Eye-tracking. Privacy interfaces for the next ubiquitous modality

Problem statement: privacy risks associated with eye-tracking

Eye-tracking is about to become the next ubiquitous input modality. On mobile phones, eye-trackers join other visual sensors such as front- and rear-facing photo/video cameras, depth cameras, and ambient light detectors. Similar to those as well as to geo-positioning capabilities and built-in microphones, eye-tracking is likely to become mainstream outside the desktop PC first: mobile devices such as phones, tablets and augmented-reality glasses, in-car computation, and single-purpose computing kiosks in public spaces (e.g., ticket machines, outdoor media / advertising billboards, self-help PCs in retailing).

Gaze-aware computers offer multiple benefits for individual users and the Internet ecosystem at large. To name a few: mobile devices can dedicate the scarce screen estate to areas of interest; cars can identify signs of fatigue and drowsiness and apply safety-enhancing countermeasures; content publishers get a valuable feedback channel about which stories or media are most interesting; pay-per-mille advertising can ultimately have a true count for ad impressions; disabled users have new ways to interact with the machine when unable to use mouse or keyboard; shoulder-surfing for security-critical input can be defeated through gaze-based entry. These promises rely on users' trust into the eye-tracking infrastructure and on striking a balance between privacy and the opportunities of tracking.

Together with my colleague Dan Liebling, I recently developed privacy considerations for a pervasive eye-tracking world (Liebling & Preibusch, 2014). We conducted a privacy impact assessment for eye-trackers to record human activity continuously and ubiquitously. Here, I summarise the main findings before deducing recommendations for privacy-aware APIs and UIs.

In a rough approximation, an eye-tracker is a fancy camera. Eye movements are returned as a time-series of screen coordinates. Some high-end devices also return pupil diameter. The eye-tracker takes care of data pre-processing and typically returns a series of fixations, as the original data is noisy due to biological noise, sensor uncertainty, and ambient illumination. When making sense of fixations, one combines gaze data with the semantics of the image displayed on screen: *where* the user looks is turned into *what* the user is looking at. Browsers have direct access to what they display and Websites can determine the positions of their constituent parts. Fixation duration is a strong indicator of users' interests and at the heart of commercial eye-tracking studies that create heat-maps or examine Website usability issues. **This approximation fails to capture privacy risks.**

Users' gaze involuntarily lies with their interests: familiar faces, the own race, the sexually interesting other or cigarettes for smokers. Pupil size varies with cognitive load or interest in certain foods or attractive actors. Eye-tracking data thus gives a rich picture of Web users' lifestyle, their interests and preferences, as well as their demographics.

Eye-tracking data is a biometric. Like other biometrics (e.g., minutiae, iris scans), gaze patterns allow the unique re-identification of an individual within a database of past recordings. The privacy community has been talking much about device or browser fingerprinting vectors in Web usage recently, such as JavaScript performance or installed fonts / plugins. These patterns can be used to re-identify a machine; gaze patterns take fingerprinting to the user level and allow **re-identification of the same individual across Websites and machines**.

For yet unknown users, biometric key indicators can be inferred from gaze and pupil dilation, such as gender, age or general health status. As outlined above, higher-level indicators such as political,

sexual, cultural and other lifestyle preferences can also be learned when combining gaze data with the content displayed to the user.

Potential remedies and avenues for privacy-enhanced eye-tracking

There is a challenge for research, engineering, policy and businesses to realise the potential of eye-tracking with privacy affordances. It is difficult, from exhausting to impossible, for users to control their eye movements and pupil dilation. Privacy as control must therefore lie with the collection and use of the data captured by an eye-tracker.

Potential privacy remedies fall into five categories:

- Destruction and obstruction of eye-trackers
- Meaningful browser APIs to access eye-tracking data that are designed with privacy in mind and reconcile user privacy with Websites' and applications' sensory needs
- Meaningful ways to give notice to users' what data is collected about them through eye-tracking
- Meaningful ways for uses to have control over their data and to choose and configure which data is released to whom
- Policy changes to provide minimum privacy guarantees for eye-tracking data

Destruction and obstruction (a no-starter)

Like a front-facing webcam, an eye-tracker needs line of sight and close proximity to observe the user (in contrast to geo-location via IP address or visible Wi-Fi networks, for instance). Users are thus able to shield the eye-tracker, by taping over the sensor, or destroying it right away. **Users can put a sticker over their eye-tracker to evade tracking. While effective, this approach offers an inferior user experience:**

