User Gaze Detection on Arbitrary Objects Using Physical Sensors and an Eye Tracker in a Real Environment

Takumi Toyama, Thomas Kieninger, Faisal Shafait, Andreas Dengel
German Research Center for Artificial Intelligence (DFKI) GmbH
Trippstadter Strasse 122, Kaiserslauern, Germany
(firstname.lastname)@dfki.de

ABSTRACT
Recent advances of mobile eye tracking technologies open up the possibilities of gaze-based human-computer interaction systems in a real environment. Since eye movements supply us with dynamic visual information, detection of gaze (observed when the person is looking at a specific region for a certain time) facilitates the system to trigger specific events when user attention is recognized. In order to detect such user gaze in a real environment, the existing approaches typically use image based object recognition methods. Such an approach limits the capability of the application because it is not applicable to unknown objects, for which the system has not been trained. This paper presents a method to detect the user gaze on arbitrary objects by using physical sensors in combination with an eye tracker. Experimental results show that the performance of the proposed method is comparable to the existing image based method but expands the applicability to arbitrary objects. Furthermore, we present a prototypical application that makes use of the method proposed in this paper to demonstrate the adaptability of this method to an open scenario.

Author Keywords
Eye tracking, gaze detection, user attention, object recognition, activity recognition.

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation (e.g. HCI): User Interfaces

General Terms
EXPERIMENTAL

INTRODUCTION

Over several decades, a number of studies have been conducted to understand the nature of human attention by analyzing human gaze [6, 8, 10, 17]. As by-products of these studies, many researchers have proposed human-computer interactive frameworks using human gaze as an interface for controlling the computer [3, 4, 12].

Typically, gaze plays a role similar to that of ordinary computer interfaces like as a computer mouse or a keyboard in these frameworks, which vary from gaming [3], controlling a web-browser [4], to typing [12]. These approaches mostly rely on developments of human-computer interaction methods using a desk-mounted eye tracker, which observes the user's eye movement on a computer display. Besides this, several applications have also been proposed using a head-mounted (mobile) eye tracker, such as [5, 9, 16]. Authors in [14] describe an approach for developing eye-controlled environments in a house. Furthermore, a blueprint of a home automation system in the near future is discussed in [5]. In [9], how human gaze can be used for *Lifelog* systems is presented. In this work, face recognition, OCR and object recognition are integrated with the system in order to recognize what is being watched by the user. In addition, we have presented in [16] how the user gaze on a specific object can be detected in order to provide an automatic audio guidance of the exhibits in a museum. These state-of-the-art applications show real potential of gaze-based human-computer interaction, particularly by inferring user attention from gaze.

The existing systems in those real environment scenarios stated above typically rely on image processing based object recognition mechanisms in order to detect at which object or position in the scene the user is gazing. However, such types of object recognition based approaches hold two crucial drawbacks. First, a known set of objects for a database is always required in order to recognize an object. Most of the systems stated above need a pre-defined object database in order to match local features, such as SIFT [11] or SURF [2] which are extracted from an image. Thus, these systems cannot deal with unknown objects, which might appear frequently in a real environment. Secondly, even though advances of the recent hardware relax the restriction of computational expense, image processing still consumes high computational costs, particularly for object recognition with a large dataset. Thus, it is hardly applicable to a real-time scenario.

This paper presents a method to detect the user gaze on particular objects or regions without using object recognition methods. Instead, we analyze the number of fixations in a particular region in a scene by combining motion data from other sensors such as accelerometer, gyroscope and mag-

Leave in blank
APCHI 2012, August 28–31, 2012, Matsue, Japan.

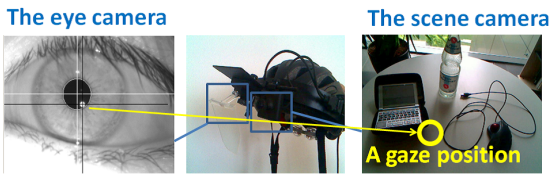

Figure 1. The SMI iViewX^{TM} HED. The left image shows a sample image of the eye camera and the right shows the image of the scene camera. For a transparent view, a special mirror is used.

netic compass with eye tracking data. An ordinary mobile eye tracker provides the gaze position as a point in an individual scene image. In this work, the sensor data is used to obtain the relative positions between each gaze position. By computing the spatial orientation of each gaze sample and aggregating this data in a *global gaze map*, the method finds the frequently fixated regions, which are considered as a gaze on a particular object. This way, the method can detect gaze on arbitrary objects without recognizing the corresponding objects.

