamic experiment set-ups could advance these and other clinical applications [7, 136, 141, 142, 2*], enabling larger-scale studies and new experiment designs. In [11†], for example, we examined the fixation patterns (with fixations pre-detected via a standard algorithm) of typically developing and autism spectrum disorder children during image viewing – a relatively standard static-stimulus scenario. Our approach, which won the corresponding IEEE ICME 2019 Grand Challenge “Saliency4ASD” [140], is based on the fixation sequence properties (duration and location of fixations, transition amplitudes), augmented by saliency- and face-related features (quantifying how well the fixation locations align with automatically predicted saliency distribution and faces on the images). While this approach demonstrated promising results, a richer, more dynamic data set (*e.g.* gaze patterns during movie watching [2*], locomotion [136], or virtual world exploration), would allow for a much more elaborate analysis. Combining such data with *e.g.* our recent computational methods for both analysing the gaze signal [1*] and predicting saliency [5*, 6*] would help produce additional insights into the neurological and behavioural underpinnings of the visual system [108].

As another example, gaze-based interaction also often relies on eye movement classification [41, 42, 143, 144, 145, 146], and should benefit from the detector quality improvement, similar to the biometric application in [62]. Extending the eye movement analyses to new contexts – such as mobile or virtual reality eye tracking [43, 44, 68, 1*] – enables novel interaction concepts and methods [147, 148].

In [149], the authors proposed using the properties of the performed smooth pursuits (with known synthetic targets) to assess the cognitive workload of the observer, discussing the applicability of this analysis in conjunction with interfaces that incorporate moving elements in order to *e.g.* perform real-time workload level adjustments. Being able to analyse SP without knowing the targets in advance [3*, 4*], and potentially also in other, more immersive contexts [1*], would enable extending such systems to new domains, keeping pace with the developments of visualisation interfaces [150, 151].

Overall, better understanding and modelling the visual system on different levels should aid a variety of research fields, from those directly related to human vision – *e.g.* detecting visual field abnormalities based on gaze patterns [135, 152, 153], to computer vision – *e.g.* finding the particularly important or informative regions of the stimuli [16, 154, 6*], to – thanks to the ubiquity of vision – seemingly unrelated areas, such as human-machine interaction [42, 144] or neurological disorder classification [136, 2*].

3.4 Conclusions

The work presented here has expanded the scope of human visual perception analyses undertaken in the literature, enabling and improving automated modelling on various fronts: from eye movement classification to saliency analysis in new domains (in omnidirectional content or related to explicitly incorporating eye movement categories into saliency prediction).

In particular, throughout this dissertation we have demonstrated that smooth pursuit represents a significant part of human gaze behaviour, and should not be discarded or ignored, and neither can it be treated in the same way as fixations. We have empirically shown the properties of SP that set it aside from other eye movement types: (i) it is more difficult to detect than the traditionally considered fixations and saccades, so specialised detectors could be designed; (ii) training to predict SP-based saliency instead of fixations improves attention model generalisability; (iii) the voluntary nature of SP (compared to fixations, which cannot be suppressed) leads to its higher sparsity and, consequently, requires adjusting the saliency evaluation pipeline for this eye movement type, as different stimuli can have widely varying amounts of it.

All this once again points out that traditional eye movement analyses that do not take SP into account are not suitable for dynamic stimulus content, even if solely fixation or saccade detection is performed. Our pipelines for eye movement annotation, classification, and evaluation, as well as saliency prediction and evaluation, were developed with pursuit in mind, improving SP handling across the board.

A

Smooth Pursuit Detection Based on Multiple Observers

This paper introduced the idea of detecting smooth pursuit (SP) by aggregating inter-observer information via clustering. In our pipeline, we first processed the recordings of different observers independently, removing saccade, blink, and fixation gaze samples from the data. We then collected all remaining gaze samples ("SP candidates") from all observers (for each stimulus video separately). These were processed with one of the three clustering techniques we tested: DBSCAN [97], level-set tree [155], and graph connectivity-based clustering.

Depending on the clustering method, some filtering of its outputs might be needed. While DBSCAN was designed to discard low-density outliers, the level-set tree implementation of [156] includes methods to prune the cluster tree with fewer elements than a given threshold (a relatively high threshold of 10% of the processed data was set by default, leading to a high-specificity but low-sensitivity detector). For graph-based clustering, we implemented cluster filtering based on the duration of the cluster (*i.e.* the time between the earliest and the latest gaze point belonging to it) and its diversity (*i.e.* how many different observers contributed samples to a given cluster).

The approach was tested on a 10% subset of the GazeCom data set [69], which we crudely labelled (windows of 250 ms labelled with a single label – either "mostly pursuit" or "mostly not pursuit") for this study. We compared our method against two literature models that represented the state of the art at the time ([54] and our re-implementation of the algorithm in [56]), demonstrating the superior performance of our algorithm. While DBSCAN and graph-based clustering performed similarly, the latter had ca. 70 times shorter runtime, in part owing to our efficient implementation of the neighbourhood relationship verification function for two gaze points.

This is a shared first authorship work with Ioannis Agtzidis, so some contributions are shared. My personal contributions consist of (i) developing the idea of using clustering to detect similar gaze movement patterns in the recordings of several observers; (ii) implementing two of the three clustering algorithms for SP detection tested in this work (graph-based clustering and level-set tree); (iii) participating in the data annotation and labelling interface implementation, and (iv) co-writing the paper.

Smooth Pursuit Detection Based on Multiple Observers

Ioannis Agtzidis[*]
Technische Universität München

Mikhail Startsev[†]
Technische Universität München

Michael Dorr
Technische Universität München

Abstract

While many elaborate algorithms to classify eye movements into fixations and saccades exist, detection of smooth pursuit eye movements is still challenging. Smooth pursuits do not occur for the predominantly studied static stimuli; for dynamic stimuli, it is difficult to distinguish small gaze displacements due to noise from smooth pursuit. We propose to improve noise robustness by combining information from multiple recordings: if several people show similar gaze patterns that are neither fixations nor saccades, these episodes are likely smooth pursuits. We evaluated our approach against two baseline algorithms on a hand-labelled subset of the GazeCom data set of dynamic natural scenes, using three different clustering algorithms to determine gaze similarity. Results show that our approach achieves a very substantial increase in precision at improved recall over state-of-the-art algorithms that consider individual gaze traces only.

Keywords: eye movements, smooth pursuit, clustering

Concepts: •**Applied computing** → **Psychology;**

1 Introduction

Humans constantly sample their visual surroundings by moving their eyes, and where they look largely determines what visual information will be processed. Naturally, eye movement patterns are therefore of interest to researchers both inside and outside the laboratory. In experiments that utilize static stimuli such as texts or pictures of natural scenes, gaze data can be broadly classified into two categories: *fixations*, relatively stationary phases that typically last several hundreds of milliseconds and during which visual information is processed from a specific image location; and *saccades*, ballistic eye movements that last only 20-80 ms and reach speeds of up to 800 deg/s. In principle, this classification should be relatively easy because of differences in speed and dispersion; in practice, this distinction is more difficult because of artefacts in contemporary eye-tracking equipment, such as jitter, post-saccadic oscillations [Nyström et al. 2013], or drift due to pupil size changes [Drewes et al. 2012].

Further eye movement types such as optokinetic nystagmus and the vestibular-ocular reflex are introduced in gaze recordings with more naturalistic stimuli and tasks, e.g. with head-mounted eye trackers and freely moving observers [Munn et al. 2008]. Even in the relatively simple case of dynamic stimuli in an otherwise static setup, i.e. videos on a computer screen, smooth pursuit eye movements may occur. Such pursuit movements serve to track and keep moving objects foveated, and may thus also be considered as a "fixation on a moving target". Because of their potentially low speed, smooth pursuit eye movements are difficult to distinguish from fixations. Nevertheless, several solutions for pursuit detection have been proposed [Ferrera 2000; Berg et al. 2009; Larsson et al. 2013]. In the absence of ground truth data, however, the choice of classification thresholds must remain arbitrary to some extent; for example, to detect smooth pursuit episodes with high specificity, one could set high speed or duration thresholds under the assumption that fixations are stationary and that noise artefacts should be transient. However, this runs the risk of discarding short and slow pursuits.

Here, we propose to increase specificity of smooth pursuit detection with a more principled approach by using information from multiple observers. Given a sufficiently large number of observers, we assume that potential pursuit targets will be tracked by more than one observer at a time. This means that our confidence that a particular gaze trace at a certain spatio-temporal location is indeed smooth pursuit is increased by the presence of similar gaze traces at the same location. Eye-tracking artefacts such as noise or post-saccadic oscillations, and other slow eye movement signals such as glissades and vergence movements, on the other hand, should be independent of pursuit targets and are thus less likely to occur in the same spatio-temporal location and with similar direction across multiple observers.

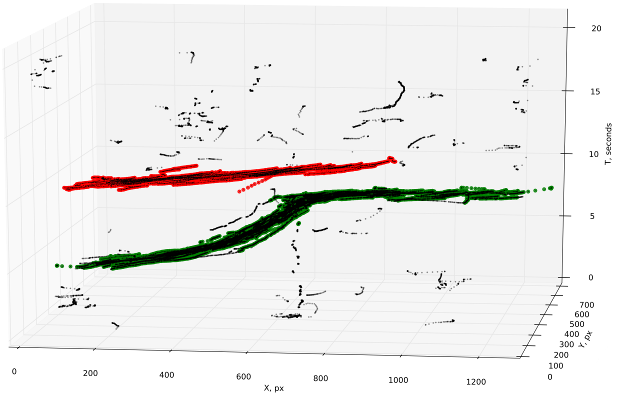


Figure 1: *Example clustering for the 'ducks_boat' movie, which shows ducks flying by a boat moored on a river. Each point represents an (x, y, t) gaze sample that was identified as neither fixation nor saccade during prefiltering. Coloured areas indicate clusters, i.e. video regions where such gaze samples from several observers coincide and likelihood of pursuit is higher; remaining samples are likely eye-tracking artefacts.*

In this paper and as a proof of principle, we apply our technique to detect smooth pursuit episodes in the GazeCom set of gaze recordings on dynamic natural scenes [Dorr et al. 2010]. Schematically, this technique is illustrated in Figure 1. Every data point in the spatio-temporal volume represents a gaze sample that could not be identified as either fixation or saccade during an initial filtering step. Highlighted in colour are those samples that fell into clusters, i.e. where different observers showed similar gaze patterns outside clearly marked fixations and saccades; for this example video, these

[*]The first two authors contributed equally.

[†]e-mail: first.last@tum.de

ETRA '16, March 14-17, 2016, Charleston, SC, USA
ISBN: 978-1-4503-4125-7/16/03
DOI: http://dx.doi.org/10.1145/2857491.2857521

clusters correspond to two ducks flying across the screen from left to right (one from 5–10 s, with a briefly upward trajectory inbetween; one from 11–13 s). For a dynamic visualization, please also see the companion video to this paper.

2 Methods

2.1 Gaze data

We ran our offline analysis on the publicly available GazeCom eye movement data set [Dorr et al. 2010] that contains gaze data from 54 subjects for 18 video clips of about 20 s duration each, totalling about 40000 saccades. These video clips show outdoor scenes in and around a city with a number of potential pursuit targets, such as moving cars and pedestrians.

2.2 Data pre-filtering

If multiple observers fixate the same image region at the same time, their gaze patterns will be similar even in the absence of smooth pursuit. We therefore pre-filtered the raw data in order to remove blinks, saccades, and clearly identified fixations. After blink removal, saccades were detected using the dual-threshold method from [Dorr et al. 2010] and subsequently removed. Following this, fixations were removed in two steps: First, all inter-saccadic intervals with a dispersion of less than 2 degrees were marked as fixations. Then, a temporal window of 50 ms width was shifted samplewise across the remaining data and a non-fixation onset (offset) was marked every time speed rose above (fell below) 2 deg/s. Finally, only episodes where non-fixation onset and offset were more than 50 ms apart were kept; overall, pre-filtering removed about 90% of the raw gaze data.

2.3 Clustering

For the purpose of this paper, we were interested in similarities of gaze trajectories across observers, i.e. spatio-temporal information rather than spatial locations only. Therefore, we clustered gaze data in three-dimensional (x, y, t) space. Whereas clustering often is aimed at maximizing the compactness of clusters, we wanted to maximize some notion of connectivity: pursuit targets in our data set followed elongated trajectories with occasional changes of direction.

In order to assess the robustness of our approach to variations in clustering results, we evaluated three different clustering approaches. We had to make changes to the standard implementations of all of these algorithms because *a priori*, there is no optimal scaling factor between space and time.

2.3.1 Graph-based clustering

This algorithm is not based on point density in the 3D space, but on an empirical concept of 'neighbour points'. Gaze data are represented as a graph, and its connected components are determined.

The nodes of the graph are all the gaze samples, and the edges exist between all the points that are close enough in the 3D space. Our definition of 'close enough' was *points A and B are neighbours if $\mid A.t - B.t \mid \leq \tau$ and the Euclidean distance between A and B in XY-plane $\rho_{xy}(A, B) \leq R$*, with R=2 deg and τ=4 ms.

Since this clustering algorithm does not rely on density information, many small and short clusters resulted; therefore, clusters with less than 10% of all observers or a duration of less than 50 ms were treated as noise.

2.3.2 DBSCAN clustering

The two following algorithms consider density as the main criterion for cluster identification. The first is an adaptation of DBSCAN [Ester et al. 1996]; as spatio-temporal distance metric, we used the spatial Euclidean distance for any point pair within a 40 ms time slice. The DBSCAN parameters eps (analogous to R above) and $minPts$ were set to 2 degrees and the number of available observers per movie, respectively.

2.3.3 Level-set tree clustering

Finally, we also used the python implementation [Kent et al. 2013] of the level set tree (LST) clustering algorithm [Chaudhuri and Dasgupta 2010].

The algorithm treats the data as realization of a distribution with an unknown density function in a d-dimensional space ($d = 3$ in our case). As before, a separate treatment of time and space coordinates was required; here, we subdivided the whole space into rectangular parallelepipeds with 'height' τ along the time axis and 'width' R along both spatial axes (same as above). Then, for each such parallelepiped only its own and its neighbours' gaze points were considered. For these points the time component was removed (resulting in a projection onto the (x, y) plane) and the density was then estimated with a kernel-density estimation with a Gaussian kernel.

3 Evaluation

We evaluated the performance of the proposed approach on a subset of the GazeCom dataset against two state-of-the-art algorithms [Berg et al. 2009] (provided through the authors' toolbox) and our own reimplementation of [Larsson et al. 2015]. SPs were hand labelled independently by three raters (the authors of this paper) and the majority vote was computed for each sample.

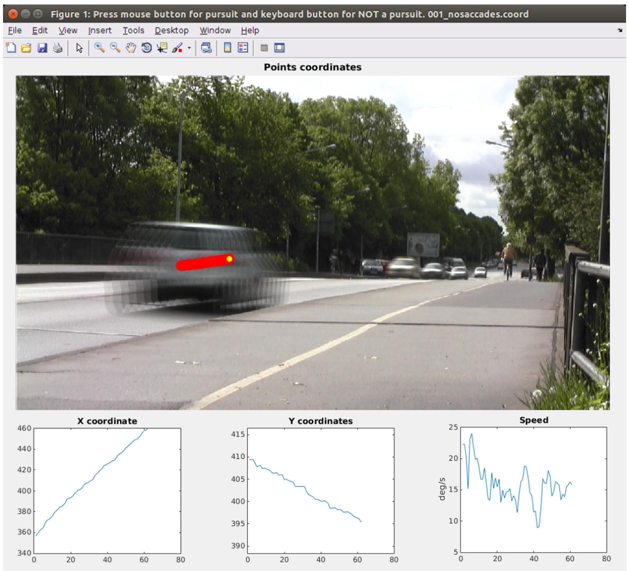


Figure 1: Press mouse button for pursuit and keyboard button for NOT a pursuit. 001_nosaccades.coord

Figure 2: *Hand labelling GUI.*

We first randomly chose one two-second interval from each video and labelled gaze episodes as pursuit or non-pursuit for each observer in these intervals. For efficiency, gaze samples were grouped into non-overlapping temporal windows of 250 ms which were then labelled as a whole. A Matlab interface was devised that presented a visualization of gaze trace and raw data x and y coordinates as well as speed in deg/s over time. To provide video context that allowed the raters to judge whether a fixated object was moving or not, gaze traces were overlaid on the temporally averaged video frames in the 250 ms window (see an example screenshot in Figure 2). Raw data plots served raters to identify abrupt changes in gaze traces that might be invisible at the coarser scale of the video frames. By either a mouse click or a key press, raters classified each window as SP or not; they were instructed to treat the whole window as SP if more than half of the gaze samples could be regarded as part of a SP.

For quantitative evaluation, we computed precision, recall, and F1 scores as well as false positive rate over the concatenation of all intervals. Overall, about 70000 gaze samples out of 385000 (approx. 18%) were labelled as SP by at least two raters. Average pair-wise inter-observer agreement rate was 89.3%.

4 Results

Evaluation results are presented in Table 1. It can be seen that both graph-based clustering and DBSCAN yield a dramatic increase in precision while substantially improving recall at the same time. The number of false positive errors was decreased about fourfold. LST clustering achieves a further considerable increase in precision while having relatively low recall.

Table 1: *Precision, recall, F1 and false positive rate (FPR) throughout all the ground truth data.*

	Precision	Recall	F1	FPR
Graph-based	0.865	**0.372**	**0.52**	0.013
DBSCAN	0.88	0.361	0.512	0.01
Level-set tree	**0.963**	0.115	0.206	**0.001**
Berg et al.	0.599	0.306	0.405	0.045
Larsson et al.	0.54	0.235	0.327	0.044

For each of the 18 movies in the GazeCom data set, the proportion of gaze samples that were labelled as pursuit is shown in Figure 3. All of our algorithms detect no pursuit episodes at all on several movies, whereas the reference algorithms never label less than 3.5%/2.7% of samples as SP per video. The maximal rate of gaze samples labelled as pursuit is 15% and 8% for graph-based/DBSCAN and LST clustering, respectively and 16%/14% for Berg et al./Larsson et al, respectively. Based on these results and recall/precision figures from the evaluation stage, we can estimate that observers spent about 2.5 s on average per movie doing smooth pursuit. DBSCAN and graph-based clustering yielded very similar rates; based on LST clustering, fewer episodes were labelled as pursuit, but the relative ranking of movies was similar for all three algorithms. Graph-based clustering not only showed better results, but also had the lowest computational cost (in our experiment, 21 s vs. 470 s and 1440 s for LST and DBSCAN, respectively for the entire GazeCom data set).

5 Discussion

Many sophisticated algorithms to automatically label raw eye movement data with meaningful descriptors exist to date. However, eye-tracking devices still suffer from recording artefacts that may differ between subjects and between different device manufacturers. Short of neurophysiological recordings, there is no 'objective' ground truth what the visual system might have intended, and thus any performance evaluations necessarily require at least some assumptions. As a consequence, it is impossible to calibrate such algorithms towards universally optimal parameters. Thus, it is still considered good practice to manually inspect at least a sample of automatically labelled data as a sanity check (and, if necessary, adjust the labels), which is time consuming and potentially introduces individual biases.

For the problem of smooth pursuit detection, we here proposed a more principled approach that is based on the idea that smooth pursuit targets will elicit similar gaze patterns across observers, while noisy artefacts should be independent of these targets. Therefore, we clustered those gaze episodes that were neither clearly identified fixations nor saccades, and assigned the pursuit label to relatively dense clusters.

Clearly, this approach also introduces some free parameters, e.g. what constitutes sufficient similarity between gaze patterns during clustering, or how many observers need to be represented in a cluster to warrant a pursuit label. However, these parameters are less susceptible to minor gaze signal perturbations than parameters that are applied directly to the raw eye movement signal. While stressing that this cannot be considered an objective ground truth, we hand-labelled pursuit episodes in almost half an hour of gaze recordings on naturalistic scenes, and evaluated three different clustering algorithms. As a baseline, we also evaluated two state-of-the-art algorithms that base their classification on the characteristics of individual gaze traces only. Our results showed that level-set tree clustering achieved almost perfect precision (0.96) at the cost of low recall. DBSCAN and graph-based clustering still gave greatly improved precision compared to the state of the art while also improving recall. It should be noted that we here only clustered raw (but pre-filtered) gaze samples; in principle, the same approach could be applied to the output of existing pursuit detection algorithms, potentially further increasing robustness.

Under laboratory conditions, it has been shown that visual processing is altered during smooth pursuit, e.g. motion perception is enhanced [Spering et al. 2011]. Pursuit behaviour may also be impaired in clinical populations [Nagel et al. 2007], and eye movement features have been previously used successfully to support diagnosis of neurological disorders [Tseng et al. 2013]. To our knowledge, however, smooth pursuit has not been specifically quantified for dynamic natural scenes yet; the authors in [Li et al. 2010] translated natural image patches on an otherwise blank background to assess the importance of target size and speed. Especially for clinical purposes, it clearly would further be desirable to base any comparative evaluation only on those gaze episodes that very likely are pursuit, and thus high precision may be more critical than recall.

In a first analysis, we therefore looked at the pursuit episodes identified by our approach. Naturally, these will be strongly determined by the data set and the range of possible pursuit targets. The GazeCom data set is limited in this regard with only 18 different scenes; unfortunately, larger gaze corpora for dynamic stimuli are typically based on professionally produced material with camera motion and scene cuts. Nevertheless, some preliminary observations can be made even on this data set already.

The relative rate of pursuit episodes was higher for movies with isolated pursuit targets that suddenly appear (e.g. a duck flying by); in contrast to this, the 'roundabout' clip constantly shows many moving targets, but elicited fewer pursuits. Whether these results are artefacts of the present data set or represent general preferences of the oculomotor system remains to be addressed in future work.

Acknowledgements

This research was supported by the Elite Network Bavaria, funded by the Bavarian State Ministry for Research and Education.

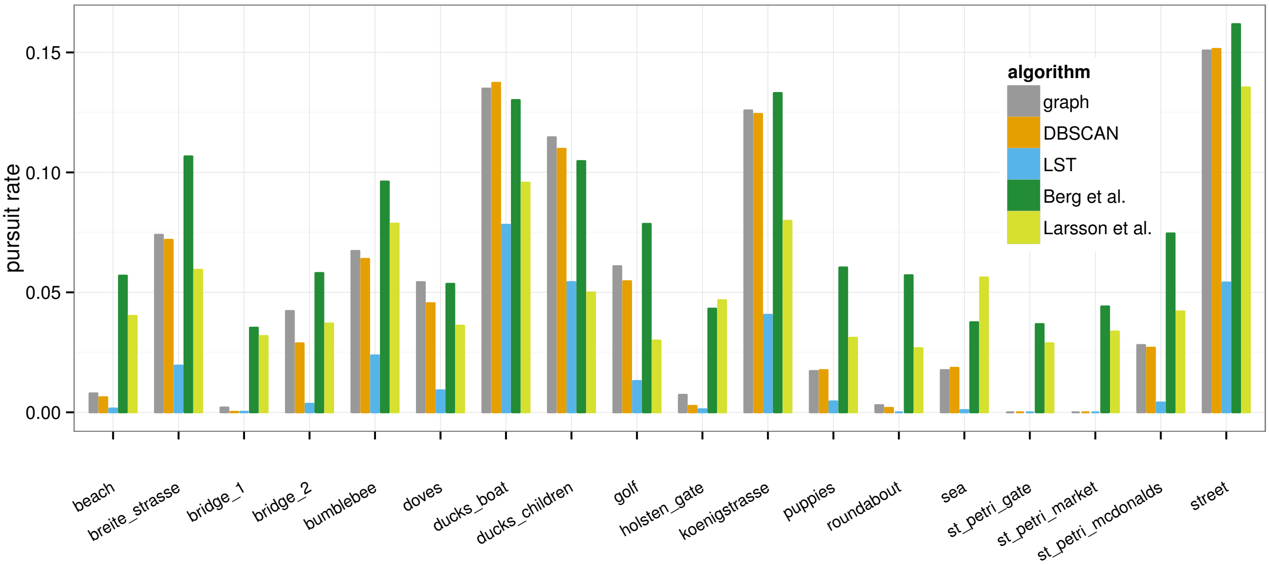


Figure 3: *Rate of gaze samples labelled as smooth pursuit for each movie of the GazeCom data set.*

References

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. 1999. Optics: Ordering points to identify the clustering structure. *ACM SIGMOD99*.

Berg, D. J., Boehnke, S. E., Marino, R. A., Munoz, D. P., and Itti, L. 2009. Free viewing of dynamic stimuli by humans and monkeys. *Journal of Vision 9*, 5 (5), 1–15.

Chaudhuri, K., and Dasgupta, S. 2010. Rates of convergence for the cluster tree. In *Advances in Neural Information Processing Systems*, 343–351.

Dorr, M., Martinetz, T., Gegenfurtner, K., and Barth, E. 2010. Variability of eye movements when viewing dynamic natural scenes. *Journal of Vision 10*, 10, 1–17.

Drewes, J., Masson, G. S., and 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*, ACM, 209–212.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD Proceedings*, vol. 96, 226–231.

Ferrera, V. P. 2000. Task-dependent modulation of the sensorimotor transformation for smooth pursuit eye movements. *Journal of Neurophysiology 84*, 6, 2725–2738.

Hartigan, J. A. 1975. *Clustering Algorithms*, 99th ed. John Wiley & Sons, Inc., New York, NY, USA.

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

Kent, B. P., Rinaldo, A., and Verstynen, T. 2013. Debacl: A python package for interactive density-based clustering. *arXiv preprint arXiv:1307.8136*.

Larsson, L., Nyström, M., and Stridh, M. 2013. Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit. *Biomedical Engineering, IEEE Transactions on 60*, 9, 2484–2493.

Larsson, L., Nyström, M., Andersson, R., and Stridh, M. 2015. Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. 145–152.

Leigh, R. J., and Zee, D. S. 2006. *The Neurology of Eye Movements*, fourth ed. Oxford University Press.

Li, F., Pelz, J. B., and Daly, S. J. 2010. Effects of stimulus size and velocity on steady-state smooth pursuit induced by realistic images. In *IS&T/SPIE Electronic Imaging*, International Society for Optics and Photonics, 752717–752717.

Munn, S. M., Stefano, L., and 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*, ACM, 33–42.

Nagel, M., Sprenger, A., Nitschke, M., Zapf, S., Heide, W., Binkofski, F., and Lencer, R. 2007. Different extraretinal neuronal mechanisms of smooth pursuit eye movements in schizophrenia: an fmri study. *Neuroimage 34*, 1, 300–309.

Nyström, M., Hooge, I., and Holmqvist, K. 2013. Post-saccadic oscillations in eye movement data recorded with pupil-based eye trackers reflect motion of the pupil inside the iris. *Vision research 92*, 59–66.

Spering, M., Schütz, A. C., Braun, D. I., and Gegenfurtner, K. R. 2011. Keep your eyes on the ball: smooth pursuit eye movements enhance prediction of visual motion. *Journal of Neurophysiology 105*, 4, 1756–1767.

Tseng, P., Cameron, I. G. M., Pari, G., Reynolds, J. N., Munoz, D. P., and Itti, L. 2013. High-throughput classification of clinical populations from natural viewing eye movements. *Journal of Neurology 260* (Jan), 275–284.

B

Characterizing and Automatically Detecting Smooth Pursuit

This work deals with both manual and algorithmic annotation of eye movements in gaze recordings. First, we here presented the manual expert annotations of the eye movements in a large data set of eye tracking recordings – GazeCom [69], which contains over 4.5 h of gaze data. We provided the definitions for the eye movement classes that we annotated, and described the annotation pipeline (two novice annotators working in parallel, then an expert annotator reconciling and amending their labels).

In the annotated data set, we, for the first time in the literature, characterised several aspects of smooth pursuit behaviour in dynamic natural scene free-viewing. For example, we quantified the spatio-temporal congruency of the annotated eye movement classes between different observers. This quantification revealed that, while only accounting for ca. 11% of the gaze samples, the samples attributed to smooth pursuits of different video viewers are more densely allocated in video volume, compared to fixations (which account for the vast majority of gaze data – over 70%).

Additionally, we first improved our previously proposed smooth pursuit detection method ([3†] and Appendix A) via random grid-based parameter optimisation, then implemented it in a form of a publicly available framework for eye movement detection. This framework not only detects the eye movements that we considered in our research (*i.e.* fixations, pursuits, saccades, and noise), but also implements a number of evaluation metrics from the literature [30, 52, 66, 3*, 7*].

We additionally thoroughly evaluated our smooth pursuit detection method to determine the influence of the number of observers, whose recordings are analysed at the same time, on the algorithm's performance. This directly demonstrated the benefits of aggregating inter-observer information for eye movement classification.

This is a shared first authorship work with Ioannis Agtzidis, so some contributions are shared. My personal contributions consist of (i) performing the spatio-temporal congruence analysis in the labelled data set, (ii) performing algorithm parameter optimisation, (iii) implementing the framework for eye movement classification, including all eye movement detectors, evaluation methods, a console interface, as well as (iv) testing the final model in various set-ups and (v) writing the larger part of the manuscript.

Characterizing and automatically detecting smooth pursuit in a large-scale ground-truth data set of dynamic natural scenes

Mikhail Startsev* Human-Machine Communication, Technical University of Munich, Munich, Germany

Ioannis Agtzidis* Human-Machine Communication, Technical University of Munich, Munich, Germany

Michael Dorr Human-Machine Communication, Technical University of Munich, Munich, Germany

Eye movements are fundamental to our visual experience of the real world, and tracking smooth pursuit eye movements play an important role because of the dynamic nature of our environment. Static images, however, do not induce this class of eye movements, and commonly used synthetic moving stimuli lack ecological validity because of their low scene complexity compared to the real world. Traditionally, ground truth data for pursuit analyses with naturalistic stimuli are obtained via laborious hand-labelling. Therefore, previous studies typically remained small in scale. We here present the first large-scale quantitative characterization of human smooth pursuit. In order to achieve this, we first provide a methodological framework for such analyses by collecting a large set of manual annotations for eye movements in dynamic scenes and by examining the bias and variance of human annotators. To enable further research on even larger future data sets, we also describe, improve, and thoroughly analyze a novel algorithm to automatically classify eye movements. Our approach incorporates unsupervised learning techniques and thus demonstrates improved performance with the addition of unlabelled data. The code and data related to our manual and automated eye movement annotation are publicly available via https://web.gin.g-node.org/ioannis.agtzidis/gazecom_annotations/.

Introduction

The rapid decrease of visual resolution away from the fovea renders the movement of the eyes essential for perception and action in our complex and dynamic visual world. Segmentation of eye movements into discrete events is an important part of eye movement research and has been investigated for decades. Although we discuss the definitions of the particular eye movement types later in the paper, reliably separating gaze events from one another enables a large number of analyses of eye tracking data sets in order to search for group differences or similarities (Dowiasch et al., 2016; Silberg et al., 2019), find the differences in viewing behavior for different stimulus types (Vig, Dorr, Martinetz, & Barth, 2011), and many other research applications, including media summarisation (Salehin & Paul, 2017).

For both precise quantification of eye movements and the development of automatic algorithms for their detection, ground truth data are required. Such data are typically acquired via manual annotation (Larsson, Nyström, & Stridh, 2013; Santini, Fuhl, Kübler, & Kasneci, 2016; Andersson, Larsson, Holmqvist, Stridh, & Nyström, 2017; Steil, Huang, & Bulling, 2018), which is a time-consuming process, often requiring the effort of multiple raters. This problem led to a relatively small scale of the previously conducted studies (for reference, the data sets in the works listed above range 3–25 min). I. T. C. Hooge, Niehorster, Nyström, Andersson, and Hessels (2018) concluded that while experienced yet untrained annotators often do not produce well-agreeing fixation annotations, human expertise still represents the gold standard for complex, ill-defined cases, which could include setting borders between fixations and postsaccadic oscillations or slow pursuits.

In order to quantify the eye movements with dynamic naturalistic stimuli on a larger scale, we here collected what is, to the best of our knowledge, the

Citation: Startsev, M., Agtzidis, I., & Dorr, M. (2019). Characterizing and automatically detecting smooth pursuit in a large-scale ground-truth data set of dynamic natural scenes. *Journal of Vision*, *19*(14):10, 1–25, https://doi.org/10.1167/19.14.10.

https://doi.org/10.1167/19.14.10 Received February 22, 2019; published December 12, 2019 ISSN 1534-7362 Copyright 2019 The Authors

largest manually annotated eye tracking data set that accounts for smooth pursuit (SP, foveating an object moving relative to the observer via an eye movement). We collected the manually annotated eye movement class labels for a set of 18 dynamic natural scenes, viewed by a multitude of observers in the established GazeCom data set (Dorr, Martinetz, Gegenfurtner, & Barth, 2010). The labelled data set amounts to a total of over 4.5 hours of eye tracking data, all samples assigned to one of the four categories: fixation, saccade, smooth pursuit, and noise. The latter was employed during blinks, out-of-monitor gaze, and naturally impossible gaze traces (i.e., the likely recording noise).

Although the size of other data sets in the literature would be sufficient for small- to medium-scale gaze pattern analysis and evaluation of eye movement detection algorithms, such amounts of data do not allow for meaningful algorithm parameter tuning, especially where machine learning is involved. For deep learning models specifically, with their thousands of parameters (Startsev, Agtzidis, & Dorr, 2019; Zemblys, Niehorster, & Holmqvist, 2019), the amount of available data, as well as their diversity, are crucial for the development and refinement of sophisticated models that could further improve the state of the art in eye movement classification. Our data set, with its millions of annotated gaze samples and tens of thousands of labelled events, sets a new yardstick for data set scale and enables the meaningful training of highly parametrized classification models, as well as makes large-scale analyses of naturalistic viewing behavior possible.

As spontaneously occurring pursuit behavior in naturalistic video viewing has not been quantified in the literature, we set out to characterise it in this study. Having manually annotated the GazeCom data set recordings, we report on the amount and properties of SP in this large-scale eye tracking data set, describing and discussing the relations between different eye movements in this context. For example, in our free-viewing gaze data we observed that pursuits cover a nonnegligible percentage of recorded gaze samples (ca. 11%), even more than is covered by saccades. We additionally explicitly explored the congruence between the eye movements performed by different observers, thus for the first time directly numerically characterizing the synchrony—in space and time—of fixations, saccades, and pursuits. We found that, even though most of the time the observers spent fixating, smooth pursuits were performed by a larger number of people at the same time and at the same place.

While this work presents a large-scale analysis of eye movements in its own right, it also demonstrates that considerable effort is required to obtain reliable annotations. To facilitate studies involving eye movements without the need to perform expert annotations for every analysed recording, algorithmic eye movement classification approaches are being constantly developed and refined. This strive for robust and accurate automatic analysis resulted in an impressive number of algorithms for eye movements classification that exist to date. Many of them rely on simple speed or dispersion thresholding (Salvucci & Goldberg, 2000; Komogortsev & Karpov, 2013), while others use more elaborate analyses such as principal component analysis (Berg, Boehnke, Marino, Munoz, & Itti, 2009; Larsson, Nyström, Andersson, & Stridh, 2015) or Bayesian inference (Santini et al., 2016). Lately, machine learning approaches have been applied to eye movement classification (Vidal, Bulling, & Gellersen, 2012; Anantrasirichai, Gilchrist, & Bull, 2016; Zemblys, Niehorster, Komogortsev, & Holmqvist, 2018) with promising results. Most recently, deep learning models have emerged as the new state of the art for eye movement detection (Startsev, Agtzidis, & Dorr, 2019; Zemblys et al., 2019).

Traditionally, automatic analysis performed based on the subjects' eye movements relied either on detecting fixations and saccades (Williams, Loughland, Gordon, & Davidson, 1999), or on analyzing the recordings that correspond to synthetic stimuli (Spering, Schütz, Braun, & Gegenfurtner, 2011), where targets for smooth pursuit, for example, are limited and have well-defined properties. Recent works show a tendency towards naturalistic stimuli, however, which include dynamic content as well (Dowiasch et al., 2016; Silberg et al., 2019). For these, even a seemingly simple analysis that is limited to fixations and saccades may be prone to errors because of the accidental inclusion of pursuit samples (Dorr et al., 2010). In their recent review, Andersson et al. (2017) indeed found that the algorithms designed without SP in mind would often falsely detect fixations instead. This accounted for the vast majority (over 70%) of misclassified gaze samples in their data, both in synthetic and realistic stimuli, albeit with the participants instructed to follow moving targets, which exacerbated this particular problem.

All this leads us to the observation that even though SP is an as important part of viewing behavior as are e.g., saccades, it is substantially underrepresented and often entirely overlooked in current eye movement detection approaches (Olsen, 2012; Mould, Foster, Amano, & Oakley, 2012; Kasneci, Kasneci, Kübler, & Rosenstiel, 2014; Anantrasirichai et al., 2016; Steil et al., 2018; Zemblys et al., 2019), highlighting the need to develop accurate pursuit classification algorithms (Andersson et al., 2017). It is of interest to note that one common property of all eye movement classification methods to date is that they only process one gaze recording of a single observer at a time, thus never accounting for the element of synchrony in the eye movements performed by various observers for the same stimulus (Startsev, Göb, & Dorr, 2019). This limitation has its benefits in terms of online applicability and the absence of additional data set restrictions, and it also seems to be sufficient for detecting saccades and fixations, which have relatively defined speed and acceleration ranges. For SPs, however, simple analysis of the speed of the gaze might not be sufficient to differentiate them from drifts (Yarbus, 1967, Chapter VI, Section 2), noisy fixations, or slow saccades (we present speed distributions later in the paper). Some algorithms, therefore, include acceleration thresholds in order to avoid misclassification of slow saccades as pursuits (e.g., (Mital, Smith, Hill, & Henderson, 2011) or the SR Research saccade detector (SR Research, 2009)). Mital et al. (2011) then simply combine all "nonsaccadic eye movements" into one category. While this is sufficient for some applications, various areas of eye movement research require distinguishing between different ways of looking at the gaze targets, in terms of execution or perception (Schütz, Braun, & Gegenfurtner, 2011; Spering et al., 2011; Silberg et al., 2019).

What additionally distinguishes pursuits is that they normally require a target in order to be executed. In artificial scenarios, where SP targets are generated with predefined speeds and trajectories, accurate detection of pursuit can be mostly achieved via matching the position of the gaze and position of the target at each given time. One should, of course, take catch-up saccades into account, but these are relatively easy to detect. In natural scenes, and in the absence of the detailed information about *all* the moving targets throughout the video, such matching is practically impossible. Dowiasch et al. (2016) computed optical flow of the video instead, using it as a substitute for gaze target speed, but during manual annotation of our data set we noticed that gaze samples were often offset relative to the targets they were following, likely due to tracking inaccuracy.

As a substitute for moving object detection in natural scenes, we recently proposed (Agtzidis, Startsev, & Dorr, 2016b) an SP detection algorithm that is based on a clustering of several observers' partial scanpaths, where fixation and saccade samples were eliminated in advance. This approach is based on the observation that multiple people will often track (pursue) the same objects of interest in natural scenes, as well as on the spatio-temporal eye movement congruency analysis performed in this work. Individual gaze traces will be noisy, so a significant portion of the gaze samples that would not be labelled as saccades or fixations could be attributed to recording or oculomotor artefacts. This noise, however, will be uncorrelated between the observers. If, on the other hand, several participants show similar gaze traces that are neither fixations nor saccades, these patterns are correlated and therefore less likely to be noise. Following this logic, we can obtain an indication of a reliably detected SP and filter out noise. A preliminary implementation of this approach (Agtzidis et al., 2016b) already demonstrated promising results for SP detection.

Figure 1 illustrates the detection patterns of this approach on an example of the *ducks_boat* video of the GazeCom data set (this video has two "main" moving targets—two ducks flying by—and several much slower moving, floating ducks). Here, the true positives (i.e., SP detected as SP, green traces), false positives (i.e., not SP labelled as SP, red traces), and false negatives (i.e., missed SP samples, blue traces) reveal both the benefits and the downsides of our approach: While most of the codirected pursuit episodes are successfully identified by our method, the nature of clustering leads to potential false detections where a dense group of samples was not discarded by the preceding steps of the algorithm, and potential missed detections, e.g., when the target was pursued by a single observer only.

The use cases and implications of the work presented in this manuscript extend beyond its immediate contributions (quantifying human eye movements in a large manually annotated data set and improving upon the state of the art of eye movement classification). The data presented in this work enables us and other researchers for the first time to quantify natural video-viewing behavior in terms of its constituent eye movements and their interactions or similarity between the observers (Startsev, Göb, & Dorr, 2019) on a comparatively large scale. The algorithmic analysis we propose allows for fully automated processing of the eye-tracking data sets, the size of which would make it difficult or well-nigh impossible to collect full expert annotations. Such analyses could further the research both in medical contexts (Lagun, Manzanares, Zola, Buffalo, & Agichtein, 2011; Tseng et al., 2013; Silberg et al., 2019), in computer vision applications dealing with human attention (Marat et al., 2009; Startsev & Dorr, 2018), and for attempting to understand the nature of human smooth pursuit in general (Hashimoto, Suehiro, Kodaka, Miura, & Kawano, 2003; Yonetani, Kawashima, Hirayama, & Matsuyama, 2012). Moreover, the unsupervised nature of our pursuit detection approach brings a unique property into the eye movement analysis field: This clustering-based algorithm is capable of improving detection quality and robustness by using more *unlabelled* data, i.e., without the need for additional annotations.

The manually labelled data set we collected is freely available via https://web.gin.g-node.org/ioannis.agtzidis/gazecom_annotations/ together with both our hand-labelling framework and automatic eye movement detection software. A detailed description of the

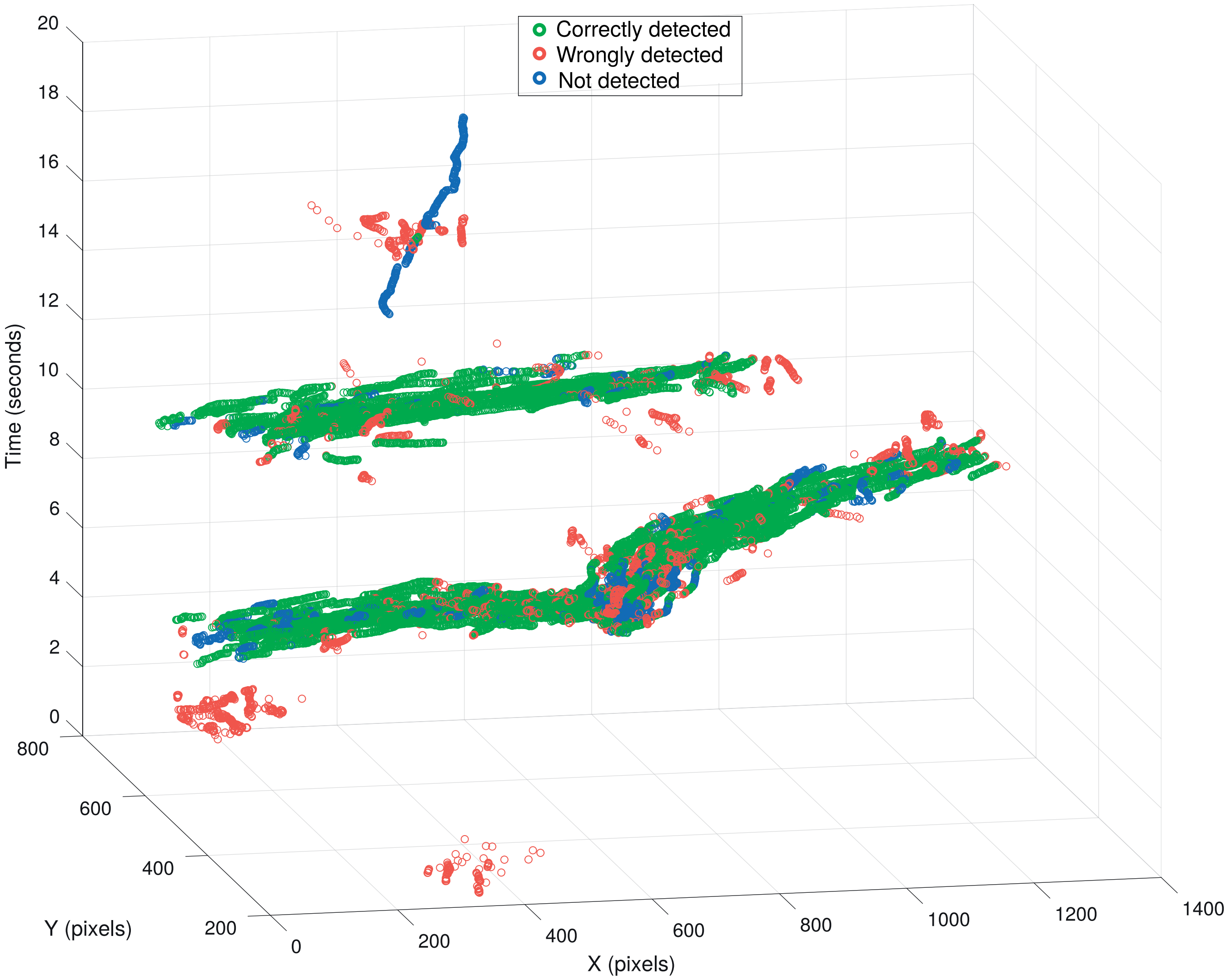


Figure 1. Visualization of clustering-based pursuit classification in one video of our data set *(ducks_boat)*. Data points for all observers are presented. Correctly detected smooth pursuit samples (in green) as well as detection errors (in red, false detections; in blue, missed samples) of our SP detection algorithm in the sp_tool framework.

latter, including the particularly relevant parameters and use cases, is provided in the Programmatic interface section.

Methods

In this section we describe the methodological details of the pipeline that we employed in order to collect a large annotated data set and construct an automatic tool for the segmentation of gaze traces into distinct eye movements. We start by describing the terminological and data-related background for this work, the labelling process that was used by the manual raters for the annotation of fixations, saccades, SP, and noise in the GazeCom data set. We then describe the classification and evaluation procedures of our eye movement detection framework.

Addressing terminological ambiguity

Before we proceed to describe further details of this work, we address several definitions that might be ambiguous or context-dependent, as they may differ in various set-ups of eye-tracking experiments or in various subfields (Hessels, Niehorster, Nyström, Andersson, & Hooge, 2018).

For example, throughout this manuscript we use the term “naturalistic” in order to describe the stimulus scenes in our data set. We use this term in the meaning of “imitating real life or nature” in accordance with other literature (Krieger, Rentschler, Hauske, Schill, & Zetzsche, 2000; Torralba, Oliva, Castelhano, & Henderson, 2006; Dorr et al., 2010; Tatler, Hayhoe, Land, & Ballard, 2011; McIlreavy, Fiser, & Bex, 2012; Smith & Mital, 2013; Parks, Borji, & Itti, 2015; Leder, Mitrovic, & Goller, 2016; Ramkumar et al., 2016; Foulsham & Kingstone, 2017; Schomaker, Walper, Wittmann, & Einhäuser, 2017; White et al., 2017). We describe our experimental set-up as naturalistic in part to contrast it with synthetic stimuli with prescribed, isolated eye movements often used for studies involving smooth pursuit (Vidal et al., 2012; Santini et al., 2016): Naturalistic stimuli represent a more complex set of visual inputs that affect oculomotor behavior (Monache, Lacquaniti, & Bosco, 2019), and the idea that the visual system is optimized to efficiently encode the inputs that surrounded our ancestors during evolution is well established (Field, 1987; Atick & Redlich, 1992).

Another terminological clarification we make (following the recommendations of (Hessels et al., 2018)) concerns the particular eye movement definitions we used for this work. We note that in our data, the head of the observer was always fixed, so when we talk about motion, we mean movement on the monitor, which necessarily implies movement relative to the observer in this set-up. Also, the eye tracker yielded point-of-regard coordinates relative to the monitor (i.e., in the world coordinate system). In this setting, we limited ourselves to four labels: fixations, saccades, smooth pursuits, and noise. For convenience of terminology, we refer to fixations as "eye movements" as well, even though they are technically defined by the absence of motion ("gaze event" might be a more accurate, but less common term).

The following definitions were employed: (a) Fixations were defined as periods of relatively stationary gaze, which was not following the motion of any moving object in the video. (b) Saccades were defined as jumps to different on-screen positions, and no specific amplitude bounds were utilized. The end of each saccade was marked when the gaze had stabilized again. Even though there is no clear definition for postsaccadic oscillations (PSOs; I. Hooge, Nyström, Cornelissen, & Holmqvist, 2015), our saccade end interpretation considers them part of respective saccades. If a different way of handling the saccade and PSOs combination is desired, additional analyses have to be carried out. (c) Special care was given to SP labelling since it can be confused with other pursuit-like motions. SP labels were assigned to the parts of the gaze recordings where the gaze point was smoothly moving itself and was following a moving object in the video, i.e., the projection of the point of regard had roughly the same velocity—speed and direction of motion—as some moving object. The spatial location of the gaze also had to approximately match that of the assumed target (some offset was allowed to account for the potential drifts in tracking). Contrarily, if the gaze was moving, even in a pursuit-like fashion, without a corresponding target, it was considered part of a drifting or noisy fixation. We observed several instances in the data where the gaze recording was smoothly moving in a direction perpendicular or even opposite to the velocity of the closest potential target. (d) Blinks, gaze reported outside of the monitor, as well as intervals where the eye tracker was yielding zero confidence, along with naturally impossible gaze traces, which could be attributed to tracking artefacts, were labelled as noise. In this work, "noise" is used to name the parts of the gaze recordings that are irrelevant to the present study, and a more precise labelling scheme might be required for different-context studies. This is why this label was also assigned to blinks, for example, even though these are a dedicated type of eye activity.

Additionally, we use the terms "event" and "episode" interchangeably when talking about eye movements, both referring to a period of time where all the gaze sample class labels (either in human annotations or in the output of an algorithmic detector) are identical. Thus, any gaze recording is subdivided into nonoverlapping eye movement events (episodes), each described by a corresponding label (in this study—one of the labels defined above).

We further note that we refer to the manual labels as the "ground truth" for eye movement classification, even though expert annotations differ between themselves (I. T. C. Hooge et al., 2018), and even such basic eye movements as fixations and saccades are differently defined in the field (Hessels et al., 2018). Therefore, the labels produced by hand-labelling the eye tracking data can only be an approximation of the eye movements that were taking place at the time. Nevertheless, we maintain the "ground truth" name for this type of data as this represents the state-of-the-art data source in eye movement classification (Zemblys et al., 2018; Startsev, Agtzidis, & Dorr, 2019; Zemblys et al., 2019), though some automatic scoring pipelines are also being developed (Larsson, Nyström, Ardö, Åström, & Stridh, 2016).

Original data set

Because the GazeCom (Dorr et al., 2010) data set forms the basis on which we build our work, we briefly describe its set-up and basic statistics here. The data set comprises 18 short naturalistic video clips (20 s each), depicting everyday scenes. These include beach scenes, pedestrian and car-filled streets, boats, animals, etc. There is little to no camera motion in the recorded clips (11 out of 18 clips lack it completely, four have slow panning camera motion, and the camera was slightly

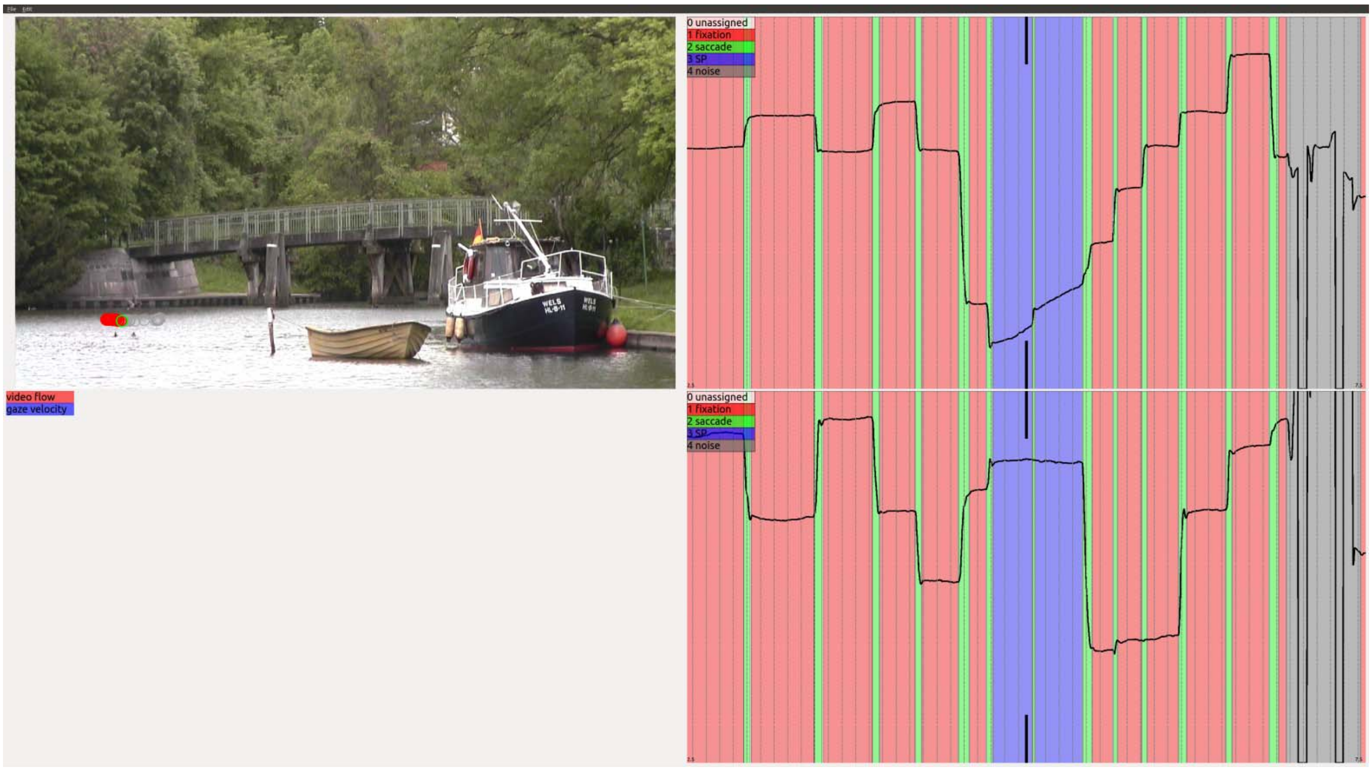


Figure 2. An example of the hand-labelling tool interface.

shaking in the other three), and the scenes themselves contain both rigid (e.g., cars) and nonrigid (e.g., human or animal) motion at a variety of speeds. These clips thereby form a set of dynamic and relatively naturalistic stimuli.

All video clips were presented at 1280 × 720 pixels, 29.97 frames per second, at a distance of 45 cm from the observers. The frames covered an area of 48 × 27 degrees of visual angle. The gaze of 54 participants was recorded at 250 Hz with an SR Research EyeLink II eye tracker. Even though the eye tracker allowed for small head motion, a chin rest was used to stabilize the participants' heads. Some recordings were discarded by the authors of the data set due to frequent (over 5%) tracking loss, leaving 844 recordings in the published data set (46.9 per clip on average). These data total 4.5 hr of gaze tracking recordings, all of which we annotate and analyze in the context of this work.

Manual eye movement annotation

We now focus on the manual annotation part of our work, for which we used the software described in (Agtzidis, Startsev, & Dorr, 2016a). The graphical interface presents an annotator with four panels (see Figure 2). The top left panel displays the video overlaid with the gaze trace (current gaze sample plus gaze positions 100 ms before and after it). The bottom left panel, which was not used during our labelling, is optional and displays the optical flow of the video. The two panels on the right display the *x* and the *y* gaze coordinates as time series, which are overlaid with color-coded boxes that correspond to the time intervals of different eye movements. These intervals could be freely created or deleted, and their borders could be freely adjusted by the manual annotators, who could also scroll through the video (to observe object motion patterns) and change the temporal scale of the displayed gaze coordinates.