- Stickers only provide a per-system on/off switch rather than per-user and per-application access permissions to certain data streams, contravening the principles of privacy-enhanced APIs (Preibusch, 2010).
- It is cumbersome and time-consuming to remove the sticker when the eye-tracker should be used. Stickers leave residue and only survive a few glue/unglue cycles (dedicated camera stickers are now sold in multi-buy packs).
- Like tinfoil hats, stickers send a 'privacy paranoia' signal to bystanders, resulting in social stigma. Through peer-group pressure, users may feel compelled to leave the tracker open against their preferences.
- The DIY workaround of a sticker is hardly compatible with stylish high-end devices.
- Users may not know where to put the sticker. A modern eye-tracker hardly looks like a webcam and certainly not like a stylised video surveillance camera.
- Users may be unable to selectively block the eye-tracker when the device is integrated with other sensors such as an ambient light or a visual camera.
- Safety regulations or contractual obligations (e.g., by the car insurance company) may prohibit the user from obstructing the eye-tracker, such as for in-car applications.
- User may not be allowed to tamper devices outside their ownership (e.g., public kiosks, rented, leased or borrowed cars).
- Stickers do not offer a feedback channel to the device: applications are unaware that gaze data is effectively unavailable when the lens was taped over but the functionality not disabled. (NB: users may find it privacy-preserving to blur the distinction between these two cases.)

Semantic eye-tracking APIs with built-in privacy (an effective near-term solution)

The W3C, associated Web Stewards and technology manufacturers are in a position to standardise privacy-considerate browser APIs for eye-tracking data *before the technology reaches the mass market*. Specification of these APIs can be achieved within the remit of existing working groups, such as the DAP (Device API). Privacy principles developed for camera, microphone or geo-location data can be applied analogously to the eye-tracking modality (Cooper, Hirsch, & Morris, 2010). A good starting point are also the APIs provided by the Kinect camera, which does not expose the raw data collected from the RGB and depth cameras, but provides a skeleton model of the gamer, for instance. Existing APIs (navigator.getUserMedia()) can be augmented to provide access to eye-tracking data.

1. Spatial abstraction. Websites should not have access to raw gaze and pupil data, but instead register regions of interests with **gaze-triggered event handlers**: ongazeenter, ongazeover, ongazeleave etc., onblink, analogous to onmouseenter and onclick. Block elements of a Web page could be annotated by a new **attribute or :seen pseudo-class** that exposes whether or not the element has been seen, analogous to :visited. This class can be queried through getComputedStyle(), although without repeating the CSS history leak. Same-origin policies must be respected here.
2. Temporal abstraction. Exact gaze timings are not typically required for commercial applications. The temporal resolution can be capped, for instance by providing readings only every second for continuous data feeds (Method example: navigator.getUserMedia({gaze: true})). It is however preferable that the browser integrates fixations over time and constructs a **heatmap**, available to Websites similar to screen-capture capabilities. Heatmaps are sufficient to identify areas of interest; users profit from enhanced privacy while Websites do not have to perform heavy processing.

Consent dialogs and notifications (an effective near-term solution)

User must be asked before the browser releases eye-racking data through the API. From that follow requirements for the UI; existing consent dialogs and notifications can be used as a starting point: microphone/camera and screen-capture access (Figure 1 and Figure 2). Interestingly, two ways to inform the user about intended data collection are found conflated in existing dialogs: on the one hand, the device is named (“microphone”), on the other hand, the actual content is shown (“your computer’s location”, thumbnail of the screen portion to be shared).

Users should be informed about the data collected by the eye-tracker rather than just the device itself. For instance, telling the user that a Website wants to record which portions of the screen the user is looking at, or whether a certain element has been seen. Browsers should also provide an explanatory example, such as the heatmap to be shared overlaid over the actual page. Similarly, browsers should provide an explanation what this data could be used for (e.g., detect elevated interest in certain content).

An indicator must be shown as long as eye-tracking is active on the Web page, similar to existing status icons (Figure 3). Such icons are already common for mobile devices even when screen space is scarce.

On larger devices (e.g., laptops), an LED status light may indicate that the eye-tracker is active. Such hardware indicators, despite their manufacturing costs and space requirements, are commonly found to indicate camera and wireless status (or hard-disk / floppy activity since the early days of desktop PCs). Similarly, a device may offer a physical and/or soft operating switch for an eye-tracker, which however does not make the in-browser dialogs redundant, as they may not be under the user’s control.

