Quality and User Experience (2019) 4:1
https://doi.org/10.1007/s41233-019-0024-6

RESEARCH ARTICLE

# CrowdWatcher: an open-source platform to catch the eye of the crowd

**Pierre Lebreton[1] · Isabelle Hupont[2] · Matthias Hirth[3] · Toni Mäki[4] · Evangelos Skodras[5] · Anton Schubert[6] · Alexander Raake[6]**

Received: 20 November 2018 / Published online: 12 March 2019
© Springer Nature Switzerland AG 2019

**Abstract**
This paper presents the open-source eye tracking platform *CrowdWatcher*. It enables researchers to measure gaze location and user engagement in a crowdsourcing context through traditional RGB webcams. The proposed platform particularly advances the field of Quality of Experience (QoE) research, as it allows the experimenter to collect remotely and with very limited effort novel information from crowds of participants, such as their commitment towards a task, attention and decision-making processes. Two different experiments are described that were conducted to demonstrate the platform's potential. The first experiment addresses the measurement of participants' behavior while performing a movie selection task. Results show that the platform provides complementary information to traditional self-reported data by taking gaze analysis into account. This is of particular relevance, since in a crowdsourcing context decision processes and attention are difficult to assess, and there is often limited control over the engagement of the test user with the task. A second experiment is conducted in the scenario of a multimedia QoE test. Prediction accuracy is compared to a professional infrared eye tracker. While *CrowdWatcher* performs less well than the professional eye tracker, it is still able to collect valuable gaze information in the far more challenging environment of crowdsourcing. As an outlook to further application domains, the usage of the platform to measure user engagement allows participants who do not pay attention to the task to be identified.

**Keywords** Eye tracking · Crowdsourcing · Engagement · User behavior · Quality of experience · Human–computer interaction

## Introduction

During daily activities and tasks, humans are faced with considerably more visual information than they are able to process. All throughout the day, we scan our environment targeting things like faces, texts, images on screens or product packages and various other objects. Selective visual attention mechanisms, both with explicit focusing via eye movements (*overt*) and without explicit focusing of the eye (*covert*), allow us to deal with this vast amount of

✉ Pierre Lebreton
lebreton@zju.edu.cn

Isabelle Hupont
hupont@isir.upmc.fr

Matthias Hirth
matthias.hirth@tu-ilmenau.de

Toni Mäki
toni.j.maki@aalto.fi

Evangelos Skodras
evskodras@upatras.gr

Anton Schubert
anton.schubert@gmx.de

Alexander Raake
alexander.raake@tu-ilmenau.de

[1] Group of Networked Sensing and Control (NeSC), Zhejiang University, Hangzhou, China

[2] Institute of Intelligent Systems and Robotics, Sorbonne University, Paris, France

[3] User-Centric Analysis of Multimedia Data Group, Technische Universität Ilmenau, Ilmenau, Germany

[4] Department of Computer Science, Aalto University, Espoo, Finland

[5] Department of Electrical and Computer Engineering, University of Patras, Patras, Greece

[6] Audio Visual Technology Group, Technische Universität Ilmenau, Ilmenau, Germany

information by prioritizing some aspects of the scene while ignoring others [8].

Such mechanisms have been widely studied in the field of Quality and User Experience, as they provide valuable insights on the subject's experience regarding multimedia contents such as images, videos, websites, etc. Due to the relevance of visual attention, a number of studies have been reported on the relation of eye movements with quality evaluation tasks [51], the visibility of coding or slicing impairments [2, 18, 19, 79], and their application to enhance video quality metrics [17, 44, 49]. In addition, eye movements were also studied in other fields such as the perception of aesthetics [59] or web design [30, 43]. As *covert* attention is not directly measurable because it involves the attentional spotlight of our minds without deploying the eyes, *overt* attention is typically addressed when it comes to quantitatively measuring visual attention. It is the gaze point that shows the visual targeting that takes place, and hence represents the fundamental information collected during eye tracking.

In parallel, the use of crowdsourcing in QoE evaluation tasks has evolved as an important growing field [16]. On the one hand, experiments are not conducted in a laboratory environment, but at the test participants' current locations using their own equipment. This provides the advantage to perform a subjective evaluation in conditions closer to the users' daily environment. It enables taking into account a large variety of technical factors such as displays, Internet access speeds, devices (smartphones, computers, tablets), environmental conditions (room lighting, participants's viewing distance, ambient sound, etc.) or locations (at home, in public places) [34, 35, 39]. It also allows reaching a larger crowd both in terms of the number of users and in terms of background (cultural differences, knowledge of subjective quality tests, attitude towards technology, etc.) [48]. On the other hand, crowdsourcing brings new challenges and paradigms regarding experimental designs and data screening [25, 35]. For example, different kinds of noise can be induced in the collected data due to the differences in test conditions that may vary across different remote users or the misunderstanding of the task, as it becomes more difficult to interact with participants who only rely on written instructions. The question of engagement into the task is also raised, as participants tend to optimize the usage of their time and may not necessarily pay full attention to the study [24, 32].

Introducing gaze analysis into crowdsourcing studies could open new opportunities. It would benefit visual attention research by allowing to easily address a larger number of users, working in very different test conditions, and to evaluate a wider variety of contents (images, videos, websites) with limited effort. Gaze tracking could bring a powerful source of implicit feedback beyond typical crowdsourcing channels (e.g. clicking, scrolling, typing) and reveal a great deal of a person's cognitive processes [68]. It would provide novel insights about engagement issues: are participants paying attention to the task, are they busy with different tasks performed in parallel, etc.?

Combining crowdsourcing and gaze analysis is not easy. The emergence of accurate eye trackers in the early 2000s opened the door to the reliable exploration of visual attention. However, traditional eye tracking relies on high-priced, professional dedicated equipment, such as infrared cameras or head-mounted devices working at 60–120 Hz, and implies a prior calibration procedure. Therefore acquiring gaze data from a large number of users is complex and time-consuming. Moreover, the use of these accurate eye trackers is impractical in remote crowdsourcing contexts, as not possessed by participants or cannot be lent to them [4].

This paper proposes a solution to obtain novel and extended measurements about participants' gaze behavior and engagement in a crowdsourcing context. A web-based eye tracking platform called *CrowdWatcher* is presented. It is provided open-source [12], enabling the community to use it and obtain valuable information on participants, such as their engagement with a task, coarse gaze location measurements, and deeper knowledge of user decision-making processes. The underlying eye tracking algorithm only requires a typical RGB webcam to run, such as the one embedded in most commercial laptop/tablets. A particular innovation lies in the proposed calibration procedure, which is non-invasive and transparent to the user. Two crowdsourcing experiments are carried out to demonstrate the potential of the platform. A preliminary test illustrates the feasibility and suitability of the proposed approach. Then, a second study presents an in-depth evaluation of the eye tracking capabilities of the platform and compares it to a in-lab professional eye tracker.

The paper is organized as follows. "Related work" section reviews previous studies aiming at measuring gaze and aspects of visual attention in a crowdsourcing context. "Platform description" section provides an overall technical description of the platform. "Application: proof of concept and performance evaluation" section presents the two experiments carried out using the platform and corresponding results. Finally, "Conclusion" section concludes the paper.

## Related work

Moving traditional eye tracking user studies from the laboratory to the "wild" of crowdsourcing implies many challenges. From a technical point of view, the subject's pupils have to robustly be detected from a standard RGB webcam without any control over scene conditions (room illumination, head pose, background, etc.). From the User Experience perspective, the main challenge is to keep the participant

engaged and focused on the task during the whole experiment. In the following, different related developments from the literature are reviewed in light of the platform presented in this paper.

## Alternatives to eye tracking for crowdsourcing

Many existing crowdsourcing studies prefer low-cost and easy-to-implement alternatives to eye tracking for evaluating visual attention. The most popular approach considers that the usage of the mouse by a participant is directly related to gaze in the case of web page navigation. Several studies have demonstrated that mouse tracking feedback can be very close to eye tracking for certain tasks [36, 46, 56]. For example, findings in [62] showed that when a user clicks on an interface element, he is looking at this element.

Self-reporting has also been used as a simple solution to gather information on visual attention from the crowd. For example, in [10, 63] users had to look at video clips for a few seconds and, immediately after the video ended, were shown a labeled grid at the position where the video was initially displayed. They were then asked to specify the grid label corresponding to the region of the video they had seen most clearly. A different type of self-reporting approach was proposed in [29, 40, 61]. They performed crowdsourced experiments in which participants were presented with images they were asked to describe during the test. Each image was blurred so that the participant needed to click to reveal "bubbles" (small circular areas at normal resolution). Because the image was uncovered tile-by-tile, the game mechanics allowed to collect information on the image segments that are most important to identify the image content.

Although mouse usage and self-reports were shown to be comparable to eye tracking data, they suffer from a number of drawbacks. On the one hand, these methods are intrusive and imply an increase in cognitive load that could influence participants' responses. On the other hand, they fail to track gaze trajectories over the whole period of time when the stimuli are displayed.

## Crowdsourcing gaze through RGB cameras

Most eye tracking techniques rely on infrared (IR) light sources and cameras. High-accuracy eye trackers generally depend on high-priced, large and invasive hardware, such as specific displays or glasses with embedded IR cameras. Well-known examples include Tobii [72], EyeLink [20] or Gazepoint [27] eye trackers.

In recent years, cheaper—yet less accurate—solutions have become feasible with reduced-size IR equipment sold below $300. Some examples are *The Eye Tribe eye tracker* [21], the *ITU Gaze Tracker* [37] or *Pupil* [58]. However, such IR hardware cannot readily be used in crowdsourcing contexts, since it is not embedded in devices such as laptops/tablets by default, and has to be used with specific software.