Prior to the hand-labelling process, the eye movements were roughly prelabelled automatically with the purpose of simplifying the annotation process (e.g., so that the manual raters would not have to insert and label as many eye movement episodes, mostly adjusting their borders). For prelabelling we used the authors' implementation of the saccade and fixation detection algorithms of Dorr et al. (2010). The rest of the samples were clustered in order to detect SP gaze samples by a very early implementation of the Agtzidis et al. (2016b) algorithm.

This technique of prelabelling the samples prior to manual annotation allowed us to roughly double the speed of the labelling process: For an expert annotator, the labelling time decreased from ca. 10 to ca. 4 min on average per single ca. 20 s recording (Agtzidis et al., 2016a). The importance of these gains becomes evident when we consider the 4.5 hr of gaze recordings of the GazeCom data set labelled by several annotators, thus saving months of manual annotation time. Even though any form of prelabelling introduces bias into the resulting labels, we note the following: (a) Most of the algorithms for eye movement detection, even the simple threshold-based ones, detect fixations and saccades reasonably well (Startsev, Agtzidis, & Dorr, 2019). Therefore, potential bias in the manual labels should not constitute a large issue. (b) For smooth pursuit, however, which is the focus point of this work, and which is harder to detect algorithmically, we specifically tested that our conclusions about the performance of the SP detector we developed were not unfairly affected by our labelling procedure (see the Validity check for algorithmic detection evaluation section).

The interface described above was used by three human annotators in order to create a complete manually labelled version of the GazeCom data set in accordance to the eye movement definitions that were given in the Addressing terminological ambiguity section. The overall process involved two novice annotators going through all the recordings twice, followed by an expert who solved conflicts in their annotations, but was still free to make any adjustments in the labels in accordance with the provided eye movement definitions.

The two novice annotators were paid undergraduate students who received basic instructions about eye movements and interpreting eye tracker data. Experts in the eye movement field were available to answer their questions at any point in the labelling process. Due to their little prior experience with hand-labelling and because we wanted their internal biases to stabilize, these two annotators went through the data set for a second time several months later. In the first pass they were provided with the prelabelled suggestions and instructed to change, add, or remove intervals accordingly. In the second pass they were presented with their own labelling and instructed to change it wherever they thought it was not accurate (with respect to the eye movement definitions). As a quality assurance measure, a third (expert) annotator (one of the authors) reexamined all the recordings in the data set with the objective of resolving conflicts between the labels of the first two annotators, also making changes where the provided eye movement definitions were violated. We report on the agreement between the raters later in the paper.

In order to describe the eye movements in our data set, we report several simple statistics. First, we computed the overall speed of the events of each eye movement class as episode amplitude divided by its duration. Similarly, to characterize the directional similarity of gaze movement within the individual eye movement episodes, we computed the angular deviation of sample-to-sample velocity vectors from the overall direction of the corresponding episode. The overall direction was computed as the vector pointing from the start to the end position of gaze for each eye movement episode. The deviations are then computed as angles between the sample-to-sample shift vectors and the respective overall direction vector. Such vectors are visualized in Figure 3 for an example fixation of GazeCom data.

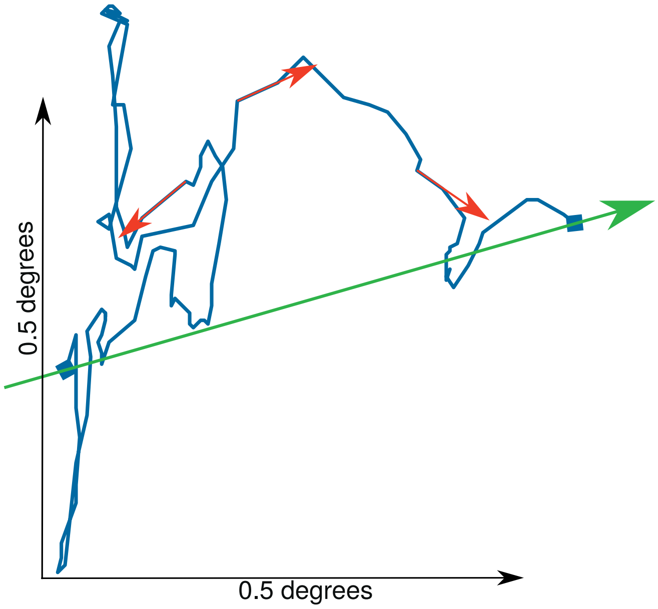


Figure 3. The sequence of gaze samples for an example fixation, with the green vector marking the overall direction of the episode and the red vectors corresponding to examples of sample-to-sample gaze shift directions. The axes' arrows indicate the scale of the plot.

To additionally quantify gaze behavior in naturalistic dynamic video viewing, we also directly assessed how synchronous were the eye movements (of the same type) of different observers. To achieve this, we computed the following for each of the eye movement types considered here: (a) For each data point, we determined the other data points belonging to its spatio-temporal neighborhood (determined by the parameters of the observer-driven clustering modification of our approach, see Appendix, Observer-driven clustering extension of DBSCAN—within 4° in the monitor space and within 20 ms in time). (b) Among these points, we computed the number of unique other observers. (c) We then measured the percentage of gaze samples (i.e., data points) that had no fewer than N other observers' gaze samples (of the same eye movement type) in their neighborhood, and plotted this over varying N (0 to 40 with a step of 1).

Automatic eye movement annotation

Manually labelling eye movements is a tedious process that requires a substantial amount of time, an order of magnitude greater than the time required to perform the recordings. Automating this process can be desirable, as long as the algorithmically produced labels offer qualitatively similar results to the manual ones. The algorithm of Agtzidis et al. (2016b) forms the basis for our automatic eye movement annotation approach. Here we provide an in-detail description of the algorithm and its implementation, which was developed in the context of this work. We further optimized the parameters of our approach (see Appendix, Parameter optimization), which has significantly improved the algorithm's performance (the values of the optimized parameters are provided below). For recommendations regarding parameter adjustment when the algorithm is to be applied to a different data set, see Appendix, Parameter adaptation for other data sets.

Our approach first removes the confidently detected saccades (along with blinks) and fixations from consideration. Saccades were detected by the dual-threshold saccade detector of Dorr et al. (2010). Saccades nearest to the tracking loss intervals (but no further than 25 ms) were marked as parts of a blink. Fixations were removed based on sliding-window analysis: All intersaccadic intervals with a gaze shift magnitude below 1.41° were first marked as fixations (value chosen via parameter grid search, see Appendix, Parameter optimization). A 100 ms sliding window was then applied to the remaining intervals to detect fixation on- and off-sets when the average gaze speed in the considered window fell below or raised above 2°/s, respectively.

After the prefiltering step, we clustered the remaining "pursuit candidate" samples with a variation of the DBSCAN clustering algorithm (Ester, Kriegel, Sander, & Xu, 1996). Importantly, the recordings of individual observers were processed *separately* for saccade, blink, and fixation detection, but the remaining SP candidate samples were *aggregated* from all the available recordings for a given stimulus (between 37 and 52 in GazeCom).

We employed DBSCAN in the 3D space consisting of time and x, y coordinates. This algorithm effectively finds densely populated areas of the considered space by subdividing all the data samples into (a) cluster core points, (b) border points, and (c) outliers. The concept of the point's neighborhood is important for these definitions, and it is usually defined as all the data points with a distance from the considered point not exceeding a user-set value (parameter ε). The core points are defined as those having at least a certain number (parameter *minPts*) of points in their respective neighborhoods. Border points are those that do not fulfil the requirements for core points but have at least one core point in their neighborhood. All other data samples are labelled as outliers (not a part of any cluster) and receive a "noise" eye movement class label.

As there is no universal way of scaling distances in time and in space, we proposed a slight modification of the original DBSCAN algorithm by splitting coordinates into groups that are considered together, and for which an independently set threshold is used. For our data, we grouped x and y and used the threshold $\varepsilon_{xy} = 4°$ of visual angle. Time t represented the other coordinate group, with the threshold $\varepsilon_t = 80$ ms. The *minPts* parameter was set to 160 following the optimization procedure in Appendix, Parameter optimization.

An important distinction of DBSCAN from many other popular clustering algorithms (e.g., k-means; MacQueen, 1967, or Gaussian mixture models) is that it does not assume that clusters can be represented by centroids, but the cluster shape is arbitrary and only determined by the data point density in the respective space. This is particularly important for detecting the grouping of smooth pursuit samples, as the trajectory of the pursued target can be arbitrary, and the dynamic nature of pursuit does not allow for its representation as a centroid, which could be appropriate for fixations, for example. Our implementation of DBSCAN not only labels all the considered data points as either belonging to a cluster or not, but also differentiates between the individual clusters by assigning a corresponding (unique) cluster ID to all the gaze samples belonging to a particular cluster.

We also note that we additionally implemented a more elegant, albeit less performant, version of the algorithm, which clusters the data based on how many unique *observers* have produced gaze samples in the spatio-temporal vicinity of the considered gaze point, instead of simply using the number of gaze samples themselves. We describe this algorithm variant and some analysis of its performance in more detail in Appendix, Observer-driven clustering extension of DBSCAN.

Programmatic interface

The implementation of our algorithm together with a wide set of evaluation measures for eye movement classification in general is available at https://web.gin.g-node.org/ioannis.agtzidis/gazecom_annotations/ (accompanying the annotated GazeCom data set) or as GitHub repository https://github.com/MikhailStartsev/sp_tool. The implementation uses Python and several external libraries (e.g., for handling ARFF data), which are listed as its dependencies. We will here briefly cover the functionality of the published framework.

The framework can be used either as a Python library that can be accessed from code or through an executable file. In both cases, the framework user will interact with the *run_detection.py* file and can set all the parameters related to saccade, blink, fixation, and pursuit detection as well as specify the path to input and output directories. Implementation details of all the detectors can be found in respective source files (e.g., *saccade_detector.py,* etc.). The parameter set that we recommend based on the results of the optimization procedure in Appendix, Parameter optimization is provided in the *default_parameters.conf.json* file, which can be modified with any text editor, if necessary.

Two data formats can be loaded natively (without preliminary conversion): ARFF (as described in Appendix, Data format) and the original format of the GazeCom data set (Dorr et al., 2010), also text-based, with a header describing experiment set-up parameters. We additionally provide conversion scripts for two popular eye-tracking recording formats: text files produced from binary SMI recording files and EyeLink ASCII format (usually *.asc* files). These conversion scripts also provide an example for programmatically populating an ARFF file structure with any data and can be found in the *examples/* directory of the source code.

Beyond this functionality, the framework provides an implementation of a diverse set of metrics (see next section), which can be computed for any ARFF data (i.e., not necessarily GazeCom, not necessarily only the eye movement types that are present in our data), provided that some form of corresponding "ground truth" and tested eye movement labels are available. The implementation of the evaluation strategies can be found in *evaluate.py,* and the evaluation script—*examples/run_evaluation.py*—can be executed directly from the command line.

Sample- and event-level evaluation

The widely used evaluation measures we implemented include sample-level accuracy/precision/recall/*F*1 scores (we recommend using *F*1 as a balanced combination of precision and recall) and Cohen's kappa. Levenshtein distances between the true and the predicted labelled sequences (of either samples or events), as proposed by Zemblys et al. (2019), evaluate the edit distances between the two sequences, though these are a relatively weak evaluation measure that might not be well suited for the eye movement classification problem (Startsev, Göb, & Dorr, 2019).

As for event-level evaluation, there is no consensus in the literature as to which measures should be used. We therefore tested several different strategies proposed in the field. We particularly want to point out the *F*1 scores as computed by I. T. C. Hooge et al. (2018), where the intersecting same-class episodes are matched. It was modified in recent works: In Zemblys et al. (2019), the events that have the largest intersection are matched (rather than the temporally first intersecting event being treated as a match, as in the original matching scheme of I. T. C. Hooge et al., 2018), and the event-level Cohen's kappa scores are computed accordingly. In Startsev, Agtzidis, and Dorr (2019), a threshold for the "quality" of the intersection was recommended, which results in no more than one potential match for each of the "true" episodes. In Startsev, Göb, and Dorr, (2019) we additionally proposed a new event-level Cohen's kappa-based statistic, which we developed after analyzing the literature evaluation strategies in the context of eye movement classification baselines. These and other evaluation methods can be found as functions of the framework we provide.

In this manuscript we will mostly rely on sample-level *F*1 scores and event-level *F*1 scores of (I. T. C. Hooge et al., 2018) for simplicity. A larger spectrum of metrics for this and other literature models is reported on the data repository page, however.

Algorithm evaluation

To put the performance of our detector in context, we compare it with three other methods that detect SP: the algorithms of Berg et al. (2009, implemented in Walther & Koch, 2006) and Larsson et al. (2015, reimplemented by our group and available for download on the data repository page), as well as I-VMP (San Agustin, 2010, implemented by Komogortsev, 2014). I-VMP, among others, was optimized in Startsev, Agtzidis, and Dorr (2019) via an exhaustive grid search of its parameters in order to deliver optimal performance on the full GazeCom data set, so its results represent an optimistic scenario. These three models (plus the approach described here) were the best nondeep-learning detectors tested in Startsev, Agtzidis, and Dorr (2019), when ranked by the average per-class sample- and event-level *F*1 scores. We use the same metrics in this paper and test all models on the full set of annotations of the GazeCom recordings that are collected as described in this work.

Beside sample- and event-level *F*1 scores, we wanted to computationally directly assess the properties of the episodes (as detected by all the algorithms) and how they compare to those of the ground truth episodes. We consider duration as an example of a widely used episode characteristic. As researchers might, for example, use SP episode durations to distinguish between

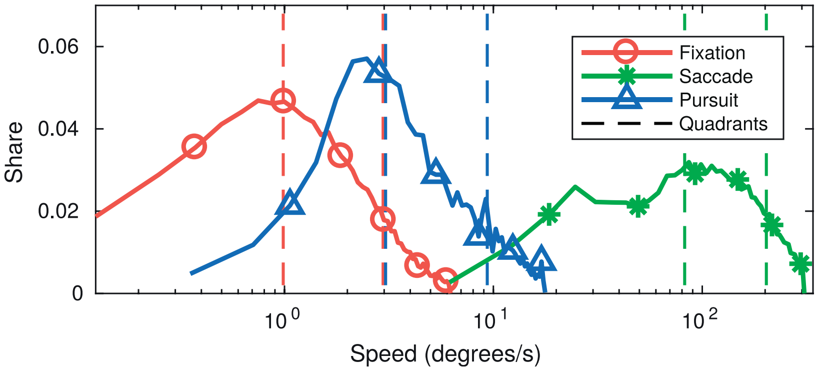


Figure 4. Overall per-episode speed distributions for fixations, saccades, and smooth pursuits. These are the (normalized) histograms, which were computed for each eye movement type independently with 50 equal-sized bins covering each respective speed range. These were then plotted here in log-scale (see *x* axis), with the *y* axis representing the share of episodes in each of the bins. The dashed vertical lines visualize the quartiles (first and third) of the respective distributions. Note that since the horizontal axis is in log-scale, it is difficult to visually compare the areas under different parts of the curves. For example, for fixations (red solid line), 50% of the labelled episodes (between the first and third quartile lines) had an overall speed between 1°/s and 3°/s, as indicated by the left and right vertical red lines, respectively.

clinical populations (Silberg et al., 2019), it would be useful to know which algorithms should be used for automatic event detection in order to obtain episodes that are closer to the ground truth in terms of the properties of interest.

Instead of comparing just average episode statistics (e.g., as in Komogortsev, Jayarathna, Koh, & Gowda, 2010), we represent episode duration distributions as histograms (of 256 bins) and evaluate their similarity with appropriate measures: Kullback–Leibler divergence (KLD; Joyce, 2011) and histogram intersection similarity (HSIM; Swain & Ballard, 1991).

Results

Eye movement properties

Overall, the GazeCom data set (in our final annotation) contains 38,629 fixations, 39,217 saccades, and 4,631 SP episodes. While the number of SP episodes may seem small, especially for training a balanced classification algorithm, there are more pursuit than saccade samples: 11% versus 10.5%. As expected, most samples were labelled as fixations (72.5%), with another ca. 6% labelled as "noise."

In this section, we visualize some basic and commonly used (e.g., Salvucci & Goldberg, 2000; Komogortsev & Karpov, 2013; Santini et al., 2016; Zemblys et al., 2018; Startsev, Agtzidis, & Dorr, 2019) statistics (speed and directional deviation) of the ground-truth fixations, saccades, and pursuits.

Figure 4 visualizes the distribution of the overall speeds of the events of each eye movement class. Notably, some average saccade speeds were lower than expected because of the inclusion of PSOs in our definition. Whereas fixations and thus-labelled saccades have almost no intersection in their speed distributions, pursuits demonstrate a sizeable overlap with the fixation class, while also extending into the territory of the speeds of slow saccades.

Figure 5 visualizes the distributions of sample-to-sample velocity vector angular deviation from the overall direction of the corresponding episode. We can observe that the three eye movement types we consider correspond to three distinct shapes of the direction deviation distribution, with saccades having the most pronounced peak (Figure 5c), followed by SPs (Figure 5b), followed by an almost uniform distribution for fixations (Figure 5a). The direction deviation distribution for fixations is not perfectly uniform because the deviations of direction are computed regardless of the gaze shift magnitude (e.g., see Figure 3), and thus any drift, however small, would result in the distribution skewing. The fact that these distributions exhibit different patterns for fixations, saccades, and pursuits indicates that gaze movement direction could be a useful feature for eye movement classification (which was also demonstrated in Larsson et al. (2016) and Startsev, Agtzidis, and Dorr (2019).

Figure 6 depicts the spatio-temporal interobserver congruency of different eye movement types, demonstrating that pursuit has the strongest synchrony between the observers, closely followed by fixations, followed by saccades, finally followed by samples labelled as noise.

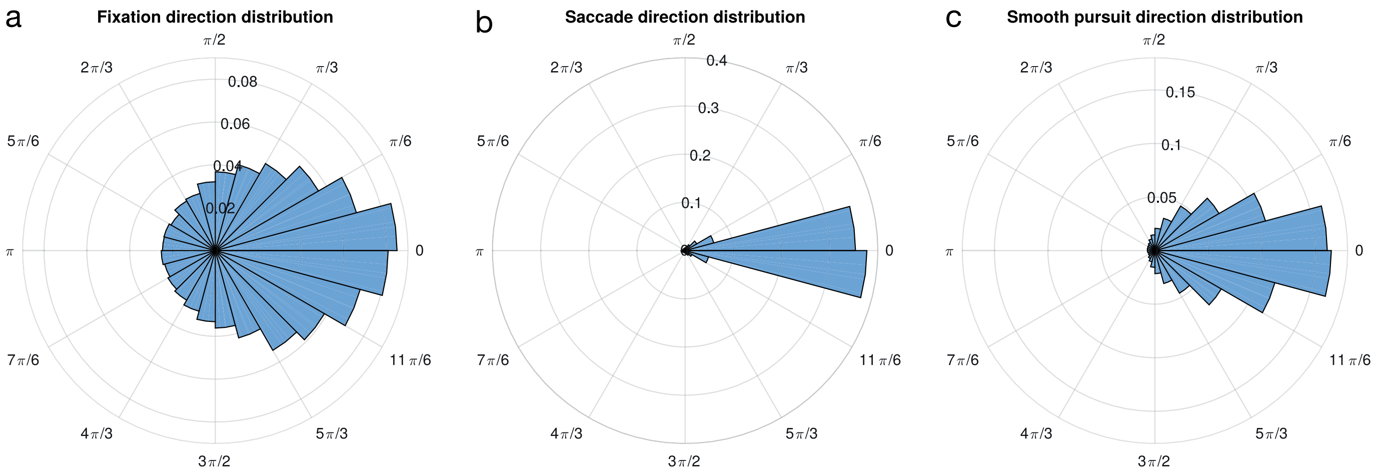


Figure 5. Directional deviation distributions for fixations (a), pursuits (b), and saccades (c), presented as circular histograms. The height of each bar represents the share of the sample-to-sample velocity vectors with the given angular deviation from the overall direction of their corresponding episode (see Figure 3). Zero deviation angle means perfect alignment with the overall direction of the respective episode.

Hand-labelling statistics

Labelling the full Gazecom data set lasted the equivalent of several months of full-time work (including the two passes through the whole data set for the first two annotators). On average for all three annotators, labelling one GazeCom recording (usually ca. 20 s) took between 5 and 6 minutes, which is equivalent to a labelling time of 15–18 s for each second of the recorded gaze signal. The labelling process also benefited from prelabeling the gaze signal, which more than doubled the labelling speed (see the Manual eye movement annotation section).

In Figure 7 we illustrate the confusion matrix between the prelabelled and hand-labelled eye movement classes, thus reporting which and how many algorithmically preassigned labels were replaced during manual annotation. The algorithmically suggested

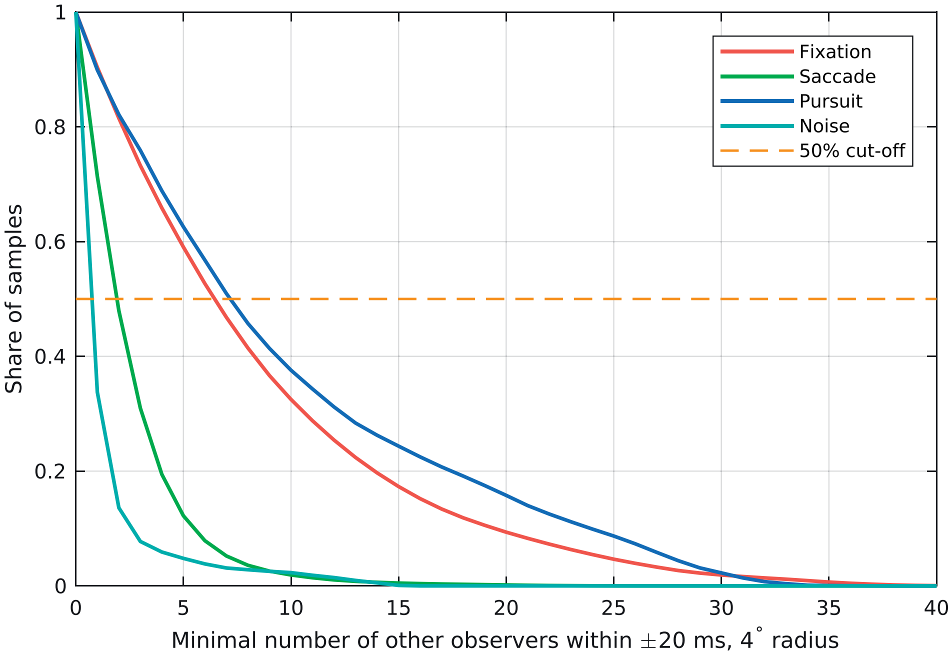


Figure 6. Visualization for the spatio-temporal congruency between same-type eye movements *of different observers*. The *y* axis portrays the share of the respective eye movement samples that are located within 20 ms and a 4° radius from the same-type samples that belong to at least as many different unique observers as denoted by the *x* axis.

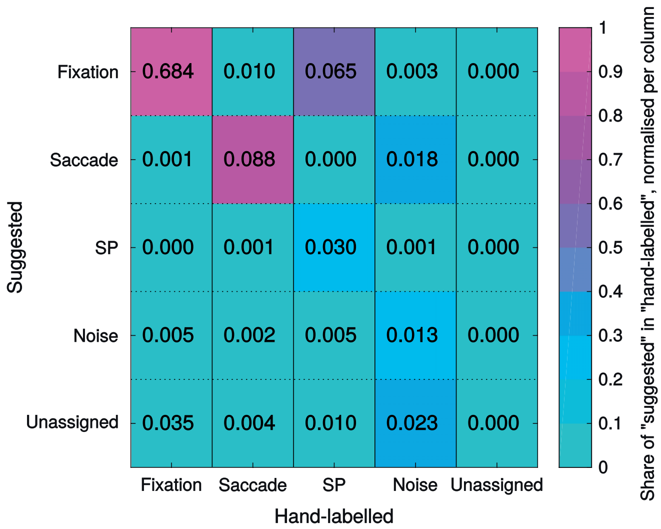


Figure 7. Confusion matrix for the prelabelled and manually annotated eye movement samples. Rows correspond to the suggested eye movement labels, columns—to the final hand-labelled classes. Cell color reflects the share of samples in the final hand-labelling that were originally prelabelled as the respective suggested classes (i.e., per-column normalization is employed; cf. the color bar on the right).

labels are represented by the matrix rows, while the final "ground truth" labels are represented by the columns. Note that the individual cells contain the overall share of the samples that had a certain suggested label and a certain final label, i.e., the whole matrix sums to 1.0, but not the individual rows or columns. The color of the cells indicates the degree of the correspondence between the originally suggested labels of each type and the final labels of each type (see color bar in Figure 7; if the prelabeling were perfect, only the diagonal would be populated). It can be observed that fixations and saccades were very well detected by the algorithms (over 90% of the final labels of these types were correctly labelled by the algorithms that were used for prelabeling). For pursuit, however, most of the finally assigned SP labels corresponded to originally suggested fixation labels (ca. 59%), only 27% being prelabelled correctly. A large share of the final noise labels (ca. 31%) correspond to prelabelled saccades, with half of them very likely being a part of blinks (closer than 200 ms to a tracking loss interval), which is common in video-oculography (Holmqvist et al., 2011, Section 5.7).

Figure 7 already reflects the proportions of samples that were prelabelled or received a manual label of a certain type (these numbers can be obtained by summing either the matrix rows or columns, respectively). We also separately report the label shares and the number of respective uninterrupted episodes in Table 1. It can be seen again that the amount of SP has increased dramatically with the manual annotation (from 3% to 11% of gaze samples, ca. 3000 to ca. 4500 episodes), whereas the amount of saccades and fixations (in terms of both samples and episodes) was prelabelled relatively accurately. This is indicative of both fixation and saccade classes being more well defined in the literature and the existing (even simple) detectors being much more accurate for these classes. Overall, we can say that the preassigned labels were changed substantially during manual annotation, mostly affecting the smooth pursuit class.

Eye movement type	Suggested label		Final expert label	
	Share	Episodes	Share	Episodes
Fixation	76.2%	39,293	72.6%	38,629
Saccade	10.7%	40,233	10.5%	39,217
SP	3.3%	2879	11%	4631
Noise	2.5%	6319	5.9%	3493
Unassigned	7.3%	27,165	0%	0

Table 1. The overall percentage of gaze samples and number of episodes of all eye movement types in the algorithmically suggested ("prelabelled") labels and the final set of labels produced in our annotation procedure.

Interrater agreement

We here report how well the three annotators agreed in their labels in terms of sample-level *F*1 scores; event-level scores were quantitatively similar because humans tend not to fragment intervals (data not shown). The scores presented in Table 2 indicate that all the annotator pairs have very high agreement levels for fixations and saccades. For pursuits, however, the agreement is substantially lower and the final annotator, who was mostly resolving the conflicts between the labels of the first two annotators, tended to mostly agree with the labelling of the first annotator. Interestingly, the agreement scores between each annotator's first and second pass labels (marked with $_{ini}$ and $_{final}$ in the table) are similar in value to the interrater agreement, confirming the difficulty of

Eye movement type	1_{ini} vs. 1_{final}	2_{ini} vs. 2_{final}	1_{final} vs. 2_{final}	1_{final} vs. final	2_{final} vs. final
Fixation	0.950	0.977	0.933	0.975	0.949
Saccade	0.904	0.951	0.863	0.937	0.883
SP	0.787	0.796	0.629	0.904	0.697

Table 2. Agreement between the initial (1_{ini} and 2_{ini}) and final (1_{final} and 2_{final}) annotations of the two nonexpert annotators, and all annotator pairs in the form of sample-level F1 scores. The "final" label refers to the annotations of the third (expert) rater, who consolidated the labels of 1_{final} and 2_{final}.

Eye movement type	sp_tool vs. 1	sp_tool vs. 2	Sp_tool vs. final
Fixation	0.883	0.882	0.886
Saccade	0.849	0.883	0.864
SP	0.626	0.602	0.646

Table 3. Agreement between our algorithmic eye movement detection framework and all of the annotators in the form of sample-level *F*1 scores.

pursuit annotation in naturalistic stimuli, compared to the labelling of fixations and saccades.

We will examine algorithmic detection in more detail in the next section, but we report the same type of agreement scores for our algorithm and all of the individual annotators in Table 3. As our detector was optimized for the final manual label, its own SP detection outputs agree more with the final annotator, but the differences are small. Generally, the agreement of our algorithm with the manual raters is close to the agreement between the raters themselves.

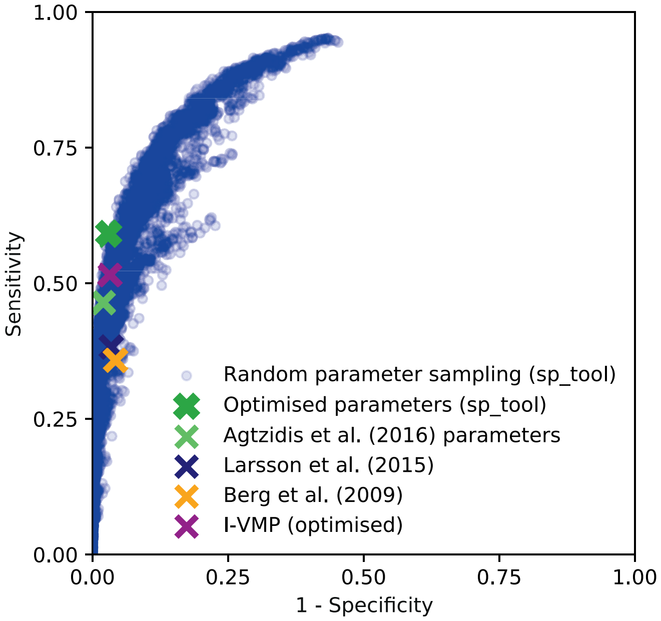


Figure 8. Smooth pursuit detection performance range of our framework, depending on the parameters.

Algorithmic detector parameter optimization results

We randomly sampled the multidimensional parameter space of our fixation and pursuit detectors (see Appendix, Parameter optimization), which enabled us to illustrate the performance range of our detector in the form of a ROC-like plot in Figure 8. The optimization procedure has substantially increased the sensitivity of the sp_tool – from 0.46 for the preliminary parameter set in (Agtzidis et al., 2016b) to 0.59 after optimization—at the cost of minimally lowered specificity (0.98 to 0.97). The optimization criteria did not account for fixation detection quality. However, this improvement in SP detection also comes with an increase in the event-level *F*1 score for fixation detection—0.75 for Agtzidis et al. (2016b) versus 0.81 for the sp_tool after parameter optimization—at a small decrease of sample-level *F*1 (0.91 to 0.89).

Quantitative evaluation

In this section we report and discuss the various performance statistics for our sp_tool detector in comparison to the other methods in the literature, which include the preliminary version of the multi-observer SP detector (Agtzidis et al., 2016b) and the algorithms of Berg et al. (2009), San Agustin (2010), and Larsson et al. (2015). Our comparison is based on several metrics: First of all, the sample- and event-level *F*1 scores were computed. Then, we numerically compared the distributions of automatically detected SP episodes with those in the ground truth via KLD and HSIM (see the Algorithm evaluation section). For *F*1 scores and HSIM, higher values are better, with a perfect algorithm scoring 1. For KLD, lower values are better (as it is a measure of divergence), with the best score of 0.

SP detection performance is separately addressed in Table 4. From these statistics it can be seen that parameter optimization positively affects both the *F*1 scores and the distributional metrics, more than halving the KLD and increasing the HSIM score over 1.5 times, compared to the Agtzidis et al. (2016b) version of the algorithm. Overall, the biggest weakness of the Agtzidis et al. (2016b) parameter set for the sp_tool lies in generating a large number of short SP episodes, which is reflected by the KLD and HSIM measures, ranking it

Algorithm	Sample *F*1 ↑	Event *F*1 ↑	Duration distr. KLD ↓	Duration distr. HSIM ↑
Ours (sp_tool): optimized	**0.646**	**0.527**	**0.620**	**0.679**
Larsson et al. (2015)	0.459	0.392	0.693	0.647
I-VMP (optimized)	0.581	**0.531**	1.154	0.602
Agtzidis et al. (2016b)	0.571	0.415	1.280	0.440
Berg et al. (2009)	0.422	0.424	1.923	0.459

Table 4. Smooth pursuit detection evaluation results on the entire GazeCom data set. *Notes*: The ↑ symbol marks the columns where the higher score is better; ↓ where the lower score is better. The rows are sorted by their average scores (KLD taken with a negative sign). Best score in each column (or within 0.01 of it) is bolded.

Algorithm	SP sample $F1$	SP event $F1$
Ours (sp_tool): optimized	**0.423**	**0.419**
I-VMP (optimized)	0.382	0.399
Berg et al. (2009)	0.240	0.316
Larsson et al. (2015)	0.207	0.239

Table 5. Partial evaluation results (only on the labels that were *changed* during the annotation), demonstrating that our labelling procedure does not unfairly favor our model. *Notes*: The rows are sorted by their average scores. Highest score in each column is bolded.

on average below (Larsson et al., 2015) and I-VMP, even though its $F1$ scores are mostly higher or on par with these models. Parameter optimization led to a significant performance increase that puts our framework higher than the competition, yielding the best results in all considered metrics except event-level $F1$ scores, where the score is slightly behind the optimized I-VMP, but only by 0.004.

The sp_tool framework also detects fixations and saccades as part of its pipeline, and we compared the algorithms employed there to the same literature models as in Table 5 for SP (for full evaluation tables, see Startsev, Agtzidis, & Dorr, 2019). For saccade detection, sp_tool and our reimplementation of Larsson et al. (2015) use the same saccade detector (Dorr et al., 2010), which yields better sample- and event-level $F1$ scores than the next best model for saccade detection in our evaluation (Berg et al., 2009): 0.86 and 0.88 versus 0.70 and 0.86, respectively. In terms of fixation detection, the sp_tool performance (0.89 and 0.81 for sample- and event-level $F1$ scores) is comparable, though slightly behind the Larsson et al. (2015) model with its scores of 0.91 and 0.87, respectively. These results indicate that the sp_tool offers an improvement to SP detection without sacrificing fixation and saccade detection performance, thus offering a balanced framework for eye movement classification.

Validity check for algorithmic detection evaluation

Here we address the issue that was raised in the Manual eye movement annotation section: Since a pilot implementation of the clustering strategy described in this work was used to algorithmically prelabel SP prior to manual annotation (to speed up the tedious process), it is possible that the potential correlation of the final labels with the algorithmically suggested labels would unfairly benefit our model's evaluation scores. We therefore tested our (postoptimization, see Appendix, Parameter optimization) and literature SP detectors on those gaze sample where the label was *changed* by the manual raters during the annotation process.

Overall, the final manual annotator "disagreed" with the algorithmically suggested labels in 18.5% of the cases. This seems low, but this encompasses 72.9% of the final SP labels, so the partial evaluation for this class is meaningful. Table 5 presents the sample- and event-level $F1$ scores for all the tested detectors on these data. It can be seen that even in these conditions our model outperforms the literature models by a noticeable margin.

It has to be additionally noted that all the results reported in this table are noticeably lower than the corresponding values in Table 4 (for the full GazeCom data set): Sample-level $F1$ scores in Table 5 are ca. 0.2 lower than on the full data set, event-level scores—between 0.1 and 0.2 lower. This leads us to argue that the SP episodes that were correctly prelabelled prior to manual annotation represent a set of easily detectable examples for any pursuit detector, so their preannotation would not bias the evaluation in favor of our approach.

Robustness to variations in the number of observers

As the approach we take to SP detection is based on analyzing the recordings of several observers at once, we tested how much its performance depends on the number of the observers whose gaze recordings are available for processing.

To be able to compare the performances of our model on the subsets of GazeCom with reduced numbers of observers, as well as to alleviate the effects of the random subsampling, we repeatedly sampled reduced observer sets for each stimulus video clip independently. We tested the subsets that included between 5 and 45 observers and sampled (without replacement) the respective number of recordings 20 times for each video. If the video had fewer recordings than required, all of the available recordings were used without duplication.

Figure 9 presents the sample- and event-level $F1$ scores for SP detection achieved by our algorithm (parameters optimized for the full GazeCom set and adjusted according to the recommendations in Appendix, Parameter adaptation for other data sets, i.e., *minPts* scaled proportionally to the number of observers) and compares those to the results of I-VMP—the literature model with the best respective scores (see Table 4).

It can be observed that sample-level performance of our model confidently exceeds that of I-VMP when 15 or more observers' recordings are processed at once, and keeps increasing. Event-level $F1$ scores for our

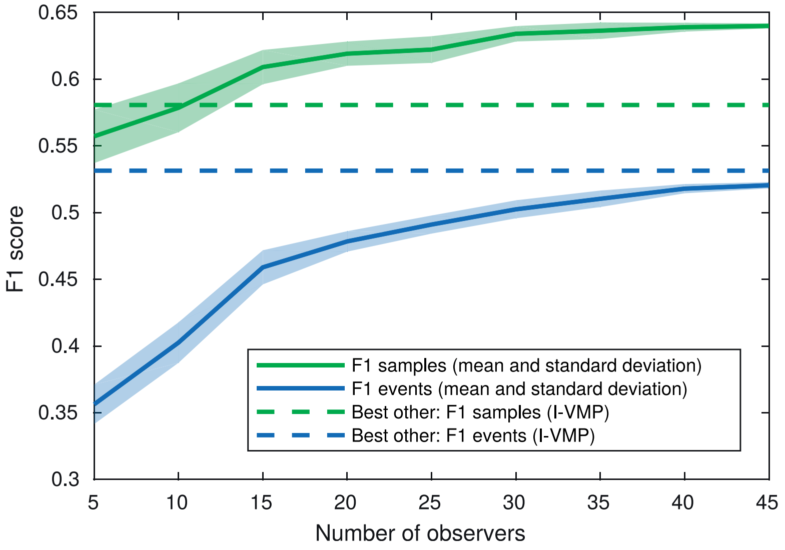


Figure 9. The dynamics of the sample- and event-level *F*1 scores of the sp_tool pursuit detection depending on the number of observers that are used for analysis simultaneously. Dashed lines indicate the scores achieved by the best other model (see Table 4). The shaded areas correspond to ±1 *SD* of the scores over 20 runs.

approach also increase with the number of observers, but only reach performance levels comparable with I-VMP when ca. 40 observers have viewed each clip.

We note that the observed dynamics in the (sample-level) *F*1 scores were due to precision rapidly increasing with the number of observers (from 0.47 to 0.7 for five and 45 observers, respectively), while recall gradually decreased (from 0.69 to 0.59). On the whole, the increase in sample-level *F*1 scores becomes incremental around the 15-observer mark. For event-level scores the same is observed only at around 30–35 recordings per stimulus.

Discussion

In this work we presented, first of all, the manual eye movement annotations for the GazeCom data set (Dorr et al., 2010). These represent, to the best of our knowledge, the largest collection of expert eye movement class labels where smooth pursuit is taken into account. Other dynamic content viewing data sets that are manually annotated are typically either small in size (Andersson et al., 2017), or focus on synthetic stimuli viewing (Santini et al., 2016). A recent work by Steil et al. (2018) only annotates the data for determining whether the gaze keeps following the same object between recording frames, which does not differentiate between fixations and pursuits, thus confounding static and dynamic gaze behaviors in its definition of "fixation." The data set presented in Agtzidis, Startsev, and Dorr (2019) annotates smooth pursuit in 360° video viewing as well, but it is much smaller in size (ca. 0.5 h). Kurzhals, Bopp, Bässler, Ebinger, and Weiskopf (2014) manually annotated only the areas of interest and not the eye movements themselves (fixations detected by a standard algorithm are also provided). The data set presented in this work will allow researchers to acquire insights into certain aspects of behavior during naturalistic video viewing, where differentiating between fixations and pursuits is of importance.

Eye movement behavior in dynamic natural scenes

Our work provides the first quantitative characterization of human pursuit behavior in dynamic natural scenes. Given the significance of this eye movement type, we argue that researchers should take smooth pursuit into account when analyzing gaze recordings for dynamic stimuli. In our experiments ca. 11% of the viewing time was spent performing smooth pursuit, which is more than the time spent during saccades. This is particularly impressive as the stimuli were not designed to induce SP (unlike commonly used artificial moving stimuli), and the participants were not instructed to specifically "follow moving objects" as in e.g., Larsson et al. (2013).

Examining the speed distribution of the occurring SP episodes in the GazeCom data set—see Figure 4—allows us to conclude that, at least for this data set, achieving accurate ternary eye movement classification (i.e., distinguishing fixations, saccades, and pursuits from one another) via any number of speed thresholds is impossible, as the three classes have an overlap in their speeds. The particular challenge is presented by the introduction of smooth pursuit: Fixations and saccades, for example, have practically no overlap in their overall speed, and could be almost perfectly separated with a simple speed threshold (in the absence of SP). SP, however, would be impossible to classify correctly using speed thresholds only (as in I-VVT; Komogortsev, Gobert, Jayarathna, Koh, & Gowda, 2010, for example), as there is a high degree of overlap with fixations, as well as some intersection with the saccade class. Of course, the speed distribution of SP is directly stimulus-dependent: Unlike fixations and saccades, which are only to some extent influenced by the observed stimulus properties (faster paced scenes could reduce average fixation durations, saccade amplitudes depend on the spatial distribution of the objects of interest on the video surface, etc.), pursuit speeds are very close to the speeds of the corresponding targets, at least up to about 100°/s (Meyer, Lasker, & Robinson, 1985). This means that in a different data set of stimuli, the overlap between the speeds of fixations, pursuits, and saccades may look different. However, we note the following: (a) The scenes in the GazeCom data set are representative of the real world (albeit without head rotation freedom for the viewer; in recording set-ups with unrestrained head, nonnegligible head movement is present for a large portion of the time—e.g., ca. 50% in (Agtzidis et al., 2019)). Therefore, our observations should be generalizable to similar conditions in other data sets. (b) The stimuli in our data contained a variety of natural and man-made targets, moving at a range of speed and directions. Since the participants were not instructed to perform a specific task or to exhibit specific viewing behavior during the gaze recording session, we can conclude that the observed SP properties are "natural" in the sense of not being stressful to perform. This means that the pursuit episodes in our data set cover some, but potentially not all of the range for spontaneously occurring SP speeds and directions, implying that the conclusions we make about the difficulty of separating the considered eye movement classes can only be *underestimating* this difficulty in a more generic set-up.

Very similar observations can be made about the plot of the directional deviations of different eye movement types in Figure 5: For these distributions as well, a typical pattern emerges—SP is somewhere "in-between" fixations and saccades, noticeably complicating classification. From the arguments above we infer that simple thresholds of basic eye movement statistics (speed, direction) are not optimal for smooth pursuit classification. Hence combinations of simple properties, higher order statistics, or either implicitly or explicitly learned (e.g., via training machine learning algorithms) complex features are more appropriate for the detection of all eye movements occurring in dynamic scene viewing. It is, however, unclear whether the modalities characterizing the gaze traces alone (speed and direction in this case) provide enough information to distinguish the eye movements from one another. Based on our previous experiments (Startsev, Agtzidis, & Dorr, 2019), we can only claim that (a) complex features learned from basic statistics on a variety of time scales improve classification beyond simple thresholding, and (b) analyzing large segments of gaze traces is much more beneficial than analyzing individual gaze sample characteristics, and increasing the temporal context size for such analysis can drastically improve the classifier.

In order to further examine the viewing behavior in our data set, as well as to quantitatively motivate our clustering-based smooth pursuit detection approach, we computed spatio-temporal synchrony in the eye movements of different types (see Figure 6). The results matched our intuitive expectations about the eye movements that are neither fixations nor saccades—the congruence between the SP samples of different observers is much higher than that for the noise samples, which could be misinterpreted for potential pursuits. In addition to this, we saw that pursuit demonstrated the highest degree of synchrony between the observers, separating it from the other classes (though the percentages for fixations performed synchronously are not much lower). Saccades, on the other hand, are rarely performed at the same time and place by different observers. Figure 6 allows us to directly quantify the synchrony of the different eye movements in our data set: Over 50% of smooth pursuit (fixation) samples are in the immediate spatio-temporal neighborhood of the samples of another seven (six) observers in the GazeCom data. Bearing in mind that GazeCom has an average of 46.9 unique observers' recordings per stimulus, we can see that 11% of smooth pursuit samples belong to episodes that are synchronous *between over half of all the observers* that watched the videos. The same can be said about just 6% of fixation samples. On the same data set as used in this work, Dorr et al. (2010) previously made a broader observation that gaze congruency between observers is the highest when a small number of moving objects are present in the scene, though without considering particular eye movement classes. Mital et al. (2011) also reported that the clustering of gaze points was predicted well by the motion in the video, meaning that pursuit targets are likely to attract attention of multiple subjects at the same time. In Startsev, Göb, and Dorr (2019), temporal interobserver synchrony of the performed eye movements is indirectly examined, but the spatial aspect is not considered.

Manual annotation and “ground truth”

We further compare the annotation pipeline in our work with a recent work by I. T. C. Hooge et al. (2018), who observed that expert annotators often disagree in their fixation annotations when they use their own implicit definitions of the eye movements. Keeping these findings in mind, we provided our annotators with a set of instructions and validated their labels with an additional correction by an expert. The first two annotators in our procedure were not field experts, but they received basic instructions regarding the eye movement types and the labelling process, with only the third annotator having prior experience and expertise in the field. Nevertheless, they demonstrated high agreement when it comes to fixation and saccade episodes both between their two passes and with the final annotator (event-level $F1$ scores for both classes $\geq$ 90% for all annotator pairs), indicating that at least the interpretation of the definitions of these eye movements was consistent between raters. SP labelling, however, is far more subjective, as it seems: Having received identical instructions, the nonexpert annotators disagreed about these labels much more than about the other classes not only between themselves, but also between the first and second pass of the same rater. This disagreement is likely due to the fact that the SP labelling instructions included somewhat intuitive concepts, such as the gaze moving smoothly and the motion of the gaze corresponding to the movement of some target in the scene. The perception of both of these can depend on the zoom level in the labelling interface and the speed at which the rater scrolled through the video frames, not to mention the subjective thresholds and criteria for the presence of motion, its smoothness, and trajectory correspondence. In subsequent versions of the annotation tool (Agtzidis et al., 2019) we have, therefore, included gaze speed plots to be able to set explicit thresholds for annotators (e.g., “a sustained gaze speed of at least X°/s can constitute an SP, provided that there is a target in the scene that moves along a similar trajectory”), thus somewhat eliminating the rater-dependent bias and the dependence on the zoom level.

In this context, it is an interesting question whether the information that is typically presented to human annotators is enough to yield quality eye movement labels. The issue is actually two-fold: (a) Whether enough information is provided to sufficiently characterize the viewing behavior (e.g., should the annotators see the gaze in relation to the stimulus) and (b) whether human annotators (with their limited numerical inference possibilities and visual perception precision) can efficiently use this provided information (with respect to the visualization scale, the necessity to combine information across different plots, or the units of the visualized values, for instance). With respect to the former, several works in the literature (Andersson et al., 2017; I. T. C. Hooge et al., 2018) use an approach where the expert is blind to the stimulus, and therefore cannot assess, for example, the number of potential gaze targets and the position of gaze with respect to them, which could potentially help disentangle a series of fixations in noisy data. In Andersson et al. (2017), the gaze trace is shown at different scales, however, one of which corresponds to the dimensions of the stimulus. Pupil diameter was additionally visualized, which is typically not taken into account by the algorithms. In this work, however, we define smooth pursuit in relation to following a moving (in world coordinates, as the observer’s head is fixed in space) target, so we argue that the visualization of gaze with respect to the video frames is essential. Taken to the extreme, as in Steil et al. (2018), a similar definition can be applied to separate the eye movements into either focusing on a target or not, regardless of whether the target is moving relative to the observer (all denoted as “fixation” in that work). This approach loses the granularity of eye movement analysis, however.

As to the second point, we note the fixed (temporal) scale and a somewhat unintuitive unit for gaze speed (px/s^2) of the visualizations in I. T. C. Hooge et al. (2018). However, providing a speed signal to the annotator could be a great help, especially when several speeds have to be compared and combined for meaningful classification (e.g., for the set-up with unrestrained head motion; Kothari et al., 2017; Agtzidis et al., 2019). As noted by Andersson et al. (2017), any particular way of presenting gaze data to annotators will inevitably bias their internal criteria for distinguishing eye movement classes. However, until bias-free ways of annotating eye movements are developed, manual annotation remains an important part of evaluating and training algorithmic detectors in this field (I. T. C. Hooge et al., 2018).

Algorithmic annotation

In another branch of our analysis, we extended and improved on our previously developed algorithm for pursuit detection (Agtzidis et al., 2016b), which uses the recordings of several observers to improve the detection quality. The optimized parameter set demonstrated excellent performance on the GazeCom data set, in terms of both sample- and event-level measures, including comparing basic episode statistics to the manually annotated events. It also demonstrated its generalizability on an independent data set of Andersson et al. (the video-viewing subset, 2017), for which results were presented in Startsev, Agtzidis, and Dorr (2019): The sp_tool model (with optimized parameters, adjusted according to Appendix, Parameter adaptation for other data sets) yielded the best mean sample- and event-level $F1$ score (averaged across fixations, saccades, and pursuit). Its event-level $F1$ score for SP (0.592) was at least 0.11 higher than that of the next best models on that data.

We discuss the strengths and weaknesses of this clustering-based SP detection approach on an example of the visualization in Figure 1. First of all, it can be seen that when the observers are following distinct targets (the "main" targets that attract most of the attention by their sudden motion onsets), SP is detected relatively well (see the green clusters in Figure 1). Only comparatively few SP episodes are missed in the vicinity of these dense clusters. However, the use of clustering here means that if certain fixation samples, for example, were not detected by the fixation detector beforehand and form dense groups, SP labels will be assigned to them (see two red clusters at the bottom of Figure 1). Similarly, if only a single observer is following a target, the corresponding SP episode(s) will likely be missed due to insufficient sample density (see the continuous blue sample sequence at the top of Figure 1).

The extensive evaluation performed in this work demonstrated that pursuit detection quality increases with the number of observers. This is not characteristic to any other eye movement detection algorithm, since recordings are usually processed independently. The machine learning-based methods (e.g., Zemblys et al., 2018; Startsev, Agtzidis, & Dorr, 2019; Zemblys et al., 2019), also benefit from additional data, but they require additional *annotated* data being provided to improve the trained models, since supervised learning is applied. Our method, on the other hand, only requires additional data *without annotations* due to the unsupervised nature of clustering. This means that the effort required in order to improve pursuit detection quality with our algorithm is much lower than in the case of other data-driven approaches: Data annotation can take up to 18 times longer than the recordings themselves (cf. the Hand labelling statistics section and I. T. C. Hooge et al., 2018); for mobile eye tracking data, the overhead can be even larger (Munn, Stefano, & Pelz, 2008).

Using several recordings per stimulus, of course, imposes certain restrictions on the applicability of the algorithm. First and foremost, there have to be several observers viewing each stimulus. This, however, is relatively typical for video-based eye tracking studies (Itti & Carmi, 2009; Kurzhals et al., 2014; Andersson et al., 2017). For experiments with synthetic stimuli, researchers sometimes randomly generate the motion of the target(s) for each observer (e.g., Santini et al., 2016). Clustering cannot be applied in such cases, but the method remains applicable when the same synthetic sequences are presented to all of the participants.

Another issue with the approach that involves clustering the gaze samples of several recordings is that this processing can only happen when all the recordings have already been collected, i.e., no online detection is possible. However, the pipeline can be modified for online detection of pursuit that occurs during the viewing of the stimuli that have already been presented to other observers. To this end, the already available prerecorded data points are clustered beforehand, and only the core points of the clusters should be retained. The newly arriving gaze coordinates can then be tested for proximity to the preclustered points in a real-time fashion.

Our high-quality algorithmic analysis of eye movement episodes enables automated processing of (large) data corpora collected for dynamic stimuli. In Silberg et al. (2019), for example, our eye movement classification framework was used to automatically detect pursuit in the recordings of 51 participants, who were shown half of the videos of the GazeCom data set (ca. 2.5 hr of eye tracking data). In Startsev and Dorr (2018), automatic eye movement classification via the framework described here was used to produce training data for saliency modelling in a more targeted way, i.e., focusing specifically on predicting human fixations or pursuit. Providing enough training data for a deep learning computer vision system would be impossible without an automated detection system: The training set of the Hollywood2 data set (Mathe & Sminchisescu, 2012), which was used in Startsev and Dorr (2018), comprises well over 30 hours of eye tracking recordings. The fact that the Startsev and Dorr (2018) saliency model that was trained on *automatically detected* pursuit performed better than all of the literature models when predicting *ground truth* pursuit on the GazeCom data set validates the fact that the SP detection method we developed here can be used to study human pursuit patterns in a data-driven way even without manual annotations.

Conclusions

In this work we presented our contributions to both the manual and the automatic analysis of eye movement events in eye tracking recordings. Firstly, we collected a data set of manual eye movement annotations for the entire GazeCom data set, which makes this the largest data set where smooth pursuit was also considered by the annotators. Based on this data set, we, for the first time, quantitatively described and characterized pursuit behavior in dynamic naturalistic scene viewing without instructions or task. We found

that the percentage of samples attributed to smooth pursuit was slightly higher than that for saccades, thus emphasizing the importance of this eye movement in studies with dynamic stimuli. Pursuit also demonstrated the highest spatio-temporal interobserver congruence across all eye movements we annotated, indicating the importance of the targets that induce this type of visual behavior. Motivated by the latter finding, we additionally described and improved our multiobserver smooth pursuit detection algorithm that outperforms other approaches in the literature. We found that the detection quality of our algorithm rises with the number of observers in the data set, which sets it aside from other detectors in the literature: The results of our model can be improved simply by increasing the pool of observers, without manual processing of the additional recordings. The implementation of this algorithm is provided as part of the sp_tool framework, which detects all major eye movement types as well. The code of our methods (including all the data handling procedures, detectors, and several evaluation strategies) is publicly available together with the manual labels we assembled for the full GazeCom data set via https://web.gin.g-node.org/ioannis.agtzidis/gazecom_annotations/.

Keywords: smooth pursuit, data set, natural scenes, eye movement classification, clustering, unsupervised learning

Acknowledgments

This research was supported by the Elite Network Bavaria, funded by the Bavarian State Ministry for Research and Education.

*MS and IA contributed equally to this article.
Commercial relationships: none.

Corresponding author: Mikhail Startsev.
Email: mikhail.startsev@tum.de.
Address: Human-Machine Communication, Technical University of Munich, Munich, Germany.

References

Agtzidis, I., Startsev, M., & Dorr, M. (2016a). In the pursuit of (ground) truth: A hand-labelling tool for eye movements recorded during dynamic scene viewing. In *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)* (pp. 65–68). Baltimore, MD: IEEE.

Agtzidis, I., Startsev, M., & Dorr, M. (2016b). Smooth pursuit detection based on multiple observers. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications* (pp. 303–306). New York, NY: ACM.

Agtzidis, I., Startsev, M., & Dorr, M. (2019). 360-degree video gaze behavior: A ground-truth data set and a classification algorithm for eye movements. In *Proceedings of the 27th ACM International Conference on Multimedia* (pp. 1007–1015). New York, NY: ACM.

Anantrasirichai, N., Gilchrist, I. D., & Bull, D. R. (2016). Fixation identification for low-sample-rate mobile eye trackers. In *2016 IEEE International Conference on Image Processing (ICIP)* (pp. 3126–3130). Phoenix, AZ: IEEE.

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

Atick, J. J., & Redlich, A. N. (1992). What does the retina know about natural scenes? *Neural Computation*, *4*(2), 196–210, https://doi.org/10.1162/neco.1992.4.2.196.