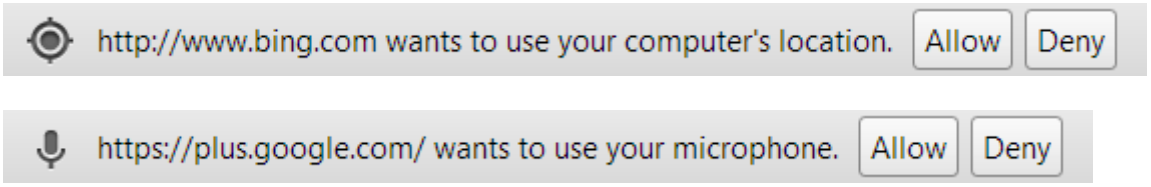

Figure 1. These (non-modal) bars are displayed by the Google Chrome browser when a Website requests access to the geo-location or to the microphone. The user can ignore, deny or allow.

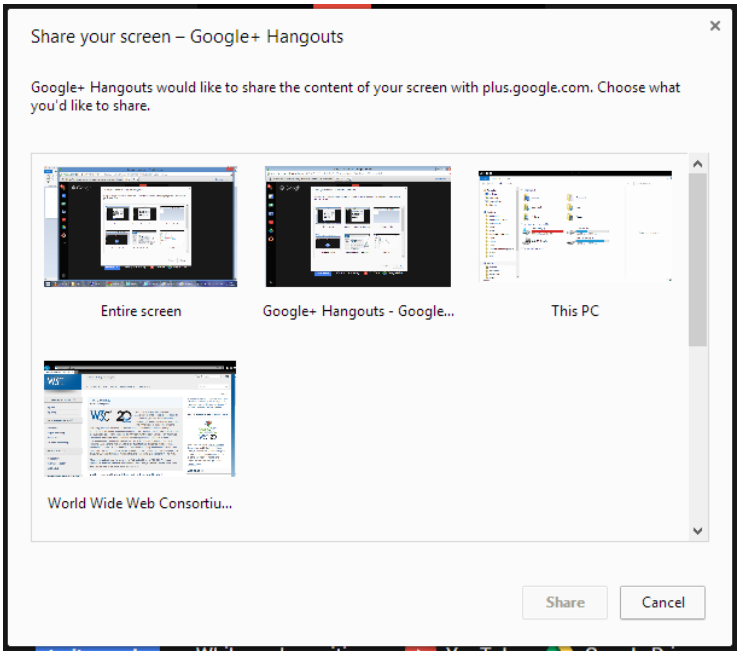

Figure 2. This (modal) dialog is displayed by the Google Chrome browser when a Website requests screen-recording. The user can cancel or choose which content shall be shared.

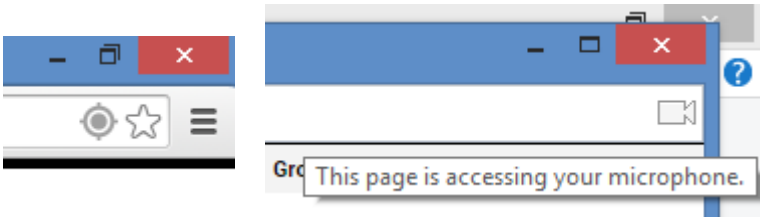

Figure 3. In the Google Chrome browser, the address bar is augmented by an icon that summarises permissions the user has granted the page to access her geo-location or microphone, as explained by a tooltip when hovering over the icon.

Policy changes (ongoing long-term activity)

Existing data protection covers eye-tracking data as another form of personally identifiable information. Importantly, its potential as a biometric must be recognised, as it allows re-identification of individuals and the inference of special data items, including health details.

New regulation and enforcement needs to address eye-tracking outside the user's realm, such as rental cars or public displays. Some countries already have standardised icons to notify citizens about video surveillance; an equivalent icon and the associated notification requirement should be developed and applied for eye-tracking.

References

Cooper, A., Hirsch, F., & Morris, J. (2010). *Device API Privacy Requirements (W3C Working Group Note 29 June 2010)*. http://www.w3.org/TR/2010/NOTE-dap-privacy-reqs-20100629/

Liebling, D. J., & Preibusch, S. (2014). Privacy considerations for a pervasive eye tracking world. *oceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp Adjunct)*, (pp. 1169--1177).

Preibusch, S. (2010). APIs and consumers' privacy decision-making. *W3C Workshop on Privacy for Advanced Web APIs.* W3C. http://www.w3.org/2010/api-privacy-ws/papers/privacy-ws-1.pdf