November 19, 2017

Gaze and attention detection API for content pre-fetching

Sandro Feuz

Sebastian Millius

Gaze and attention detection API for content pre-fetching

ABSTRACT

As software applications become increasingly sophisticated and complex, it is harder to ensure speed and responsiveness. For example, a sizeable fraction of content displayed by mobile applications is fetched in real time from the internet, because, e.g., it is not already available on the user device. To improve speed and responsiveness, applications attempt to anticipate user action and pre-fetch content. Techniques of this disclosure determine the screen area where a user is focusing by using gaze detection techniques, for users that provide permission for such detection. The gaze detection is performed using a front facing sensor of a user device and a model, e.g., a trained machine-learning model, or heuristics. With user permission, detected gaze data is provided to applications that can utilize such data to prefetch content, thereby optimizing usage of the data network and allowing for faster app interaction. Gaze detection can be implemented in an operating system, or within an application.

KEYWORDS

- Gaze detection
- Attention detection
- Pre-fetching
- Operating system
- Gaze API

BACKGROUND

As software applications become increasingly sophisticated and complex, it is harder to ensure speed and responsiveness. For example, a sizeable fraction of content displayed by mobile applications is fetched in real time from the internet, because, e.g., it is not already available on the user device. For example, a video-streaming application fetches videos on demand, once the user selects a video, e.g., by selecting a thumbnail. Similarly, a mapping application stores only certain zoom-levels and sections of a map on a user device, and fetches additional content when the user selects or zooms into other regions of the map. A web browser application fetches linked pages or not-yet visible parts of web pages only upon the user selecting those portions, e.g., by clicking (or tapping on) a link, scrolling the page, etc.

One strategy to speed up user interaction is to anticipate user action and pre-fetch content that the user is likely to need soon. Traditionally, this requires considerable knowledge of user interaction with apps in general, and with the specific app in particular.

DESCRIPTION

Techniques of this disclosure describe gaze detection, e.g., implemented in an operating system, browser, or other application. When users permit gaze detection to be enabled, a trained machine learning model and the front-facing camera or other sensor, e.g., of a user device such as a smartphone, tablet, laptop, etc., are used to determine the part of the currently displayed screen content the user is focusing on. Further, a gaze detection application programming interface (API) is provided, e.g., by the operating system. Apps for which the user has granted permission to use gaze detection data are granted access to the API to request gaze information, e.g., for use to optimize content pre-fetching. Per techniques of this disclosure, gaze detection is implemented in two parts:

1. a machine learning model that infers parts of the screen of current user focus from images generated by a front-facing camera or another sensor; and
2. an API that provides gaze data, e.g., screen areas of current user focus to applications that have been permitted by the user to access such data.

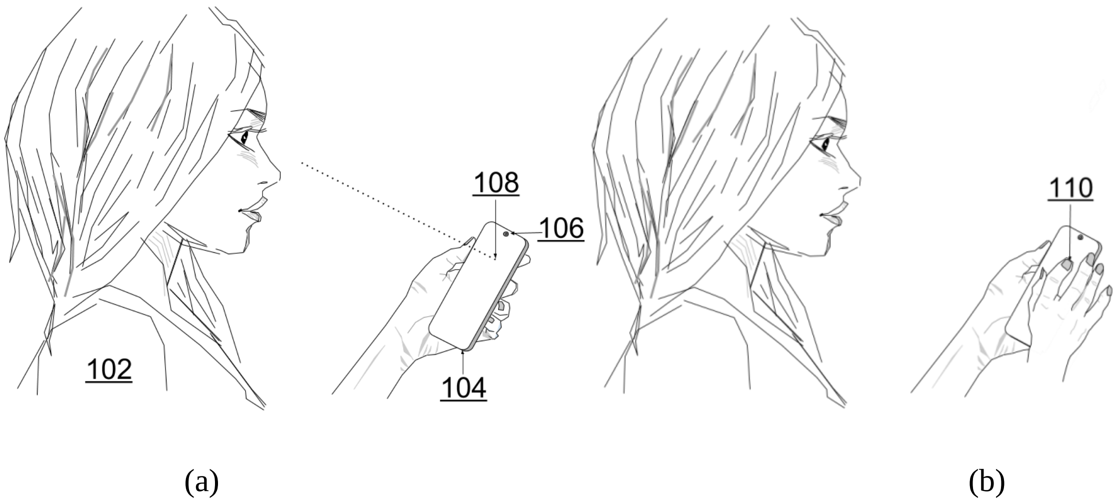

Fig. 1: Machine learning to infer user focus (a) User's gaze on a certain region of the screen is followed by (b) interaction with that region of the screen

Fig. 1 illustrates the use of a machine learning model to infer user focus, with user's consent and permission. The techniques are implemented using images and data from a front-facing camera and/or other sensors. For example, one or more front facing cameras can be used. In another example, a three-dimensional face detection sensor or sensor array can be used. Such sensors or sensor arrays obtain data including depth values corresponding to the user's face. In some examples, when users permit, a three-dimensional model of the user's face may be generated based on data obtained from the front-facing camera and/or sensors.

