Eye Gaze Estimation and its Applications

Xucong Zhang[1], Seonwook Park[1] and Anna Maria Feit[2]

Abstract The human eye gaze is an important non-verbal cue that can unobtrusively provide information about the intention and attention of a user to enable intelligent interactive systems. Eye gaze can also be taken as input to systems as a replacement of the conventional mouse and keyboard, and can also be indicative of the cognitive state of the user. However, estimating and applying gaze in real-world applications poses significant challenges. In this chapter, we first review the development of gaze estimation methods in recent years. We especially focus on learning-based gaze estimation methods which benefit from large-scale data and deep learning methods that recently became available. Second, we discuss the challenges of using gaze estimation for real-world applications and our efforts towards making these methods easily usable for the Human-Computer Interaction community. At last, we provide two application examples, demonstrating the use of eye gaze to enable attentive and adaptive interfaces.

1 Introduction

The human eye has the potential to serve as a fast, pervasive, and unobtrusive way to interact with the computer. Reliably detecting where a user is gazing at allows the eyes to be used as an explicit input method. Such a new way of interaction has been shown to outperform traditional input devices such as the mouse due to the ballistic movement of eye gaze [42, 52]. Moreover, it allows interaction under circumstances where no external input device is available or operable by the user [27]. Beyond explicit input, the movement patterns of a user's eyes reveal information about the cognitive processes, level of attention, and interests or abilities [6]. This offers

[1]ETH Zürich, Department of Computer Science, Stampfenbachstrasse 48, 8092 Zürich, Switzerland. e-mail: `firstname.lastname@inf.ethz.ch`
[2]Saarland University, Saarland Informatics Campus Saarbrücken, Germany. e-mail: `feit@cs.uni-saarland.de`

exciting opportunities to develop novel intelligent and interactive systems that truly understand the user.

In this chapter, we focus on remote camera-based gaze estimation and its applications. This is typically done in a setting such as that depicted in Figure 1 where a camera is positioned at a certain distance from and facing the user's eyes. The problem these methods aim to solve is to infer the 3D gaze direction or the 2D on-screen point-of-gaze (PoG) from images recorded by the camera. The 3D gaze origin is often defined to be at the center of the eye or face. Note that these gaze estimation approaches can also be adapted to head-mounted devices such as those used in AR and VR settings, though we do not discuss them in this chapter [23].

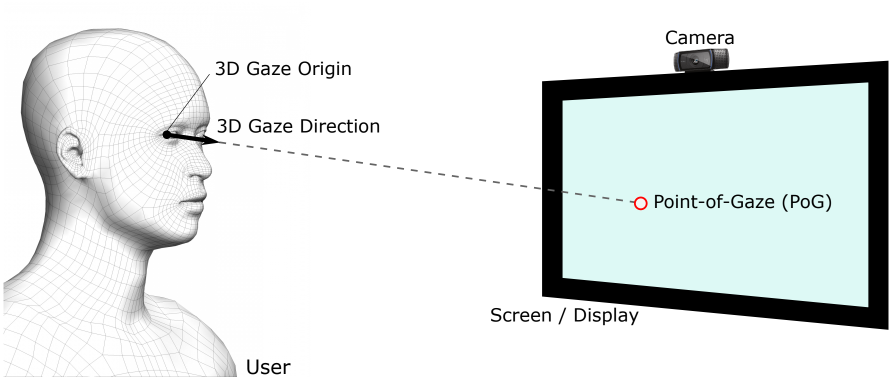


Fig. 1: The standard setting for remote camera-based gaze estimation. A camera captures images of the user's face region. The problem of gaze tracking is to infer the 3D gaze direction in the camera coordinate system or the 2D point-of-gaze (PoG) on the screen from the image recorded by the camera.

Estimating the gaze position of a user is a challenging task. Subtle movements of the eyeball can change the gaze direction dramatically and the difficulty of the task varies greatly across people. Reliably determining where a user is looking on a screen or inside a room has been an active research topic for several decades. Classic gaze estimation methods often use high-resolution machine vision cameras and corneal reflections from infra-red illuminators to determine the gaze direction [16]. These methods can provide reasonable gaze estimation accuracy of around one degree after personal calibration in well-controlled environments. However, dedicated hardware is essential for their performance, which limits their use in real-world applications.

The rise of AI methods, such as deep learning approaches, has advanced the use of *learning-based gaze estimation methods*. In contrast to classic methods, learning-based methods are based on purposefully designed machine learning models, for example neural networks, for the gaze estimation task. These learning-based methods either estimate the gaze position directly from an image of the user's eye or face [56, 24, 61], or derive intermediate eye features for gaze direction regression [33, 47]. This group of methods often assume an unmodified environment. That is, no additional infra-red illumination is available to provide reflections on the surface of the cornea. Hence, learning-based gaze estimation methods can work with a single off-the-shelf webcam [56, 55, 33, 34, 36]. This makes these approaches more widely and more easily applicable for human-computer interaction (HCI) in everyday settings [57, 54].

Still, many challenges persist in making gaze tracking practicable for computer interaction. For example, personal calibration plays a major role in gaze estimation and also has an impact on user experience. The calibration procedure often requires users to focus on designated points for a period of time. This can be cumbersome or in some cases even impossible and disturbs the user experience. Nevertheless, personal calibration is crucial for many gaze estimation methods to perform accurately. Thus, recently researchers have built on AI-based techniques for gaze redirection to generate additional eye images for personalization and thus reduce the number of calibration samples [51]. Other researchers have worked on providing easy-to-use software toolkits for making learning-based gaze estimation methods accessible to HCI researchers and developers [59].

Designing useful and usable gaze-aware interfaces is another major challenge. In practice, tracking accuracy and precision vary largely depending on factors such as the tracking environment, user characteristics, and others [7]. In comparison to mouse or touch input, eye tracking might yield a highly noisy signal with poor accuracy. Still, information about eye gaze, even from noisy data, can enable novel and useful interactions. However, design guidelines developed for traditional interfaces cannot be applied here. Instead, we need new design approaches making efficient use of the noisy gaze signal.

In this chapter, we first provide some background on the problem of gaze tracking. We then offer an overview of recent approaches towards improving performance on the gaze estimation task with the power of AI. We then discuss the practical challenges when applying gaze estimation methods for computer interaction and designing gaze-aware interfaces, offering concrete design guidelines and actionable insights for the HCI community. Finally, we describe two application examples: (1) gaze-aware interaction with real-life objects and (2) automatic interface adaptation by assessing information relevance from users' eye movements. These examples showcase the exciting opportunities gaze tracking offers for Human-Computer Interaction.

2 Background

In the following, we start with a brief introduction to the human eye, its movements and the relation to human attention. We then discuss different categories of gaze estimation methods and introduce learning-based methods. Lastly, we briefly discuss the need for the personal calibration of gaze estimators and how this has been done in existing works.

2.1 The Human Eye Gaze

The human visual field is about 114° [20] large of which we can only see sharply in an area of 1° [2] during so-called *fixations*, when the gaze is focused on a fixed position in the environment. To perceive information from a larger area, the eyes perform *saccades*, fast ballistic movements that allow us to move between fixation points to integrate information from other areas. See for example [40] for further introduction into the working principles of the human eye gaze. The duration and frequency of such fixations and saccades can provide information about a user's attention. It can be used by interactive systems in combination with their awareness of the visual stimulus or interface to enable explicit gaze input or make further inferences about a user's cognitive state.

However, a person does not always consciously control their eye gaze. Often, it is stimulus-driven and attracted by visual features, or "idles" in uninteresting regions. Thus, there is a difference between the eyes focusing on a point and a person's covert attention (i.e. their *mental* focus). Even when focusing a certain point, people can shift their conscious attention within the larger field of view similar to a spotlight and to some extent independent of the gaze position. This allows them to not just passively perceive information but visually process and encode it for further cognitive processing [39]. A major challenge for using gaze for HCI is to isolate and analyze the underlying cognitive processes from such noisy gaze behavior where overt and covert attention are mixed. In the later part of this chapter we describe some applications that aim to make sense of noisy gaze behavior [7, 54].

2.2 Gaze estimation methods

The gaze estimation methods considered here try to infer information about where a person is looking at from an image of the users eyes or face image. They can be categorized into three groups: model-based, feature-based and appearance-based methods [16]. In both model- and feature-based methods, key landmarks are often required to be detected, such as the pupil center, eye corner and iris contour. Generally speaking, model-based methods fit a pre-defined 3D eyeball model to the detected eye landmarks and take the direction from eyeball center to the pupil center as the gaze direction [48, 49]. The eyeball model can optionally incorporate an offset parameter which can be determined with personal calibration data [46]. Feature-based methods take eye-region landmarks as features for the direct regression of gaze direction [41]. Since the input feature dimension is limited by the number of determined key points, these methods often cannot handle complex changes such as large head movements. Both model-based and feature-based methods conventionally demand accurate eye landmark detection, often necessitating complex or expensive hardware setups. For example, multiple high-resolution infrared-light cameras along with optimal infrared-light light sources are the standard hardware configuration for most of these methods. Appearance-based methods directly learn the mapping from the eye or face image to the gaze direction [44]. Since there is no need for explicit eye landmarks detection (and corresponding training data annotation in the real-world), appearance-based methods can work with a single webcam without any additional light source. However, these methods can be sensitive to illumination condition changes or unfamiliar facial appearances due to the scarcity of training data.

2.3 Learning-based gaze estimation methods

Recent developments in deep learning have given rise to a large array of promising learning-based gaze estimation methods. We refer to these methods as being *learning-based*, in order to encompass hybrid methods [34, 50] as well as appearance-based methods that benefit from large amounts of training data and highly complex neural network architectures [53, 24]. In particular, appearance-based gaze estimation methods work with just a single webcam under challenging lighting conditions even over long operating distances of up to 2 meters [59, 10]. This is because deep convolutional neural networks - when given large and varied amounts of training data - are effective at defining useful image-based features, and thus often outperform hand-defined features. Importantly, this allows for the new task of person-independent gaze estimation. That is, a generic learning-based gaze estimation model can be directly applied to a previously unseen user and achieve 4° to 6° of mean angular error even in very challenging conditions.

Integrating known priors such as the 3D structure of the eyeball or eyelids into neural networks is a promising direction of research. A hierarchical generative model has been proposed for improving gaze estimation by understanding how to control and generate eye shape [47]. A so-called *gazemaps* representation has been used to implicitly encode a 3D eyeball model and then taken as an intermediate output for gaze direction regression [33]. Applying deep learning based landmark localization architectures for eye-region landmarks detection has also been shown to be more effective than traditional edge or contour-based methods [34, 11].

