Combining Gaze Estimation and Optical Flow for Pursuits Interaction

Mihai Bâce*, Vincent Becker*
Department of Computer Science
ETH Zürich
mbace | vbecker @ethz.ch

Chenyang Wang*
Dept. of Information Technology and Electrical Engineering, ETH Zürich
wangche@ethz.ch

Andreas Bulling
Institute for Visualisation and Interactive Systems, Univ. of Stuttgart
andreas.bulling@vis.uni-stuttgart.de

ABSTRACT

Pursuit eye movements have become widely popular because they enable spontaneous eye-based interaction. However, existing methods to detect smooth pursuits require special-purpose eye trackers. We propose the first method to detect pursuits using a single off-the-shelf RGB camera in unconstrained remote settings. The key novelty of our method is that it combines appearance-based gaze estimation with optical flow in the eye region to jointly analyse eye movement dynamics in a single pipeline. We evaluate the performance and robustness of our method for different numbers of targets and trajectories in a 13-participant user study. We show that our method not only outperforms the current state of the art but also achieves competitive performance to a consumer eye tracker for a small number of targets. As such, our work points towards a new family of methods for pursuit interaction directly applicable to an ever-increasing number of devices readily equipped with cameras.

CCS CONCEPTS

• **Human-centered computing** → **Interaction techniques**.

KEYWORDS

Smooth Pursuit, Pursuit Interaction, Gaze Estimation, Optical Flow

ACM Reference Format:
Mihai Bâce, Vincent Becker, Chenyang Wang, and Andreas Bulling. 2020. Combining Gaze Estimation and Optical Flow for Pursuits Interaction. In *Symposium on Eye Tracking Research and Applications (ETRA '20 Full Papers), June 2–5, 2020, Stuttgart, Germany.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3379155.3391315

1 INTRODUCTION

In recent years, *Pursuits* has emerged as the first gaze interaction technique that allows for both natural and spontaneous, calibration-free interaction with dynamic user interfaces. It relies on smooth pursuit eye movements that are performed when following a target moving along a continuous trajectory at an appropriate speed. By correlating these eye trajectories with those of on-screen objects, the single target the user is following with his/her eyes can be robustly identified. The original paper introducing *Pursuits* [Vidal et al. 2013] has therefore spurred a large number of works resulting in both variants of the method itself [Khamis et al. 2016a] as well as applications in various interactive settings such as public displays [Khamis et al. 2016b], smartwatches [Esteves et al. 2015], or virtual reality [Khamis et al. 2018b].

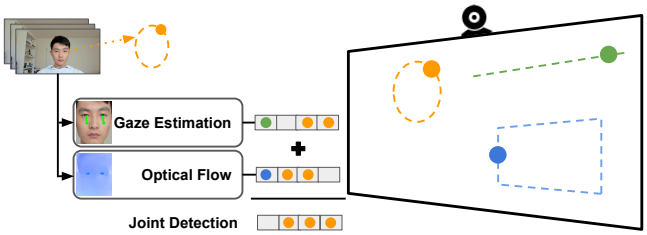

Figure 1: We propose a novel method to detect pursuits using a single off-the-shelf RGB camera. Our method jointly analyses the eye gaze direction and optical flow in the eye region to identify the target the user is following.

However, all of these works require special-purpose eye tracking equipment, i.e. dedicated devices built and sold specifically for this task. Unlike other gaze interaction techniques such as dwelling [Bednarik et al. 2009] or gaze gestures [Bâce et al. 2016; Drewes and Schmidt 2007], Pursuits does not require a calibrated eye tracker. However, robust tracking of the relative movement of the eyes is fundamental to the technique. Moreover, dedicated eye trackers may not always be available or may be difficult to integrate into the small form factor of some devices, such as mobile phones. Several works have proposed methods that implement Pursuit-like interactions for other body parts [Clarke et al. 2016], such as the hands or arms [Carter et al. 2016], based on off-the-shelf cameras and computer vision. However, computer vision-based techniques for Pursuit interaction using gaze have not been explored as of yet.

We propose a novel method to detect pursuits using a single off-the-shelf RGB camera in unconstrained remote settings. Our method combines appearance-based gaze estimation and optical flow into a joint pipeline and is, therefore, able to capture both the gaze direction and the eye movement dynamics during a pursuit movement (Figure 1). On the one hand, the full-face appearance-based gaze estimator [Zhang et al. 2017] predicts the 2D point of regard from a single image. By correlating the gaze estimates and the position of the moving target using the Pearson product-moment correlation, we obtain a first target candidate. On the other hand, our method uses dense optical flow [Farnebäck 2003] in the eye region extracted from a series of normalised face images to estimate the eye movement direction. Our method then correlates the eye

*The first three authors contributed equally to this research.

ETRA '20 Full Papers, June 2–5, 2020, Stuttgart, Germany

ACM ISBN 978-1-4503-7133-9/20/06…$15.00
https://doi.org/10.1145/3379155.3391315

movement directions and the target motions using cosine similarity to obtain a second target candidate. By combining the two different perspectives on eye movements and aggregating the outputs from both approaches, our method shows increased robustness.

The specific contributions of our work are two-fold. First, we present a novel method to detect pursuits in unconstrained remote settings, which does not require any special-purpose eye tracking equipment but only a standard off-the-shelf RGB camera. Second, we evaluate our method in a 13-participant user study and show that it outperforms the current state of the art EyeFlow [Hassoumi et al. 2019] by a large margin. For a small number of targets, our method achieves over 90% accuracy (the percentage of correctly identified targets) and is even competitive with a consumer eye tracker. As such, our work paves the way for a new class of methods that enable spontaneous Pursuits interaction in the wild.

2 RELATED WORK

Our research relates to previous work on 1) smooth pursuit interaction, 2) gaze estimation, and 3) optical flow estimation.

2.1 Smooth Pursuit Interaction

Selecting a target from multiple user interface (UI) elements is a key task in gaze-based interaction [Sibert and Jacob 2000]. Pursuits [Vidal et al. 2013] is a recent alternative to pointing that has a wide range of applications, including interaction with public displays [Khamis et al. 2015, 2016b] or smartwatches [Esteves et al. 2015]. Other works investigated the use of *Pursuits* with other body parts, e.g. to control ambient devices with the hands [Velloso et al. 2016] or to provide secure input of PINs [Cymek et al. 2014] or passwords [Almoctar et al. 2018]. An advantage of the technique is that it is robust to partially hidden trajectories [Mattusch et al. 2018]. Further work optimised the method itself [Velloso et al. 2018] and extended possible use cases to novel tasks, such as text entry [Drewes et al. 2019]. Rather than using *Pursuits* for interaction, others instead used it as an implicit calibration method for eye trackers [Celebi et al. 2014; Pfeuffer et al. 2013].