Berg, D. J., Boehnke, S. E., Marino, R. A., Munoz, D. P., & Itti, L. (2009, 05). Free viewing of dynamic stimuli by humans and monkeys. *Journal of Vision*, *9*(5):19, 1–15, https://doi.org/10.1167/9.5.19. [PubMed] [Article]

Dorr, M., Martinetz, T., Gegenfurtner, K. R., & Barth, E. (2010). Variability of eye movements when viewing dynamic natural scenes. *Journal of Vision*, *10*(10):28, 1–17, https://doi.org/10.1167/10.10.28. [PubMed] [Article]

Dowiasch, S., Backasch, B., Einhäuser, W., Leube, D., Kircher, T., & Bremmer, F. (2016). Eye movements of patients with schizophrenia in a natural environment. *European Archives of Psychiatry and Clinical Neuroscience*, *266*(1), 43–54, https://doi.org/10.1007/s00406-014-0567-8.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD Proceedings*, *96*, 226–231. Portland, OR: AAAI.

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, *4*(12), 2379–2394, http://josaa.osa.org/abstract.cfm?URI=josaa-4-12-2379.

Foulsham, T., & Kingstone, A. (2017). Are fixations in static natural scenes a useful predictor of attention in the real world? *Canadian Journal of Experimental*

Psychology/Revue Canadienne de Psychologie Expérimentale, *71*(2), 172–181.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations Newsletter*, *11*(1), 10–18, http://doi.acm.org/10.1145/1656274.1656278.

Hashimoto, K., Suehiro, K., Kodaka, Y., Miura, K., & Kawano, K. (2003). Effect of target saliency on human smooth pursuit initiation: Interocular transfer. *Neuroscience Research*, *45*(2), 211–217, https://doi.org/10.1016/S0168-0102(02)00227-4.

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

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

Hooge, I., 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.

Hooge, I. T. C., Niehorster, D. C., Nyström, M., Andersson, R., & Hessels, R. S. (2018). Is human classification by experienced untrained observers a gold standard in fixation detection? *Behavior Research Methods*, *50*(5), 1864–1881, https://doi.org/10.3758/s13428-017-0955-x.

Itti, L., & Carmi, R. (2009). *Eye-tracking data from human volunteers watching complex video stimuli*. Retrieved from https://crcns.org/data-sets/eye/eye-1/

Joyce, J. M. (2011). Kullback-leibler divergence. In M. Lovric (Ed.), *International encyclopedia of statistical science* (pp. 720–722). Berlin, Heidelberg: Springer Berlin Heidelberg, https://doi.org/10.1007/978-3-642-04898-2_327.

Kasneci, E., Kasneci, G., Kübler, T. C., & Rosenstiel, W. (2014). The applicability of probabilistic methods to the online recognition of fixations and saccades in dynamic scenes. In *Proceedings of the 2014 Symposium on Eye Tracking Research & Applications* (pp. 323–326). New York, NY: ACM.

Komogortsev, O. V. (2014). *Eye movement classification software*. Retrieved from http://cs.txstate.edu/~ok11/emd_offline.html

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.

Komogortsev, O. V., Jayarathna, S., Koh, D. H., & Gowda, S. M. (2010). Qualitative and quantitative scoring and evaluation of the eye movement classification algorithms. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (pp. 65–68). New York, NY: ACM.

Komogortsev, O. V., & Karpov, A. (2013). Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behavior Research Methods*, *45*(1), 203–215.

Kothari, R., Binaee, K., Bailey, R., Kanan, C., Diaz, G., & Pelz, J. (2017). Gaze-in-world movement classification for unconstrained head motion during natural tasks. *Journal of Vision*, *17*(10):1156, https://doi.org/10.1167/17.10.1156. [Abstract]

Krieger, G., Rentschler, I., Hauske, G., Schill, K., & Zetzsche, C. (2000). Object and scene analysis by saccadic eye-movements: An investigation with higher-order statistics. *Spatial Vision*, *13*(2–3), 201–214.

Kurzhals, K., Bopp, C. F., Bässler, J., Ebinger, F., & Weiskopf, D. (2014). Benchmark data for evaluating visualization and analysis techniques for eye tracking for video stimuli. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (pp. 54–60). New York, NY: ACM.

Lagun, D., Manzanares, C., Zola, S. M., Buffalo, E. A., & Agichtein, E. (2011). Detecting cognitive impairment by eye movement analysis using automatic classification algorithms. *Journal of Neuroscience Methods*, *201*(1), 196–203, https://doi.org/10.1016/j.jneumeth.2011.06.027.

Larsson, L., Nyström, M., Andersson, R., & Stridh, M. (2015). Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. *Biomedical Signal Processing and Control*, *18*, 145–152.

Larsson, L., Nyström, M., Ardö, H., Åström, K., & Stridh, M. (2016). Smooth pursuit detection in binocular eye-tracking data with automatic video-based performance evaluation. *Journal of Vision*, *16*(15):20, 1–18, https://doi.org/10.1167/16.15.20. [PubMed] [Article]

Larsson, L., Nyström, M., & Stridh, M. (2013). Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit. *IEEE Transactions on Biomedical Engineering*, *60(9)*, 2484–2493.

Leder, H., Mitrovic, A., & Goller, J. (2016). How beauty determines gaze! Facial attractiveness and gaze duration in images of real world scenes. *i-Perception*, *7*(4), 1–12, https://doi.org/10.1177/2041669516664355.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics* (pp. 281–297). Berkeley, CA: University of California Press.

Marat, S., Ho Phuoc, T., Granjon, L., Guyader, N., Pellerin, D., & Guérin-Dugué, A. (2009). Modelling spatio-temporal saliency to predict gaze direction for short videos. *International Journal of Computer Vision*, *82*(3), 231–243, https://doi.org/10.1007/s11263-009-0215-3.

Mathe, S., & Sminchisescu, C. (2012). Dynamic eye movement datasets and learnt saliency models for visual action recognition. In *Proceedings of the 12th European Conference on Computer Vision* (Vol. 2, pp. 842–856). Berlin, Heidelberg: Springer-Verlag.

McIlreavy, L., Fiser, J., & Bex, P. J. (2012). Impact of simulated central scotomas on visual search in natural scenes. *Optometry and Vision Science: Official Publication of the American Academy of Optometry*, *89*(9), 1385–1394, https://doi.org/10.1097/OPX.0b013e318267a914.

Meyer, C. H., Lasker, A. G., & Robinson, D. A. (1985). The upper limit of human smooth pursuit velocity. *Vision Research*, *25*(4), 561–563, https://doi.org/10.1016/0042-6989(85)90160-9.

Mital, P. K., Smith, T. J., Hill, R. L., & Henderson, J. M. (2011). Clustering of gaze during dynamic scene viewing is predicted by motion. *Cognitive Computation*, *3*(1), 5–24, https://doi.org/10.1007/s12559-010-9074-z.

Monache, S. D., Lacquaniti, F., & Bosco, G. (2019). Ocular tracking of occluded ballistic trajectories: Effects of visual context and of target law of motion. *Journal of Vision*, *19*(4):13, 1–21, https://doi.org/10.1167/19.4.13. [PubMed] [Article]

Mould, M. S., Foster, D. H., Amano, K., & Oakley, J. P. (2012). A simple nonparametric method for classifying eye fixations. *Vision Research*, *57*, 18–25, https://doi.org/10.1016/j.visres.2011.12.006.

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). New York, NY: ACM.

Olsen, A. (2012). *The Tobii I-VT fixation filter*. Retrieved from https://stemedhub.org/resources/2173/download/Tobii_WhitePaper_TobiiIVTFixationFilter.pdf

Parks, D., Borji, A., & Itti, L. (2015). Augmented saliency model using automatic 3D head pose detection and learned gaze following in natural scenes. *Vision Research*, *116*, 113–126, https://doi.org/10.1016/j.visres.2014.10.027.

Ramkumar, P., Lawlor, P. N., Glaser, J. I., Wood, D. K., Phillips, A. N., Segraves, M. A., & Kording, K. P. (2016). Feature-based attention and spatial selection in frontal eye fields during natural scene search. *Journal of Neurophysiology*, *116*(3), 1328–1343, https://doi.org/10.1152/jn.01044.2015.

Salehin, M. M., & Paul, M. (2017). A novel framework for video summarization based on smooth pursuit information from eye tracker data. In *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)* (pp. 692–697). Hong Kong, China: IEEE.

Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications (ETRA)* (pp. 71–78). Palm Beach Gardens, FL: ACM.

San Agustin, J. (2010). *Off-the-shelf gaze interaction* (Doctoral dissertation, IT-Universitetet i Kbenhavn, Copenhagen, Denmark).

Santini, T., Fuhl, W., Kübler, T., & Kasneci, E. (2016). Bayesian identification of fixations, saccades, and smooth pursuits. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications* (pp. 163–170). New York, NY: ACM.

Schomaker, J., Walper, D., Wittmann, B. C., & Einhäuser, W. (2017). Attention in natural scenes: Affective-motivational factors guide gaze independently of visual salience. *Vision Research*, *133*, 161–175, https://doi.org/10.1016/j.visres.2017.02.003.

Schütz, A. C., Braun, D. I., & Gegenfurtner, K. R. (2011). Eye movements and perception: A selective review. *Journal of Vision*, *11*(5):9, 1–30, https://doi.org/10.1167/11.5.9. [PubMed] [Article]

Silberg, J. E., Agtzidis, I., Startsev, M., Fasshauer, T., Silling, K., Sprenger, A., ... Lencer, R. (2019, Jun 01). Free visual exploration of natural movies in schizophrenia. *European Archives of Psychiatry and Clinical Neuroscience*, *269*(4), 407–418, https://doi.org/10.1007/s00406-017-0863-1.

Smith, T. J., & Mital, P. K. (2013, 07). Attentional synchrony and the influence of viewing task on gaze behavior in static and dynamic scenes. *Journal of Vision*, *13*(8):16, 1–24, https://doi.org/10.1167/13.8.16. [PubMed] [Article]

Spering, M., Schütz, A. C., Braun, D. I., & Gegenfurtner, K. R. (2011). Keep your eyes on the ball: Smooth pursuit eye movements enhance prediction of visual motion. *Journal of Neurophysiology*, *105*(4), 1756–1767, http://jn.physiology.org/content/105/4/1756, https://doi.org/10.1152/jn.00344.2010.

SR Research. (2009). *Eyelink 1000 user manual. Version 1.5.0*. Retrieved from http://sr-research.jp/support/EyeLink%201000%20User%20Manual%201.5.0.pdf

Startsev, M., Agtzidis, I., & Dorr, M. (2019). 1D CNN with BLSTM for automated classification of fixations, saccades, and smooth pursuits. *Behavior Research Methods*, *51*(2), 556–572, https://doi.org/10.3758/s13428-018-1144-2.

Startsev, M., & Dorr, M. (2018). Increasing video saliency model generalizability by training for smooth pursuit prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (pp. 2050–2053). Salt Lake City, UT: IEEE.

Startsev, M., Göb, S., & Dorr, M. (2019). A novel gaze event detection metric that is not fooled by gaze-independent baselines. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (pp. 22:1–22:9). New York, NY: ACM.

Steil, J., Huang, M. X., & Bulling, A. (2018). Fixation detection for head-mounted eye tracking based on visual similarity of gaze targets. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (pp.23:1–23:9). New York, NY: ACM.

Swain, M. J., & Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, *7*(1), 11–32, https://doi.org/10.1007/BF00130487.

Tatler, B. W., Hayhoe, M. M., Land, M. F., & Ballard, D. H. (2011). Eye guidance in natural vision: Reinterpreting salience. *Journal of Vision*, *11*(5):5, 1–23, https://doi.org/10.1167/11.5.5. [PubMed] [Article]

Torralba, A., Oliva, A., Castelhano, M. S., & Henderson, J. M. (2006). Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological Review*, *113*(4), 766–786, http://content.apa.org/journals/rev/113/4/766, https://doi.org/10.1037/0033-295X.113.4.766.

Tseng, P.-H., Cameron, I. G. M., Pari, G., Reynolds, J. N., Munoz, D. P., & Itti, L. (2013). High-throughput classification of clinical populations from natural viewing eye movements. *Journal of Neurology*, *260*(1), 275–284, https://doi.org/10.1007/s00415-012-6631-2.

Vidal, M., Bulling, A., & Gellersen, H. (2012). Detection of smooth pursuits using eye movement shape features. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (pp. 177–180). New York, NY: ACM.

Vig, E., Dorr, M., Martinetz, T., & Barth, E. (2011). Eye movements show optimal average anticipation with natural dynamic scenes. *Cognitive Computation*, *3*(1), 79–88, https://doi.org/10.1007/s12559-010-9061-4.

Walther, D., & Koch, C. (2006). Modeling attention to salient proto-objects. *Neural Networks*, *19*(9), 1395–1407.

White, B. J., Berg, D. J., Kan, J. Y., Marino, R. A., Itti, L., & Munoz, D. P. (2017). Superior colliculus neurons encode a visual saliency map during free viewing of natural dynamic video. *Nature Communications*, *8*, 14263.

Williams, L. M., Loughland, C. M., Gordon, E., & Davidson, D. (1999). Visual scanpaths in schizophrenia: Is there a deficit in face recognition? *Schizophrenia Research*, *40*(3), 189–199, https://doi.org/10.1016/S0920-9964(99)00056-0.

Yarbus, A. L. (1967). Eye movements during perception of moving objects. (B. Haigh, Trans.). In L. A. Riggs (Trans. Ed.), *Eye movements and Vision* (pp. 159–170). Boston, MA: Springer US.

Yonetani, R., Kawashima, H., Hirayama, T., & Matsuyama, T. (2012). Mental focus analysis using the spatio-temporal correlation between visual saliency and eye movements. *Journal of Information Processing*, *20*(1), 267–276.

Zemblys, R., Niehorster, D. C., & Holmqvist, K. (2019). gazeNet: End-to-end eye-movement event detection with deep neural networks. *Behavior Research Methods*, *51*(2), 840–864, https://doi.org/10.3758/s13428-018-1133-5.

Zemblys, R., Niehorster, D. C., Komogortsev, O., & Holmqvist, K. (2018). Using machine learning to detect events in eye-tracking data. *Behavior Research Methods*, *50*(1), 160–181, https://doi.org/10.3758/s13428-017-0860-3.

Appendix

Data format

In this section we present the Attribute-Relation File Format (ARFF) that is used throughout our work for eye-tracking data representation. Its description should facilitate the interpretation and usage of our data and algorithms. ARFF is a popular file format in the data mining/machine learning community but largely unknown in the eye-tracking community. We will, therefore, briefly explain it here. A more detailed explanation is given in Agtzidis et al. (2016a). ARFF is an extendible, text-based file format, where all of its keywords are case insensitive and start with the "@" symbol. The "@attribute" keyword is needed to describe each of the columns of the data in the file, specifying its name and type (could be integer, real, or categorical). After the attributes are defined, the "@data" keyword begins the section of the file that contains the set of samples. Each line in this section is a comma-separated list of values corresponding to all of the declared attributes.

As this format does not allow for storing any metadata that characterize the entire recording (e.g., the experimental set-up) and not each individual sample, we extended this format. However, since we wanted to maintain compatibility of our ARFF files with third-party software, e.g., WEKA (Hall et al., 2009), we introduced a special format for the comments in the ARFF files (lines starting with "%"), which starts with "%@metadata" and contains the name and the value of the described meta-attribute (e.g., "%@metadata width_px 1280"). Such comments are correspondingly processed by our software but are safely ignored by other toolkits.

Using this notation enables the storage and extraction of the information specific to the eye tracking experiment, such as the dimensions and properties of the monitor and the eye tracking set-up by simply adding meta-attributes to the header of the ARFF file. We used the following attributes for each recording: the dimensions of the stimulus displayed on the screen in pixels ("width_px" and "height_px") and millimetres ("width_mm" and "height_mm"), as well as the distance from the observer's eyes to the monitor in millimetres ("distance_mm"). These sufficiently define the monitor-based experimental set-ups with fixed head position to compute the pixels-per-degree (PPD) value, which can be used to convert the on-screen gaze position units to visual angle units. This format is flexible enough to allow for effortless extensions to more complex scenarios such as head-mounted display experiments in (Agtzidis et al., 2019).

Parameter optimization

To optimize the parameters of our eye movement classification framework, we tested a random subset of a grid of plausible parameter combinations for our fixation and pursuit detectors. We considered the parameters of the fixation detector even though pursuit detection was of main interest to us because our clustering approach only processes the gaze samples that were not labelled as fixations. If some pursuit samples receive a label of fixation, there is no possibility to retrieve them with our approach. For example, the confusion matrix in Figure 7 demonstrates the performance of a reasonable fixation detector from the literature that has not been optimized together with the subsequent SP detector. This detector labels just under 60% of the "true" SP samples as fixations, which would result in very poor sensitivity.

For fixation detection, we optimized (a) the upper limit for the gaze shift during an intersaccadic interval (intervals with shifts below this threshold were marked as parts of a fixation right away)—0.7° to 2.8°, (b) the lower limit on the intersaccadic interval duration that sets the condition for applying sliding window-based steps to it (intervals with lower durations ignored at this step)—75 to 300 ms, (c) the moving average window size that was applied to every remaining intersaccadic interval to suppress recording noise—3, 5, 7, or 11 samples, (d) the length of the sliding window that was used for analysis—35 to ca. 140 ms, (e) the upper speed threshold for fixation samples—0.7°/s to 4°/s, as well as (f) the minimal plausible SP duration, which was used to label as noise all nonfixation episodes of a shorter duration—35 ms to ca. 140 ms.

For smooth pursuit detection (i.e., the parameters of our DBSCAN modification), we optimized (g) the spatial distance threshold ε_{xy}—1° to 4°, (h) the temporal distance threshold ε_t—0 to 160 ms, and (i) the *minPts* parameter—20 to 320, as well as setting *minPts* to the number of observers, whose recordings are being processed for a given stimulus (the latter was the value used in Agtzidis et al., 2016b).

The parameters marked with (a), (b), (d), (e), (f), and (g) were randomly sampled on the logarithmic grid with the base of $\sqrt{2}$; those marked with (h) and (i), with the base of 2. The grid was constructed to explore parameter combinations with values both lower and greater than in the parameter set of Agtzidis et al. (2016b). A total of 2.25 million combinations of these values are possible. We randomly sampled ca. 6500 of those to assess the possible performance range of the algorithm.

To make sure the parameter set we would choose based on this optimization was relatively stable to fluctuations in the data, as well as to ensure some degree of the best parameters' ability to be generalized, we performed this optimization on two nonoverlapping subsets of the data separately, and then selected a parameter set that performed consistently well on both subsets. Recordings were split based on the corresponding stimuli (half of the GazeCom video clips in each part). We split the recordings this way as we intuitively suspected that decreasing the number of observers will have a negative impact on the algorithm's performance (we tested this experimentally in the Robustness to variations in the number of observers section). Splitting the data set by video clips rather than by the individual observers has proven to have another positive effect in our recent work (Startsev, Agtzidis, & Dorr, 2019): We found that optimizing an algorithm for all clips, but only a subset of observers, leads to more prominent overfitting behavior than optimizing it for all observers, but only a subset of clips. This effect was especially noticeable for SP detection, which is the main target of our optimization here.

Therefore, out of the tested ca. 6500 parameter combinations, we selected the top 25 (less than 0.5%) for each of the two data subsets independently, ranked by the F1 score for smooth pursuit samples. This yielded six parameter combinations that were within the selected percentile for both subsets simultaneously. We chose the parameter set that resulted in the best average *F*1 score across the subsets. We provide the full parameter sets corresponding both to the original method (Agtzidis et al., 2016b) and to the optimized version, which we obtained here, together with the code of our model on the code repository page.

Parameter adaptation for other data sets

Here we describe the adjustment that has to be made to the parameters of our algorithm to adapt it to be used with a different data set. The full set of parameters is stored in a configuration file and can be accessed and adjusted with a text editor.

Only a minor change is required to adapt the clustering algorithm for a different use case, however. It has to do with the *minPts* parameter, which defines the number of gaze points in the spatio-temporal vicinity of the considered gaze point that is necessary to make this point a core point of a cluster. This number has a linear dependency on (a) the sampling rate and (b) the number of observers in the data set. The *minPts* parameter has to be scaled accordingly. GazeCom has the sampling rate of $F_{GazeCom} = 250$ Hz and $N_{GazeCom} = 46.9$ observers per clip on average. Therefore, in order to use our algorithm on a new data set with the sampling frequency $\hat{F}$ and $\hat{N}$ observers for each clip, the parameter has to be updated as follows:

$$minPts = minPts_{GazeCom} * \frac{\hat{F}}{F_{GazeCom}} * \frac{\hat{N}}{N_{GazeCom}}, \quad (1)$$

where $minPts_{GazeCom} = 160$, taken from our optimized parameter set. We used this correction formula for our experiments with reducing the number of observers in the Robustness to variations in the number of observers section, and in Startsev, Agtzidis, and Dorr (2019) to adapt the parameters of this method to the data set of Andersson et al. (2017).

In case data quality is substantially different from the GazeCom data, other parameters might need to be altered as well. For example, it would make sense to increase ε_{xy} for noisy recordings, and larger ε_t could be advisable for lower frequency data.

Observer-driven clustering extension of DBSCAN

While the regular DBSCAN determines whether each data point belongs to a dense cluster by comparing the number of unique *gaze samples* in its neighborhood to a fixed threshold, we propose considering the number of unique *observers* with their samples in this neighborhood (see Figure A1). The number of unique observers' gaze traces in the vicinity of the considered gaze point will be then compared to a threshold, to which we refer as *minObservers,* analogously to the *minPts* parameter of the original DBSCAN algorithm. In the sp_tool framework, the *minObservers* parameter can be set either to an integer (in which case it is directly used for thresholding) or to a floating point value in the [0, 1] range (in which case it indicates the share of the number of participants that have viewed each individual stimulus). The actual threshold in the latter case is then computed for each stimulus individually. If the *minObservers* threshold is set as a proportion of the total number of observers, there are no parameter adjustments that need to be made to adapt the clustering scheme to other data sets, as this density criterion does not directly depend on the absolute number of observers in the data set or the sampling frequency of its recordings (though ε_t might need to be increased if the sampling rate is too low—the optimal ε_t for this version of the algorithm was 20 ms, which is shorter than the sampling interval of some eye trackers).

We optimized the parameters for this DBSCAN variation in the same way as for its *minPts* version (see Appendix, Parameter optimization) and provide the optimal parameter set together with the source code. The *minObservers* threshold that yielded the best performance in our random search (values from 0.05 to 0.2 were tested, with the log-scale grid with the base of $\sqrt{2}$) was 0.14 (for the full GazeCom data set this is on average equivalent to six observers).

This parameter combination was additionally tested on the subsets of the GazeCom data with a varying number of observers (same as for the *minPts* version in the the Robustness to variations in the number of

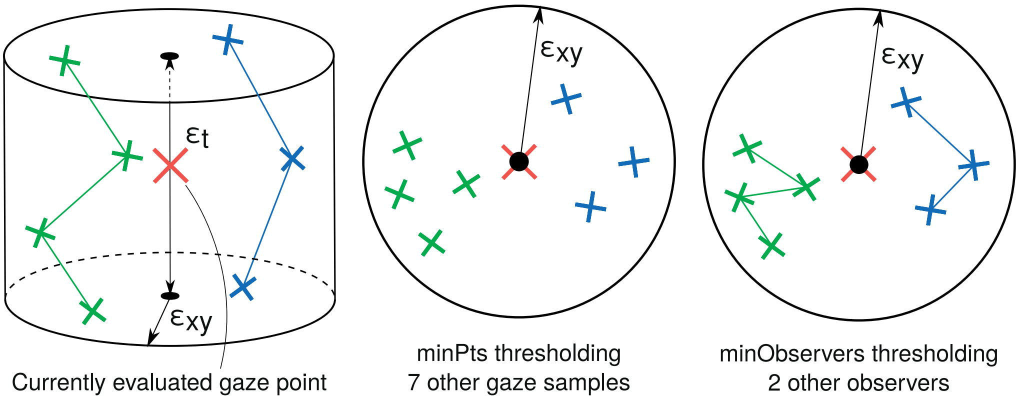


Figure 10. DBSCAN modification specifically for eye tracking recordings: In order to ascertain whether each considered data point (on the left side, together with its spatio-temporal neighborhood) belongs to a cluster, traditional DBSCAN checks the number of (other) data points in its vicinity (middle). Our proposed modification would consider the number of (other) *observers'* gaze traces (right side) in the neighborhood of the considered data point.

observers section), without any parameter correction required whatsoever. We observed performance patterns similar to those in Figure 9, but the values for the *minObservers* version of the algorithms were always below those for the *minPts* variant: The sample-level F1 scores were typically 0.02 worse; the event-level scores, ca. 0.1 lower. Based on this, we cannot recommend using the observer-based modification of our algorithm when detection performance is the key issue. It may, however, serve as an easier generalizable solution and an example of tailoring generic data analysis strategies specifically to eye-tracking recording processing.

C

1D CNN with BLSTM for Eye Movement Classification

This work has proposed utilising deep learning for eye movement classification. We based our architecture on a combination of convolutional and (bidirectional) long short-term memory (LSTM) layers, as is common for video data processing in computer vision. The model we developed operates on windows of gaze data, producing windows of per-class probabilities as an output. We used four eye movement classes in this work – fixations, saccades, smooth pursuits (SPs), and noise, but the model can be used for any set of labels, *e.g.* to include post-saccadic oscillations (PSOs) [52], or expanded to produce several labels per sample, as could be necessary in head-mounted display set-ups [1*].

In this work, while keeping the model architecture constant, we performed a variety of tests regarding the features extracted from the eye tracking data prior to it being passed to the network. While the convolutional part of our network is capable of implicitly learning useful representations from the x, y signal directly, it substantially benefited from simple feature extraction. In particular, gaze speed (on five temporal scales) has proven to improve the model's performance, as well as its combination with gaze direction.

As our model can operate on arbitrary-length temporal windows of gaze data (given the hardware constraints, and as long as it is trained and tested similarly), we analysed the influence of the size of such windows on its performance. While increasing the temporal context size improved the detection of all considered eye movement classes, it particularly benefited SP, emphasising both the room for improvement for its detectors, and their sensitivity to the temporal window, in which the analysis is performed.

We also developed a new event-level evaluation procedure for the considered problem, matching the events in the ground truth and their detections based on the degree of their overlap – intersection-over-union ratio (IoU) – rather than the absolute overlap duration [52]. Finally, we proposed evaluating the models at a range of strictness degrees for the event matching criterion, enabling a more rigorous evaluation.

My personal contributions include (i) the idea, design, and implementation of the full learning pipeline, including the newly developed evaluation techniques, (ii) designing and carrying out experiments on two data sets, (iii) developing and testing a cross-validation procedure specifically for eye tracking experiments, and (iv) writing the manuscript.

Behavior Research Methods (2019) 51:556–572
https://doi.org/10.3758/s13428-018-1144-2

1D CNN with BLSTM for automated classification of fixations, saccades, and smooth pursuits

Mikhail Startsev[1] · Ioannis Agtzidis[1] · Michael Dorr[1]

Published online: 8 November 2018

Abstract

Deep learning approaches have achieved breakthrough performance in various domains. However, the segmentation of raw eye-movement data into discrete events is still done predominantly either by hand or by algorithms that use hand-picked parameters and thresholds. We propose and make publicly available a small 1D-CNN in conjunction with a bidirectional long short-term memory network that classifies gaze samples as fixations, saccades, smooth pursuit, or noise, simultaneously assigning labels in windows of up to 1 s. In addition to unprocessed gaze coordinates, our approach uses different combinations of the speed of gaze, its direction, and acceleration, all computed at different temporal scales, as input features. Its performance was evaluated on a large-scale hand-labeled ground truth data set (GazeCom) and against 12 reference algorithms. Furthermore, we introduced a novel pipeline and metric for event detection in eye-tracking recordings, which enforce stricter criteria on the algorithmically produced events in order to consider them as potentially correct detections. Results show that our deep approach outperforms all others, including the state-of-the-art multi-observer smooth pursuit detector. We additionally test our best model on an independent set of recordings, where our approach stays highly competitive compared to literature methods.

Keywords Eye-movement classification · Deep learning · Smooth pursuit

Introduction

Eye-movement event detection is important for many eye-tracking applications as well as the understanding of perceptual processes. Automatically detecting different eye movements has been attempted for multiple decades by now, but evaluating the approaches for this task is challenging, not least because of the diversity of the data and the amount of manual labeling required for a meaningful evaluation. To compound this problem, even manual annotations suffer from individual biases and implicitly used thresholds and rules, especially if experts from different sub-areas are involved (Hooge, Niehorster, Nyström, Andersson, & Hessels, 2017). For smooth pursuit (SP), even detecting episodes[1] by hand is not entirely trivial (i.e., requires additional information) when the information about their targets is missing. Especially when pursuit speed is low, it may be confused with drifts or oculomotor noise (Yarbus, 1967).

Therefore, most algorithms to date are based on hand-tuned thresholds and criteria (Larsson, Nyström, Andersson, & Stridh, 2015; Berg, Boehnke, Marino, Munoz, & Itti, 2009; Komogortsev, Gobert, Jayarathna, Koh, & Gowda, 2010). The few data-driven methods in the literature either do not consider smooth pursuit (Zemblys, Niehorster, Komogortsev, & Holmqvist, 2017), operate on data produced with low-variability synthetic stimuli (Vidal, Bulling, & Gellersen, 2012), or are not yet publicly available (Hoppe & Bulling, 2016).

Here, we propose and make publicly available a neural network architecture that differentiates between the three major eye-movement classes (fixation, saccade, and smooth

✉ Mikhail Startsev
mikhail.startsev@tum.de

Ioannis Agtzidis
ioannis.agtzidis@tum.de

Michael Dorr
michael.dorr@tum.de

[1] Technical University of Munich, Institute for Human-Machine Communication, Arcisstr. 21, Munich, 80333, Germany

[1] We use the terms "event" and "episode" interchangeably, both referring to an uninterrupted interval, where all recorded gaze samples have been assigned the same respective label, be that in the ground truth or in algorithmic labels.

pursuit) while also taking potential noise (e.g., blinks or lost tracking) into account. Our approach learns from data (simple features) to assign sequences of labels to sequences of data points with a compact one-dimensional convolutional neural network (1D-CNN) and bidirectional long short-term memory block (BLSTM) combination. Evaluated on a fully annotated[2] (Startsev, Agtzidis, & Dorr, 2016) GazeCom data set of complex natural movies (Dorr, Martinetz, Gegenfurtner, & Barth, 2010), our approach outperforms all 12 evaluated reference models, for both sample- and event-level detection. We additionally test our method's generalization ability on the data set of Andersson, Larsson, Holmqvist, Stridh, and Nyström (2017).

Related work

Data sets

Despite the important role of smooth pursuit in our perception and everyday life (Land, 2006), its detection in free-viewing scenarios has been somewhat neglected. At the very least, it should be considered in event detectors to avoid false detections of other eye-movement types (Andersson et al., 2017). Even when taking into account information about gaze patterns of dozens of observers at once (Agtzidis, Startsev, & Dorr, 2016b), there is a dramatic gap between the performance of detecting saccades or fixations, and detecting smooth pursuits (Startsev et al., 2016).

We will, therefore, use the largest publicly available manually annotated eye-tracking data set that accounts for smooth pursuit to train and validate our models: GazeCom (Dorr et al., 2010; Startsev et al., 2016) (over 4 h of 250-Hz recordings with SR Research EyeLink II). Its data files contain labels of four classes, with noise labels (e.g., blinks and tracking loss) alongside fixations, saccades, and smooth pursuits. We maintain the same labeling scheme in our problem setting (including the introduced, yet unused, "unknown" label).

We additionally evaluate our approach on a small high-frequency data set that was originally introduced by Larsson, Nyström, and Stridh (2013) (data available via (Nyström, 2015); ca. 3.5 min of 500-Hz recordings with SensoMotoric Instruments Hi-Speed 1250), also recently used in a larger review of the state of the art by Andersson et al. (2017). This data set considers postsaccadic oscillations in manual annotation and algorithmic analysis, which is not common yet for eye-tracking research.

Another publicly available data set that includes smooth pursuit, but has low temporal resolution, accompanies the work of Santini, Fuhl, Kübler, and Kasneci (2016) (available at Santini, 2016; ca. 15 min of 30-Hz recordings with a Dikablis Pro eye tracker). This work, however, operates in a different data domain: pupil coordinates on raw eye videos. Because it was not necessary for the algorithm, no eye tracker calibration was performed, and therefore no coordinates are provided with respect to the scene camera. Post hoc calibration is possible, but it is recording-dependent. Nevertheless, the approach of Santini et al. (2016) presents an interesting ternary (fixations, saccades, smooth pursuit) probabilistic classifier.

Automatic detection

Many eye-movement detection algorithms have been developed over the years. A simple, yet versatile toolbox for eye-movement detection is provided by Komogortsev (2014). It contains Matlab implementations for a diverse set of approaches introduced by different authors. Out of the eight included algorithms, five (namely, I-VT and I-DT (Salvucci & Goldberg, 2000), I-HMM (Salvucci & Anderson, 1998), I-MST (Goldberg & Schryver, 1995; Salvucci & Goldberg, 2000), and I-KF Sauter, Martin, Di Renzo, & Vomscheid, 1991) detect fixations and saccades only (cf. Komogortsev et al. 2010 for details).

I-VVT, I-VMP, and I-VDT, however, detect the three eye-movement types (fixations, saccades, smooth pursuit) at once. I-VVT (Komogortsev & Karpov, 2013) is a modification of the I-VT algorithm, which introduces a second (lower) speed[3] threshold. The samples with speeds between the high and the low thresholds are classified as smooth pursuit. The I-VMP algorithm (San Agustin, 2010) keeps the high speed threshold of the previous algorithm for saccade detection, and uses window-based scoring of movement patterns (such as pair-wise magnitude and direction of movement) for further differentiation. When the score threshold is exceeded, the respective sample is labeled as belonging to a smooth pursuit. I-VDT (Komogortsev & Karpov, 2013) uses a high speed threshold for saccade detection, too. It then employs a modified version of I-DT to separate pursuit from fixations.

Dorr et al. (2010) use two speed thresholds for saccade detection alone: The high threshold is used to detect the peak-speed parts in the middle of saccades. Such detections are then extended in time as long as the speed stays above the low threshold. This helps filter out tracking noise and other artifacts that could be mistaken for a saccade, if only

[2] The recordings were algorithmically pre-labeled to speed up the hand-labeling process, after which three manual annotators have verified and adjusted the labeled intervals in all of the files.

[3] Unfortunately, in the eye-movement literature, the term "velocity" (in physics, a vector) often is used to refer to "speed" (the scalar magnitude of velocity). Here, we try to be consistent and avoid using "velocity" when it is not justified.

the low threshold was applied. Fixations are determined while trying to avoid episode contamination with smooth pursuit samples. The approach uses a sliding window inside intersaccadic intervals, and the borders of fixations are determined via a combination of modified dispersion and speed thresholds.

Similarly, the algorithm proposed by Berg et al. (2009) was specifically designed for dynamic stimuli. Here, however, the focus is on distinguishing saccades from pursuit. After an initial low-pass filtering, the subsequent classification is based on the speed of gaze and principal component analysis (PCA) of the gaze traces. If the ratio of explained variances is near zero, the gaze follows an almost straight line. Then, the window is labeled either as a saccade or smooth pursuit, depending on speed. By combining information from several temporal scales, the algorithm is more robust at distinguishing saccades from pursuit. The samples that are neither saccade nor pursuit are labeled as fixations. The implementation is part of a toolbox (Walther & Koch, 2006).

An algorithm specifically designed to distinguish between fixations and pursuit was proposed by Larsson et al. (2015), and its re-implementation is provided by Startsev et al. (2016). It requires a set of already detected saccades, as it operates within intersaccadic intervals. Every such interval is split into overlapping windows that are classified based on four criteria. If all criteria are fulfilled, the window is marked as smooth pursuit. If none are fulfilled, the fixation label is assigned. Windows with one to three fulfilled criteria are labeled based on their similarity to already-labeled windows.

Several machine learning approaches have been proposed as well. Vidal et al. (2012) focus solely on pursuit detection. They utilize shape features computed via a sliding window, to then use k-NN based classification. The reported accuracy of detecting pursuit is over 90%, but the diversity of the data set is clearly limited (only purely vertical and horizontal pursuit), and reporting accuracy without information about class balance is difficult to interpret.

Hoppe and Bulling (2016) propose using convolutional neural networks to assign eye-movement classes. Their approach, too, operates as a sliding window. For each window, the Fourier-transformed data are fed into a network, which contains one convolutional, one pooling, and one fully connected layer. The latter estimates the probabilities of the central sample in the window belonging to a fixation, a saccade, or a pursuit. The network used in this work is rather small, but the collected data seems diverse and promising. The reported scores are fairly low (average F1 score for detecting the three eye movements of 0.55), but without the availability of the test data set, it is impossible to assess the relative performance of this approach.

An approach by Anantrasirichai, Gilchrist, and Bull (2016) is to identify fixations in mobile eye tracking via an SVM, while everything else is attributed to the class "saccades and noise". The model is trained with gaze trace features, as well as image features locally extracted by a small 2D CNN. The approach is interesting, but the description of the data set and evaluation procedure lacks details.

A recently published work by Zemblys et al. (2017) uses Random Forests with features extracted in 100–200-ms windows that are centered around respective samples. It aims to detect fixations, saccades, and postsaccadic oscillations. This work also employs data augmentation to simulate various tracking frequencies and different levels of noise in data, which adds to the algorithm's robustness.

Unfortunately, neither of these machine learning approaches are publicly available, at least in such a form that allows out-of-the-box usage (e.g., the implementation of Zemblys et al. (2017) lacks a pre-trained classifier).

All the algorithms so far process gaze traces individually. Agtzidis et al. (2016b) (toolbox available at Startsev et al., 2016) detect saccades and fixations in the same fashion, but use inter-observer similarities for the more challenging task of smooth pursuit detection. The samples remaining after saccade and fixation detection are clustered in the 3D space of time and spatial coordinates. Smooth pursuit by definition requires a target, so all pursuits on one scene can only be in a limited number of areas. Since no video information is used, inter-observer clustering of candidate gaze samples is used as a proxy for object detection: If many participants' gaze traces follow similar paths, the chance of this being caused by tracking errors or noise is much lower. To take advantage of this effect, only those gaze samples that were part of some cluster are labeled as smooth pursuit. Those identified as outliers during clustering are marked as noise. This way, however, many pursuit-like events can be labeled as noise due to insufficient inter-observer similarity, and multiple recordings for each clip are required in order to achieve reliable results.

Our approach

We here propose using a bidirectional long short-term memory network (one-layer, in our case), which follows the processing of the input feature space by a series of 1D convolutional layers (see Fig. 1). Unlike (Hoppe & Bulling, 2016) and the vast majority of other literature approaches, we step away from single sample classification in favor of simultaneous window-based labeling, in order to capture temporal correlations in the data. Our network receives and outputs a window of samples-level features and labels, respectively. Unlike most of the methods in the literature, we also assign the "noise" label, which does not force our model to choose only between the meaningful classes, when this is not sensible.

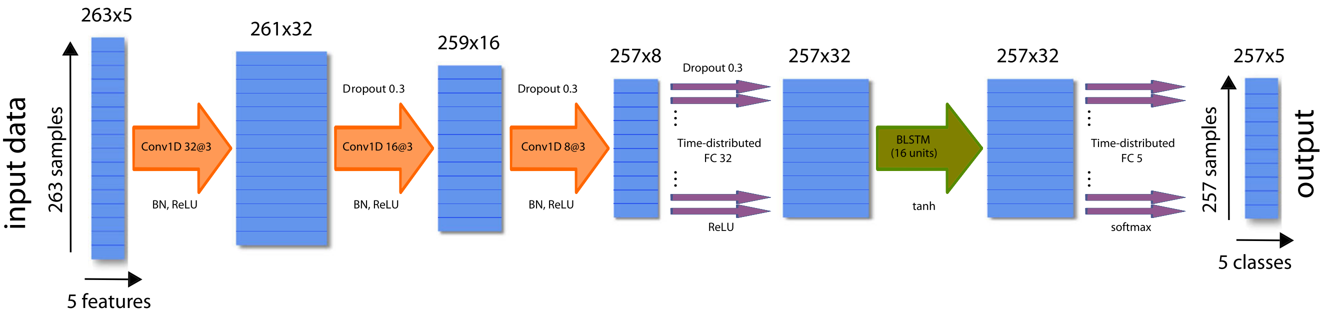


Fig. 1 The architecture of our 1D CNN-BLSTM network. BN stands for "batch normalization", FC – for "fully connected". In this figure, the input is assumed to contain the five different-scale speed features, and the context window size that is available to the network is just above 1 s

Features for classification

We used both the raw x and y coordinates of the gaze and simple pre-computed features in our final models. To avoid overfitting for the absolute gaze location when using the xy-coordinates, we used positions relative to the first gaze location in each processed data window. Our initial experiments showed that a small architecture such as ours noticeably benefits from feature extraction on various temporal scales prior to passing the sequence to the model. This is especially prominent for smooth pursuit detection. With a limited number of small-kernel convolutional layers, network-extracted features are influenced only by a small area in the input-space data (i.e., the feature extracting sub-network has a small receptive field, seven samples, or 28 ms, in the case of our network). With this architecture it would thus be impossible to learn motion features on coarser temporal scales, which are important, e.g., for detecting the relatively persistent motion patterns, which characterize smooth pursuits. To overcome this, we decided to use precomputed features; specifically, we included speed, acceleration, and direction of gaze, all computed at five temporal scales in order to capture larger movement patterns on the feature level already.

The speed of gaze is an obvious and popular choice (Sauter et al., 1991; Salvucci & Goldberg, 2000; Komogortsev et al., 2010; Komogortsev & Karpov, 2013). Acceleration could aid saccade detection, as it is also sometimes used in the literature (Collewijn & Tamminga, 1984; Nyström & Holmqvist, 2010; Behrens, MacKeben, & Schröder-Preikschat, 2010; Larsson et al., 2013) as well as in SR Research's software for the EyeLink trackers.

The effect of using the direction of gaze is less obvious: Horizontal smooth pursuit, for instance, is more natural to our visual system (Rottach, Zivotofsky, Das, Averbuch-Heller, Discenna, Poonyathalang, & Leigh, 1996). The drifts that occur due to tracking artifacts are, however, more pronounced along the vertical axis (Kyoung Ko, Snodderly, & Poletti, 2016).

We consider five different temporal scales for feature extraction: 4, 8, 16, 32, and 64 ms. The speed (in $^\circ/s$) and direction (in radians, relative to the horizontal vector from left to right) of gaze for each sample were computed via calculating the displacement vector of gaze position on the screen from the beginning to the end of the temporal window of the respective size, centered around the sample. Acceleration (in $^\circ/s^2$) was computed from the speed values of the current and the preceding samples on the respective scale (i.e., numerical differentiation; acceleration for the first sample of each sequence is set to 0). If a sample was near a prolonged period of lost tracking or either end of a recording (i.e., if gaze data in a part of the temporal window was missing), a respectively shorter window was used.

We additionally conducted experiments on feature combinations, concatenating feature vectors of different groups, in order to further enhance performance.

Data sets

GazeCom

We used the GazeCom (Dorr et al., 2010; Startsev et al., 2016) recordings for both training and testing (manual annotation in Agtzidis, Startsev, & Dorr, 2016a), with a strict cross-validation procedure. This data set consists of 18 clips, around 20 s each, with an average of 47 observers per clip (total viewing time over 4.5 h). The total number of individual labels is about 4.3 million (including the samples still recorded after a video has finished; 72.5, 10.5, 11, and 5.9% of all samples are labeled as parts of fixations, saccades, pursuits, or noise, respectively). Event-wise, the data set contains 38629 fixations, 39217 saccades, and 4631 smooth pursuits. For training (but not for testing) on this data set, we excluded gaze samples with timestamps over 21 s (confidently outside video durations) and re-sampled to 250-Hz recordings of one of the observers (SSK), whose files had a higher sampling frequency.

We used leave-one-video-out (LOVO) cross-validation for evaluation: The training and testing is run 18 times, each time training on all the data for 17 videos and testing on all the eye-tracking data collected for the remaining video clip.

This way, the model that generates eye-movement labels on a certain video had never seen any examples of data with this clip during training. We aggregate the labeled outputs for the test sets of all splits before the final evaluation.

There are two major ways to fully utilize an eye-tracking data set in the cross-validation scenario, in the absence of a dedicated test subset. The first, LOVO, is described above, and it ensures that no video clip-specific information can benefit the model. The second would ensure that no observer-specific information would be used. For this, we used a leave-n-observers-out (LnOO) procedure. In our case, to maintain symmetry with the 18 splits of LOVO, we introduced the same number of splits in our data, each containing three unique randomly selected observers (54 participants in total).

We hypothesize that LOVO should be less susceptible to overfitting than LnOO, since smooth pursuit is target-driven, and the observers' scanpaths tend to cluster in space and time (Agtzidis et al., 2016b), meaning that their characteristics for different observers would be similar for the same stimulus. We test this hypothesis with several experiments, where the only varied parameter is the cross-validation type.

Nyström-Andersson data set

We used an independent set of recordings in order to additionally validate our model. For this, we took the manually annotated eye-tracking recordings that were utilized in a recent study (Andersson et al., 2017). These contain labels provided by two manual raters: One rater ("CoderMN") originally labeled the data for Larsson et al. (2013), another ("CoderRA") was added by Andersson et al. (2017). The annotations of both raters include six labeled classes: fixations, saccades, postsaccadic oscillations (PSOs), smooth pursuit, blinks, and undefined events.

The whole data set comprises three subsets that use moving dots, images, and video clips as stimuli. We focus our evaluation on the "video" part, since our model was trained on this domain.

We will refer to this subset by the abbreviations of the manual labelers' names (in chronological order of publications, containing respective data sets): "MN-RA-data". In total, it contains ca. 58000 gaze samples (or about 2 min at 500 Hz). Notably, only half of this data consists of "unique" samples, the second half being duplicated, but with different ground truth labels (provided by the second rater). 37.7% of all the samples were labeled as fixation, 5.3% as saccade, 3% as PSO, 52.2% as pursuit, 1.7% as blink, and 0.04% as "unknown". Counting events yields 163 fixations, 244 saccades, 186 PSOs, 121 pursuits, and 8 blinks. The high ratio of pursuit is explained by the explicit instructions given to the participants ("follow [...] moving objects for video clips" Larsson et al. 2013) vs. free viewing in GazeCom (Dorr et al., 2010).

As in Andersson et al. (2017), we evaluated all the considered automatic approaches and both manual raters against the "average coder" (i.e., effectively duplicating each recording, but providing the "true" labels by MN in one case and by RA in the other).

Model architecture

We implemented a joint architecture that consists of a small one-dimensional temporal convolutional network and a bidirectional LSTM layer, with one time-distributed dense layer both before and after the BLSTM (for higher-level feature extraction and to match the number of classes in the output without limiting the number of neurons in the BLSTM, respectively). In this work, we implement multi-class classification with the following labels: fixation, saccade, smooth pursuit, noise (e.g., blinks or tracking loss), also "unknown" (for potentially using partially labeled data), in order to comply with the ground truth labeling scheme in Startsev et al. (2016). The latter label was absent in both the training data and the produced outputs. The architecture is also illustrated in Fig. 1 on an example of using a five-dimensional feature space and simultaneously predicting labels in a window equivalent to about 1 s of 250-Hz samples.

The network used here is reminiscent of other deep sequence-processing approaches, which combine either 2D (Donahue, Anne Hendricks, Guadarrama, Rohrbach, Venugopalan, Saenko, & Darrell, 2015) or, more recently, 3D (Molchanov, Yang, Gupta, Kim, Tyree, & Kautz, 2016) convolutions with recurrent units for frame sequence analysis. Since our task is more modest, our parameter count is relatively low (ca. 10000, depending on the input feature space dimensionality, compared to millions of parameters even in compact static CNNs (Hasanpour, Rouhani, Fayyaz, & Sabokrou, 2016), or ca. 6 million parameters only for the convolutional part (Tran, Bourdev, Fergus, Torresani, & Paluri, 2015) of Molchanov et al., 2016).

The convolutional part of our architecture contains three layers with a gradually decreasing number of filters (32, 16, and 8) with the kernel size of 3, and a batch normalization operation before activation. The subsequent fully connected layer contains 32 units. We did not use pooling, as is customary for CNNs, since we wanted to preserve the one-to-one correspondence between inputs and outputs. This part of the network is therefore not intended for high-level feature extraction, but prepares the inputs to be used by the BLSTM that follows.

All layers before the BLSTM, except for the very first one, are preceded by dropout (rate 0.3), and use ReLU as activation function. The BLSTM (with 16 units) uses *tanh*, and the last dense layer (5 units, according to the number of classes) – softmax activation.

The input to our network is a window of a pre-set length, which we varied to determine the influence of context size on the detection quality for each class. To minimize the border effects, our network uses valid padding, requiring its input to be padded. For both training and testing, we only mirror-pad the whole sequence (of undetermined length; typically ca. 5000 in our data), and not each window. We pad by three samples on each side (since each of the three convolutional layers uses valid padding and a kernel of size 3). For our context-size experiments, this means that for a prediction window of 129 samples (i.e., classification context size 129), for example, windows of length 135 must be provided as input. So when we generate the labels for the whole sequence, the neighboring input windows overlap by six samples, but the output windows do not.

We balanced neither training nor test data in any way.

To allow for fair comparison of results with different context sizes, we attempted to keep the training procedure consistent across experiments. To this end, we fixed the number of windows (of any size) that are used for training (90%) and for validation (10%) to 50000. For experiments with context no larger than 65 samples, we used windows without overlap and randomly sampled (without replacement) the required number of windows for training. For larger-context experiments, the windows were extracted with a step of 65 samples (at 250 Hz, equivalent to 260 ms).

We initialized convolutional and fully connected layers with random weights from a normal distribution (mean 0, standard deviation 0.05), and trained the network for 1000 iterations with batch size 5000 with the RMSprop optimizer (Tieleman & Hinton, 2012) with default parameters from the Keras framework (Chollet et al., 2015) (version 2.1.1; learning rate 0.001 without decay) and categorical cross-entropy as the loss function.

Evaluation

Metrics

Similar to Agtzidis et al. (2016b), Startsev et al. (2016), and Hoppe and Bulling (2016), we evaluated *sample-level* detection results. Even though all our models (and most of the baseline models) treat eye-movement classification as a multi-class problem, for evaluation purposes we consider each eye movement in turn, treating its detection as a binary classification problem (e.g., with labels “fixation” and “not fixation”). This evaluation approach is commonly used in the literature (Larsson et al., 2013; Agtzidis et al., 2016b; Andersson et al., 2017; Hoppe & Bulling, 2016). We can then compute typical performance metrics such as precision, recall, and F1 score.

As for the *event-level* evaluation, there is no consensus in the literature as to which methodology should be employed. Hoppe and Bulling (2016), for example, use ground truth event boundaries as pre-segmentation of the sequences: For each event in the ground truth, all corresponding individual predicted sample labels are considered. The event is classified by the majority vote of these labels. As Hoppe and Bulling (2016) themselves point out, this pre-segmentation noticeably simplifies the problem of eye-movement classification. Additionally, the authors only reported confusion matrices and respective per-class hit rates, which conceal the problem of false detections. Andersson et al. (2017) only assess the detected events in terms of the similarity of event duration distribution parameters to those of the ground truth.

In Hooge, Niehorster, Nyström, Andersson, and Hessels (2017), *event-level* fixation detection was assessed by an arguably fairer approach with a set of metrics that includes F1 scores for fixation episodes. We computed these for all three main event types in our data (fixations, saccades, and smooth pursuits): For each event in the ground truth, we look for the earliest algorithmically detected event of the same class that intersects with it. Only one-to-one matching is allowed. Thus, each ground truth event can be labeled as either a “hit” (a matching detected event found) or a “miss” (no match found). The detected events that were not matched with any ground truth events are labeled as “false alarms”. These labels correspond to true positives, false negatives, and false positives, which are needed to compute the F1 score.

One drawback of such matching scheme is that the area of event intersection is not taken into account. This way, for a ground truth event E_{GT}, the earlier detected event E_A will be preferred to a later detected event E_B, even if the intersection with the former is just one sample, i.e., $|E_{GT} \cap E_A| = 1$, while the intersection with E_B is far greater. Hooge et al. (2017) additionally compute measures such as relative timing offset and deviation of matched events in order to tie together agreement measures and eye-movement parameters, which would also penalize potential poor matches. These, however, have to be computed for both onset and offset of each event type, and are more suited for in-detail analysis of particular labeling patterns rather than for a concise quantitative evaluation. We propose using a typical object detection measure (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010; Everingham, Eslami, Van Gool, Williams, Winn, & Zisserman, 2015), the ratio of the two matched events’ intersection to their union (often referred to as “intersection over union”, or IoU). If a ground truth event is labeled as a “miss”, its corresponding “match IoU” is set to 0. This way, the average “match IoU” is influenced both by the number of detected and missed events, and by the quality of correctly identified events.

We report this statistic for all event types in the ground truth data, in addition to episode-level F1 scores of Hooge et al. (2017) as well as sample-level F1 scores (for brevity, F1 scores are used instead of individual statistics such as sensitivity or specificity; this metric represents a balanced combination of precision and recall—their harmonic mean), for both GazeCom and MN-RA-data.

Another idea, which we adapt from object detection research, is only registering a "hit" when a certain IoU threshold is reached (Ren, He, Girshick, and Sun, 2015), thus avoiding the low-intersection matches potentially skewing the evaluation. The threshold that is often employed is 0.5 (Everingham et al., 2010). In the case of one-dimensional events, this threshold also gains interpretability: This is the lowest threshold that ensures that no two detected events can be candidate matches for a single ground truth event. Additionally, if two events have the same duration, their relative shift can be no more than one-third of their duration. For GazeCom, we further evaluate the algorithms at different levels of the IoU threshold used for event matching.

We also compute basic statistics over the detected eye-movement episodes, which we compare to those of the ground truth. Among those are the average duration (in milliseconds) and amplitude (in degrees of visual angle) of all detected fixation, saccade, and smooth pursuit episodes. Even though fixations are not traditionally characterized by their amplitude, it reflects certain properties of fixation detection: For instance, where does a fixation end and a saccade begin? While this choice has relatively little bearing on saccade amplitudes, it might significantly affect the amplitudes of fixations.

Data set-specific settings

For MN-RA-data, we focused only on our best (according to GazeCom evaluation) model.

Since MN-RA-data were recorded at 500 Hz (compared to 250 Hz for GazeCom), we simply doubled the sample-level intervals for feature extraction, which preserves the temporal scales of the respective features (as described in the "Features for classification" above). We also used a model that classifies 257-sample windows at once (our largest-context model). This way, the temporal context at 500 Hz is approximately equivalent to that of 129 samples at 250 Hz, which was used for the majority of GazeCom experiments. Notably, the model used for MN-RA-data processing was trained on 250-Hz recordings and tested on the ones with double the sampling frequency. Our estimate of the model's generalization ability is, therefore, conservative.

Due to cross-validation training on GazeCom, and in order to maximize the amount of pursuit examples in the training data, we predict labels in MN-RA-data using a model trained on all GazeCom clips except one without smooth pursuit in its ground truth annotation ("bridge_1").

Andersson et al. (2017) ignore smooth pursuit detection in most of their quantitative evaluation (while separating it from fixations is a challenging problem on its own), and focus on postsaccadic oscillations instead (which our algorithm does not label), so we cannot compare with the reported results directly. However, on the MN-RA-data, we additionally followed the evaluation strategies of Andersson et al. (2017).

In order to compare our approach to the state-of-the-art performances on MN-RA-data that were reported in Andersson et al. (2017), we computed the Cohen's kappa statistic (for each major eye-movement class separately).

