Detection of Smooth Pursuits Using Eye Movement Shape Features

Mélodie Vidal* Lancaster University

Andreas Bulling† University of Cambridge Lancaster University

Hans Gellersen‡ Lancaster University

Abstract

Smooth pursuit eye movements hold information about the health, activity and situation of people, but to date there has been no efficient method for their automated detection. In this work we present a method to tackle the problem, based on machine learning. At the core of our method is a novel set of *shape features* that capture the characteristic shape of smooth pursuit movements over time. The features individually represent incomplete information about smooth pursuits but are combined in a machine learning approach. In an evaluation with eye movements collected from 18 participants, we show that our method can detect smooth pursuit movements with an accuracy of up to 92%, depending on the size of the feature set used for their prediction. Our results have twofold significance. First, they demonstrate a method for smooth pursuit detection in mainstream eye tracking, and secondly they highlight the utility of machine learning for eye movement analysis.

CR Categories: I.5.2 [Pattern Recognition]: Design Methodology—Pattern analysis; I.5.4 [Pattern Recognition]: Applications—Signal Processing;

Keywords: Eye Tracking, Eye Movement Analysis, Smooth Pursuit Movements, Feature Extraction

1 Introduction

Smooth pursuit movements constitute a significant proportion of eye activity but have not been embraced in mainstream eye tracking. Smooth pursuits occur when the eyes "latch onto" a smoothly moving object and follow it at a linearly related velocity. They contrast saccadic movements that occur when the eyes move rapidly from one visual target to another, or follow an object that is suddenly displaced. On visual inspection smooth pursuit movements can often be easily made out, in particular if their occurrence is expected as for example in a controlled test.

The nature and mechanics of smooth pursuits are well understood [Robinson 1965] but in contrast to other eye movements - specifically saccades, fixations and blinks - there has been virtually no work on their automated detection [Koh et al. 2010]. Others before us have argued that detection of smooth pursuits would have useful applications in human-computer interaction, and noted that the lack of a generic method represents a significant gap in eye movement research [Grindinger 2006; Koh et al. 2010]. Some work however has been reported on the use of Kalman filters for processing smooth pursuits [Cecchin et al. 1990; Abd-almageed et al. 2002]. More recently, a common approach is to classify pursuits based on both dispersion and velocity of the signal, but the thresholds vary greatly depending on the data [San Agustin Lopez 2009; Larsson 2010; Liang and Lee 2008; Koh et al. 2010]. Another approach is to label smooth pursuits as what falls in between velocity thresholds set for fixations and saccades [Grindinger 2006; Komogortsev and Khan 2007]. Further insight into the classification problem was provided by work that analysed a larger variety of features for discrimination of different types of eye movement [Vidal et al. 2011].

In this work we propose a novel approach that is based on capturing a multitude of features and to use machine learning techniques for smooth pursuit detection. Machine learning is well established for pattern recognition but is novel in the context of eye movement analysis. The principle is to consider a set of features that represent incomplete information about a statistical phenomenon, in our case smooth pursuit movements, and to use a learning algorithm to select features that in combination predict the phenomenon.

The challenge and novelty in the approach lie in the development of features that capture the dynamics of smooth pursuits, while standard algorithms are used for feature selection and pattern classification. The features we introduce are based on the observation that smooth pursuits exhibit less variance in basic signal characteristics than other eye movements when analysed over several consecutive time windows. In order to capture these characteristics, we define *shape features* that represent signal shapes of characteristically low variance. For evaluation, we collect a large dataset of eye movements from 18 participants containing sequences of smooth pursuits amongst other movements and show the feasibility of detecting pursuit movements with a machine learning approach.

2 Pursuit Detection Using Shape Features

Our method for detection of smooth pursuits is based on the following concepts, which we will elaborate subsequently:

1. We use a set of features that capture different aspects of eye movement, and use machine learning to combine features for detection of smooth pursuits.

2. We use a sliding window approach for feature extraction, as this permits the translation of our results into a detection method that can be implemented in real-time.

3. In each time step, we extract seven signal measures, such as slope and range, to capture different aspects of the eye movement signal under the window. In an initial training phase, a representative sample of smooth pursuit movements is used to compute mean and standard deviation for each of these measures.

4. We compute shape features over a succession of windows, to capture whether measures' values conform with the trained model (in terms of a certain maximum variance from the mean) not only in a single window but sustained over several windows. The classification algorithm for detection of smooth pursuits is entirely based on the shape features.