While *Pursuits* enables novel interactive experiences, the need for special-purpose eye tracking equipment hinders its broader applicability. In contrast, our method only requires a single RGB camera, e.g. a webcam. The closest work to ours is EyeFlow [Hassoumi et al. 2019], but that work was designed for a wearable, head-mounted setting in which the camera is mounted close to the eyes, and therefore requires a high-resolution eye image. Moreover, their method assumes that the camera is rigidly attached to the user's head – an assumption that no longer holds in remote settings. As such, we are first to propose a method that tackles the particularly challenging remote setting in which users are at a distance and the user's eyes constitute only a small, low-resolution part of the full camera view.

2.2 Gaze Estimation

Gaze estimation is the task of estimating a user's 3D gaze direction or 2D point of regard. While early works required pupil detection or (infrared) illumination [Li et al. 2005; Morimoto et al. 2000], more recent methods directly use the face's and eye's appearance leveraging large datasets and machine learning [Zhang et al. 2015, 2019b]. For example, Zhang et al. proposed a full-face appearance-based gaze estimator trained on the MPIIFaceGaze dataset [Zhang et al. 2017], while Krafka et al. introduced a convolutional neural network (CNN) trained on the large-scale GazeCapture dataset for mobile gaze estimation [Krafka et al. 2016]. Learning-based methods have already outperformed feature- or model-based approaches and have shown increased robustness in unconstrained settings even without calibration. However, angular errors between 4° and 6° still prevent them from being used in high-accuracy applications.

Part of our method uses one such generic appearance-based gaze estimator [Zhang et al. 2019a; Zhang et al. 2017] trained on the GazeCapture dataset. User or model adaptation through calibration could have further increased its performance [Zhang et al. 2018a], but this contradicts the concept of spontaneous calibration-free interaction. Instead, for increased robustness, we propose to additionally capture eye movement dynamics using optical flow.

2.3 Optical Flow Estimation

Optical flow estimation is a computational task in computer vision with the goal of estimating the apparent motion of the pixels in the image plane. It is widely used in applications such as object detection and tracking [Hua et al. 2018], semantic segmentation [Sevilla-Lara et al. 2016], or activity recognition [Simonyan and Zisserman 2014] because it serves as an approximation of the real physical motion. Methods that estimate sparse optical flow only examine a reduced number of pixels or features in an image [Lucas and Kanade 1981]. In contrast, dense optical flow estimates the flow vectors for all pixels in the entire image [Farnebäck 2003], which leads to increased performance at the cost of being computationally slower. The current state of the art in optical flow estimation is end-to-end deep learning models such as FlowNet2 [Ilg et al. 2017].

We are the first to leverage both static and dynamic information about eye motion to detect smooth pursuit eye movements. To estimate the motion of the eye, for maximum performance, we leverage a dense optical flow method [Farnebäck 2003].

3 METHOD

To detect whether a user is following a moving UI element, our method combines appearance-based gaze estimation and optical flow in the eye region to jointly analyse the eye movement dynamics during a pursuit (Figure 2 for an illustration). Given a sequence of images, our method first detects the user's face and the facial landmarks in each image individually. Face bounding boxes are detected using three multi-task CNNs [Zhang et al. 2016] and an hourglass network [Deng et al. 2018] then predicts 68 facial landmarks inside the detected bounding box, which are then further used in the single components.

3.1 Gaze Estimation

The goal of the gaze estimation component is to correlate the 2D gaze estimates from a window of N images to the 2D coordinates of all moving UI elements. The output of this component is either a candidate target if the correlation value is over a predefined threshold, or *None* if no target can be detected.

Face Image Normalisation. Face image normalisation is an effective preprocessing step in appearance-based gaze estimation [Zhang et al. 2018b, 2015]. By rotating and scaling the input image, normalisation cancels out differences in user appearance caused by

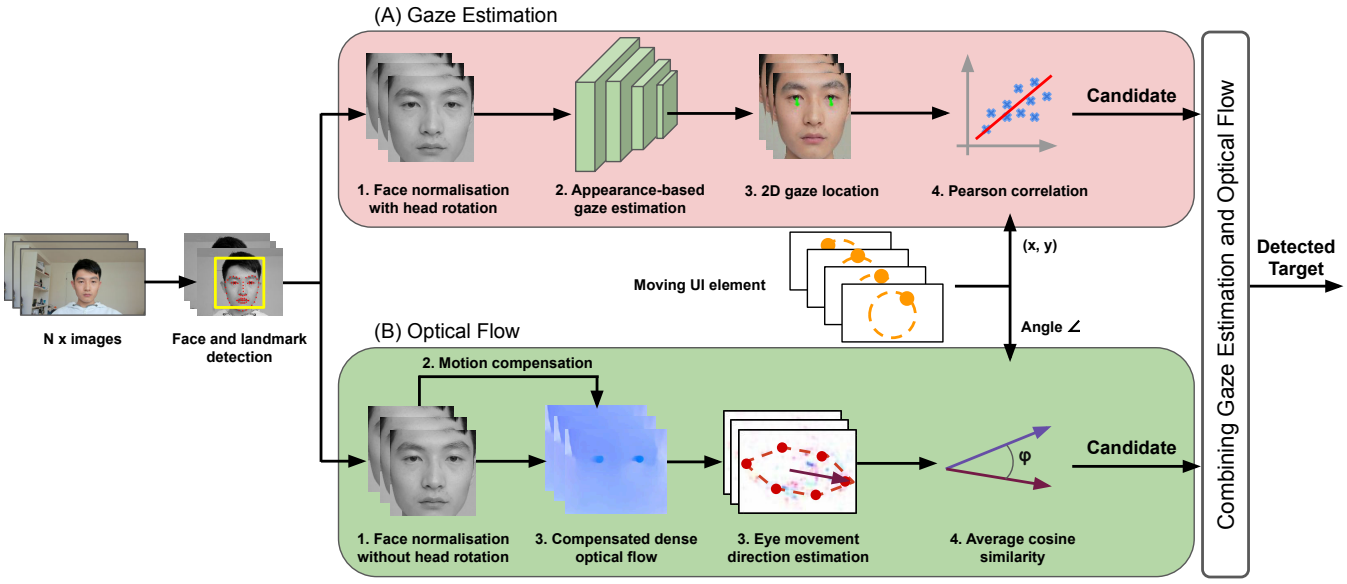

Figure 2: Overview of our method that consists of two components running gaze estimation and optical flow to first independently correlate eye movements to the objects' motion trajectories and thereby estimate the target the user is following. These estimates are then combined to make a joint decision about the single most likely followed on-screen target.

the user-camera distance or different hardware setups. Normalisation requires an estimate of the user's head pose that can be defined as the 3D translation and rotation of the head relative to the camera. To estimate the 3D head pose, we opted to use the method proposed in [Bâce et al. 2019] that has shown improved robustness to appearance variability across users and head pose angles. Given the 3D head pose, the input image is warped to a normalised space with fixed parameters and cropped to a size of 448x448 px (Figure 2 A1). We use these images to train the appearance-based gaze estimator.