Cohen's kappa κ for two binary sets of labels (e.g., A and B) can be computed via the observed proportion of samples, where A and B agree on either accepting or rejecting the considered eye-movement label, p_{obs}, and the chance probability of agreement. The latter can be expressed through the proportions of samples, where each of A and B has accepted and rejected the label. We denote those as p_+^A, p_-^A, p_+^B, and p_-^B, respectively. In this case,

$$p_{chance} = p_+^A * p_+^B + p_-^A * p_-^B, \tag{1}$$

$$\kappa(A, B) = \frac{p_{obs} - p_{chance}}{1 - p_{chance}}. \tag{2}$$

Cohen's kappa can assume values from [−1; 1], higher score is better. We also considered the overall sample-level error rate (i.e., proportion of samples where the models disagree with the human rater, when all six ground truth label classes are taken into account). For this, we consider all "noise" labels assigned by our algorithm as blink labels, as blinks were the primary cause of "noise" samples in the GazeCom ground truth. It has to be noted that all sample-level metrics are, to some extent, volatile with respect to small perturbations in the data—changes in proportions of class labels, almost imperceptible relative shifts, etc. We would, therefore, recommend using event-level scores instead.

Baselines

For both data sets, we ran several literature models as baselines, to give this work's evaluation more context.

These were: Agtzidis et al. (2016b) (implementation by Startsev et al. (2016)), Larsson et al. (2015) (re-implemented by Startsev et al. (2016)), Dorr et al. (2010) (the authors' implementation), Berg et al. (2009) (toolbox implementation Walther & Koch, 2006), I-VMP, I-VDT, I-VVT, I-KF, I-HMM, I-VT, I-MST, and I-DT (all as implemented by Komogortsev (2014), with fixation interval merging enabled). For their brief descriptions, see "Automatic detection".

Since several of the baselines (Dorr et al., 2010; Agtzidis et al., 2016b, the used implementation of Larsson et al., 2015) were either developed in connection with or optimized for GazeCom, we performed grid search optimization (with respect to the average of all sample- and event-wise F1 scores, as reported in Table 2) of the parameters of those algorithms in Komogortsev (2014) that detect smooth pursuit: I-VDT, I-VVT, and I-VMP. The ranges and the parameters of the grid search can be seen in Fig. 2. Overall, the best parameter set for I-VDT was $80\,^{\circ}/s$ for the speed threshold and $0.7\,^{\circ}$ for the dispersion threshold. For I-VVT, the low speed threshold of $80\,^{\circ}/s$ and the high threshold of $90\,^{\circ}/s$ were chosen. For I-VMP, the high speed threshold parameter was fixed to the same value as in the best parameter combination of I-VVT ($90\,^{\circ}/s$), and the window duration and the "magnitude of motion" threshold were set to 400 ms and 0.6, respectively.

It is an interesting outcome that, when trying to optimize the scores, half of which depend on events, I-VVT abandons pursuit detection (by setting very high speed thresholds) in favor of better-quality saccade and fixation identification. If optimization with respect to sample-level scores only is performed, this behavior is not observed. This indicates that simple speed thresholding is not sufficient for reasonable pursuit episode segmentation. We have, therefore, tried different speed thresholds for I-VMP, but $90\,^{\circ}/s$ was still the best option.

We have to note that taking the best set of parameters selected via an exhaustive search on the entire test set is prone to overfitting, so the reported performance figures for these baseline methods should be treated as optimistic estimates.

Since the fixation detection step of Dorr et al. (2010) targeted avoiding smooth pursuit, we treat missing labels (as long as the respective samples were confidently tracked) as pursuit for this algorithm. We also adapted the parameters of I-MST to the sampling frequency of the data set (for both data sets).

Just as Andersson et al. (2017), we did not re-train any of the models before testing them on the MN-RA-data.

For the model of Agtzidis et al. (2016a), however, we had to set the density threshold ($minPts$), which is a parameter of its clustering algorithm. This value should be set proportionally to the number of observers, and the sampling frequency (Startsev et al., 2016). If was, therefore, set to $160 * N_{observers}/46.9 * 500/250$, where $N_{observers}$ is the number of recordings for each clip (i.e., four for "dolphin", six for "BergoDalbana", and eight for "triple_jump"). GazeCom has an average of 46.9 observers per clip, and is recorded at 250 Hz. MN-RA-data, as mentioned already, consists of recordings at 500 Hz. The resulting $minPts$ values were 28, 40, and 54, respectively.

For both data sets, we additionally implemented a random baseline model, which assigns one of the three major eye-movement labels according to their frequency in the ground truth data.

Results and discussion

Cross-validation procedure selection

We first address the cross-validation type selection. We considered two modes, leave-one-video-out (LOVO) and leave-n-observers-out (LnOO, $n = 3$). If the two cross-validation procedures were identical in terms of the danger of overfitting, we would expect very similar quantitative results. If one is more prone to overfitting behavior than the other, its results would be consistently higher. In this part of the evaluation, therefore, we are, somewhat counterintuitively, looking for a validation procedure with *lower* scores.

We conducted several experiments to determine the influence of the cross-validation procedure on the performance

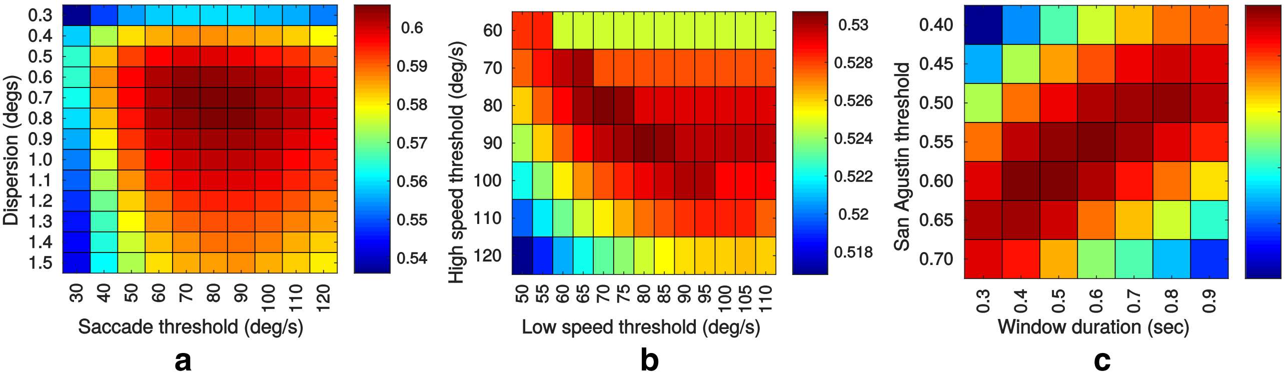


Fig. 2 Grid search average F1 scores on GazeCom for I-VDT (**2a**), I-VVT (**2b**), and I-VMP (**2c**). Default parameters were (ordered as (x, y) pairs): I-VDT – (70 °/s, 1.35 °), I-VVT – (20 °/s, 70 °/s), I-VMP – (0.5 s, 0.1). These are not optimal for our data set

Table 1 Experiment on the choice of a suitable cross-validation technique for our 1D CNN-BLSTM model with speed and direction features and a context size of 129 samples (equivalent to ca. 0.5 s at 250 Hz)

Metric	LOVO	LnOO
Fixations sample F1	0.937	0.939
Saccade sample F1	0.892	0.892
Pursuit sample F1	0.680	**0.706**
Fixations episode F1	0.888	0.887
Saccade episode F1	0.944	0.946
Pursuit episode F1	0.585	0.583
Fixations episode F1 ($IoU >= 0.5$)	0.854	0.855
Saccade episode F1 ($IoU >= 0.5$)	0.922	0.922
Pursuit episode F1 ($IoU >= 0.5$)	0.456	**0.466**
Fixations episode IoU	0.902	0.905
Saccade episode IoU	0.858	0.857
Pursuit episode IoU	0.521	**0.549**

LOVO refers to the leave-one-video-out approach, LnOO – to leave-n-observers-out. Both methods split the data in 18 non-overlapping groups of recordings in our case (18 videos in the data set, 18 groups of three observers each). Differences no less than 0.01 are **boldified**. This suggests that LOVO provides a more conservative estimate, compared to LnOO

estimates (while keeping the rest of the training and testing parameters fixed), all of which revealed the same pattern: While being comparable in terms of fixation and saccade detection, these strategies differ consistently and noticeably for smooth pursuit detection (see the results of one of these experiments in Table 1).

LnOO tends to overestimate the performance of the models, yielding higher scores on most of the metrics. We conclude that LOVO is less prone to overfitting and conduct the rest of the experiments using this technique. We note that overfitting seems to affect the detection of the stimulus characteristics-dependent eye-movement type—smooth pursuit—the most. For stimuli that only induce fixations and saccades, the concern of choosing an appropriate cross-validation technique is alleviated.

We conclude that excluding the tested stimulus (video clip, in this case) must be preferred to excluding the tested observer(s), if some form of cross-validation has to be employed, especially if the evaluation involves highly stimulus-driven eye-movement classes (e.g., smooth pursuit).

GazeCom results overview

An overview of all the evaluation results on the full GazeCom data set is presented in Table 2. It reports the models' performance on the three eye movement classes (fixations, saccades, and pursuits) for both sample- and event-level detection. Table 3 additionally provides the IoU values for all the tested algorithms. Bold numbers mark best performance in each category.

Most of our BLSTM models were trained with the context window of 129 samples (ca. 0.5 s) at the output layer, as it presented a reasonable trade-off between training time (ca. 3 s per epoch on NVIDIA 1080Ti GPU) and the saturation of the effect that context size had on performance.

Individual feature groups

Looking at individual feature sets for our model (raw xy-coordinates, speed, direction, and acceleration), we find that speed is the best individual predictor of eye-movement classes.

Acceleration alone, not surprisingly, fails to differentiate between fixations and smooth pursuit (the largest parts of almost 90% of the smooth pursuit episodes are covered by fixation labels), since both perfect fixation and perfect pursuit lack the acceleration component, excepting onset and offset stages of pursuits. Saccade detection performance is, however, impressive.

Interestingly, direction of movement provides a decent feature for eye-movement classification. One would expect that within fixations, gaze trace directions are distributed almost uniformly because of (isotropic) oculomotor and tracker noise. Within pursuits, its distribution should be pronouncedly peaked, corresponding to the direction of the pursuit, and even more so within saccades due to their much higher speeds. Figure 3 plots these distributions of directional features for each major eye-movement type. The directions were computed at a fixed temporal scale of 16 ms and normalized per-episode so that the overall direction is at 0. Unlike perfect fixations, which would be completely stationary, fixations in our data set contain small drifts (mean displacement amplitude during fixation of 0.56° of visual angle, median 0.45°), so the distribution in Fig. 3 is not uniform. Gaze direction features during saccades and pursuits predictably yield much narrower distribution shapes.

Using just the xy coordinates of gaze has an advantage of its simplicity and the absence of any pre-processing. However, according to our evaluation, the models that use either speed or direction features instead generally perform better, especially for smooth pursuit detection. Nevertheless, our model without any feature extraction already outperforms the vast majority of the literature approaches.

Feature combinations

Experimenting with several feature sets at once, we found that acceleration as an additional feature did not improve average detection performance, probably due to its inability to distinguish pursuit from fixation. The combination

Table 2 GazeCom evaluation results as F1 scores for *sample-level* and *episode-level* detection (sorted by the average of all columns)

		Sample-level F1			Event-level F1		
Model	average F1	Fixation	Saccade	SP	Fixation	Saccade	SP
1D CNN-BLSTM: speed + direction^{+}	**0.830**	**0.939**	**0.893**	**0.703**	0.898	**0.947**	**0.596**
1D CNN-BLSTM: speed + direction	0.821	0.937	0.892	0.680	0.888	0.944	0.585
1D CNN-BLSTM: speed	0.808	0.932	0.891	0.675	0.877	0.942	0.529
(Agtzidis et al., 2016b)	0.769	0.886	0.864	0.646	0.810	0.884	0.527
1D CNN-BLSTM: direction	0.769	0.919	0.802	0.621	0.862	0.911	0.499
1D CNN-BLSTM: xy	0.752	0.913	0.855	0.517	0.861	0.932	0.435
(Larsson et al., 2015)	0.730	0.912	0.861	0.459	0.873	0.884	0.392
I-VMP**	0.718	0.909	0.680	0.581	0.792	0.815	0.531
(Berg et al., 2009)	0.695	0.883	0.697	0.422	0.886	0.856	0.424
(Dorr et al., 2010)	0.680	0.919	0.829*	0.381	**0.902**	0.854	0.193*
1D CNN-BLSTM: acceleration	0.668	0.904	0.876	0.160	0.877	0.943	0.245
I-VDT**	0.606	0.882	0.676	0.321	0.823	0.781	0.152
I-KF	0.563	0.892	0.736	–	0.877	0.876	–
I-HMM	0.546	0.891	0.712	–	0.817	0.857	–
I-VVT**	0.531	0.890	0.686	0.000	0.778	0.816	0.013
I-VT	0.528	0.891	0.705	–	0.761	0.810	–
I-MST	0.497	0.875	0.570	–	0.767	0.773	–
I-DT	0.480	0.877	0.478	–	0.759	0.765	–
Random Baseline	0.201	0.750	0.105	0.114	0.098	0.121	0.020

CNN-BLSTM results are for the context window size of just over 0.5 s (129 samples), except where marked with $^{+}$ (1 s, 257 samples). The * signs mark the numbers where the label was assumed from context and not actually assigned by the algorithm – i.e. missing labels were imputed. Performance estimates for models marked with ** are optimistic (thresholds were optimized on the entire test set). In each column, the highest value is **boldified**

of direction and speed, however, showed a noticeable improvement over using them separately, and the results for these features we present in the tables.

We retrained the model that uses direction and speed features for a larger context size (257 samples, ca. 1 s). This model demonstrates the highest F1 scores (or within half a percent of the best score achieved by any model) for all eye-movement types in both evaluation settings. It outperforms the nearest literature approach by 2, 2.9, and 5.7% of the F1 score for fixations, saccades, and smooth pursuits, respectively. The gap is even wider (6.3 and 6.5% for saccades and smooth pursuit, respectively) for episode-level evaluation. Only for fixation episode detection, the Dorr et al. (2010) model performs slightly better (by 0.004). In terms of IoU values, our model improves the state-of-the-art scores by 0.04, 0.05, and 0.09 for fixations, saccades,

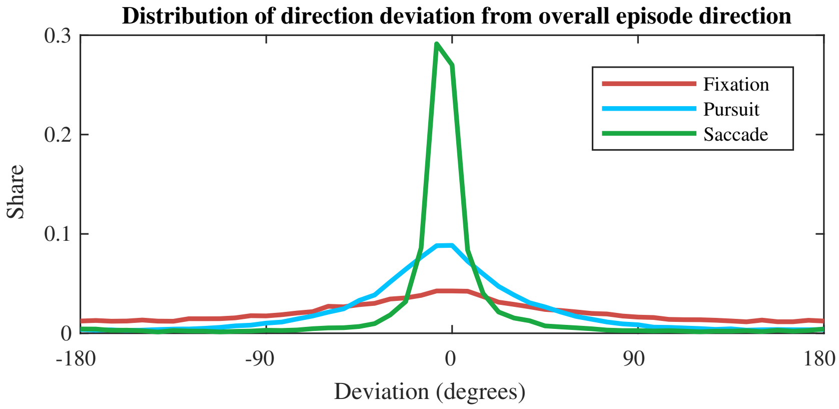


Fig. 3 Histogram of sample-level direction features, when normalized relative to the overall direction of each respective episode

Table 3 GazeCom evaluation results for *event-level* detection as intersection-over-union values (sorted by the average of all columns)

Model	Average IoU	Fixation IoU	Saccade IoU	SP IoU
1D CNN-BLSTM: speed + direction$^+$	**0.768**	**0.906**	**0.858**	0.541
1D CNN-BLSTM: speed	0.763	0.885	0.856	**0.547**
1D CNN-BLSTM: speed + direction	0.760	0.902	**0.858**	0.521
1D CNN-BLSTM: xy	0.665	0.880	0.801	0.313
(Dorr et al., 2010)	0.663	0.815	0.808	0.367*
(Agtzidis et al., 2016b)	0.663	0.742	0.799	0.448
1D CNN-BLSTM: direction	0.631	0.834	0.718	0.341
(Larsson et al., 2015)	0.625	0.789	0.809	0.277
I-VMP**	0.613	0.828	0.556	0.454
1D CNN-BLSTM: acceleration	0.606	**0.906**	0.834	0.077
I-VDT**	0.558	0.760	0.555	0.359
(Berg et al., 2009)	0.541	0.774	0.499	0.351
I-KF	0.504	0.842	0.671	–
I-HMM	0.501	0.870	0.633	–
I-VT	0.492	0.868	0.607	–
I-VVT**	0.477	0.863	0.567	0.000
I-MST	0.364	0.694	0.399	–
I-DT	0.313	0.592	0.347	–
Random baseline	0.055	0.077	0.071	0.017

CNN-BLSTM results are for the context window size of just over 0.5 s (129 samples), except where marked with $^+$ (1 s, 257 samples). The * signs mark the numbers where the label was assumed from context and not actually assigned by the algorithm – i.e. missing labels were imputed. Performance estimates for models marked with ** are optimistic (thresholds were optimized on the entire test set). In each column, the highest value is **boldified**

and pursuits, respectively, indicating the higher quality of the detected episodes across the board.

We also varied the IoU threshold that determines whether two episodes constitute a potential match, computing episode-level F1 scores every time (see Fig. 4). From this evaluation it can be seen that not only does our deep learning model outperform all literature models, but it maintains this advantage even when a stricter criterion for an event "hit" is considered (even though it was trained to optimize pure sample-level metrics). For fixations, while similar to the performance of Dorr et al. (2010) at lower IoU thresholds, our model is clearly better when it comes to higher thresholds. For saccades, it has to be noted that the labels of Dorr et al. (2010) were used as initialization for the manual annotators in order to obtain ground truth event labels for GazeCom. This results in a higher number of perfectly matching saccade episodes for Dorr et al. (2010) (as well as for Agtzidis et al. (2016b) and our implementation of Larsson et al. (2015), both of which use a very similar saccade detection procedure), when the manual raters decided not to change the borders of certain saccades.

As mentioned already, a threshold of 0.5 has its theoretical benefits (no two detected episodes can both be matches for a single ground truth event, some interpretability). Here, we can see practical advantages as well, thanks to the *Random Baseline* model. Due to the prevalence of fixation samples in the GazeCom data set, assigning random labels with the same distribution of classes results in many fixation events, which occasionally intersect with fixations in the ground truth. In the absence of any IoU thresholding (the threshold of 0 in Fig. 4), the F1 scores for fixations and saccades are around 10%. Only by the threshold level of 0.5 does the fixation event-wise F1 score for the *Random Baseline* reach zero.

Common eye-tracking measures

In order to directly compare the average properties of the detected events to those in the ground truth, we compute the mean durations and amplitudes for the episodes of the three major eye-movement types in our data. The results are presented in Table 4. For this part of the evaluation, we consider only our best model (referred to as *1D CNN-BLSTM (best)* in the tables), which uses speed and direction features at a context size of roughly 1 s. We compare it to all the baseline algorithms that consider smooth pursuit.

For both measures, our algorithm is ranked second, while providing average fixation and saccade amplitudes that are

Table 4 Average durations (in milliseconds) and amplitudes (in degrees of visual angle) of different event types, as labelled by manual annotators (first row) or algorithmic procedures

	Average event duration, ms				Average event amplitude, degrees			
Algorithm	Fixation	Saccade	SP	(rank) avg. Δ	Fixation	Saccade	SP	(rank) avg. Δ
Ground truth	315.2	41.5	405.2		0.56	6.84	2.38	
1D CNN-BLSTM (best)	281.1	38.4	217.0	(2) 75.2	**0.53**	**6.66**	1.44	(2) 0.38
(Agtzidis et al., 2016b)	229.5	40.1	244.6	(3) 82.6	0.40	7.19	**1.91**	(1) **0.33**
(Larsson et al., 2015)	**335.3**	41.0	**320.1**	(1) **35.2**	0.70	7.45	3.15	(3) 0.51
I-VMP**	256.3	20.8	217.8	(4) 89.0	0.52	6.03	1.64	(4) 0.53
(Berg et al., 2009)	282.6	66.0	164.3	(5) 99.3	0.51	8.26	1.20	(6) 0.88
(Dorr et al., 2010)	340.3	**41.7**	68.2*	(6) 120.7	**0.53**	7.41	1.09*	(5) 0.63
I-VDT**	284.7	19.0	45.7	(7) 137.5	0.47	5.41	0.49	(8) 1.13
I-VVT**	261.1	21.4	0.5	(8) 159.7	0.64	6.11	0.04	(7) 1.05

In each column, the value with the lowest absolute difference to the respective ground truth value is **boldified**. The averages of these absolute differences for event durations and amplitudes occupy the fifth and the last columns, respectively, along with the rank of each considered model (lower is better). The rows are sorted as the respective rows of our main evaluation table – Table 2. The * signs mark the numbers where the label was assumed from context and not actually assigned by the algorithm – i.e. missing labels were imputed. Performance estimates for models marked with ** are optimistic (thresholds were optimized on the entire test set)

the closest to the ground truth. We note that the approaches with the average duration and amplitude of events closest to the ground truth differ for the two measures (Larsson et al., 2015; Agtzidis et al., 2016b, respectively).

From this evaluation, we can conclude that our algorithm detects many small smooth pursuit episodes, resulting in comparatively low average smooth pursuit duration and amplitude. This is confirmed by the relatively higher event-level false positive count of our algorithm (3475, compared to 2207 for Larsson et al., 2015). Our model's drastically lower false negatives count (1192 vs. 2966), however, allows it to achieve much higher F1 score for pursuit event detection.

We also have to stress that simple averages do not provide a comprehensive insight into the performance of an eye-movement detection algorithm, but rather offer an intuitively interpretable, though not entirely reliable, measure of detected event quality. There is no matching performed here, the entire set of episodes of a certain type is averaged at once. This is why we recommend using either the temporal offsets of matched episode pairs as introduced by Hooge et al. (2017), or IoU averaging or thresholding, as we suggest in "Metrics". The latter allows for evaluating episode-level eye-movement detection performance at varying levels of match quality, which is assessed via a relatively easily interpretable IoU metric.

Context size matters

We also investigated the influence of the size of the context, where the network simultaneously assigns labels, on detection scores (see Fig. 5). We did this by running the cross-validation process at a range of context sizes, with five speed features defining the input space. We tested contexts of 1, 2 + 1, 4 + 1, 8 + 1, . . ., 256 + 1 samples. For the GazeCom sampling frequency, this corresponds to 4, 12, 20, 36 ms, . . ., 1028 ms. Training for larger context sizes was computationally impractical.

Context size had the biggest influence on smooth pursuit detection. For speed features, when the context window size was reduced from just over 1 s of gaze data to merely one sample being classified at a time, the F1 scores for fixation and saccade samples decreased (in terms of absolute values) only by 2.8 and 5.1%, respectively, whereas smooth pursuit sample detection performance plummeted (decreased by over 40%).

For all eye movements, however, there is a general positive impact of expanding the context of the analysis. This effect seems to reach saturation point by the 1 s mark, with absolute improvements in all detection F1 scores being not much higher than 1% (except for smooth pursuit episodes, which could potentially benefit from even larger context sizes).

At the largest context size, this model is better at detecting smooth pursuit (for both sample- and event-level problem settings) than any baseline smooth pursuit detector, including the multi-observer approach in Agtzidis et al. (2016b), which uses information from up to 50 observers at the same time, allowing for higher-level heuristics.

Generalizability

To test our model on additional independent data, we present the evaluation results of our best model (speed and direction used as features) with the context size ca. 0.5 s and all the literature models we tested on MN-RA-data set as sample- and event-level F1 scores (Table 5), as well as average IoU values (Table 6). This is the model with the largest context we trained, 257 samples. The duration in seconds is reduced due the doubling of the sampling frequency, compared to GazeCom.

Table 7 combines our evaluation results with the performances reported in Andersson et al. (2017) in the form of Cohen's kappa values and overall error rates for the MN-RA-data (for video stimuli). Evaluation results from Andersson et al. (2017) were included in the table if they represent the best performance with respect to at least one of the statistics that we include in this table. BIT refers to the algorithm in van der Lans, Wedel, and Pieters (2011), LNS—in Larsson et al. (2013).

For our model, performance on this data set is worse compared to GazeCom, but even human raters show substantial differences in labeling the "ground truth" (Hooge et al., 2017; Andersson et al., 2017).

Nevertheless, the average out-of-the-box performance of our algorithm compares favorably to the state of the art in terms of sample-level classification (see Table 5). In terms of episode-level evaluation, our model shows somewhat competitive F1 scores (Table 5), but makes up for it in the average intersection over union statistic, which accounts for both the number of correctly identified episodes and the quality of the match (see Table 6). While its error rate is similar to that of I-VMP, the F1 and IoU scores are, on average, higher for our model, and its Cohen's κ scores are consistently superior to I-VMP across the board.

Our algorithm's 34% error rate may still be unacceptable for many applications, but so could be the manual rater disagreement of 19% as well. Our algorithm further demonstrates the highest Cohen's Kappa score for smooth pursuit detection, and second highest for fixation detection.

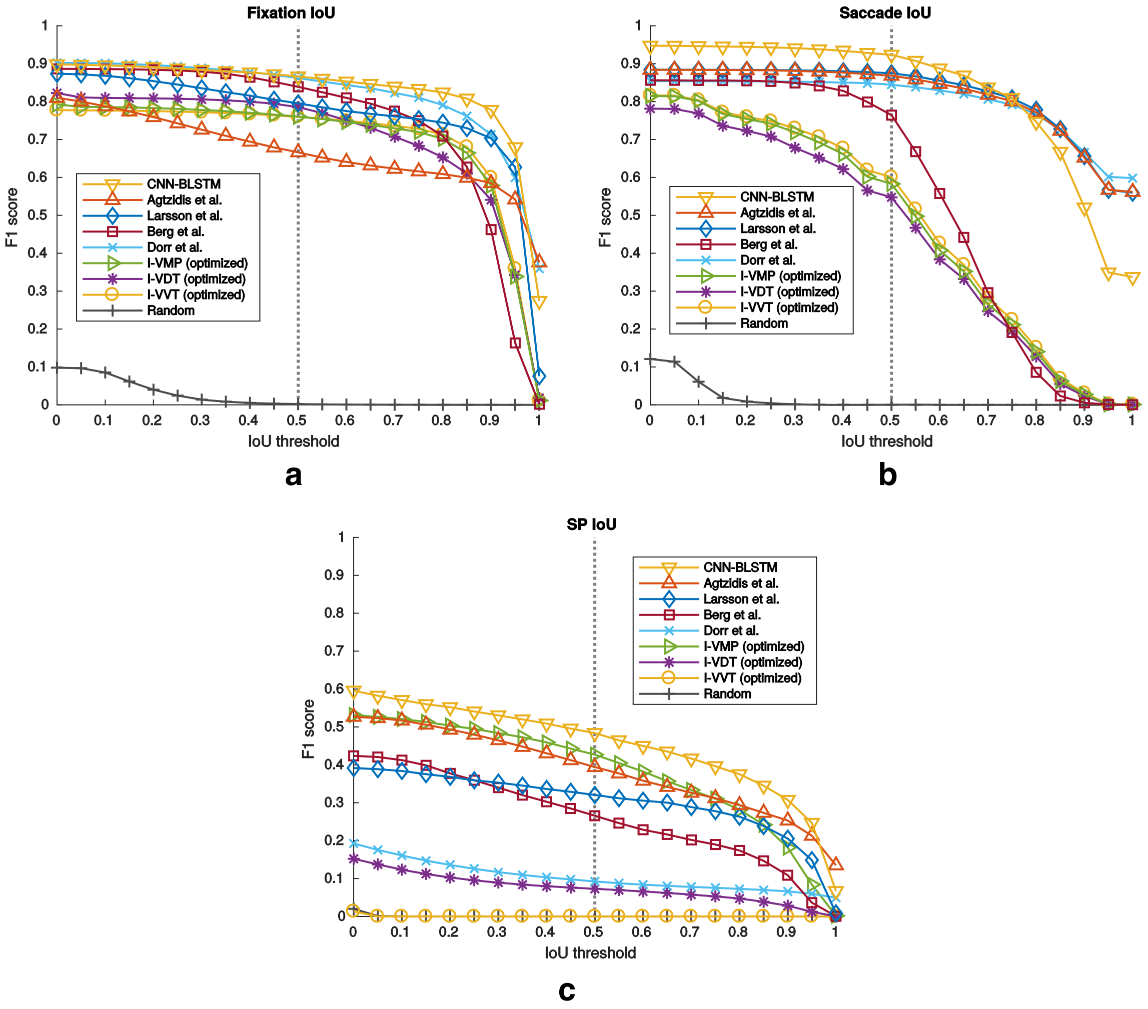


Fig. 4 Episode-level F1 scores at different IoU thresholds: At 0.0, the regular episode F1 score is computed; At 1.0, the episodes have to match exactly; The thresholds in-between represent increasing levels of episode match strictness. The *vertical dashed line* marks the threshold, which is typically used when considering IoU scores

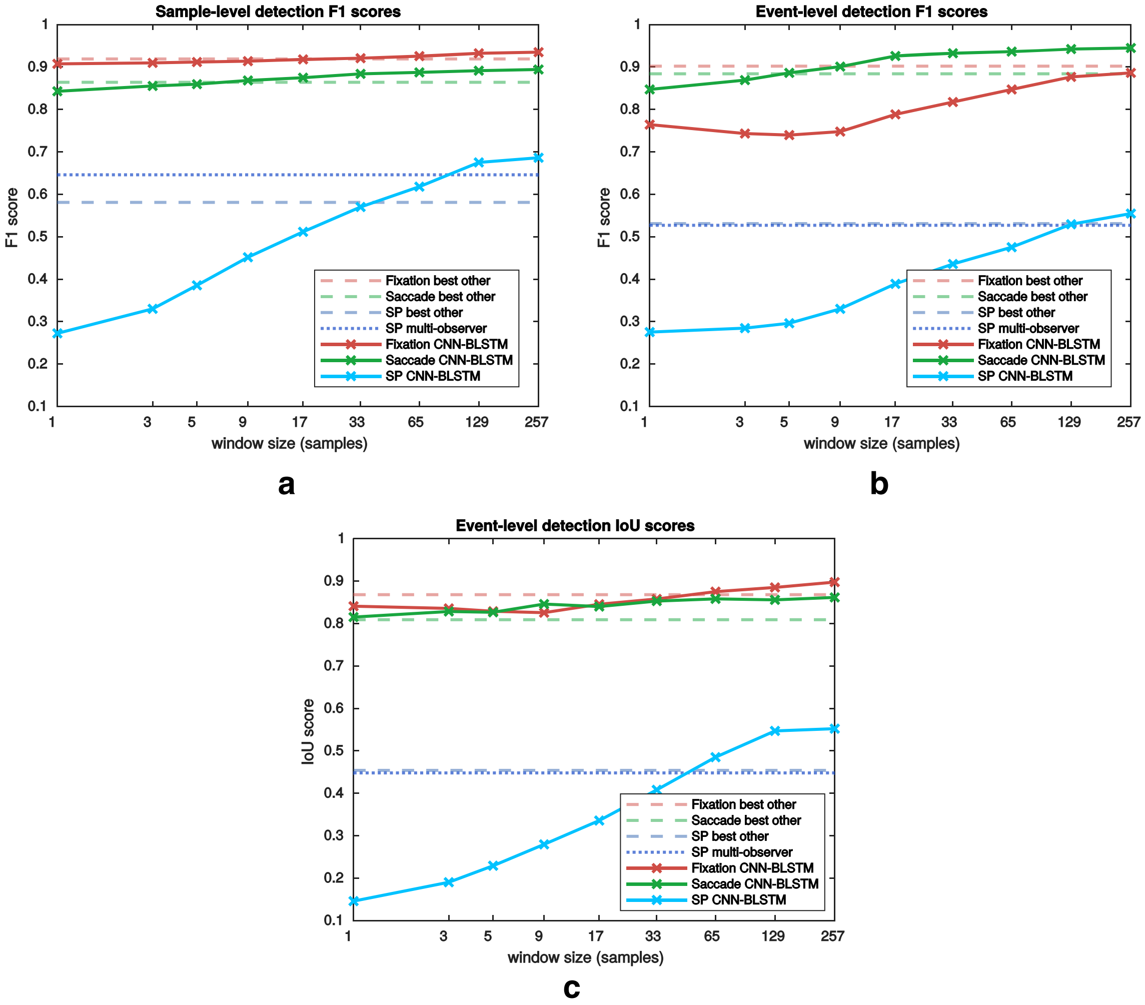


Fig. 5 Detection quality plotted against the context size (in samples at 250 Hz; log-scale) that the network classifies at once. *Dashed lines* represent individually chosen reference algorithms that perform best with respect to each eye movement. For both sample- and event-level F1 evaluation (**5a** and **5b**, respectively), fixation detection results of Dorr et al. (2010) are taken, for saccades – Startsev et al. (2016), for pursuits – I-VMP. For event-level IoU evaluation (**5c**), "best other" fixation detection IoU is taken from I-HMM, for saccades – Larsson et al. (2015), for pursuits – I-VMP. We separately display the smooth pursuit detection results of the multi-observer algorithm's toolbox (Startsev et al., 2016) (*the dotted line*), as it belongs to a different class of algorithms

The best saccade detection quality is achieved by LNS, which explicitly labels postsaccadic oscillations and thus increases saccade detection specificity.

For sample-level F1-score evaluation (Table 5), our model achieves second highest scores for fixation (with a very narrow margin) and pursuit detection, outperforming all competition in saccade detection.

Conclusions

We have proposed a deep learning system for eye-movement classification. Its overall performance surpasses all considered reference models on an independent small-scale data set. For the naturalistic larger-scale GazeCom, our approach outperforms the state of the art with respect to the three major eye-movement classes: fixations, saccades, and smooth pursuits. To the best of our knowledge, this is the first inherently temporal machine learning model for eye-movement event classification that includes smooth pursuit. Unlike (Agtzidis et al., 2016b), which implicitly uses full temporal context, and explicitly combines information across a multitude of observers, our model can be adapted for online detection (by re-training without using look-ahead features and preserving the LSTM states[4]). The classification time is kept short due to the low—for deep-learning models, at least—parameter count of the trained models. Furthermore, we introduced and analyzed a new

[4] For online detection, one would need to either use a unidirectional LSTM and process the samples as they are recorded, or assemble windows of samples that end with the latest available ones and process the full windows with a BLSTM model.

Table 5 MN-RA-data evaluation results as F1 scores for *sample-level* and *episode-level* detection (sorted by the average of all columns). CNN-BLSTM results here are for the window size of just over 0.5s (257 samples at 500 Hz). The * signs mark the numbers where the label was assumed from context and not actually assigned by the algorithm – i.e. missing labels were imputed. In each column, the highest value is **boldified**

		Sample-level F1			Event-level F1		
Model	average F1	Fixation	Saccade	SP	Fixation	Saccade	SP
(Agtzidis et al., 2016b)	**0.653**	**0.670**	0.699	0.638	0.455	0.860	**0.592**
1D CNN-BLSTM: speed + direction	0.650	0.667	**0.720**	0.663	0.550	0.826	0.475
(Larsson et al., 2015)	0.633	0.609	0.698	0.424	**0.741**	**0.871**	0.456
(Dorr et al., 2010)	0.630	0.614	0.691	0.446*	0.710	0.841	0.476*
I-VMP	0.572	0.593	0.699	**0.739**	0.455	0.564	0.385
(Berg et al., 2009)	0.533	0.609	0.625	0.176	0.683	0.730	0.374
I-VDT	0.474	0.595	0.694	0.222	0.443	0.561	0.329
I-KF	0.444	0.578	0.643	–	0.639	0.806	–
I-HMM	0.421	0.577	0.711	–	0.535	0.702	–
I-DT	0.381	0.573	0.439	–	0.599	0.678	–
I-VT	0.375	0.575	0.701	–	0.412	0.560	–
I-VVT	0.365	0.573	0.701	0.242	0.067	0.560	0.046
I-MST	0.363	0.560	0.444	–	0.603	0.568	–
Random Baseline	0.180	0.387	0.051	0.535	0.023	0.066	0.017

Table 6 MN-RA-data evaluation results for *event-level* detection as intersection-over-union values (sorted by the average of all columns)

Model	Fixation ep. IoU	Saccade ep. IoU	SP ep. IoU
1D CNN-BLSTM: speed + direction	0.705	0.543	0.398
I-VMP	0.368	0.623	**0.619**
I-VDT	0.744	0.647	0.127
(Agtzidis et al., 2016b)	0.626	0.469	0.420
(Larsson et al., 2015)	0.754	0.514	0.215
I-VT	0.798	**0.665**	–
I-HMM	**0.81**	0.627	–
(Dorr et al., 2010)	0.646	0.461	0.284*
I-KF	0.785	0.543	–
(Berg et al., 2009)	0.706	0.353	0.075
I-DT	0.628	0.328	–
I-VVT	0.181	**0.665**	0.023
I-MST	0.438	0.208	–
Random baseline	0.024	0.038	0.016

BLSTM results here are for the window size of just over 0.5s (257 samples at 500 Hz). The * signs mark the numbers where the label was assumed from context and not actually assigned by the algorithm–i.e., missing labels were imputed. In each column, the highest value is **boldified**

Table 7 Cohen's kappa (higher is better) and overall error rates (lower is better) for the MN-RA-data set

Group	Error rate	Fixation κ	Saccade κ	SP κ
CoderMN	19%	0.83	0.94	0.83
CoderRA	19%	0.82	0.94	0.83
1D CNN-BLSTM: speed + direction	**34%**	0.41	0.70	**0.41**
I-VMP	**34%**	0.38	0.68	0.40
(Agtzidis et al., 2016b)	38%	**0.43**	0.68	0.40
(Dorr et al., 2010)	46%	0.25	0.67	0.20*
(Larsson et al., 2015)	47%	0.23	0.68	0.19
I-VDT	53%	0.16	0.67	0.09
Random Baseline	56%	0.00	0.00	0.00
(Berg et al., 2009)	57%	0.21	0.60	0.07
I-VVT	55%	0.14	0.68	0.02
I-HMM**	59%	0.13	0.71	–
I-VT**	59%	0.13	0.76	–
I-DT	60%	0.09	0.40	–
I-MST	61%	0.04	0.43	–
I-KF**	62%	0.14	0.59	–
BIT**	67%	0.14	0.00	–
LNS**	92%	0.00	**0.81**	–

BLSTM here uses speed and direction features and the context of ca. 0.5 s (257 samples at 500 Hz). The * signs mark the numbers where the label was assumed from context and not actually assigned by the algorithm—i.e., missing labels were imputed. The scores for the algorithms marked with the ** were taken directly from Andersson et al. (2017). The rows are sorted by their error rate. In each column, the best value a chieved by any algorithm is **boldified** (the first two rows correspond to manual annotators)

event-level evaluation protocol that considers the quality of the matched episodes through enforcing restrictions on the pair of events that constitute a match. Our experiments additionally highlight the importance of temporal context, especially for detecting smooth pursuit.

The code for our model and results for all evaluated algorithms are provided at http://www.michaeldorr.de/smoothpursuit.

Acknowledgements This research was supported by the Elite Network Bavaria, funded by the Bavarian State Ministry for Research and Education.

References

Agtzidis, I., Startsev, M., & Dorr, M. (2016a). In the pursuit of (ground) truth: A hand-labelling tool for eye movements recorded during dynamic scene viewing. In *2016 IEEE second workshop on eye tracking and visualization (ETVIS)* (pp. 65–68).

Agtzidis, I., Startsev, M., & Dorr, M. (2016b). Smooth pursuit detection based on multiple observers. In *Proceedings of the ninth biennial ACM symposium on eye tracking research & applications, ETRA '16* (pp. 303–306). New York: ACM.

Anantrasirichai, N., Gilchrist, I. D., & Bull, D. R. (2016). Fixation identification for low-sample-rate mobile eye trackers. In *2016 IEEE international conference on image processing (ICIP)* (pp. 3126–3130).

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

Behrens, F., MacKeben, M., & Schröder-Preikschat, W. (2010). An improved algorithm for automatic detection of saccades in eye movement data and for calculating saccade parameters. *Behavior Research Methods*, *42*(3), 701–708.

Berg, D. J., Boehnke, S. E., Marino, R. A., Munoz, D. P., & Itti, L. (2009). Free viewing of dynamic stimuli by humans and monkeys. *Journal of Vision*, *9*(5), 1–15.

Chollet, F., et al. (2015). Keras. https://github.com/keras-team/keras

Collewijn, H., & Tamminga, E. P. (1984). Human eye movements during voluntary pursuit of different target motions on different backgrounds. *The Journal of Physiology*, *351*(1), 217–250.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Dorr, M., Martinetz, T., Gegenfurtner, K. R., & Barth, E. (2010). Variability of eye movements when viewing dynamic natural scenes. *Journal of Vision*, *10*(10), 28–28.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The Pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, *111*(1), 98–136.

Goldberg, J. H., & Schryver, J. C. (1995). Eye-gaze-contingent control of the computer interface: Methodology and example for zoom detection. *Behavior Research Methods Instruments, & Computers*, *27*(3), 338–350.

Hasanpour, S. H., Rouhani, M., Fayyaz, M., & Sabokrou, M. (2016). Lets keep it simple, using simple architectures to outperform deeper and more complex architectures. CoRR, arXiv:1608.06037

Hooge, I. T. C., Niehorster, D. C., Nyström, M., Andersson, R., & Hessels, R. S. (2017). Is human classification by experienced untrained observers a gold standard in fixation detection? *Behavior Research Methods*.

Hoppe, S., & Bulling, A. (2016). End-to-end eye movement detection using convolutional neural networks. ArXiv e-prints.

Komogortsev, O. V. (2014). Eye movement classification software. http://cs.txstate.edu/ok11/emd_offline.html

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.

Komogortsev, O. V., & Karpov, A. (2013). Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behavior Research Methods*, *45*(1), 203–215.

Kyoung Ko, H., Snodderly, D. M., & Poletti, M. (2016). Eye movements between saccades: Measuring ocular drift and tremor. *Vision Research*, *122*, 93–104.

Land, M. F. (2006). Eye movements and the control of actions in everyday life. *Progress in Retinal and Eye Research*, *25*(3), 296–324.

Larsson, L., Nyström, M., Andersson, R., & Stridh, M. (2015). Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. *Biomedical Signal Processing and Control*, *18*, 145–152.

Larsson, L., Nyström, M., & Stridh, M. (2013). Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit. *IEEE Transactions on Biomedical Engineering*, *60*(9), 2484–2493.

Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., & Kautz, J. (2016). Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Nyström, M. (2015). Marcus Nyström — Humanities Lab, Lund University. http://www.humlab.lu.se/en/person/MarcusNystrom

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

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., & Garnett, R. (Eds.) *Advances in neural information processing systems 28* (pp. 91–99). Curran Associates, Inc.

Rottach, K. G., Zivotofsky, A. Z., Das, V. E., Averbuch-Heller, L., Discenna, A. O., Poonyathalang, A., & Leigh, R. J. (1996). Comparison of horizontal, vertical and diagonal smooth pursuit eye movements in normal human subjects. *Vision Research*, *36*(14), 2189–2195.

Salvucci, D. D., & Anderson, J. R. (1998). Tracing eye movement protocols with cognitive process models. In *Proceedings of the 20th annual conference of the cognitive science society* (pp. 923–928). Lawrence Erlbaum Associates Inc.

Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on eye tracking research & applications, ETRA '00* (pp. 71–78). New York: ACM.

San Agustin, J. (2010). *Off-the-shelf gaze interaction*. PhD thesis, IT-Universitetet i København.

Santini, T. (2016). Automatic identification of eye movements. http://ti.uni-tuebingen.de/Eye-Movements-Identification.1845.0.html

Santini, T., Fuhl, W., Kübler, T., & Kasneci, E. (2016). Bayesian identification of fixations, saccades, and smooth pursuits. In *Proceedings of the ninth biennial ACM symposium on eye tracking research & applications, ETRA '16* (pp. 163–170). New York: ACM.

Sauter, D., Martin, B. J., Di Renzo, N., & Vomscheid, C. (1991). Analysis of eye tracking movements using innovations generated by a Kalman filter. *Medical and Biological Engineering and Computing*, *29*(1), 63–69.

Startsev, M., Agtzidis, I., & Dorr, M. (2016). Smooth pursuit. http://michaeldorr.de/smoothpursuit/

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, *4*(2), 26–31.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In *2015 IEEE international conference on computer vision (ICCV)* (pp. 4489–4497).

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.

Vidal, M., Bulling, A., & Gellersen, H. (2012). Detection of smooth pursuits using eye movement shape features. In *Proceedings of the symposium on eye tracking research & applications, ETRA '12* (pp. 177–180). New York: ACM.

Walther, D., & Koch, C. (2006). Modeling attention to salient proto-objects. *Neural Networks*, *19*(9), 1395–1407.

Yarbus, A. L. (1967). *Eye movements during perception of moving objects* (pp. 159–170). Boston: Springer.

Zemblys, R., Niehorster, D. C., Komogortsev, O., & Holmqvist, K. (2017). Using machine learning to detect events in eye-tracking data. *Behavior Research Methods*, *50*(1), 160–181. https://doi.org/10.3758/s13428-017-0860-3

D

A Novel Gaze Event Detection Metric

In this work, we have for the first time proposed a way to systematically objectively analyse evaluation metrics for eye movement detection. To this end, we suggested using "baseline" eye movement classifiers, which would be designed in such a way that their performance should be unsatisfactory (*e.g.* randomly assigning the labels, with some plausibility constraints). If a certain metric would, for some baselines, yield a score that is superior to that of a well-established detection algorithm, when tested against the ground truth in a diverse and high-quality data set, the considered metric is likely not adequately reflecting the actual performance of the classifiers in an intuitive way.

Having analysed the evaluation strategies in the literature (*e.g.* in [30, 52, 3*]), we discovered their biases and flaws. Some of the findings were to be expected – *e.g.* as the classes are not balanced (the vast majority of samples would be typically labelled as fixations), sample-level F1 scores for the most frequent class would be relatively high, even if the labels were assigned randomly. Other findings, on the other hand, were more unexpected – for instance, while quantifying the differences between the event sequences in true and automatically labelled data via Levenshtein distance (the number of simple sequence-editing operations required to transform one sequence into the other) would be very intuitive [52], it turned out that this metric would rank six out of the seven tested algorithms for smooth pursuit (SP) detection below a random baseline.

Having determined the strong and weak properties of existing metrics, we proposed a new approach that combined the idea of applying Cohen's kappa analysis to event-level evaluation [52] with variable event matching strictness [3*]. We re-formulated the computation procedure for this event-level kappa score used in [52], aligning the resulting metric with its intuitive definition (the advantage of the algorithm-assigned event detections over randomly allocating those). Our proposed metric – adjusted event-level Cohen's kappa – yielded nearly zero scores for all the developed baselines.

My personal contributions consist of (i) designing, implementing, and evaluating the baselines for eye movement classification (except for the implementation of the "video-only" baseline) and the proposed novel metric, (ii) testing our pipeline on three data sets in addition to the mainly used GazeCom [69], and (iii) writing the manuscript.

A Novel Gaze Event Detection Metric That Is Not Fooled by Gaze-independent Baselines

Mikhail Startsev
Technical University of Munich
Munich, Germany
mikhail.startsev@tum.de

Stefan Göb
Technical University of Munich
Munich, Germany
stefan.goeb@tum.de

Michael Dorr
Technical University of Munich
Munich, Germany
michael.dorr@tum.de

ABSTRACT

Eye movement classification algorithms are typically evaluated either in isolation (in terms of absolute values of some performance statistic), or in comparison to previously introduced approaches. In contrast to this, we first introduce and thoroughly evaluate a set of both random and above-chance baselines that are completely independent of the eye tracking signal recorded for each considered individual observer. Surprisingly, our baselines often show performance that is either comparable to, or even exceeds the scores of some established eye movement classification approaches, for smooth pursuit detection in particular. In these cases, it may be that (i) algorithm performance is poor, (ii) the data set is overly simplistic with little inter-subject variability of the eye movements, or, alternatively, (iii) the currently used evaluation metrics are inappropriate. Based on these observations, we discuss the level of stimulus dependency of the eye movements in four different data sets. Finally, we propose a novel measure of agreement between true and assigned eye movement events, which, unlike existing metrics, is able to reveal the expected performance gap between the baselines and dedicated algorithms.

CCS CONCEPTS

• **Applied computing → Psychology**.

KEYWORDS

eye movement classification, event detection, random baseline

ACM Reference Format:
Mikhail Startsev, Stefan Göb, and Michael Dorr. 2019. A Novel Gaze Event Detection Metric That Is Not Fooled by Gaze-independent Baselines. In *2019 Symposium on Eye Tracking Research and Applications (ETRA '19), June 25–28, 2019, Denver, CO, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3314111.3319836

1 INTRODUCTION

Eye movement classification or detection is a popular analysis tool for eye tracking sessions. Often, the algorithms that are built into the eye trackers are utilised, but developing specialised [Agtzidis et al. 2016b; Larsson et al. 2015, 2013; Otero-Millan et al. 2014; Steil et al. 2018; Vidal et al. 2012] or generic [Berg et al. 2009; Hoppe and Bulling 2016; Kasneci et al. 2015; Komogortsev and Karpov 2013; Nyström and Holmqvist 2010; Santini et al. 2016; Startsev et al. 2018; Zemblys et al. 2018a,b] eye movement classification approaches is a large and rapidly developing field of research in its own right, which has recently seen the advent of deep learning models [Startsev et al. 2018; Zemblys et al. 2018a].

Despite (or perhaps because of) the multitude of existing eye movement classification methods (for a recent review, see [Andersson et al. 2017]) and toolboxes [Komogortsev 2014; Santini 2016; Startsev et al. 2016; Zemblys et al. 2018a], the field has a variety of factors hindering its development, starting from the researchers disagreeing [Hessels et al. 2018] on just what the most basic eye movement types – fixations and saccades – should represent, to the evergrowing number of metrics and evaluation procedures that are to be employed to find “the best” model [Andersson et al. 2017; Hooge et al. 2017; Komogortsev et al. 2010; Komogortsev and Karpov 2013; Startsev et al. 2018], to the question of which eye movement classes should be detected at all (some consider post-saccadic oscillations [Larsson et al. 2013; Zemblys et al. 2018a,b], some – microsaccades [Engbert and Kliegl 2003; Engbert and Mergenthaler 2006; Otero-Millan et al. 2014]), to a lack of widely used data sets or baselines.

The latter issue often leads to researchers either reporting performance statistics of their model without a comparison to other algorithms [Behrens et al. 2010; Vidal et al. 2012], or picking relatively weak but easily implementable reference models [Hoppe and Bulling 2016; Kinsman et al. 2012; Nyström and Holmqvist 2010; Steil et al. 2018] such as I-VT or I-DT [Salvucci and Goldberg 2000]. [Startsev et al. 2018] consider random independent assignment of eye movement labels, but this clearly is a very weak baseline, especially when event-level evaluation strategies [Hooge et al. 2017; Startsev et al. 2018; Zemblys et al. 2018a] are employed.

In the field of saliency prediction [Judd et al. 2009] – i.e. attempting to computationally model human visual attention – which is closely related to eye movement research, a substantial number of baselines are used in order to give context to the models' performance [Judd et al. 2012]. Not only do these baselines provide a set of performance levels, to which the trained models can be compared, but they also reveal certain characteristics of the ground truth data, and therefore the respective data sets in general. For example, the performance of the “centre baseline” is directly tied to the amount of centre bias in the ground truth saliency maps. The fact that this baseline performs better than a large number of saliency models [Judd et al. 2012] calls into question how much of the claim that saliency models reflect the brain is really true. The performance of the “one human” baseline, where the gaze locations

ETRA '19, June 25–28, 2019, Denver, CO, USA

ACM ISBN 978-1-4503-6709-7/19/06…$15.00
https://doi.org/10.1145/3314111.3319836

Table 1: List of proposed baselines, with * marking those selected for comparison to the dedicated eye movement classifiers.

Baseline name	Description
Random independent samples	Individual sample-level labels are generated with "correct" a priori probabilities
Random samples sequences	Individual sample-level labels are generated with "correct" a priori and transition probabilities
Random independent events	Plausible-duration events are generated with "correct" a priori probabilities
Random event sequences*	Plausible-duration events are generated with "correct" a priori and transition probabilities
Majority label	All observers' gaze samples for each video frame are labelled with the same (most frequent) class
Majority label + sub-segmentation*	Additionally split up overlong fixations and pursuits similar to *random event sequences*
Video-only	A model predicting the *majority* baseline labels (pursuit vs. fixation) based on the video frames
Video-only + sub-segmentation*	Additionally split up overlong fixations and pursuits similar to *random event sequences*
Inter-observer*	Label each observer's gaze samples with the ground truth labels of another observer

of a single human observer are compared to those of the whole set of subjects relates to the level of congruency between individuals.

Some of the saliency baselines are either inapplicable (e.g. the "centre baseline"), or very difficult and costly to estimate in the context of eye movement classification (e.g. the "infinite humans" baseline [Judd et al. 2012] would be equivalent to getting a very large pool of annotators to label the recordings for all present eye movements, then taking their consensus labels as predictions of this baseline). Still, eye movement research can benefit from the works on saliency prediction, especially since the latter profits from the former by better distinguishing between the eye movement types such as fixations and pursuits [Startsev and Dorr 2018]. Similar to the saliency baselines, we here want to quantify how much useful information do the existing algorithms extract from the gaze data, and what can be achieved without directly analysing that data.

In this paper, we proposed several baselines for eye movement classification. We draw inspiration from both the saliency baselines and the assumptions and domain knowledge that human experts implicitly incorporate in their annotations. Our set of baselines includes (i) several random approaches, generating labels either on the level of samples or whole events, with or without modelling the transitions between label types, (ii) two baselines that are related to the inter-observer similarity of the simultaneously performed eye movements, as well as (iii) a purely video-content based eye movement prediction method, which we implemented with the help of hand-crafted features and a small deep architecture. Approaches (i) and (iii) can be used regardless of how many observers viewed the stimuli and whether their recordings are synchronised or not.

Just as the saliency prediction baselines, these methods we introduced are not meant to be used in practice in place of dedicated eye movement classifiers. They are intended exclusively (i) to reveal certain properties of the eye tracking data sets (the scores of the baselines depend on the characteristics of the data sets such as variability in the eye movements performed by different subjects simultaneously, etc.), and (ii) to be compared to the dedicated algorithms in order to provide context for their performance (e.g. how much does algorithmic classification gain over randomly labelling events?). We report the statistics on (i) for several data sets, reflecting on their complexity and diversity, while testing (ii) by comparing the performance of the developed baselines to both classical and state-of-the-art algorithms for eye movement classification with the help of 2 overall and 6 per-class literature metrics.

As a result, we concluded that current metrics are biased. We therefore proposed a new metric κ_{adjusted} for evaluating algorithmically detected events in the eye tracking recordings, which adjusts the event-level Cohen's kappa computation for the observed biases.

2 METHODS

In this section we describe the baselines we proposed in order to test the robustness of existing evaluation strategies. The overview of the baselines can be found in Table 1. The individual sub-sections below will first introduce our main data set, followed by the description and motivation of all proposed baseline methods in detail. The novel metric we developed is described separately in Section 5, as it is motivated by the observations on the existing literature evaluation strategies in the context of our baseline methods.

2.1 Data

For the most part, we developed and evaluated our baselines on the GazeCom data set [Dorr et al. 2010] of eye tracking recordings with 18 dynamic natural scenes presented as stimuli. This data set choice was motivated by its size (over 4.5 h in total; ca. 39,000 fixations and saccades, 5000 pursuits) and the publicly available [Startsev et al. 2016] manual annotations. Since the data set is relatively large, we will not have to run our random baselines several times over to achieve a good estimate of their average performance. Another property that interested us for the data set selection was the presence of smooth pursuit in the stimulus, since we were specifically targeting to distinguish fixation- and pursuit-dominated video frames with our video-only baseline.

2.2 Random Baselines

The simplest baseline that we used randomly assigns labels to the individual gaze samples, but with probabilities that are proportional to the number of respective samples in real eye-movement data (we consider fixation, saccade, and smooth pursuit samples). This *random independent samples* baseline was used in [Startsev et al. 2018] as well in order to motivate a stricter event-level evaluation procedure. Here, this was the least sophisticated baseline, so further on we will test and discuss this and other evaluation techniques.