*e-mail: m.vidal@lancaster.ac.uk
†e-mail: andreas.bulling@acm.org
‡e-mail: hwg@comp.lancs.ac.uk

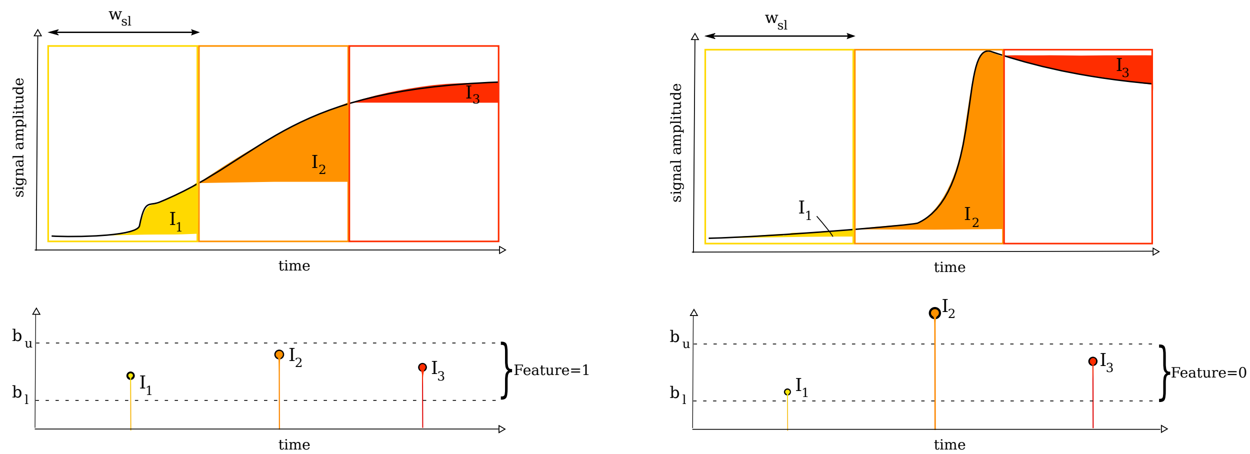

Figure 1: *Example binary shape feature calculated from the integral measure for a smooth pursuit (a) and a saccade (b) using the same window size w_{sl}. The integrals I_n stay within the given lower boundary b_l and upper boundary b_u over several consecutive windows for the smooth pursuit (feature value 1) while I_2 changes considerably in the case of the saccade (feature value 0).*

2.1 Extraction of Signal Measures

We use a sliding window approach for feature extraction. In this approach a window of size w_{sl} is moved over the data with a step size s_{sl}, and measures are computed to represent the signal under the window. This contrasts analysis of the whole data set after it has been captured in its entirety, and has the advantage that it can not only be used for offline analysis (as shown here) but also for online detection with a real-time algorithm.

In each step of the sliding window, we extract seven signal measures from the data in the current window, as described in Table 1). The measures are extracted separately from the vertical and horizontal movement components, as well as from a combination of both components. Some of these measures were inspired by previous work on eye movement analysis and other physiological signals [Miller 2008; Vidal et al. 2011]. Others were motivated by the specific characteristics of smooth pursuit movements.

By analysing the shape of different eye movements, we found that the slope of the signal is quite stable for fixations and typically increases sharply for saccades and blinks. It also increases for smooth pursuit movements but more slowly. The same applies to the mean velocity, range, variance, integral and energy of the signal. For the latter, smooth pursuit movements evolve more slowly than saccades but faster than fixations. Finally, the waveform length is the length a string would have if we would fit it to follow the shape of the signal. The waveform length measure should be very low for fixations, very high for blinks and in-between for saccades and smooth pursuits.

2.2 Shape Features

Shape features are complex features we developed on top of the signal measures. We refer to them as shape features, as they integrate information on the shape of smooth pursuits. The rationale for these features is that smooth pursuits tend to evolve more slowly than other movements, and in a relatively regular way. We capture this by analysing over consecutive windows whether signal measures conform with values expected for smooth pursuits. The range of expected values is derived from training data, and expressed in terms of lower and upper boundaries for each measure: $boundaries_{mes} = mean_{mes} \pm \alpha std_{mes}$. The coefficient α is a parameter for scaling variance.

Figure 1 illustrates how shape features are computed from signal measures, in this case the integral of the signal. Shape features evaluate to 1 when the measure remains within the boundaries associated with smooth pursuit over a succession of windows, and are otherwise set to 0.