2.4 Person-specific gaze estimator calibration

While learning-based methods perform well in the person-independent setting, the error of 4° to 6° may be unsuitable for applications that require higher accuracy. When sufficient data is provided from the target user, such methods were shown to perform at an average gaze estimation error of 2.5° in-the-wild [56]. Reducing this performance gap increases the efficacy and applicability of learning-based methods greatly. In this section, we describe why this performance gap exists and discuss how recent learning-based methods reduce it.

A primary reason for this performance gap is the so-called “angle kappa” as the angular difference between the line-of-sight of a user (actual axis along which an eye “sees”) and the optical axis of their eyeball (defined by the geometry of the head and eye). For a more principled definition of angle-kappa, please refer to [29]. This difference varies greatly across people with typical differences being two to three degrees [29]. Importantly, the line-of-sight cannot be measured by a camera alone as it is defined by the position of the fovea, which cannot be observed. The optical axis, on the other hand, can be reasonably estimated from the appearance of the eye-region.

The classic literature tackles this issue by explicitly defining the kappa angle as a parameter to an optimization problem. In all gaze estimator calibration methods, a user is asked to gaze at specified points on a screen or in space. An optimization-based scheme is then often applied to determine the user-specific parameters of the model. An important consideration in these schemes is in requiring minimal “calibration samples” from the end-user such as to make the experience less cumbersome and also to enable spontaneously interactive applications in everyday scenarios. Conventional approaches are quite effective in clean and controlled laboratory settings where the position and shape of the iris and eyeball can be reasonably measured. In-the-wild settings and unconstrained head movements of the user, however, pose significant challenges that learning-based methods can easily address. However, learned models can be tricky to adapt as user-specific parameters are usually not explicitly defined.

Several feasible calibration strategies have been suggested recently for learning-based gaze estimation, either via optimization of user-specific parameters defined at specific parts of the network, or via eye-region image synthesis for personalized training data generation. The more direct and effective approaches define parameters which can be adapted based on a few labeled samples from the target user. Approaches have been proposed to apply these parameters at the input [17, 25] and output [4] of the neural network. As the primary factor in the difference between users is the angle-kappa, such low-dimensional definitions of user parameters are surprisingly effective. Yet other approaches have been proposed for learning a light-weight regression model from penultimate layer activations [34, 24], or gradient-based meta-learning as a method for effective few-shot neural network adaptation [35]. A unique approach suggests correcting an initially estimated gaze direction based on changes in the appearance of the presented visual stimuli [36]. Importantly, this approach does not require any explicit calibration but instead relies on the model having been trained on paired eye gaze and visual stimulus data.

An alternative area of research is in “gaze redirection”, where the objective is in accurate and high-quality eye image generation with control of gaze direction. While earlier learning-based methods in this area focused only on the image synthesis aspect [13], later works have shown that generating person-specific eye images with varying gaze directions can allow for an alternative method of personal calibration. That is, given a few samples from the target user, gaze redirection methods can be used to create a training dataset tailored to the target user [51]. Though not directly related to personal calibration, later works [18, 62] have further improved the accuracy and quality of generated images and have shown that limited gaze datasets can be augmented via such synthesis schemes.

3 Learning-based Gaze Estimation Methods

We refer to “learning-based gaze estimation methods” as the set of methods that take advantage of modern machine learning techniques for the gaze estimation task. Nowadays, these are mostly enabled by deep learning techniques together with a large amount of training data. The input to these methods are monocular images of the eye or face region, and the models either directly estimate eye gaze, extract intermediate features for the gaze task, or otherwise work towards the improvement of gaze estimation performance via approaches such as data synthesis. Such strategies have been advancing rapidly with the recent development of convolutional neural networks. Alongside, several datasets have been introduced covering an increasingly wider variety of human appearances and temporal information to improve generalization and provide novel challenges to existing data-driven models.

3.1 Gaze estimation method pipeline

The gaze estimation method we proposed in [53] was the first work to use a convolutional neural network architecture for gaze estimation. Our later works extended the architecture to much deeper networks such as VGG-16 and ResNet-50 [56, 60]. These works introduce a basic pipeline for image-based gaze estimation. That is, given an input image taken from a single webcam, we learn a direct mapping to the gaze direction (see Figure 2). The first step in this pipeline is face detection and facial landmarks localization. Then, we fit a pre-defined 3D face model to the detected facial landmarks to estimate the rotation and translation of the head. With this head pose information, we perform a procedure known as “data normalization” to cancel the rotation around the roll-axis and crop the eye or face image to a consistently defined size [43]. This data normalization procedure was later optimized further to increase its effectiveness towards improving gaze estimation performance [58]. Finally, the cropped image, together with the head pose, are fed into the convolution neural networks to regress to the final gaze direction in the camera coordinate system. The gaze direction can be presented as a three-dimension vector in the Cartesian coordinate system. We choose to convert it to a two-dimensional vector in the spherical coordinate system representing the polar angle and azimuthal angle. In this way, we reduce one degree of freedom from the gaze direction vector and center the output values around zero, for better ease of regression.

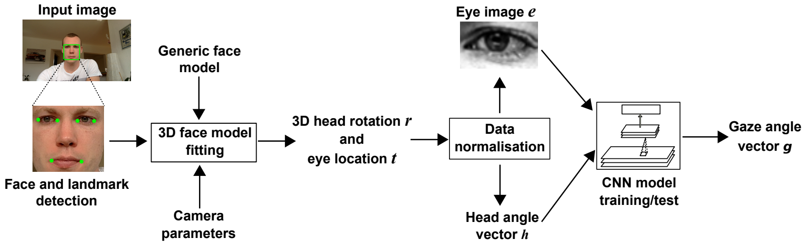


Fig. 2: The gaze estimation pipeline proposed in [56] describes a pre-processing procedure for extracting eye patches which are then input to a convolutional neural network that directly predicts the gaze direction of the imaged eye.

3.2 3D and 2D gaze estimation

The final output of the pipeline shown in Figure 2 is a 3D gaze direction value. Alternatively, a different group of methods directly output the 2D gaze location on a screen that the user is assumed to be gazing at. See [55] for an example of experiments directly comparing between 2D and 3D gaze estimation. Apart from a change in the dimensionality of the output value, 3D and 2D gaze estimation differ in practice in terms of how the head position is integrated into the estimation pipeline. 3D gaze estimation methods typically insert the 3D head orientation value (often referred to as head pose) directly into the network as input to one of the last fully-connected layers. The task of 2D screen-space point-of-regard regression (2D gaze estimation) however theoretically requires more complex information such as the definition of the pose, scale, and bounds of the screen plane as well as a reliable estimation of the translation of the head in relation to the screen. This can be approximated by providing a binary "face grid" where the number of black pixels (as opposed to white pixels) indicate the size and position of the user's face [24]. While this alternative 2D problem formulation tackles the gaze estimation task more directly, its main drawback is that the trained model is specific to the device used in the training data. Hence, 2D models are not robust to changes such as the camera hardware, screen size and pose, and other factors pertaining to the camera-screen relationship. 3D gaze direction estimation is thus a more generic approach that can consolidate data samples from different devices both at training time and test time.

3.3 Input for gaze estimation methods

Early works in gaze estimation only take a single eye image as input since it is often deemed to be sufficient in inferring gaze direction [43, 53]. However, learning-based methods can be surprisingly effective in extracting information from seemingly redundant image regions and thus regions beyond the single eye could be helpful for training neural networks. Taking a face image along with both eye images [24] or simply the two eye patches [10, 5] as input for gaze estimation can achieve better performance than a model taking single eye input. We were the first to use a single full-face image as input for the gaze estimation task [55] showing that this achieved the best results compared to other kinds of input regions. Furthermore, to fully use the information of the full-face, we proposed a soft attention mechanism [55] and a hard attention mechanism [61] to efficiently learn information form the full-face patch. In [55], we allow the neural network to self-predict varying weights for different regions of the input face in order to make model training efficient. However, in contrast to object classification tasks where the scale of activation values of each feature map is correlated to the importance of a template or object class, gaze estimation as a regression task can benefit from an attention mechanism that goes beyond activation value modulation. Our later work proposes a hard attention mechanism to force the model to focus on the sub-regions of the face [61]. Taken a full-face patch as input, our method first crops sub-regions with multiple region selection networks. These sub-regions are then passed as input to the gaze regression network which predicts gaze direction. Since each sub-region is resized to be the same as the original input face image, the receptive field is enlarged, thus, the gaze regression model can extract large and informative feature maps from the sub-regions. This method successfully picks the most appropriate eye region for gaze estimation depending on different input image conditions such as occlusion and lighting conditions. However, the model itself can be difficult to train and take much time to converge. How to efficiently learn the information from the full-face patch is still an ongoing research topic.

3.4 Representation learning for gaze estimation

In addition to studying various methods of input region selection for gaze estimation, we also suggest various approaches to learning unique gaze-specific representations in neural networks. Such representations can be explicitly defined or implicitly learned. The first representation as proposed in [34] is explicitly defined as being eye-region landmark coordinates. The fully convolutional network proposed in this work is able to detect eye-region landmarks from images captured with a single webcam, even under challenging lighting conditions. Compared to the classic edge-based eye landmark detection method [16, 48], the convolutional network provides more robust landmark prediction. These detected landmarks are then be used for model-based or feature-based gaze estimation. However, since it still requires eye landmark detection, this method can only work in settings with the close distance between user and camera such as the laptop and desktop setting [59] and relies on high-quality synthetic training data. We further improve our method by first predicting a novel pictorial representation that we call a "gazemap", then use it as input for a light-weight gaze regression network [33]. In this work, the proposed method leverages the power of hourglass networks to extract this image-based "gazemap" feature which is composed of silhouettes of the eyeball and the iris. It is an abstract, pictorial representation of eyeball structure which is the minimally essential information necessary for the gaze estimation task. The gazemap representation is not explicitly correlated with key landmarks in the input eye image and can be generated from the 3D gaze direction labels. Hence, the latter approach can be applied to models that need to be trained directly on real-world data. The alternative is to train on synthetic data, which can result in a model that does not perform sufficiently well due to the domain gap between synthetic and real data domains.