Appearance-based Gaze Estimation. Our method uses a full face appearance-based gaze estimator [Zhang et al. 2019a; Zhang et al. 2017] trained on the GazeCapture dataset [Krafka et al. 2016] to predict 3D gaze directions. While collected for gaze estimation on mobile devices, training on this dataset is still beneficial given its large number of participants (over 1'400) and training images (over two million). The gaze estimation is, therefore, able to better abstract away data-specific biases caused, for example, by variability in user appearance. On GazeCapture the average angular error is around 4.3° while in a cross-dataset setting the error increases to around 5.3° on the MPIIFaceGaze dataset [Zhang et al. 2017], which is state of the art for appearance-based gaze estimation.

Motion Correlation. For every image captured by the camera, the appearance-based gaze estimator outputs the 3D gaze direction in terms of *yaw* and *pitch* angles in the camera coordinate system. We intersect this vector with the XY camera plane ($z = 0$) to obtain the 2D point of gaze. Given multiple 2D gaze points over time, we use the Pearson product-moment correlation for the X and Y axis separately to identify which moving target is the most similar. We also account for cases in which it is impossible to calculate the correlation because one series has a variance of zero, i.e. the series together constitute a horizontal or vertical line, by rotating that line including the corresponding gaze estimates by 45°, similar to [Velloso et al. 2018]. In the original approach [Vidal et al. 2013], the axis with zero variance is discarded. However, this results in an unnecessary loss of information. When the correlation values for both axes are over a threshold, a candidate target is identified. In the case of multiple candidates, the method selects the one with the maximum sum of both the horizontal and vertical similarity.

3.2 Optical Flow

The goal of the optical flow component is to complement the gaze estimator and provide more robust correlations between the movement of the eyes and UI elements. This dual-branch approach exploits a fundamental difference between optical flow and gaze estimation. Gaze estimation only considers a single image, i.e. predicts the 3D gaze direction or 2D point of regard from one image frame. Optical flow, on the other hand, can capture the movement between consecutive frames and analyse in which direction the eyes move, and outputs a motion vector instead of an absolute point of regard.

Face Image Normalisation. The normalisation step is very similar to the one performed for the gaze estimation task except for one key difference. In the original method [Zhang et al. 2018b], the input image was rotated so that the X-axis of the head coordinate system is parallel to the X-axis of the camera coordinate system. By applying such a rotation, we would also rotate the movement direction of the eyes, which would no longer match the movement of the UI element shown on the screen. Therefore, in the face image normalisation step for optical flow, we transform the input image to a space where the normalised camera points at the centre of the face, yet without cancelling out the head roll, i.e. rotation of the X axis. The image is still scaled so that the resulting normalised face has a similar appearance in terms of shape and size across users. Normalising the input image is beneficial to the optical flow

estimation task since the same set of parameters (e.g. similarity threshold) will work for different users or hardware configurations.

Dense Optical Flow and Motion Compensation. We use dense optical flow to estimate the pattern of apparent motion between two image frames [Farnebäck 2003]. If the camera has a clear view of only the eye (e.g. EyeFlow [Hassoumi et al. 2019], the optical flow between two consecutive frames provides robust information on eye motion. In contrast, in remote settings, users are at a distance and the camera cannot capture the eye in high resolution. As such, calculating the optical flow between every two consecutive frames introduces too much noise relative to the little actual motion in the eye. In our method, we therefore instead calculate the optical flow every couple of frames, leading to more robust motion estimates. We set this parameter, the *compute-rate*, to 5 empirically.

Our method estimates dense optical flow between two normalised face images (Figure 2 B2). In general, optical flow is either due to object motion in the image or camera motion. Image normalisation transforms the input image to an image with fixed camera parameters. Since the position of the normalised camera depends on the detected facial landmarks and the head pose estimate, it slightly differs between frames, which causes the camera to appear as if it was moving. Directly estimating the optical flow would lead to incorrect flow vectors. To address this, we apply motion compensation to decompose visual motion into dominant motion caused by the camera or the background as well as residual motion caused by the user. Specifically, we use a method that estimates a 2D affine transformation model between two frames to cancel out the dominant motion, i.e. the normalised camera motion in our case [Jain et al. 2013]. The output of this step is the motion compensated optical flow.

Eye Movement Direction Estimation. To estimate the eye movement direction, we use the compensated dense optical flow in the eye region. Our method requires only one eye, and in our implementation, we chose the user's right eye. We select the flow vectors that are inside the eye polygon defined by the six eye landmarks (Figure 2 B3). We then compute Shannon's entropy to measure the amount of disagreement among the flow vectors as proposed in [Hassoumi et al. 2019] and drop frames with an entropy exceeding a certain threshold. The eye movement direction is calculated by averaging the flow vectors in Cartesian space.

Eye and Object Motion Correlation. For the same pair of time points for which the optical flow vector was computed, we calculate a motion vector for each object by computing the difference vector between the objects' positions. We can then calculate the similarity between eye motion and each object's motion by computing the cosine similarity of the optical flow vector and the object motion vector. This value ranges from -1 for vectors pointing in opposite directions (a 180° angle) to 1 for vectors which point in the same direction (a 0° angle). To obtain similarities between a sequence of optical flow motion vectors and a set of objects, we compute the mean similarity μ_o over the similarities of each optical flow vector to the corresponding objects' motion vectors for each object o. While the object motion is a precisely known quantity, the optical flow vectors might contain outliers pointing away from the general direction, which are detrimental to the goal of a robust average similarity. To avoid this effect, we remove similarities which lie outside of the interval $[\mu_o - 2\sigma_o, \mu_o + 2\sigma_o]$, where σ_o is the standard deviation over all similarities for object o. Finally, among all objects whose mean similarities exceed a certain threshold, we select the one with the maximum mean similarity to the target. If the threshold is not reached at all, we return *None*.

3.3 Joint Gaze Estimation and Optical Flow

To combine our two sub-components, we calculate target detections from each of them. This can either be a specific object or *None* if no object was determined to be the target for that window. We simply merge detections by checking whether they agree or not, where *None* counts as neutral, i.e. if one of the two detections is *None*, we use the other as the merged window detection (Figure 1). If both detections are *None* or they contradict each other, we select *None*. Naturally, if the detections agree, we select that common detection. Running our method results in a sequence of merged detections representing (overlapping) windows of frames. To improve the robustness of our method, instead of simply selecting the first merged detection that is not *None* as the target, we take several consecutive detections into account. We examined several voting schemes, both on fixed or variable length voting windows of detections. The best voting scheme we found is to select the object which is the first to be jointly detected as target three times, i.e. obtains three votes. These three detections do not have to be in direct succession.

Overall, our method has several free parameters: the window size, i.e. the number of image frames in a window; the stride, i.e. the number of frames two subsequent windows differ in; the threshold for the gaze correlation values; the threshold for the cosine similarity in the optical flow component; and the number of necessary votes for a target to be selected when combining detections.