Visible light (VL) gaze tracking does not require special hardware and aims to solve the task by means of standard RGB cameras. However, several factors make VL eye tracking a far more challenging task than IR eye tracking. The most important aspect is related to the accuracy of the pupil detection and the influence of head movements. IR illumination creates much more contrast around pupil contours, which can be tracked with high accuracy and makes eye tracking far less susceptible to head movements. VL gaze tracking requires the use of more sophisticated computer vision algorithms. An exhaustive state of the art on VL pupil detection techniques can be found in [22, 31]. In this section, we rather focus on design considerations that must be undertaken to obtain accurate and long-term VL gaze tracking in a crowdsourcing context.

The main objective in this context is to keep the user engaged and focused throughout the task, while maintaining an accurate gaze estimation over time. To achieve it, a good design of the task (e.g. in the form of instructions given to the participants) may help to ensure that the user's head is as stable, well illuminated and frontal as possible. Also, attention has to be paid to calibration, the key delicate procedure that allows matching pupil and screen positions. Even small head movements may cause large errors in the estimations of a calibrated tracker. Therefore efficient yet non-disturbing calibration and recalibration mechanisms must be established to guarantee the long-term success of eye tracking.

Traditional approaches for the calibration of eye trackers are based on explicitly asking the subject to look at several targets in known screen positions before starting the task. Applied to crowdsourcing, [78] maintain this philosophy in their platform, where the participant is asked to look at specific locations on the screen to perform calibration. Once calibrated, the gaze evaluation of an image is performed. This approach has, however, the drawback that it requires intrusive recalibration between images due to head movements, which implies participant flow of attention to be frequently interrupted.

Several recent works propose alternatives to make calibration process easier, less intrusive and more suitable for crowdsourcing. The approach in [3] is based on the similarity of human gaze patterns, and makes use of other users' gaze patterns to auto-calibrate the current user's gaze. Another approach to collect calibration data in a transparent manner is to let participants operate the computer normally and take calibration samples during mouse clicks. This method is grounded on the assumption that the user looks to the mouse pointer while clicking. In [71] the authors used this strategy for a web navigation task, achieving promising

results. Nevertheless, their task implied a number of clicks per participant ranging from 600 to 1300. Other kinds of tasks requiring fewer mouse interactions have not been explored.

## Commercial and open-source gaze tracking platforms

While VL gaze tracking has become a hot topic in academia, the industry is not trailing far behind either [22]. Several commercial solutions for VL gaze tracking are available. For instance, *GazeHawk* [26] and *Sticky* [69] deploy JavaScript-based services enabling their customers to convey remote eye tracking studies inside the user's browser. Using these systems, gaze data can be recorded on the test participants' computers and then be uploaded to a study server for the generation of analytic reports and visualizations.

Another popular business model is in the form of Software Development Kits (SDKs) to be used in third-party applications. Some examples are: *xLabs SDK* [77], which is also available as a Chrome extension; *SentiGaze* [50], which provides an SDK for developers targeting the Windows platform; *Face-Track* [74], a C++ SDK that also offers detailed information about the mouth contour, chin pose and eye openness; and *InSight SDK* [67], that combines gaze information with mood, age and gender estimation. Although some of these SDKs could be used to analyze facial videos recorded in crowdsourcing experiments, they have been more oriented to other kinds of applications requiring real-time gaze interaction (e.g. for video gaming or as an accessibility tool for disabled users).

Only few works on gaze tracking have released their source code, as it is the case for the framework presented in this paper. Examples include: *OpenGazer* [53], a C++ and Python eye tracker; *NetGazer* [80], the porting of *OpenGazer* for Windows; and *CVC Eye Tracker* [13], a fork of *OpenGazer* actively supported. These frameworks have been created primarily targeting desktop applications and are based on C++. Hence, they are less well suited for crowdsourcing.

Three recent open-source platforms that could be applied in a crowdsourcing context are available on *GitHub*. They are completely implemented in JavaScript, which makes them platform-independent and suitable for gathering gaze information via a web browser. The first one is *CamGaze* [7], which computes binocular gaze estimations and maps them to screen positions by using the information obtained from a prior grid-based calibration procedure. The second is *TurkerGaze* [57, 78], a library presenting an interface for calibration and verification that comes with a small application for analyzing the gaze patterns recorded during a given experiment. It is conceived for one particular task: exploring users' gaze patterns while watching still images. It has the drawback of asking for constant recalibration between two images, which makes the process intrusive for the user. The most recent platform is *WebGazer* [54, 75]. Its eye tracking model self-calibrates by watching web page visitors' interactions (clicks and cursor movements), and it can be integrated into any website by adding only a few lines of JavaScript code. However, it has been validated only for one use case: dynamic web browsing. It is therefore unclear if the system could still remain accurate for tasks that do not require constant user interaction, such as video watching.

## Measuring engagement

It is important to highlight that none of the existing eye tracking platforms presented in "Commercial and open-source gaze tracking platforms" section provides mechanisms to measure to what extent the user has been engaged with the task. They make the assumption that the participant will be looking at the screen and focused all throughout the task, which is difficult to ensure in real crowdsourcing environments.

Engagement is of the utmost importance in Human–Computer Interaction [55]. The extensive review of engagement definitions presented in [28] discusses a set of concepts that are strongly related to engagement and sometimes even used interchangeably. These concepts include attention, involvement, interest and stance. Studying the degree of involvement of participants with respect to the focus of a given task is also essential in crowdsourcing [14].

Engagement assessment is a recent field which has been mostly studied in laboratory environments and in the context of human–agent interaction. It has generally been tackled through verbal self-reports, where participants themselves or external annotators are asked to judge the degrees of attention, boredom, enjoyment or distraction [65]. Although self-reports provide first-hand information, they are not suitable for crowdsourcing, as they are time consuming, participants may cheat to be paid and they may divert their attention during the task. Some techniques, referred to as "honeypots", have been developed in crowdsourcing to guard against distracted or low performing participants. Examples include asking explicitly verifiable questions to reduce invalid responses, or measuring timings related to task completion [47]. However, it is difficult to apply these techniques to the case of eye tracking.

Previous work has supported the idea of deriving the field of attention from head pose [70], which can be automatically extracted by means of computer vision techniques [15]. The study of head position across time allows to capture certain gestures, such as tilts and large head movements, which may appear when someone is distracted or bored, allowing addressing user's engagement and focus on the task [33, 64].

However, this idea has not been brought to crowdsourcing yet.

### Contributions

The framework described in this paper results from a longer-term research (first proof of concept published in Feb. 2015 [42]), and has several key contributions compared to state-of-the-art platforms. Regarding VL measurement of gaze locations, *CrowdWatcher* differs from most previous works as, except for *xLabs* and *WebGazer*, it does not require a standard calibration phase and allows continuous non-intrusive recalibration along the test based on the user's actions. Also, the proposed approach brings new quality measurements not addressed by other frameworks. It allows monitoring and providing on-line feedback to the user about the test conditions to ensure reliable data, and provides confidence intervals along with gaze predictions, allowing the experimenter to be informed on the quality of measurements. Finally, to the best of our knowledge, it is the first crowdsourcing platform able to perform non-intrusive vision-based evaluation of participants' engagement.

## Platform description

This section describes the *CrowdWatcher* platform. It is based on the use of conventional RGB cameras, such as the one provided with a laptop, to determine where participants are looking on the screen. One of the key aspects of the platform is to use the interactions of the user with the computer as a way to perform an online calibration of the eye tracker. Indeed, it is expected that when a participant performs an action such as clicking on an item, he will be looking at the position on the screen where the click was performed. Therefore, at the very moment of the click, it is possible to relate pupils' position to a position on the screen. Based on this principle, a browser-based solution was developed enabling performing eye tracking tests in a crowdsourcing context.

### Architecture overview

The platform has two parts: a client side and a server side. The client side employs WebRTC[1] to turn on the webcam of the participant and record the face while performing the task. In parallel, it also records the actions of the user with the platform. These actions include clicks while filling forms, or while performing different kinds of tasks such as gaming, interacting with a video player, or manipulating graphical elements on web pages (buttons, lists, sliders, etc.). After agreement from the participant, all this information is transmitted to the server. The server side of the platform is in charge of delivering the test contents to the client, retrieving information from the client side, and performing the estimation of gaze locations based on the webcam video stream and user action logs.

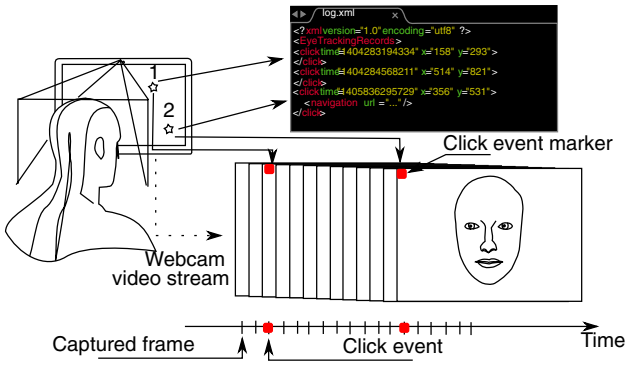


**Fig. 1** *CrowdWatcher*'s client side. The face of the user is recorded along with actions. Two different clicks (1) and (2) are performed. A red marker is then added to the video stream, enabling to synchronize timestamps in the logs and video frames

#### Client side

From the client point of view, several aspects are of interest to enable high-quality data.

*Temporal alignment of time series* The main goal of the client side is to record participants' face and actions. However, facial video and participants' actions are recorded separately. To temporally align the data, every time the participant performs a click, a timestamped click position is logged into an XML file and a red marker is added to the corresponding frame of the camera video stream (see Fig. 1).

*Calibration data points* Contrary to other popular platforms, *CrowdWatcher* performs continuous and non-intrusive recalibration along the test. Calibration data points are collected every time the participant clicks using the mouse. Actions are recorded and used for training the gaze tracking algorithm.

*Online screening* An important novel aspect of the platform is the monitoring of test conditions to ensure reliable data. Participant's lighting conditions and distance to the camera must be checked in order to guarantee the success of the pupil center extraction task. Firstly, a frontal face detector based on the Viola and Jones Haar Cascade algorithm [73] is applied on the client side from the browser, to determine if the face of the participant is visible from the webcam. Once detected, the bounding box around the participant's face is used to estimate the viewing distance, by computing the ratio between the bounding box's height and the frame height (see top Fig. 2). This allows ensuring

[1] https://webrtc.org Accessed Feb. 2019.

