Defining Gaze Interaction Events

Jari Kangas, Oleg Špakov, Päivi Majaranta, Jussi Rantala, Poika Isokoski, and Roope Raisamo
Tampere Unit for Computer-Human Interaction
School of Information Sciences
University of Tampere, Finland
{jari.kangas, oleg.spakov, paivi.majaranta , jussi.e.rantala, poika.isokoski, roope.raisamo}@uta.fi

Abstract
In this position paper we argue of the need for a more complete definition of gaze interaction events. The need is especially clear in non-desktop use scenario. We ask on one hand what the needed gaze events are and on the other hand what can eye tracking technology possibly deliver in dynamic everyday settings. Current understanding on gaze events works well for static scenes often encountered in laboratory studies and desktop computing, but we need to expand or modify the notions when moving out from lab. We need a better understanding of what is meant by gaze events and what kind of events need to be recognized to push forward the utilization of gaze tracking technology.

CHI 2013 Workshop on "Gaze Interaction in the Post-WIMP World", April 27, 2013, Paris, France.
ACM 978-1-4503-1952-2.

Author Keywords
Gaze tracking; gaze event; gaze interaction.

ACM Classification Keywords
H.5.2. User interfaces: Input devices and strategies.

General Terms
Human Factors; Design.

Introduction
There is a long history of gaze tracking technology research [6,11], and a number of different use cases and applications have been identified for eye movement data. For examples, see, [3,6,9,11].

In some areas, like marketing, usability and user experience studies, gaze tracker is a common tool to analyze user behavior. Gaze tracker is used also in neuroscience and psychology to get insights of physiological and cognitive processes. In these applications the gaze tracker is usually passive as it records eye movements for later off-line analysis.

A gaze tracker can also be an active component in real-time systems, where gaze data provides input for human-computer interaction. Eyes are then often used intentionally to point or select [5,9].

Another important active use is to observe and assess the user's cognitive context from his/her gaze data [1].

This information could be used, for example, in interactive advertising that changes content depending on products that were just looked at and interested the user.

All the examples described above would require the gaze tracker and other parts of the system to generate relevant gaze interaction events. In some cases, like when moving outdoors, a very rich set of events is available, of gaze related to user, environment, and objects of interest.

Most gaze trackers are still not mass produced and, therefore, expensive. It is expected, however, that the technologies will mature and the price will become reasonable for increased use (e.g. [7,8]).

In the next section we discuss the use of gaze interaction in different environments. Gaze interaction events are important, and need to be defined with care.

Need for Gaze Interaction Events

An active, gaze tracker based system would consist of several parts. The gaze tracker itself would locate the eyes and measure the gaze direction. If display is used, some parts of the system would generate graphics for it, while other parts would handle the keyboard and other accessories. Still other parts would relate the position and pose of the user to the environment and objects. The logic of application would run separately, collecting events from all these parts, changing internal states and sending commands, as needed.

The current definitions of gaze interaction events work well when in front of a fixed display (e.g., GazeEnter and GazeLeave in [12]). As we know the layout of the objects, the gaze interaction events can easily be defined similarly to mouse events.

However, if we use a gaze tracker when the view is not fixed, gaze interaction events are different. We still know and notice the direction and movements of the gaze. However, eyes may be tracking moving objects and the object fixated on may change without eye movements.

A natural member of the set of low-level events is the gaze fixation [10]. Another often used event type is the gaze gesture, which consist of a sequence of distinct gaze directions [4]. The gestures can be defined without a reference location [2], or tied to a certain object or locations in environment [5]. The active analysis of context information from the user's eye gaze information [1] might also generate gaze events.

In an everyday setting, say, a person sitting and looking a moving car passing by, the gaze tracking system would give, at least, the following events at regular intervals: GazeDirection, GazeMove (as eyes move rel. to head), GazeEnvMove (as gaze moves with car), GazeObjectOver (as gaze aligns with object), GazeObjectContinue (as gaze is still fixated).