Machine learning and classification of smooth pursuits versus other movements operate entirely on the level of shape features. Measures are not taken into account directly in the learning and classification process, but used solely to derive shape features.

3 Evaluation

3.1 Experiment

We evaluated the performance of the smooth pursuit detection algorithm on data collected from a previous experiment. While the goal of the user study was to collect ground truth annotated eye movement data containing smooth pursuit movements, saccades, fixations, vestibulo-ocular reflex movements and blinks, we only used the smooth pursuit movements, regarding the rest of the data as a "null" class for the classification, that is to say the data the smooth pursuits are classified against. The classification problem was thus a two-class problem. The study, which was approximately 20 minutes long, involved 18 participants to perform a sequence of these eye movements by looking at a visual stimulus shown on a computer screen. The smooth pursuits were both horizontal and vertical, with amplitudes of 2, 6, 12, 22 and 28° of visual angle, shown at speeds of 15 and 30°/s. The eye data we analyse here was recorded using a Dikablis mobile infrared eye tracker from Ergoneers GmbH running at 25Hz. Afterwards, the data was segmented into different eye movement instances, on which we evaluate the shape features method through classification. Our method is based on the assumption that nothing else than the eye data is available, which is why we do not use any information from the scene camera. This method can thus be exported to other eye movement acquisition techniques such as electro-oculography (EOG).

3.2 Data Processing and Classification

We started the evaluation by extracting the seven signal measures for five different window sizes from the horizontal component x, the vertical component y and for x and y combined ($w_{sl} \in \{80, 100, 120, 140, 160\}ms$). The minimum window size, $w_{sl} = 80ms$, was set by the limitations from the sampling frequency. The step sizes (respectively $s_{sl} \in \{50, 60, 70, 80, 90\}ms$)

Measure	Formula	Description	Combined Method
Slope	$S = \frac{ampl(t)-b}{t}$	Slope of a first-order polynomial fitted to the signal	$S(ampl_x) + S(ampl_y)$
Range	$R = max(ampl) - min(ampl)$	Range of the signal	$R(\sqrt{ampl_x^2 + ampl_y^2})$
Mean Velocity	$MV = mean(\lvert\frac{dampl(t)}{dt}\rvert)$	Mean of the point-to-point signal velocity	$MV(\sqrt{ampl_x^2 + ampl_y^2})$
Variance	$V = \frac{1}{N-1}\sum_{k=1}^{N}(ampl(k) - mean(ampl))^2$	Mean of the squared deviation of the signal from its mean	$V(\sqrt{ampl_x^2 + ampl_y^2})$
Integral	$I = (t_N - t_1)\frac{ampl(1)+ampl(N)}{2}$	Area under the signal using trapezoidal approximation	$I(ampl_x) + I(ampl_y)$
Energy	$E = \sum_{k=1}^{N}\lvert ampl(k)\rvert^2$	Integral of the squared signal	$E(ampl_x) + E(ampl_y)$
Waveform Length	$WL = \sum_{k=2}^{N}\lvert ampl(k) - ampl(k-1)\rvert$	Straight length of the signal	$WL(\sqrt{ampl_x^2 + ampl_y^2})$

Table 1: *Measures extracted from the signal in each step of the sliding window. $ampl$ is the amplitude of the eye position, N is the number of time samples in the window, b is the y-intercept of the polynomial. For the measures of integral and energy, the window is first normalised so that the initial position of the eye does not influence the measure's value.*

were designed to cover partially the previous window without covering more than half of it to avoid data redundancy.

For the evaluation, instead of using a detection algorithm we classify smooth pursuit against all other movements recorded during the experiment. This is thus a two-class classification problem. We used a k-nearest neighbours classifier ($k = 3$) and calculated its performance for each five window sizes and for different values of α. We used a 10-fold cross-validation: for each repetition, we selected 10% of all smooth pursuit instances for testing and the remaining 90% for training. All other movement instances were in the "null" class against which the pursuits were classified. The boundaries for the shape features were only calculated on the training set with $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$, using only one value of α per trial. We generated the shape features for 3, 4 and 5 consecutive windows. As we used the seven measures, calculated for horizontal channel, vertical channel and their combination, using three different numbers of consecutive windows, this resulted in a total number of 63 shape features. Finally, prior to the classification we ran a minimum redundancy maximum relevance (mRMR) feature selection algorithm and classified the testing instances for each possible feature set size. The classifier's output for each movement was then compared to the actual labels of the movement, generating the measure of accuracy we use in Figure 2.