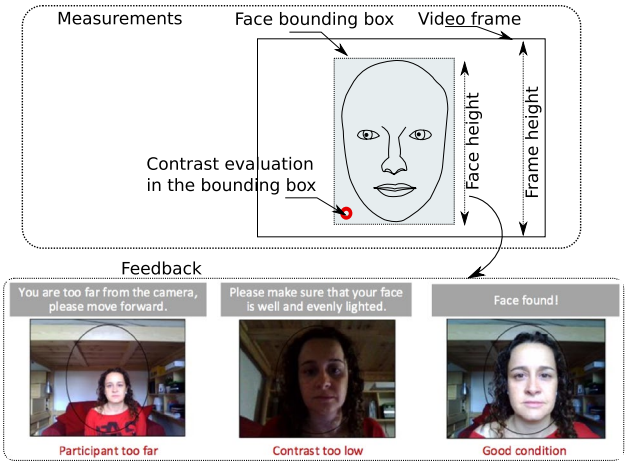


**Fig. 2** Online screening of viewing conditions. User's position and face lighting are checked before the test

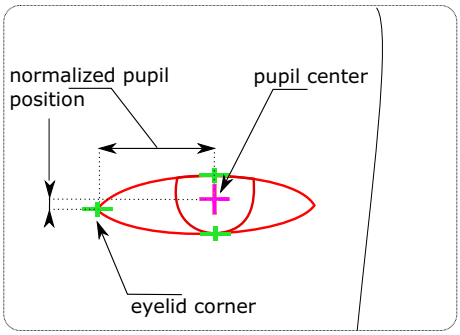


**Fig. 3** Normalization of pupil center position based on stable facial landmarks

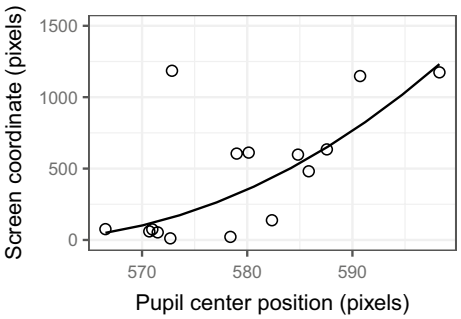


**Fig. 4** Model relating normalized pupil center positions to screen coordinates. Black dots represent pupil/click pairs used to fit the model

that the participant is not seated too far from the camera, which would result in low accuracy in pupil centers estimation. Secondly, the contrast around the face is evaluated to determine whether the pupils can be extracted. To this aim, the no-reference metric proposed by [1] is applied inside the area of the face bounding box. It was chosen as it is computationally light and conceived to estimate contrast. Using this measurement, feedback is provided to the participant with indications on how to adjust position and lighting conditions (c.f. gray boxes in Fig. 2). An oval representing optimal face position is also overlaid on the screen as a guide (see bottom Fig. 2). Participants can move to the next step only after having completed these preliminary checks.

### Server side

This subsection addresses the main different steps performed in *CrowdWatcher*'s server side for predicting gaze locations into screen coordinates, based on click logs and facial recordings from webcams.

*Pupil center extraction* Pupil center estimation on the recorded video stream is performed using the open-source framework *OpenFace* [76]. It allows extracting a set of facial landmarks positions in absolute frame coordinates, including eyelid corners ($P_{ec}$) and pupil center ($P_{abs}$). As stable facial landmarks, eyelid corners are used to normalize the pupil center position and thus to increase the robustness of gaze location predictions due to head movements (Fig. 3). The normalized pupil center position $P$ is computed as:

$$P = P_{abs} - P_{ec} \tag{1}$$

*Calibration* Based on extracted pupil centers and click positions, the eye tracking model is trained. The first step is to perform the alignment of the data using the markers on the videos and click timestamps. Then, a second-order polynomial fit using the RANSAC algorithm [23] is used to relate normalized pupil center positions and click positions. Hence, the eye tracking model allowing to predict gaze location in screen coordinates $S$ from a normalized pupil position $P$ follows the equation:

$$S = a \cdot P^2 + b \cdot P + c \tag{2}$$

where $a$, $b$ and $c$ are coefficients learned from data. Once the eye tracking model is trained, it can be applied to all the video frames (Fig. 4).

*Quality control* Large head movements or bad lighting conditions can result in an inaccurate pupil center detection, and thus in a noisy predicted gaze location. This type of noise is frequent in crowdsourcing environments, and thus it is important to provide quality control mechanisms to experimenters. *CrowdWatcher* provides, along with gaze location predictions, an estimate of their accuracy in the form of Confidence Intervals (CIs). This information is obtained by comparing the predicted value to the actual click position (ground truth) in the frames where clicking events occur. In the process of CI estimation, the accuracy of a given prediction is obtained after removing its corresponding pupil/

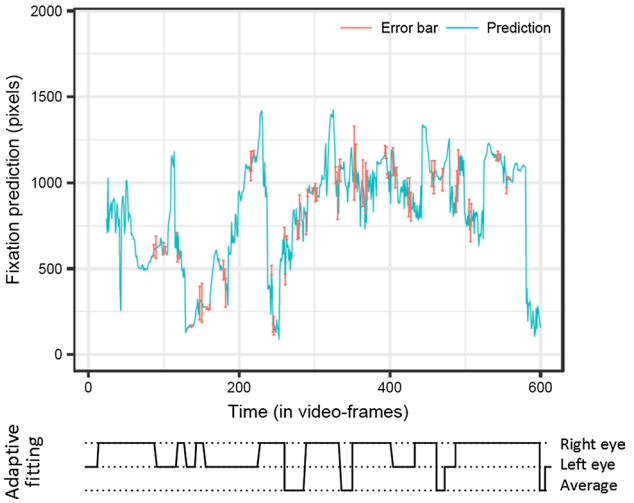


**Fig. 5** Gaze location predictions at every frame with corresponding Confidence Intervals (CIs) when available, i.e. when clicks occur. Real data taken from the first experiment in the horizontal axis are shown as an example. The model selected in the adaptive fitting strategy is also provided, informing about which eye's data was used to perform each prediction

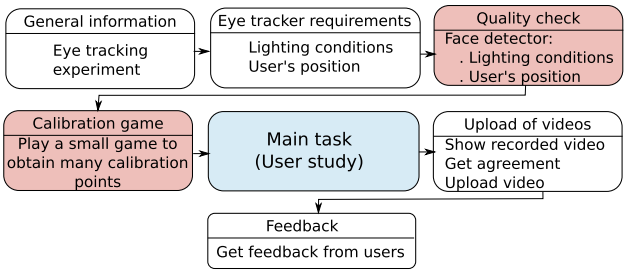


**Fig. 6** General flow of a crowdsourced eye tracking test in *CrowdWatcher*

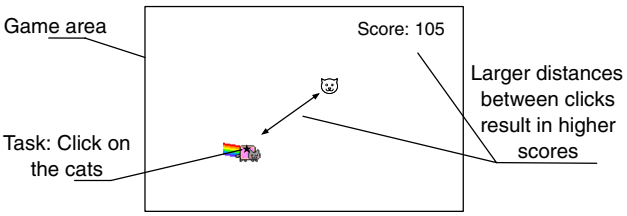


**Fig. 7** Initial calibration game. Participants have to click on as many moving objects as they can

click position pair from the model's training data. Thus, the model is trained using remaining training data, and this trained model is used to predict the current gaze location. This allows to measure the prediction accuracy by comparing it with the ground truth click position, without any overfitting. This process is repeated for each click data, allowing to study the temporal evolution of prediction accuracy. An exemplary result is shown in Fig. 5. It can be seen that, in some cases, the error is found to be large, meaning that gaze location prediction should be taken with care.

*Adaptive fitting* It must be noticed that it is possible to fit an eye tracking model for each of the two eyes. In *CrowdWatcher*, each eye's model is trained independently from the other, and gaze location predictions can be based either on one eye (left or right) or on a combination of the two. The goal of *adaptive fitting* is to find an appropriate pooling strategy to obtain the most reliable prediction at each frame. The most accurate prediction is not always necessarily the one obtained using averaged data from both eyes, as lighting conditions on the face may not be uniform, resulting in different accuracy while extracting each pupil center. Consequently, different pooling strategies are considered and the final gaze location prediction can be obtained: (i) out of one of the models trained on the left or the right eye, or (ii) by averaging the screen positions predicted using both models. To select the best pooling strategy, the quality of the fit for each prediction is studied as described in the previous paragraph (c.f. *quality control*), and the model with the highest quality is chosen. In the case of frames not corresponding to click events, the decision is based on the weighted means of prediction errors made by each model (left eye, right eye or average) at the neighboring frames where click events occurred. The weighted mean takes into account the time difference between the considered frame and these neighboring frames. Following this approach, *CrowdWatcher* automatically identifies for each frame which of the three models is the most suitable to predict gaze location, as illustrated in Fig. 5 (bottom).