Categories of Gaze interaction events

The following table is one categorization of gaze interaction events that we are working on for a generic use of gaze trackers:

Generic events based only on eye properties and dynamics	**Head centric** events recognized by gaze tracker, directions relative to head of user	**Environment centric** events where gaze direction relative to static environment, needs some external sensors	**Object centric** events where gaze direction relative to objects in view, can be static on display, can be moving objects
EyesOpen, eyes are found and open	**GazeFound**, gaze direction has been determined		
EyesClosed, eyes are closed (not found)	**GazeLost**, gaze direction has been lost		
	GazeDirection, regular update of gaze direction	**GazeEnvDirection**, regular update of gaze direction	
	GazeMove, direction changed	**GazeEnvMove**, direction changed	
	GazeDwell, triggered when gaze has stayed still for predefined duration	**GazeEnvDwell**, gaze still on environment for predefined duration	**GazeObjectDwell**, gaze stays still on object for predefined duration
	FixationStart, gaze on one direction long enough, can be noticed after a delay	**FixationEnvStart**, gaze on one direction for long enough	**FixationObjectStart**, gaze on one object for long enough
	FixationContinue, gaze stays on same direction	**FixationEnvContinue**, gaze stays on same direction	**FixationObjectContinue**, gaze stays on same object
	FixationEnd, gaze changes direction	**FixationEnvEnd**, gaze changes direction	**FixationObjectEnd**, gaze direction moves from object
			GazeObjectEnter, gaze direction becomes aligned with an object
			GazeObjectLeave, gaze direction moves from object
			GazeObjectOver, gaze direction is aligned with object
	GazeGesture, tracker recognizes a gaze pattern	**GazeEnvGesture**, gaze pattern or gesture	**GazeObjectGesture**, gaze pattern or gesture on object

Conclusion

Plenty of work has been done on gaze tracker technology. The technology is now reaching the state where it will move from laboratories to wider use, and we need to prepare for that. The applications, so far, have been mostly restricted to rather small trials and very tightly defined use cases, and they have usually used a small set of situation-specific definitions of gaze interaction units, events.

In this paper we want to discuss possibilities for better formalization of gaze interaction events especially in dynamic situations. What types of gaze events are needed now and in the foreseeable future? What types of software components and interfaces would be useful in general?

Acknowledgements

We thank the members of TAUCHI unit who provided helpful comments on different versions of this paper. This work was funded by the Academy of Finland, project Haptic Gaze Interaction (decisions numbers 260026 and 260179).

References

[1] Bulling, A., Roggen, D., and Tröster, G. What's in the Eyes for Context-Awareness? *IEEE Pervasive Computing*, 10(2), 48-57, 2011.

[2] Drewes, H., and Schmidt, A. Interacting with the Computer Using Gaze Gestures. In: *INTERACT 2007*, LNCS 4663, Part II, pp- 475-488, 2007.

[3] Duchowski, A.T. A Breadth-First Survey of Eye-Tracking Applications. *Behavior Research Methods, Instruments, & Computers*, 2002, 34 (4), 455-470.

[4] Isokoski, P.: Text input methods for eye trackers using off-screen targets. *ETRA '00*, 15–21. ACM Press, 2000.

[5] Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., and Vickers. S. Designing Gaze Gestures for Gaming: an Investigation of Performance. *ETRA 2010*, pp. 323-330, ACM Press, 2010.

[6] Jacob, R.J.K. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems (TOIS)*, 9(3), 152-169, 1991.

[7] Lemahieu, W. and Wyns, B. Low cost eye tracking for human-machine interfacing. Journal of Eyetracking, Visual Cognition and Emotion, 1(1), 1-12, 2011.

[8] Li, D., Babcock, J., and Parkhurst, D. openEyes: a low-cost head-mounted eye-tracking solution. *ETRA'06*, 95-100, ACM Press, 2006.

[9] Majaranta, P., and Räihä, K.-J. Twenty Years of Eye Typing: Systems and Design Issues. *ETRA'02*, 15-22, ACM Press, 2002.

[10] Salvucci, D.D. and Goldberg, J.H. Identifying Fixations and Saccades in Eye-Tracking Protocols. *ETRA '00,* 71 – 78, ACM Press, 2000.

[11] Ware, C., and Mikaelian, H.H. An Evaluation of an Eye Tracker as a Devices for Computer Input. *Proc. CHI + GI'87*, 183-188, ACM Press, 1987.

[12] Wassermann, B., Hardt, A. and Zimmermann, G. Generic Gaze Interaction Events for Web Browsers – Using the Eye Tracker as Input Device *WWW2012 Workshop: Emerging Web Technologies, Facing the Future of Education*, 2012.