Recent publications demonstrate a noticeable trend in the field towards testing eye movement classification with event-level metrics [Hooge et al. 2017; Startsev et al. 2018; Zemblys et al. 2018a]. For this class of measures, randomly assigning independent samples will drastically overestimate the number of eye movement episodes, yielding event sequences that very poorly match the "ground truth" on any metric. To remedy this, we wanted to explicitly leverage our (average) expectations of eye movement durations, and thus simulate an expert assigning event classes with a knowledge of their typical characteristics, but completely oblivious of the eye tracking signal. To this end, we stored all the occurring durations separately for all the considered eye movement classes in our data set to generate plausible events by directly sampling this set of realistically occurring durations (sampling from a fitted Gaussian or algorithmically detected event duration distributions delivered very similar results). We could then randomly generate events of naturally-distributed durations one after another until each individual recording has been fully labelled. This is the *random independent events* baseline. The a priori probability of each event type was computed based on the frequency of such events in the ground truth.

Independently generating both samples and events means that the resulting sequences are not controlled for their plausibility. For *random independent samples* this leads to the distinct possibility of generating "events" that last one sample only. For *random independent events*, on the contrary, the problem lies in the possibility of generating e.g. two events of the same class one after the other, meaning that they will be merged into one, much larger, episode, thus skewing the distribution of event durations. To solve both of these issues, we proposed generating sequences of both samples and events in a simple Markov chain process. To this end, we also computed the conditional transition probabilities from a sample or event of one class to a sample or event of another, respectively: $p(EM_{\text{next}} \mid EM_{\text{current}})$, where EM stands for the eye movement class label of a single sample or a whole event, depending on the setting. We then generated the eye movement type only of the first sample or event in each labelled recording with the respective a priori distribution, and proceeded to generate all the subsequent sample or event labels using the conditional transition probabilities. We refer to this approach as generating *random sample sequences* or *random event sequences*, respectively. This sequence generation approach was partly inspired by the saccadic models (e.g. [Meur and Liu 2015]) that generate scanpaths in a Markov process that is conditioned by the oculomotor biases of human observers, which are reflected in the distributions of saccade amplitudes and directions.

We consistently found that generating events results in better overall performance than generating samples, and utilising sequences of samples or events is preferable to generating them independently, so for the purpose of brevity we will only report the evaluation results for the *random event sequences* baseline.

2.3 Majority-label Baselines

Here we attempted to achieve the best performance without differentiating between the subjects or explicitly taking gaze movement data into account. This baseline "learns" from the annotated data: For every stimulus video clip, it takes the majority label assigned to any of the observers' (min. 37, max. 52 in GazeCom) gaze samples (at 250 Hz) that occur during each video frame (at 29.97 Hz). In case of ties, the label with the lower numerical value (in the labelling scheme of [Agtzidis et al. 2016a]) was preferred. This majority label is then assigned to all of the considered samples for this frame, regardless of the particular recording or gaze movement statistics.

This baseline, of course, works at the level of samples, so the generated events are not plausibly distributed: E.g. we expected an overwhelming number of samples to be labelled as part of a fixation with this approach. However, it has been noted before that the congruency between subjects' gaze traces tends to increase when a small number of moving objects are present in the scene [Dorr et al. 2010], so the gaze samples (of different observers) that belong to a pursuit of the same object will likely be clustered in space and time [Agtzidis et al. 2016b]. [Mital et al. 2011] also noted that scene motion is highly predictive of the clustering of the viewers' points of regard. We therefore also anticipated a non-negligible amount of smooth pursuit assigned by this baseline.

2.3.1 Large Event Sub-segmentation.

As explained above, the fixations (and, potentially, pursuits) typically produced by this baseline will be exceedingly long (e.g. average "fixation" duration of ca. 3 s). This is a significant downside of this method, especially in the context of event-level evaluation. We therefore split large episodes (more than one standard deviation above the respective mean, in our implementation) into smaller, more plausible ones. We did this in the same way as described for the *random event sequences* baseline above, except for only using fixation and saccade events when sub-segmenting fixations, and only smooth pursuit and saccade events when processing overlong pursuit episodes.

We found that this post-processing step slightly lowered the sample-level statistics, but increased all tested event-level measures, sometimes more than threefold. Consequently, we only report the results for the sub-segmented version of the *majority-label* baseline.

2.4 Video-only Baseline

To build on the *majority-label* baseline, which still uses the manual annotations of the subjects, whose recordings it classifies, albeit in an accumulated, "anonymised" fashion, we trained a machine learning model to identify the frames in the video sequence that are more likely to elicit smooth pursuits vs. those that are more likely to elicit non-pursuit eye movements (we assumed those are fixations, which is overwhelmingly the case in GazeCom anyway). We stress that this is a proof-of-concept model, and its generalisability to other data sets (as a pre-trained predictor or via re-training, especially for videos from head-mounted cameras) needs to be tested further.

For this classification, we used eight hand-crafted features, all computed for the median-filtered (11 × 11 for 720 × 1280 frames) optical flow between the current video frame and the previous one. We typically observed one of three consistently occurring patterns during the frames that have smooth pursuit as the most frequent eye movement type: a moving object has recently entered the frame (e.g. see Figure 1a), a singular moving object is traversing the otherwise relatively static scene (e.g. see Figure 1b), or the camera itself is moving (mostly in just one clip in GazeCom, e.g. see Figure 1c).

To create a feature that tests for an object that has recently entered the frame, we first found the pixel with the highest magnitude of the optical flow. With the assumption that this was the "main"

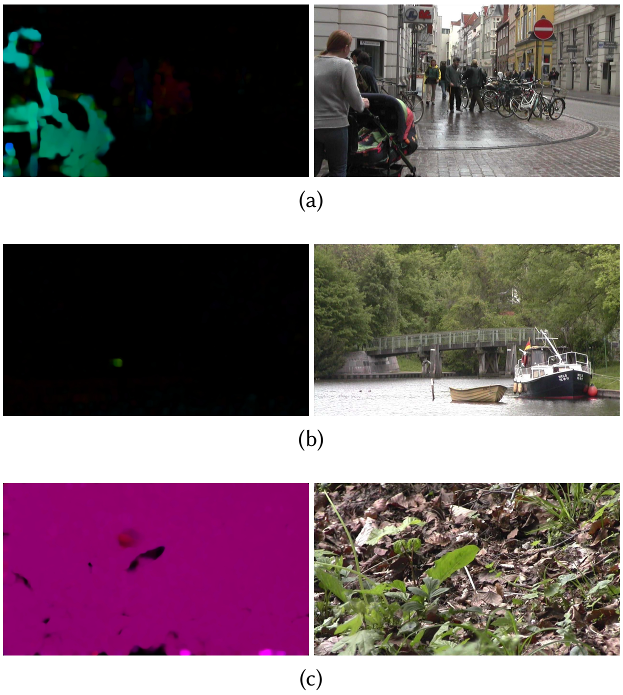


Figure 1: Optical flow (on the left) and corresponding video (on the right) frame examples, where most of the observers are performing smooth pursuit. These illustrate the general patterns we observed: (1a) an object has recently entered the scene, (1b) a single target is moving on a mostly static background, and (1c) the camera is in motion.

moving object, we estimated the time that this pixel was already visible (1 feature) by assuming that it has been moving with a constant speed – a vector equal to its current optical flow.

In order to somewhat describe the case of a single moving object in the video while the rest is mostly motionless, we computed the median and the maximum magnitudes of the optical flow vectors of the frame (2 features). Ideally, these would describe the speed of the moving object and the background scene motion, respectively. We additionally separately computed the gradient of the optical flow in vertical and horizontal axes, and extracted its maximal magnitude (1 value) to test for strong edges in the optical flow frame.

We used the mean and standard deviation of the optical flow (separately for its horizontal and vertical components) as features to capture camera movement (4 features).

We tried several model types and configurations, discovering that using a sequence of feature vectors (extracted for a sequence of video frames) and processing those with a long short-term memory network (LSTM) [Hochreiter and Schmidhuber 1996] yields better results. We eventually decided to use a small network that consists of one LSTM layer with 16 units, followed by four fully-connected layers (8 neurons in each, ReLU activations) and a final fully-connected output layer (1 neuron, sigmoid activation function). We placed dropout [Srivastava et al. 2014] layers (0.5 rate) in front of every fully-connected layer. We used binary cross-entropy as loss and the Adam optimiser [Kingma and Ba 2014] with default parameters (Keras version 2.1.6-tf [Chollet et al. 2015]). This model was trained on sequences of 12 feature vectors to predict the label (i.e. “pursuit” or “non-pursuit”) of the last considered frame. The training was carried out for up to 200 epochs, but could be stopped if no improvement in validation accuracy was observed for the last 20 epochs. We balanced the dataset so that both classes had roughly the same number of examples by oversampling the “pursuit” class.

To produce the labels for our data set, we used a cross-validation pipeline, where a separate model was tested for each video. The 17 remaining videos were split into the validation (first 3) and training (another 14) sets. The model with the best validation accuracy during training was then used for the prediction of the final labels of the *video-only* baseline for the corresponding video.

For this baseline, just like for the *majority-label* above, we only report the performance for its modification where we randomly subdivided the overlong fixations and pursuits (see Section 2.3.1), as the overall performance was consistently improved by this.

2.5 Inter-observer Baseline

This baseline tested the assumption that different observers will make similar eye movements when viewing the same stimulus (provided that their recordings are synchronised). Each recording was matched to a random other observer's recording for the same stimulus (all recordings used as a match once). We then produced the labelling of all the samples by using the matched recordings.

If the recorded gaze data were perfect, we could simply match (one-to-one) each of the gaze samples (and, therefore, their eye movement class labels) of one observer with all of the samples of the other, as every such pair of samples would be recorded at precisely the same time (relative to the stimulus onset), and no samples would be missing. With real data, however, we have to deviate from this procedure in two ways: First, we allowed for a temporal shift between the matched pair of samples of up to $\Delta_\tau = 20$ ms. Second, in case of missing samples (due to recording artefacts) we filled the gaps by the last matched class label. The temporal tolerance threshold Δ_τ plays the role of limiting the boundaries of what could be considered as temporal synchrony between two recordings and should not noticeably alter the results of this baseline. The only recommendation is not to set Δ_τ below one half of the eye tracker's sampling interval, as the recordings could otherwise be temporally misaligned in such a way that none of their samples can be matched.

3 EVALUATION

3.1 Algorithms and Data Sets

Since data-driven algorithmic annotation is not the focus point of this work, we simply tested a large crop of algorithms that were evaluated on the GazeCom data set in a recent work [Startsev et al. 2018] (correspondingly labelled files already provided via [Startsev et al. 2016]). In our evaluation we only included the algorithms that annotate smooth pursuit, which left us with the updated [Startsev et al. 2016] version of the [Startsev et al. 2018] algorithm, the approaches by [Agtzidis et al. 2016b], [Larsson et al. 2015], [Berg et al. 2009], and [Dorr et al. 2010], as well as three more algorithms implemented by [Komogortsev 2014]: I-VMP, I-VVT, and I-VDT. The last three were optimised in [Startsev et al. 2018] to deliver the best (on average across eye movement types and metrics) performance on GazeCom. We excluded I-VVT from our analysis since the optimisation procedure resulted in it effectively ignoring smooth pursuit detection in favour of better classifying other eye movement types.

We wanted to more extensively evaluate our inter-observer baseline, since it is a very easy-to-implement approach that directly relates to the diversity of salient targets in each individual stimulus in the data set. To this end, we tested it on three mode data sets (in addition to GazeCom, which we described Section 2.1) and compared the results. We considered the video-viewing subset of the data used in [Andersson et al. 2017], where we only used the labels of one of the annotators ("RA"), since his labels are present for a larger number of recordings, and it is important for the inter-observer baseline to have a representative set of observers.

We also considered the [Santini et al. 2016] data set for an example of fully artificial stimuli, but the mobile eye tracker recordings for different participants are not synchronised and, furthermore, the stimuli were randomly generated for each observer. Despite these by-design differences between the observers' recordings, we still performed the inter-observer analysis on this data.

Finally, we manually annotated a subset (50 clips, ca. 13 observers per video) of the Hollywood2 [Mathe and Sminchisescu 2012] data set of high-frequency (500 Hz) eye tracking recordings We annotated approximately 13,000 fixations, 15,000 saccades, and 5000 pursuits following the methodology of [Agtzidis et al. 2016a].

3.2 Metrics

We selected a wide range of commonly used evaluation measures from the literature. Sample-level Cohen's kappa [Andersson et al. 2017; Hooge et al. 2017; Larsson et al. 2015; Santini et al. 2016; Zemblys et al. 2018b] and F1 scores [Agtzidis et al. 2016b; Hoppe and Bulling 2016; Kasneci et al. 2015; Startsev et al. 2018] (also in the form of sensitivity and specificity [Larsson et al. 2013] and precision and recall [Anantrasirichai et al. 2016; Santini et al. 2016] reported separately) are frequently used by researchers, for example.

[Komogortsev et al. 2010] proposed a number of metrics for the evaluation of eye movement classifiers, most of which (e.g. FQnS, FQIS, SQnS) are only applicable when the stimulus is known and controlled. The authors also used average fixation durations, saccade amplitudes, and the number of eye movement events. Though many works report such average statistics [Andersson et al. 2017; Larsson et al. 2013; Nyström and Holmqvist 2010; Startsev et al. 2018; Zemblys et al. 2018b]), we did not include these in our evaluation since most of our baselines were by-design generating roughly "correct" numbers of episodes of each eye movement type with durations similar to those in the ground truth, so comparing the literature models to the baselines would be pointless. Additionally, these metrics (unlike any of the others we compute) are easily interpretable by field experts without an additional numerical yardstick.

Of course, just matching the number of events and their average statistics is not sufficiently quantifying the quality of the detected episodes. In addition to these measures, [Hooge et al. 2017] reported relative timing offset (RTO) and relative timing deviation (RTD), albeit for comparing human coders and not automatic detectors to the ground truth, but the principle is exactly the same. These are the mean and standard deviation of the difference in on- or offset timing of the "true" and detected events. This requires matching the events from the two labelled sets of eye tracking recordings, for which [Hooge et al. 2017] proposed finding the (temporally) first event in the second set that intersects with the considered event in the first set. After such matching is performed, event-level F1 scores can be computed (used in our evaluation as well). As for the RTO and RTD measures, they need to be reported separately for on- and offset of every event type, resulting in many statistics that need to be compared. Such an approach was appropriate in [Hooge et al. 2017], where only one event type (fixation) was considered, and the main focus was not on a compact evaluation of performance, but on finding systematic differences between human coders. [Startsev et al. 2018] proposed using the *intersection-over-union* ratio (IoU) statistic to compare the detected events to the ground truth ones in a less interpretable but more concise fashion, which we employed in our evaluation. Average IoU value across all ground truth episodes of a certain type is computed, with 0.0 corresponding to the missed events. This way, the quality of matched episodes as well as the number of correctly matched ground truth events are assessed at the same time, though not in an obvious combination.

Two very recent works [Startsev et al. 2018; Zemblys et al. 2018a] both suggest modifications to the event-level evaluation strategy of [Hooge et al. 2017]. [Zemblys et al. 2018a] mainly use Cohen's kappa for both sample- and event-level evaluation. For the event evaluation, they modify the procedure of matching the algorithmically detected events to the eye movement episode in the manual annotations. Instead of finding the earliest detected event that intersects with the considered "true" event, they find the one with the largest overlap area. We slightly modified this strategy by looking for the largest IoU, rather than for the largest intersection only: Suppose a case when a long fixation and a short saccade were detected, both intersecting with a ground truth saccade by 1/3 of its length. In terms of overlap size, these are indistinguishable, but the IoU will give a clear preference to the shorter saccade.

Following the implementation of [Zemblys et al. 2018a], after such matching is performed we paired the unmatched events in both the ground truth and the algorithmically detected sets with "empty" phantom events, and then directly computed the Cohen's kappa value of the resulting aligned sequences. Additionally, [Zemblys et al. 2018a] calculate sample and event "error rates", computed via normalising the Levenshtein edit distance (we divided it by the length of the largest of the compared label sequences), so we will refer to them as $\mathscr{L}_{\text{sample}}$ and $\mathscr{L}_{\text{event}}$, respectively, and tested our baselines against those measures as well. Contrary to what is stated in [Zemblys et al. 2018a], normalised Levenshtein distance between two sequences of equal length does *not* necessarily equal the misclassification rate, but rather does not exceed it (since the set of edit operations allowed under the Hamming distance definition is a strict subset of the corresponding operations allowed under the definition of the Levenshtein distance) [Navarro 2001].

We also followed the evaluation strategies of [Startsev et al. 2018], where the modification of the [Hooge et al. 2017] event matching scheme consists of limiting the possible event matches to those with the IoU no lower than a certain fixed threshold. The value recommended for theoretical interpretability is 0.5, since at this level no more than one match candidate can exist. For lower thresholds, the earliest-occurring of the matches is preferred (same as in [Hooge et al. 2017]). [Startsev et al. 2018] further argue for the practical preferableness of limiting acceptable IoU to values ≥ 0.5 by analysing the "random baseline" model (same as our *independent random samples* baseline). Here we will test whether this matching scheme is robust enough with our other random baselines as well.

To sum up, we used the following metrics: On the sample level, we tested the algorithms and baselines with (i) per-class F1 scores, (ii) per-class Cohen's kappa statistic κ_{sample}, and (iii) normalised Levenshtein distance measure $\mathscr{L}_{\text{sample}}$. On the event level, we computed (i) per-class F1 scores [Hooge et al. 2017], (ii) per-class Cohen's kappa κ_{event} [Zemblys et al. 2018a], (iii) normalised event-level Levenshtein distance $\mathscr{L}_{\text{event}}$, (iv) average per-class *IoU* ratio, as well as (v) per-class F1 scores at a varying IoU threshold for matched events, e.g. $\text{F1}_{\text{IoU}\geq 0.5}$ [Startsev et al. 2018].

We also proposed a novel metric based on the event-level Cohen's kappa of [Zemblys et al. 2018a]. We denote it κ_{adjusted} and separately explain its details in Section 5 as its motivation and implementation are drawing on the conclusions about other metrics.

4 RESULTS AND DISCUSSION

We computed all the selected metrics for the predictions of our baselines and the considered literature algorithms on the GazeCom data set. For space reasons, we only provide most of the values for smooth pursuit evaluation (Table 2); full, sortable tables are provided at https://github.com/MikhailStartsev/sp_tool/baselines.

4.1 Fixation and Saccade Detection

We first examine the evaluation procedures in the context of fixation and saccade detection (pursuit will be covered separately below). We omitted the evaluation results for these eye movements from Table 2 since they were not particularly surprising: All the baselines performed worse than the literature algorithms, especially on the saccade detection task, where the baselines' sample- and event-level F1 scores, for example, were between ca. 0.1 and 0.2 with all of the literature models scoring between 0.65 and 0.95.

For fixations, however, which are by far and away the most common eye movement label in GazeCom (72.5% samples), both the sample- and the event-level F1 scores (computed without any IoU thresholding) put the baselines very close to the scores of data-driven models: 0.85 for the best baseline vs. 0.88 for the worst literature model on the level of samples, and 0.73 vs. 0.79 on the level of events. It has to be noted that introducing a 0.5 IoU threshold makes this metric more discriminative (the score gap grows to ca. 0.4). Unlike the simplest random baseline in [Startsev et al. 2018], however, all our selected baselines scored above 0.2 on the $\text{F1}_{\text{IoU}\geq 0.5}$ metric for fixation detection. This means that the episode matching criteria can and should be made even stricter (i.e. the IoU threshold further increased): Event-level F1 scores for fixations only dropped below 0.05 for our baselines at the much higher IoU threshold of 0.8.

The IoU metric itself as well as sample- and event-level Cohen's kappa scores are very well distinguishing between the baselines and the other models in terms of fixation and saccade detection. The event-level κ_{event} exhibited slightly unusual behaviour when evaluating saccade detection: Almost all of our baselines received a negative score in the vicinity of -0.25. This is especially strange for the baselines that randomly assign event labels and their sequences, as this case should be the closest one can get to randomly assigning event labels; with Cohen's kappa this should correspond to a "perfect" zero-score. The current implementation of κ_{event} [Zemblys et al. 2018a], however, assumes a different source of randomness – arbitrarily assigning labels to pre-segmented events rather than arbitrarily labelling events in the signal, i.e. it is entirely possible under these assumptions that a 20 ms interval would be labelled as a fixation. This inherently biases the evaluation of short events, the timing of which is randomly offset: The probability of two such events being matched is proportional to the product of their respective lengths. On average in the GazeCom data set, ground truth fixations are 7 times longer in durations than saccades, meaning that the randomly-positioned (in time) saccades are almost 50 times less likely to match the "true" events of their own type.

4.2 Overall Performance and Smooth Pursuit Detection

For the simultaneous evaluation of all sample or event labels we used normalised Levenshtein distances (as proposed in [Zemblys et al. 2018a]). While no baseline scored higher than any literature model with respect to $\mathscr{L}_{\text{sample}}$ (though the closest scores were very similar – 0.279 or 0.254 for the majority vote baseline with or without sub-segmentation vs. 0.246 for the [Berg et al. 2009] algorithm), *all* literature approaches except for the deep model of [Startsev et al. 2018] had a larger event-level Levenshtein distance to the ground truth than three out of four baselines in Table 2. *All but two* models received a worse score than the video-only baseline there, which does not take *any* eye tracking or ground truth information into consideration. This leads us to argue that both Levenshtein distances are relatively weak metrics for eye movement detectors, which is likely due to equal assumed costs of mislabelling and omitting a sample (an event): Deleting even one fixation from a labelled sequence could lead to *all* of the subsequent labels not corresponding to the gaze recording. This error is, however, treated as equivalent to labelling this fixation as pursuit.

We now focus on the evaluation of smooth pursuit detection. Table 2 highlights (in magenta) the instances where a data-driven algorithm scored lower than at least one of the selectively presented baselines (grey rows). For now, we ignore the κ_{adjusted} metric, which we introduce together with our general recommendations about the metrics that are to be used for eye movement classification studies in Section 5. It can be clearly observed that a substantial part of literature models' scores are surprisingly worse than those of the baselines we proposed. The only metric for which none of the algorithms scored below any of the baselines is intersection-over-union, IoU. Under any other evaluation conditions, I-VDT and [Dorr et al. 2010] were consistently worse at detecting pursuit than some of our baselines (not shown in Table 2: for sample-level F1, the best baseline score was achieved by the *majority vote* baseline without event sub-segmentation – 0.404). We note that the majority voting with sub-segmentation as well as the inter-observer baseline scored particularly well in the event-level evaluation part for the detection of pursuit, with the video-only approach a little behind.

Among the tested data-driven algorithms, only the [Startsev et al. 2018] approach was consistently ahead of the baselines, with

Table 2: Sample- and event-level performance: for normalised Levenshtein distances, lower is better, for all other statistics, higher is better. The baseline rows are highlighted in grey. Individual cells are highlighted in magenta if the corresponding score is worse than at least one of the baselines evaluated here. Best algorithmic and baseline scores in each column are boldified. The rows are sorted by average performance (mean of all columns, Levenshtein distances used with a negative sign).

	All classes		Pursuit samples		Pursuit events				
Model	$\mathcal{L}_{\text{sample}}$	$\mathcal{L}_{\text{event}}$	F1	κ_{sample}	F1	IoU	$\text{F1}_{\text{IoU}\geq 0.5}$	κ_{event}	κ_{adjusted}
[Startsev et al. 2018]	**0.103**	**0.140**	**0.707**	**0.672**	**0.629**	**0.621**	**0.542**	**0.539**	**0.313**
[Agtzidis et al. 2016b]	0.219	0.351	0.646	0.607	0.527	0.448	0.394	0.444	0.199
I-VMP (optimised)	0.167	0.331	0.581	0.536	0.531	0.454	0.427	0.418	0.181
[Larsson et al. 2015]	0.154	0.400	0.459	0.407	0.392	0.277	0.321	0.339	0.171
[Berg et al. 2009]	0.246	0.278	0.422	0.365	0.424	0.351	0.266	0.259	0.118
Majority vote + sub-segment.	**0.279**	**0.245**	**0.379**	**0.328**	**0.439**	0.128	**0.115**	**0.216**	0.007
[Dorr et al. 2010]	0.178	0.309	0.381	0.299	0.193	0.367	0.092	0.044	0.064
Inter observer	0.328	0.266	0.321	0.236	0.388	0.142	0.114	0.191	**0.008**
Video only + sub-segment.	0.342	0.289	0.317	0.208	0.282	**0.166**	0.069	0.105	0.003
I-VDT (optimised)	0.231	0.468	0.321	0.230	0.152	0.359	0.073	0.032	0.044
Random event sequences	0.355	0.251	0.118	0.005	0.162	0.052	0.030	0.028	-0.001

[Agtzidis et al. 2016a] and I-VMP only being subservient to the baselines in one metric ($\mathcal{L}_{\text{event}}$).

About the κ_{sample} scores we note in particular that these were very close to zero for all of our random baselines. For non-random baselines, however, these were sometimes higher than the corresponding values for literature models (see Table 2), meaning that κ scores are good at discerning randomness in label assignment, but not necessarily so for comparing algorithms to one another.

4.3 Inter-observer Similarity of Eye Movements in Four Data Sets

Much like [Dorr et al. 2010] and [Mital et al. 2011] analysed spatio-temporal congruency of gaze points of multiple subjects, our *inter-observer* baseline can be used to assess the amount of similarities between the eye movements elicited by the same stimulus in different participants. As can be seen from Table 3, even when free-viewing video stimuli (GazeCom and Hollywood2 rows), the directly stimulus-dependent eye movements (smooth pursuit, in this case, since it cannot be performed without a moving target) were, to a large extent, similarly performed and timed between subjects, often resulting in performance better than dedicated algorithms (see Table 2). However, the scores of the baseline were the lowest on GazeCom, pointing towards its higher diversity.

When instructions are added to the video viewing (e.g. in the data set used by [Andersson et al. 2017], where participants were instructed to follow the moving objects in videos [Larsson et al. 2013]), the congruency further increases. We used a slightly different (more recordings, but only annotated by one expert instead of two) version of the data set than [Andersson et al. 2017], but since the algorithms' performance there is reported as largely similar when compared to either of the experts, and the difference in the metrics is fairly large, we feel confident in the generalisation of the following: While none of the algorithms compared in [Andersson et al. 2017] accounted for smooth pursuit, their best overall performance (reported as error rates) on the video-viewing data was at 61% disagreement rate, while the inter-observer baseline had a better score of 43%. As for the detection of fixations, the best sample-level Cohen's kappa score for algorithmic detection as reported in [Andersson et al. 2017] was 0.14, while the inter-observer baseline scored 0.24. Several pursuit-detecting algorithms were evaluated on the [Andersson et al. 2017] data in [Startsev et al. 2018]. The inter-observer κ_{sample} score of 0.25 is better than that for six out of nine approaches tested there. The corresponding F1 scores for this baseline (see Table 3) were slightly inferior just to one algorithm in either sample- or event-level setting: 0.7 vs. 0.74 for I-VMP or 0.58 vs. 0.59 for [Agtzidis et al. 2016b], respectively.

Table 3: A selection of metrics for smooth pursuit detection with the inter-observer baseline on multiple data sets. The video-viewing subset of [Andersson et al. 2017] data set is considered. The rows are sorted by the average score.

Data set	κ_{sample}	sample F1	event F1	event $\text{F1}_{\text{IoU}\geq 0.5}$
[Andersson et al. 2017]	0.25	**0.70**	**0.58**	0.15
Hollywood2	**0.30**	0.50	0.53	0.15
[Santini et al. 2016]	0.25	0.36	0.54	**0.20**
GazeCom	0.24	0.32	0.39	0.11

The recordings in [Santini et al. 2016] are not temporally synchronised, and stimuli in this data set were uniquely generated for each subject. Nevertheless, the similarities in the pseudo-randomly generated sequences of artificial target movements allowed the inter-observer baseline to come close (on some metrics) to the performance of the I-BDT algorithm, which was developed together with this data set: For the event-level F1 scores of [Hooge et al. 2017] for pursuit, I-BDT is only 0.06 higher than this baseline (0.6 vs. 0.54). When the IoU thresholding is applied, however, the differences were much more noticeable (0.53 vs. 0.2). Once again, these results highlight the importance of the evaluation measure choice.

Similar to the findings of [Dorr et al. 2010], we observed that Hollywood2 recordings contain much less variation of eye movement types (see Table 3), in addition to being less spatially variant, compared to the naturalistic videos of the GazeCom data set.

The inter-observer baseline can be additionally interpreted as the quality of labels one would get, if only one recording (of an "average" observer) per stimulus was manually annotated. Alternatively, if the stimuli are synthetic or observers receive instructions regarding their viewing behaviour, the information about the likely eye movements may be already available to the researchers and could be used to "automatically" label the recordings. Comparing algorithms to the inter-observer baseline would, in this case, characterise the gain these methods deliver over simply assuming the performed eye movement types based on this information.

When analysing a data set, we recommend assessing the diversity of its recordings, for instance by testing the inter-observer baseline on its labels, which would reveal how much similarity is shared between the simultaneously performed eye movements of different subjects. In some cases, a very simple assumption that participants will perform the eye movements that they "should" (either based on the nature of the stimuli or on the instructions) could yield greater performance than a generic eye movement detection algorithm.

5 ADJUSTING THE COHEN'S KAPPA METRIC

Having evaluated our baselines on a number of existing metrics from the literature, we focused on combining their strengths while avoiding the observed weaknesses. Below, we first outline the advice that can be drawn from our experiments and thus motivate the introduction of a new evaluation strategy.

Based on the observations made in Section 4.2, we advise against using Levenshtein distances. We also recommend using stronger matching criteria for event-level evaluation, increasing the thresholds for acceptable matches even further than in [Startsev et al. 2018], since the IoU scores themselves differentiated well between baselines and dedicated models, but setting a threshold at 0.5 did not achieve the desired strictness of event matching. Overall, it seems that only evaluating the presence of event matches between the detected and the "true" ones (e.g. as F1 scores or Cohen's kappa do) is not sufficient, and the quality of events needs to be assessed with a special metric (e.g. IoU). IoU itself demonstrated the greatest power to distinguish the baselines from the dedicated algorithms, but there is no obvious interpretation of its way of reflecting on both the quality of matched events and the amount of missed ones.

We, therefore, proposed a novel event evaluation strategy to overcome the shortcomings of the existing approaches. Our main motivation sources were (i) making event matching stricter while maintaining the clear separation between how events are matched and how the matched events are evaluated (like IoU thresholding and unlike IoU averaging in [Startsev et al. 2018]), (ii) clearly separating baselines and algorithmic approaches by their performance, as well as (iii) obtaining close-to-zero scores for the random baselines, as κ_{sample} does, while (iv) avoiding the bias of the [Zemblys et al. 2018a] κ_{event} against short event detection (see Section 4.1).

To this end, we combined the event-matching strategy of [Startsev et al. 2018] with the modified Cohen's kappa scoring procedure of [Zemblys et al. 2018a]: First of all, two events can be matched only if they have an IoU over a certain threshold, which we set to 0.8 (see Section 4.1 for the source of this value; this is a parameter that might need to be increased as algorithmic detection improves further), meaning that no more than one algorithmically detected eye movement episode can form a potential match with one event of the ground truth set. Second, to obtain the chance-level performance (to normalise the observed agreement between the true and detected events, just as in the traditional Cohen's kappa formula) we randomly re-shuffled the detected events, preserving their type and duration. This can be repeated several times, but on the large GazeCom data set we did not observe large inter-run variance. Lastly, if a certain eye movement type label is evaluated, the "agreement" between the two sets of labels was computed by considering only the events of this class, since correctly matching the timing of the negative-label events should not contribute to the performance estimate of the positive-label event detection. We provide the implementation of this and all other metrics at https://github.com/MikhailStartsev/sp_tool.

Table 2 demonstrates that no baseline performed better than any considered algorithm with respect to this metric, κ_{adjusted}, with the lowest score for a literature algorithm over five times higher than the highest baseline score. Also, all of our random baselines scored between -0.002 and 0.002 on this metric for all eye movement types, so we do not observe any eye movement-specific bias either.

As to the limitations of the proposed metric, its high IoU threshold might render κ_{adjusted} unsuitable for being used as the basis for a loss function for gradual parameter tuning of machine learning models: If the initial score is poor, it is unlikely to change with small parameter alterations. We would recommend gradually increasing the IoU threshold as the training progresses, similar to learning rate decay [Smith et al. 2018], or combining this metric with others.

6 CONCLUSIONS

Here, we first proposed and tested several eye movement classification baselines that either only randomly model the sequences of samples or events of different eye movement types, or make no distinction between individual recordings. Notably, none of our baselines are gaze data-dependent, i.e. the properties of the recorded eye tracking signal are never considered.

We then further tested the baseline that directly leverages the (likely stimulus- or instructions-driven) similarities between the eye movements of different subjects on three more independent data sets, which allowed us to compare their inter-observer diversity, as well as obtain some context for the algorithmic performance on these, since this baseline approach approximates sparsely annotating the recordings and re-using the labels for other recordings on the same stimuli.

The results of the gaze data-free baselines reflect, in part, on the metrics used for evaluating the classification models and reveal their shortcomings. To avoid the observed weaknesses and biases of currently used measures, the new κ_{adjusted} metric should be used for the evaluation of eye movement event classification algorithms.

ACKNOWLEDGMENTS

This research was supported by the Elite Network Bavaria, funded by the Bavarian State Ministry of Science and the Arts.

REFERENCES

Ioannis Agtzidis, Mikhail Startsev, and Michael Dorr. 2016a. In the pursuit of (ground) truth: A hand-labelling tool for eye movements recorded during dynamic scene viewing. In *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*. 65–68. https://doi.org/10.1109/ETVIS.2016.7851169

Ioannis Agtzidis, Mikhail Startsev, and Michael Dorr. 2016b. Smooth pursuit detection based on multiple observers. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 303–306.

Nantheera Anantrasirichai, Iain D Gilchrist, and David R Bull. 2016. Fixation identification for low-sample-rate mobile eye trackers. In *2016 IEEE International Conference on Image Processing (ICIP)*. 3126–3130. https://doi.org/10.1109/ICIP.2016.7532935

Richard Andersson, Linnea Larsson, Kenneth Holmqvist, Martin Stridh, and Marcus Nyström. 2017. One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms. *Behavior Research Methods* 49, 2 (01 Apr 2017), 616–637. https://doi.org/10.3758/s13428-016-0738-9

F. Behrens, M. MacKeben, and W. Schröder-Preikschat. 2010. An improved algorithm for automatic detection of saccades in eye movement data and for calculating saccade parameters. *Behavior Research Methods* 42, 3 (01 Aug 2010), 701–708. https://doi.org/10.3758/BRM.42.3.701

David J. Berg, Susan E. Boehnke, Robert A. Marino, Douglas P. Munoz, and Laurent Itti. 2009. Free viewing of dynamic stimuli by humans and monkeys. *Journal of Vision* 9, 5 (5 2009), 1–15. https://doi.org/10.1167/9.5.19 arXiv:/data/journals/jov/932860/jov-9-5-19.pdf

François Chollet et al. 2015. Keras. https://github.com/keras-team/keras.

Michael Dorr, Thomas Martinetz, Karl R Gegenfurtner, and Erhardt Barth. 2010. Variability of eye movements when viewing dynamic natural scenes. *Journal of Vision* 10, 10 (2010), 28–28.

Ralf Engbert and Reinhold Kliegl. 2003. Microsaccades uncover the orientation of covert attention. *Vision Research* 43, 9 (2003), 1035 – 1045. https://doi.org/10.1016/S0042-6989(03)00084-1

Ralf Engbert and Konstantin Mergenthaler. 2006. Microsaccades are triggered by low retinal image slip. *Proceedings of the National Academy of Sciences* 103, 18 (2006), 7192–7197. https://doi.org/10.1073/pnas.0509557103 arXiv:http://www.pnas.org/content/103/18/7192.full.pdf

Roy S. Hessels, Diederick C. Niehorster, Marcus Nyström, Richard Andersson, and Ignace T. C. Hooge. 2018. Is the eye-movement field confused about fixations and saccades? A survey among 124 researchers. *Royal Society Open Science* 5, 8 (2018). https://doi.org/10.1098/rsos.180502 arXiv:http://rsos.royalsocietypublishing.org/content/5/8/180502.full.pdf

Sepp Hochreiter and Jürgen Schmidhuber. 1996. LSTM Can Solve Hard Long Time Lag Problems. In *Proceedings of the 9th International Conference on Neural Information Processing Systems (NIPS'96)*. MIT Press, Cambridge, MA, USA, 473–479. http://dl.acm.org/citation.cfm?id=2998981.2999048

Ignace T. C. Hooge, Diederick C. Niehorster, Marcus Nyström, Richard Andersson, and Roy S. Hessels. 2017. Is human classification by experienced untrained observers a gold standard in fixation detection? *Behavior Research Methods* (19 Oct 2017). https://doi.org/10.3758/s13428-017-0955-x

Sabrina Hoppe and Andreas Bulling. 2016. End-to-End Eye Movement Detection Using Convolutional Neural Networks. *ArXiv e-prints* (Sept. 2016). arXiv:cs.CV/1609.02452

Tilke Judd, Frédo Durand, and Antonio Torralba. 2012. A benchmark of computational models of saliency to predict human fixations. http://hdl.handle.net/1721.1/68590.

T. Judd, K. Ehinger, F. Durand, and A. Torralba. 2009. Learning to predict where humans look. In *2009 IEEE 12th International Conference on Computer Vision*. 2106–2113. https://doi.org/10.1109/ICCV.2009.5459462

Enkelejda Kasneci, Gjergji Kasneci, Thomas C. Kübler, and Wolfgang Rosenstiel. 2015. Online Recognition of Fixations, Saccades, and Smooth Pursuits for Automated Analysis of Traffic Hazard Perception. In *Artificial Neural Networks*, Petia Koprinkova-Hristova, Valeri Mladenov, and Nikola K. Kasabov (Eds.). Springer International Publishing, Cham, 411–434.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). arXiv:1412.6980 http://arxiv.org/abs/1412.6980

Thomas Kinsman, Karen Evans, Glenn Sweeney, Tommy Keane, and Jeff Pelz. 2012. Ego-motion Compensation Improves Fixation Detection in Wearable Eye Tracking. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, NY, USA, 221–224. https://doi.org/10.1145/2168556.2168599

Oleg V. Komogortsev. 2014. Eye Movement Classification Software. http://cs.txstate.edu/~ok11/emd_offline.html.

Oleg V. Komogortsev, Sampath Jayarathna, Do Hyong Koh, and Sandeep Munikrishne Gowda. 2010. Qualitative and Quantitative Scoring and Evaluation of the Eye Movement Classification Algorithms. In *Proceedings of the 2010 Symposium on Eye-Tracking Research &#38; Applications (ETRA '10)*. ACM, New York, NY, USA, 65–68. https://doi.org/10.1145/1743666.1743682

Oleg V. Komogortsev and Alex Karpov. 2013. Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behavior Research Methods* 45, 1 (2013), 203–215.

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

Linnéa Larsson, Marcus Nyström, and Martin Stridh. 2013. Detection of Saccades and Postsaccadic Oscillations in the Presence of Smooth Pursuit. *IEEE Transactions on Biomedical Engineering* 60, 9 (Sept 2013), 2484–2493. https://doi.org/10.1109/TBME.2013.2258918

Stefan Mathe and Cristian Sminchisescu. 2012. Dynamic Eye Movement Datasets and Learnt Saliency Models for Visual Action Recognition. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II (ECCV'12)*. Springer-Verlag, Berlin, Heidelberg, 842–856. https://doi.org/10.1007/978-3-642-33709-3_60

Olivier Le Meur and Zhi Liu. 2015. Saccadic model of eye movements for free-viewing condition. *Vision Research* 116 (2015), 152 – 164. https://doi.org/10.1016/j.visres.2014.12.026 Computational Models of Visual Attention.

Parag K. Mital, Tim J. Smith, Robin L. Hill, and John M. Henderson. 2011. Clustering of Gaze During Dynamic Scene Viewing is Predicted by Motion. *Cognitive Computation* 3, 1 (01 Mar 2011), 5–24. https://doi.org/10.1007/s12559-010-9074-z

Gonzalo Navarro. 2001. A Guided Tour to Approximate String Matching. *ACM Comput. Surv.* 33, 1 (March 2001), 31–88. https://doi.org/10.1145/375360.375365

Marcus Nyström and Kenneth Holmqvist. 2010. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods* 42, 1 (01 Feb 2010), 188–204. https://doi.org/10.3758/BRM.42.1.188

Jorge Otero-Millan, Jose L. Alba Castro, Stephen L. Macknik, and Susana Martinez-Conde. 2014. Unsupervised clustering method to detect microsaccades. *Journal of Vision* 14, 2 (2014), 18. https://doi.org/10.1167/14.2.18 arXiv:/data/journals/jov/932814/i1534-7362-14-2-18.pdf

Dario D. Salvucci and Joseph H. Goldberg. 2000. Identifying Fixations and Saccades in Eye-tracking Protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications (ETRA '00)*. ACM, New York, NY, USA, 71–78. https://doi.org/10.1145/355017.355028

Thiago Santini. 2016. Automatic Identification of Eye Movements. http://ti.uni-tuebingen.de/Eye-Movements-Identification.1845.0.html.

Thiago Santini, Wolfgang Fuhl, Thomas Kübler, and Enkelejda Kasneci. 2016. Bayesian Identification of Fixations, Saccades, and Smooth Pursuits. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 163–170. https://doi.org/10.1145/2857491.2857512

Samuel L. Smith, Pieter-Jan Kindermans, and Quoc V. Le. 2018. Don't Decay the Learning Rate, Increase the Batch Size. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1Yy1BxCZ

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

Mikhail Startsev, Ioannis Agtzidis, and Michael Dorr. 2016. Smooth Pursuit. http://michaeldorr.de/smoothpursuit/.

Mikhail Startsev, Ioannis Agtzidis, and Michael Dorr. 2018. 1D CNN with BLSTM for automated classification of fixations, saccades, and smooth pursuits. *Behavior Research Methods* (08 Nov 2018). https://doi.org/10.3758/s13428-018-1144-2

Mikhail Startsev and Michael Dorr. 2018. Increasing Video Saliency Model Generalizability by Training for Smooth Pursuit Prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Julian Steil, Michael Xuelin Huang, and Andreas Bulling. 2018. Fixation Detection for Head-mounted Eye Tracking Based on Visual Similarity of Gaze Targets. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications (ETRA '18)*. ACM, New York, NY, USA, Article 23, 9 pages. https://doi.org/10.1145/3204493.3204538

Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2012. Detection of Smooth Pursuits Using Eye Movement Shape Features. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, NY, USA, 177–180. https://doi.org/10.1145/2168556.2168586

Raimondas Zemblys, Diederick C. Niehorster, and Kenneth Holmqvist. 2018a. gazeNet: End-to-end eye-movement event detection with deep neural networks. *Behavior Research Methods* (17 Oct 2018). https://doi.org/10.3758/s13428-018-1133-5

Raimondas Zemblys, Diederick C. Niehorster, Oleg V. Komogortsev, and Kenneth Holmqvist. 2018b. Using machine learning to detect events in eye-tracking data. *Behavior Research Methods* 50, 1 (01 Feb 2018), 160–181. https://doi.org/10.3758/s13428-017-0860-3

E

360-aware Saliency Estimation

This work has contributed to the field of saliency prediction in the omnidirectional domain. Our approach produces high-quality saliency maps for equirectangular 360° images by applying traditional, two-dimensional image saliency models to a set of manipulated versions of the original 360° image. Combining this general idea with an ensemble of three existing saliency predictors [74, 157, 158] (without any re-training) demonstrated excellent performance, winning the corresponding IEEE ICME "Salient360!" Grand Challenge in 2017 [95] (best head and eye movement prediction).

In particular, we focused on two major problems with saliency prediction directly in equirectangular content: (i) the discrepancy between immersive nature of the stimulus and the presence of borders in the image representation and (ii) the distortions resulting from projecting a spherical image onto a rectangular shape. We utilised two approaches to overcome these artefacts: (i) Rotating the spherical image before projection and (ii) projecting the spherical scene onto a set of six cube faces instead of a single image.

In traditional saliency predictors, pixels near image borders often receive noticeably lower saliency scores, compared to pixels further away from the borders, regardless of the depicted content. Predicting the spherical saliency at several different rotations (approach (i) above) ensures that equirectangular image borders would correspond to different locations on the sphere. When aggregated via pixel-wise maximum, these saliency predictions form a final output with greatly diminished border artefacts.

The second approach allows to undistort the equirectangular projection images, producing more traditional views of the depicted objects. We countered the appearance of many additional borders artefacts as a result of this (near the borders of each of the cube faces) via a rotation procedure similar to the one describe above.

For the final model, we combined the two proposed approaches, computing the saliency maps from both the rotated input image (around the vertical axis only) and the top and bottom cube faces, *i.e.* where distortions are most severe. This approach was demonstrated to outperform the two individual manipulation strategies above.

My personal contributions include (i) designing and implementing the equirectangular image manipulation methods; (ii) integrating our proposed pipeline with the saliency model implementations of [74, 157, 158], slightly modifying those to avoid unnecessary central biasing, normalisation, and quantisation; (iii) writing the manuscript.

Signal Processing: Image Communication 69 (2018) 43–52

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

360-aware saliency estimation with conventional image saliency predictors

Mikhail Startsev *, Michael Dorr

Technical University of Munich, Institute for Human-Machine Communication, Arcisstr. 21, Munich, 80333, Germany

ARTICLE INFO

Keywords:
Saliency prediction
Equirectangular projection
Panoramic images

ABSTRACT

This work explores saliency prediction for panoramic 360°-scenes stored as equirectangular images, using exclusively regular "flat" image saliency predictors. The simple equirectangular projection causes severe distortions in the resulting image, which need to be compensated for sensible saliency prediction in all viewports. To address this and other arising issues, we propose several ways of interpreting equirectangular images and analyse how these affect the quality of the resulting saliency maps. We perform our experiments with three popular conventional saliency predictors and achieve excellent results on the "Salient360!" Grand Challenge data set (ranked 1st among the blind-test submissions in the Head–Eye Saliency Prediction track).

1. Introduction

Even though we seemingly perceive our entire surrounding as a whole, this is impossible because of the physical constraints of our visual system. Only a small part of our visual field is projected onto a high-resolution part of the retina — the area called *fovea*. This foveation reduces the computational load on the visual cortex and bandwidth requirements on the optic nerve, but forces our eyes to constantly scan the scene to obtain the "full picture". This means that from such fragmented input our brain has to reconstruct a comprehensive model of what surrounds us. The strategy of visual exploration is therefore an important factor of human adaptation, which had both social and environmental factors impact its development.

Being able to predict or model the process of this "biologically-approved" attention allocation can aid various computer vision-related areas in the struggle for sparsity [1,2], help action recognition [3,4] and semantic segmentation [5], or even potentially shed light on and aid diagnosis of mental disorders [6,7]. With 360°-content becoming more and more widespread on popular image- and video-sharing platforms, as well as with the rise of consumer-oriented virtual reality applications and 360-camera set-ups, the saliency models for such stimuli can facilitate its analysis and compression, for example in order to enhance user immersion.

Working with the panoramic image scenario is generally beneficial for understanding attention. First of all, whereas conventional 2D image saliency data sets are often recorded under restrictive laboratory conditions, the free head motion of 360°-recordings means this scenario is much closer to real-life viewing behaviour.

Just as regular image or video saliency, this scenario does not yet introduce the social aspects of attention, such as avoiding either prolonged eye contact with strangers [8] or even looking at people when they are close-by in a genuine social context altogether [9], or seeking out familiar faces in crowds. But the prioritisation of observers' attention has a different component to it, making it two levels deep: first the head rotation, and then the eye gaze direction.

Compared to fully-unconstrained complex recording scenarios, static 360°-stimuli allow us to analyse common objects and regions of interest for multiple observers without having to match the contents of the foveated patches with one another, or deal with depth perception or occlusions. This eases the transition from numerous readily available 2D image saliency predictors, which have much larger data sets that could be used for training and evaluation. This work explores the possibilities and needed image transformations to perform this very transition.

In this work we have, therefore, proposed a range of transformations of the input equirectangular images, which we call "interpretations", that allow us to predict 360° saliency using any existing 2D attention model. In our experiments, we used three publicly available saliency prediction algorithms that model different levels of the visual processing hierarchy. Our approach demonstrated excellent results on a data set of omnidirectional images without any training or parameter adjustments.

In contrast to the work in [10,11], for example, which presents a CNN-based approach, where the network is fitted for the available set of the equirectangular images, and several strategies to prevent overfitting had to be applied as a consequence of the data set size, our approach does not require any additional training and can be used with any conventional pre-trained saliency model. In [12], an approach involving

[*] Corresponding author.
E-mail address: mikhail.startsev@tum.de (M. Startsev).

https://doi.org/10.1016/j.image.2018.03.013
Received 7 September 2017; Received in revised form 19 March 2018; Accepted 20 March 2018
Available online 6 April 2018

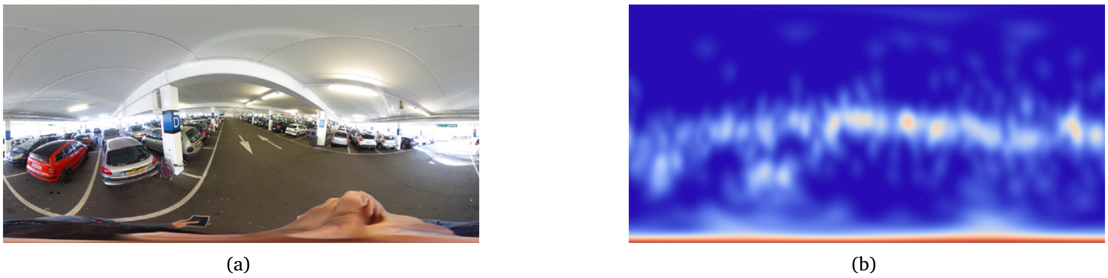


Fig. 1. Equirectangular image example 1(a) and its ground-truth saliency map 1(b).

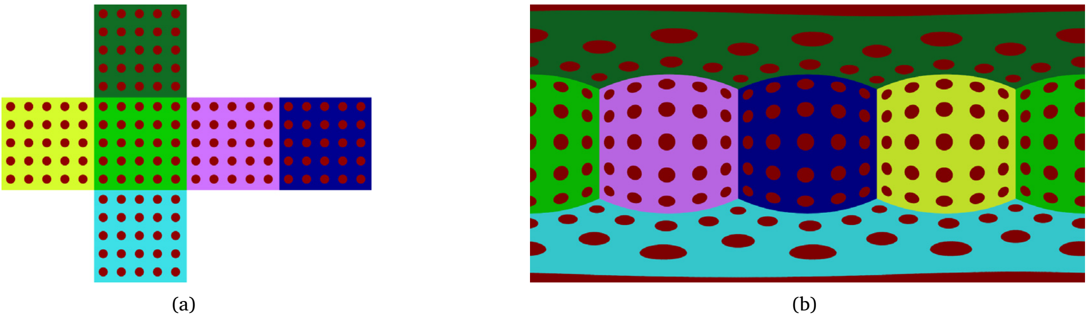


Fig. 2. Distortion visualization for equirectangular projection 2(b) and the set of corresponding cube map faces 2(a). Note that the red bottom and top stripes in 2(b) each represent just one disk on top and bottom faces of 2(a).

an idea similar to what we here call "interpretations" was applied for predicting salient viewports as a post-processing step for conventional saliency predictors' outputs, but it does not get rid of all the issues that arise for the eye gaze-based saliency prediction, originally only addressing the centre bias.

2. Proposed approach

Dealing with omnidirectional images is a challenge on its own, as the "perfect" way to store and process them is yet to be developed: So far, there is always a trade-off between efficiency, visual interpretability, and convenience of use. The data set that we use in this work (described in Section 3.1) employs equirectangular projection, so we first examine its artefacts, and then describe how they can be mitigated for a better saliency prediction via proposed interpretations.

2.1. Motivation

Aside from the obvious unnatural visual stretching of the objects at the top and the bottom of any equirectangular image (for an example, see Fig. 1(a)), there are several issues that are particularly prominent when such an image is being processed automatically, for instance by attention predictors (for an example of an empirical ground truth saliency map, see Fig. 1(b)). In [13], a similar data set to the one used here was introduced, and the authors reported some preliminary findings regarding the equirectangular image peculiarities in the context of subjective and objective quality evaluation. In [12], the authors investigated the prediction of head rotation-based saliency and examined the artefacts occurring in such "head saliency maps".

A regular saliency predictor expects its input to be a 2D image, and does not rely on any additional information about it. Below we describe several reasons why directly applying a saliency prediction models to equirectangular images might not be wise. First, the already mentioned image structure distortions might result in irregular feature responses. A significant part of an image produced through equirectangular projection suffers little to moderate shape distortion, but the parts close to its top and bottom are noticeably malformed, enough for a human not to recognise a shape right away (see an example image pair for a set of simplest shapes in Fig. 2; also, can you recognise a human head in Fig. 1?).

The second issue is related to the well-known centre bias effect, observed at least as early as 1935 [14], which is very noticeable in regular image saliency data sets (see Fig. 3(a)), and is extremely persistent across different data sets, tasks, image feature distributions, or forced first fixation location for static images [15], as well as for videos of dynamic natural scenes [16,17].

This effect is very different for 360° images (see Fig. 3(b)). Instead, we see attention bias along the vertical axis, with the central, the topmost, and the bottom-most locations of the equirectangular images all accumulating significant portions of the overall saliency distribution. This was also observed in [18], as well as in [12], in that case even more prominently so for the head-only saliency. The term "equator bias" was used in the latter to describe this effect, and a general way to overcome the centre bias tendency in regular saliency predictions was introduced.

The two issues described above lead in turn to a third problem. The border artefacts that could be neglected for regular image saliency prediction, in part due to the centre bias (on average, only a small part of saliency is allocated close to the image borders), can be neglected no more. From the theoretical standpoint, there were no actual borders in the stimulus, the viewport never contained a discontinuous image during recording. Now from the practical point of view, directly applying a regular saliency model to an equirectangular stimulus will most likely generate some border effects, both vertical (i.e. neglecting horizontal continuity; the object right behind the starting point of the observation is basically cut in half and is not seen as a set of closely located pixels by the saliency predictor) and horizontal (which means that the most prominent parts of the average ground truth empirical saliency map in Fig. 3(b) are likely to fall into the border effect zone). Example saliency maps produced by the three saliency predictors we use in our experiments (see Section 3) can be found in Fig. 4.

2.2. Outline

We propose to deal with these issues with what we call "interpretations" of the equirectangular image format. In our approach (see Fig. 5

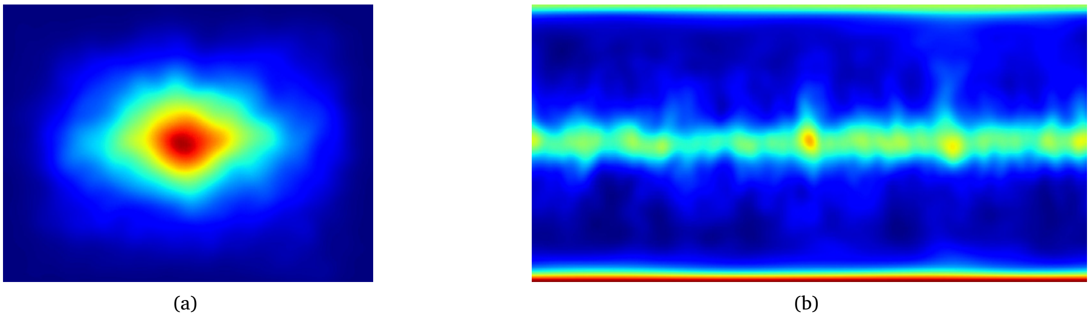


Fig. 3. "Centre bias" visualized as empirical mean saliency maps for the MIT1003 data set [19] of regular 2D images 3(a) and for the "Salient360!" [18] training set 3(b).