## General flow of a test

This section describes the different steps a participant goes through when using the platform. Figure 6 illustrates the general flow of a crowdsourcing test in *CrowdWatcher*. Firstly, general information is provided to participants indicating that an eye-tracking experiment will be carried out, and that it requires to allow the use of the webcam of their computer. Secondly, explanations about how the participant must position himself in front of the camera and adjust illumination are given. Then, the platform performs quality checks to verify both viewing distance and lighting conditions, as described in "Client side" section. It is only after having passed this quality control that participants are allowed to continue through the test. The next step is a calibration game where participants have to click on moving objects on the screen (see Fig. 7). The gamification of this initial calibration procedure ensures that test participants are dedicated to the task and always looking at the mouse pointer when clicking. Moreover, the scoring system favors

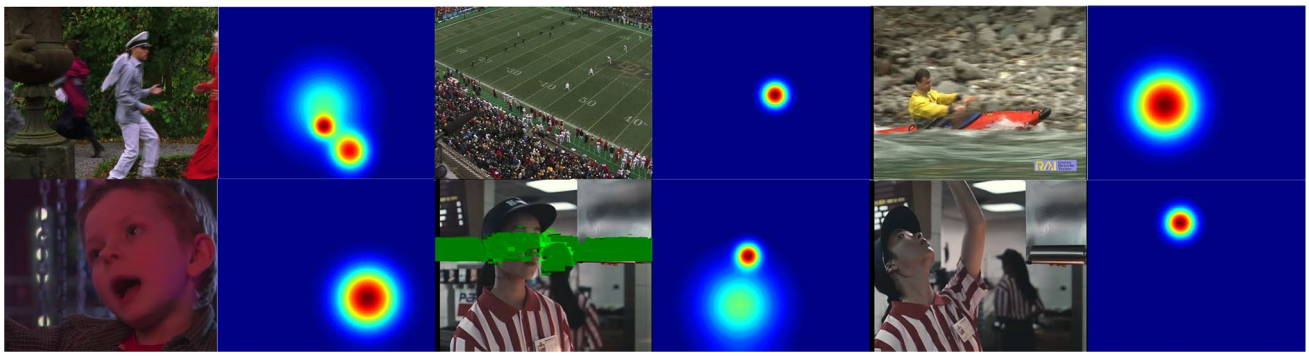

**Fig. 8** Examples of heatmaps that can be automatically obtained from *CrowdWatcher*. They were built from real data collected during our second experiment

large inter-click distances, allowing to collect optimal calibration points (i.e. those with larger inter-click distance). The use of this game is optional, as calibration data can be obtained all throughout the test, but it provides a quick way to collect many initial calibration points for the eye tracker. Once the game is over, the main study can be performed. It can be an image/video quality test, a web-browsing task, a behavior evaluation study, etc. After completion of the study, the recorded video is presented to the participants, letting them decide if they agree to send it to the server. When the video is received on the server side, participants are finally asked to provide feedback on the test.

## Advanced measurements

Beyond gaze location prediction, the platform allows to obtain heatmap visualizations and to measure participants' engagement.

### Heatmaps

In the case of an IR light-based eye tracker, the generation of heatmaps is well known [5]. However, several challenges arise in a crowdsourcing context: firstly, the viewing distance is unknown and not necessarily constant; secondly, gaze location predictions come with an associated uncertainty (CI). To address the first issue, *CrowdWatcher* considers an average viewing distance of 55 cm, based on values found in previous User Experience literature for web navigation tasks[2] [9, 60]. Regarding the second problem, uncertainty is taken into account for the generation of heatmaps. As prediction accuracy is only provided for a limited number of gaze points corresponding to click events, prediction errors are linearly interpolated between known key points. Heatmaps are then generated by summing up the result of the convolution of gaze locations by two 2D Gaussian kernels: one representing the drop-off of visual acuity around the fovea, and one considering prediction accuracy ($CI_i$) at frame $i$. The standard deviation of the Gaussian kernel associated with prediction accuracy was set to $\frac{CI_i}{6}$, 6 being a scaling factor defined empirically. This process can be applied to every participant individually. To remove inter-participant variance, heatmaps are averaged across all participants and normalized to the range [0; 1]. Figure 8 provides examples of heatmap obtained from the data collected in our second experiment ("Performance analysis: application to multimedia QoE" section).

### Engagement

Besides facial landmark positions (c.f. "Server side" section), *OpenFace* also provides, for each video, a timestamped log file containing per-frame values of: yaw-roll-pitch head pose angles, XYZ position of the head center, and an associated 0-to-1 confidence value. *CrowdWatcher* firstly unbiases resulting head angles and positions time series, by using their mean as the bias value and subtracting it from all the samples. Then, the following metrics are computed for each video:

*Percent of face detection* A face is considered to be correctly detected in a given frame when its associated confidence value is above 0.5. This metric computes the percent of frames in the video where the face was detected according to this criterion. Low values of this metric generally mean that the image quality was not good enough to perform face detection (because of bad illumination, low face resolution, face cropped, etc.).

*Percent of attention focused* The approximate immediate field of view of a human eye is in the range of −30° to 30° for yaw and −25° to 30° for pitch [66]. Using these values as thresholds in head pose time series, it can be determined for each frame whether the participant was focusing attention

[2] Note that this average viewing distance is a configurable parameter in *CrowdWatcher*, and it can be easily changed to other values depending on the required test setup.

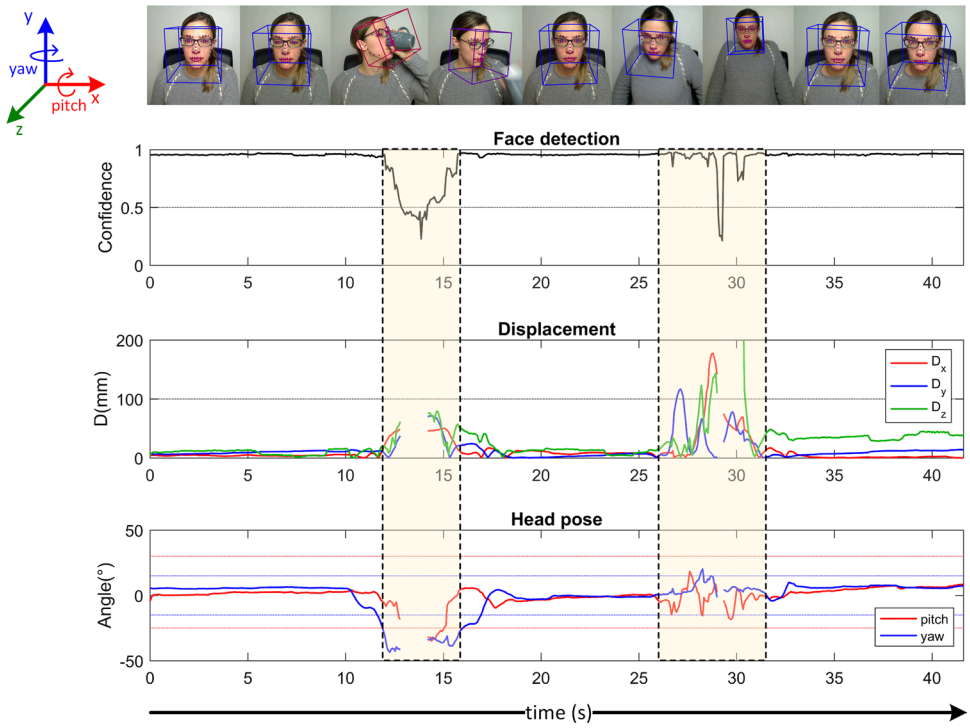


**Fig. 9** Metrics of engagement automatically computed from facial videos by *CrowdWatcher*. Highlighted yellow areas correspond to moments where the participant is not paying attention to the task

towards the screen or not. This metric computes the percent of frames in the video in which the participant was focused, allowing to detect abnormal periods of non-frontal head poses. It provides valuable insights about possible distractions that caused the participant to look away from the screen. It may also translate states of sleepiness, as excessive head-nodding and head-lowering have been widely established as good indicators of drowsiness [11, 45, 52].

*Percent of large displacements* Displacements in X, Y and Z directions ($D_x$, $D_y$ and $D_z$, respectively, in millimeters) indicate the difference in terms of head center position with respect to each time series' mean. A large displacement is considered to happen in a given frame if its value in any of the 3 axes is above 100 mm. This metric provides the percentage of frames in which there is a large displacement. Frequent displacements may imply that the participant is distracted from the task.

An example of engagement time series that can be obtained from *CrowdWatcher* is illustrated in Fig. 9.

## Open-source access

*CrowdWatcher* and its related tools are available open-source, enabling researchers to use it in the context of their studies. The source code and documentation can be downloaded from *GitHub*.[3] Its installation requires the setup of a web server with a Ruby on Rails framework. To simplify the installation of the server side, the platform is also provided as a pre-configured package in a virtual machine archive. Thanks to its design, the platform is flexible and can easily be used for any test where the interface is based on web technologies. The integration of the main task with the platform is simply performed by indicating to *CrowdWatcher* the address of the webpage where the main task is located. Indeed, the main task can be developed independently of the eye tracking platform and only needs a JavaScript library to communicate with it.

Once a participant completes an experiment, the facial video and click records are stored automatically on the server side. A provided script can be used to process the videos and log files, enabling to obtain gaze location predictions and associated quality information. Further scripts are also available to compute heatmaps from gaze locations, and engagement metrics from head pose logs.

## Application: proof of concept and performance evaluation

In the previous section, the platform was described from a conceptual and technical point of view. In this section, different experiments are described illustrating its performance and the type of information which can be collected from it.

[3] https://github.com/Telecommunication-Telemedia-Assessment/CrowdWatcher Accessed Feb. 2019.

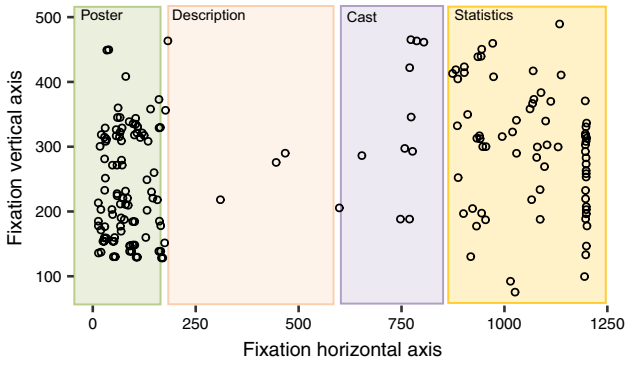


**Fig. 10** First experiment snapshot: the user is asked to select a movie. For visualization purposes, we colored the areas of interest internally used to compute gaze statistics. Exemplar gaze locations obtained for one participant are also overprinted (black dots)

## Proof of concept: user behavior analysis

The first experiment is a user behavior study, which was carried out with a preliminary version of the framework (release from July 2015). Our goal in this section is to provide a concrete example based on a conducted experiment of what can be achieved with *CrowdWatcher*. In this test, participants were asked to choose a movie from a list and then had to report the main reasons motivating their decision. Eye tracking data and self-reported metrics from the users were compared, with the aim to determine whether participants' actions reflected their answers. A more detailed description of the experiment and its results can be found in [41].