Additionally, the sensor data is also used to recognize the user activity. This enables the system to treat the gaze behaviour differently according to the user activity. We only focus on two types of activities, *walking* and *standing*[1]. A *global gaze map* is created only when the user is in a *standing* state because the accurate relative gaze position is hardly available when the user is walking.

We compare this method with the method proposed in [16] to show this method can reasonably detect user gaze compared to the object recognition based method while it is also applicable to other arbitrary objects. In addition, in order to demonstrate in which application this method can be used, we present a *visual diary* system that provides the user with a collection of the pictures that attracted the user's interest in his daily life.

APPARATUS

Figure 1 and Figure 2 show pictures of the SMI iViewX^{TM} HED [2] we used to obtain user gaze in a scene. It provides us the gaze position on the image from the scene camera.

We also mounted an Android phone onto the eye tracker to use the sensor integrated in the phone. It is connected to a laptop computer via a USB cable and sends data of its acceleration and cardinal direction by using an accelerometer, a gyroscope and a magnetic compass. The phone must be fixed on the top of the helmet firmly so that the sensor can collect the data properly.

SMI has recently released a new glasses-type wearable eye tracker which is more flexible and easy to handle than the former one. A picture of the new eye tracker is shown in Figure 3. The apparatus we used in this paper can soon be succeeded by the new one, which is more convenient and comfortable with combination of the sensor board instead of the phone. However, the technical basis we propose in this paper works same with the new apparatus.

[1] *standing* also includes the activity that the person is sitting

[2] http://www.smivision.com/en/gaze-and-eye-tracking-systems/products/iview-x-hed.html

Figure 2. The Android phone mounted on the eye tracker helmet. the phone is tied up with a plastic cable and two cushions are put between the phone and the helmet.

Figure 3. The new eye tracking device (top) and the sensor board (bottom). The board can be attached to the side of the eye tracker.

METHODS

Eye Tracking

Several methods for an eye tracking system have been proposed to find out at which position the user is looking. The iViewX^{TM} employs the dark pupil system. In this system, the user's eye is illuminated by infrared light and the eye camera of this eye tracker captures mirrored images of the illuminated eye. Then, an image analysis software in the system maps the center of the pupil in the image into the scene camera as shown in Figure 1.

Head Movements Tracking Using Sensors

An Android phone provides the APIs to send its acceleration vector and orientation vector from the accelerometer,

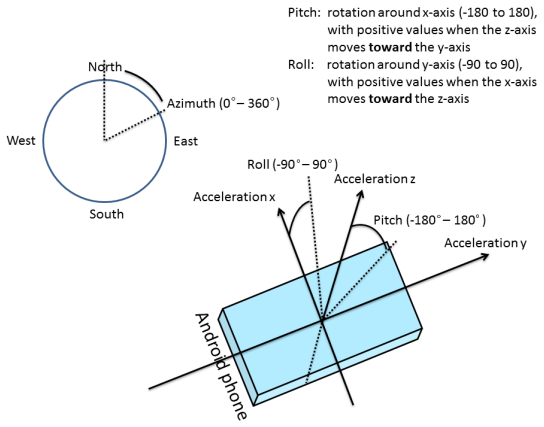

Figure 4. The values of the sensor of an Android phone.

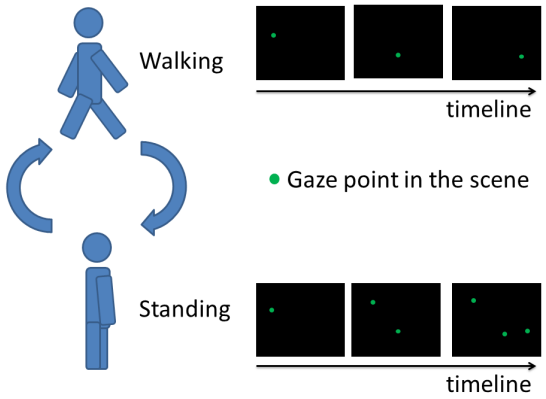

Figure 5. State transition between two activities. When walking activity is detected, the system resets all gaze position in the scene.

gyroscope and magnetic compass. Acceleration vector consists of three values, which correspond to acceleration values of axis x, y and z and orientation vector also consists of three values, which correspond to Azimuth, Pitch and Roll as shown in Figure 4.