3.5 Gaze estimation datasets

To train a generic gaze estimator that can be applied to a large variety of conditions and devices, it is critical that learning-based gaze estimation methods are trained with datasets that have good coverage of real-world conditions. Unless the model has had a chance to encounter data with large variations, it could suffer due to overfitting to the more limited training data and perform in unexpected ways outside of the original data regime. Essentially, we should not expect learned models to handle samples that are out-of-distribution. Specifically for assessing a dataset for the gaze estimation task, there are several factors that should be considered, such as the range of gaze direction, range of head poses, diversity of lighting conditions, variety of personal appearances and input image resolution.

Early datasets mainly focus on the head pose and gaze direction coverage under controlled lighting conditions such as UT-Multiview [43] and EYEDIAP [12]. Our MPIIGaze dataset, as the first of its own kind, brought the task of gaze estimation out from the conventional and controlled laboratory setting out into the real-world setting which covers different lighting conditions [53, 56]. This was done by installing a data recording software on 15 participants' laptop computers and prompting the participant every 10 minutes to ask for 20 gaze data samples. In this task, participants were asked to look at dots on the screen as they appear, then pressed the space button to confirm that he/she was looking at the dot. By this way, we could record the dot that the participant was looking at, and at the same time, the position of the on-screen dot was stored, along with an image of the participant's face taken with the built-in camera of the laptop. Since the data samples were collected without restriction on location and time, we were able to collect samples under many different lighting conditions with natural head movement. However, since the MPIIGaze dataset was collected with laptop devices, the head pose and gaze direction ranges are limited to the size of typical laptop screens. Therefore, models trained only on MPIIGaze data may not apply well to settings with larger displays and viewing distances, for example, participants gazing at a TV in a public space.

Such limitation by the capture device appears in many existing datasets. Similar to our MPIIGaze, the GazeCapture dataset limited themselves with small ranges of head poses and gaze directions due to using mobile phone and tablet devices for data collection [24]. The EYEDIAP dataset is designed specifically for head poses

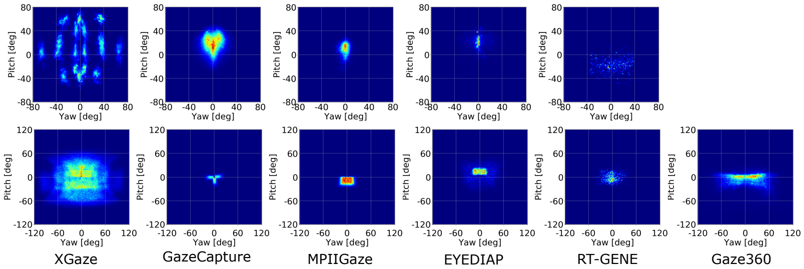
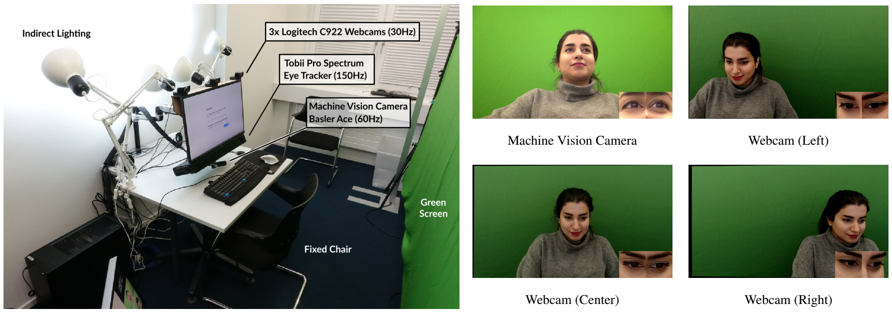


Fig. 3: Head pose (top row) and gaze direction (bottom row) distributions of different datasets. The head pose of Gaze360 is not shown here since it is not provided by the dataset. The figure is adapted from [60].

and gaze directions of the desktop setting [12]. The RT-GENE dataset tried to use a head-mounted eye tracker to provide accurate gaze direction ground-truth and large spatial coverage of head poses and gaze directions [10]. The recent Gaze360 dataset used a moving camera to simulate different head poses [21]. However, the image and ground-truth quality were not guaranteed with these datasets, and the coverage of head poses and gaze directions were not properly designed.

We provide the ETH-XGaze dataset consisting of over one million high-resolution images of varying gaze directions under extreme head poses [60]. This dataset was collected with a custom setup of devices including a screen to show visual content from a projector, four lighting boxes to simulate different lighting conditions, and 18 digital SLR cameras which can capture high-resolution (6000 × 4000 pixels) images. The cameras were arranged such as to cover different perspectives of the face of the participant, effectively making each camera position correspond to one "head orientation" in the final processed dataset. Since the participant was placed close to the screen, a large range of gaze directions were captured during each recording session. A comparison of head pose and gaze direction ranges are made between our ETH-XGaze dataset and other datasets in Figure 3. From the figure, we can see that our ETH-XGaze dataset provides the largest range of head poses and gaze directions compared to previous datasets. ETH-XGaze is a milestone towards providing full robustness to extreme head orientations and gaze directions and should enable the development of interesting novel methods that better incorporate understandings of the geometry of the human head and the eyeball within.

In addition to exploring the spatial dimension with the 18-camera ETH-XGaze dataset, we chose to explore the temporal dimension of gaze tracking in an end-to-end fashion. That is, we aimed to go beyond the static face images provided by most gaze estimation datasets by providing video data. In addition, we observed that when humans gaze at objects or other visual stimuli, their eye movements are often correlated with particular changes or movements in the stimuli. Yet, no large-scale video-based dataset exists to relate the change in the appearance of the human directly to a video of the visual stimulus. To fill this gap, we proposed another novel dataset called EVE to provide temporal information of both the human face and


Indirect Lighting
3x Logitech C922 Webcams (30Hz)
Tobii Pro Spectrum Eye Tracker (150Hz)
Machine Vision Camera Basler Ace (60Hz)
Green Screen
Fixed Chair
Machine Vision Camera
Webcam (Left)
Webcam (Center)
Webcam (Right)

(a) Collection Setup

(b) Sample frames (each 1080p)

Fig. 4: EVE data collection setup and example of (undistorted) frames collected from the 4 camera views with example eye patches shown as insets [36]

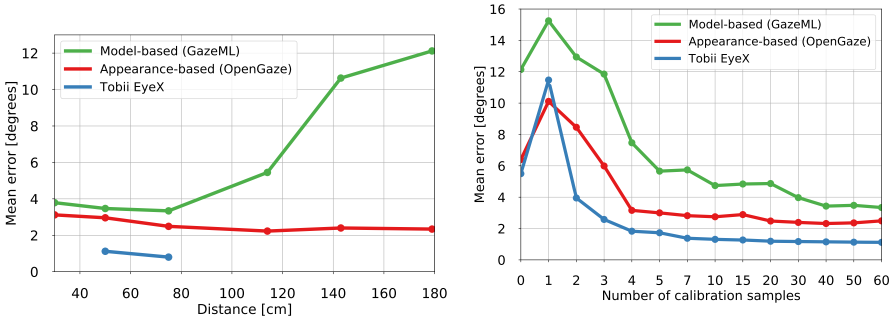
corresponding visual stimulus for improving the temporal gaze estimation task [36]. The EVE dataset was recorded with four video cameras facing the participants, with various visual contents shown on the screen. The custom multi-view data capture setup and example frames are shown in Figure 4. The custom setup synchronized information from three webcams running at 30Hz, one machine-vision camera running at 60Hz, and one Tobii Pro Spectrum eye tracker running at 150Hz. A large variety of visual stimuli were presented to our participants including images, videos, and Wikipedia webpages. We ensured that each participant observes 60 image stimuli (for three seconds each), at least 12 minutes of video stimuli, and six minutes of Wikipedia stimulus (three 2-minute sessions). To our understanding, EVE is the first dataset to provide continuous video recordings of both the user and the visual stimuli while the user is free-viewing the presented visual stimuli. Alongside the dataset, we propose a method which shows that when a video of the user and screen content are taken as input, it is possible to correct for biases in a pre-trained gaze estimator by relating changes in the screen content with eye movement. Effectively, this allows for calibration-free performance improvements, finally yielding 2.5° of mean angular error.

3.6 Comparison of learning-based and commercial gaze estimation methods

Learning-based gaze estimation methods have developed rapidly and now begin to challenge classical methods. However, an accurate comparison of different gaze estimation methods is not a trivial task since they have different requirements in terms of capture hardware and lighting conditions. In [59], we compared three typical gaze estimation methods including two of our webcam-based methods [34, 55] and the commercial Tobii EyeX eye tracker on data collected from 20 participants. Our method proposed in [34] uses a neural network to predict eye-region landmarks which are then used for model-based gaze estimation. Our method proposed in [55] directly learns the mapping from the input face image to the gaze direction with a neural network as an appearance-based method. We do not know what exact method the commercial Tobii EyeX eye tracker uses for gaze estimation and calibration.

We mounted both a webcam and the Tobii EyeX below a screen, and then asked participants to look at displayed point stimuli from different distances to the screen. In this way, we collected the gaze direction ground-truth of the participants. We resized the region on the screen to show the visual stimuli according to the different distances such that gaze direction ranges for each distance were the same. We recorded 80 samples of which the first 60 were for personal calibration and the rest (20 samples) were for testing. The data collection setup was designed to be highly controlled to allow for a reliable comparison of performance across different gaze estimation methods for varying amount of user-camera distances and number of gaze tracker calibration samples. This was done by fixing the lighting conditions in the environment, asking the participants to keep their head relatively still, and collecting the calibration and test samples in a single session.

The main results of our comparison are shown in Figure 5. From the figure on the left, we can see that our model-based method [34] can work well for close distances while it becomes much worse when the distance between the user and camera increases. This is because this method relies on accurate estimations of eye-region landmarks. The Tobii EyeX eye tracker achieves the best performance (with lowest gaze estimation error) since it has dedicated hardware including high-resolution cameras and active lighting sources. However, our appearance-based method [55] provides the robust gaze estimation performance across different distances between the user and camera. This means that the appearance-based gaze estimation method can be applied to many more applications, for example, room-level human attention estimation from cameras placed far away from the users. On the right of Figure 5,



(a) Collection Setup

Fig. 5: Gaze estimation errors of different methods in degrees across distances between the user and camera (left), and number of samples for the personal calibration. Dots are results averaged across all 20 participants and we linked them by lines [59].

we can see that three approaches only need a few personal calibration samples to reach reasonable accuracy. However, the number of necessary calibration samples may increase for real-world applications compared to this simple setting.