4 EVALUATION

4.1 Evaluation Dataset

For evaluation, we collected a novel dataset from 13 participants (three female, M=28.9 years old, SD=8.06, age range 23 to 50). Participants were seated in front of a 24" computer display to which we attached a Microsoft LifeCam Cinema Business webcam (recording resolution: 1280x720 px at 30 fps) on the top and a dedicated eye tracker, the Tobii 4C, which we use for comparison, on the bottom edge as recommended by the manufacturer. We did not constrain the participants in any way in terms of position, distance to the screen, or head movement. The only constraint was imposed by the Tobii eye tracker, which has an operating distance between 50 cm and 95 cm. When users are 75 cm away from the screen, the tracking box is 40 x 30 cm, as defined by the manufacturer.

In terms of head movement, the majority of the image samples in the normalised camera space cover vertical angles between 0° and −25° (min −40.83°, max 73.6°) and horizontal angles between −20° and 20° (min −37.13°, max 31.95°). Most vertical angles, i.e. the pitch, are negative due to the head's position relative to the camera: The camera was mounted above the screen while the participants were looking at the screen. The user's distance to the screen and camera resolution also influence the number of image pixels available in the eye region. In our dataset, the right eye, used for the optical flow computation, filled an area of 243.3 px (SD=98.8 px).

Type	No. of trials	Velocity
Linear	8	*height-of-screen*/π m/s
Linear fast	10	1.5x linear speed
Circular	7	0.8 rad/s (~45°/s)
Circular fast	4	1.1 rad/s
Orbits	11	1.1 rad/s
Alternating orbits	11	1.1 rad/s
Orbits fast	16	1.4 rad/s
Alternating orbits fast	16	1.4 rad/s
Rectangle	3	as for linear

Table 1: We study four types of trajectories with different characteristics: Linear, circular, orbits, and rectangular. The velocity for each trajectory type was set empirically after initial testing. The number of trials is per participant.

Before each session, we calibrated the eye tracker for each participant using the routine provided by the manufacturer. Following prior work on pursuits interaction [Vidal et al. 2013], a single session consisted of several experiments with different trajectory types:

(1) *Linear* trajectories moved back and forth on a straight line (tilted at a certain angle)
(2) *Circular* trajectories followed a circle, with all competitors being on the same circle
(3) *Orbital* trajectories, which also followed a circle, but competitors moved on circles with different radii
(4) *Alternating orbital* trajectories, where every other object moved in the reverse direction
(5) *Rectangular* trajectories followed a rectangle, with all competitors on the same rectangle at an equal distance apart from each other

For each type of trajectory, we used a single size, as both similarity measures we employ are invariant to scaling. However, except for rectangular trajectories, we collected data at two different velocities, which were determined in pilot experiments. The details are given in Table 1. The screen showed a single red dot following the specific trajectory that participants were asked to follow while we recorded the video from the webcam and gaze data from the Tobii eye tracker. To not distract participants while doing the study, only the actual target was shown.

For each trajectory type, we conducted different trials in which we varied the starting positions and the movement direction of the dot. We created a variable number of trajectories based on the target trajectory to simulate the presence of a certain number of competing targets in the interface. In the simulation, for any given number of competitors, we maximised the difference within the set of objects, consisting of the target and the competitors. For linear trajectories, we created a new trajectory for each competitor and rotated it around the centre point of the plane so that the angle between the trajectories was maximised. For circular and orbital trajectories, we maximised the phase shift between the objects. For example, for a circular trajectory with two competitors, there were three objects used in the evaluation, each one with a phase shift of 120°. Similarly, competitors for rectangular trajectories travel on the same shape as the target, but at a distance from each other.

4.2 Baseline Methods

To evaluate the performance and robustness of our method, we compared it to the following two baselines:

(1) Pursuits with Tobii 4C. We implemented the original *Pursuits* technique [Vidal et al. 2013] that correlates the X and Y axes of the Tobii 4C gaze estimates and object trajectories independently. If both correlation values are above a threshold, the target with the maximum correlation sum is selected. The threshold was set to 0.9.

(2) EyeFlow. We implemented the original method [Hassoumi et al. 2019] that was not designed for remote settings but head-mounted cameras. This is a fundamental difference, yet it is the closest to our work. To simulate this configuration, we cropped a fixed eye patch from the original eye image. Using the six eye landmarks, we calculated the centre, height and width of the eye patch. The cropped image was resized to 72x120 px (similar to other works [Zhang et al. 2015], where the eye patch size was 36x60 px). The location of the eye crop was calculated every ten frames to account for possible movement of the participants.

In the implementation of our method, for the gaze estimator, the correlation threshold was set to 0.6, while the cosine similarity threshold for optical flow was set to 0.8. The window size for all methods was set to 30 frames (i.e. 1 s) with a stride of one. When combining gaze estimation and optical flow, the number of necessary votes was three. In all experiments that follow, we evaluated the different methods in terms of the mean and standard deviation of the *accuracy* across participants. The accuracy is calculated as the total number of correctly detected targets divided by the total number of trials. If no target was detected during a trial, it was counted as a false detection negatively influencing the accuracy.

4.3 Pursuits Detection Performance

We first use all of the recorded experiments and run the different methods for up to 15 competitors. As explained in Subsection 4.1, this means that for each number of competitors, we simulate competitor trajectories based on the displayed target trajectory and run each of the methods on this data, i.e. each pair of specific trial and number of competitors constitutes an independent execution of the different methods. In all experiments, we discarded the first second from the video recordings to account for the possible time it took participants to follow the moving target.

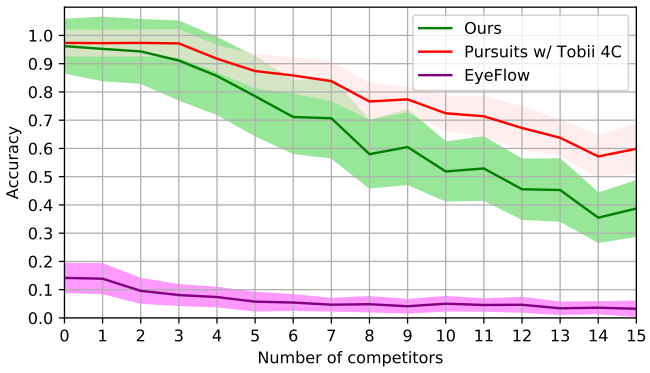

Figure 3: Pursuit detection accuracy averaged across participants. The coloured bands show the standard deviation.

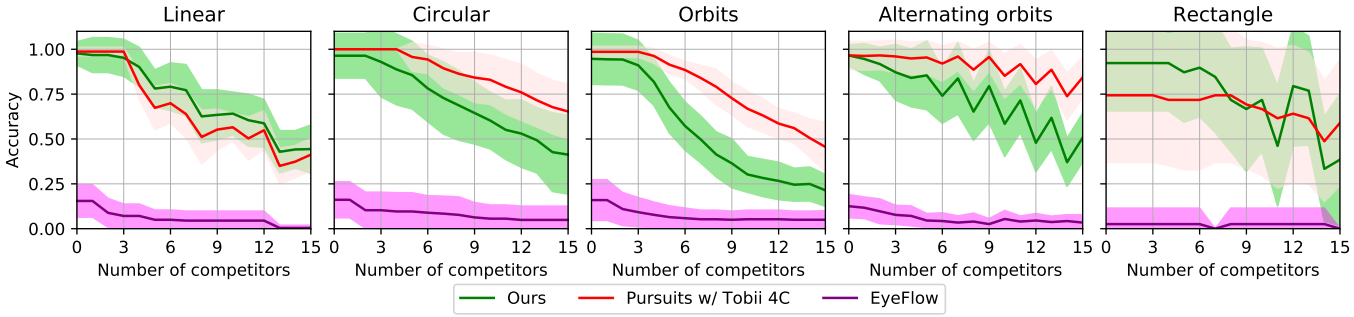

Figure 4: Mean detection accuracy per trajectory type across participants with coloured bands showing the standard deviation.