These values indicate at which direction the user's head aims and how fast the user's head moves when the movements are occurred. If the acceleration vector is perfectly accurate and data sampling rate is fast enough, we might as well map the absolute spatial position of the user in the scene so that the system can reconstruct the 3D map of the user position and his/her gaze movements. However, the technology available today is not advanced enough to reconstruct such a map only by using an accelerometer and gyroscope, so instead, we adopt an alternative method based on the following observations.

- The gaze movements differ whether the user is walking or standing (or sitting).
- If the user position remains the same, the direction of the user's head measured by the orientation vector has the same origin.

Thus in this method, when "walking" action is detected, the system resets the scene. Only when the user activity is "standing", it aggregates all gaze samples in a scene as a global gaze map as shown in Figure 5 in order to analyze gaze pattern.

Activity Recognition

As stated in the previous section, this method distinguishes walking and standing in order to switch the mechanism for organizing the gaze samples differently. To distinguish these two actions, we use the acceleration vectors obtained from the phone. Although a number of approaches have been proposed for activity recognition to date [1, 13], these approaches were intended to be applied for recognition of several activities (typically more than 8 different activities) and therefore, they adopt relatively complex features and classifiers. Since our intended activities are only two simple ones,

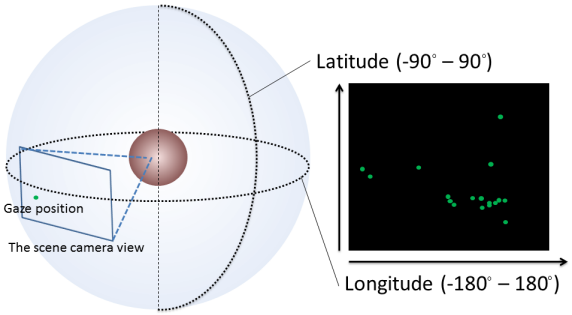

Figure 6. A global map. The globe represents a sphere where the user's head locates the center.

we only use the mean acceleration value of the signals over a period, which is also used as one of the features in [1]. By thresholding the mean value, we classify the activities into walking or standing. The threshold value is obtained from the mean value of the average value of the training samples from each activity.

Gaze Vector Computation in a Scene

While "standing" activity is being detected, the system maps all the gaze samples in a 2-dimensional plane. The plane represents the global map where user's head is centered as shown in Figure 6. The axes are introduced as longitude and latitude, which range from $-180°$ to $180°$ and from $-90°$ to $90°$ respectively. Gaze direction is obtained from the eye tracker and its vector in the 3D space is added to the head direction vector obtained as the orientation vector of the phone as shown in Figure 7. The green dot here represents each sample of gaze. Thus, all gaze samples in the standing state are aggregated in one scene to observe how the user looks at the scene even when the user's head direction has changed.

Gaze Detection

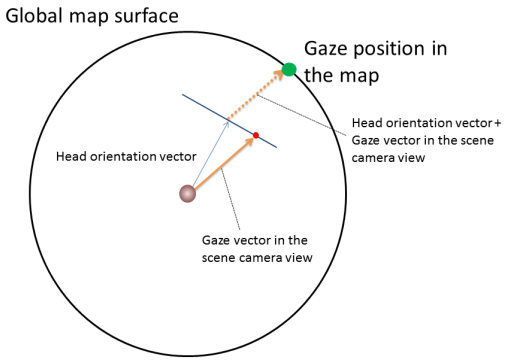

Figure 7. Gaze vector is computed as the sum of two vectors.