4 Making gaze tracking practicable for computer interaction

Applying webcam-based gaze estimation methods to real-world interactive applications poses practical challenges. One of the key issues is that collecting personal calibration data is tedious for the user. However, without it the gaze estimation accuracy is poor. Even after personal calibration, the predicted gaze positions often show a large amount of noise, also for commercial eye trackers. Therefore, using gaze for interaction requires carefully designed user interfaces (UI) that take into account this potential noise and thus the uncertainty of the input signal. Otherwise, it results in bad user experience or interaction is not even possible. Another problem is that unlike existing commercial eye-tracking devices that can be directly used out-of-the-box, learning-based methods are still under development and may not lend themselves as simple solutions for novice users. In this section, we discuss our efforts toward making gaze tracking practicable for human-computer interaction.

4.1 Personalizing gaze tracking methods

In principle, there are two main challenges for learning-based gaze estimation methods caused by personal differences. The first one is the kappa angle which varies by around two to three degrees on average across people [29]. The second challenge is personal eye appearance differences such as the shape of the eye, colour of the iris, etc. The eye appearance is also affected by changes in gaze directions and head poses, which is further connected to the image capturing or personal computing device. For example, a gaze estimator trained on images captured on a smartphone device that is held closer to the user may not perform well when directly applied to a large public display such as an advertisement board in a shopping mall. This could be caused by loss of image resolution and quality and unfamiliar head poses during operation. Due to these challenges, learning-based methods may benefit from further adaptation in challenging conditions that are not covered by the training data.

A basic experimental observation is that increasing the number of dataset participants results in improved general gaze estimation performance [24]. That is, learning from more peoples' data allows for a method that generalizes better to previously unknown users. However, as introduced in Section 2.4, there still exists a large performance gap that can be recovered when using just a few samples from the final target user to adapt learned models.

Nevertheless, collecting personal calibration data is still an effective way for good gaze estimation performance. In our work [57], we proposed to use multiple types of devices to collect samples for specific users, then aggregate all of these samples to train a joint model for that specific user across devices. The intuition behind this work is that the personal appearance should be the same for different devices which we can learn with the shared layers in the middle of our model. Our approach can benefit applications that are expected to be used by a user over a long period of time and across multiple devices, with personal calibration data being collected occasionally.

An alternative and promising method of increasing the amount of training data for specific persons is in generative modelling. Given a few labeled samples, a high-quality generative model would be able to create tailored training data from which a robust yet personalized gaze estimation model could be learned. Our first work in this direction used an architecture based on CycleGAN [63] for realistic eye image generation, where gaze direction is provided as an input condition to the network and training is supervised via perceptual and gaze direction losses [18]. Although this method is successful at generating photo-realistic images of the eye, it is not aware of head orientation and cannot easily be trained with noisy real-world images. We later proposed a transforming encoder-decoder architecture to tackle these issues, where features pertaining to gaze direction, head orientation, and other appearance-related factors are explicitly defined at the bottleneck of the autoencoder [62]. To truly enable training on in-the-wild datasets, we allow for the definition of implicitly defined "extraneous" factors at the bottleneck. The reconstruction objectives allow for these extraneous factors to encode information that is task-irrelevant yet allow for the satisfaction of the image-to-image translation objective. This approach, in particular, was shown to improve performance in the person-independent cross-dataset setting, but with further development, it should be possible to demonstrate improvements in the personalizing of gaze trackers. While personalized data collection is a promising and active direction of research, much work is yet needed for it to be effective.

Alternatively, our other research works show that learning-based gaze estimator calibration is definitely possible with tens of samples using simple regression schemes and with as few as one to three samples when using a more advanced meta-learning scheme. By defining input features using eye-region landmarks detected by a fully convolutional neural network, we show that a support vector regression model is capable of improving performance significantly with as few as 10 calibration samples. An appearance-based gaze estimator taking full-face input images was shown to be effective in tandem with a simple polynomial regression scheme taking point-of-regard as input, resulting in less than 4° of error with just 4 calibration samples [59], albeit in controlled experimental settings. When training on real-world data, a transforming encoder-decoder architecture coupled with a gradient-based meta-learning scheme was shown to be highly effective, with as few as one to three calibration samples yielding close to 3° of error on challenging in-the-wild datasets [35]. The code for the latter two systems are open-source and thus contribute towards effecting real improvements with regards to the applicability of learning-based gaze estimation methods to HCI applications.

4.2 Design of robust interfaces

The estimated gaze data can be highly noisy and inaccurate. Nevertheless, it can be used for computer input, to improve user experience or otherwise enable new interaction if potential noise is taken into account during the design of gaze-aware interfaces. To this end, we have studied tracking performance in practical set ups to derive design guidelines and actionable insights for the design of robust gaze-aware interfaces.

In [7], we collected eye-tracking data of 80 people in a calibration-style task, where participants were asked to fixate randomly positioned targets on the screen for two seconds. We used two different eye trackers (Tobii EyeX and SMI REDn scientific, both at 60Hz) under two lighting conditions (closed room with artificial lighting, room with large windows facing the tracker) in a controlled but practical setup. In contrast to many lab studies, we did not exclude any participant due to insufficient tracking quality. Instead, we were interested in learning about the possible variations in tracking accuracy (the offset from the true gaze point) and precision (the spread of tracked gaze points). These could be due to the independent variables of our study (lighting, tracker, screen regions), as well as due to external factors that we did not control but that are typical for real-life set ups (participants wearing glasses or mascara, varying eye physiology, etc.).

The collected data reveals large variations of tracking quality in such a practical setup. Figure 6 shows the average accuracy and precision across all focused targets for different percentiles of participants. Very accurate fixations (25th percentile) are only 0.15 *cm* in the x- and 0.2 *cm* in the y-direction offset from the target. On the other hand, inaccurate fixations (90th percentile) can be as far offset as 0.93 *cm* in the x- and 1.19 *cm* in the y-direction — a more than six-fold difference. Similar to the spread of the gaze points. Additionally, we found the precision of the estimated gaze points to be worse toward the right and bottom edge of the screen, as shown in Figure 7. The ellipses represent the covariance matrix computed over all gaze points from all participants. In contrast, we found no significant variation across the screen for accuracy.

With data from such a calibration-style task, we can derive appropriate sizes for gaze targets, i.e. the regions in a UI that recognize if the user's gaze falls inside its borders. Given the gaze points belonging to a fixation, we can assume they are normally distributed in x- and y- direction independently, with an offset $O_{x/y}$ (accuracy) from the center of the fixated target and a standard deviation $\sigma_{x/y}$ (precision). From these, we can compute the necessary width and height for a gaze-aware element to be usable under such tracking conditions with the following equation:

$$S_{w/h} = 2(O_{x/y} + 2\sigma_{x/y}) \tag{1}$$

Multiplying $\sigma_{x/y}$ by 2 results in about 95% of gaze points falling inside the target, according to the properties of a normal distribution. While this seems conservative, an error rate of more than 5% (every 20th gaze point falling outside the target area) might slow down performance and lead to errors that can be hard to recover from.

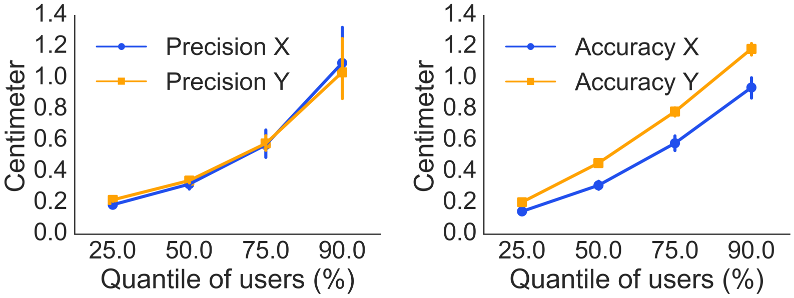


Fig. 6: Accuracy and precision of gaze tracking varies largely across users. Values increase steeply for different percentiles of users both in x- and y-direction.

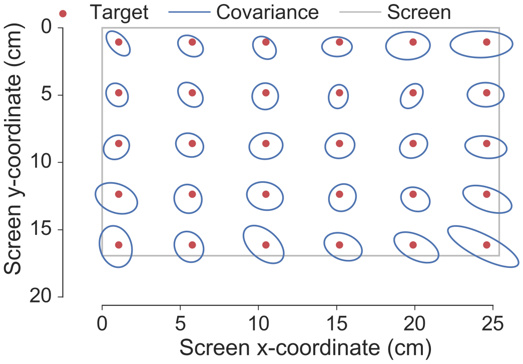


Fig. 7: The precision of the estimated gaze points varies for different screen regions. For each target, the ellipse shows the 2D Gaussian distribution fitted to the estimated gaze points of all participants fixating that target [7].

Figure 8 visualizes the size computation and shows two example cases with good and bad tracking quality. In [7] we give explicit target sizes for different percentiles of users. They vary from $0.94 \times 1.24\ cm$ for users that track well (25th percentile), up to $5.96 \times 6.24\ cm$ if we want to allow robust interaction for nearly all users in the dataset (90th percentile).

Target sizes can be significantly reduced if the gaze data is first filtered to remove noise artifacts and reduce signal dispersion. However, in contrast to laboratory studies, interactive applications cannot post-process the gaze data but must filter it in real-time. This makes the recognition of outliers and artifacts difficult since it can introduce delays of several frames. Gaze filters must also account for the quick and sudden changes between saccades and fixations. In contrast, eye tremor, microsaccades, and noise should be filtered in order to stabilize the signal and improve precision. This makes commonly used methods, such as moving average, Kalman filter, or Savytzki-Golay filter less useful [45].

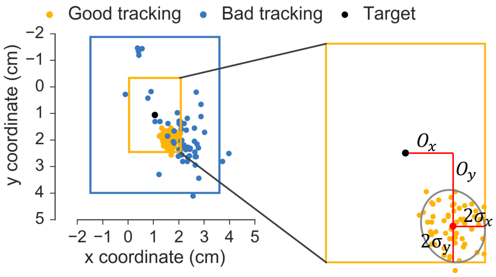


Fig. 8: Using the accuracy ($O_{x/y}$) and precision ($\sigma_{x/y}$) of the estimated gaze points belonging to a fixation we can compute target sizes that would allow for robust interaction with an interface. The plot shows examples from fixations of two different users one with good and one with poor tracking quality [7]

The choice of the filter and its parameters can be seen as a trade-off between the target size required for robust interaction and the signal delay in following a saccade. In [7] we proposed a method to optimize the parameters for any filter given gaze data from a calibration-style data as described earlier. In a grid search, it instantiates a filter with each possible parameter, computes the resulting target size after filtering the data, and simulates saccades between such targets to determine any signal delay. The result is a pareto front of parameter combinations that yield the minimum target size for a specific delay.