### Experiment description

The experiment described in this section corresponds to the "main task" inside the general flow of a crowdsourced eye tracking test (c.f. "General flow of a test" section). The main task started with instructions about the movie selection. It was then followed by a basic demographic questionnaire including questions about the user's film genre preferences. Consequently, the user was shown information about three different movies. This information was queried from the Internet Movie Database (IMDb[4]) and included the poster, a brief synopsis, the cast description and some statistics (such as its budget and the mean rating of the public), as illustrated in Fig. 10. Users were asked to select the movie they would like to watch. They could also indicate if they had previously seen any of the movies presented. Once the movie was selected, users needed to indicate at least three criteria that brought them to the final decision from the options: "title", "description", "cast", "poster", "director", "box office", "release date", "ratings" or "I knew the movie". Once accomplished, this process of selecting a movie and justifying replies was repeated twice. Then, the main task was over and the further steps of usage of the crowdsourcing platform as described in Fig. 6 were pursued.

### Participants and campaign information

Two crowdsourcing campaigns were conducted. The first involved volunteer online testers recruited during a science show and the second was a paid campaign carried out using the *Microworkers*[5] platform. In the first campaign, 10 participants completed the test. For the second campaign, 29 *Microworkers* users from English-speaking countries executed the work between 16th and 20th of June 2015 and were rewarded with $1 after providing the required proof. The uploaded data of 13 *Microworkers* users could not be considered in the evaluation because the respective videos were not properly received and task token was released before the end of the upload resulting in users quitting the platform before the end of the uploading process onthe server. This software defect was fixed in following *CrowdWatcher* releases, but resulted in this test in a lower number of available data. From the remaining videos of both campaigns, another set had to be rejected due to too bad lighting conditions and/or large head movements, which have then motivated the development of additional quality control metrics as reported in the platform description. In the end, data from 16 participants (13 males) was available for evaluating this test.

### Results

The proposed framework allowed to perform eye tracking over the entire length of the experiment. Figure 10 (black dots) illustrates the different gaze locations obtained for an exemplar participant while choosing a movie. It can be seen that this participant mainly focused on the poster column and to a lower extent on the metadata and cast information. To study the relationship between what participants answered and how they focused their gaze, the movie selection page was divided into 4 categories (see colored areas in Fig. 10). Then, the time users spent watching each category was calculated from eye tracking data. Results are depicted in Fig. 11 (left). It can be observed, for example, that the first participant spent 40% of the time on the "poster" category and 40% of the time in the movie "description" category.

After each selection, participants were explicitly asked to report which criteria motivated their decision (at least three criteria in the order of importance). A chart based on their replies, shown in Fig. 11 (right), was then built to compare

[4] http://www.imdb.com Accessed Feb. 2019.

[5] http://www.microworkers.com Accessed Feb. 2019.

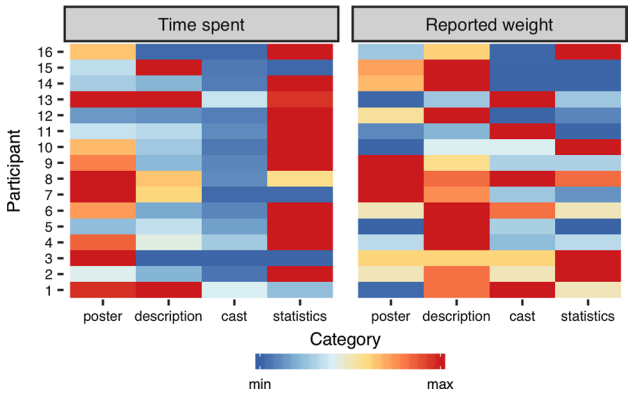


**Fig. 11** First experiment results. Left: eye tracker data—time spent by each participant at looking at each category. Right: survey data—category reported to be used as the main factor for the decision

their answers to eye tracking data. The selected main criterion was assigned a weight of 3, the second a weight of 2, and the third a weight of 1. The weights of all the selections were summed up per user and category to find out the main criteria used in choosing the movies. From this chart, it can be seen that the first participant based the decision mainly on the movie "description" category but also on the "cast", indicating that survey and eye tracking data reveal different influencing factors. To further extend this analysis, a Spearman's rank correlation between reported and eye tracking data was performed. Very different correlation values were obtained from one participant to another. For example, observers 7 and 15 are found having a correlation beyond 0.94, while observers 6 and 13 show negative correlations of $-0.74$ and $-0.83$, respectively. 9 users out of 16 show an absolute value of correlation lower than 0.4. On average across participants, the Spearman correlation is null. The lack of positively high correlation does not imply that participants were lying. It is possible, e.g., that statistics may have influenced their decisions, even though they do not admit the effect or they are not even aware of it. This study demonstrates that self-reports do not necessarily provide the same results as eye tracking measurements, as self-reports reflect mostly the conscious aspects of users' decisions. What participants primarily focus on may therefore have an indirect effect on their decisions.

This first experiment has been carried out in the context of a common crowdsourcing scenario, where participants are asked to fill in questionnaires. Beyond explicitly self-reported information, *CrowdWatcher* has demonstrated to provide extended knowledge on attention and decision-making processes motivating participants' decisions. In our next experiment, we demonstrate that the platform can also go beyond typical crowdsourcing scenarios and be applied to the more complex field of video QoE research.

## Performance analysis: application to multimedia QoE

As a second experiment, a comparison between a professional eye-tracker and the *CrowdWatcher* platform was performed. The considered use case is a crowdsourcing video quality test. This use case is a challenging task for the *CrowdWatcher* platform, as the performance of the platform relies on users' actions and participants do not perform any click when watching videos. The demonstration of *CrowdWatcher*'s performance in a video quality test is one of the key contributions of this paper compared to previous work.

### Experiment description

The goal of this experiment was to evaluate the suitability of *CrowdWatcher* to obtain eye tracking data in a crowdsourcing context where participants have very few interactions with the task interface, like when watching videos. To quantify the performance of the platform, an open database of videos providing ground truth heatmaps was employed [17]. This database contains video sequences having standard definition (SD) resolution. Twenty 10s long source sequences (SRCs) are provided. These sequences were processed by four hypothetical reference circuits (HRCs), in addition to the reference. The different HRCs correspond to packet loss and respective slicing degradations happening or not in salient regions, and with two different types of group of pictures length (20 and 30 frames). This results in 100 processed videos (PVSs). All the videos were evaluated by 30 participants using a professional IR eye tracker, which is provided as ground truth gaze data.

Using this database, this second experiment aimed at replicating the in-lab original experiment in a crowdsourcing context, and investigating how the proposed framework performs when fewer interactions from users are available. *CrowdWatcher*'s performance was then compared to the one of the professional IR eye tracker.

A subset of 5 SRCs was used. All 4 different HRCs for the selected 5 SRCs were included. The videos from the database were provided in interlaced scan, thus requiring prior deinterlacing. To do so, the filter from FFmpeg[6] was used. It addition, it was not practical to send RAW video files to a distant user via HTTPS, and therefore videos were encoded at a bitrate of 3 Mbps in H.264 using FFmpeg. This choice is reasonable to obtain visually unimpaired PVSs, considering the employed SD resolution and the strength of the distortions already included in the PVSs (slicing).

[6] https://ffmpeg.org/ Accessed Feb. 2019.

### Task description

The general flow of the experiment was as described in "General flow of a test" section and Fig. 6. The main task started with specific instructions on the video quality test to be performed: participants were indicated to later be watching videos with degradations and to evaluate the quality of these videos. Then, participants went through a training phase: A video sequence was presented to them, and at the end of the playback they were asked to rate its quality on a 5-grade Absolute Category Rating (ACR) scale with the labels "excellent", "good", "fair", "poor" and "bad". Using this setting, we made sure that workers interacted with the task naturally as if they were performing a regular video quality rating task. For the later evaluation, the subjective ratings were not of interest, as the platform's performance is evaluated based on the accuracy of the gaze estimations compared to the ground truth that is provided by the professional eye-tracker.

Concerning the video playback, several points need to be stressed. Firstly, the video playback could only start after having fully downloaded the video sequence. This was meant in order to avoid unexpected stalling events which were not part of the tested conditions. Secondly, once the video playback could start, the video player attempted to switch to full screen mode. However, to avoid intrusive websites, standard web page design does not allow to enforce the use of full screen mode without user agreement. In the current setup, it was then not guaranteed that the participant watched the videos in full-screen. The use of full-screen versus a smaller-size window was recorded during the tests.

Participants were allowed to redo the training if they did not feel comfortable with the task yet. Otherwise, they could perform the main part of the task, consisting in watching and evaluating the quality of 3 videos. These videos were chosen randomly across the 5 different sources and 4 different processings, ensuring that each participant watched 3 different SRCs. It took approximately 4 minutes to complete the entire test including the eye-tracker related steps and the main task.

A total of 45 participants from the *Microworkers* platform participated in the experiment and were paid $1 after completion of the test. 37 participants came from Bangladesh, 4 from Malaysia, 3 from Europe, and 1 from Russia.

### Results

Four main aspects are addressed in this section: general statistics about *CrowdWatcher*'s performance in predicting gaze, how the platform can be used to measure user engagement and the relationship between the platform performance and the captured image quality.