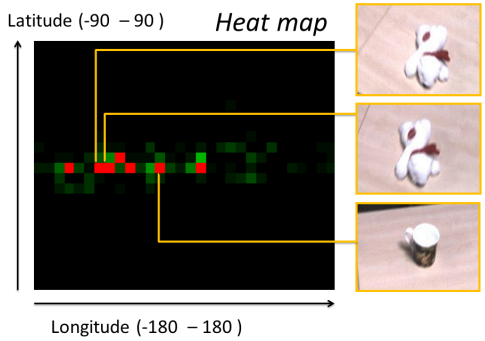

Figure 8. A heat map of user gaze and images from respective cells. Each axis is divided into n_{lat} and n_{lon}. Therefore, the map has $n_{lat} \times n_{lon}$ cells. In this example, $n_{lat} = 25$ and $n_{lon} = 30$. The contiguous cells capture the same object (a toy bear). Red cells are detected gaze regions.

We detect the user gaze by counting the number of gaze samples located in the same region of a global map. Each axis of the global map is divided into n_{lat} and n_{lon} respectively as shown in Figure 8. When the cell $R(n_{lat}, n_{lon})$ has more than T_N samples, the system outputs the region as the detected gaze. The color intensity of a cell represents the number of gaze samples located in the cell. If a gaze event is detected, the color of the cell turns into red. If the detected gaze cell is contiguous to the cell where already gaze exists, these cells are considered to cover the same object. Therefore, if that is the case, the system ignores the event.

Since the proposed method employs simple and light-weight algorithms, it can be run with 15 fps while object recognition based methods may miss important gaze information due to their lower frame processing rates (6 fps).

Image Labeling

Once the gaze on a particular region is detected, the system saves the image of the gazed region. It crops the local region of the image centered around the detected gaze position. The saved images contain the objects that drew the user attention. The labeling of the image can be done either manually by the user or automatically by using an image retrieval engine, such as Google Goggles[3] or IQ Engines[4]. Especially, IQ Engines provides APIs to query an image from our own programs. In our following visual diary system, we use our own image retrieval system used in [16] in combination with IQ Engines to get a label of the image. Note that we only use the image retrieval system in order to get the label of the image but the detection of gaze is done without using any image retrieval (object recognition) methods.

EXPERIMENTS AND EVALUATION

In order to evaluate how the proposed method efficiently detects user gaze in real environments, we compare the proposed gaze detection method with the object recognition based method presented in [16]. In that paper, we evaluated whether the system can find the user gaze when the user looks at the object for a certain duration. Here, we acquire training data for object recognition and test data in the same way as in [16].

The point of this experiment is only the comprison of gaze detection method, which have nothing to do with object recognition in our proposed method. Therefore, note that even though we apply the same object recognition method as in [16] in order to acquire the label of the object for each detected image and compare it with the ground truth, it does not affect the performance of gaze detection.

Data Acquisition

First, we put ten different objects well spaced-out on a table. Then, we asked ten test persons to wear the eye tracker and to look at the objects naturally, that is, to look at the objects with a certain attention if the object is interesting or otherwise just give a glance. After labelling the recorded video frames as the identity of the object being indicated by the user gaze, the ground truth of the user gaze are obtained by using the same method as described in [16]. The ground truth data consists of the frame number of the beginning of the gaze, the frame number of ending of the gaze and the label of the object. Evaluation is done by checking whether the system can detect the gaze on the labelled object during the period indicated by the beginning and the ending. The total gaze events obtained in this experiments were 72.

In order to test with realistic conditions, we simulate a real-time environment. All the experiments are done by sending video frames with the same speed of the scene camera sampling rate to the gaze detection system. The sample rate of the scene camera of the eye tracker and the eye tracking was 25 fps. The sensor provides data immediately when a motion is detected. All the experiments were done on an Intel Core i5 M560 2.67GHz CPU with 8GB RAM.

The Conventional Method

The conventional method applies an object recognition process to the image when it has a gaze position. Since the

[3]http://www.google.com/mobile/goggles/
[4]http://developer.iqengines.com/

eye tracker does not always provide the gaze position due to several reasons, such as the user's blink or the failure of the image processing, only when the gaze position is available, object recognition is done. However, if the system runs in a real-time environment, it misses some gaze positions during the processing. Thus, all the frames are not necessarily processed even if the frames have the gaze position.

This method counts the number of frames that have the same label of object recognition result. When the number of such frames reaches a threshold value while accepting a certain number of noise frames, the system outputs as the result that the user is gazing at the object.