Using this method, we compare five commonly used gaze filters with three different kernel filters: the Stampe filter, the 1€ filter, a set of weighted average filters with linear, triangular, and Gaussian kernel functions, an extension with saccade detection and one with additional outlier correction. See [7] for a description of each filter. The filters differ in the trade-offs they achieve for target size and signal delay. Generally, we found that a weighted average filter with a saccade detection performs best in terms of target size when signal delay should be short (up to one frame or 32 ms with a 30Hz tracker). The best performance is achieved with additional outlier correction at the cost of $2--2.5$ frames delay.

The use of a filter with optimized parameters can reduce the target sizes by up to 42% (see Table 1). However, the filter can only improve the precision of the data, not its accuracy. Simulation-based on real data yields important insights into the effect of filters on the signal. Filters that by design should not introduce any or only a short signal delay, in practice introduce much larger delays to the gaze signal. For example, depending on the noise and set parameters, it may wrongly detect saccades as outliers or as part of fixation and either remove them or heavily smooth the signal. In such cases, an additional delay occurs before the filtered signal follows a saccade to a new fixation point. See [7] for an in-depth discussion of the tested filters.

We can summarize our analysis in a set of concrete design guidelines for gaze-enabled applications:

	Width (cm)			**Height (cm)**		
	Raw	Filtered	Improv.	Raw	Filtered	Improv.
Overall	3.0	2.02	33%	3.14	2.19	30%
Percentile						
25%	0.94	0.58	38%	1.24	0.8	35%
50%	1.8	1.12	38%	2.26	1.48	35%
75%	3.28	1.9	42%	3.78	2.35	38%
90%	5.96	3.9	35%	6.24	4.24	32%

Table 1: Recommended target sizes for robust interaction by eye gaze. The values for raw and filtered show the improvement that can be achieved by filtering the gaze data. The percentiles show how much target sizes can vary for different levels of tracking quality [7].

- **Target sizes** of at least 1.9×2.35 *cm* allow for reliable interaction for at least 75% of users if optimal filtering is used.
- **Target dimensions** should take into account the larger spread of gaze points in y-direction we observed. Thus, the height should be somewhat larger than the width.
- **Visual representation** of elements can be smaller in which case the element should have a transparent margin that is also reactive to the user's gaze.
- **Placement** of targets should avoid the bottom or right edge of the screen, for which accuracy and precision was found to be significantly worse.
- **Filter** gaze points using a weighted average filter (over 36/40 frames in x/y direction) with a Gaussian or Triangular kernel and saccade detection (threshold of 1.45/1.65 cm in x/y direction). Additional outlier correction can further improve precision but at the cost of a two-sample delay.

4.3 Make single-webcam based methods accessible for HCI researchers

To allow learning-based gaze estimation methods to be used out-of-the-box in a similar manner to commercial eye trackers, we published OpenGaze[1]. OpenGaze includes the entire gaze estimation pipeline, beginning from the acquisition of a single RGB image to prediction of the gaze direction in the camera coordinate system. Therefore, it can be used with just a single webcam as the input device. OpenGaze is based on the appearance-based method in [55] that directly learns the mapping from input face image to the gaze direction without explicit eye landmarks detection. Therefore, it is particularly effective when the distance between the user and camera is high. The full description of OpenGaze and evaluation can be found in our paper [59].

[1] `http://www.opengaze.org`

We also publish GazeML[2] which is a demonstration of the approach in [34]. It uses stacked-hourglass networks to predict eye-region landmark heatmaps and an estimate of gaze direction. As it was only built for demonstrative purposes, its outputs are not suitable for actual gaze estimation nor can the software be easily adapted for HCI applications. Yet, it is an interesting demonstration of the possibility of real-time gaze estimation using deep convolutional neural networks.

5 Applications

Eye-tracking provides information on where a user is looking at, the dynamics of the gaze behaviour, or the simple presence of the eyes on an object or screen. Such information offers a range of opportunities for computer interaction (see for example [28]). On the one hand, *explicit eye input* allows controlling an interface by fixating the corresponding UI elements or executing a prescribed series of fixations, saccades, or smooth pursuits. This requires users to consciously control their eyes which can be difficult but useful when other input modalities are not available or impractical. Explicit gaze input is used for example in virtual or augmented reality applications [19, 22] or to enable interaction for people with motor impairments [27]. On the other hand, *attentive interfaces* use information about the natural gaze behaviour of users often without them noticing. They can obtain insights on the user's experience with an interface, their cognitive processes, their skills or struggles, their intentions or preferences [14, 26]. In this section, we focus on such attentive interfaces that make implicit use of the gaze information. We present two applications that use this data in different ways: (1) as a way to establish a user's intention to interact with a device by tracking the location of their natural gaze, and (2) for adapting the interface to make the displayed information more relevant to a user by observing their gaze behaviour over time.

5.1 Gaze-aware real-life objects

Gaze-awareness, that is recognizing when a user is looking at a specific element, is an important functionality of intelligent interactive system and the core of attentive interaction [3]. Also in real-life settings, interactive systems can benefit from sensing where or which object a user is looking at in their environment. However, the position of interactive devices can be arbitrary inside a room, making it difficult to identify the layout of multiple potential objects. In our work [54], we proposed a novel method for user-object eye contact detection that combines state-of-the-art learning-based gaze estimation [55] with a novel approach for unsupervised gaze target discovery, i.e. without the need for tedious and time-consuming manual data annotation.

[2] `https://github.com/swook/GazeML`

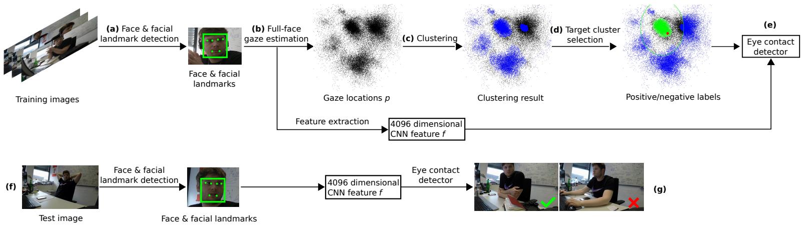


Fig. 9: Overview of our method in [54]. Taking images from the camera as input, our method first detects the face and facial landmarks (a). It then estimates the gaze directions $\boldsymbol{p}$ and extracts CNN features f using a full-face appearance-based gaze estimation method (b). During training, the gaze estimates are clustered (c) and samples in the cluster closest to the camera get a positive label while all others get a negative label (d). These labelled samples are used to train a two-class SVM for eye contact detection (e). During testing (f), the learned features $\boldsymbol{f}$ are fed into the two-class SVM to predict eye contact on the desired target object or face (g).

Our method works with the assumption that the target object is the one closest to the camera, thus, our method only requires a single off-the-shelf RGB camera placed close to the target object. Once the camera is placed, the approach does not require any personal or camera-object calibration . As illustrated in Figure 9, the input to our method is the video sequence from the camera over a period of time. During the training, our method runs the gaze estimation pipeline introduced in our work [55] to obtain the estimated gaze direction. Assuming dummy camera parameters, the estimated gaze direction vector $\boldsymbol{g}$ is projected to the camera image plane and converted to on-plane gaze locations $\boldsymbol{p}$. While the gaze estimation results are used for sample clustering, we extract a 4096-dimensional face feature vector $\boldsymbol{f}$ from the first fully-connected layer of the neural networks.

As we stated in [55], the estimated gaze direction $\boldsymbol{g}$ is not accurate enough without personal calibration, and it cannot be mapped directly to the physical space without the camera-object relationship parameter. However, it indicates the relative gaze direction of the user from the camera position. Hence these estimated gaze directions can be grouped into multiple clusters corresponding to several objects in front of the user. Given that our method assumes that the target object is the one closest to the camera, the sample cluster of the target object is identified as the cluster closest to the origin point of the camera coordinate system. Other clusters are assumed to correspond to other objects, and samples from these clusters are used as negative samples.

Labelled samples obtained from the previous step are used to train the eye contact classifier. This is a two-class classifier that determines if the user is looking at the target object or not in the current input frame. We use a high-dimensional feature vector $\boldsymbol{f}$ extracted from the gaze estimation network to leverage richer information instead of only gaze locations. Furthermore, we apply principal component analysis

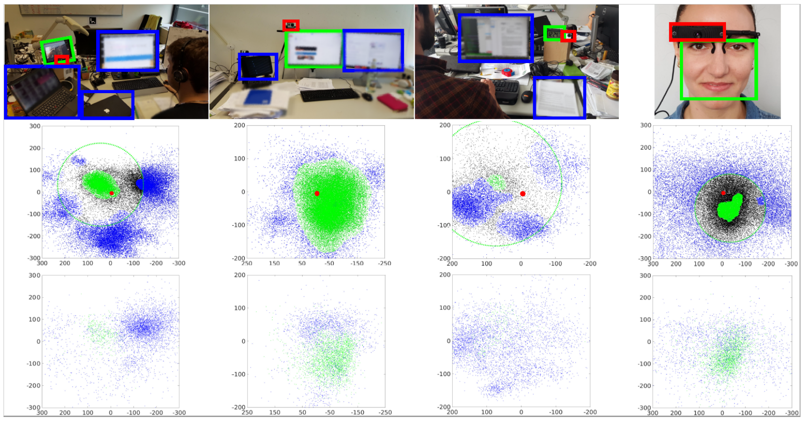


Fig. 10: Examples of gaze locations distribution for the object-mounted (tablet, display, and clock) and head-mounted settings from [54]. The first row shows the recording setting with marked target objects (green), camera (red), and other distraction objects (blue). The second row shows the gaze locations clustering results with the target cluster in green and negative cluster in blue. The third row shows the ground-truth gaze locations from a subset of 5,000 manually annotated images with positive (green) and negative (blue) samples.

(PCA) to the training data and reduce the dimension of feature vector $\boldsymbol{f}$ that the subspace retains the 95% variance.

During testing, input images are fed into the same pre-processing pipeline with the face and facial landmark detection, and feature $\boldsymbol{f}$ is extracted from the same gaze estimation neural networks. It is then projected to the PCA subspace, and the SVM classifier is applied to output eye contact labels. Note that during both the training and test phase, we neither need to label the input frame sample nor calibrate the camera-object relationship.