Figure 3 shows the results of this analysis for our method, *Pursuits with Tobii 4C*, and *EyeFlow*. The coloured bands depict the standard deviation across participants for each method and number of competitors. For zero competitors, we measured whether the methods were able to detect the target at all. As shown by the figure, for all methods, the accuracy decreases as the number of competitors increases. This is because the detection problem becomes more challenging with a greater number of objects to choose from. *Pursuits with Tobii 4C* achieves the highest accuracy for any number of competitors, with an accuracy of over 95% for up to three competitors, nearly linearly declining to 60% for 15 competitors. It also exhibits a lower standard deviation than our method, which also shows a high accuracy of over 90% for up to three competitors, however, then declines slightly steeper to 40% for 15 competitors. Nevertheless, our method clearly outperforms *EyeFlow*, which only achieves a very low accuracy overall. From six competitors onwards, our method shows a slight zig-zag trend, which is caused by performance differences for even and odd numbers of competitors for alternating orbital trajectories, as explained in the next section.

4.4 Influence of Trajectory Type and Speed

We further examined the influence of the type and speed of a trajectory on the accuracy. We ran the same evaluations as before, however, aggregated by trajectory type or velocity. Figure 4 shows the accuracy of the different methods per trajectory type and the coloured bands indicate the standard deviation across participants. The results for circular and orbital trajectories reflect the overall results and performance ranking among the three methods, with perfect or close to perfect detections for *Pursuits with Tobii 4C* for few competitors. When there are more than three competitors, our method outperforms *Pursuits with Tobii 4C* for linear trajectories.

Note that for alternating orbital trajectories, there are performance differences depending on whether the number of competitors is even or odd. Because we keep the phase shift consistent with the shift for circular or non-alternating orbits, the minimum phase shift between a competitor moving in the same direction as the target and the target itself is smaller for an even number of competitors than for a close odd number (e.g. Figure 5). For seven competitors, the phase shift between the target and the closest object moving in the same direction is 90°. For six competitors, however, this difference is only 51°, making discrimination more challenging although the number of competitors is smaller. Our method also outperforms *Pursuits with Tobii 4C* for rectangular trajectories with up to seven competitors; however, for both methods, the standard deviation is very high. As in the overall evaluation, *EyeFlow* achieves a poor accuracy below 20% independent of trajectory type.

For each type of trajectory, we recorded two different experiments, displaying the target at two different velocities, except for the rectangular case, which we, therefore, excluded in this specific comparison. The results depending on the velocity settings are illustrated in Figure 6. While the average performance for all three methods is similar, the standard deviation is smaller for *Pursuits with Tobii 4C* and *Ours* at higher velocity.

4.5 Ablation Study

Our method detects pursuits by jointly analysing the gaze direction and optical flow in the eye region. In this experiment, we evaluated the performance of each of the two independently and how much each component contributed to the joint result. Figure 7

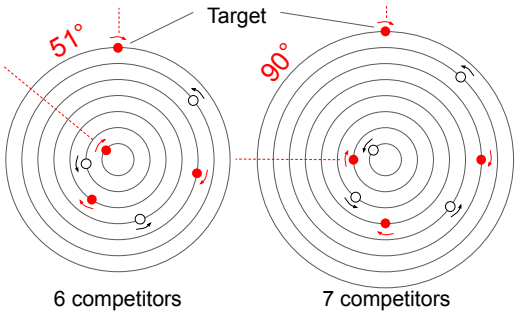

Figure 5: Alternating orbital trajectories for six and seven competitors.

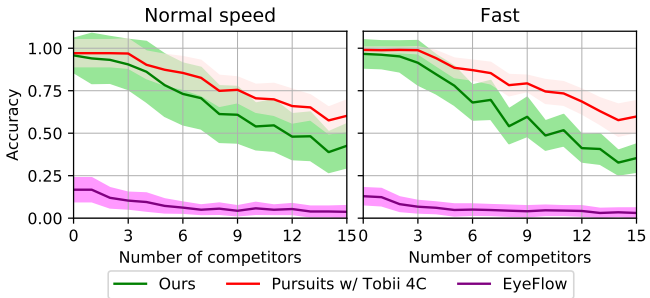

Figure 6: Pursuit detection accuracy at normal and higher velocity. We excluded the rectangular one from this evaluation since we only collected data for a single speed setting.

shows the results of the same procedure as in Subsection 4.3 for our combined method and for single components, *Gaze Estimation* and *Optical Flow*. For the single components, we employ the same voting strategy as for the combined method. When the number of competitors is low, the results for *Gaze Estimation* alone are similar to *Ours*. With a larger number of competitors, the performance of the *Gaze Estimation* component declines to the level of the *Optical Flow* results. In such cases, *Ours*, the combined method, maintains a positive performance difference of about 8 to 10 percentage points.

To investigate the contribution of each component to the overall results, we removed the voting scheme in the last processing step and simply returned the first merged detection that is not *None*. This detection is then counted towards the contribution of the component that provided it, or as a common contribution if both agreed. Figure 8 shows these contributions. We separated the contributions into correct, below the accuracy curve, and incorrect, above. The correct contributions naturally sum up to the accuracy of the combined method. Note that since we removed the voting strategy to obtain the clear origin of the detection, the results are different from those in Subsection 4.3. The results show that, overall, especially for a small number of competitors, *Gaze Estimation* has the strongest influence for both correct and incorrect detections, while for larger numbers, the contributions are balanced.

4.6 Response Time

Performance in terms of accuracy only highlights one aspect of a method's capabilities. For interactive systems, another important characteristic is the response time that can be defined as the amount of time needed until a pursuit is detected. In our evaluations, we used a window size of 30 frames, which amounts to one second for the camera we used. As such, the lower bound for the response time is 1 s, which is the time needed to fill the first window.

We calculated the response time for each trial and then average across all trials. When the number of competitors is between 0 and 7, the average response time was 1.44 s (SD=0.01 s) for *Tobii*, 1.45 s (SD=0.06 s) for *Gaze Estimation*, 1.94 s (SD=0.07 s) for *Ours*, and 2.34 s (SD=0.08 s) for *Optical Flow*. For 8 to 15 competitors, the mean response time was 1.43 s (SD=0.01 s) for *Tobii*, 1.36 s (SD=0.01 s) for *Gaze Estimation*, 2.27 s (SD=0.11 s) for *Ours*, and 2.22 s (SD=0.01 s) for *Optical Flow*. We did not analyse the response time for *EyeFlow* since this method has a very low overall accuracy when used in remote settings, which makes it practically unusable.