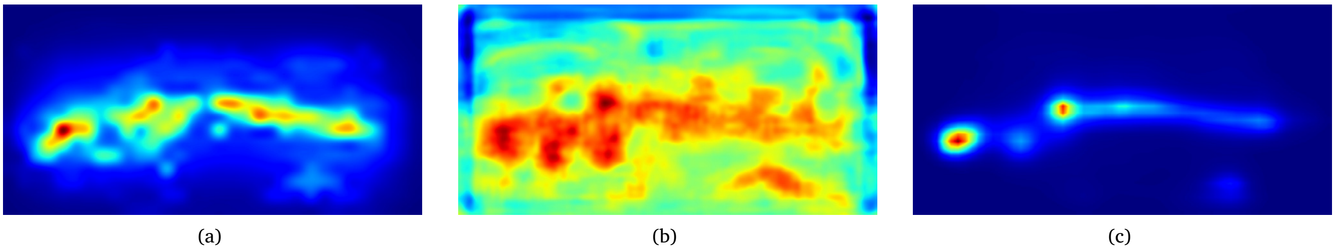


Fig. 4. Example saliency map predictions directly on an equirectangular image with the three existing predictor models we use in Section 3.3: GBVS [20] 4(a), eDN [21] 4(b) and SAM-ResNet [22] 4(c). Here we take the image in Fig. 1(a) as input. The ground-truth saliency map in Fig. 1(b) has its highest values along the bottom border, and the vertical borders neither on the left nor on the right side affect the continuity of the central "saliency strip". Both these observations do not hold for either of the directly predicted saliency maps.

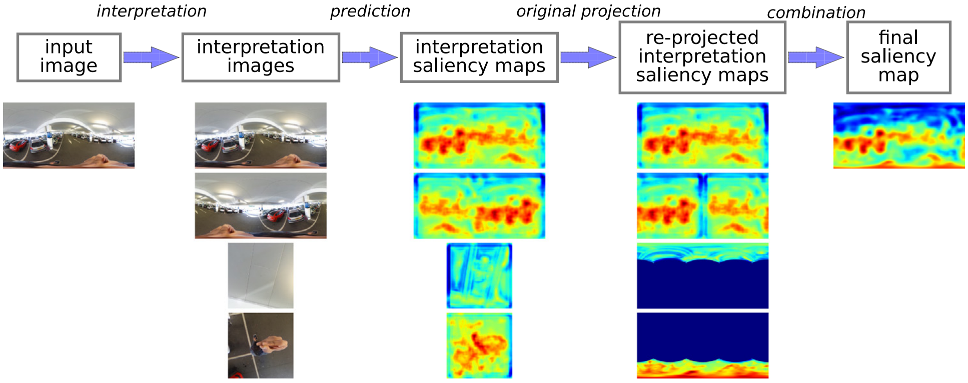


Fig. 5. The sequence of steps in our approach (top row). Example data samples from each stage are presented towards the bottom of the figure.

for the overview of its stages), we first create a set of intermediary images derived from the input image (this derivation is what we mean by "interpretation"). For these, respective saliency maps are predicted and subsequently re-projected into the equirectangular space corresponding to the input image, before they are combined into a final saliency map.

In order to combine several overlapping saliency maps into one, the final map is produced by taking the greatest predicted value in each individual pixel (i.e. applying the pixel-wise maximum operation). If we were to use the mean of predicted values, the pixels that were affected by border-related effects at least in one of the intermediary saliency maps would be at great disadvantage, compared to pixels that were never close to saliency map borders. Since we cannot guarantee the uniformity of the individual, interpretation-, model-, and content-dependent border effects across all pixels of the final saliency map, a reasonable solution would be to ignore the saliency values that were affected by being too close to borders. Using pixel-wise maximum achieves just that, discarding the very low intermediate saliency scores along the borders, provided that the respective values have been re-computed in any of the other saliency maps with a higher estimated saliency score.

The resulting saliency map is always smoothed with a Gaussian filter (σ proportional to the image size, $\sigma = 16\,\text{px}$ for input image height of $1024\,\text{px}$), and normalized to contain only non-negative values that sum to 1 over the entire map.

The following sections provide a detailed description of the several interpretation techniques we have explored.

2.3. Continuity-aware interpretation

To address the artefacts occurring at the left and the right borders of the input equirectangular images, we can use the knowledge that those edges can be seamlessly stitched. We therefore compute the saliency maps both for the original image without any preprocessing, and an image that has its left and right halves swapped (this is equivalent to looking in the direction opposite to the starting gaze direction, i.e. backwards). The reverse transformation is applied to the respective

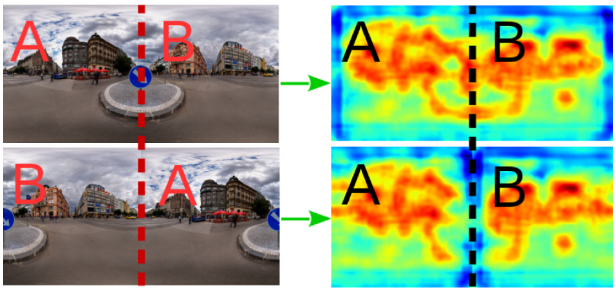


Fig. 6. Saliency map computation via continuity-aware interpretation. The final saliency map is obtained with a pixel-wise maximum operation on the two saliency maps on the right, which counteracts the artefacts seen on each of the two maps as dark to light blue vertical stripes either at the left and right borders, or near the dotted cut line.

saliency map (the "original projection" step in Fig. 5). The idea is graphically explained in Fig. 6.

This is similar to the Fused Saliency Map post-processing method in [12], where the equirectangular input was translated horizontally several times before saliency maps were predicted, and weighted averaging was applied to the prediction results in order to cancel out the centre bias effect of individual predictions. Here we need fewer rotations (2 instead of 4), since we attempted to switch off the centre bias for our models, where possible, so we mostly needed the rotation just to deal with the border artefacts of our saliency predictors, i.e. help preserve local scene context for feature computation near the borders.

2.4. Cube map-based interpretations

The continuity-aware interpretation only deals with left and right input image borders. Projection distortions, as well as the top and bottom border artefacts are not addressed. To remove the distortions of the equirectangular projection, we can convert the input 360-image to six faces of the cube centred around the camera position. The reverse projection brings the saliency maps from the cube map domain back into the equirectangular one.

Another benefit of this interpretation can be inferred from Fig. 2. For example, since the entire bottom stripe of the equirectangular image is produced from just one disk in the centre of the bottom cube map face, the saliency values in this stripe will be extracted from the middle of the respective cube face, which is unaffected by any potential border effects. As a result, the equirectangular saliency map produced with this interpretation in mind will be devoid of the top and the bottom border artefacts (for an example, see Fig. 7(a)).

The use of cube maps for omnidirectional scenes is not novel: In [23], several sphere-to-planar projections were examined in search for alternatives to the equirectangular format, in order to reduce bitrate or increase video quality at a given bitrate. Even though the cube map was not the best one overall, it was still an improvement over the equirectangular projection, while being natively supported by modern software. The authors of [24] also looked at a set of projections in the context of using the geometric structure of the projection layouts to select the "Quality Emphasized Regions" (QOR) for full-quality rendering. The quality of the respective spherical video presented to the observer was evaluated at a fixed bit-rate. The cube map layout yielded the best results in this study. Using saliency maps to prioritize different viewports was also suggested there (for selecting the QORs, adapted to scene content). This generally indicates that the cube map "interpretation" is not foreign to the field of 360°-scenes.

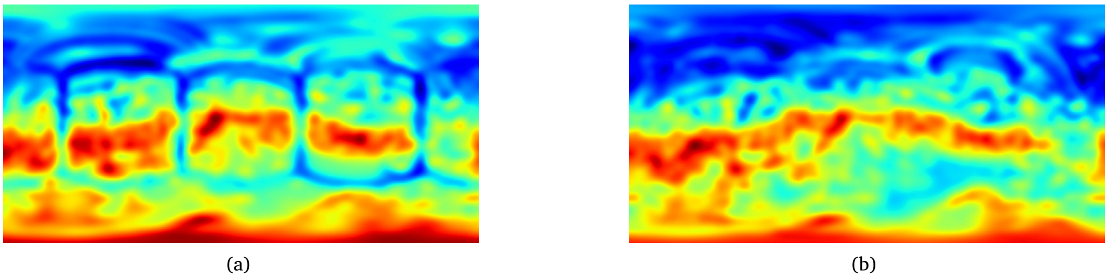
We explored multiple ways of leveraging this particular interpretation of the scene. First, we directly generated the saliency maps for all the cube faces and assembled them into an equirectangular saliency map (an example can be seen in Fig. 7(a)). This approach, however, loses the global context and introduces as many as 24 smaller border artefacts (4 for each face) that greatly deteriorated the quality of the final saliency prediction.

To compensate for these borders, one can generate a larger set of intermediary images and respective saliency maps by extracting the faces at several different rotations of the underlying cubic representation. This way we shift the borders between the stitched faces around the equirectangular saliency map (after the re-projection step), thus lessening the effect of these borders on the final map (see Fig. 7(b)). We take five different cube orientations: its original orientation, rotated by 45° relative to each axis separately, and rotated by 45° relative to the first two axes at the same time.

We can observe that the resulting saliency map does not exhibit any artefacts around its borders (e.g. the lower border accumulates significant amount of saliency, just as in the ground truth saliency map for this input in Fig. 1(b)). The context of the full scene is, however, still lost for the saliency predictors, since they only process one individual cube face at a time.

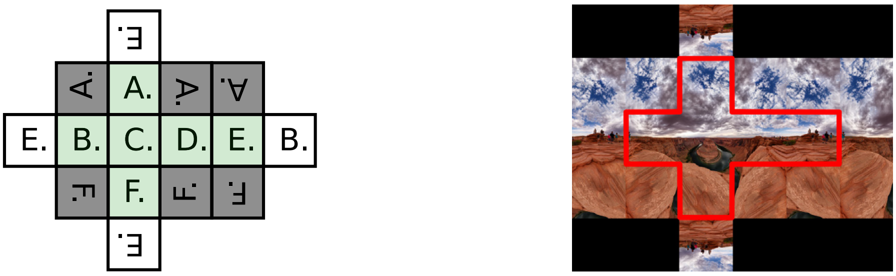
To preserve all the original image information and context in one image, one can assemble a cube map cutout, which will look similar to that in Fig. 2(a), with the faces replaced with pixels from the "main" cutout – the highlighted part – of Fig. 8(b). This does not fully get rid of the border artefacts, since five out of the six faces have at least two problematic edges either at the image border or due to bordering with an empty part of the cutout (only face "C" in Fig. 8(a) would have no discontinuities at its borders). A *filled cutout*, which is an image consisting of a grid of 3 × 4 cube faces stitched together (see the shaded areas of Fig. 8(a)), just like the central rectangle around the main cutout in Fig. 8(b), resolves only part of the border issues (four of the six main cutout faces are still at the image border). To further minimize these, we introduce an *extended cutout*, which augments the "main" and the "filled" cutouts in such a way that all of the six original cube map faces share all their borders with another face (see the additional "E" and "B" faces to the left and right of the centre row, and the inverted "E"-faces at top and bottom in Fig. 8(a)).

We then compute the saliency map for the whole extended cutout at once (see Fig. 9(a)), extract the maps for all the cube faces of the


(a)

(b)

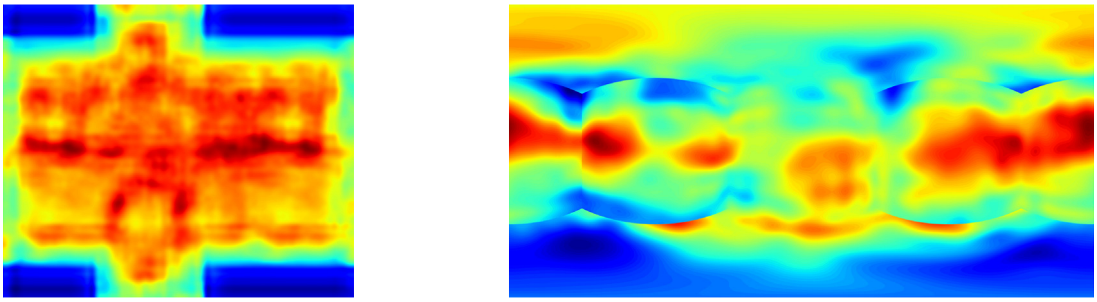
Fig. 7. An example of an equirectangular saliency map assembled from the individual saliency maps for the cube faces 7(a) and a combination of five such maps, produced at different cube rotation angles 7(b).



(a)

(b)

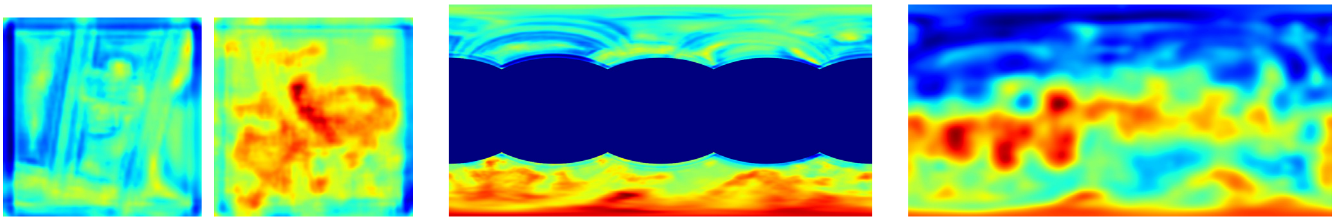
Fig. 8. Extended cutout construction scheme 8(a) and an image example 8(b). The "main", not-extended cutout is highlighted in light green on 8(a) and in red on 8(b). A filled cutout consists of all the shaded cube faces in 8(a).



(a)

(b)

Fig. 9. An example extended cutout raw saliency map 9(a) and its respective equirectangular projection 9(b).



(a)

(b)

(c)

Fig. 10. For the input image in Fig. 1(a): saliency maps for its top and bottom cube faces 10(a), their combined projection onto the partial equirectangular map 10(b), and the final saliency map 10(c), achieved by taking a pixel-wise maximum of the maps in Figs. 6 and 10(b), plus blurring. Note that value ranges for Figs. 6 and 10(b) are different.

"main" cutout and project them back onto the equirectangular map (see Fig. 9(b)).

This approach preserves the global context of the scene, even though it over-represents parts of the panorama (in particular, the top and the bottom faces are repeated more than the rest; if these contain highly salient objects, this can have noticeable effects on the final prediction). Distortions are cancelled out, but the stitching in Fig. 8(b) is not perfect (e.g. "A" in Fig. 8(a) wrongly borders on rotated versions of itself in order to fulfil continuity constraints for "B" and "D"). This interpretation also has the scene continuity information built into the cutout, since the objects at the borders of the main cutout are now augmented with the scene parts from the neighbouring cube faces, thus preserving local context. These trade-offs and limitations can be partly visually observed in the saliency maps produced with this interpretation (see Fig. 9).

We experimentally concluded that the extended cutout was the best cube map-based interpretation we considered (see Section 4.2).

2.5. Combined interpretation

With this interpretation, we try to combine the benefits of both ideas above: the **continuity-aware** interpretation makes use of all the available contextual information in an equirectangular image without any artificial over-representation, while the **cube map** interpretation helps undo the distortions introduced by the projection, as well as does away with border effects at the top and the bottom of the input image.

The idea here is to now use the cube map interpretation for the two most distorted cube faces only: the top and the bottom ones ("A" and "F" in Fig. 8(a)). The two resulting saliency maps (see Fig. 10(a)) are projected onto the partial equirectangular map (see Fig. 10(b)), and then combined (see Fig. 10(c)) with the full saliency map produced by the continuity-aware approach (as in Fig. 6). This interpretation was used to give example visualizations for the pipeline of our approach in Fig. 5, so it can be consulted for a better overview.

This way, the resulting map (in Fig. 10(c)) has no left or right vertical border artefacts due to the continuity-awareness, and no horizontal

Table 1
The overview of the used 2D image saliency models' performance, as the rank of each respective model in the MIT300 benchmark [27].

	KLD[a] rank	CC[a] rank	NSS[a] rank	AUC[a] rank	Balanced AUC rank
GBVS	**9**	27	28	22	14
eDN	43	35	39	18	**7**
SAM-ResNet	59	**4**	**2**	**5**	30

[a] These metrics were also used in the "Salient360!" Grand Challenge [18,25].

border artefacts due to the top and the bottom cube map faces being processed separately. The distortions are addressed where it is needed the most, and the scene context was not disbalanced during prediction.

3. Experimental methods

In this section we outline the experiments we performed and the evaluation procedures employed in the context of this work.

3.1. Data set

The data sets used in this work were provided by the "Salient360!" Grand Challenge at the IEEE International Conference on Multimedia & Expo (ICME) 2017 [18,25]. For head–eye saliency (i.e. for each viewport, the direction of eye gaze was considered; this is a natural extension of regular 2D saliency for the 360°-image domain), a training set of 40 images and corresponding scanpaths and fixation heat maps were provided. During the eye tracking recordings, the images were presented for 25 s with identical starting observation direction for all observers (at least 40 for each image). The stimuli were presented with an HMD Oculus-DK2 at 75 Hz and with a resolution of 960 × 1080 px per eye. Gaze data was recorded binocularly with an SMI tracker at 60 Hz.

The test set consisted of 25 spherical images, with their respective ground truth collected under conditions identical to those of the training set. Both the test image set and its ground-truth empirical saliency maps were hidden at the time of submission to the Grand Challenge.

All the 360° images and heat maps were represented as flat 2D images through the equirectangular projection. Scanpath coordinates were also given relative to this projection. An example image of the data set that visualizes this projection is shown in Fig. 1, along with its empirical saliency map.

3.2. Evaluation

For evaluation, the Grand Challenge used four saliency map metrics [18,25]: (i) two *density-based* metrics, which compare the entire saliency map to the empirical "ground truth" map: Kullback–Leibler divergence (KLD) and Correlation Coefficient (CC), and (ii) two *location-based* metrics, which consider only a set of selected locations on the saliency map: Normalized Scanpath Saliency (NSS) and Area Under the Curve (AUC, no class balancing; it technically considers the entire set of pixels of the saliency map by sampling all the possible locations, but the thresholds for building the Receiver Operating Characteristic (ROC) only iterate through the values at fixated locations).

3.3. Saliency predictors

As for this work we focused on already existing pre-trained models for image saliency prediction, we took three different, well-performing open-source models from the MIT300 image saliency benchmark [26,27] (probably the most widespread and established benchmark for image saliency; the ground truth saliency maps are not publicly available, and each submitted model is evaluated by the benchmark organizers, after which the scores with respect to eight popular quality metrics are published on the website). No additional training was performed.

Small modifications were applied to all the models (where possible and necessary) in order to (i) support varying image ratios by implementing adaptive downscale parameter choice (since the original images are 1:2, and our input interpretations in Section 2 additionally produce 1:1, 3:4 and 5:6 images, scaling all of them to one size would impede accurate saliency prediction); (ii) yield saliency maps without any post-processing, such as blurring and normalization (which would otherwise make the saliency values incomparable when combining several saliency maps into one); and (iii) store saliency maps to disk using matrix-based formats instead of images to avoid 8-bit quantization.

Below we describe the three literature models that were used in this work, in chronological order. Graph-based visual saliency (GBVS) was introduced in 2006 [20]. This approach uses a set of Gabor filter responses, local contrast, and luminance maps as features on several spatial scales. The feature maps are heavily downsampled, after which sophisticated activation and normalization steps are applied.

Ensemble of deep networks (eDN), introduced in 2014 [21], was a precursor of the deep learning methods for saliency prediction that have afterwards become very popular. The model's architecture can be represented as a combination of six multilayer structures (one to three layers) of operations that were inspired by their biological counterparts that take place in the visual cortex. Both the final combination and each individual layered structure of the richly-parameterized operations were obtained through hyper-parameter optimisation. A simple linear classifier is used to distinguish salient and non-salient image locations.

Saliency Attentive Model (SAM) is a recently (in 2016) introduced model [22] that extracts image features via a dilated ResNet architecture [28] (in the version used for this work; the framework also includes an option to use dilated VGG-16 [29] for feature extraction). It then employs a convolutional Long Short-Term Memory (LSTM) network, which recurrently attends to different locations of the feature tensor.

As saliency prediction is a multifaceted problem, there is no one definitive metric for model evaluation, and hence no one best model. If we use the well-established MIT300 benchmark [26,27] to compare the three models listed above, each of them comes out on top of the others according to at least one metric. Table 1 contains an overview of the models' performance in the form of their ranks (out of 74 models) with respect to several metrics [30] (the ranking snapshot was taken on the date of the Grand Challenge submission deadline, May 2017). It can be seen that all the models have their strengths and weaknesses, but SAM-ResNet is probably the more consistently well-performing one.

To enhance the performance of our saliency prediction, we also combined the final saliency maps generated by the three models above. The benefits of combining several saliency predictions into one have been thoroughly discussed in [31], as well as earlier in [32]. Taking the mean of the predicted saliency maps falls under the category of non-learning based approaches described in [31], and was shown to outperform all of the baseline saliency models, especially when averaging only over a small set of best performers. The work in [32] only considered summation (with different weighting schemes) and multiplication approaches, concluding that the simple mean performed best. We therefore computed the average of the final saliency maps produced with all three base saliency predictors (after the normalization step).

Table 2
Training-set performance of the cube map interpretation variations (with eDN as saliency predictor)

Metric	Filled cutout		Cube faces		Cube faces (5 rotations)		**Extended cutout**
KLD	0.76	≺	0.74	≺	0.71	≺	**0.69**
CC	0.28	≺	0.33	≈	0.33	≺	**0.35**
NSS	0.30	≺	0.31	≺	0.40	≺	**0.50**
AUC	0.59	≈	0.59	≺	0.61	≺	**0.64**

The symbol ≺ indicates inferiority of the number on the left to the number on the right (i.e. greater for KLD and lower for the rest of the metrics).

Table 3
Saliency maps evaluation results, depending on the equirectangular image interpretation and the saliency predictor model. Best results for each metric are boldified.

Metric	Predictor	Continuity-aware	Extended cutout	Combined
KLD	GBVS	0.67	0.76	0.66
	eDN	0.67	0.64	0.62
	SAM-ResNet	0.55	0.74	0.48
	average	0.50	0.58	**0.45**
CC	GBVS	0.35	0.29	0.35
	eDN	0.41	0.40	0.43
	SAM-ResNet	0.54	0.31	0.56
	average	0.55	0.41	**0.58**
NSS	GBVS	0.73	0.46	0.64
	eDN	0.75	0.63	0.67
	SAM-ResNet	0.84	0.56	0.70
	average	**0.92**	0.69	0.81
AUC	GBVS	0.71	0.64	0.70
	eDN	0.72	0.68	0.69
	SAM-ResNet	**0.75**	0.67	0.71
	average	**0.75**	0.69	0.73

3.4. Experiments

In our work we tested various combinations of interpretations (see Section 2) and saliency predictors (see Section 3.3). Most of the preliminary experiments were performed with eDN, whereas the final selection of interpretations was tested with all the models. We selected a subset of interesting combinations for submission to the Grand Challenge.

4. Results and discussion

First, we here discuss the limitations and related preliminary experiment of each interpretation group. Section 4.4 summarises the performance figures of all evaluated saliency predictors.

4.1. Continuity-aware interpretation

This is the simplest approach of the ones we have used, which essentially changes the location of the vertical border in the equirectangular image by rotating the spherical image representation by 180° in the horizontal plane. Since we do not know whether any objects happen to be located at the stitching line, neither before nor after the rotation, we simply combine the saliency maps produced for the original image and the shifted one.

Another approach here could be finding such a stitching point on the image, where no object would be bisected, and only predicting the saliency map for one equirectangular image. It is, however, not guaranteed that such a point always exists, and the resulting saliency map would still have noticeable visually unnatural artefacts near the stitching line.

A similar approach could be additionally applied to eliminate vertical borders, but this requires more complex spherical image manipulations (e.g. converting to a cube map, rotating by 90° in the respective plane, and projecting back onto the equirectangular surface, with corresponding reverse transformations taking place after saliency prediction), whereas this interpretation was intended as the simplest way of incorporating additional information into the prediction process.

4.2. Cube map interpretations

As described in Section 2.4, there are multiple ways to use a cube map to produce equirectangular saliency maps. We evaluated (on the training set) four of them to find the best one: individual cube faces (as in Fig. 7(a)), individual cube faces at five different rotations of the spherical image (same as in Fig. 7(b)), filled cutout (the shaded areas in Fig. 8(a)), and extended cutout (all cube faces in Fig. 8(a)). Their performance figures are summarised in Table 2. The trend is the same for all the four metrics: a filled cutout is inferior to using the individual cube map faces, which is in turn improved by using several rotated versions of the cube map, and the extended cutout outperforms the rest (marked in bold in the table).

4.3. Combined interpretation

For this interpretation, we have additionally experimented with the way of computing the saliency maps for the top and the bottom cube faces: either separately, or as part of an extended cutout. The former approach proved to outperform the latter with big margins (on the training set, with eDN used for saliency prediction): 0.65 vs. 0.57 AUC, 0.36 vs. 0.29 CC, 0.68 vs. 0.75 KLD, 0.53 vs. 0.24 NSS, respectively.

One adjustment we had to make for this approach was concerning one of the saliency predictors (namely SAM-ResNet), which in this set-up tended to over-represent the top and bottom cube planes (see Fig. 11(b)), probably because of the lacking context. We therefore attempted to quantitatively examine this disbalance. To this end, we split each of the resulting saliency maps in two parts: part A — the middle third (horizontally) — and part B — the rest of the map. We then computed the ratio of the maximal saliency value in part B to that in part A for each individual saliency map.

It turned out that the ground truth maps and both the eDN and the GBVS saliency maps (produced via the combined interpretation) all had the mean of these ratios around 1 (0.73 for the ground truth to 1.16 for GBVS). For the SAM-ResNet saliency maps it was, however, 4.51. We therefore divided all the values in the partial (for the top and the bottom cube map faces, see Fig. 10(b)) equirectangular SAM saliency map by this coefficient prior to combining it with the continuity-aware saliency maps (see Fig. 11(c)). The improvement of this rescaling is again quantitatively noticeable: 0.68 vs. 0.62 AUC, 0.53 vs. 0.4 CC, 0.51 vs. 0.7 KLD, 0.48 vs. 0.1 NSS, with and without this modification, respectively.

4.4. All results

For a more complete evaluation of our approach, we can consider using each of the selected 360°-image interpretations (i.e. *continuity-aware*, *extended cutout* and *combined*) with each of the employed saliency predictors (i.e. GBVS, eDN, SAM-ResNet, and their average) in turn. The full table for all results of our predictor–interpretation pairs can be found in Table 3.

Additionally, to any of the resulting saliency maps we can optionally add the mean ground truth saliency map (of the training set) with a certain weight. We empirically determined 0.2 to be a good choice. This way, we explicitly take into account the “vertical centre bias” that was observed in Fig. 3(b). This gives us a total of 24 models.

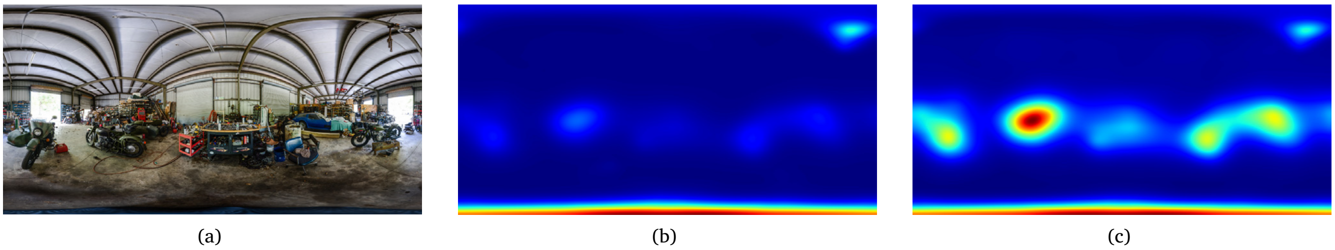


Fig. 11. The input image 11(a), the respective SAM-ResNet saliency maps produced with the **combined interpretation** without 11(b) and with 11(c) the rescaling factor for the partial saliency map.

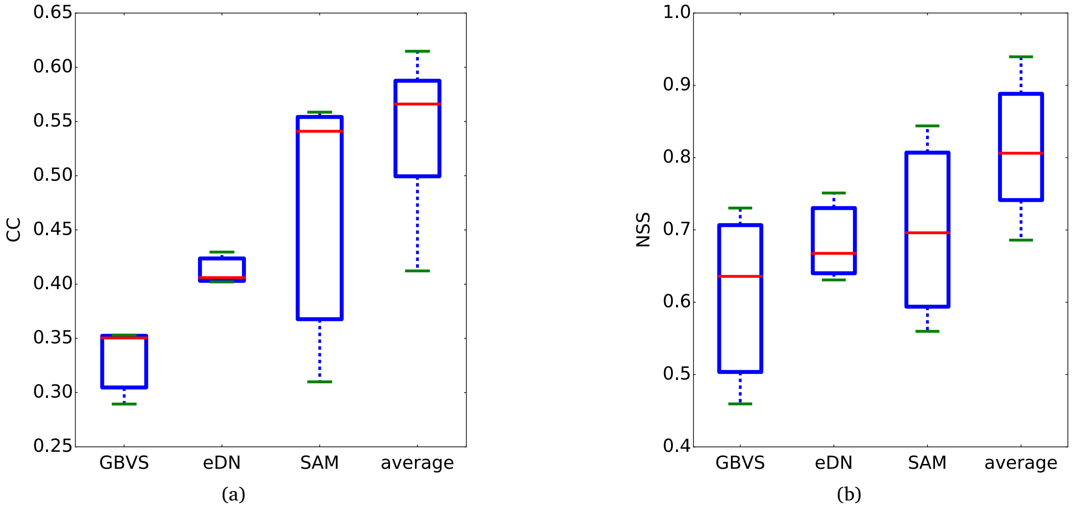


Fig. 12. Performance summary of all the models, split by the saliency predictor: correlation coefficient 12(a) and normalized scanpath saliency 12(b). A similar trend is observed for the other metrics as well.

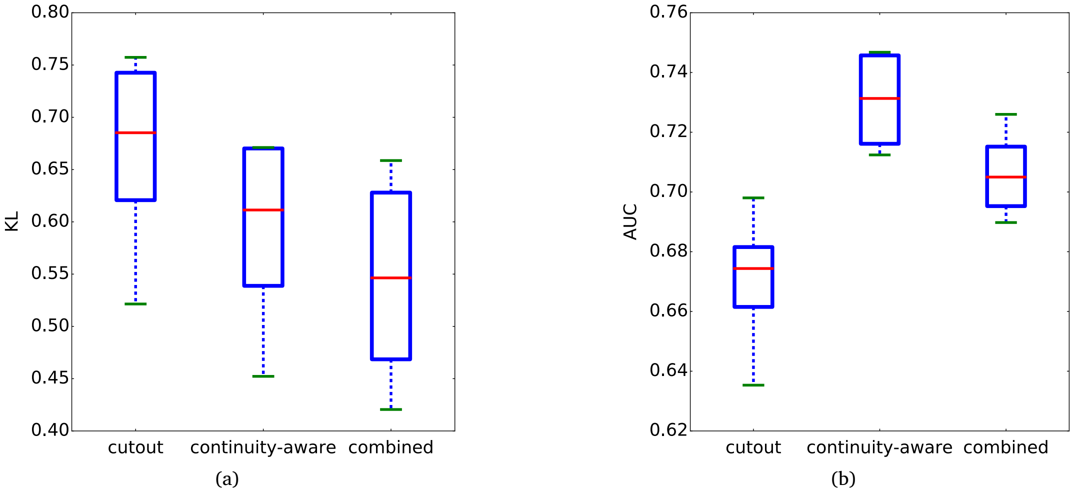


Fig. 13. Performance summary of all the models, split by the input image interpretation: Kullback–Leibler divergence (13(a), similar results for correlation coefficient) and area under the curve (13(b), similar results for normalized scanpath saliency).

To better analyse the evaluation results, we can differently group them: If we group the entire set of models by the saliency predictor, we can see that the "newer" model's performance is consistently superior to that of an "older" one, while the average model outperforms all of the individual models (see Fig. 12).

If we now group by the interpretation method, the conclusions become less clear-cut. For both density-based metrics, the combined interpretation performs best, followed by the continuity-aware interpretation (see Fig. 13(a)). For both location-based metrics, the continuity-aware interpretation is now the one in the lead, closely followed by the combined interpretation (see Fig. 13(b)).

For the average saliency predictor, however, it turned out that some of these differences were not statistically significant, and so *the combined interpretation with the average saliency predictor* was ranked 1st for all the

Table 4
The "Salient360!" Grand Challenge official unbiased results for the Head–Eye Saliency track, top-5 snippet and our extended cutout interpretation-based model. The rank (within each metric) was only increased if the difference between the respective sets of performance figures was statistically significant. 16 models were submitted to the challenge in total, with the worst average rank of 14.25.

	KLD (rank)	CC (rank)	NSS (rank)	AUC (rank)	mean rank
Combined interp. + avg. saliency model + centre bias	0.42 **(1)**	0.62 **(1)**	0.81 **(1)**	0.72 **(1)**	**1**
Combined interp. + avg. saliency model	0.45 **(1)**	0.58 **(1)**	0.81 **(1)**	0.73 **(1)**	**1**
Zhu et al. [33]	0.48 **(1)**	0.53 (6)	0.92 **(1)**	0.74 **(1)**	2.25
Ling et al. [34]	0.51 (5)	0.54 (6)	0.94 **(1)**	0.74 **(1)**	3.25
Continuity-aware interp. + avg. saliency model	0.50 (5)	0.55 (6)	0.92 **(1)**	0.75 **(1)**	3.25
…	…	…	…	…	…
Extended cutout interp. + avg. saliency model	0.58 (5)	0.41 (12)	0.69 (8)	0.69 (6)	7.75

metrics in the "Salient360!" Grand Challenge [25], for some metrics tied in the first place with several other approaches, including *the continuity-aware interpretation with the average saliency predictor* (see Table 4).

It is also interesting to note that the worst (on average) saliency predictor – GBVS – in combination with the best (on average) interpretation – combined – performs better than the best (on average) predictor – SAM-ResNet – with the worst (on average) interpretation – the extended cutout.

All the qualitative results were reproduced both on the training and the test set. We see that the optimal choice of the interpretation can depend on the metric choice, but the combined interpretation generally fares rather well, delivering the best-ranked results (out of the models submitted before the test set was released) for all metrics at the "Salient360!" Grand Challenge in the "Head–Eye saliency prediction" track [18,25]. It also yields the best (in terms of absolute values) average scores for KLD and CC metrics across all submitted models.

Naturally, saliency prediction can benefit from specialized models, which were trained with the information about the equirectangular format of the images and the 360° nature of the scenes in mind, so training a dedicated model for this kind of stimuli is still worthwhile. It seems, however, that using pretrained state-of-the-art image saliency predictors to tackle the 360°-scene saliency prediction problem could suffice, at least as a first approximation, for some applications. For a minimal-effort model, one can therefore focus on an appropriate stimulus interpretation rather than on developing and training a whole new prediction model. Combining input interpretations and dedicated training procedure may yield even better results.

The source code of our approach is publicly available at http://www.michaeldorr.de/salient360.

5. Conclusion

In this work we have explored the applicability of regular image saliency models for the panoramic image case with a full 360° field of view. To this end we proposed several ways of "interpreting" the input equirectangular image, which would deal with the projection-related issues. We used three well-performing regular 2D image saliency predictors (and their combination via averaging). Our best-performing input interpretation is a combination of the continuity-aware and the cube map approach, and requires computing four saliency maps: one for the frontal equirectangular view, one for the "rear view" (i.e. looking backwards from the starting viewing position), and one saliency map for each of the top and the bottom cube map faces. Combined with the average saliency predictor, this took the first prize at the Head–Eye Saliency Prediction track of the "Salient360!" Grand Challenge.

Acknowledgements

This research was supported by the Elite Network Bavaria, funded by the Bavarian State Ministry for Research and Education.

References

[1] N. Ouerhani, J. Bracamonte, H. Hugli, M. Ansorge, F. Pellandini, Adaptive color image compression based on visual attention, in: Proceedings 11th International Conference on Image Analysis and Processing, 2001, pp. 416–421. http://dx.doi.org/10.1109/ICIAP.2001.957045.

[2] C. Guo, L. Zhang, A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression, IEEE Trans. Image Process. 19 (1) (2010) 185–198. http://dx.doi.org/10.1109/TIP.2009.2030969.

[3] K. Rapantzikos, Y. Avrithis, S. Kollias, Dense saliency-based spatiotemporal feature points for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1454–1461. http://dx.doi.org/10.1109/CVPR.2009.5206525.

[4] E. Vig, M. Dorr, D. Cox, Space-variant descriptor sampling for action recognition based on saliency and eye movements, in: Computer Vision — ECCV 2012: 12th European Conference on Computer Vision, Proceedings, Part VII, Florence, Italy, October 7–13, 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 84–97. https://doi.org/10.1007/978-3-642-33786-4_7.

[5] Y. Wei, X. Liang, Y. Chen, X. Shen, M.M. Cheng, J. Feng, Y. Zhao, S. Yan, STC: A simple to complex framework for weakly-supervised semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (11) (2017) 2314–2320. http://dx.doi.org/10.1109/TPAMI.2016.2636150.

[6] S. Wang, M. Jiang, X. Duchesne, E. Laugeson, D. Kennedy, R. Adolphs, Q. Zhao, Atypical visual saliency in autism spectrum disorder quantified through model-based eye tracking, Neuron 88 (3) (2015) 604–616. http://dx.doi.org/10.1016/j.neuron.2015.09.042. http://www.sciencedirect.com/science/article/pii/S0896627315008314.

[7] J.E. Silberg, I. Agtzidis, M. Startsev, T. Fasshauer, K. Silling, A. Sprenger, M. Dorr, R. Lencer, Free visual exploration of natural movies in schizophrenia, Eur. Arch. Psychiatry Clin. Neurosci. (2018) 1–12. https://doi.org/10.1007/s00406-017-0863-1.

[8] E. Goffman, Behavior in Public Places, Simon and Schuster, 2008.

[9] T. Foulsham, E. Walker, A. Kingstone, The where, what and when of gaze allocation in the lab and the natural environment, Vis. Res. 51 (17) (2011) 1920–1931. http://dx.doi.org/10.1016/j.visres.2011.07.002. http://www.sciencedirect.com/science/article/pii/S0042698911002392.

[10] M. Assens, K. McGuinness, X. Giro-i-Nieto, N.E. O'Connor, SaltiNet: Scan-path prediction on 360 degree images using saliency volumes, 2017, pp. 1–8. ArXiv e-prints arXiv:1707.03123.

[11] M.A. Reina, K. McGuinness, X. Giro-i-Nietro, N.E. O'Connor, Scanpath and saliency prediction on 360 degree images, Signal Process., Image Commun. (2018).

[12] A.D. Abreu, C. Ozcinar, A. Smolic, Look around you: Saliency maps for omnidirectional images in VR applications, in: Ninth International Conference on Quality of Multimedia Experience, QoMEX, 2017, pp. 1–6. http://dx.doi.org/10.1109/QoMEX.2017.7965634.

[13] E. Upenik, M. Řeřábek, T. Ebrahimi, Testbed for subjective evaluation of omnidirectional visual content, in: Picture Coding Symposium, PCS, 2016, pp. 1–5. http://dx.doi.org/10.1109/PCS.2016.7906378.

[14] G.T. Buswell, How People Look at Pictures: A Study of the Psychology of Perception in Art, University of Chicago Press, Chicago, 1935.

[15] B.W. Tatler, The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions, J. Vision 7 (14) (2007) 4. http://dx.doi.org/10.1167/7.14.4. arXiv:/data/journals/jov/932846/jov-7-14-4.pdf.

[16] P.-H. Tseng, R. Carmi, I.G.M. Cameron, D.P. Munoz, L. Itti, Quantifying center bias of observers in free viewing of dynamic natural scenes, J. Vision 9 (7) (2009) 4. http://dx.doi.org/10.1167/9.7.4. arXiv:/data/journals/jov/932863/jov-9-7-4.pdf.

[17] M. Dorr, T. Martinetz, K.R. Gegenfurtner, E. Barth, Variability of eye movements when viewing dynamic natural scenes, J. Vision 10 (10) (2010) 28. http://dx.doi.org/10.1167/10.10.28. arXiv:/data/journals/jov/932797/jov-10-10-28.pdf.

[18] Y. Rai, J. Gutiérrez, P. Le Callet, A dataset of head and eye movements for 360 degree images, in: Proceedings of the 8th ACM on Multimedia Systems Conference, MMSys'17, ACM, New York, NY, USA, 2017, pp. 205–210. http://dx.doi.org/10.1145/3083187.3083218. http://doi.acm.org/10.1145/3083187.3083218.

[19] T. Judd, K. Ehinger, F. Durand, A. Torralba, Learning to predict where humans look, in: IEEE 12th International Conference on Computer Vision, 2009, pp. 2106–2113. http://dx.doi.org/10.1109/ICCV.2009.5459462.

[20] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: Advances in Neural Information Processing Systems, 2007, pp. 545–552.

[21] E. Vig, M. Dorr, D. Cox, Large-scale optimization of hierarchical features for saliency prediction in natural images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2798–2805.

[22] M. Cornia, L. Baraldi, G. Serra, R. Cucchiara, Predicting human eye fixations via an lstm-based saliency attentive model, CoRR (2016) 1–13. abs/1611.09571. arXiv:1611.09571 http://arxiv.org/abs/1611.09571.

[23] M. Yu, H. Lakshman, B. Girod, A framework to evaluate omnidirectional video coding schemes, in: IEEE International Symposium on Mixed and Augmented Reality, 2015, pp. 31–36. http://dx.doi.org/10.1109/ISMAR.2015.12.

[24] X. Corbillon, G. Simon, A. Devlic, J. Chakareski, Viewport-adaptive navigable 360-degree video delivery, 2016, pp. 1–7. ArXiv e-prints arXiv:1609.08042.

[25] J. Gutiérrez, E. David, Y. Rai, P. Le Callet, Toolbox and dataset for the development of saliency and scanpath models for omnidirectional/360° still images, Signal Process., Image Commun. (2018).

[26] T. Judd, F. Durand, A. Torralba, A benchmark of computational models of saliency to predict human fixations, in: MIT Technical Report, 2012.

[27] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, A. Torralba, MIT Saliency Benchmark.

[28] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778.

[29] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, pp. 1–14. ArXiv e-prints arXiv:1409.1556.

[30] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, F. Durand, What do different evaluation metrics tell us about saliency models?, CoRR (2016) 1–24. abs/1604.03605. arXiv:1604.03605 http://arxiv.org/abs/1604.03605.

[31] J. Wang, A. Borji, C.C.J. Kuo, L. Itti, Learning a combined model of visual saliency for fixation prediction, IEEE Trans. Image Process. 25 (4) (2016) 1566–1579. http://dx.doi.org/10.1109/TIP.2016.2522380.

[32] A. Borji, D.N. Sihite, L. Itti, Salient object detection: A benchmark, in: Proceedings of the 12th European Conference on Computer Vision, Volume Part II, ECCV'12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 414–429. http://dx.doi.org/10.1007/978-3-642-33709-3_30.

[33] Y. Zhu, G. Zhai, X. Min, The prediction of head and eye movement for 360 degree images, Signal Process., Image Commun. (2018).

[34] J. Ling, K. Zhang, Y. Zhang, D. Yang, Z. Chen, A saliency prediction model on 360 degree images using color dictionary based sparse representation, Signal Process., Image Commun. (2018).

F

Supersaliency: Predicting Smooth Pursuit-based Attention

As a foundation of this work, we have noted that while saliency prediction for videos is often referred to as "fixation prediction", smooth pursuit (SP) is responsible for a non-negligible part of dynamic viewing behaviour [47, 53, 4*, 9†]. Fixation detectors used in the literature to create the ground truth are often supplied with the eye tracker [71, 82] or never described in the respective papers [70, 94]. Given the relative rarity of SP detectors, and the fact that SP is almost never mentioned in the saliency data set or model articles (and never, as of yet, detected by the eye trackers' software), it is clear that the fields of video saliency and eye movement classification have developed mostly in parallel, with the most recent advancements in the latter field not affecting the former.

To amend this lack of synchronisation, we used our recent eye movement classification framework [4*] in order to systematically differentiate between gaze events in the eye tracking recordings, which form the ground truth sets for saliency prediction. Having obtained the labels of this algorithm, we proposed to separately consider two formulations of the video attention prediction problem: fixation prediction and SP prediction. We refer to these as saliency and supersaliency prediction, the latter name owing to the much greater selectivity of SP, as well as its other properties that we described in this paper.

We tested two different saliency model architectures (proof-of-concept and end-to-end deep networks), training these to predict either saliency or supersaliency ground truth maps. For both architectures, training to predict SP yielded saliency predictors with better generalisation properties: When tested directly without fine-tuning, the scores of supersaliency-trained models on two independent unseen data sets were consistently higher than the corresponding scores for the saliency-trained version of the same model. This demonstrates the potential of principled eye movement class separation for saliency modelling, and, in particular, the benefits of accounting for SP in the analysis.

My personal contributions consist of (i) conceiving the hypothesis and experimental design for this work; (ii) re-processing the saliency data sets to separate fixations and SPs in their ground truth; (iii) developing and testing the predictive models; (iv) motivating and performing adjustments in the saliency evaluation pipeline that is traditionally used in the literature in order to account for the sparsity of SP; (v) writing the manuscript.

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 30, 2019, accepted December 12, 2019, date of publication December 23, 2019, date of current version January 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961835

Supersaliency: A Novel Pipeline for Predicting Smooth Pursuit-Based Attention Improves Generalisability of Video Saliency

MIKHAIL STARTSEV AND MICHAEL DORR
Chair of Human-Machine Communication, Department of Electrical and Computer Engineering, Technical University of Munich, 80333 München, Germany
Corresponding author: Mikhail Startsev (mikhail.startsev@tum.de)

This work was supported in part by the Elite Network Bavaria, funded by the Bavarian State Ministry for Research and Education.

ABSTRACT Predicting attention is a popular topic at the intersection of human and computer vision. However, even though most of the available video saliency data sets and models claim to target human observers' fixations, they fail to differentiate them from smooth pursuits (SPs), a major eye movement type that is unique to perception of dynamic scenes. In this work, we strive for a more meaningful prediction and conceptual understanding of saliency in general. Because of the higher attentional selectivity of smooth pursuit compared to fixations modelled in traditional saliency research, we refer to the problem of SP prediction as ''supersaliency''. To make this distinction explicit, we (i) use algorithmic and manual annotations of SPs and fixations for two well-established video saliency data sets, (ii) train Slicing Convolutional Neural Networks for saliency prediction on either fixation- or SP-salient locations, and (iii) evaluate our and 26 publicly available dynamic saliency models on three data sets against traditional saliency and supersaliency ground truth. Overall, our models outperform the state of the art in both the new supersaliency and the traditional saliency problem settings, for which literature models are optimised. Importantly, on two independent data sets, our supersaliency model shows greater generalisation ability than its counterpart saliency model and outperforms all other models, even for fixation prediction. Furthermore, we tested an end-to-end video saliency model, which also showed systematic improvements when smooth pursuit was predicted either exclusively or together with fixations, with the best performance achieved when the model was trained for the supersaliency problem. This demonstrates the practical benefits and the potential of principled training data selection based on eye movement analysis.

INDEX TERMS Eye movements, saliency, smooth pursuit prediction.

I. INTRODUCTION

Saliency prediction has a wide variety of applications, be it in computer vision, robotics, or art [1], ranging from image and video compression [2], [3] to such high-level tasks as video summarisation [4], scene recognition [5], or human-robot interaction [6]. Its underlying idea is that in order to efficiently use the limited neural bandwidth, humans sequentially sample informative parts of the visual input with the high-resolution centre of the retina, the *fovea*. The prediction of gaze should thus be related to the classification of informative and uninformative video regions. However, humans use two different processes to foveate visual content. During fixations, the eyes remain mostly stationary; during smooth pursuit (SP), in contrast, a moving target is tracked by the eyes to maintain foveation. Notably, SP is impossible without such a target, and it needs to be actively initiated and maintained. For models of attention, this is a critical distinction: Because the eyes are stationary (''fixating'') in their default state, ''spurious'' fixations may be detected even if a subject is not attentively looking at the input; SP, however, always co-occurs with attention. In addition, visual sensitivity seems to be improved during SP (e.g. higher chromatic contrast sensitivity [7] and enhanced visual motion prediction [8]).

The ultimate goal of all eye movements and perception is to facilitate action in the real world. In a seminal paper [11], Land showed that gaze strategies, and SP in particular, play a critical role during many everyday activities. Similar results have been found for driving scenarios, where attention is crucial. Studies show that tangential [12] and target [13]

The associate editor coordinating the review of this manuscript and approving it for publication was Jianqing Zhu.

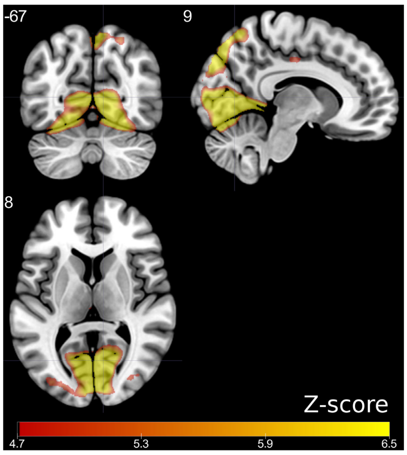


FIGURE 1. Empirically observed neurological differences between fixation and smooth pursuit: Large brain areas (highlighted) show significantly increased activation levels during pursuits compared to fixations (detected by [9]) in the *studyforrest* data set [10]; none demonstrate the inverse effects. A set of representative slices along orthogonal planes for a model brain is presented in this figure (slice numbers labelled on the figure) for the visualisation of the differences between fixation and pursuit conditions. Significance was determined via analysing the "standard score", or "Z-score" values.

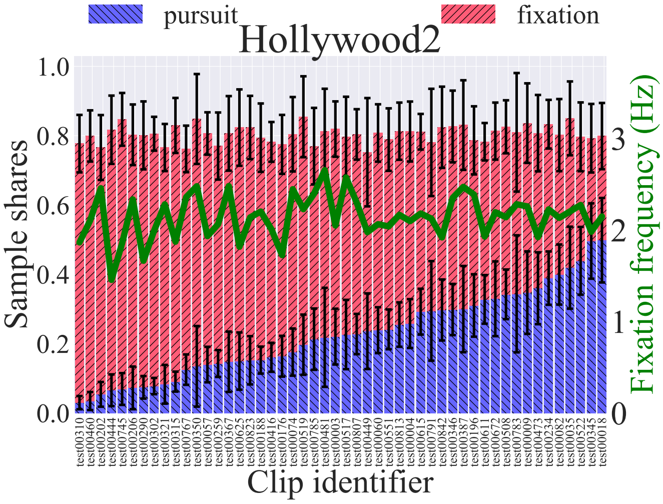


FIGURE 2. Behavioural differences between fixation and smooth pursuit: Saliency metrics typically evaluate against fixation onsets, which, as detected by a traditional approach [21] (green line), are roughly equally frequent across videos. However, applying a more principled approach to separating smooth pursuit from fixations [9] reveals great variation in the number of fixation (red bars) and pursuit (blue bars) samples (remaining samples are saccades, as well as blinks and other unreliably tracked samples).

locations during curve driving are "fixated" with what actually consists, in part, of SP. In natural driving, roadside objects are often followed with pure SP, without head motion [14]. Following objects that are moving relative to the car with gaze (by turning the head, via an SP eye movement, or a combination of both) is a clearer sign of attentive viewing, compared to the objects of interest crossing the line of sight.

In practice, it is difficult to segment the – often noisy – eye tracking signal into fixations and SPs, and thus many researchers combine all intervals where the eyes are keeping track of a point or an object into "fixations" [15]. Nevertheless, it is well established that e.g. individuals with schizophrenia show altered SP behaviour [16], [17], and recently new methods for gaze-controlled user interfaces based on SP have been presented [18]–[20]. This demonstrates some of the practical benefits of carefully separating the eye movements that make up the human gaze behaviour.

FIGURE 1 and FIGURE 2 show two analyses corroborating the importance of SP for models of attention in the context of a more tractable task of video watching. In FIGURE 1, data from the publicly available *studyforrest* data set[1] [10], which combine functional brain imaging and eye tracking during prolonged movie watching, were comparatively evaluated for SP vs. fixation episodes in a preliminary study. The highlighted voxels show that large brain areas are more active during SP compared to fixations; notably, no brain areas were more active during fixation than during SP. In other words, SP is representative of greater neurological engagement. The sparser selectivity of SP is demonstrated in FIGURE 2, where the relative share of SP and fixation gaze samples is plotted for 50 randomly selected clips from Hollywood2 [22]. Even though the number of traditionally detected fixations (but not their duration) is roughly the same for all clips, the amount of SP ranges from almost zero to half of the viewing time.

Taken together, these observations let us hypothesise that SP is used to selectively foveate video regions that demand greater cognitive resources, i.e. contain more information. In practice, automatic pursuit classification as applied to the *studyforrest* and Hollywood2 data sets may not be perfect, but the results in FIGURE 1 corroborate that even with potentially noisy detections, SP corresponds to higher brain activity, and thus to more meaningful saliency.

Therefore, explicitly modelling SP in a saliency pipeline should benefit the classification of informative video regions. Beyond a better understanding of attention, there might also be direct applications of SP prediction itself, e.g. in semi-autonomous driving (verification of attentive supervision), telemedicine (monitoring of SP impairment as a vulnerability marker for schizotypal personality disorder [23], e.g. during TV or movie watching [17]), or gaze-based interaction (analysis of potential distractors in user interfaces for AR/VR).

Despite the fundamental differences between SP and fixations, however, available saliency data sets ignore this distinction, and the computational models naturally follow suit [24], [25]. In fact, not one of the video saliency models we came across mentions the tracking of objects performed via SP, and the only data set we found to purposefully attempt separating SP from fixations is GazeCom [21], which simply discarded likely pursuits in order to achieve cleaner fixation detection.

We argue that processing the eye tracking recordings in a systematic and comprehensively described way in order

[1] These data were obtained from the OpenfMRI database. Its accession number is ds000113d.