3.3 Overall Recognition Performance

Figure 2 gives us an understanding of the influence of the feature set size on the performance. We plotted the feature set size against the accuracy of the classification, and Figure 2 shows that larger window sizes give better results. However, these results are reached with a larger number of features. Interestingly, we found a saturation effect of the window size: the overall accuracy of the 80ms window stays the same from a set of 25 features onwards even if the feature set size increases. This saturation effect appears later with larger window sizes: around 40 features for $w_{sl} = 120ms$ and 50 features for $w_{sl} = 160ms$. This means that it is possible to get a better performance with a larger window size but it comes with the trade-off of manipulating a set of features that grows with the desired performance. Figure 2 also indicates the best and worst accuracy for each window size depending on the α selected. The coloured area is the span of the different results we obtained by sweeping the α value for the boundaries calculation. We can see that generally the best performances are reached for an α of 0.3 or 0.4. If α is too big or too small (not shown here as the worst accuracy always came with the biggest α), the classifier is not as accurate.

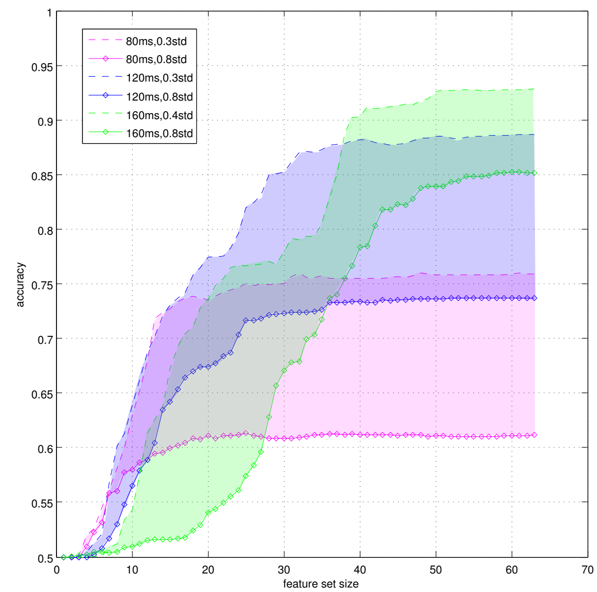

Figure 2: *The accuracy of the classification depending on the feature set size for window sizes $w_{sl} = 80ms, 120ms, 160ms$. The plot shows the best and worst accuracy for each of these three window sizes, depending on the coefficient α. A saturation effect appears on the feature set size: although it does not reach a maximum accuracy as good as larger window sizes, a smaller window size will reach its top accuracy with a smaller number of features.*

4 Discussion

4.1 Shape-Based Detection of Smooth Pursuit

The shape-based approach proved to be well suited for smooth pursuit detection. The proposed shape features efficiently encode the temporal characteristics of smooth pursuit movements. While the approach was developed for detecting smooth pursuits, it is generic and thus not limited to only such movements. In order to use it for other movements, the only modification would be to code the shape features differently by adapting the pattern they seek to recognise in the data. For smooth pursuits the pattern was stability of values over several windows, while for saccades it could be a drastic change between two windows. In future work we plan to apply it on the problem of detecting other types of eye movements, such as certain instances of the vestibulo-ocular reflex.

One drawback of the proposed machine learning based approach is that it requires a certain amount of training data. This is in contrast to established methods that rely on single features calculated using signal processing techniques, such as the velocity profile or the dispersion of gaze points. For a real-world implementation, however, such data would only need to be recorded once and used to identify the initial parameters and train the classifier. During operation, the trained parameters, such as the window size or the boundaries, can then be adapted online to the specific user or task at hand. In combination with unsupervised machine learning techniques, such as clustering [Jain et al. 1999] or self organising maps [Kohonen and Somervuo 1998], this may allow us to develop a robust, fully automatic smooth pursuit recognition system.

Finally, the shape features provide an interesting opportunity for a future extension of the algorithm. At the moment, the features are binary and do not provide any information on how close the shape of the signal is to that of a typical smooth pursuit movement. The difference between the boundary and the features values could be used as a measure of the quality of the classifier's prediction output. This so-called confidence value would then allow the system to take more fine-grained decisions or to rank or compare different instances of smooth pursuit movements. This would potentially improve the recognition rate on a more diverse set of pursuits than the subset investigated in this work.