Results

Figures 9 and 10 show the results of gaze detection. Figure 9 shows the system recall rates, which indicate to what degree the system can detect the manually labelled ground truths. In these graphs, the results for different combinations of n_{lon} and n_{lat} are shown respectively. The horizontal axes represent T_N value. As shown in this graph, as T_N value increases, the recall rates drops gradually. The exceptions are $n_{lon} = 20, n_{lat} = 15$ where $T_N = 6$ and $n_{lon} = 40, n_{lat} = 30$ where $T_N = 10$, that the recall rate is lower than others. There are two possible reasons for that. First, when T_N value is small, the system outputs more regions as gazed regions. Therefore, since the method treats contiguous cells as an identical gaze region, if one cell is recognized as gaze and the recognition fails (or the result is rejected), all the contiguous cells cannot be detected as gaze even if it is actual gaze. Secondly, the larger region is covered by one cell, that is the smaller n_{lon} and n_{lat} are, the more gaze events are aggregated as one gaze event. Hence, when a region is too large, two distinct gaze events are not distinguishable.

Next, the system precision rates are shown in Figure 10. Theses rates indicate how precise the detections are. The results show that when T_N value is too large, the precision rate starts to decrease. This is mainly because of the failure of object recognition which is caused by the selection of different frames from the video. The system waits until the number of gaze samples mapped in a particular cell reaches T_N and then it picks up the frame for a query image. Therefore, sometimes the recognition fails even if the same object is appearing in frames when the images are distorted by several factors such as image blur, which frequently occurs when the user's head moves.

The precision rate and the recall rate of the conventional method were both 0.61. This is slightly better than the result from the proposed method, whose recall rate was 0.58 and precision rate was 0.64 when $T_N = 8$ and $n_{lon} = 30, n_{lat} = 25$. However, the results show the proposed method is still competitive even in the limited scenario that set of objects are all-known.

VISUAL DIARY - A PROTOTYPICAL APPLICATION

Our previous work focused on a museum scenario [15], in which we can obtain images of all objects. The goal of the

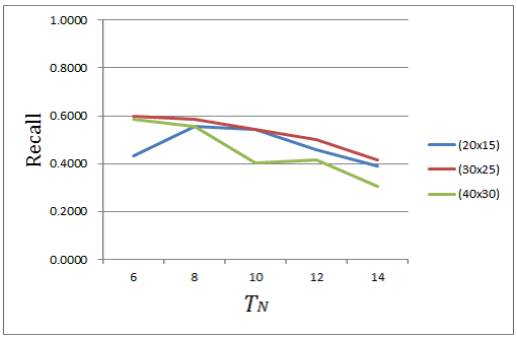

Figure 9. Recall rates for the proposed method for different cell sizes. Each curve represents a particular $(n_{lon} \times n_{lat})$ pairs. The values mostly decrease as T_N increases.

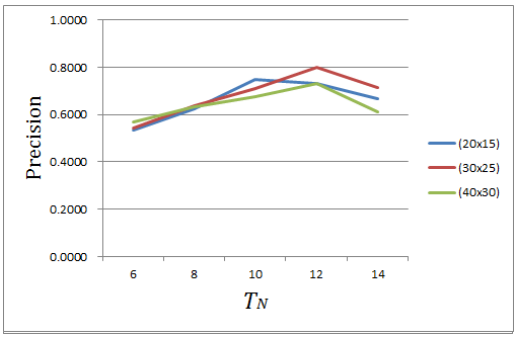

Figure 10. Precision rates for the proposed method. The graphs peak where $T_N = 10$ and $T_N = 12$.

Figure 11. A screenshot of the visual diary prototype. This application shows the image from the scene camera (the left), the detected gaze image (the right-most), the selected image by the user and the tag of the image. This tag (Whiteboard) was obtained from IQ Engines. The user can also edit tag here.

new method is to open up such a closed environment in order to increase the possibility of gaze-based interfaces. By detecting gaze on arbitrary objects with this method, we also build a prototype of a new application. Lifelog [7] systems have got much more attention in recent years. Gaze detection plays a quite important role in such a system. In this paper, we show a screenshot of *visual diary system*, a form of lifelog system, that provides the user with a collection of images of objects which drew the user attention in Figure 11. The images in the right column show the detected objects and they are automatically tagged if the recognition is successfully done. First, the image is matched with a local reference database. Then, if the respective result cannot be found in the local database, the image is sent to IQ Engines to get the label. Otherwise, the user can also tag on his/her own.