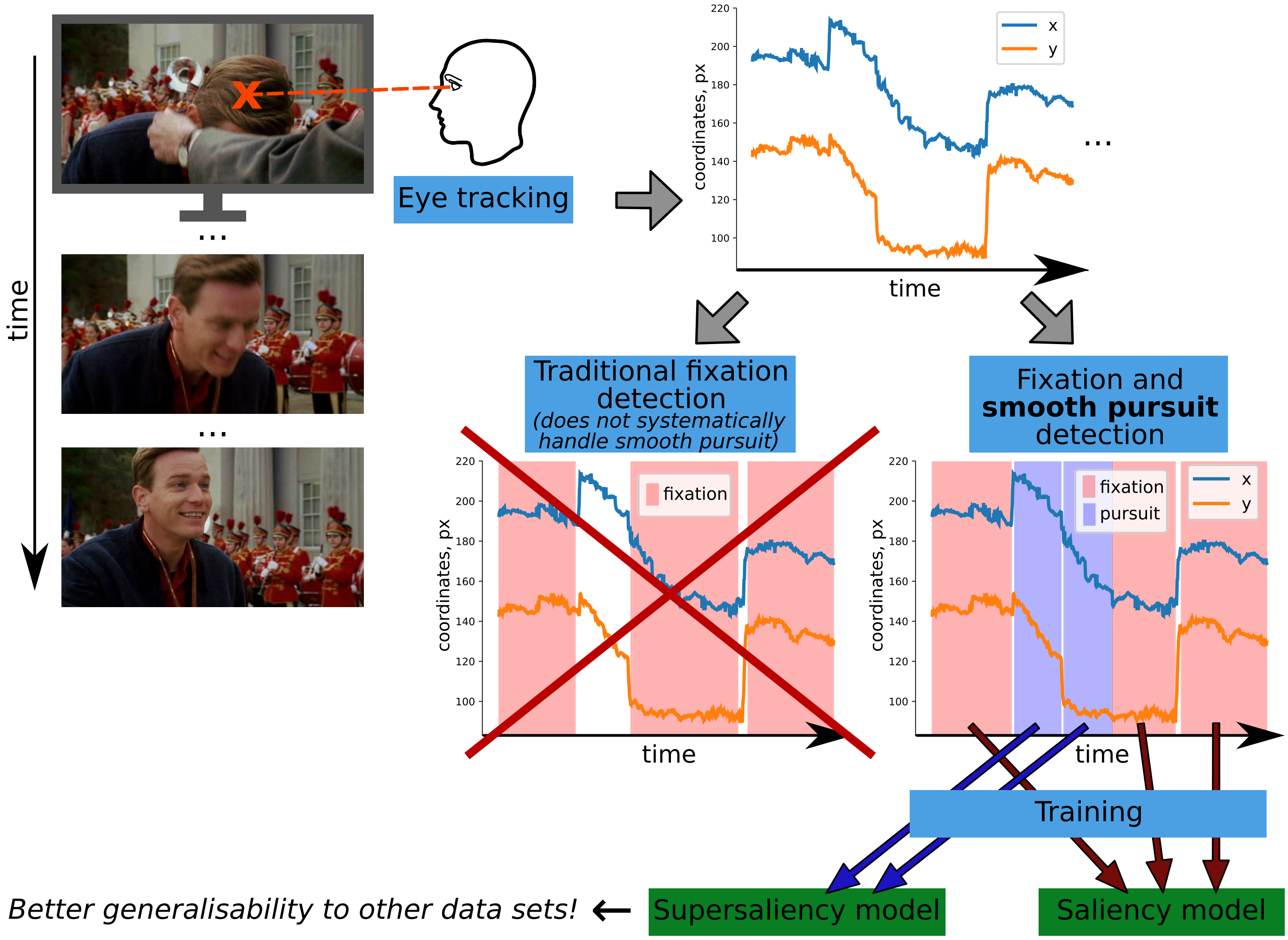

time
...
...
Eye tracking
coordinates, px
220
200
180
160
140
120
100
x
y
time
...
Traditional fixation detection
(does not systematically handle smooth pursuit)
fixation
Fixation and smooth pursuit detection
fixation
pursuit
Training
Better generalisability to other data sets!
Supersaliency model
Saliency model

FIGURE 3. Overview of the proposed pipeline.

to extract moments of attention, be that fixations or smooth pursuits, is a vital first step in any pipeline of modelling human attention. This would allow for saliency to be treated not as a purely computational challenge of predicting some heat map frames for a video input, but as a task that could help us better understand human perception and attention.

In this manuscript, we extend our previous work [26] and make the following contributions: First, we introduce the problem of smooth pursuit prediction – *supersaliency*, so named due to the properties separating it from traditional, fixation-based saliency (e.g. see FIGURE 1 and FIGURE 2). In this problem setting, the saliency map values correspond to how likely a certain input video location is to induce SP. We then provide automatically labelled [9], large-scale training and test sets for this problem (building on the Hollywood2 data set [22]), as well as a manually labelled, smaller-scale test set of more complex scenes in order to test the generalisability of saliency models (building on the GazeCom data set [21], [27]). For both, we provide SP-only and fixation-only ground truth saliency maps. We also discuss the necessary adjustments to the evaluation of supersaliency (and video saliency in general) due to its high inter-video variance, introducing weighted averaging of individual clip scores.

Furthermore, we propose a deep dynamic saliency model for (super)saliency prediction, which is based on the slicing convolutional neural network (S-CNN) architecture [28]. After training our proposed model for both saliency and supersaliency prediction on the same overall data set, we demonstrate that our models excel at their respective problems in the test subset of the large-scale data set, compared to over two dozen literature models. Finally, we show that training for predicting smooth pursuit reduces data set bias: The supersaliency-trained model better generalises to two independent sets (without any additional training) and performs best even for traditional saliency prediction. We demonstrate the same pattern with an additional, end-to-end video saliency model we introduce in this work.

The overview of the (super)saliency modelling pipeline we are proposing in this work can be seen in Figure 3.

II. RELATED WORK

Predicting saliency for images has been a very active research field. A widely accepted benchmark is represented by the

MIT300 data set [29], [30], which is currently dominated by deep learning solutions. Saliency prediction for videos, however, lacks an established benchmark. It generally is a challenging problem, because, in addition to larger computational cost, objects of interest in a dynamic scene may be displayed only for a limited time and in different positions and contexts, so attention prioritisation is more crucial.

Taking this prioritisation principle to the extreme, works on salient object *detection* typically attempt to identify an object of interest in each frame (usually the same throughout a processed video clip). Saliency *prediction*, on the contrary, is not attempting to identify a single attention centre in the video, but aims at predicting the overall distribution of attention in the video as a heat map sequence. The salient object detection task is, therefore, much closer to segmentation at its core, with the added aspect of automatically selecting the dominant object in the scene. Despite the difference in problem formulations, both video saliency prediction and salient object detection essentially belong to the class of video-to-video transformation tasks, so some methodology can be shared between the two. We therefore include several works on both problems in our literature overview, when the methods are either directly or potentially applicable to the problem posed in this study.

Somewhat bridging these two saliency-related areas, [31] enabled attention shifting in the domain of salient object detection. That work directly tied the annotated objects of interest to human gaze directions, and therefore allowed for the objects to become or stop being salient as the scene unfolds.

A. SALIENCY PREDICTION

A variety of algorithms has been introduced to deal with human attention prediction [1]. Video saliency approaches broadly fall into two groups: Published algorithms mostly operate either in the original pixel domain [2], [24], [32], [33] and its derivatives (such as optic flow [34] or other motion representations [35]), or in the compression domain [25], [36], [37]. Transferring expert knowledge from images to videos in terms of saliency prediction is consistent with pixel-domain approaches, and the mounting evidence that motion attracts our eyes contributed to the development of compression-domain algorithms.

Traditionally, from the standpoint of perception, saliency models are also separated into two categories based on the nature of the features and information they employ. Bottom-up models focus their attention (and assume human observers do the same) on low-level features such as luminance, contrast, or edges. For videos, local motion can also be added to the list, together with the video encoding information. Hence, all the currently available compression-domain saliency predictors are effectively bottom-up.

Top-down models, on the contrary, use high-level, semantic information, such as concepts of objects, faces, etc. These are notoriously hard to formalise. One way to do so would be to detect certain objects in the video scenes, as was done in [22], where whole human figures, faces, and cars were detected. Another way would be to rely on developments in deep learning and the field's endeavour to implicitly learn important semantic concepts from data. In [38], either RGB space or contrast features are augmented with residual motion information to account for the dynamic aspect of the scenes (i.e. motion is processed before the CNN stage in a handcrafted fashion). The work in [39] uses a 3D CNN to extract features, plus an LSTM network to expand the temporal span of the analysis. Other researchers use further additional modules, such as the attention mechanism [40] or object-to-motion sub-network [41]. In [42], a modified convolutional LSTM (using multi-scale dilations) is employed to accurately detect salient objects in video sequences. In a similar vein of research, [43] also modified the typical convolutional LSTM structure for video-to-video prediction by developing a parallel multi-dimensional extension of this structure. This modification allows for a much more complete utilisation of the relevant past information for each pixel. While our work does not focus on the architecture design, it would doubtlessly be interesting to explore the effects of systematically differentiating between fixations and smooth pursuits in the context of saliency prediction with a wider spectrum of computational models (our work tested two different approaches).

Whereas using a convolutional neural network in itself does not guarantee the top-down nature of the resulting model, its multilayer structure fits the idea of hierarchical computation of low-, mid-, and high-level features. A work by Krizhevsky *et al.* [44] pointed out that while the first convolutional layers learned fairly simplistic kernels that target frequency, orientation, and colour of the input signal, the activations in the last layer of the network corresponded to a feature space, in which conceptually similar images are close, regardless of the distance in the low-level representation space. Another study [45] concluded that, just like certain neural populations of a primate brain, deep networks trained for object classification create such internal representation spaces, where images of objects in the same category get similar responses, and images of differing categories get dissimilar ones. Other properties of the networks discussed in that work indicate potential insights into the visual processing system that can be gained from them.

B. VIDEO SALIENCY DATA SETS

A broad overview of existing data sets is given in [46]. Here, we dive into the aspect particularly relevant to this study – the identification of ''salient'' locations of the videos, i.e. how did the authors deal with dynamic eye movements. For the most part, this question is addressed inconsistently. The majority of the data sets either make no explicit mention of separating smooth pursuit from fixations (ASCMN [47], SFU [48], two Hollywood2-based sets [22], [49], DHF1K [40]) or rely on the event detection built into the eye tracker, which in turn does not differentiate SP from fixations (TUD [50], USC CRCNS [51], CITIUS [24], LEDOV [41]). IRCCyN/IVC (Video 1) [52] does not mention any eye movement types at all, whereas IRCCyN/IVC (Video 2) [53] only names SP in passing.

There are two notable exceptions from this logic. First, DIEM [54], which comprises video clips from a rich spectrum of sources, including amateur footage, TV programs, and movie trailers, so one would expect a hugely varying fixation–pursuit balance. The respective paper touches on the properties of SP that separate it from fixations, but in the end only distinguishes between blinks, saccades, and non-saccadic eye movements, referring to the latter as generic *foveations*, which combine fixations and SPs.

GazeCom [21], on the other hand, explicitly acknowledges the difficulty of distinguishing between fixations and smooth pursuits in dynamic scenes. The used fixation detection algorithm employed a dual criterion based on gaze speed and dispersion. However, the recently published manually annotated ground truth data [27] show that these coarse thresholds are insufficient to parse out SP episodes.

Part of this work's contribution is, therefore, to provide a large-scale supersaliency (SP) and saliency (fixations) data set based on Hollywood2, as well as establishing a pipeline for (super)saliency evaluation.

III. SALIENCY AND SUPERSALIENCY

In this section, we describe the methodology behind the (super)saliency prediction in this work. Our approach relies on two main components: A large-scale data set of human video free-viewing, where the raw eye tracking data are available, and a computational model. Such data set would allow us to analyse the gaze recordings to parse out the episodes of either fixations or smooth pursuits. The detected samples of the two eye movements can be then directly used to train the proposed model.

A. DATA SETS AND THEIR ANALYSIS

GazeCom [21], which we used because it is the only saliency data set that also provides full manual annotation of eye movement events [27], [55], contains eye tracking data for 54 subjects, with 18 dynamic natural scenes used as stimuli, around 20 seconds each. At over 4.5 total hours of viewing time, this is the largest manually annotated eye tracking data set that accounts for SP. A high number of observers and the hand-labelled eye movement type information make this a suitable benchmark set. FIGURE 4a displays an example scene, together with its empirical saliency maps for both fixations and smooth pursuits, and the same frames in saliency maps predicted by different models.

Hollywood2 [22], selected for its diversity and the sheer amount of eye tracking recordings, contains about 5.5 hours of video (1707 clips, split into training and test sets), viewed by 16 subjects. The movies have all types of camera movement, including translation and zoom, as well as scene cuts. While the full training subset was used, we randomly selected 50 clips from the test subset (same as in FIGURE 2) for testing all the models. Example frames and respective (super)saliency maps can be seen in FIGURE 4b. Since manual labelling is impractical due to the data set size (over 70 h of total viewing time), we used our publicly available toolbox [27] implementing a state-of-the-art SP and fixation detection algorithm [9], [55]. A large-scale evaluation of this toolbox was performed in [56], where it demonstrated excellent performance when compared to the GazeCom ground truth data, and generalised well to an independent data set.

CITIUS [24] was recently used for a large-scale evaluation of the state of the art in connection with a novel model (AWS-D). It contains both real-life and synthetic video sequences, split into subcategories of static and moving camera. For our evaluation, we used the real-life part, **CITIUS-R** (22 clips totalling ca. 7 minutes, 45 observers). Only fixation onset and duration data are provided by the authors, so SP analysis was impossible.

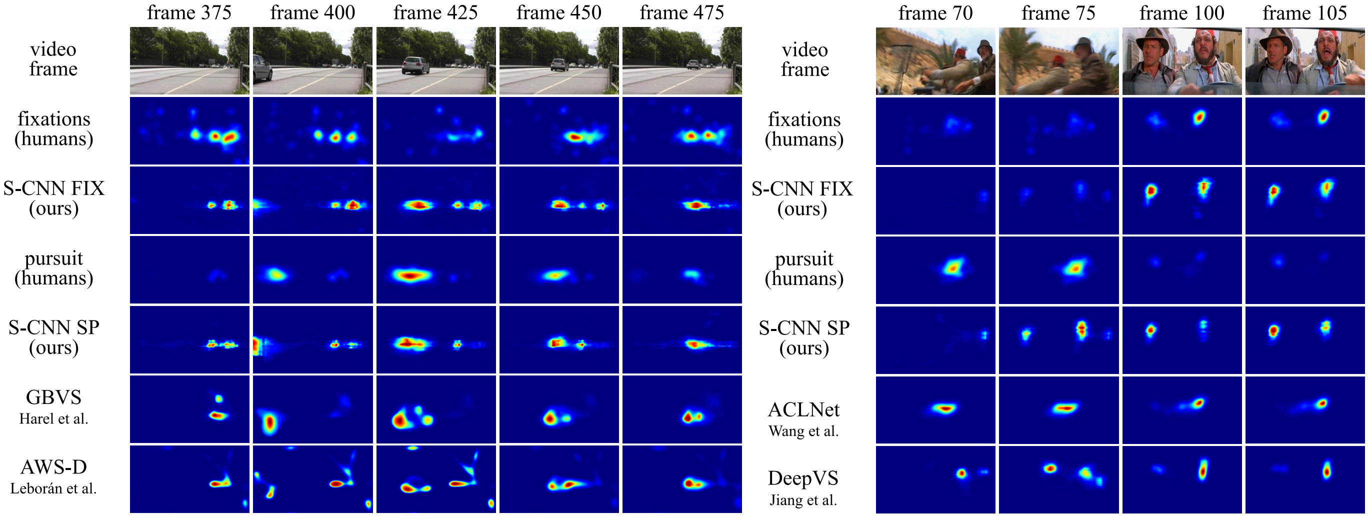
By definition, fixations are almost stationary, so that a single point (usually, mean gaze position placed at temporal onset) sufficiently describes an entire fixation. In line with the literature, we evaluated the prediction of such fixation onsets in the "onset" condition (detected by a standard algorithm [21] for GazeCom and Hollywood2, provided with the data set for CITIUS-R). Notably, the reference models are likely optimised for this problem setting.

To describe the trajectory of an SP episode, however, all its gaze samples need to be taken into account. Accordingly, both the GazeCom ground truth and the toolbox [27] we used for Hollywood2 provide sample-level annotations. These annotations were used for evaluating the prediction of pursuit-based attention in the "SP" condition, i.e. model predictions were tested against the set of individual pursuit gaze samples. The "FIX" condition utilised individual fixation samples as well (similar to [54]), and is, in principle, not very different from "onset". By directly mirroring the implementation of the "SP" condition, however, it allowed for a fairer comparison between the two.

B. SLICING CNN SALIENCY MODEL

We adopted the slicing convolutional neural network (S-CNN) architecture [28]. To achieve saliency prediction, we extended patch-based image analysis (e.g. [57] for image saliency, and [38] for individual video frames) to subvolume-based video processing. This way, we are still able to capture motion patterns, while maintaining a relatively straightforward binary classification-based architecture – (super)salient vs. non-salient subvolumes. Initially, we did not use more complex end-to-end approaches in order to keep the proof-of-concept implementation of fixation- and pursuit-based training as straightforward as possible, without intermediate steps of having to convert locations of corresponding samples into continuous saliency maps. These steps would introduce additional data parametrisation and, potentially, biases into the pipeline. However, we additionally validate the idea of supersaliency prediction with an end-to-end model in Section IV.

S-CNN [28] takes an alternative approach to extracting motion information from a video sequence. Instead of



(a) GazeCom (“street” clip)

(b) Hollywood2 (“actioncliptest00416” clip)

FIGURE 4. Frame examples from GazeCom (a) and Hollywood2 (b) videos (first row), with their respective empirical ground truth fixation-based saliency (second row) and smooth pursuit-based supersaliency (fourth row) ground truth maps. Algorithmic predictions (all identically histogram-equalised, for fair visual comparison) occupy the rest of the rows. The choice of saliency models for visual comparison was based on best average performance on the respective data set.

handcrafted motion descriptors [38], 3D convolutions [58], or recurrent structures [39], S-CNN achieves temporal integration by rotating the feature tensors after initial individual frame-based feature extraction. This way, time (frame index) is one of the axes of the subsequent convolutions. The architecture is based on VGG-16 [59], with the addition of dimension swapping operations and temporal pooling. The whole network would consist of three branches, in each of which the performed rotation is different, and the ensuing convolutions are performed in the planes *xy* (equivalent to no rotation), *xt*, or *yt* (branches are named respectively). Due to the size of the complete model, only one branch could be trained at a time. We decided to use the *xt*-branch for our experiments (see FIGURE 5), since it yielded the best individual results in [28], and the horizontal axis seems to be more important for human vision [60] and SP in particular [61]. We also tested the other branches separately and the late fusion of their results, but the *xt* branch was the best individual performer, and the fusion did not produce sufficient performance gains to justify the tripled computation time. Therefore, we do not report these results in this paper. Similarly, our preliminary tests with 3D-CNN architectures, similar to results in [28], led us to opt for the better-performing S-CNNs instead.

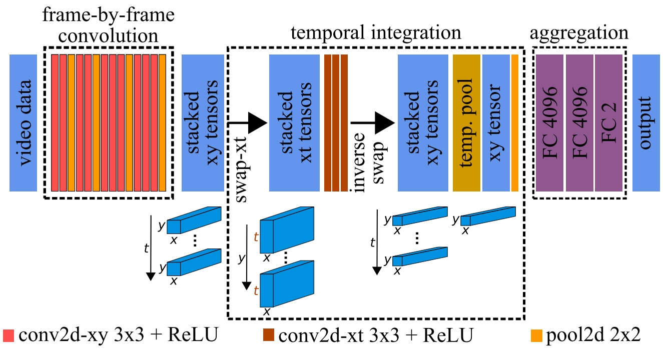


FIGURE 5. The *xt* branch of the S-CNN architecture for binary salient vs. non-salient video subvolume classification. Temporal integration is performed after the *swap-xt* operation via the convolutions operating in the *xt* plane and temporal pooling.

As input to our model, we used RGB video subvolumes $128\,\mathtt{px} \times 128\,\mathtt{px} \times 15\,\mathtt{frames}$ ($\mathtt{px}$ denoting pixels) around the pixel to be classified. Similar subvolumes were used in [62] for unsupervised feature learning. Unlike [38], we did not extract motion information explicitly, but relied on the network architecture entirely without any further input manipulations in order to achieve a simpler data processing pipeline.

To go from binary classification to generating a continuous (super)saliency map, we took the probability for the positive class at the soft-max layer of the network (for each respective surrounding subvolume of each video pixel). To reduce computation time, we only did this for every 10th pixel along both spatial axes. We then upscaled the resulting low-resolution map to the desired dimensions. For GazeCom and Hollywood2, we generated saliency maps at $640 \times 360\,\mathtt{px}$, whereas for CITIUS-R, the original resolution of 320 × 240 was used.

C. TRAINING DETAILS

Out of 823 training videos in Hollywood2, 90% (741 clips) were used for training and 10% for validation. Before extracting the subvolumes centred around positive or negative locations of our videos, these were rescaled to 640 × 360 pixels size and mirror-padded to reduce boundary effects. In total, the 823 clips contain 4,520,813 unique SP and 10,448,307 unique fixated locations. To assess the influence of the eye movement type in the training data, we fitted the same model twice for two different purposes. First, we trained the *S-CNN SP* model for predicting *supersaliency*, so the positive locations were those where SP had occurred. Analogously, for the *S-CNN FIX* model predicting purely fixation-based (i.e. excluding SP) *saliency*, the input video subvolumes where observers had fixated were labelled as positive.

For both *S-CNN SP* and *S-CNN FIX*, the training set consisted of 100,000 subvolumes, half of which were positives (as described above, randomly sampled from the respective eye movement locations in the training videos), half negatives (randomly selected in a uniform fashion to match the number of positive samples per video, excluding the subvolumes already in the positive set). For validation, 10,000 subvolumes were used, same sampling procedure as for the training set.

Convolutional layers were initialised with pre-trained VGG-16 weights, fully-connected layers were initialised randomly. We used a batch size of 5, and trained both models for 50,000 iterations with stochastic gradient descent (with momentum of 0.9, learning rate starting at 10^{-4} and decreasing 10-fold after every 20,000 iterations), at which point both loss and accuracy levelled out.

D. ADAPTIVE CENTRE BIAS

Since our model is inherently spatial bias-free, as it deals purely with individual subvolumes of the input video, we applied an adaptive solution to each frame – the gravity centre bias approach of Wu *et al.* [34], which emphasises not the centre of the frame, but the centre of mass in the saliency distribution. At this location, a single unit pixel is placed on the bias map, which is then blurred with a Gaussian filter (σ equivalent to three degrees of the visual field was chosen) and normalised to contain values ranging from 0 to the highest saliency value of the currently processed frame. Each frame of the video saliency map was then linearly mixed with its respective bias map (with a weight of 0.4 for the bias, and 0.6 for the original frame, as in [34]).

IV. VALIDATION WITH A MORE COMPLEX MODEL

As discussed in Section II-A, more sophisticated architectures have been developed over time to better handle both the spatial and the temporal aspects of deep video processing. While the slicing CNN model we used in Section III-B allowed us to avoid any additional steps when going from concept to implementation, end-to-end architectures provide a more modern and efficient tool for saliency prediction.

In order to investigate whether the benefits of supersaliency hold for an end-to-end model, we implemented an architecture combining two recent works: (i) the fully-convolutional deep DenseNet from [63] for efficient information extraction from each 2D frame, and (ii) the introduction of several convolutional LSTMs into an encoder-decoder network [64] for temporal integration. Thus, we replaced the encoder part of the network in [64] with a DenseNet structure as in [63], keeping the decoder simple. The dense blocks were modified to process the video frames in a time-distributed fashion (i.e. identical operations applied to all frames). The model is sketched in FIGURE 6. A detailed model description can be found in the supplementary material.

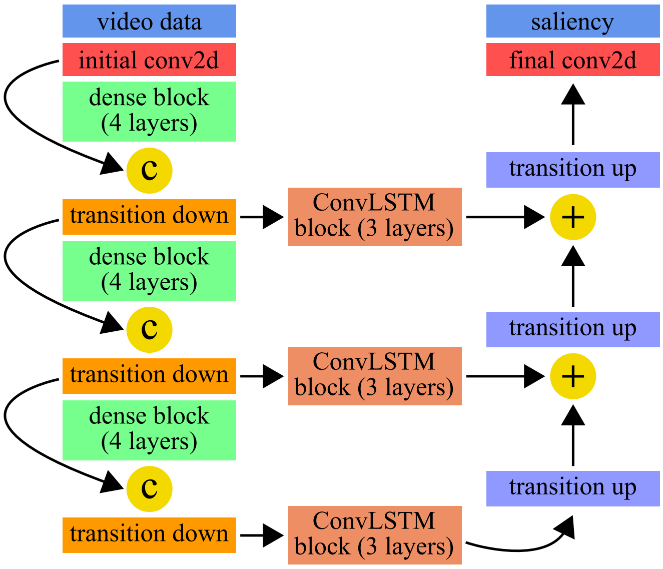


FIGURE 6. The outline of the end-to-end architecture we used for additional testing of our pipeline. In this scheme, “c” stands for the concatenation operation, “+” – for addition. In our experiments, ground truth saliency is provided as related to solely fixations, solely smooth pursuits, or both eye movements together.

A. TRAINING DETAILS

The Hollywood2 training set was randomly subdivided in the following way: 770 clips (ca. 200,000 frames) were used for training, 53 clips (ca. 15,000 frames) – for validation. The ground truth saliency map sequences were generated in the same way as for evaluation (see Section V-B). For this experiment, we trained the model to predict the saliency maps produced either for fixation or smooth pursuit samples only, or for the combination of both. The first two conditions correspond to purely fixation-based (traditional) saliency and purely pursuit-based supersaliency; the latter is very similar to only removing the saccades, and aggregating all the remaining gaze samples, as e.g. in [54].

We used Kullback-Leibler divergence as loss on the three-dimensional tensors of saliency (time × x × y), and trained the model for 10 epochs (500 iterations in each) with Adam optimiser [65] with default parameters (cf. Keras 2.2.4). The final model was selected based on the validation loss. Due to GPU memory constraints, we limited the input to this relatively large model to sequences of 12 frames (at $128 \times 72\,\mathrm{px}$) and used a batch size of 4. During training, the model produced sequences of 12 corresponding saliency frames for each input sequence. During testing, no video subdivision was performed.

Since this model operated on relatively low-resolution clips, we did not expect its saliency prediction to achieve benchmark-beating performance, but separately evaluated it and used its results to support our argument about the potential of the supersaliency problem setting, and the importance of the smooth pursuit eye movement for saliency in general.

V. EVALUATION

A. REFERENCE MODELS

We compared our approach to a score of publicly available dynamic saliency models. For compression domain models, we followed the pipeline and provided source code of Khatoonabadi *et al.* [25], generating the saliency maps for all videos at 288 pixels in height, and proportionally scaled width for PMES [66], MAM [67], PIM-ZEN [68], PIM-MCS [69], MCSDM [70], MSM-SM [71], PNSP-CS [72], and a range of OBDL-models [25], as well as pixel-domain GBVS [32], [73] and STSD [74]. Instead of the static AWS [75] that was used in [25], we evaluated AWS-D [24], its recent extension to dynamic stimuli (for GazeCom, after downscaling to $640 \times 360\,\mathtt{px}$ due to memory constraints, other data sets – at their original resolution). We also computed the three invariants (H, S, and K) of the structure tensor [76] at fixed temporal (second) and spatial (third) scales. For Hollywood2, the approach of Mathe and Sminchisescu [77], combining static (low-, mid-, and high-level) and motion features, was evaluated as well.

Deep models for saliency prediction on videos are much scarcer than such models for static images. As of yet, the problem of finding reference models in this domain is further confounded by the absence of publicly available code or data of some approaches, e.g. [39], and the popularity of salient object detection approaches and data sets, e.g. [78]–[80]. Included in our set of reference models are two recent approaches: DeepVS (OMCNN-2CLSTM) [41] – code available via [81] – and ACLNet [40] – code available via [82]. We ran both with default parameters on all three data sets.

B. BASELINES

The set of baselines was inspired by the works of Judd *et al.* [29], [30]: *Chance*, *Permutation*, *Centre*, *One Human*, and *Infinite Humans* (as a limit). The latter two cannot be computed unless gaze data *for each individual observer* are available (i.e. not possible for CITIUS). All the random baselines were repeated five times per video of each data set. The *ground truth saliency maps* were obtained via superimposing spatio-temporal Gaussians at every attended location of all the considered observers. The two spatial sigmas were set to one degree of visual angle (commonly used in the literature as the approximate fovea size, e.g. [29], [83]; [77] uses 1.5°). The temporal sigma was set to a frame count equivalent of $1/3$ of a second (so that the effect would be mostly contained within one second's distance).

C. METRICS

For a thorough evaluation, we took a broad spectrum of metrics (all computed the same way for fixation samples and onsets – saliency – and smooth pursuit samples – supersaliency – for the data sets described in Section III-A), mostly based on [83]: AUC-Judd, AUC-Borji, shuffled AUC (sAUC), normalised scanpath saliency (NSS), histogram similarity (SIM), correlation coefficient (CC), and Kullback-Leibler divergence (KLD), as well as Information Gain (IG) [84]. We additionally computed balanced accuracy (same positive and negative location sets as for AUC-Borji; accuracy at the equal error rate point).

In our implementation of sAUC and IG, in order to obtain salient locations of other clips, we first rescaled their temporal axes to fit the duration of the evaluated clip, and then sampled not just spatial (like e.g. [24]), but also temporal coordinates. This preserves the temporal structure of the stimulus-independent bias: E.g. the first fixations after stimulus display tend to have heavier centre bias than subsequent ones in both static images [85] and videos [86].

For GazeCom and Hollywood2, we fixed all saliency maps to $640 \times 360\,\mathtt{px}$ resolution during evaluation, either for memory constraints, or for symmetric evaluation in case of differently shaped videos. For CITIUS, the native resolution of $320 \times 240\,\mathtt{px}$ was maintained.

1) METRIC AVERAGING

Due to its selectivity (i.e. observers can decide not to pursue anything), SP is sparse and highly unbalanced between videos (see FIGURE 2). Simply averaging the performance scores across all videos of the data set could introduce artefacts for many metrics. For AUC-based metrics, for example, there exists a "perfect" aggregated score, which could be computed by combining the data over all the videos *before* computing the metric, i.e. merging all positives and all negatives beforehand. This is, however, not always possible, as many models use per-video or even per-frame normalisation as the final step, either to allow for easier visualisation, or to use the full spectrum of the 8-bit integer range, if the result is stored as a video. To demonstrate this averaging problem, we randomly sampled non-trivial subsets of video clips (100 times for all the possible subset sizes) of all three utilised test sets, and computed per-clip AUC-Borji and sAUC scores for our *S-CNN SP* model (without any normalisation of its outputs). We combined these via either regular or weighted (according to the number of SP- or fixation-salient locations samples, depending on the problem setting) averaging. This combination is then compared to the perfect score, as described above. We found that averaging per-video AUC scores is a significantly poorer approximation of the ideal score than their *weighted mean* ($p \ll 0.01$, for (super)saliency prediction on GazeCom and Hollywood2, see Table 1).

We will, therefore, present the weighted averaging results for supersaliency prediction. Since fixations suffer from this problem to a lesser extent, this adjustment is not essential there. However, in the data sets with great variation of fixation samples' share (e.g. Hollywood2: 30% to 78% in our 50-clip subset), we would generally recommend using weighting for fixation prediction evaluation as well. Conventional mean

TABLE 1. Means and standard deviations of the absolute error of "perfect AUC" estimation with *regular* and *weighted* averaging, as well as one-sided two-sample Kolmogorov-Smirnov test p-values (with the null hypothesis that regular averaging, as a way to estimate the perfect AUC score, produces absolute errors that are smaller than or equal to those of weighted averaging). Except for CITIUS-R, weighted averaging always demonstrates a statistically significant ($p \ll 0.01$) advantage over regular averaging.

Statistic	Absolute error properties	GazeCom			Hollywood2 (50 clips)			CITIUS-R
		SP	FIX	onsets	SP	FIX	onsets	onsets
AUC-Borji	mean (*regular* averaging)	0.038	0.011	0.012	0.022	0.017	0.018	0.0125
	mean (*weighted* averaging)	0.011	0.01	0.01	0.008	0.012	0.009	0.0135
	SD (*regular* averaging)	0.03	0.007	0.008	0.009	0.009	0.008	0.0079
	SD (*weighted* averaging)	0.01	0.007	0.007	0.004	0.004	0.004	0.0075
	p-value	9e-205	4e-16	8e-16	0e+00	0e+00	0e+00	0.92
sAUC	mean (*regular* averaging)	0.039	0.013	0.014	0.038	0.029	0.031	0.0173
	mean (*weighted* averaging)	0.015	0.011	0.011	0.011	0.015	0.012	0.0169
	SD (*regular* averaging)	0.031	0.008	0.009	0.016	0.014	0.013	0.0092
	SD (*weighted* averaging)	0.014	0.008	0.008	0.006	0.005	0.005	0.0089
	p-value	1e-137	4e-20	2e-25	0e+00	0e+00	0e+00	0.02

results for fixations are, nevertheless, presented for comparability with the literature (weighted results reveal a similar picture).

2) CROSS-AUC

Another point we raise in our evaluation is directly distinguishing SP-salient from fixation-salient pixels based on the saliency maps. To this end, we introduced *cross-AUC (xAUC)*: The AUC is computed for the positive samples' set of all pursuit-salient locations, with an equal number of randomly selected fixation-salient locations for the same stimulus used as negatives. The baselines' performance on this metric will be indicative of how well the targets for these two eye movements can be separated (in comparison to the separation of salient and non-salient locations). If a model scores above 50% on this metric, it on average favours (i.e. assigns higher saliency scores to) pursuit-salient locations over fixation-salient ones (since SP is chosen as the positive class). For the purpose of distinguishing the two eye movement types, the scores of 70% and 30% are, however, equivalent: Such scores would reveal that a model favours either SPs over fixations, or vice versa, respectively, with the same bias from not displaying any preference whatsoever (and the corresponding xAUC of 50%).

VI. RESULTS AND DISCUSSION

A. SLICING CNN RESULTS

We tested the outputs of 26 published dynamic saliency models, including two deep learning-based solutions, as well as our own S-CNN models – SP and fixation predictors both with and without the additional post-processing step of gravity centre bias. For brevity and because there is no principled way of averaging different metrics numerically, we present the results as average ranks (over the 9 metrics we used – see Section V-C) in Table 2. Complete tables of all metric scores for all 7 data types (corresponding to the columns of Table 2) and 35 baselines and models can be found in the supplementary material.

Traditional saliency prediction commonly evaluates only one sample per fixation, as we did in the "onset" condition. For supersaliency, however, all gaze samples need to be predicted individually, and for consistency we did the same for fixations in the "FIX" condition. In principle, this should give greater weight to longer fixations with more samples, but our results show that differences between evaluating in the "FIX" and "onset" conditions are small in practice (cf. respective columns in Table 2).

On average, our pursuit prediction model, combined with adaptive centre bias (*S-CNN SP* + *Gravity CB*), performs best, almost always making it to the first or the second position (and always in the top-4). Remarkably, this holds true both for the prediction of smooth pursuits and the prediction of fixations, despite training exclusively on SP-salient locations as positive examples. The success of our pursuit prediction approach in predicting fixations can be potentially attributed to humans pursuing and fixating similar targets, but the relative selectivity of SP allows the model to focus on the particularly interesting objects in the scene. Even without the gravity centre bias, both our saliency *S-CNN FIX* and supersaliency *S-CNN SP* models outperform the models from the literature on the whole, with their average rank at least two positions better than that of the next best model (ACLNet).

The fact that all our *S-CNN* models consistently outperform the traditional "shallow" reference models for both saliency and supersaliency prediction on all data sets demonstrates the potential of deep video saliency models. This is in line with the findings in e.g. [39], [87], where a deep architecture has shown superior fixation prediction performance, compared to non-CNN models. On Hollywood2, due to the very centre biased nature of the gaze locations [21], for example, only the deep learning models (*S-CNN*, ACLNet, and DeepVS) rank higher than the Centre Baseline or achieve non-negative information gain scores (cf. Table 2 and the tables in the supplementary material).

Only in the fixation prediction task on the Hollywood2 data set, the results of our best model are inferior to the two deep reference approaches (and only to those) – DeepVS and ACLNet. On both other data sets (GazeCom and CITIUS-R), as well as for supersaliency prediction on Hollywood2, our model is outperforming all reference algorithms. The two evaluated deep literature approaches are particularly weak on the GazeCom data set, and especially in the task of predicting

TABLE 2. Evaluation results, presented as the mean of rank values for all the metrics we compute (except for xAUC). "Onset" refers to evaluation against fixation onsets ("traditional" saliency). Where marked with *, ranking was computed for the weighted average of the scores. The rows with gray background correspond to baselines. Top-3 non-baseline results in each category are boldified.

Model	GazeCom			Hollywood2 (50 clips)			CITIUS-R	average rank
	SP*	FIX	onset	SP*	FIX	onset	onset	
Infinite Humans	1.0	1.0	1.0	1.0	1.0	1.0	–	1.0
S-CNN SP + Gravity CB	**4.9**	**2.9**	**2.9**	**4.0**	5.1	5.0	**3.3**	**4.0**
S-CNN FIX + Gravity CB	12.2	**2.8**	**2.8**	5.3	**4.6**	**4.1**	**3.9**	**5.1**
S-CNN SP	**3.0**	**4.4**	**4.1**	6.2	7.6	7.4	4.8	**5.4**
S-CNN FIX	**9.1**	4.6	4.8	7.7	6.7	6.8	5.6	6.4
ACLNet	24.3	11.0	10.7	**4.3**	**2.9**	**3.4**	**3.3**	8.6
DeepVS (OMCNN-2CLSTM)	25.4	9.8	11.0	**5.0**	**4.7**	**4.7**	8.2	9.8
GBVS	11.1	11.2	10.1	11.6	11.3	11.1	7.6	10.6
OBDL-MRF-O	13.8	12.2	11.9	13.7	13.3	11.8	9.9	12.4
OBDL-MRF-OC	15.1	13.9	13.7	14.8	14.2	12.9	11.2	13.7
AWS-D	14.9	7.9	7.4	24.0	18.0	18.2	7.2	14.0
OBDL-MRF-TO	18.6	13.9	14.8	12.6	14.3	15.2	13.3	14.7
OBDL-MRF	18.9	16.0	16.3	13.8	11.3	12.8	13.7	14.7
Centre	29.4	16.8	16.4	9.6	10.3	10.1	10.6	14.7
OBDL-MRF-T	23.1	13.2	14.9	12.6	12.2	15.1	15.3	15.2
One Human	18.7	19.2	22.6	11.1	9.8	10.6	–	15.3
OBDL-T	13.7	15.1	12.4	17.9	18.7	18.6	11.1	15.3
OBDL-MRF-C	16.3	16.1	16.0	15.9	15.9	14.2	13.1	15.4
OBDL-MRF-TC	20.6	12.9	14.2	12.8	16.8	17.1	15.6	15.7
OBDL-S	14.7	19.8	18.4	19.1	20.2	19.9	17.3	18.5
Mathe	–	–	–	20.7	21.7	21.9	–	21.4
Invariant-K	11.3	20.8	19.6	30.2	25.0	25.0	20.8	21.8
STSD	18.0	21.6	21.6	27.4	25.4	25.1	18.0	22.4
OBDL	22.4	23.2	22.4	22.9	22.9	22.8	20.8	22.5
PMES	11.8	27.8	27.0	22.0	27.0	27.4	27.0	24.3
PIM-ZEN	13.6	26.4	26.3	24.0	26.6	27.1	26.8	24.4
PIM-MCS	14.3	25.9	26.1	25.8	26.7	27.2	26.2	24.6
Invariant-S	28.6	21.9	22.2	32.0	27.7	27.2	22.8	26.0
MSM-SM	16.8	33.1	32.3	21.7	28.4	27.2	25.4	26.4
PNSP-CS	13.9	28.7	28.7	28.1	30.1	30.1	27.2	26.7
Permutation	33.4	29.3	29.4	27.7	23.4	22.3	24.8	27.2
MCSDM	13.4	28.4	28.3	30.7	31.0	31.6	28.1	27.4
Invariant-H	29.8	23.7	24.4	33.3	30.9	30.9	25.8	28.4
Chance	31.0	28.0	28.0	32.4	31.8	31.9	28.2	30.2
MAM	27.9	31.6	32.1	28.3	32.6	32.2	31.1	30.8

pursuit-based supersaliency. Qualitatively, we observed that their predicted saliency distributions tend to miss moving salient targets, unless these are close to the centre of the frame.

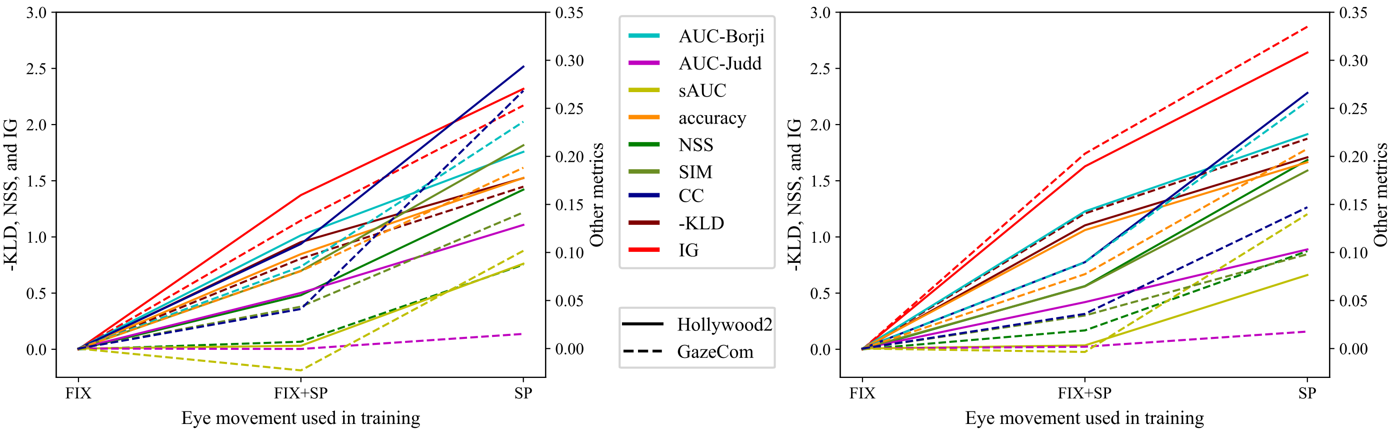
Both with and without the gravity centre bias, our supersaliency *S-CNN SP* models perform better than our respective saliency *S-CNN FIX* models (with the difference in average rank values of ca. one position). We emphasise that these models were only trained on the Hollywood2 training set. On the Hollywood2 test set, maybe not surprisingly, the fixation-predicting models perform better for fixation-based saliency and SP-predicting models perform better for pursuit-based supersaliency. On the two other data sets, however, the models that were trained for SP prediction generally perform better than their fixation-trained counterparts, indicating their greater generalisation capability.

To find informative video regions, we use humans as a yardstick, since they clearly excel at real-world tasks despite their limited perceptual throughput. Smooth pursuit is more selective than fixations and thus likely restricted to particularly interesting objects. The use of such sparser (yet more densely concentrated [27]), higher-quality training data could explain the superior generalisability of the supersaliency models to independent data sets.

For visual comparison, example saliency map sequences are presented in FIGURE 4a and FIGURE 4b for select GazeCom and Hollywood2 clips, respectively. It can be seen, for example, that our *S-CNN FIX* model differentiates well between fixation-rich and SP-rich frames in an example Hollywood2 clip.

B. END-TO-END VALIDATION

To additionally highlight the importance of pursuit and supersaliency in the context of a more state-of-the-art-like architecture, we trained a model encompassing both DenseNet and convolutional LSTM elements (see Section IV) in several set-ups: While keeping the training pipeline the same, we differently generated the ground truth saliency maps. We examined three conditions: (i) fixation-only attention, (ii) fixation- and pursuit-based attention, and (iii) pursuit-only attention. Taking the performance in the first condition as a baseline, we plot the absolute improvements of the saliency metrics in other conditions in



(a) Fixation prediction scores improvement

(b) Pursuit prediction scores improvement

FIGURE 7. Absolute improvement in the scores of our end-to-end saliency prediction model (see Section IV) due to the type of training data used (see *x* axis). FIGURE 7a reports the improvements of fixation-based saliency prediction, while FIGURE 7b depicts the same improvements for pursuit-based supersaliency prediction. Including pursuit-based attention into training (FIX+SP condition) is beneficial for the vast majority of metrics, compared to training for predicting purely fixation-based attention (FIX condition). Notably, training the model for the supersaliency problem directly (SP condition) always benefited our model, when tested for both the traditional saliency (7a) and supersaliency (7b) tasks.

FIGURE 7 (full absolute performance scores can be found in the supplementary material).

In these plots, the models trained in one of the three conditions are tested on the task of either fixation prediction (FIGURE 7a) or smooth pursuit prediction (FIGURE 7b). For both of the tasks, on GazeCom and Hollywood2 data sets alike, the values of performance measures are almost always improved when pursuit samples are added to fixation-only attention modelling (transition from "FIX" to "FIX+SP" conditions in the figures). Most importantly, performance of the model is invariably and noticeably improved when only pursuit samples are used for training. Only AUC-Judd on the GazeCom data set is just slightly improving between these conditions, because the metric is not class-balanced and is saturating at high saliency map resolutions. Evaluation at a lower resolution of saliency maps yields much more noticeable performance improvements for AUC-Judd as well (data not shown).

The results on CITIUS-R are qualitatively and quantitatively similar, and are not depicted for better figure readability. This again points to the greater across-data set generalisation capability of a model that was trained to predict supersaliency maps, compared to an identically trained model for saliency map prediction.

C. DISTINGUISHING FIXATION AND PURSUIT TARGETS

In the task of separating SP- and fixation-salient locations (the xAUC metric), most models yield a result above 0.5 on GazeCom, which means that they still, by chance or by design, assign higher saliency values to SP locations (unlike e.g. the centre baseline with xAUC score of 0.44, which implies that fixations on this data set are more centre biased than pursuits). Probably due to their emphasis on motion information, the top of the chart with respect to this metric is heavily dominated by compression-domain approaches (top-7 non-baseline models for GazeCom, top-4 for Hollywood2, cf. tables in the supplementary material). Even though in the limit (*Infinite Humans* baseline) this metric's weighted average can be confidently above 0.9, the best model's (MSM-SM [71]) result is just below 0.74 for GazeCom, and below 0.6 for Hollywood2. This particular aspect needs more investigation and, possibly, dedicated training: Notably, the models proposed in this work were not trained to maximise xAUC, but rather to achieve better general-purpose saliency prediction, conditioned on one eye movement type or the other.

D. GENERAL IMPLICATIONS

The work presented here points out a major methodological concern: While smooth pursuit comprises a significant part of the viewing behaviour, is has never been systematically analysed in the context of saliency prediction. This lack of specialised analysis means that the gaze samples corresponding to the form of attention expressed as smooth pursuit will be either discarded in the analysis, or labelled inconsistently.

Analysing attention means analysing both fixations *and* smooth pursuits, with a caveat: Fixations are not always intentional and can correspond to inattentive viewing or mind wandering [88], [89]. Typical works on saliency prediction only talk about fixations, never accounting for what can be called their attentiveness. Our work, on the contrary, demonstrates that using only smooth pursuit gaze samples – i.e. those when the eye movements reveal attentive viewing by following a moving target – can help improve on traditional saliency approaches.

This, however, is not the end of the story: We only consider pure eye movement information to uncover something about the observer's attention. Instead, e.g. pupil size can be used to infer attention (see e.g. [90] for a review of the works on connecting pupil size dynamics to a variety of perception aspects; [91], [92]), though the analysis might be more complex. If an

EEG signal is recorded simultaneously with eye tracking data, this can be analysed to infer periods of attentive viewing as well [93], [94]. Recent technological advancements have enabled simultaneous fMRI and eye tracking recording [10], [95], which could open the next frontier for analysing attention allocation with the help of brain imaging.

Directly tying together pursuit, saliency, and brain activity, albeit with synthetic stimuli and single-neuron recordings, [96] examined neuron spiking in monkeys, comparing the extent to which different regions in the brain encode visual saliency (in a low-level sense). Generalising such conclusions to more naturalistic [97] and realistic visual stimuli would require a better method to analyse naturally occurring smooth pursuit, and could further our understanding of what exactly contemporary saliency models learn.

VII. CONCLUSION

In this paper, we introduced the concept of *supersaliency* – smooth pursuit-based attention prediction. We argue that pursuit exhibits properties that set it apart from fixations in terms of perception and behavioural consequences, and that predicting smooth pursuit should thus be studied separately from fixation prediction. To this end, we provide our pipeline and the ground truth for saliency and supersaliency problems for the large-scale Hollywood2, as well as for the manually annotated GazeCom at `https://gin.g-node.org/MikhailStartsev/supersaliency`.

To better understand a model's behaviour on supersaliency data, we introduced the cross-AUC metric that assesses an algorithm's preference for pursuit vs. fixation locations, thus describing its ability to distinguish between the two. Whereas the human data showed that there are clear systematic differences between the two target types, it remains an open question how to reliably capture these differences with video-based saliency models.

Finally, we proposed and evaluated a deep saliency model with the slicing CNN architecture, which we trained for both smooth pursuit and fixation-based attention prediction. In both settings, our model outperformed all 26 tested dynamic reference models. Importantly, training for supersaliency yielded better results even for traditional fixation-based saliency prediction on two additional independent data sets. The same trend was observed with an additionally introduced deep end-to-end saliency model, further validating our conclusions that supersaliency demonstrates better generalisability. These findings demonstrate the potential of smooth pursuit modelling and prediction.

REFERENCES

[1] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013.

[2] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 185–198, Jan. 2010.

[3] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 19–33, Jan. 2014.

[4] S. Marat, M. Guironnet, and D. Pellerin, "Video summarization using a visual attention model," in *Proc. 15th Eur. Signal Process. Conf.*, Sep. 2007, pp. 1784–1788.

[5] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 300–312, Feb. 2007.

[6] Y. Nagai and K. J. Rohlfing, "Computational analysis of motionese toward scaffolding robot action learning," *IEEE Trans. Auton. Mental Develop.*, vol. 1, no. 1, pp. 44–54, May 2009.

[7] A. C. Schütz, D. I. Braun, and K. R. Gegenfurtner, "Eye movements and perception: A selective review," *J. Vis.*, vol. 11, no. 5, p. 9, 2011, doi: 10.1167/11.5.9.

[8] M. Spering, A. C. Schütz, D. I. Braun, and K. R. Gegenfurtner, "Keep your eyes on the ball: Smooth pursuit eye movements enhance prediction of visual motion," *J. Neurophysiol.*, vol. 105, no. 4, pp. 1756–1767, 2011. [Online]. Available: http://jn.physiology.org/content/105/4/1756

[9] I. Agtzidis, M. Startsev, and M. Dorr, "Smooth pursuit detection based on multiple observers," in *Proc. 9th Biennial ACM Symp. Eye Tracking Res. Appl. (ETRA)*, New York, NY, USA, 2016, pp. 303–306, doi: 10.1145/2857491.2857521.

[10] M. Hanke, N. Adelhöfer, D. Kottke, V. Iacovella, A. Sengupta, F. R. Kaule, R. Nigbur, A. Q. Waite, F. Baumgartner, and J. Stadler, "A studyforrest extension, simultaneous fMRI and eye gaze recordings during prolonged natural stimulation," *Sci. Data*, vol. 3, Oct. 2016, Art. no. 160092.

[11] M. F. Land, "Eye movements and the control of actions in everyday life," *Prog. Retinal Eye Res.*, vol. 25, no. 3, pp. 296–324, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1350946206000036

[12] C. N. Authié and D. R. Mestre, "Optokinetic nystagmus is elicited by curvilinear optic flow during high speed curve driving," *Vis. Res.*, vol. 51, no. 16, pp. 1791–1800, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0042698911002173

[13] O. Lappi and E. Lehtonen, "Eye-movements in real curve driving: Pursuit-like optokinesis in vehicle frame of reference, stability in an allocentric reference coordinate system," *J. Eye Movement Res.*, vol. 6, no. 1, pp. 1–13, 2013. [Online]. Available: https://bop.unibe.ch/JEMR/article/view/2352

[14] O. Lappi, P. Rinkkala, and J. Pekkanen, "Systematic observation of an expert driver's gaze strategy—An on-road case study," *Frontiers Psychol.*, vol. 8, p. 620, Apr. 2017. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2017.00620

[15] J. B. Pelz and R. Canosa, "Oculomotor behavior and perceptual strategies in complex tasks," *Vis. Res.*, vol. 41, no. 25, pp. 3587–3596, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0042698901002450

[16] A. B. Sereno and P. S. Holzman, "Antisaccades and smooth pursuit eye movements in schizophrenia," *Biol. Psychiatry*, vol. 37, no. 6, pp. 394–401, 1995.

[17] J. E. Silberg, I. Agtzidis, M. Startsev, T. Fasshauer, K. Silling, A. Sprenger, M. Dorr, and R. Lencer, "Free visual exploration of natural movies in schizophrenia," *Eur. Arch. Psychiatry Clin. Neurosci.*, vol. 269, pp. 407–418, Jan. 2018, doi: 10.1007/s00406-017-0863-1.

[18] M. Vidal, A. Bulling, and H. Gellersen, "Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, New York, NY, USA, 2013, pp. 439–448. [Online]. Available: http://doi.acm.org/10.1145/2493432.2493477

[19] A. Esteves, E. Velloso, A. Bulling, and H. Gellersen, "Orbits: Gaze interaction for smart watches using smooth pursuit eye movements," in *Proc. 28th Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, New York, NY, USA, 2015, pp. 457–466. [Online]. Available: http://doi.acm.org/10.1145/2807442.2807499

[20] S. Schenk, P. Tiefenbacher, G. Rigoll, and M. Dorr, "SPOCK: A smooth pursuit oculomotor control kit," in *Proc. CHI Conf. Extended Abstr. Hum. Factors Comput. Syst. (CHI EA)*, New York, NY, USA, 2016, pp. 2681–2687. [Online]. Available: http://doi.acm.org/10.1145/2851581.2892291

[21] M. Dorr, T. Martinetz, K. R. Gegenfurtner, and E. Barth, "Variability of eye movements when viewing dynamic natural scenes," *J. Vis.*, vol. 10, no. 10, p. 28, 2010, doi: 10.1167/10.10.28.

[22] S. Mathe and C. Sminchisescu, "Dynamic eye movement datasets and learnt saliency models for visual action recognition," in *Proc. 12th Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer-Verlag, 2012, pp. 842–856, doi: 10.1007/978-3-642-33709-3_60.

[23] L. J. Siever, R. D. Coursey, I. S. Alterman, M. S. Buchsbaum, and D. L. Murphy, "Impaired smooth pursuit eye movement: Vulnerability marker for schizotypal personality disorder in a normal volunteer population," *Amer. J. Psychiatry*, vol. 141, no. 12, pp. 1560–1566, 1984, doi: 10.1176/ajp.141.12.1560.

[24] V. Leborán, A. García-Díaz, X. R. Fdez-Vidal, and X. M. Pardo, "Dynamic whitening saliency," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 893–907, May 2017.

[25] S. H. Khatoonabadi, N. Vasconcelos, I. V. Bajić, and Y. Shan, "How many bits does it take for a stimulus to be salient?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5501–5510.

[26] M. Startsev and M. Dorr, "Increasing video saliency model generalizability by training for smooth pursuit prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2018, pp. 2050–20503.

[27] M. Startsev, I. Agtzidis, and M. Dorr, "Characterizing and automatically detecting smooth pursuit in a large-scale ground-truth data set of dynamic natural scenes," *J. Vis.*, vol. 19, no. 14, p. 10, Dec. 2019, doi: 10.1167/19.14.10.

[28] J. Shao, C.-C. Loy, K. Kang, and X. Wang, "Slicing convolutional neural network for crowd video understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5620–5628.

[29] T. Judd, F. Durand, and A. Torralba. (2012). *A Benchmark of Computational Models of Saliency to Predict Human Fixations*. [Online]. Available: http://hdl.handle.net/1721.1/68590

[30] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. *MIT Saliency Benchmark*. Accessed: Nov. 23, 2018. [Online]. Available: http://saliency.mit.edu/

[31] D.-P. Fan, W. Wang, M.-M. Cheng, and J. Shen, "Shifting more attention to video salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8554–8564.

[32] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 545–552.

[33] J. Wang, H. R. Tavakoli, and J. Laaksonen, "Fixation prediction in videos using unsupervised hierarchical features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2225–2232.

[34] Z. Wu, L. Su, Q. Huang, B. Wu, J. Li, and G. Li, "Video saliency prediction with optimized optical flow and gravity center bias," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2016, pp. 1–6.

[35] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *Proc. 14th ACM Int. Conf. Multimedia (MM)*, New York, NY, USA, 2006, pp. 815–824. [Online]. Available: http://doi.acm.org/10.1145/1180639.1180824

[36] Y. Li and Y. Li, "A fast and efficient saliency detection model in video compressed-domain for human fixations prediction," *Multimedia Tools Appl.*, vol. 76, pp. 1–23, Dec. 2016, doi: 10.1007/s11042-016-4118-3.