*User statistics* From the 45 *Microworkers* participants who took part into the test, the videos from 7 of them were not usable. This has been a strong improvement compared to the previous experiment, thanks to strengthening the on-line verification of the test conditions (c.f. "Client side" section). The 7 participants were rejected on the basis of engagement metrics, as will be further detailed in the following paragraph. These failures are due to the current limitation, where the screening of viewing condition is only performed at the very beginning of the test. Once participants pass the screening, it is possible for them to perform unexpected changes of the viewing conditions, even though they were told not to do so. This could have been addressed via an on-line screening all along the test, but it was not implemented due to the high processing power requirements on the user-side.

During the test, the resolution of the screen of the participants was recorded. 80% of the users had a resolution of $1366 \times 768$, 10% had lower resolutions with a lower bound of $1024 \times 768$, and 10% had resolutions higher than $1366 \times 768$ with a higher bound of $1920 \times 1080$.

Once a video was ready to play, the user was asked to confirm a dialog box which turned on the full-screen mode. Although participants were explicitly asked to use the full-screen mode, it was observed that 21% of them did not confirm the dialog, which resulted to playing back the video in a window of $560 \times 448$ instead of the screen resolution.

*Engagement statistics CrowdWatcher*'s module on user engagement (c.f. "Engagement" section) was applied to this crowdsourced experiment. The objective was to exclude from further analyses participants that were potentially distracted or not paying enough attention to the task. A participant was discarded if one or more of the following criteria occurred: (i) *percent of face detection* was below 90%, (ii) *percent of attention focused* was below 75% or (iii) *percent of large displacements* was above 5%. As a result, 7 participants were discarded: 4 because of criterion (i), 2 due to criterion (ii) and the remaining mainly for reason (iii).

*Performance statistics* As detailed in "Server side" section, it is possible to measure the accuracy of predicted gaze location for every calibration point. The left and right graphs in Fig. 12 depict prediction performances per user on the horizontal and vertical axis, respectively. Concerning the units, performances are provided in pixels and not in degrees. This is due to the crowdsourcing context, where the viewing distance is unknown. Therefore, only the prediction error in pixels can accurately be provided. Boxplots show information on the distribution of error values. Each point corresponds to the absolute value of a measured prediction error. Points beyond the whiskers indicate outliers. Additionally, the 25, 50 and 75 quartiles of prediction errors are included via the box diagram. It should be stated that 2 users were removed in Fig. 12-left (users 26 and 37), as predictions appeared to have failed on the horizontal axis

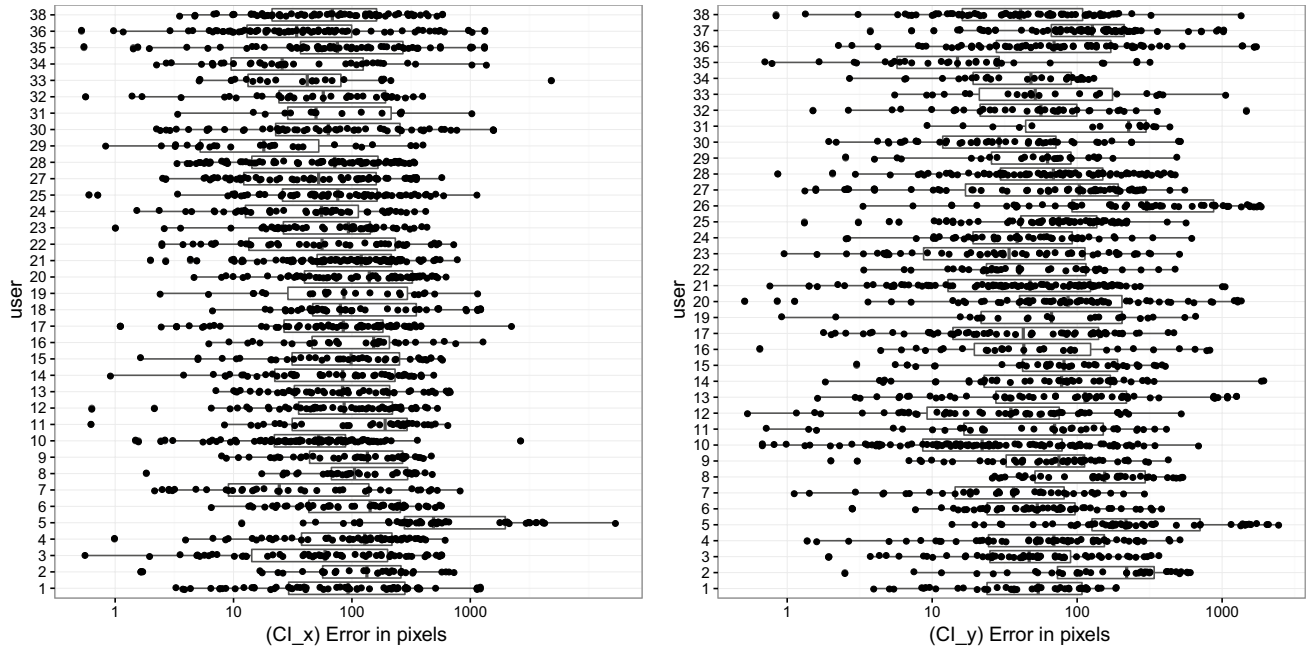


**Fig. 12** *CrowdWatcher*'s accuracy in predicting gaze locations on the horizontal (left) and vertical (right) axis, for all the users. Each point corresponds to a measured prediction error. Boxplots provide the 25, 50 and 75 quartiles, and points beyond the whiskers correspond to outliers

and masked other users' results in the figure. It must thus be noticed that prediction can fail in one axis without affecting the other axis, as users 26 and 37 appear to have reasonable results on the vertical axis. Including these participants, it can be concluded that the 90% of users have prediction errors corresponding to the 25 and 75 quartiles between 56 and 332 pixels. Similarly, 90% of the users have a median and mean value of at most 158 and 277 pixels, respectively.

*Performance and technical factors* To better characterize the requirements for ensuring good quality results, several analyses were performed to study the influence of technical factors on prediction error values. Firstly, a study addressing the viewing position of the participants was performed. A regression analysis was performed to put into relation the size of participants' face on the videos and the size of the confidence intervals (CI). The effect of the face size was almost found to be significant at 95% ($F = 4.206$, $p = 0.0501$). More precisely, considering that the videos sent by participants had a constant resolution of $640 \times 480$, it was found that when the size of the face is below 250px, there is a linear relationship between the average error bar size and the participant's face size according to the equation:

$$\sqrt{CI_x^2 + CI_y^2} = 319.782 - 1.319 \times Face_h \quad (3)$$

with $Face_h$ being the height of the face in pixels, and $CI_x$ and $CI_y$ the average size in pixels of the confidence intervals. Beyond the threshold of 250px, no clear relationship between face size and performance can be observed.

Pursuing user-related analyses, we studied to what extend participant's head motion in front of the camera impacts performance. To this aim, the average CI value was put into relation with the average difference of head center position between consecutive frames, which was considered as the measure of user's motion. Figure 13 shows the result of such analysis. It can be observed that users mostly remain still in front of the camera, resulting in relatively small average motion values. In case of larger motion, it can be observed that the performance of the platform decreases. However, when users do not move large confidence intervals can still appear. Therefore, user motion is not the only factor influencing the performance of the platform.

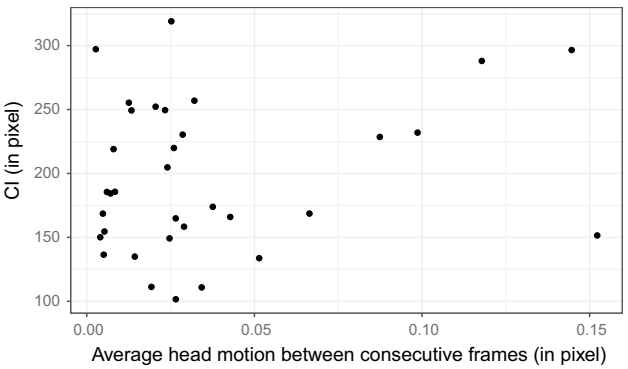


**Fig. 13** Relationship between user motion and *CrowdWatcher*'s accuracy

**Table 1** Performance evaluation of the platform, compared to the ground truth data collected with the professional IR eye tracker

| PVS | KL | CC | NSS | Judd AUC |
|---|---|---|---|---|
| SRC01 HRC02 | 2.312 | 0.363 | 0.1657 | 0.6023 |
| SRC06 HRC02 | 2.330 | 0.356 | 0.1227 | 0.6215 |
| SRC08 HRC02 | 5.227 | − 0.0425 | − 0.1008 | 0.5515 |
| SRC15 HRC02 | 5.120 | 0.1930 | 0.2308 | 0.6326 |
| SRC17 HRC02 | 22.52 | − 0.1438 | 0.3822 | 0.6979 |

The second direction for performance analysis is to relate the image quality of the video provided by the participants and performance. It was observed that the bitrate of the videos received on the server (i.e. the face recordings) range from 300 Kbps up to 4 Mbps, resulting in different image sharpness. A regression analysis was performed, and a linear model $CI = A \times bitrate + B$ was fit to study the significance of the contribution of the variable bitrate to the model performance. Using such analysis, no clear relation between bitrate and CI size was found in our data ($F = 1.097$, $p = 0.303$). Aiming at providing a better characterization of the relationship between image sharpness and *CrowdWatcher*'s accuracy, the set of no-reference image quality indicators provided by AGH University[7] within the context of the VQEG project "MOAVI"[8] was used to characterize the video properties according a large variety of image quality indicators. These include: blockiness, spatial activity, blur, exposure, contrast and noise. Similarly to the bitrate analysis, we performed multiple single regression analysis relating each individual measurement and confidence interval size. It was found that that the noise metric from Janowski and Papir [38] has the highest relation with the framework's performance, although it was not significant ($F = 2.164$, $p = 0.154$). It can thus be concluded that it is not straightforward to link the quality of webcam images and the framework's performance. The position and motion of the observer was a more critical aspect in our data. This is to be expected as participants' face size (and then participants' pupil size) has the most straightforward effect on pupil center detection.