CONCLUSION

This paper introduced a method to detect the user gaze on objects without using object recognition based approaches so that the method is adaptable for a wide variety of applications without the restriction of an object image database. The experimental results clearly show that the proposed method is competitive to the existing method. Furthermore, we also presented a prototypical application using the proposed method.

Future work is to expand the activity recognition in order to apply the method to other scenarios that contain more user activities and to analyze the user gaze to recognize which type of object is being paid attention by the user.

ACKNOWLEDGMENTS

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project Perspecting (01 IW 08002). We also wish to thank SensoMotoric Instruments (SMI) GmbH for providing us with an iView X$^{\mathrm{TM}}$ HED mobile eye tracker for building the application and conducting all the experiments.

REFERENCES

1. Bao, L., and Intille, S. S. Activity recognition from user-annotated acceleration data. *Pervasive Computing LNCS 3001* (2004), 1–17.
2. Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. Speeded-up robust features (surf). *Comput. Vis. Image Underst. 110* (June 2008), 346–359.
3. Bee, N., Wagner, J., Andre, E., Charles, F., Pizzi, D., and Cavazza, M. Interacting with a gaze-aware virtual character. In *Proceedings of the International workshop on eye gaze in intelligent human machine interaction* (Hong Kong, China, 2010).
4. Biedert, R., Buscher, G., Schwarz, S., Hees, J., and Dengel, A. Text 2.0. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems* (Atlanta, USA, 2010), 4003–4008.
5. Bonino, D., Castellina, E., Corno, F., Gale, A., Garbo, A., Purdy, K., and Shi, F. A blueprint for integrated eye-controlled environments. *Universal Access in the Information Society 8*, 4 (2009), 311–321.
6. Buswell, G. T. *How people look at pictures*. The University of Chicago press, Chicago, 1935.
7. Byrne, D., Doherty, A., C.G.M, S., Jones, G., and Smeaton., A. F. Everyday concept detection in visual lifelogs: Validation, relationships and trends. *Multimedia Tools and Applications Journal* (2009).
8. Henderson, J. M. Human gaze control during real-world scene perception. *Trends in Cognitive Sciences 7*, 11 (2003), 498–504.
9. Ishiguro, Y., Mujibiya, A., Miyaki, T., and Rekimoto, J. Aided eyes: eye activity sensing for daily life. In *The 1st Augmented Human International Conference (AH2010)*, H. Saito, J.-M. Seigneur, G. Moreau, and P. Mistry, Eds., ACM (2010), 25.
10. Kovic, V., Plunkett, K., and Westermann, G. Eye-tracking study of animate objects. *Psihologija 42*, 3 (2009), 307–327.
11. Lowe, D. Object Recognition from Local Scale-Invariant Features. In *International Conference on Computer Vision* (Kekyra, Greece, September 1999), 1150–1157.
12. Majaranta, P., Ahola, U.-K., and Špakov, O. Fast gaze typing with an adjustable dwell time. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, ACM (Boston, MA, USA, 2009), 357–360.
13. Reiss, A., Weber, M., and Stricker, D. Exploring and extending the boundaries of physical activity recognition. In *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (oct. 2011), 46 –50.
14. Shi, F., Gale, A., and Purdy, K. A new gaze-based interface for environmental control. In *Proceedings of the 4th international conference on Universal access in human-computer interaction: ambient interaction*, UAHCI'07, Springer-Verlag (Berlin, Heidelberg, 2007), 996–1005.
15. Toyama, T., Kieninger, T., Shafait, F., and Dengel, A. Museum guide 2.0 - an eye-tracking based personal assistant for museums and exhibits. In *Re-Thinking Technology in Museums 2011: Emerging Experiences*, L. Ciolfi, K. Scott, and S. Barbieri, Eds., University of Limerick (5 2011).
16. Toyama, T., Kieninger, T., Shafait, F., and Dengel, A. Gaze guided object recognition using a head-mounted eye tracker. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, ACM (New York, NY, USA, 2012), 91–98.
17. Yarbus, A. L. Eye movements and vision. *Plenum Press* (1967).