4.7 Runtime Analysis

We evaluated the runtime of our pipeline on a desktop PC equipped with an Intel i7-4790K CPU @ 4.00GHz and an Nvidia GeForce 1080 ti GPU. Face detection and landmark localisation use state-of-the-art neural networks [Deng et al. 2018; Zhang et al. 2016], which require a GPU. Face detection takes about 70 ms for an image of 1280x720 px (like the ones used in our evaluations). Reducing the resolution to 640x360 px decreases the face detection runtime to about 20 ms per image, but we did not investigate the effect of lower resolution images on performance. Landmark localisation including face image normalisation takes around 570 ms for a window of 30 images (~19 ms per frame). The gaze estimation CNN needs around 50 ms for a batch of 30 images (~1.7 ms per frame). Optical flow

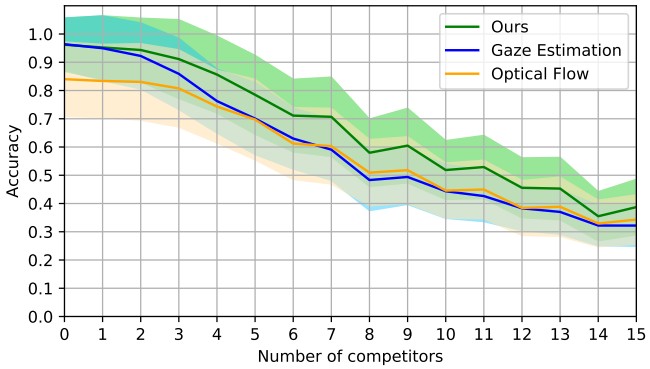

Figure 7: Ablation study comparing the pursuit detection performance of *Ours* to the individual components of our method, *Gaze Estimation* and *Optical Flow*.

estimation takes around 45 ms, motion compensation around 38 ms and, because of the *compute-rate*=5, they are calculated six times in a window of 30 frames. The two components of our method can run in parallel, gaze estimation on the GPU and optical flow on the CPU. In our current implementation, the runtime is bounded by the gaze estimation pipeline, which takes around 93 ms per image.

5 DISCUSSION

In this work, we proposed to combine appearance-based gaze estimation and optical flow in the eye region to jointly detect pursuits without the need for any special-purpose eye tracking equipment. Our method only requires a single RGB camera, which is included in an ever-increasing number of devices [Khamis et al. 2018a].

Evaluations on a novel real-world dataset showed that our method could robustly detect the correct pursuit target with over 90% accuracy for up to four moving UI elements, independent of trajectory type (Figure 3). It not only outperforms the current state of the art EyeFlow [Hassoumi et al. 2019] but is also competitive to a commercial, consumer-grade eye tracker. EyeFlow shows a low overall accuracy, and this could be, in part, because of the optical flow calculation: EyeFlow was proposed for head-mounted settings where the camera has a close-up, high-resolution view of the eye. In remote settings, users are at a distance, and only a few pixels are available in the eye region to calculate the flow (in our case, around 240 px per eye). Therefore, optical flow calculations across two consecutive frames will lead to increased noise and incorrect estimates of eye movement direction. By calculating the dense optical flow every five frames, our method is able to better approximate the eye motion since the change between two images will be more significant. For a large number of moving UI elements, all methods suffer from a drop in performance, including the one that uses a dedicated eye tracker. Other research-grade eye trackers such as the Tobii Pro Spectrum may achieve better performance. However, they are not only expensive but also not targeted towards end users.

In all our evaluations, to ensure reliable pursuit samples, we only showed participants a single target that they had to follow (similar to [Vidal et al. 2013]) and generated all the competitors post hoc. In a real application, multiple moving elements might be present at the same time, potentially distracting users. However, in a multi-target environment, neuroscience literature suggests

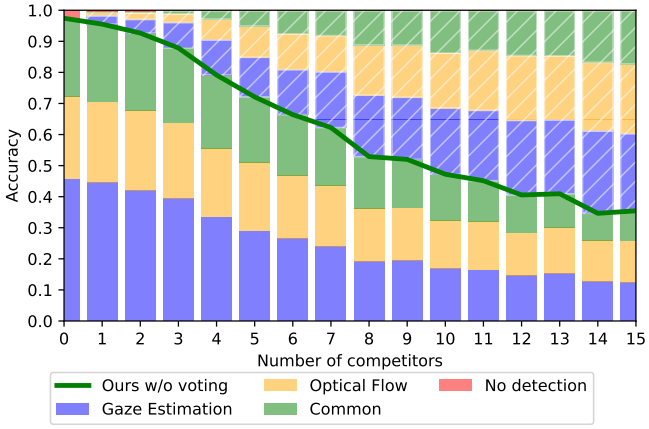

Figure 8: Ablation study quantifying the contribution of each component, i.e. *Gaze Estimation*, *Optical Flow*, or *Common*, towards the final decision. The bars below the accuracy curve, which is without the voting strategy, indicate correct contributions. The ones above show incorrect contributions, including a segment when nothing was detected.

that the brain suppresses other non-tracked targets [Leigh and Zee 2015] and the perception of competitors is reduced during smooth pursuit [Khurana and Kowler 1987]. Moreover, a field study in a real environment has shown that users can reliably select the desired target in spite of multiple visible competitors [Vidal et al. 2013].

An in-depth analysis showed that there are also differences between the methods when looking at different trajectory types. As expected, when the number of moving targets is low, the *Pursuits with Tobii 4C* method performs best, yet ours follows closely. More interesting is that for linear trajectories, when the number of targets increases, our method shows increased robustness and even outperforms *Pursuits with Tobii 4C* (Figure 4). This could be explained by the use of the cosine similarity in the optical flow component. It appears that our method also outperforms the dedicated eye tracker for rectangular trajectories. However, given the little number of trials per participant, it is difficult to draw general conclusions. In our analysis, we also evaluated the influence of the target velocity on all the methods' performance (Figure 6). The overall average accuracy is similar, yet the standard deviation is smaller, which implies that faster targets lead to more stable results across participants.

To further understand how and which of the two components contribute to the final results, we did an ablation study (Figure 7 and Figure 8). Of the two components, *Gaze estimation* has a higher overall accuracy than *Optical Flow*. However, when the number of targets is five or more, the methods perform quite similarly. Nevertheless, Figure 7 clearly shows that by combining both of them, the accuracy can improve by as much as 10 percentage points.

We also compared the different methods in terms of response time that is of particular practical relevance in interactive systems. *Pursuits with Tobii 4C* is, as expected, the fastest of all the methods with around 1.4 s. Our method is between *Gaze Estimation* and *Optical Flow* with a response time between 1.9 and 2.2 s. This represents a trade-off between accuracy and response time. Only using gaze estimates will lead to a faster, but not as accurate decision.