To evaluate our method for eye contact detection, we collected two datasets for two challenging real-world scenarios: office scenario and interaction scenario. The example of the two scenarios is shown in Figure 10. For the office scenario, the camera is object-mounted as the camera was mounted or placed very near to the target object, and we aimed to detect eye contact of a single user with these target objects during everyday work at their workplace. We recorded 14 participants in total (five females) and each of them recorded four videos for different target objects: one for the clock, one for the tablet, and two for the display with two different camera positions. The recording duration for each participant ranged between three and seven hours.

In the interaction scenario (see far right of Figure 10), a user was wearing a head-mounted camera while being engaged in everyday social interactions. This scenario was complementary to the office scenario in that the face of the user became the target and we aimed to detect eye contact of the second person who talked with the user. We recruited three users (all male) and recorded them while they interviewed multiple people on the street.

The example of gaze location distribution for the two scenarios is shown in Figure 10. In the first row we show the recording settings for the different target objects. We mark the target object (green rectangle), camera (red rectangle) positions and other distraction objects (blue rectangle) in the figure. The second row of Figure 10 shows sample clustering results where we mark the target cluster with green dots while all other negative sample clusters are marked with blue dots. Noise samples are marked as black and the big red dot is the camera position as the origin of the camera coordinate system. The third row shows the corresponding ground-truth annotated by two annotators.

From the second row of Figure 10, we can see that the grouped sample clusters can be associated with objects in front of the camera, especially for the office scenarios as object layout is fixed. For the interaction scenario, we can observe one centered cluster and other random distributed samples. This is due to the fact that there is no fixed attractive object next to the user's face. Our sample clustering method can achieve good clustering result and successfully pick the cluster belongs to the target object. It can also be easily extended to include objects that are newly added to the scene by updating the clusters. However, our method requires sufficient data for good clustering - usually about few hours recording. Besides, the target object should attract enough attention to the user and it has to be isolated from other objects. Nonetheless, our method provides a way of eye contact detection with a single RGB camera without neither tedious personal calibration nor complex camera-object relationship calibration.

5.2 Adapting a UI to improve information relevance

The user's gaze behavior can reveal whether the content displayed to a user is useful and relevant to their current task. In particular when making a decision, showing the right information to the user is crucial for the decision quality [31]. For example, a user might look at the details of a product for deciding whether to buy it or check the weather forecast to decide whether to go for a hike. If important information is missing from an interface, a user might be affected and make a wrong decision. On the other hand, displaying all available information might not be effective due to device constraints (e.g. on a small screen of a mobile phone) or because it might lead to information overload and a bad user experience. What makes the design of such interfaces challenging is also that users perceive the relevance of information differently [30], an aspect that cannot be foreseen at design time but must be detected and accounted for at run-time. However, the challenge is how to infer the relevance of the displayed information online, without having to interrupt users in their task.

Eye gaze has proven to be an unobtrusive and objective measure for a person's attention [37]. In this section, we show how we can analyze this data during the decision process of a user to obtain insights on the relevance of the displayed

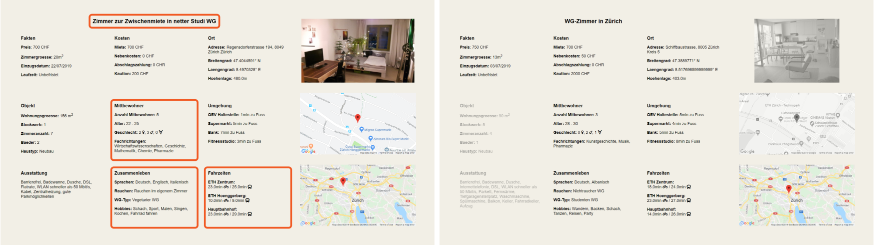
	Orientation	
TFF	*Time to First Fixation*	The time elapsed between the presentation of a stimulus and the first time that gaze enters a given AOI. A low TFF value indicates high relevance.
FPG	*First Pass Gaze*	The sum of duration of fixations on an AOI during the first pass, i.e. when the gaze first enters and leaves the AOI. A high FPG value indicates high relevance.
	Evaluation	
SPG	*Second Pass Gaze*	The sum of duration of fixations on an AOI during the second pass. A high SPG value indicates high relevance.
RFX	*Refixations Count*	The number of times an AOI is revisited after it is first looked at. A high RFX value indicates high relevance.
	Verification	
SFX	*Sum of Fixations*	The total number of fixations within an AOI. A high SFX value indicates high relevance.
ODT	*Overall Dwell Time*	The total time spent looking at an AOI including fixations and saccades. A high ODT value indicates high relevance.

Table 2: For recognizing information relevance from gaze behavior we combine six well-established gaze metrics which we can associate with the three cognitive stages of decision making. [8]

information [8]. This requires no explicit user input but analyzes the natural gaze behavior of the user while they focus on their decision-making task. In contrast to simpler, visual search tasks, the challenge is that the gaze behavior varies drastically during the decision process as users transition from obtaining an overview of the UI to comparing relevant information to finally validating their decision [15, 38].

To account for this variation, we select six different gaze metrics which were all shown to effectively infer a person's covert attention in simpler search tasks. However, the gaze behavior during decision-making is more complex and affected by the three cognitive stages the user goes through. Each metric captures a different aspect of these stages. Following Russo and Leclerc [38], we refer to them as (1) Orientation, (2) Evaluation, and (3) Verification. In the first stage, the user obtains an overview of the available information, characterized by a scanning pattern of shorter fixations without many return-fixations. The user then compares the information determined as relevant going back and forth between the same UI elements. Finally, short fixations on the most relevant information are used to validate the decision. While a clear separation of these stages is difficult, they inform our selection of gaze metrics that capture the different gaze characteristics during decision-making. These are shown in Table 2.

Each metric can be seen as a weak classifier which outputs a binary decision whether a UI element is considered relevant by the user or not. By allowing multiple metrics to vote on an element's relevance, we imitate a multiple-classifier system while avoiding the need for training data. We say that a metric casts a vote for a UI element as being relevant if its standard score (z-score) for the element is positive.



Zimmer zur Zwischenmiete in netter Studi WG

Fakten
Preis: 700 CHF
Zimmergroesse: 20m^2
Einzugsdatum: 22/07/2019
Laufzeit: Unbefristet

Kosten
Miete: 700 CHF
Nebenkosten: 0 CHF
Abschlagszahlung: 0 CHR
Kaution: 200 CHF

Ort
Adresse: Regensdorferstrasse 194, 8049 Zürich Zürich
Breitengrad: 47.4044591° N
Laengengrad: 8.4970328° E
Hoehenlage: 480.0m

Objekt
Wohnungsgroesse: 156 m^2
Stockwerk: 1
Zimmeranzahl: 7
Baeder: 2
Haustyp: Neubau

Mittbewohner
Anzahl Mitbewohner: 5
Alter: 22 - 25
Geschlecht: 2 ♀, 3 ♂, 0 ⚥
Fachrichtungen: Wirtschaftswissenschaften, Geschichte, Mathematik, Chemie, Pharmazie

Umgebung
OEV Haltestelle: 1min zu Fuss
Supermarkt: 5min zu Fuss
Bank: 7min zu Fuss
Fitnessstudio: 3min zu Fuss

Ausstattung
Barrierefrei, Badewanne, Dusche, DSL, Flatrate, WLAN schneller als 50 Mbit/s, Kabel, Zentralheizung, gute Parkmöglichkeiten

Zusammenleben
Sprachen: Deutsch, Englisch, Italienisch
Rauchen: Rauchen im eigenem Zimmer
WG-Typ: Vegetarier WG
Hobbies: Schach, Sport, Malen, Singen, Kochen, Fahrrad fahren

Fahrzeiten
ETH Zentrum: 23.0min 🚲 / 25.0min 🚌
ETH Hoenggerberg: 10.0min 🚲 / 9.0min 🚌
Hauptbahnhof: 23.0min 🚲 / 29.0min 🚌

WG-Zimmer in Zürich

Fakten
Preis: 750 CHF
Zimmergroesse: 13m^2
Einzugsdatum: 03/07/2019
Laufzeit: Unbefristet

Kosten
Miete: 700 CHF
Nebenkosten: 50 CHF
Abschlagszahlung: 0 CHR
Kaution: 2000 CHF

Ort
Adresse: Schiffbaustrasse, 8005 Zürich Kreis 5
Breitengrad: 47.3889771° N
Laengengrad: 8.517696599999999° E
Hoehenlage: 403.0m

Objekt
Wohnungsgroesse: 90 m^2
Stockwerk: 5
Zimmeranzahl: 4
Baeder: 1
Haustyp: Neubau

Mittbewohner
Anzahl Mitbewohner: 3
Alter: 28 - 30
Geschlecht: 0 ♀, 2 ♂, 1 ⚥
Fachrichtungen: Kunstgeschichte, Musik, Pharmazie

Umgebung
OEV Haltestelle: 5min zu Fuss
Supermarkt: 4min zu Fuss
Bank: 5min zu Fuss
Fitnessstudio: 8min zu Fuss

Ausstattung
Barrierefrei, Badewanne, Dusche, Internettelefonie, DSL, WLAN schneller als 50 Mbit/s, Parkett, Fernwärme, Tiefgaragenstellplatz, Waschmaschine, Spülmaschine, Balkon, Keller, Fahrradkeller, Aufzug

Zusammenleben
Sprachen: Deutsch, Albanisch
Rauchen: Nichtraucher WG
WG-Typ: Studenten WG
Hobbies: Wandern, Backen, Schach, Tanzen, Reisen, Party

Fahrzeiten
ETH Zentrum: 18.0min 🚲 / 24.0min 🚌
ETH Hoenggerberg: 23.0min 🚲 / 27.0min 🚌
Hauptbahnhof: 14.0min 🚲 / 26.0min 🚌

Fig. 11: Two alternative ways to adapt a room search interface. Left: relevant content is highlighted through color boxes. Right: irrelevant information is suppressed by greying it out [8].

Intuitively this means that for that element, the gaze metric deviates from the average across all elements indicating a different gaze behavior of the user. To establish the relevance of an element, we count the number of votes cast by the 6 metrics and compare it to a threshold. Requiring a higher number of votes yields a lower number of elements being detected as relevant. This is further discussed below. In any case, this approach does not assume a fixed number of relevant elements a priori. Also, it is training-free and requires no ground-truth data.