### Comparison with a professional IR eye tracker

To evaluate the performance of *CrowdWatcher* with regard to state-of-the-art platforms, it was compared to a high-cost eye tracker using IR lighting. The open video database used in the experiment also provides ground truth eye tracking data collected from such a professional IR eye tracker in the laboratory [17]. As gaze locations are time-dependent and vary from one observer to another, heatmaps generated based on averages across all participants are compared here.

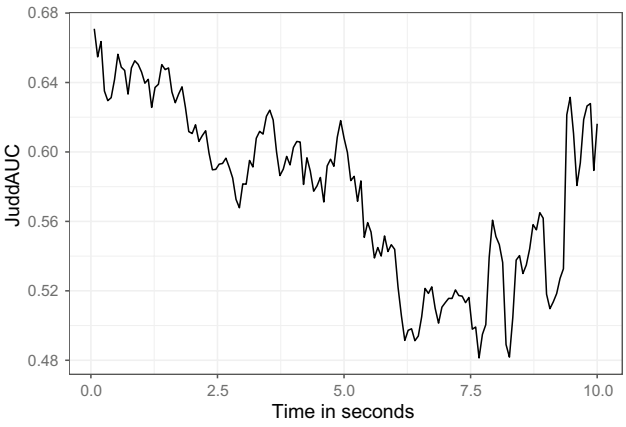


**Fig. 14** Temporal evolution of performance in terms of Judd AUC compared to ground-truth data. The curve shows an average of Judd AUC over all SRCs and HRCs

The method described in "Heatmaps" section was applied to generate heatmaps from crowdsourced data. As the ground truth was provided with both heatmaps and eye tracking data, four different metrics were computed to evaluate the performance of the platform: Kullback-Leibler divergence (KL), Pearson's correlation (CC), normalized scanpath saliency (NSS) and Judd AUC [6]. The overall results can be found in Table 1. This information is provided for every SRC of one HRC. Considering that video sequences are addressed, these metrics can be computed on a per-frame basis, and therefore the median value over time is reported. In can be observed that *CrowdWatcher*'s performance is generally low when compared to the data collected with the IR professional eye tracker. This will be further discussed in "Discussion and lessons learned" section.

Considering that the platform depends on actions, a key point is to study how long calibration can hold while the user is only staring at the screen without interacting with the computer. To this aim, the average Judd AUC score over all the SRCs and HRCs is computed per-frame, allowing to observe the temporal evolution of performance as a function of time. Results are depicted in Fig. 14 and show that a reasonable performance can be maintained on average for 5 s. Beyond this threshold, it drops severely. Additionally, an increase of performance towards the end of video sequences can be observed. This is due to the presence of new clicks from the user after having seen the video, while filling a final evaluation sheet. Indeed, as the proposed platform performs eye tracking in an off-line manner, both past and future click events can be used to predict gaze locations.

[7] http://vq.kt.agh.edu.pl/metrics.html Accessed Feb. 2019.

[8] https://www.its.bldrdoc.gov/vqeg/projects/moavi/moavi.aspx Accessed Feb. 2019.

### Discussion and lessons learned

Besides using a professional vs. a RGB-based eye tracker, several other reasons may explain the low performance of *CrowdWatcher* with regard to the original in-lab scenario. Firstly, 45 participants were involved in the crowdsourcing experiment. Due to the fact that the entire test should not be longer than 5 minutes, it was not possible to show every PVS to each participant. Therefore only 3 repetitions for each PVS were available, and not 30 as in the original work. This has led to noisier heatmaps compared to the ground truth. A second issue is that the videos from the selected database have a SD resolution ($720 \times 576$ pixels). As described in "Results" section, quartiles Q1 and Q3 of prediction errors have values of 60 and 400 pixels, respectively, with a median of 160. Thus, the accuracy of the platform may have resulted insufficient for this video resolution. Moreover, it was observed that 21% of the observers did not use the full-screen mode and watched the video in a window of $560 \times 448$ pixels (see "Results" section), which has made the prediction errors of the platform even more critical.

All the aforementioned issues combined, may have caused a limited performance in this particular test scenario when compared with its in-lab IR counterpart. Nevertheless, it has been demonstrated that *CrowdWatcher* is still able to collect useful gaze and user engagement information in the far more challenging environment of crowdsourcing.

As a lesson learned, it appears that in the crowdsourcing context there is even more special attention to pay in the design of the experiment, if it is expected to obtain more accurate heatmaps from videos. This includes the selection of appropriate SRCs to fully cover the screen of the user, and ensure that the full-screen mode is used. Also, a larger number of participants may need to be recruited in light of a higher number of repetitions per video while maintaining the 5 min length constraint. From the current results on heatmaps estimation, *CrowdWatcher* is highly recommended for collecting information about the general tendency of where the participant looked on the screen (top left, middle left, bottom right, up right, etc.). However, when it comes to obtaining heatmaps with very high precision, further validation is needed according to the observations raised in this section.

## Conclusion

In this paper, the platform *CrowdWatcher* for assessing gaze and user engagement was presented. It is provided open-source, enabling researchers to reuse this work for their own experiments. Together with the description of the platform, two different subjective experiments were presented as prototypical examples of use. The first one addressed the measurement of participants' behavior while performing the task of selecting a movie from a list of options. Results showed that the platform can provide complementary information to self-reported data, as users do not necessarily behave in the same manner as they report to do. The second experiment covered a video streaming QoE test scenario, and allowed comparing *CrowdWatcher*'s performance to a professional IR eye tracker. Even though the platform showed deviations from the results obtained with the professional eye tracker, it is able to provide valuable general information about the attention of participants and where they were looking at the screen. This platform is indeed—to the best of our knowledge—the first in the literature that allows the automatic measure of user engagement from RGB cameras and to identify participants not paying attention to the crowdsourced task.

It must be finally highlighted that *CrowdWatcher*'s extended measurements, namely environmental conditions, gaze locations and user engagement, may raise some privacy concerns for participants. It is therefore strongly recommended to the experimenter who will use the platform to inform test users about what kind of data will be collected before accepting the task.

**Acknowledgements** The authors thank Microworkers.com for sponsoring some of the crowdsourcing experiments. The research leading to these results received funding from the Deutsche Forschungsgemeinschaft (DFG) under Grants HO4770/2-2, TR257/38-2.

## References

1. Agaian SS, Lentz KP, Grigoryan AM (2000) A new measure of image enhancement. In: International conference on signal processing & communication
2. Akamine WY, Farias MC (2014) Incorporating visual attention models into video quality metrics. In: SPIE-IS&T electronic imaging—image quality and system performance, vol 9016
3. Alnajar F, Gevers T, Valenti R, Ghebreab S (2013) Calibration-free gaze estimation using human gaze patterns. In: IEEE international conference on computer vision, pp 137–144
4. Bielikova M, Konopka M, Simko J, Moro R, Tvarozek J, Hlavac P, Kuric E (2018) Eye-tracking en masse: group user studies, lab infrastructure, and practices. J Eye Mov Res 11(3):6
5. Blignaut P (2010) Visual span and other parameters for the generation of heatmaps. In: Symposium on eye-tracking research & applications, pp 125–128
6. Bylinskii Z, Judd T, Oliva A, Torralba A, Durand F (2018) What do different evaluation metrics tell us about saliency models? IEEE Trans Pattern Anal Mach Intell 3:740–757
7. Camgaze: Eye tracking in visible light from a webcam. https://github.com/wallarelvo/camgaze. Accessed Feb 2019
8. Carrasco M (2011) Visual attention: the past 25 years. Vis Res 51(13):1484–1525