Based on the findings from our evaluations, given the high accuracy for up to four moving UI elements, our method could be applied in several practical applications. For example, similar to Orbits [Esteves et al. 2015], a music player could have pursuit-enabled controls. On a public display, a few moving targets could show tourists interesting facts about a city. Another, less obvious application could be eye tracker calibration. Similarly to how pursuits can be used to calibrate an eye tracker [Pfeuffer et al. 2013], our method could be used to collect implicit calibration samples for personalised appearance-based gaze estimation [Zhang et al. 2018a].

While our work is the first to propose a method to detect pursuits in unconstrained remote settings with a single RGB camera, it also has several limitations that we will address in future work. First, the application design space is limited by the number of targets that can be detected reliably. As such, our method is robust for up to four targets of the same kind. It would be interesting to investigate not only combinations of different trajectory types but finding optimal combinations that would maximise the overall performance for a given number of targets. Besides predefined trajectories, we can imagine users creating personalised trajectories, e.g. by drawing them by hand on a smartphone or tablet, thereby designing their own interfaces. Second, the performance may be influenced by users' blinking. Our recordings naturally also contain such events and, while we did not investigate the effect of blinks, our results implicitly include them. Blinks may lead to incorrect gaze estimates or optical flow vectors, which, if filtered out, may increase performance. Catch-up saccades, which correct the eye's position relative to the target, are another factor to consider when designing interactions with pursuits. This is a consequence of targets moving too fast (e.g. 120°/s), which then leads to a drop in performance [Esteves et al. 2015]. Third, the fact that the participants knew they were taking part in a study might have altered their behaviour, e.g. in terms of head movement. A follow-up study could examine performance in a real-world deployment, such as playing a game on a public display [Vidal et al. 2013]. Lastly, we also intend to optimise the runtime in order to obtain a real-time system. Face detection, which dominates the runtime, could be replaced with a fast object tracker such as KCF [Henriques et al. 2015].

6 CONCLUSION

In this work, we proposed a novel method to detect pursuits by combining appearance-based gaze estimation and optical flow to jointly analyse the eye movement dynamics. Our method only requires images captured with a single off-the-shelf camera placed at a distance from the user. Through in-depth evaluations on the data collected from a 13-participant user study, our method shows a significant performance increase in comparison to the current state of the art. Moreover, for up to four moving UI elements, our method achieves an average accuracy of over 90%, which is competitive with the performance of a dedicated eye tracker. Taken together, these results are significant because they, for the first time, point towards a new class of methods that enable pursuit interactions with nothing more than a standard off-the-shelf RGB camera.

ACKNOWLEDGEMENTS

A. Bulling was supported by the European Research Council (ERC; grant agreement 801708). We thank the study participants for their contribution and Alexander Kayed for his help with data collection.

REFERENCES

Hassoumi Almoctar, Pourang Irani, Vsevolod Peysakhovich, and Christophe Hurter. 2018. Path Word: A Multimodal Password Entry Method for Ad-hoc Authentication Based on Digits' Shape and Smooth Pursuit Eye Movements. In *Proceedings of the International Conference on Multimodal Interaction (ICMI '18)*. ACM, New York, NY, USA, 268–277. https://doi.org/10.1145/3242969.3243008

Mihai Bâce, Teemu Leppänen, David Gil de Gomez, and Argenis Ramirez Gomez. 2016. ubiGaze: Ubiquitous Augmented Reality Messaging Using Gaze Gestures. In *SIGGRAPH ASIA Mobile Graphics and Interactive Applications (SA '16)*. ACM, New York, NY, USA, Article 11, 5 pages. https://doi.org/10.1145/2999508.2999530

Mihai Bâce, Sander Staal, and Andreas Bulling. 2019. Accurate and Robust Eye Contact Detection During Everyday Mobile Device Interactions. *CoRR* abs/1907.11115 (2019), 12. arXiv:1907.11115 http://arxiv.org/abs/1907.11115

Roman Bednarik, Tersia Gowases, and Markku Tukiainen. 2009. Gaze interaction enhances problem solving: Effects of dwell-time based, gaze-augmented, and mouse interaction on problem-solving strategies and user experience. *Journal of Eye Movement Research* 3, 1 (Aug. 2009), 10. https://doi.org/10.16910/jemr.3.1.3

Marcus Carter, Eduardo Velloso, John Downs, Abigail Sellen, Kenton O'Hara, and Frank Vetere. 2016. PathSync: Multi-User Gestural Interaction with Touchless Rhythmic Path Mimicry. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3415–3427. https://doi.org/10.1145/2858036.2858284

Feridun M. Celebi, Elizabeth S. Kim, Quan Wang, Carla A. Wall, and Frederick Shic. 2014. A Smooth Pursuit Calibration Technique. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '14)*. ACM, New York, NY, USA, 377–378. https://doi.org/10.1145/2578153.2583042

Christopher Clarke, Alessio Bellino, Augusto Esteves, Eduardo Velloso, and Hans Gellersen. 2016. TraceMatch: A Computer Vision Technique for User Input by Tracing of Animated Controls. In *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 298–303. https://doi.org/10.1145/2971648.2971714

Dietlind Helene Cymek, Antje Christine Venjakob, Stefan Ruff, Otto Hans-Martin Lutz, Simon Hofmann, and Matthias Roetting. 2014. Entering PIN codes by smooth pursuit eye movements. *Journal of Eye Movement Research* 7, 4 (Sep. 2014), 11. https://doi.org/10.16910/jemr.7.4.1

Jiankang Deng, Yuxiang Zhou, Shiyang Cheng, and Stefanos Zaferiou. 2018. Cascade Multi-View Hourglass Model for Robust 3D Face Alignment. In *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition (FG '18)*. IEEE, 399–403. https://doi.org/10.1109/FG.2018.00064

Heiko Drewes, Mohamed Khamis, and Florian Alt. 2019. DialPlates: Enabling Pursuits-based User Interfaces with Large Target Numbers. In *Proceedings of the International Conference on Mobile and Ubiquitous Multimedia (MUM '19)*. ACM, New York, NY, USA, Article 10, 10 pages. https://doi.org/10.1145/3365610.3365626

Heiko Drewes and Albrecht Schmidt. 2007. Interacting with the Computer Using Gaze Gestures. In *Proceedings of the IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II (INTERACT '07)*. Springer-Verlag, Berlin, Heidelberg, 475–488. http://dl.acm.org/citation.cfm?id=1778331.1778385

Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: Gaze Interaction for Smart Watches Using Smooth Pursuit Eye Movements. In *Proceedings of the Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 457–466. https://doi.org/10.1145/2807442.2807499

Gunnar Farnebäck. 2003. Two-Frame Motion Estimation Based on Polynomial Expansion. In *Proceedings of the Scandinavian Conference on Image Analysis*, Vol. 2749. Springer, Berlin, Heidelberg, 363–370. https://doi.org/10.1007/3-540-45103-X_50