Once we know whether the displayed information is relevant for the user's decision we can adapt the interface to facilitate the decision process. Broadly speaking, many of the adaptation techniques proposed in the literature (see e.g. [9]) can be divided into two types: (1) emphasizing relevant content (e.g. coloring, rearranging or replicating elements) or (2) suppressing irrelevant information (e.g. greying out, removing, or moving elements to less prominent positions). See Figure 11 for an example application. To obtain a benefit from adaptation, it is critical to minimize the risk of usability issues due to wrong adaptations. For example, wrongly highlighting seldom used elements in a menu can induce a performance cost that exceeds the benefit of adaptation [9]. On the other hand, failing to highlight an important menu item might not bring any benefit to the user but induces no cognitive cost either. Thus, when emphasizing content, a successful relevance detector should identify the subset of relevant UI elements (i.e. true positives) while minimizing the risk to detect irrelevant ones as important (i.e. false positives). When suppressing content, on the other hand, we are interested in recognizing the non-relevant elements (i.e. true negatives) while avoiding suppressing any relevant ones (i.e. false negatives) which might induce a high cognitive cost.

We can easily tune the different recognition rates of the relevance detector (true/false positive/negative) by varying the number of votes required to recognize an element as being relevant. Different voting schemes are possible. We can require a minimum of 1-6 gaze metrics to cast a vote or we can be more selective and consider votes from metrics of the same stage (see Table 2 as redundant. In this case, we might require votes from a minimum of 2 different or all 3 stages. Figure 12 shows the resulting trade-off between the true positive (relevant elements correctly

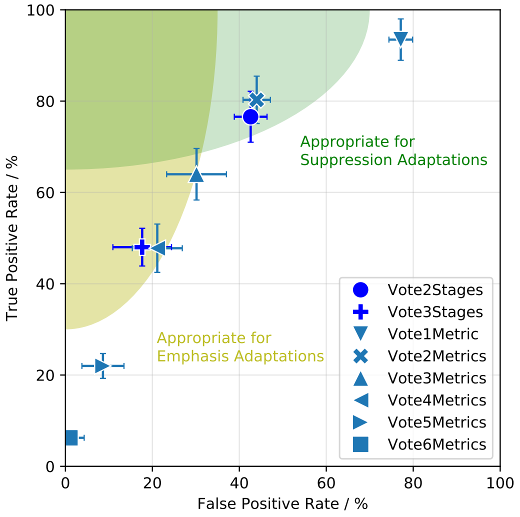


Fig. 12: The number of votes required to detect an element as relevant changes the trade-off between the true positive and false positive rate. This voting approach allows to choose the right trade-off must depending on the adaptation scheme. `VoteXMetrics` denotes a minimum of 1-6 votes. `VoteXStages` refers to a minimum of 2 or 3 votes which must come from different stages [8].

detected) and the false positive rate (irrelevant elements detected as relevant) depending on the voting scheme. The shaded areas indicate rates that seem acceptable for emphasizing or suppressing information. Data comes from an empirical study capturing the gaze behavior of 12 participants during interaction with a financial trading interface with information about a specific stock. Participants should decide whether to invest in the stock or not. Details are given in [8].

The figure shows that the true/false positive rate of the recognizer can easily be adjusted in a predictable manner to account for the requirements of different adaptation schemes. We recommend the following vote thresholds:

- For **emphasizing** relevant information, we recommend a minimum of 3 votes each from a different stage (`Vote3Stages` in Figure 12). This yields a low false positive rate, reducing the risk of inducing any cognitive dissonance by emphasizing irrelevant information. At the same time, it ensures that only the most relevant information is emphasized.
- For **suppressing** irrelevant information, we recommend a minimum of any 2 votes (`Vote2Metrics` in Figure 12). This yields a high true positive rate, ensuring that relevant information is not suppressed in any way, which could lead to higher interaction costs. A high false positive rate is acceptable in this case, which means that some less relevant content is not suppressed.

6 Discussion and Outlook

The improvements in AI have given rise to a new class of learning-based gaze estimation methods which make eye tracking more practicable and more widely applicable in everyday computer interaction. In contrast to traditional gaze estimation methods, the recently developed learning-based approaches do not require specialized hardware and can operate with just a single webcam and at a much larger operation distance. As these methods further improve, they will allow for HCI applications to consequently use gaze outside the lab and in the everyday interaction with computers.

Two major challenges remain open in enabling out-of-the-box learning-based gaze estimation solutions. One is in improving the generalization of models to previously unseen users, environments, eyeglasses, makeup, camera specifications and other confounding factors. This can be tackled by the non-trivial task of collecting datasets with high-quality ground-truth annotations from a large number of people [60, 36] and designing novel neural network architectures for better generalization [55, 33] - both directions which we have extensively studied. The other challenge is due to person-specific biases, which must be accounted for when higher performance is required by the interactive application. This challenge exists not only because of the kappa angle but also the variations in the appearance of the eye and face region in the real-world. While we have explored several methods to this end in terms of few-shot adaptation [59, 35], further research must be conducted to efficiently collect data from the end-user without compromising user experience, such as via so-called implicit calibration [57].

A problem in developing gaze-based interfaces is that the accuracy and precision of the tracked gaze vary largely depending on many factors, such as the tracking method, the environment, human features, and others. The application receiving the gaze information must process a series of noisy data points. We have shown how, to some extent, a signal can be stabilized by filtering data. However, this does not account for its inaccuracy. For that, we have made recommendations for designing gaze-aware applications in a robust way such that they are usable under most conditions [7]. However, such a conservative approach might unnecessarily slow down or complicate interaction in cases where the gaze is tracked well. An alternative approach is to develop *error-aware applications* that recognize the uncertainty in the signal and adapt to it [1, 7]. As tracking quality decreases, a gaze-aware UI element could be enlarged, replaced by a more robust alternative, or deactivated entirely to avoid errors that might be hard to recover from. For such an approach to be useful, it is crucial to optimize for the time-point of UI adaptation. To this end, future work is needed that investigates how to trade-off potential gains through adaptation with the cost for the user to get used to a new interface. For taking into account personal preferences, such adaptations could even be done after explicitly querying the user.

We have seen that data about where a user is looking cannot only be used for explicit interaction but also to make predictions about the user's cognitive processes, abilities, or intentions. Such attentive applications do not require the user to consciously control their gaze which can be cumbersome. Instead, they process the natural gaze behavior of the user with the goal to facilitate interaction. However, approaches to interpreting the eye gaze are often tailored to specific application cases and general solutions are rare. The voting-scheme presented in this chapter (Section 5.2) is a first attempt to develop a more general approach for estimating the relevance of displayed information to a specific user and was shown to work across different decision-making tasks [8]. More work is needed though to develop general methods for inferring a user's intent, difficulties, or preferences from their gaze data and thus facilitate the design of intelligent user interfaces.

Once we can reliably derive information about the user's attention and intention from the estimated gaze, it is important to consider how to make effective use of this data in practice. In a user study conducted in [8], the large majority of participants confirmed that the tested application could correctly detect content relevant for their decision making. Many also preferred the adapted version of the interface. However, the specific highlighting and suppression adaptations (see Figure 11) did not lead to measurable improvements in terms of task execution time, users' perceived information load, or their confidence in their decision. Future work needs to develop better approaches to utilize such relevant information and develop UI adaptation schemes that facilitate the decision-making process for the user [14, 26]. Such work should also consider how adaptive interfaces can build trust to resolve users' concern of being manipulated by the interface [8, 32].

7 Conclusion

The advancement of AI techniques is boosting gaze estimation to become one of the major interactive signals for modern human-computer interaction. New learning-based methods have been developed for appearance-based, model-based and hybrid gaze estimation methods. In particular, these learning-based methods can work with just a single webcam under challenging lighting conditions even over long operating distances of up to 2 meters. The gaze estimation error is maintained to about 4° without personal calibration and 2° with personal calibration under variant challenging conditions. However, learning-based methods rely on large and varied datasets of different conditions and devices. Therefore, multiple datasets have been proposed that capture variations in head poses, gaze directions, lighting conditions, personal appearances, or input image resolutions. Although there is still a gap between methods using a single webcam and those with dedicated hardware, the presented research indicates promising efforts yield a performance that is close to that of traditional methods.

One of the key issues of the gaze estimation task is that collecting personal calibration data is tedious for the user. There are two ways to tackle this issue. the first one is efficiently using the personal calibration data with few-shot learning or synthetically generate more training samples with few calibration images. The second effective way is carefully designing user interfaces (UI) that take into account this potential noise and thus the uncertainty of the input signal. We proposed actionable design guidelines for gaze-enabled applications including appropriate target sizes, target dimensions, visual representations, placement and optimal parameter settings for different gaze filters. In addition, we introduced the OpenGaze and GazeML open-source toolkits which make the entire gaze estimation pipeline easily accessible to HCI researchers.

We presented two examples that show how the use of implicit gaze information can enable entirely new interactive concepts. Gaze-aware real-life objects can recognize when a user is looking at them without any specific camera-object or user calibration. In the second case, we showed that the user's gaze behaviour can reveal whether displayed content is useful and relevant to the current task of a user. Such information can be used for example to adapt the user interface accordingly. Both examples work without requiring the user to explicitly control their eye gaze but analyze their natural gaze behaviour during interaction.

In summary, AI-inspired methods have revolutionized approaches for estimating where a person is looking at on a screen or in the 3D world. Already now, learning-based approaches enable sufficiently good gaze estimation in many real-world environments with just a single camera, bringing eye-tracking out of the lab and into our everyday interaction with computers. This makes gaze not only a viable input method in situations where keyboard or touch input is not available or not feasible but also opens the doors for entirely new interactions and applications that can take into account gaze as an additional information source about the user's state.