9. Charness N, Dijkstra K, Jastrzembski T, Weaver S, Champion M (2008) Monitor viewing distance for younger and older workers. In: Human factors and ergonomics society annual meeting, vol 52, pp 1614–1617
10. Cheng S, Sun Z, Ma X, Forlizzi JL, Hudson SE, Dey A (2015) Social eye tracking: Gaze recall with online crowds. In: 18th ACM conference on computer supported cooperative work & social computing, pp 454–463
11. Choi IH, Jeong CH, Kim YG (2016) Tracking a driver's face against extreme head poses and inference of drowsiness using a Hidden Markov Model. Appl Sci 6(5):137
12. CrowdWatcher: An open source platform to catch the eye of the crowd. https://github.com/Telecommunication-Telemedia-Assessment/CrowdWatcher
13. CVC: CVC eye tracker. https://github.com/tiendan/OpenGazer. Accessed Feb 2019
14. De Vreede T, Nguyen C, De Vreede GJ, Boughzala I, Oh O, Reiter-Palmon R (2013) A theoretical model of user engagement in crowdsourcing. In: International conference on collaboration and technology, pp 94–109
15. Drouard V, Horaud R, Deleforge A, Ba S, Evangelidis G (2017) Robust head-pose estimation based on partially-latent mixture of linear regressions. IEEE Trans Image Process 26(3):1428–1440
16. Egger-Lampl S, Redi J, Hoßfeld T, Hirth M, Möller S, Naderi B, Keimel C, Saupe D (2017) Crowdsourcing quality of experience experiments. In: Archambault D, Purchase H, Hoßfeld T (eds) Evaluation in the crowd. Crowdsourcing and human-centered experiments. Springer, Berlin, pp 154–190
17. Engelke U, Barkowsky M, Callet PL, Zepernick HJ (2010) Modelling saliency awareness for objective video quality assessment. In: International workshop on quality of multimedia experience
18. Engelke U, Pepion R, Callet PL, Zepernick HJ (2010) Linking distortion perception and visual saliency in h.264/avc coded video containing packet loss. In: SPIE 7744, Visual communications and image processing
19. Engelke U, Zepernick HJ (2010) A framework for optimal region-of-interest based quality assessment in wireless imaging. J Electron Imaging 19(1):1–13
20. EyeLink: 1000 Plus Eye Tracker. https://www.sr-research.com/products/eyelink-1000-plus/. Accessed Feb 2019
21. EyeTribe: The Eye Tribe eye tracker. http://theeyetribe.com/theeyetribe.com/about/index.html. Accessed Feb 2019
22. Ferhat O, Vilariño F (2016) Low cost eye tracking: the current panorama. Comput Intell Neurosci 5:2–14
23. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
24. Gadiraju U, Kawase R, Dietze S, Demartini G (2015) Understanding malicious behavior in crowdsourcing platforms: the case of online surveys. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems. ACM, pp 1631–1640
25. Gadiraju U, Möller S, Nöllenburg M, Saupe D, Egger-Lampl S, Archambault D, Fisher B (2017) Crowdsourcing versus the laboratory: towards human-centered experiments using the crowd. In: Archambault D, Purchase H, Hoßfeld T (eds) Evaluation in the crowd. Crowdsourcing and human-centered experiments. Springer, Berlin, pp 6–26
26. GazeHawk: Eye tracking for everyone. http://www.gazehawk.com/. Accessed Feb 2019
27. Gazepoint: Eye tracking systems. https://www.gazept.com/. Accessed Feb 2019
28. Glas N, Pelachaud C (2015) Definitions of engagement in human–agent interaction. In: International workshop on engagment in human computer interaction, pp 944–949
29. Gomez S, Jianu R, Cabeen R, Guo H, Laidlaw D (2016) Fauxvea: crowdsourcing gaze location estimates for visualization analysis tasks
30. Grier RA (2004) Visual attention and web design. Ph.D. Thesis, University of Cincinnati, Cincinnati, USA
31. Hansen DW, Ji Q (2010) In the eye of the beholder: a survey of models for eyes and gaze. IEEE Trans Pattern Anal Mach Intell 32(3):478–500
32. Hauser DJ, Schwarz N (2016) Attentive turkers: Mturk participants perform better on online attention checks than do subject pool participants. Behavior Res Methods 48(1):400–407
33. Hernandez J, Liu Z, Hulten G, DeBarr D, Krum K, Zhang Z (2013) Measuring the engagement level of tv viewers. In: IEEE international conference on automatic face and gesture recognition, pp. 1–7
34. Hirth M, Hoßfeld T, Mellia M, Schwartz C, Lehrieder F (2015) Crowdsourced network measurements: benefits and best practices. Comput Netw 90:85–98
35. Hossfeld T, Keimel C, Hirth M, Gardlo B, Habigt J, Diepold K, Tran-Gia P (2014) Best practices for QoE crowdtesting: QoE assessment with crowdsourcing. Trans Multimedia 16:541–558
36. Huang J, White RW, Buscher G (2012) User see, user point: gaze and cursor alignment in web search. In: Conference on human factors in computing systems
37. ITU: Open source gaze tracking library. https://sourceforge.net/projects/gazetrackinglib/. Accessed Feb 2019
38. Janowski L, Papir Z (2009) Modeling subjective tests of quality of experience with a generalized linear model. In: International workshop on quality of multimedia experience
39. Keimel C, Habigt J, Diepold K (2012) Challenges in crowd-based video quality assessment. In: Forth international workshop on quality of multimedia experience (QoMEX 2012), pp 13–18
40. Kim NW, Bylinskii Z, Borkin MA, Gajos KZ, Oliva A, Durand F, Pfister H (2017) Bubbleview: an interface for crowdsourcing image importance maps and tracking visual attention. ACM Trans Comput Hum Interact 24(5):36
41. Lebreton P, Hupont I, Mäki T, Skodras E, Hirth M (2015) Eye tracker in the wild, the delta between what is said and done in a crowdsourcing experiment. In: International ACM workshop on crowdsourcing for multimedia. Brisbane, Australia
42. Lebreton P, Mäki T, Skodras E, Hupont I, Hirth M (2015) Bridging the gap between eye tracking and crowdsourcing. In: SPIE 9394, Human vision and electronic imaging XX
43. Lindgaard G, Fernandes G, Dudek C, Brown J (2006) Attention web designers: you have 50 milliseconds to make a good first impression!. Behav Inf Technol 25(2):115–126
44. Lu Z, Lin W, Ong E, Yang X, Yao S (2003) PQSM-based RR and NR video quality metrics. In: International society for optical engineering (SPIE), vol 5150, pp 633–640
45. Lyu J, Yuan Z, Chen D (2018) Long-term multi-granularity deep framework for driver drowsiness detection. arXiv preprint arXiv:1801.02325
46. Mancas M, Ferrera VP (2016) How to measure attention? In: From human attention to computational attention, pp 21–38
47. Mao A, Kamar E, Horvitz E (2013) Why stop now? Predicting worker engagement in online crowdsourcing. In: AAAI conference on human computation and crowdsourcing
48. Martin D, Carpendale S, Gupta N, Hoßfeld T, Naderi B, Redi J, Siahaan E, Wechsung I (2017) Understanding the crowd: ethical and practical matters in the academic use of crowdsourcing. In: Archambault D, Purchase H, Hoßfeld T (eds) Evaluation in the crowd. Crowdsourcing and human-centered experiments. Springer, Berlin, pp 27–69
49. Meur OL, Ninassi A, Callet PL, Barba D (2010) Overt visual attention for free-viewing and quality assessment tasks impact of

the regions of interest on a video quality metric. Signal Process Image Commun 25:547–558
50. NEUROTechnology: SentiGaze SDK. http://www.neurotechnology.com/sentigaze.html. Accessed Feb 2019
51. Ninassi A, Meur OL, Callet PL, Barba D, Tirel A (2006) Task impact on the visual attention in subjective image quality assessment. In: European signal processing conference
52. Oliveira L, Cardoso JS, Lourenço A, Ahlström C (2018) Driver drowsiness detection: a comparison between intrusive and non-intrusive signal acquisition methods. In: 7th European workshop on visual information processing (EUVIP), pp 1–6
53. OpenGazer: Open-source gaze tracker for ordinary webcams. http://www.inference.phy.cam.ac.uk/opengazer/. Accessed Feb 2019
54. Papoutsaki A, Sangkloy P, Laskey J, Daskalova N, Huang J, Hays J (2016) Webgazer: scalable webcam eye tracking using user interactions. In: International joint conference on artificial intelligence, pp 3839–3845
55. Peters C, Castellano G, de Freitas S (2009) An exploration of user engagement in HCI. In: International workshop on affective-aware virtual agents and social robots, p 9
56. Poletti M, Rucci M (2016) A compact field guide to the study of microsaccades: challenges and functions. Vis Res 118:83–97
57. PrincetonVision: TurkerGaze GitHub repository. https://github.com/PrincetonVision/TurkerGaze. Accessed Feb 2019
58. PupilLabs: Platform for eye tracking and egocentric vision research. https://pupil-labs.com/pupil/. Accessed Feb 2019
59. Redi JA, Povoa I (2013) The role of visual attention in the aesthetic appeal of consumer images: a preliminary study. In: Visual communications and image processing
60. Rempel D, Willms K, Anshel J, Jaschinski W, Sheedy J (2007) The effects of visual display distance on eye accommodation, head posture, and vision and neck symptoms. Hum Factors 49(5):830–838
61. Riegler M, Eg R, Calvet L, Lux M, Halvorsen P, Griwodz C (2015) Playing around the eye tracker—a serious game based dataset. In: GamifIR, pp 34–40
62. Rodden K, Fu X, Aula A, Spiro I (2008) Eye-mouse coordination patterns on web search results pages. In: CHI'08 extended abstracts on Human factors in computing systems, pp 2997–3002
63. Rudoy D, Goldman D, Shechtman E, Zelnik-Manor L (2012) Crowdsourcing gaze data collection. In: Collective intelligence conference
64. Salam H, Celiktutan O, Hupont I, Gunes H, Chetouani M (2016) Fully automatic analysis of engagement and its relationship to personality in human-robot interactions. IEEE Access 5:705–721
65. Salam H, Chetouani M (2015) A multi-level context-based modeling of engagement in human-robot interaction. In: 2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG), vol 3. IEEE, pp 1–6
66. Savino PJ, Danesh-Meyer HV (2012) Color Atlas and Synopsis of Clinical Ophthalmology-Wills Eye Institute-Neuro-Ophthalmology. Lippincott Williams & Wilkins, Philadelphia
67. SightCorp: InSight SDK. http://sightcorp.com/insight/. Accessed Feb 2019
68. Simko J, Bielikova M (2015) Gaze-tracked crowdsourcing. In: International workshop on semantic and social media adaptation and personalization, pp 1–5
69. Sticky: Visual Measurement Tool. https://sticky.ai/. Accessed Feb 2019
70. Stiefelhagen R (2002) Tracking focus of attention in meetings. In: IEEE international conference on multimodal interfaces, p 273
71. Sugano Y, Matsushita Y, Sato Y, Koike H (2015) Appearance-based gaze estimation with online calibration from mouse operations. IEEE Trans Hum Mach Syst 45(6):750–760
72. Tobii: Eye tracking products. https://www.tobii.com/. Accessed Feb 2019
73. Viola P, Jones M (2004) Robust real-time face detection. Int J Comput Vis 57(2):137–154
74. VisageTechnologies: FaceTrack SDK. http://visagetechnologies.com/products-and-services/visagesdk/facetrack/eye-and-gaze-tracking/. Accessed Feb 2019
75. WebGazer: WebGazer library
76. Wood E, Baltrusaitis T, Zhang X, Sugano Y, Robinson P, Bulling A (2015) Rendering of eyes for eye-shape registration and gaze estimation. In: IEEE international conference on computer vision
77. xLabs: xLabs SDK for eye, gaze and head tracking. http://xlabsgaze.com/. Accessed Feb 2019
78. Xu P, Ehinger KA, Zhang Y, Finkelstein A, Kulkarni SR, Xiao J (2015) TurkerGaze: Crowdsourcing saliency with webcam based eye tracking. arXiv preprint arXiv:1504.06755
79. You J (2013) Attention driven visual QOE: mechanism and methodologies. In: International conference on signal and information processing (ChinaSIP)
80. Zielinski P, NetGazer. https://sourceforge.net/projects/netgazer/. Accessed Feb 2019

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.