Almoctar Hassoumi, Vsevolod Peysakhovich, and Christophe Hurter. 2019. EyeFlow: Pursuit Interactions Using an Unmodified Camera. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '19)*. ACM, New York, NY, USA, Article 3, 10 pages. https://doi.org/10.1145/3314111.3319820

Joao F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. 2015. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 3 (Mar 2015), 583–596. https://doi.org/10.1109/tpami.2014.2345390

Shuai Hua, Manika Kapoor, and David C. Anastasiu. 2018. Vehicle Tracking and Speed Estimation from Traffic Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '18)*. IEEE, 153–1537. https://doi.org/10.1109/CVPRW.2018.00028

Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. IEEE, 9. http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17

Mihir Jain, Hervé Jégou, and Patrick Bouthemy. 2013. Better Exploiting Motion for Better Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*. IEEE, 2555–2562. https://doi.org/10.1109/CVPR.2013.330

Mohamed Khamis, Florian Alt, and Andreas Bulling. 2015. A Field Study on Spontaneous Gaze-based Interaction with a Public Display Using Pursuits. In *Adjunct Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the International Symposium on Wearable Computers (UbiComp/ISWC '15 Adjunct)*. ACM, New York, NY, USA, 863–872. https://doi.org/10.1145/2800835.2804335

Mohamed Khamis, Florian Alt, and Andreas Bulling. 2018a. The Past, Present, and Future of Gaze-enabled Handheld Mobile Devices: Survey and Lessons Learned. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18)*. ACM, New York, NY, USA, 38:1–38:17. https://doi.org/10.1145/3229434.3229452

Mohamed Khamis, Carl Oechsner, Florian Alt, and Andreas Bulling. 2018b. VRpursuits: Interaction in Virtual Reality Using Smooth Pursuit Eye Movements. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI '18)*. ACM, New York, NY, USA, Article 18, 8 pages. https://doi.org/10.1145/3206505.3206522

Mohamed Khamis, Ozan Saltuk, Alina Hang, Katharina Stolz, Andreas Bulling, and Florian Alt. 2016a. TextPursuits: Using Text for Pursuits-Based Interaction and Calibration on Public Displays. In *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp '16)*. ACM, New York, NY, USA, 274–285. https://doi.org/10.1145/2971648.2971679

Mohamed Khamis, Ozan Saltuk, Alina Hang, Katharina Stolz, Andreas Bulling, and Florian Alt. 2016b. TextPursuits: Using Text for Pursuits-based Interaction and Calibration on Public Displays. In *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 274–285. https://doi.org/10.1145/2971648.2971679

Beena Khurana and Eileen Kowler. 1987. Shared attentional control of smooth eye movement and perception. *Vision Research* 27, 9 (1987), 1603–1618. https://doi.org/10.1016/0042-6989(87)90168-4

Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye Tracking for Everyone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*. IEEE, 2176–2184. https://doi.org/10.1109/CVPR.2016.239

R. John Leigh and David S. Zee. 2015. *The Neurology of Eye Movements*. Oxford University Press, Oxford, UK, Chapter 5, 299. https://oxfordmedicine.com/view/10.1093/med/9780199969289.001.0001/med-9780199969289

Dongheng Li, David Winfield, and Derrick J. Parkhurst. 2005. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '05)*. IEEE, 79–79. https://doi.org/10.1109/CVPR.2005.531

Bruce D. Lucas and Takeo Kanade. 1981. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the International Joint Conference on Artificial Intelligence - Volume 2 (IJCAI '81)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 674–679. http://dl.acm.org/citation.cfm?id=1623264.1623280

Thomas Mattusch, Mahsa Mirzamohammad, Mohamed Khamis, Andreas Bulling, and Florian Alt. 2018. Hidden Pursuits: Evaluating Gaze-selection via Pursuits when the Stimuli's Trajectory is Partially Hidden. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '18)*. ACM, New York, NY, USA, Article 27, 5 pages. https://doi.org/10.1145/3204493.3204569

Carlos H. Morimoto, Dave Koons, Arnon Amir, and Myron Flickner. 2000. Pupil detection and tracking using multiple light sources. *Image and Vision Computing* 18, 4 (2000), 331–335. https://doi.org/10.1016/S0262-8856(99)00053-0

Ken Pfeuffer, Melodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2013. Pursuit Calibration: Making Gaze Calibration Less Tedious and More Flexible. In *Proceedings of the Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 261–270. https://doi.org/10.1145/2501988.2501998

Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J. Black. 2016. Optical Flow with Semantic Segmentation and Localized Layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*. IEEE, 3889–3898.

Linda E. Sibert and Robert J. K. Jacob. 2000. Evaluation of Eye Gaze Interaction. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '00)*. ACM, New York, NY, USA, 281–288. https://doi.org/10.1145/332040.332445

Karen Simonyan and Andrew Zisserman. 2014. Two-stream Convolutional Networks for Action Recognition in Videos. In *Proceedings of the International Conference on Neural Information Processing Systems - Volume 1 (NIPS '14)*. MIT Press, Cambridge, MA, USA, 568–576. http://dl.acm.org/citation.cfm?id=2968826.2968890

Eduardo Velloso, Flavio Luiz Coutinho, Andrew Kurauchi, and Carlos H Morimoto. 2018. Circular Orbits Detection for Gaze Interaction Using 2D Correlation and Profile Matching Algorithms. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '18)*. ACM, New York, NY, USA, Article 25, 9 pages. https://doi.org/10.1145/3204493.3204524

Eduardo Velloso, Markus Wirth, Christian Weichel, Augusto Esteves, and Hans Gellersen. 2016. AmbiGaze: Direct Control of Ambient Devices by Gaze. In *Proceedings of the Conference on Designing Interactive Systems (DIS '16)*. ACM, New York, NY, USA, 812–817. https://doi.org/10.1145/2901790.2901867

Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2013. Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets. In *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 439–448. https://doi.org/10.1145/2493432.2493477

Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters* 23 (04 2016). https://doi.org/10.1109/LSP.2016.2603342

Xucong Zhang, Michael Xuelin Huang, Yusuke Sugano, and Andreas Bulling. 2018a. Training Person-Specific Gaze Estimators from User Interactions with Multiple Devices. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 624, 12 pages. https://doi.org/10.1145/3173574.3174198

Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 2018b. Revisiting Data Normalization for Appearance-Based Gaze Estimation. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '18)*. ACM, New York, NY, USA, Article 12, 9 pages. https://doi.org/10.1145/3204493.3204548

Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 2019a. Evaluation of Appearance-Based Methods and Implications for Gaze-Based Applications. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 416, 13 pages. https://doi.org/10.1145/3290605.3300646

Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2015. Appearance-based Gaze Estimation in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*. IEEE, 4511–4520. https://doi.org/10.1109/CVPR.2015.7299081

Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. It's Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '17)*. IEEE, 2299–2308. https://doi.org/10.1109/CVPRW.2017.284

Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2019b. MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 41, 1 (2019), 162–175. https://doi.org/10.1109/TPAMI.2017.2778103