References

[1] Barz M, Daiber F, Sonntag D, Bulling A (2018) Error-aware gaze-based interfaces for robust mobile gaze interaction. In: Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, Association for Computing Machinery, New York, NY, USA, ETRA '18, DOI 10.1145/3204493.3204536, URL `https://doi.org/10.1145/3204493.3204536`

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

[3] Bulling A (2016) Pervasive attentive user interfaces. IEEE Computer 49(1):94–98

[4] Chen Z, Shi B (2020) Offset calibration for appearance-based gaze estimation via gaze decomposition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)

[5] Cheng Y, Zhang X, Lu F, Sato Y (2020) Gaze estimation by exploring two-eye asymmetry. IEEE Transactions on Image Processing 29:5259–5272

[6] Eckstein MK, Guerra-Carrillo B, Singley ATM, Bunge SA (2017) Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development? Developmental cognitive neuroscience 25:69–91

[7] Feit AM, Williams S, Toledo A, Paradiso A, Kulkarni H, Kane S, Morris MR (2017) Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In: Proceedings of the 2017 CHI Conference on

Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA, CHI '17, p 1118–1130, DOI 10.1145/3025453.3025599, URL https://doi.org/10.1145/3025453.3025599

[8] Feit AM, Vordemann L, Park S, Berube C, Hilliges O (2020) Detecting relevance during decision-making from eye movements for ui adaptation. In: ACM Symposium on Eye Tracking Research and Applications, Association for Computing Machinery, New York, NY, USA, ETRA '20 Full Papers, DOI 10.1145/3379155.3391321, URL https://doi.org/10.1145/3379155.3391321

[9] Findlater L, Gajos KZ (2009) Design space and evaluation challenges of adaptive graphical user interfaces. AI Magazine 30(4):68–73, DOI 10.1609/aimag.v30i4.2268

[10] Fischer T, Jin Chang H, Demiris Y (2018) Rt-gene: Real-time eye gaze estimation in natural environments. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 334–352

[11] Fuhl W, Santini T, Kasneci G, Rosenstiel W, Kasneci E (2017) Pupilnet v2. 0: Convolutional neural networks for cpu based real time robust pupil detection. arXiv preprint arXiv:171100112

[12] Funes Mora KA, Monay F, Odobez JM (2014) Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In: Proceedings of the Symposium on Eye Tracking Research and Applications, pp 255–258

[13] Ganin Y, Kononenko D, Sungatullina D, Lempitsky V (2016) Deepwarp: Photorealistic image resynthesis for gaze manipulation. In: European conference on computer vision, Springer, pp 311–326

[14] Gebhardt C, Hecox B, van Opheusden B, Wigdor D, Hillis J, Hilliges O, Benko H (2019) Learning cooperative personalized policies from gaze data. In: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, Association for Computing Machinery, New York, NY, USA, UIST '19, p 197–208, DOI 10.1145/3332165.3347933, URL https://doi.org/10.1145/3332165.3347933

[15] Gidlöf K, Wallin A, Dewhurst R, Holmqvist K (2013) Using Eye Tracking to Trace a Cognitive Process: Gaze Behaviour During Decision Making in a Natural Environment. Journal of Eye Movement Research 6(1), DOI https://doi.org/10.16910/jemr.6.1.3, URL https://bop.unibe.ch/index.php/JEMR/article/view/2351

[16] Hansen DW, Ji Q (2009) In the eye of the beholder: A survey of models for eyes and gaze. IEEE transactions on pattern analysis and machine intelligence 32(3):478–500

[17] He J, Pham K, Valliappan N, Xu P, Roberts C, Lagun D, Navalpakkam V (2019) On-device few-shot personalization for real-time gaze estimation. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp 0–0

[18] He Z, Spurr A, Zhang X, Hilliges O (2019) Photo-realistic monocular gaze redirection using generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp 6932–6941

[19] Hirzle T, Gugenheimer J, Geiselhart F, Bulling A, Rukzio E (2019) A design space for gaze interaction on head-mounted displays. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA, CHI '19, p 1–12, DOI 10.1145/3290605.3300855, URL https://doi.org/10.1145/3290605.3300855

[20] Howard IP, Rogers BJ, et al. (1995) Binocular vision and stereopsis. Oxford University Press, USA

[21] Kellnhofer P, Recasens A, Stent S, Matusik W, Torralba A (2019) Gaze360: Physically unconstrained gaze estimation in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp 6912–6921

[22] Khamis M, Oechsner C, Alt F, Bulling A (2018) Vrpursuits: Interaction in virtual reality using smooth pursuit eye movements. In: Proceedings of the 2018 International Conference on Advanced Visual Interfaces, Association for Computing Machinery, New York, NY, USA, AVI '18, DOI 10.1145/3206505.3206522, URL https://doi.org/10.1145/3206505.3206522

[23] Kim J, Stengel M, Majercik A, De Mello S, Dunn D, Laine S, McGuire M, Luebke D (2019) Nvgaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp 1–12

[24] Krafka K, Khosla A, Kellnhofer P, Kannan H, Bhandarkar S, Matusik W, Torralba A (2016) Eye tracking for everyone. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2176–2184

[25] Lindén E, Sjostrand J, Proutiere A (2019) Learning to personalize in appearance-based gaze tracking. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp 0–0

[26] Lindlbauer D, Feit AM, Hilliges O (2019) Context-aware online adaptation of mixed reality interfaces. In: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, pp 147–160

[27] Majaranta P (2011) Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies. IGI Global

[28] Majaranta P, Bulling A (2014) Eye Tracking and Eye-Based Human–Computer Interaction, Springer London, London, pp 39–65. DOI 10.1007/978-1-4471-6392-3_3, URL https://doi.org/10.1007/978-1-4471-6392-3_3

[29] Moshirfar M, Hoggan RN, Muthappan V (2013) Angle kappa and its importance in refractive surgery. Oman journal of ophthalmology 6(3):151

[30] Orquin JL, Loose SM (2013) Attention and choice: A review on eye movements in decision making. ACTPSY 144:190–206, DOI 10.1016/j.actpsy.2013.06.003, URL http://dx.doi.org/10.1016/j.actpsy.2013.06.003

[31] Papismedov D, Fink L (2019) Do Consumers Make Less Accurate Decisions When They Use Mobiles? In: International Conference on Information Systems, Munich

[32] Park S, Gebhardt C, Rädle R, Feit A, Vrzakova H, Dayama N, Yeo HS, Klokmose C, Quigley A, Oulasvirta A, Hilliges O (2018) AdaM: Adapting Multi-

User Interfaces for Collaborative Environments in Real-Time. In: SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, CHI '18

[33] Park S, Spurr A, Hilliges O (2018) Deep pictorial gaze estimation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 721–738

[34] Park S, Zhang X, Bulling A, Hilliges O (2018) Learning to find eye region landmarks for remote gaze estimation in unconstrained settings. In: Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, pp 1–10

[35] Park S, Mello SD, Molchanov P, Iqbal U, Hilliges O, Kautz J (2019) Few-shot adaptive gaze estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp 9368–9377

[36] Park S, Aksan E, Zhang X, Hilliges O (2020) Towards end-to-end video-based eye-tracking. In: European Conference on Computer Vision, Springer, pp 747–763

[37] Qvarfordt P, Zhai S (2005) Conversing with the user based on eye-gaze patterns. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA, CHI '05, p 221–230, DOI 10.1145/1054972.1055004, URL https://doi.org/10.1145/1054972.1055004

[38] Russo JE, Leclerc F (1994) An Eye-Fixation Analysis of Choice Processes for Consumer Nondurables. Journal of Consumer Research 21(2):274–290, DOI 10.1086/209397, URL https://doi.org/10.1086/209397, https://academic.oup.com/jcr/article-pdf/21/2/274/5093700/21-2-274.pdf

[39] Salvucci DD (2001) An integrated model of eye movements and visual encoding. Journal of Cognitive Systems Research 1:201–220, URL www.elsevier.com/locate/cogsys

[40] Salvucci DD, Goldberg JH (2000) Identifying fixations and saccades in eye-tracking protocols. In: Proceedings of the 2000 symposium on Eye tracking research & applications, pp 71–78

[41] Sesma L, Villanueva A, Cabeza R (2012) Evaluation of pupil center-eye corner vector for gaze estimation using a web cam. In: Proceedings of the symposium on eye tracking research and applications, pp 217–220

[42] Sibert LE, Jacob RJ (2000) Evaluation of eye gaze interaction. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp 281–288

[43] Sugano Y, Matsushita Y, Sato Y (2014) Learning-by-synthesis for appearance-based 3d gaze estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1821–1828

[44] Tan KH, Kriegman DJ, Ahuja N (2002) Appearance-based eye gaze estimation. In: Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings., IEEE, pp 191–195

[45] Špakov O (2012) Comparison of eye movement filters used in hci. In: Proceedings of the Symposium on Eye Tracking Research and Applications,

Association for Computing Machinery, New York, NY, USA, ETRA ’12, p 281–284, DOI 10.1145/2168556.2168616, URL https://doi.org/10.1145/2168556.2168616

[46] Wang K, Ji Q (2017) Real time eye gaze tracking with 3d deformable eye-face model. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV)

[47] Wang K, Zhao R, Ji Q (2018) A hierarchical generative model for eye image synthesis and eye gaze estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 440–448

[48] Wood E, Bulling A (2014) Eyetab: Model-based gaze estimation on unmodified tablet computers. In: Proceedings of the Symposium on Eye Tracking Research and Applications, pp 207–210

[49] Wood E, Baltrusaitis T, Zhang X, Sugano Y, Robinson P, Bulling A (2015) Rendering of eyes for eye-shape registration and gaze estimation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV)

[50] Yu Y, Liu G, Odobez JM (2018) Deep multitask gaze estimation with a constrained landmark-gaze model. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 0–0

[51] Yu Y, Liu G, Odobez JM (2019) Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 11937–11946

[52] Zhai S, Morimoto C, Ihde S (1999) Manual and gaze input cascaded (magic) pointing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA, CHI ’99, p 246–253, DOI 10.1145/302979.303053, URL https://doi.org/10.1145/302979.303053

[53] Zhang X, Sugano Y, Fritz M, Bulling A (2015) Appearance-based gaze estimation in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4511–4520

[54] Zhang X, Sugano Y, Bulling A (2017) Everyday eye contact detection using unsupervised gaze target discovery. In: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, pp 193–203

[55] Zhang X, Sugano Y, Fritz M, Bulling A (2017) It’s written all over your face: Full-face appearance-based gaze estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 51–60

[56] Zhang X, Sugano Y, Fritz M, Bulling A (2017) Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. IEEE transactions on pattern analysis and machine intelligence 41(1):162–175

[57] Zhang X, Huang MX, Sugano Y, Bulling A (2018) Training person-specific gaze estimators from user interactions with multiple devices. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp 1–12

[58] Zhang X, Sugano Y, Bulling A (2018) Revisiting data normalization for appearance-based gaze estimation. In: Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, pp 1–9

[59] Zhang X, Sugano Y, Bulling A (2019) Evaluation of appearance-based methods and implications for gaze-based applications. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp 1–13

[60] Zhang X, Park S, Beeler T, Bradley D, Tang S, Hilliges O (2020) Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In: European Conference on Computer Vision, Springer, pp 365–381

[61] Zhang X, Sugano Y, Bulling A, Hilliges O (2020) Learning-based region selection for end-to-end gaze estimation. In: British Machine Vision Virtual Conference (BMVC)

[62] Zheng Y, Park S, Zhang X, De Mello S, Hilliges O (2020) Self-learning transformations for improving gaze and head redirection. Advances in Neural Information Processing Systems 33

[63] Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Computer Vision (ICCV), 2017 IEEE International Conference on