[37] M. Xu, L. Jiang, X. Sun, Z. Ye, and Z. Wang, "Learning to detect video saliency with hevc features," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 369–385, Jan. 2017.

[38] S. Chaabouni, J. Benois-Pineau, O. Hadar, and C. B. Amar, "Deep learning for saliency prediction in natural video," *CoRR*, vol. abs/1604.08010, pp. 1–34, Apr. 2016. [Online]. Available: https://dblp.uni-trier.de/rec/bibtex/journals/corr/ChaabouniBHA16

[39] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[40] W. Wang, J. Shen, F. Guo, M.-M. Cheng, and A. Borji, "Revisiting video saliency: A large-scale benchmark and a new model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4894–4903.

[41] L. Jiang, M. Xu, T. Liu, M. Qiao, and Z. Wang, "DeepVS: A deep learning based video saliency prediction approach," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018.

[42] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper ConvLSTM for video salient object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 715–731.

[43] W. Byeon, Q. Wang, R. K. Srivastava, and P. Koumoutsakos, "ContextVP: Fully context-aware video prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[45] C. F. Cadieu, H. Hong, D. L. K. Yamins, N. Pinto, D. Ardila, E. A. Solomon, N. J. Majaj, and J. J. DiCarlo, "Deep neural networks rival the representation of primate IT cortex for core visual object recognition," *PLoS Comput. Biol.*, vol. 10, no. 12, pp. 1–18, Dec. 2014, doi: 10.1371/journal.pcbi.1003963.

[46] S. Winkler and R. Subramanian, "Overview of eye tracking datasets," in *Proc. 5th Int. Workshop Qual. Multimedia Exper. (QoMEX)*, Jul. 2013, pp. 212–217.

[47] N. Riche, M. Mancas, D. Culibrk, V. Crnojevic, B. Gosselin, and T. Dutoit, "Dynamic saliency models and human attention: A comparative study on videos," in *Computer Vision*. Berlin, Germany: Springer, 2013, pp. 586–598, doi: 10.1007/978-3-642-37431-9_45.

[48] H. Hadizadeh, M. J. Enriquez, and I. V. Bajić, "Eye-tracking database for a set of standard video sequences," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 898–903, Feb. 2012.

[49] E. Vig, M. Dorr, and D. Cox, "Space-variant descriptor sampling for action recognition based on saliency and eye movements," in *Computer Vision*. Berlin, Germany: Springer, 2012, pp. 84–97, doi: 10.1007/978-3-642-33786-4_7.

[50] H. Alers, J. A. Redi, and I. Heynderickx, "Examining the effect of task on viewing behavior in videos using saliency maps," *Proc. SPIE, Hum. Vis. Electron. Imag.*, vol. 8291, Feb. 2012, Art. no. 82910X.

[51] R. Carmi and L. Itti, "The role of memory in guiding attention during natural vision," *J. Vis.*, vol. 6, no. 9, p. 4, 2006, doi: 10.1167/6.9.4.

[52] F. Boulos, W. Chen, B. Parrein, and P. L. Callet, "Region-of-interest intra prediction for H.264/AVC error resilience," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 3109–3112.

[53] U. Engelke, R. Pepion, P. L. Callet, and H.-J. Zepernick, "Linking distortion perception and visual saliency in H.264/AVC coded video containing packet loss," *Proc. SPIE, Vis. Commun. Image Process.*, vol. 7744, Jul. 2010, Art. no. 774406.

[54] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson, "Clustering of gaze during dynamic scene viewing is predicted by motion," *Cogn. Comput.*, vol. 3, no. 1, pp. 5–24, 2011, doi: 10.1007/s12559-010-9074-z.

[55] I. Agtzidis, M. Startsev, and M. Dorr, "In the pursuit of (ground) truth: A hand-labelling tool for eye movements recorded during dynamic scene viewing," in *Proc. IEEE 2nd Workshop Eye Tracking Vis. (ETVIS)*, Oct. 2016, pp. 65–68.

[56] M. Startsev, I. Agtzidis, and M. Dorr, "1D CNN with BLSTM for automated classification of fixations, saccades, and smooth pursuits," *Behav. Res. Methods*, vol. 51, no. 2, pp. 556–572, Apr. 2019, doi: 10.3758/s13428-018-1144-2.

[57] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2106–2113.

[58] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[59] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–14. [Online]. Available: http://arxiv.org/abs/1409.1556

[60] E. Vig, M. Dorr, and E. Barth, "Contribution of spatio-temporal intensity variation to bottom-up saliency," in *Bio-Inspired Models of Network, Information, and Computing Systems*. Berlin, Germany: Springer, 2012, pp. 469–474, doi: 10.1007/978-3-642-32615-8_44.

[61] K. G. Rottach, A. Z. Zivotofsky, V. E. Das, L. Averbuch-Heller, A. O. Discenna, A. Poonyathalang, and R. Leigh, "Comparison of horizontal, vertical and diagonal smooth pursuit eye movements in normal human subjects," *Vis. Res.*, vol. 36, no. 14, pp. 2189–2195, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0042698995003029

[62] G. Chen, D. Clarke, M. Giuliani, A. Gaschler, D. Wu, D. Weikersdorfer, and A. Knoll, "Multi-modality gesture detection and recognition with un-supervision, randomization and discrimination," in *Computer Vision*, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Cham, Switzerland: Springer, 2015, pp. 608–622, doi: 10.1007/978-3-319-16178-5_43.

[63] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jul. 2017, pp. 1175–1183.

[64] S. S. Nabavi, M. Rochan, and Y. Wang, "Future semantic segmentation with convolutional LSTM," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*. Newcastle, U.K.: Northumbria Univ., Sep. 2018, p. 137. [Online]. Available: http://bmvc2018.org/contents/papers/0559.pdf

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, Dec. 2015, pp. 1–15. [Online]. Available: http://arxiv.org/abs/1412.6980

[66] Y.-F. Ma and H.-J. Zhang, "A new perceived motion based shot content representation," in *Proc. Int. Conf. Image Process.*, vol. 3, 2001, pp. 426–429.

[67] Y.-F. Ma and H.-J. Zhang, "A model of motion attention for video skimming," in *Proc. Int. Conf. Image Process.*, vol. 1, 2002, pp. I-129–I-132.

[68] G. Agarwal, A. Anbu, and A. Sinha, "A fast algorithm to find the region-of-interest in the compressed MPEG domain," in *Proc. Int. Conf. Multimedia Expo (ICME)*, vol. 2, Jul. 2003, pp. II-133–II-136.

[69] A. Sinha, G. Agarwal, and A. Anbu, "Region-of-interest based compressed domain video transcoding scheme," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, May 2004, pp. iii-161-iii-164.

[70] Z. Liu, H. Yan, L. Shen, Y. Wang, and Z. Zhang, "A motion attention model based rate control algorithm for H. 264/AVC," in *Proc. 8th IEEE/ACIS Int. Conf. Comput. Inf. Sci.*, Jun. 2009, pp. 568–573.

[71] K. Muthuswamy and D. Rajan, "Salient motion detection in compressed domain," *IEEE Signal Process. Lett.*, vol. 20, no. 10, pp. 996–999, Oct. 2013.

[72] Y. Fang, W. Lin, Z. Chen, C.-M. Tsai, and C.-W. Lin, "A video saliency detection model in compressed domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 27–38, Jan. 2014.

[73] J. Harel. *A Saliency Implementation in MATLAB*. Accessed: Nov. 23, 2018. [Online]. Available: http://www.klab.caltech.edu/~harel/share/gbvs.php

[74] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *J. Vis.*, vol. 9, no. 12, p. 15, 2009, doi: 10.1167/9.12.15.

[75] A. García-Díaz, X. R. Fdez-Vidal, X. M. Pardo, and R. Dosil, "Saliency from hierarchical adaptation through decorrelation and variance normalization," *Image Vis. Comput.*, vol. 30, no. 1, pp. 51–64, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885611001235

[76] E. Vig, M. Dorr, T. Martinetz, and E. Barth, "Intrinsic dimensionality predicts the saliency of natural dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1080–1091, Jun. 2012.

[77] S. Mathe and C. Sminchisescu, "Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1408–1424, Jul. 2015.

[78] Y. Tang, W. Zou, Z. Jin, and X. Li, "Multi-scale spatiotemporal Conv-LSTM network for video saliency detection," in *Proc. ACM Int. Conf. Multimedia Retr.*, 2018, pp. 362–369.

[79] G. Ding and Y. Fang, "Video saliency detection by 3D convolutional neural networks," in *Digital TV and Wireless Multimedia Communication*, G. Zhai, J. Zhou, and X. Yang, Eds. Singapore: Springer, 2018, pp. 245–254.

[80] W. Wang, J. Shen, and L. Shao, "Video salient object detection via fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 38–49, Jan. 2018.

[81] L. Jiang, M. Xu, and Z. Wang. (2017). *OMCNN_2CLSTM: The Model of DeepVS: A Deep Learning Based Video Saliency Prediction Approach (ECCV2018)*. [Online]. Available: https://github.com/remega/OMCNN_2CLSTM

[82] W. Wang, J. Shen, F. Guo, M.-M. Cheng, and A. Borji. (2017). *DHF1K: Revisiting Video Saliency: A Large-Scale Benchmark and a New Model (CVPR)*. [Online]. Available: https://github.com/wenguanwang/DHF1K

[83] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 740–757, Mar. 2019.

[84] M. Kümmerer, T. S. A. Wallis, and M. Bethge, "Information-theoretic model comparison unifies saliency metrics," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 52, pp. 16054–16059, 2015. [Online]. Available: http://www.pnas.org/content/112/52/16054.abstract

[85] B. W. Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *J. Vis.*, vol. 7, no. 14, p. 4, 2007, doi: 10.1167/7.14.4.

[86] P.-H. Tseng, R. Carmi, I. G. M. Cameron, D. P. Munoz, and L. Itti, "Quantifying center bias of observers in free viewing of dynamic natural scenes," *J. Vis.*, vol. 9, no. 7, p. 4, 2009, doi: 10.1167/9.7.4.

[87] C. Bak, A. Kocak, E. Erdem, and A. Erdem, "Spatio-temporal saliency networks for dynamic saliency prediction," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1688–1698, Jul. 2018.

[88] J. Smallwood, E. Beach, J. W. Schooler, and T. C. Handy, "Going AWOL in the brain: Mind wandering reduces cortical analysis of external events," *J. Cogn. Neurosci.*, vol. 20, no. 3, pp. 458–469, 2008, doi: 10.1162/jocn.2008.20037.

[89] J. Smallwood, "Mind-wandering while reading: Attentional decoupling, mindless reading and the cascade model of inattention," *Lang. Linguistics Compass*, vol. 5, no. 2, pp. 63–77, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1749-818X.2010.00263.x

[90] S. Mathôt and S. V. der Stigchel, "New light on the mind's eye: The pupillary light response as active vision," *Current Directions Psychol. Sci.*, vol. 24, no. 5, pp. 374–378, 2015, doi: 10.1177/0963721415593725.

[91] O. E. Kang, K. E. Huffer, and T. P. Wheatley, "Pupil dilation dynamics track attention to high-level information," *PLoS ONE*, vol. 9, no. 8, pp. 1–6, Aug. 2014, doi: 10.1371/journal.pone.0102463.

[92] S. M. Wierda, H. van Rijn, N. A. Taatgen, and S. Martens, "Pupil dilation deconvolution reveals the dynamics of attention at high temporal resolution," *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 22, pp. 8456–8460, 2012. [Online]. Available: https://www.pnas.org/content/109/22/8456

[93] V. Balasubramanian, K. Adalarasu, and A. Gupta, "EEG based analysis of cognitive fatigue during simulated driving," *Int. J. Ind. Syst. Eng.*, vol. 7, no. 2, pp. 135–149, 2011. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJISE.2011.038563

[94] N.-H. Liu, C.-Y. Chiang, and H.-C. Chu, "Recognizing the degree of human attention using EEG signals from mobile sensors," *Sensors*, vol. 13, no. 8, pp. 10273–10286, Aug. 2013, doi: 10.3390/s130810273.

[95] M. Kanowski, J. W. Rieger, T. Noesselt, C. Tempelmann, and H. Hinrichs, "Endoscopic eye tracking system for fMRI," *J. Neurosci. Methods*, vol. 160, no. 1, pp. 10–15, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165027006003931

[96] B. J. White, L. Itti, and D. P. Munoz, "Superior colliculus encodes visual saliency during smooth pursuit eye movements," *Eur. J. Neurosci.*, vol. 8, no. 14263, pp. 1–9, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/ejn.14432

[97] B. J. White, D. J. Berg, J. Y. Kan, R. A. Marino, L. Itti, and D. P. Munoz, "Superior colliculus neurons encode a visual saliency map during free viewing of natural dynamic video," *Nature Commun.*, vol. 8, pp. 1–9, Jan. 2017.

MIKHAIL STARTSEV received the Diplom degree in computational mathematics and informatics from the Lomonosov Moscow State University (LMSU), Russia, in 2015, where he was a member of the Graphics and Media Lab. He is currently pursuing the Ph.D. degree with the Chair of Human–Machine Communication, Department of Electrical and Computer Engineering, Technical University of Munich (TUM), Germany, as a part of an International Junior Research Group "Visual Efficient Sensing for the Perception-Action Loop" (VESPA). His researches focus on the human visual system, with an emphasis on the eye movements, and computer vision, in particular saliency.

MICHAEL DORR received the Dr.-Ing. degree in computer science from the University of Lübeck, Germany, in 2010. He completed his postdoctoral training at The Schepens Eye Research Institute, Harvard Medical School, before joining the Technical University of Munich (TUM) to lead the International Junior Research Group "Visual Efficient Sensing for the Perception-Action Loop" (VESPA), in 2014. He conducts research at the intersection of human and machine vision, with a focus on models of eye movements and selective attention in complex dynamic environments.

References

[1] K. Koch, J. McLean, R. Segev, M. A. Freed, M. J. Berry, V. Balasubramanian, and P. Sterling, “How much the eye tells the brain,” *Current Biology*, vol. 16, no. 14, pp. 1428–1434, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960982206016393

[2] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, J. Halszka, and J. van de Weijer, *Eye Tracking : A Comprehensive Guide to Methods and Measures.* Oxford University Press, 2011.

[3] R. Bixler and S. D’Mello, “Automatic gaze-based user-independent detection of mind wandering during computerized reading,” *User Modeling and User-Adapted Interaction*, vol. 26, no. 1, pp. 33–68, Mar 2016. [Online]. Available: https://doi.org/10.1007/s11257-015-9167-1

[4] Z. Bylinskii, P. Isola, A. Torralba, and A. Oliva, “How you look at a picture determines if you will remember it,” *Eye*, vol. 65, no. 70, p. 75, 2015.

[5] A. Ajanki, D. R. Hardoon, S. Kaski, K. Puolamäki, and J. Shawe-Taylor, “Can eyes reveal interest? implicit queries from gaze patterns,” *User Modeling and User-Adapted Interaction*, vol. 19, no. 4, pp. 307–339, Oct 2009. [Online]. Available: https://doi.org/10.1007/s11257-009-9066-4

[6] S. Wang, M. Jiang, X. Duchesne, E. Laugeson, D. Kennedy, R. Adolphs, and Q. Zhao, “Atypical visual saliency in autism spectrum disorder quantified through model-based eye tracking,” *Neuron*, vol. 88, no. 3, pp. 604–616, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0896627315008314

[7] M. Vidal, J. Turner, A. Bulling, and H. Gellersen, “Wearable eye tracking for mental health monitoring,” *Computer Communications*, vol. 35, no. 11, pp. 1306–1311, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366411003549

[8] B. Law, M. S. Atkins, A. E. Kirkpatrick, and A. J. Lomax, “Eye gaze patterns differentiate novice and experts in a virtual laparoscopic surgery training environment,” in *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ser. ETRA ’04. New York, NY, USA: ACM, 2004, pp. 41–48. [Online]. Available: http://doi.acm.org/10.1145/968363.968370

[9] S. Eivazi, R. Bednarik, M. Tukiainen, M. von und zu Fraunberg, V. Leinonen, and J. E. Jääskeläinen, “Gaze behaviour of expert and novice microneurosurgeons differs during observations of tumor removal recordings,” in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA ’12. New York, NY, USA: ACM, 2012, pp. 377–380. [Online]. Available: http://doi.acm.org/10.1145/2168556.2168641

[10] T. Tien, P. H. Pucher, M. H. Sodergren, K. Sriskandarajah, G.-Z. Yang, and A. Darzi, “Differences in gaze behaviour of expert and junior surgeons performing open inguinal hernia repair,” *Surgical Endoscopy*, vol. 29, no. 2, pp. 405–413, Feb 2015. [Online]. Available: https://doi.org/10.1007/s00464-014-3683-7

[11] S.-H. Lee, J.-K. Shin, and M. Lee, “Non-uniform image compression using biologically motivated saliency map model,” in *Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004.*, Dec 2004, pp. 525–530.

[12] C. Guo and L. Zhang, “A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression,” *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 185–198, Jan 2010.

[13] H. Hadizadeh and I. V. Bajić, “Saliency-aware video compression,” *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 19–33, Jan 2014.

[14] A. Mazumdar, B. Haynes, M. Balazinska, L. Ceze, A. Cheung, and M. Oskin, “Perceptual compression for video storage and processing systems,” in *Proceedings of the ACM Symposium on Cloud Computing*, ser. SoCC ’19. New York, NY, USA: ACM, 2019, pp. 179–192. [Online]. Available: http://doi.acm.org/10.1145/3357223.3362725

[15] E. Vig, M. Dorr, and D. Cox, “Space-variant descriptor sampling for action recognition based on saliency and eye movements,” in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 84–97.

[16] M. M. Salehin and M. Paul, “A novel framework for video summarization based on smooth pursuit information from eye tracker data,” in *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2017, pp. 692–697.

[17] K. Fujii, G. Gras, A. Salerno, and G.-Z. Yang, “Gaze gesture based human robot interaction for laparoscopic surgery,” *Medical Image Analysis*, vol. 44, pp.

196–214, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841517301809

[18] T. Louw and N. Merat, "Are you in the loop? Using gaze dispersion to understand driver visual attention during vehicle automation," *Transportation Research Part C: Emerging Technologies*, vol. 76, pp. 35–50, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X17300013

[19] A. Doshi and M. Trivedi, "Investigating the relationships between gaze patterns, dynamic vehicle surround analysis, and driver intentions," in *2009 IEEE Intelligent Vehicles Symposium*, June 2009, pp. 887–892.

[20] D. J. Liebling and S. Preibusch, "Privacy considerations for a pervasive eye tracking world," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, ser. UbiComp '14 Adjunct. New York, NY, USA: ACM, 2014, pp. 1169–1177. [Online]. Available: http://doi.acm.org/10.1145/2638728.2641688

[21] J. P. Hansen, K. Tørning, A. S. Johansen, K. Itoh, and H. Aoki, "Gaze typing compared with input by head and hand," in *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ser. ETRA '04. New York, NY, USA: ACM, 2004, pp. 131–138. [Online]. Available: http://doi.acm.org/10.1145/968363.968389

[22] D. Zhu, T. Gedeon, and K. Taylor, "Head or gaze?: Controlling remote camera for hands-busy tasks in teleoperation: A comparison," in *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*, ser. OZCHI '10. New York, NY, USA: ACM, 2010, pp. 300–303. [Online]. Available: http://doi.acm.org/10.1145/1952222.1952286

[23] L. E. Sibert and R. J. K. Jacob, "Evaluation of eye gaze interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '00. New York, NY, USA: ACM, 2000, pp. 281–288. [Online]. Available: http://doi.acm.org/10.1145/332040.332445

[24] C. Dickie, J. Hart, R. Vertegaal, and A. Eiser, "Lookpoint: An evaluation of eye input for hands-free switching of input devices between multiple computers," in *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*, ser. OZCHI '06. New York, NY, USA: ACM, 2006, pp. 119–126. [Online]. Available: http://doi.acm.org/10.1145/1228175.1228198

[25] A. Santella, M. Agrawala, D. DeCarlo, D. Salesin, and M. Cohen, "Gaze-based interaction for semi-automatic photo cropping," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '06. New York, NY, USA: ACM, 2006, pp. 771–780. [Online]. Available: http://doi.acm.org/10.1145/1124772.1124886

[26] M. Dorr, M. Böhme, T. Martinetz, and E. Barth, "Gaze beats mouse: a case study," *Proceedings of COGAIN*, pp. 16–19, 2007.

[27] V. D. Rajanna, "Gaze and foot input: Toward a rich and assistive interaction modality," in *Companion Publication of the 21st International Conference on Intelligent User Interfaces*, ser. IUI '16 Companion. New York, NY, USA: ACM, 2016, pp. 126–129. [Online]. Available: http://doi.acm.org/10.1145/2876456.2876462

[28] F. Roider and T. Gross, "I see your point: Integrating gaze to enhance pointing gesture accuracy while driving," in *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. AutomotiveUI '18. New York, NY, USA: ACM, 2018, pp. 351–358. [Online]. Available: http://doi.acm.org/10.1145/3239060.3239084

[29] S. Rivu, Y. Abdrabou, T. Mayer, K. Pfeuffer, and F. Alt, "GazeButton: Enhancing buttons with eye gaze interactions," in *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–7. [Online]. Available: https://doi.org/10.1145/3317956.3318154

[30] I. T. C. Hooge, D. C. Niehorster, M. Nyström, R. Andersson, and R. S. Hessels, "Is human classification by experienced untrained observers a gold standard in fixation detection?" *Behavior Research Methods*, vol. 50, no. 5, pp. 1864–1881, Oct 2018.

[31] R. S. Hessels, D. C. Niehorster, M. Nyström, R. Andersson, and I. Hooge, "Is the eye-movement field confused about fixations and saccades? A survey among 124 researchers," *Royal Society Open Science*, vol. 5, no. 8, pp. 1–23, 2018.

[32] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ser. ETRA '00. New York, NY, USA: ACM, 2000, pp. 71–78. [Online]. Available: http://doi.acm.org/10.1145/355017.355028

[33] C. D. Holland and O. V. Komogortsev, "Complex eye movement pattern biometrics: Analyzing fixations and saccades," in *2013 International Conference on Biometrics (ICB)*, June 2013, pp. 1–8.

[34] E. Kasneci, G. Kasneci, T. C. Kübler, and W. Rosenstiel, "The applicability of probabilistic methods to the online recognition of fixations and saccades in dynamic scenes," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '14. New York, NY, USA: ACM, 2014, pp. 323–326. [Online]. Available: http://doi.acm.org/10.1145/2578153.2578213

[35] G. Bird, C. Press, and D. C. Richardson, "The role of alexithymia in reduced eye-fixation in autism spectrum conditions," *Journal of Autism and Developmental*

Disorders, vol. 41, no. 11, pp. 1556–1564, Nov 2011. [Online]. Available: https://doi.org/10.1007/s10803-011-1183-3

[36] D. Melcher and C. L. Colby, "Trans-saccadic perception," *Trends in Cognitive Sciences*, vol. 12, no. 12, pp. 466–473, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364661308002325

[37] A. C. Schütz, D. I. Braun, and K. R. Gegenfurtner, "Eye movements and perception: A selective review," *Journal of Vision*, vol. 11, no. 5, pp. 9:1–9:30, 2011. [Online]. Available: http://dx.doi.org/10.1167/11.5.9

[38] M. Spering, A. C. Schütz, D. I. Braun, and K. R. Gegenfurtner, "Keep your eyes on the ball: Smooth pursuit eye movements enhance prediction of visual motion," *Journal of Neurophysiology*, vol. 105, no. 4, pp. 1756–1767, 2011. [Online]. Available: http://jn.physiology.org/content/105/4/1756

[39] A. M. Penkar, C. Lutteroth, and G. Weber, "Designing for the eye: Design parameters for dwell in gaze interaction," in *Proceedings of the 24th Australian Computer-Human Interaction Conference*, ser. OzCHI '12. New York, NY, USA: ACM, 2012, pp. 479–488. [Online]. Available: http://doi.acm.org/10.1145/2414536.2414609

[40] R. Engbert and R. Kliegl, "Microsaccades uncover the orientation of covert attention," *Vision Research*, vol. 43, no. 9, pp. 1035–1045, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0042698903000841

[41] A. Esteves, E. Velloso, A. Bulling, and H. Gellersen, "Orbits: Gaze interaction for smart watches using smooth pursuit eye movements," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ser. UIST '15. New York, NY, USA: ACM, 2015, pp. 457–466. [Online]. Available: http://doi.acm.org/10.1145/2807442.2807499

[42] S. Schenk, P. Tiefenbacher, G. Rigoll, and M. Dorr, "SPOCK: A smooth pursuit oculomotor control kit," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '16. New York, NY, USA: ACM, 2016, pp. 2681–2687. [Online]. Available: http://doi.acm.org/10.1145/2851581.2892291

[43] N. Anantrasirichai, I. D. Gilchrist, and D. R. Bull, "Fixation identification for low-sample-rate mobile eye trackers," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 3126–3130.

[44] J. Steil, M. X. Huang, and A. Bulling, "Fixation detection for head-mounted eye tracking based on visual similarity of gaze targets," in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '18. New York, NY, USA: ACM, 2018, pp. 23:1–23:9. [Online]. Available: http://doi.acm.org/10.1145/3204493.3204538

[45] P. Blignaut, “Fixation identification: The optimum threshold for a dispersion algorithm,” *Attention, Perception, & Psychophysics*, vol. 71, no. 4, pp. 881–895, May 2009. [Online]. Available: https://doi.org/10.3758/APP.71.4.881

[46] O. V. Komogortsev, D. V. Gobert, S. Jayarathna, D. H. Koh, and S. M. Gowda, “Standardization of automated analyses of oculomotor fixation and saccadic behaviors,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2635–2645, Nov 2010.

[47] R. Andersson, L. Larsson, K. Holmqvist, M. Stridh, and M. Nyström, “One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms,” *Behavior Research Methods*, vol. 49, no. 2, pp. 616–637, Apr 2017. [Online]. Available: https://doi.org/10.3758/s13428-016-0738-9

[48] A. Mihali, B. van Opheusden, and W. J. Ma, “Bayesian microsaccade detection,” *Journal of Vision*, vol. 17, no. 1, pp. 13:1–13:23, 01 2017. [Online]. Available: https://doi.org/10.1167/17.1.13

[49] M. Juhola, “Detection of nystagmus eye movements using a recursive digital filter,” *IEEE Transactions on Biomedical Engineering*, vol. 35, no. 5, pp. 389–395, May 1988.

[50] J. Otero-Millan, J. L. A. Castro, S. L. Macknik, and S. Martinez-Conde, “Unsupervised clustering method to detect microsaccades,” *Journal of Vision*, vol. 14, no. 2, pp. 18:1–18:17, 02 2014. [Online]. Available: https://doi.org/10.1167/14.2.18

[51] L. Larsson, M. Nyström, and M. Stridh, “Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit,” *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 9, pp. 2484–2493, Sept 2013.

[52] R. Zemblys, D. C. Niehorster, and K. Holmqvist, “gazeNet: End-to-end eye-movement event detection with deep neural networks,” *Behavior Research Methods*, vol. 51, no. 2, pp. 840–864, Apr 2019.

[53] V. I. Nicholls, G. Jean-Charles, J. Lao, P. de Lissa, R. Caldara, and S. Miellet, “Developing attentional control in naturalistic dynamic road crossing situations,” *Scientific Reports*, vol. 9, no. 1, p. 4176, 2019.

[54] D. J. Berg, S. E. Boehnke, R. A. Marino, D. P. Munoz, and L. Itti, “Free viewing of dynamic stimuli by humans and monkeys,” *Journal of Vision*, vol. 9, no. 5, pp. 19:1–19:15, 05 2009.

[55] E. Kasneci, G. Kasneci, T. C. Kübler, and W. Rosenstiel, “Online recognition of fixations, saccades, and smooth pursuits for automated analysis of traffic hazard perception,” in *Artificial Neural Networks*, P. Koprinkova-Hristova, V. Mladenov, and N. K. Kasabov, Eds. Springer International Publishing, 2015, pp. 411–434.

[56] L. Larsson, M. Nyström, R. Andersson, and M. Stridh, “Detection of fixations and smooth pursuit movements in high-speed eye-tracking data,” *Biomedical Signal Processing and Control*, vol. 18, pp. 145–152, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1746809414002031

[57] J. San Agustin, “Off-the-shelf gaze interaction,” Ph.D. dissertation, IT-Universitetet i København, 2010.

[58] L. Larsson, M. Nyström, H. Ardö, K. Åström, and M. Stridh, “Smooth pursuit detection in binocular eye-tracking data with automatic video-based performance evaluation,” *Journal of Vision*, vol. 16, no. 15, pp. 20:1–20:18, 12 2016.

[59] A. H. Dar, A. S. Wagner, and M. Hanke, “REMoDNaV: Robust eye movement detection for natural viewing,” *bioRxiv*, pp. 1–18, 2019. [Online]. Available: https://www.biorxiv.org/content/early/2019/04/26/619254

[60] M. Vidal, A. Bulling, and H. Gellersen, “Detection of smooth pursuits using eye movement shape features,” in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA ’12. New York, NY, USA: ACM, 2012, pp. 177–180. [Online]. Available: http://doi.acm.org/10.1145/2168556.2168586

[61] T. Santini, W. Fuhl, T. Kübler, and E. Kasneci, “Bayesian identification of fixations, saccades, and smooth pursuits,” in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA ’16. New York, NY, USA: ACM, 2016, pp. 163–170. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857512

[62] R. Zemblys, D. C. Niehorster, O. Komogortsev, and K. Holmqvist, “Using machine learning to detect events in eye-tracking data,” *Behavior Research Methods*, vol. 50, no. 1, pp. 160–181, Feb 2018. [Online]. Available: https://doi.org/10.3758/s13428-017-0860-3

[63] J. Goltz, M. Grossberg, and R. Etemadpour, “Exploring simple neural network architectures for eye movement classification,” in *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA ’19. New York, NY, USA: ACM, 2019, pp. 4:1–4:5. [Online]. Available: http://doi.acm.org/10.1145/3314111.3319813

[64] G. Boccignone and M. Ferraro, “Gaze shifts as dynamical random sampling,” in *2010 2nd European Workshop on Visual Information Processing (EUVIP)*, July 2010, pp. 29–34.

[65] A. Bahill, M. R. Clark, and L. Stark, “The main sequence, a tool for studying human eye movements,” *Mathematical Biosciences*, vol. 24, no. 3, pp. 191–204, 1975. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0025556475900759

[66] S. Hoppe and A. Bulling, “End-to-end eye movement detection using convolutional neural networks,” *CoRR*, vol. abs/1609.02452, 2016. [Online]. Available: http://arxiv.org/abs/1609.02452

[67] M. E. Bellet, J. Bellet, H. Nienborg, Z. M. Hafed, and P. Berens, “Human-level saccade detection performance using deep neural networks,” *Journal of Neurophysiology*, vol. 121, no. 2, pp. 646–661, 2019, PMID: 30565968. [Online]. Available: https://doi.org/10.1152/jn.00601.2018

[68] R. Kothari, Z. Yang, C. Kanan, R. Bailey, J. Pelz, and G. Diaz, “Gaze-in-wild: A dataset for studying eye and head coordination in everyday activities,” pp. 1–23, 2019.

[69] M. Dorr, T. Martinetz, K. R. Gegenfurtner, and E. Barth, “Variability of eye movements when viewing dynamic natural scenes,” *Journal of Vision*, vol. 10, no. 10, pp. 28:1–28:17, 08 2010.

[70] W. Wang, J. Shen, F. Guo, M.-M. Cheng, and A. Borji, “Revisiting video saliency: A large-scale benchmark and a new model,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 4894–4903.

[71] V. Leborán, A. García-Díaz, X. R. Fdez-Vidal, and X. M. Pardo, “Dynamic whitening saliency,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 893–907, May 2017.

[72] A. Borji and L. Itti, “CAT2000: A large scale fixation dataset for boosting saliency research,” *CoRR*, vol. abs/1505.03581, 2015. [Online]. Available: http://arxiv.org/abs/1505.03581

[73] M. Cerf, J. Harel, W. Einhaeuser, and C. Koch, “Predicting human gaze using low-level saliency combined with face detection,” in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 241–248. [Online]. Available: http://papers.nips.cc/paper/3169-predicting-human-gaze-using-low-level-saliency-combined-with-face-detection.pdf

[74] E. Vig, M. Dorr, and D. Cox, “Large-scale optimization of hierarchical features for saliency prediction in natural images,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 2798–2805.

[75] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, and N. E. O’Connor, “Shallow and deep convolutional networks for saliency prediction,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 598–606.

[76] N. Riche and M. Mancas, “Bottom-up saliency models for still images: A practical review,” in *From Human Attention to Computational Attention: A*

Multidisciplinary Approach, M. Mancas, V. P. Ferrera, N. Riche, and J. G. Taylor, Eds. New York, NY: Springer New York, 2016, pp. 141–175. [Online]. Available: https://doi.org/10.1007/978-1-4939-3435-5_9

[77] T. Judd, F. Durand, and A. Torralba, "A benchmark of computational models of saliency to predict human fixations," in *MIT Technical Report*, 2012.

[78] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "SALICON: Saliency in context," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1072–1080.

[79] S. Winkler and R. Subramanian, "Overview of eye tracking datasets," in *2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, July 2013, pp. 212–217.

[80] A. Borji, H. R. Tavakoli, D. N. Sihite, and L. Itti, "Analysis of scores, datasets, and models in visual saliency prediction," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013, pp. 921–928.

[81] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson, "Clustering of gaze during dynamic scene viewing is predicted by motion," *Cognitive Computation*, vol. 3, no. 1, pp. 5–24, Mar 2011. [Online]. Available: https://doi.org/10.1007/s12559-010-9074-z

[82] L. Jiang, M. Xu, T. Liu, M. Qiao, and Z. Wang, "DeepVS: A deep learning based video saliency prediction approach," in *The European Conference on Computer Vision (ECCV)*, September 2018, pp. 602–617.

[83] B. W. Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *Journal of Vision*, vol. 7, no. 14, pp. 4:1–4:17, 11 2007. [Online]. Available: https://doi.org/10.1167/7.14.4

[84] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein, "Saliency in VR: How do people explore virtual environments?" *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1633–1642, April 2018.

[85] H.-T. Cheng, C.-H. Chao, J.-D. Dong, H.-K. Wen, T.-L. Liu, and M. Sun, "Cube padding for weakly-supervised saliency prediction in 360° videos," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 1420–1429.

[86] Y. Rai, J. Gutiérrez, and P. Le Callet, "A dataset of head and eye movements for 360 degree images," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17. New York, NY, USA: ACM, 2017, pp. 205–210. [Online]. Available: http://doi.acm.org/10.1145/3083187.3083218

[87] E. J. David, J. Gutiérrez, A. Coutrot, M. P. Da Silva, and P. L. Callet, "A dataset of head and eye movements for 360° videos," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: ACM, 2018, pp. 432–437. [Online]. Available: http://doi.acm.org/10.1145/3204949.3208139

[88] A. Bolshakov, M. Gracheva, and D. Sidorchuk, "How many observers do you need to create a reliable saliency map in vr attention study?" in *Abstract Book of the European Conference on Visual Perception (ECVP)*, 2017, pp. 99–99.

[89] Y.-C. Su and K. Grauman, "Learning spherical convolution for fast features from 360° imagery," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 529–539. [Online]. Available: http://papers.nips.cc/paper/6656-learning-spherical-convolution-for-fast-features-from-360-imagery.pdf

[90] Y. Huang, M. Cai, Z. Li, and Y. Sato, "Predicting gaze in egocentric video by learning task-dependent attention transition," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[91] Y. Li, A. Kanemura, H. Asoh, T. Miyanishi, and M. Kawanabe, "A sparse coding framework for gaze prediction in egocentric video," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 1313–1317.

[92] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 740–757, March 2019.

[93] M. Kümmerer, T. S. A. Wallis, and M. Bethge, "Information-theoretic model comparison unifies saliency metrics," *Proceedings of the National Academy of Sciences*, vol. 112, no. 52, pp. 16 054–16 059, 2015. [Online]. Available: https://www.pnas.org/content/112/52/16054

[94] S. Mathe and C. Sminchisescu, "Dynamic eye movement datasets and learnt saliency models for visual action recognition," in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 842–856.

[95] J. Gutiérrez, E. David, Y. Rai, and P. L. Callet, "Toolbox and dataset for the development of saliency and scanpath models for omnidirectional/360° still images," *Signal Processing: Image Communication*, vol. 69, pp. 35–42, 2018, Salient360: Visual attention modeling for 360° Images. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596518304594

[96] D. Mahapatra, S. Winkler, and S.-C. Yen, "Motion saliency outweighs other low-level features while watching videos," in *Human Vision and Electronic*

Imaging XIII, B. E. Rogowitz and T. N. Pappas, Eds., vol. 6806, International Society for Optics and Photonics. SPIE, 2008, pp. 246–255. [Online]. Available: https://doi.org/10.1117/12.766243

[97] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD Proceedings*, vol. 96, no. 34. Portland, OR, USA: AAAI, 1996, pp. 226–231.

[98] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297.

[99] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[100] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov 1997.

[101] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrożny, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 799–804.

[102] X. Zeng and T. R. Martinez, "Distribution-balanced stratified cross-validation for accuracy estimation," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 12, no. 1, pp. 1–12, 2000. [Online]. Available: https://doi.org/10.1080/095281300146272

[103] P. Kasprowski and J. Ober, "Eye movements in biometrics," in *Biometric Authentication*, D. Maltoni and A. K. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 248–258.

[104] U. Saeed, "Eye movements during scene understanding for biometric identification," *Pattern Recognition Letters*, vol. 82, pp. 190–195, 2016, An insight on eye biometrics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865515001919

[105] J. Shao, C.-C. Loy, K. Kang, and X. Wang, "Slicing convolutional neural network for crowd video understanding," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 5620–5628.

[106] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017, pp. 11–19.

[107] S. shahabeddin Nabavi, M. Rochan, and Y. Wang, "Future semantic segmentation with convolutional LSTM," in *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, 2018, p. 137. [Online]. Available: http://bmvc2018.org/contents/papers/0559.pdf

[108] B. J. White, D. J. Berg, J. Y. Kan, R. A. Marino, L. Itti, and D. P. Munoz, "Superior colliculus neurons encode a visual saliency map during free viewing of natural dynamic video," *Nature Communications*, vol. 8, pp. 1–9, 2017.

[109] B. J. White, L. Itti, and D. P. Munoz, "Superior colliculus encodes visual saliency during smooth pursuit eye movements," *European Journal of Neuroscience*, vol. 8, no. 14263, pp. 1–9, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/ejn.14432

[110] O. V. Komogortsev and A. Karpov, "Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades," *Behavior Research Methods*, vol. 45, no. 1, pp. 203–215, 2013.

[111] T. Urruty, S. Lew, C. Djeraba, and D. A. Simovici, "Detecting eye fixations by projection clustering," in *14th International Conference of Image Analysis and Processing - Workshops (ICIAPW 2007)*, Sep. 2007, pp. 45–50.

[112] R. S. Hessels, D. C. Niehorster, C. Kemner, and I. T. C. Hooge, "Noise-robust fixation detection in eye movement data: Identification by two-means clustering (I2MC)," *Behavior Research Methods*, vol. 49, no. 5, pp. 1802–1823, Oct 2017. [Online]. Available: https://doi.org/10.3758/s13428-016-0822-1

[113] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 369–376. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143891

[114] I. T. Hooge, R. S. Hessels, and M. Nyström, "Do pupil-based binocular video eye trackers reliably measure vergence?" *Vision Research*, vol. 156, pp. 1–9, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0042698919300070

[115] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, pp. 1–14, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[116] M. Assens, X. G. i Nieto, K. McGuinness, and N. E. O'Connor, "Scanpath and saliency prediction on 360 degree images," *Signal Processing: Image Communication*, vol. 69, pp. 8–14, 2018, Salient360: Visual attention modeling for 360$^c irc$ Images. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596518306209

[117] Y.-C. Su and K. Grauman, "Kernel transformer networks for compact spherical convolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 9442–9451.

[118] Z. Zhang, Y. Xu, J. Yu, and S. Gao, "Saliency detection in 360° videos," in *The European Conference on Computer Vision (ECCV)*, September 2018, pp. 488–503.

[119] B. Coors, A. Paul Condurache, and A. Geiger, "SphereNet: Learning spherical representations for detection and classification in omnidirectional images," in *The European Conference on Computer Vision (ECCV)*, September 2018, pp. 518–533.

[120] P. Mazumdar and F. Battisti, "A content-based approach for saliency estimation in 360 images," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 3197–3201.

[121] M. Xu, L. Yang, X. Tao, Y. Duan, and Z. Wang, "Saliency prediction on omnidirectional images with generative adversarial imitation learning," *CoRR*, vol. abs/1904.07080, 2019. [Online]. Available: http://arxiv.org/abs/1904.07080

[122] B. Dedhia, J. Chiang, and Y. Char, "Saliency prediction for omnidirectional images considering optimization on sphere domain," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 2142–2146.

[123] P. Mazumdar, G. Arru, M. Carli, and F. Battisti, "Face-aware saliency estimation model for 360° images," in *2019 27th European Signal Processing Conference (EUSIPCO)*, Sep. 2019, pp. 1–5.

[124] S. Lee, D. Jang, J. Jeong, and E.-S. Ryu, "Motion-constrained tile set based 360-degree video streaming using saliency map prediction," in *Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, ser. NOSSDAV '19. New York, NY, USA: ACM, 2019, pp. 20–24. [Online]. Available: http://doi.acm.org/10.1145/3304112.3325614

[125] R. Monroy, S. Lutz, T. Chalasani, and A. Smolic, "SalNet360: Saliency maps for omni-directional images with cnn," *Signal Processing: Image Communication*, vol. 69, pp. 26–34, 2018, Salient360: Visual attention modeling for 360° Images. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596518304685

[126] Y. Zhang, F. Dai, Y. Ma, H. Li, Q. Zhao, and Y. Zhang, "Saliency prediction network for 360° videos," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–11, 2019.

[127] S. M. Wierda, H. van Rijn, N. A. Taatgen, and S. Martens, "Pupil dilation deconvolution reveals the dynamics of attention at high temporal resolution," *Proceedings of the National Academy of Sciences*, vol. 109, no. 22, pp. 8456–8460, 2012. [Online]. Available: https://www.pnas.org/content/109/22/8456

[128] N.-H. Liu, C.-Y. Chiang, and H.-C. Chu, "Recognizing the degree of human attention using EEG signals from mobile sensors," *Sensors*, vol. 13, no. 8, pp. 10 273–10 286, Aug 2013. [Online]. Available: http://dx.doi.org/10.3390/s130810273

[129] M. Hanke, N. Adelhöfer, D. Kottke, V. Iacovella, A. Sengupta, F. R. Kaule, R. Nigbur, A. Q. Waite, F. Baumgartner, and J. Stadler, "A studyforrest extension, simultaneous fMRI and eye gaze recordings during prolonged natural stimulation," *Scientific Data*, vol. 3, 2016.

[130] X. F. Amador, H. A. Sackeim, S. Mukherjee, R. Halperin, P. Neeley, E. Maclin, and D. Schnur, "Specificity of smooth pursuit eye movement and visual fixation abnormalities in schizophrenia: Comparison to mania and normal controls," *Schizophrenia Research*, vol. 5, no. 2, pp. 135–144, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/092099649190040X

[131] D. L. Levy, P. S. Holzman, S. Matthysse, and N. R. Mendell, "Eye Tracking Dysfunction and Schizophrenia: A Critical Perspective," *Schizophrenia Bulletin*, vol. 19, no. 3, pp. 461–536, 01 1993. [Online]. Available: https://doi.org/10.1093/schbul/19.3.461

[132] L. M. Williams, C. M. Loughland, E. Gordon, and D. Davidson, "Visual scanpaths in schizophrenia: is there a deficit in face recognition?" *Schizophrenia Research*, vol. 40, no. 3, pp. 189–199, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920996499000560

[133] R. G. Ross, A. Olincy, J. G. Harris, B. Sullivan, and A. Radant, "Smooth pursuit eye movements in schizophrenia and attentional dysfunction: Adults with schizophrenia, ADHD, and a normal comparison group," *Biological Psychiatry*, vol. 48, no. 3, pp. 197–203, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0006322300008258

[134] J. Egaña, C. Devia, R. Mayol, J. Parrini, G. Orellana, A. Ruiz, and P. Maldonado, "Small saccades and image complexity during free viewing of natural images in schizophrenia," *Frontiers in Psychiatry*, vol. 4, p. 37, 2013. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyt.2013.00037

[135] D. P. Crabb, N. D. Smith, and H. Zhu, "What's on TV? Detecting age-related neurodegenerative eye disease using eye movement scanpaths," *Frontiers in Aging Neuroscience*, vol. 6, p. 312, 2014. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnagi.2014.00312

[136] S. Dowiasch, B. Backasch, W. Einhäuser, D. Leube, T. Kircher, and F. Bremmer, "Eye movements of patients with schizophrenia in a natural environment," *European Archives of Psychiatry and Clinical Neuroscience*, vol. 266, no. 1, pp. 43–54, Feb 2016. [Online]. Available: https://doi.org/10.1007/s00406-014-0567-8

[137] A. Tales, J. Muir, R. Jones, A. Bayer, and R. J. Snowden, "The effects of saliency and task difficulty on visual search performance in ageing and alzheimer's disease," *Neuropsychologia*, vol. 42, no. 3, pp. 335–345, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0028393203001994

[138] E. Y. Uc, M. Rizzo, S. W. Anderson, J. Sparks, R. L. Rodnitzky, and J. D. Dawson, "Impaired visual search in drivers with Parkinson's disease," *Annals of Neurology*, vol. 60, no. 4, pp. 407–413, 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ana.20958

[139] D. M. Riby and P. J. Hancock, "Viewing it differently: Social scene perception in Williams syndrome and autism," *Neuropsychologia*, vol. 46, no. 11, pp. 2855–2860, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0028393208001917

[140] H. Duan, G. Zhai, X. Min, Z. Che, Y. Fang, X. Yang, J. Gutiérrez-Cillán, and P. L. Callet, "A dataset of eye movements for the children with autism spectrum disorder," in *Proceedings of the 10th ACM Multimedia Systems Conference*. ACM, 2019, pp. 255–260.

[141] M. Fetter and T. Haslwanter, "3D eye movements–basics and clinical applications," *Journal of Vestibular Research*, vol. 9, no. 3, pp. 181–187, 1999.

[142] M. Hanley, D. M. Riby, T. McCormack, C. Carty, L. Coyle, N. Crozier, J. Robinson, and M. McPhillips, "Attention during social interaction in children with autism: Comparison to specific language impairment, typical development, and links to social cognition," *Research in Autism Spectrum Disorders*, vol. 8, no. 7, pp. 908–924, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1750946714000725

[143] O. V. Komogortsev, Y. S. Ryu, D. H. Koh, and S. M. Gowda, "Instantaneous saccade driven eye gaze interaction," in *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*, ser. ACE '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 140–147. [Online]. Available: https://doi.org/10.1145/1690388.1690412

[144] M. Vidal, K. Pfeuffer, A. Bulling, and H. W. Gellersen, "Pursuits: Eye-based interaction with moving targets," in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 3147–3150. [Online]. Available: https://doi.org/10.1145/2468356.2479632

[145] E. Møllenbach, J. Hansen, and M. Lillholm, "Eye movements in gaze interaction," *Journal of Eye Movement Research*, vol. 6, no. 2, pp. 1–15, 2013.

[146] H. Drewes, M. Khamis, and F. Alt, "Smooth pursuit target speeds and trajectories," in *Proceedings of the 17th International Conference on*

Mobile and Ubiquitous Multimedia, ser. MUM 2018. New York, NY, USA: Association for Computing Machinery, 2018, pp. 139–146. [Online]. Available: https://doi.org/10.1145/3282894.3282913

[147] T. Pfeiffer, M. E. Latoschik, and I. Wachsmuth, "Evaluation of binocular eye trackers and algorithms for 3D gaze interaction in virtual reality environments," *Journal of Virtual Reality and Broadcasting*, vol. 5(2008), no. 16, pp. 1–14, 2008. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:0009-6-16605

[148] M. Khamis, C. Oechsner, F. Alt, and A. Bulling, "VRpursuits: Interaction in virtual reality using smooth pursuit eye movements," in *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, ser. AVI '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–8. [Online]. Available: https://doi.org/10.1145/3206505.3206522

[149] T. Kosch, M. Hassib, P. W. Woundefinedniak, D. Buschek, and F. Alt, "Your eyes tell: Leveraging smooth pursuit for assessing cognitive workload," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–13. [Online]. Available: https://doi.org/10.1145/3173574.3174010

[150] C. Donalek, S. G. Djorgovski, A. Cioc, A. Wang, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, A. Drake, S. Davidoff, J. S. Norris, and G. Longo, "Immersive and collaborative data visualization using virtual reality platforms," in *2014 IEEE International Conference on Big Data (Big Data)*, Oct 2014, pp. 609–614.

[151] M. Babaee, S. Tsoukalas, G. Rigoll, and M. Datcu, "Immersive visualization of visual data using nonnegative matrix factorization," *Neurocomputing*, vol. 173, pp. 245–255, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231215012606

[152] N. D. Smith, D. P. Crabb, F. C. Glen, R. Burton, and D. Garway-Heath, "Eye movements in patients with glaucoma when viewing images of everyday scenes," *Seeing and Perceiving*, vol. 25, no. 5, pp. 471–492, 2012. [Online]. Available: https://brill.com/view/journals/sp/25/5/article-p471_6.xml

[153] E. Kasneci, A. A. Black, and J. M. Wood, "Eye-tracking as a tool to evaluate functional ability in everyday tasks in glaucoma," *Journal of Ophthalmology*, vol. 2017, pp. 1–10, 2017.

[154] X. Wang, L. Gao, J. Song, and H. Shen, "Beyond frame-level CNN: Saliency-aware 3D CNN with LSTM for video action recognition," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 510–514, April 2017.

[155] K. Chaudhuri and S. Dasgupta, "Rates of convergence for the cluster tree," in *Advances in Neural Information Processing Systems 23*, J. D.

Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 343–351. [Online]. Available: http://papers.nips.cc/paper/4068-rates-of-convergence-for-the-cluster-tree.pdf

[156] B. P. Kent, A. Rinaldo, and T. Verstynen, "DeBaCl: A python package for interactive density-based clustering," *CoRR*, vol. abs/1307.8136, pp. 1–28, 2013. [Online]. Available: http://arxiv.org/abs/1307.8136

[157] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in Neural Information Processing Systems*, 2007, pp. 545–552.

[158] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5142–5154, Oct 2018.

Full Journal and Conference Publications

[1*] I. Agtzidis, M. Startsev, and M. Dorr. 360-degree video gaze behaviour: A ground-truth data set and a classification algorithm for eye movements. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, pages 1007–1015, New York, NY, USA, 2019. ACM.

[2*] J. E. Silberg, I. Agtzidis, M. Startsev, T. Fasshauer, K. Silling, A. Sprenger, M. Dorr, and R. Lencer. Free visual exploration of natural movies in schizophrenia. *European Archives of Psychiatry and Clinical Neuroscience*, 269(4):407–418, Jun 2019.

[3*] M. Startsev, I. Agtzidis, and M. Dorr. 1D CNN with BLSTM for automated classification of fixations, saccades, and smooth pursuits. *Behavior Research Methods*, 51(2):556–572, Apr 2019.

[4*] M. Startsev, I. Agtzidis, and M. Dorr. Characterizing and automatically detecting smooth pursuit in a large-scale ground-truth data set of dynamic natural scenes. *Journal of Vision*, 19(14):10:1–10:25, 12 2019.

[5*] M. Startsev and M. Dorr. 360-aware saliency estimation with conventional image saliency predictors. *Signal Processing: Image Communication*, 69:43–52, 2018. Salient360: Visual attention modeling for 360° Images.

[6*] M. Startsev and M. Dorr. Supersaliency: A novel pipeline for predicting smooth pursuit-based attention improves generalisability of video saliency. *IEEE Access*, 8:1276–1289, 2020.

[7*] M. Startsev, S. Göb, and M. Dorr. A novel gaze event detection metric that is not fooled by gaze-independent baselines. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, pages 22:1–22:9, New York, NY, USA, 2019. ACM.

Other Publications

[1†] I. Agtzidis, A. Alashqar, R. Judeh, M. Startsev, E. Vig, and M. Dorr. Eye movement prediction for naturalistic videos using C3D features. In *International Conference on Computer Vision Workshop on Mutual Benefits of Cognitive and Computer Vision*, Venice, Italy, Oct 2017. (presented as a poster).

[2†] I. Agtzidis, M. Startsev, and M. Dorr. In the pursuit of (ground) truth: A hand-labelling tool for eye movements recorded during dynamic scene viewing. In *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*, pages 65–68, Oct 2016.

[3†] I. Agtzidis, M. Startsev, and M. Dorr. Smooth pursuit detection based on multiple observers. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ETRA '16, pages 303–306, New York, NY, USA, 2016. ACM.

[4†] M. D. Ioannis Agtzidis, Mikhail Startsev. Individual smooth pursuit strategies in dynamic natural scene perception. In *The 19th European Conference on Eye Movements: Abstract book*, ECEM '17, pages 125–125, 2017. (presented as a talk).

[5†] J. E. Silberg, I. Agtzidis, M. Startsev, T. Fasshauer, K. Silling, A. Sprenger, M. Dorr, and R. Lencer. Smooth pursuit disturbances in schizophrenia during free visual exploration of dynamic natural scenes. In *The 19th European Conference on Eye Movements: Abstract book*, ECEM '17, pages 125–125, 2017. (presented as a talk).

[6†] M. Startsev, I. Agtzidis, and M. Dorr. Manual & automatic detection of smooth pursuit in dynamic natural scenes. In *The 19th European Conference on Eye Movements: Abstract book*, ECEM '17, pages 246–246, 2017. (presented as a poster).

[7†] M. Startsev, I. Agtzidis, and M. Dorr. Deep learning *vs.* manual annotation of eye movements. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, pages 1–3, New York, NY, USA, 2018. Association for Computing Machinery. (presented as a demo).

[8†] M. Startsev, I. Agtzidis, and M. Dorr. Sequence-to-sequence deep learning for eye movement classification. In *Proceedings of the 41st European Conference on Visual Perception (ECVP) 2018 Trieste*, pages 200–200. SAGE Publications, 2018. (presented as a talk).

[9†] M. Startsev, I. Agtzidis, and M. Dorr. Eye movements during movie viewing: An annotated data set and a benchmark for algorithmic eye movement detection. In *Proceedings of the 42nd European Conference on Visual Perception (ECVP) 2019 Leuven*, pages 105–105. SAGE Publications, 2019. (presented as a poster).

[10†] M. Startsev and M. Dorr. Increasing video saliency model generalizability by training for smooth pursuit prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2050–2053, Salt Lake City, UT, USA, June 2018. IEEE.

[11†] M. Startsev and M. Dorr. Classifying autism spectrum disorder based on scanpaths and saliency. In *2019 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 633–636, July 2019.

[12†] M. Startsev and M. Dorr. Improving the state of the art in eye movement event detection via trainable label correction. In *The 20th European Conference on Eye Movements: Abstract book*, ECEM '19, pages 135–135, 2019. (presented as a talk).

[13†] M. Startsev, A. T.-Y. Lee, and M. Dorr. Optimizing clustering-based smooth pursuit detection. In *Proceedings of the 40th European Conference on Visual Perception (ECVP) 2017 Berlin*, pages 24–24. SAGE Publications, 2017. (presented as a poster).

[14†] M. Startsev and R. Zemblys. Discussion and standardisation of the metrics for eye movement detection. ETRA '19. Tutorial presented at the the 11th ACM Symposium on Eye Tracking Research & Applications, 2019. `https://etra.acm.org/2019/tutorials.html`; slides available via `https://docs.google.com/presentation/d/1_BOS51Wgu_2t8CB_SUYOYD8x0dni2FMwyNF7xSBxKA4/`.