4.2 Features

From the analysis of the different feature set sizes we can derive some interesting findings. The number of features considerably influences the overall recognition performance. Generally speaking, the more features are used during classification the better the recognition performance. This validates our approach for not using a single feature to discriminate smooth pursuits against other movements but rather use a combination of features. However, as can be seen from Figure 2, recognition performance does not improve significantly beyond a saturation point that varies together with the window size. The detection of this saturation point is important as it means manipulating less features for the same performance, which accelerates the data analysis.

We observed that during feature extraction, the larger the window size the better the performance (see Figure 2). This makes sense given that using a larger window size more information is available for calculating the different shape features. Using such a large window size may cause problems for real-world applications as a large window size prevents the algorithms from detecting smooth pursuit movements shorter than this window size. Thus, for detecting smooth pursuits of arbitrary length, a small window size is generally desirable. In addition, the sliding window approach can be an obstacle to detecting the precise onset and offset of the movement.

5 Conclusion

In this work we have focused on the problem of detecting smooth pursuit movements and introduced a novel approach to tackle it, based on machine learning. We have described a concrete method that is centred around shape features, a new type of feature we designed to capture signal shape characteristics that are stable over consecutive windows. The method was evaluated based on a multi-user dataset of different types of eye movement, achieving a detection performance of up to 92%. More importantly, the evaluation has also provided insight into the influence of factors such as window size and feature set size on overall performance.

We draw a number of key conclusions from this work. First, it demonstrates the feasibility of smooth pursuit detection with a robust approach that, importantly, lends itself to implementation in a real-time system. This is significant in view of enabling applications in everyday settings, for instance context-aware support of everyday situations. Secondly, the work highlights the utility of machine learning for eye movement analysis. Eye movement and visual behaviour are complex and as a result hard to describe with one-fits-all features. As we have shown, machine learning has the potential to capture complex patterns on the basis of larger feature sets. Thirdly, we have shown that shape features are effective for capturing temporal behaviour of signals. We have introduced the shape features specifically for smooth pursuit detection but the concept is generic and can be applied to learning and detection of other types of eye activities.

References

ABD-ALMAGEED, W., FADALI, M. S., BEBIS, G., SAMI, M., AND BEBIS, F. G. 2002. A non-intrusive kalman filter-based tracker for pursuit eye movement. In *In American Control Conference*.

CECCHIN, T., SAUTER, D., BRIE, D., AND MARTIN, B. 1990. On-line separation of smooth pursuit and saccadic eye movements. In *Proc. of the 12th International Conference of the IEEE Engineering in Medicine and Biology Society*, 777 –778.

GRINDINGER, T. 2006. *Eye Movement Analysis and Prediction with the Kalman Filter*. Master's thesis, Clemson University, USA.

JAIN, A. K., MURTY, M. N., AND FLYNN, P. J. 1999. Data clustering: a review. *ACM Computing Surveys 31*, 264–323.

KOH, D. H., MUNIKRISHNE GOWDA, S., AND KOMOGORTSEV, O. 2010. Real time eye movement identification protocol. In *Extended Abstracts of the 28th International Conference on Human Factors in Computing Systems*, 3499–3504.

KOHONEN, T., AND SOMERVUO, P. 1998. Self-organizing maps of symbol strings. *Neurocomputing 21*, 1-3, 19 – 30.

KOMOGORTSEV, O. V., AND KHAN, J. I. 2007. Kalman filtering in the design of eye-gaze-guided. *System*, 679–689.

LARSSON, L. 2010. *Event detection in eye-tracking data*. Master's thesis, Lund University.

LIANG, Y., AND LEE, J. D. 2008. Driver cognitive distraction detection using eye movements. In *Passive Eye Monitoring*, R. I. Hammoud, Ed., Signals and Communication Technology. Springer Berlin Heidelberg, 285–300.

MILLER, C. J. 2008. *Real-Time Feature Extraction and Classification of Prehensile EMG Signals*. Master's thesis, San Diego State University, USA.

ROBINSON, D. 1965. The mechanics of human smooth pursuit eye movement. *The Journal of Physiology 180*, 3 (Oct.), 569–591.

SAN AGUSTIN LOPEZ, J. 2009. *Off-the-shelf Gaze Interaction*. PhD thesis, IT University of Copenhagen.

VIDAL, M., BULLING, A., AND GELLERSEN, H. 2011. Analysing eog signal features for the discrimination of eye movements with wearable devices. In *Proc. of the 1st International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*, 15–20.