As illustrated in Fig. 1(a), a user (102) is gazing at a particular region (108) of mobile device (104). An image of the user looking at the screen is obtained from the front-facing camera (106). Shortly thereafter, as illustrated in Fig. 1(b), the user interacts (110) with the device, e.g., by touching the region of the screen just gazed upon.

With user permission, data from the front-facing camera or other sensor, together with the interaction data, are used as training data for a machine learning model. Having trained on such data, a machine learning model, e.g., a deep neural network, a convolutional neural network, etc., can determine user gaze from an image of the user generated by the front-facing camera or another sensor.

Alternately, training data may be obtained using a dedicated application for this purpose. Such an application instructs the user (e.g., a user who volunteers to provide training data, or a user that consents to use of such data for training) to focus on a pre-selected and known region of the screen. Using additional hardware, such as special glasses, the focal point of the eyes of the user is determined and communicated to the application. In this manner, accurate measurement of user gaze along with data from front facing camera or another sensor is obtained. A machine-learning model is trained based on such data.

The trained machine-learning model is deployed on the user device when user provides consent for gaze detection. The final output layer of the machine-learning model provides user-focus scores for screen areas, e.g., for each pixel or abstraction thereof, e.g., subsampled sections of the screen). For example, the model predicts a user-focus score as a number between 0 (no focus at all) and 1 (total focus) for each pixel or region (e.g., 10x10 pixels). Gaze detection can also be performed using heuristics, e.g., instead of a machine-learning model.

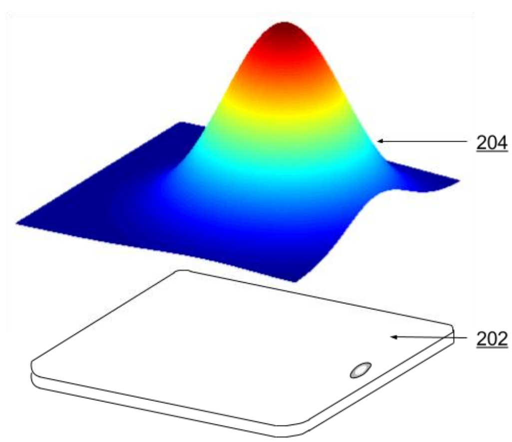

Fig. 2: An example heat map generated by the machine learned model indicating regions of likely user focus

Either at regular intervals or upon request from a running application, techniques of this disclosure obtain images from the output of the front-facing camera or other sensor and provide the image data to the trained machine learning model. The model provides as output, the user's current focus area. For example, the output is in the form of a heat map over pixel space, as illustrated in Fig. 2. In Fig. 2, heat map (204) represents the user-focus score for different portions of the screen (202) of a mobile device. In Fig. 2, the red regions represent regions with higher user-focus scores, near or equal to 1, and blue regions representing areas with lower user-focus score, e.g., near or equal to zero.

In operation, the gaze detection API can provide heat map to an application that requests gaze data. Alternatively, the techniques use the current view hierarchy from the requesting application, and provide user-focus scores relative to the different views. For example, the boundaries of all visible views are determined, and the average user-focus score of each view within its boundary is reported to a calling application. Using output generated by the techniques herein, an application can sort views, pixels, or screen subsections by gaze score, and take actions, e.g., pre-fetch content over a network.

Aside from apps using the API to prefetch content, the API can also be used in other situations with user's consent and permission, e.g., to trigger other anticipatory tasks such as pre-rendering, to assess user attention, e.g., when viewing online advertisements, to trigger re-marketing, to improve precision and/or recall for content recommendation systems based on assessed attention to particular content, to improve the responsiveness and accuracy of a virtual assistant, to reduce typing errors when a user uses a predictive software keyboard by using user-focus scores on keys as prior probabilities within the decoder of such a keyboard, etc.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user's social network, social actions or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

Techniques of this disclosure determine the screen area where a user is focusing by using gaze detection techniques, for users that provide permission for such detection. The gaze detection is performed using a front facing sensor of a user device and a model, e.g., a trained machine-learning model, or heuristics. With user permission, detected gaze data is provided to applications that can utilize such data to prefetch content, thereby optimizing usage of the data network and allowing for faster app interaction. Gaze detection can be implemented in an operating system, or within an application.