Building Implicit Interfaces for Wearable Computers with Physiological Inputs: *Zero Shutter Camera and Phylter*

Tomoki Shibata, Evan M. Peck, Daniel Afergan, Samuel W. Hincks, Beste F. Yuksel, Robert J.K. Jacob
Human-Computer Interaction Laboratory @ Tufts University

Tufts UNIVERSITY | School of Engineering

Abstract

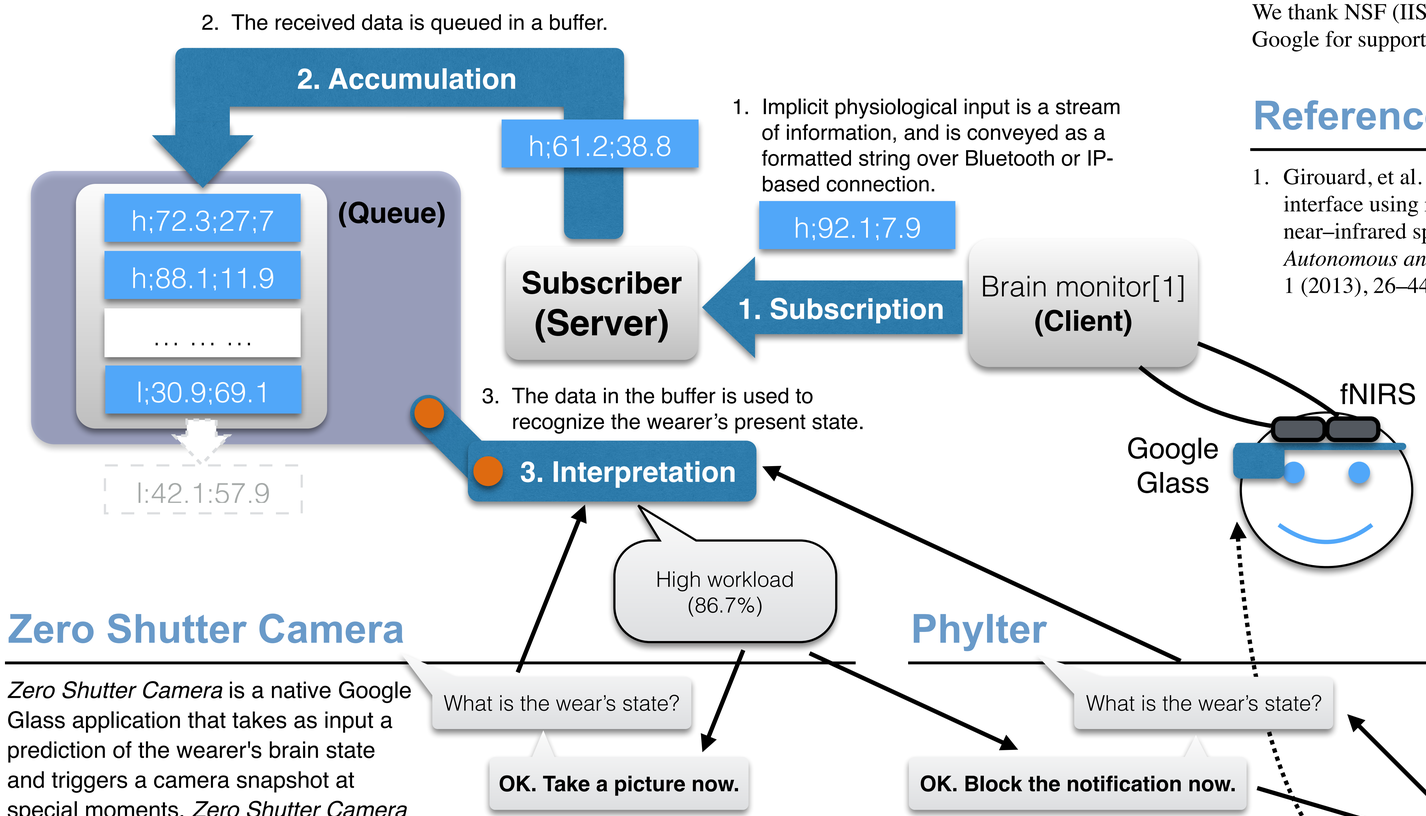
- **We propose implicit interfaces** that use passive physiological input as additional communication channels between wearable devices and wearers.
- **We introduce a core framework** to support building implicit interface, such that the framework follows the three key principles: Subscription, Accumulation, and Interpretation of implicit inputs.
- Unlike a conventional event driven system, our framework subscribes to continuous streams of input data, accumulates the data in a buffer, and subsequently attempts to recognize patterns in the accumulated data -- upon request from the application, rather than directly in response to the input events.
- To embody the impacts of implicit interfaces in the real world, we introduce two prototype applications for Google Glass, ***Zero Shutter Camera*** and ***Phylter*** the both leverage the wearer's physiological state information.

Key Contribution

We consolidate a core framework aiming to accommodate implicit input for building implicit interfaces. The core framework adhere to *the three key principles*:

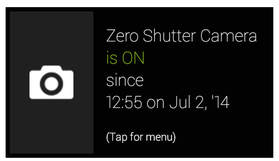
1. ***Subscription***: Ability to continuously receive information since implicit physiological input is a stream of information. (Client-Server architecture over a Bluetooth, IP-based or even wired connection would be applicable to achieve this.)
2. ***Accumulation***: Ability to hold the received information for a certain duration in order to allow pattern recognition over recent past data.
3. ***Interpretation***: Ability to recognize the wearer's present physiological state based on the accumulated information when the wearable computer asks. (Since there may be more than one implicit channel, this ability is also responsible to encompass all implicit input channels.)

Core Framework Implementation



Zero Shutter Camera

Zero Shutter Camera is a native Google Glass application that takes as input a prediction of the wearer's brain state and triggers a camera snapshot at special moments. *Zero Shutter Camera* determines when to take pictures by continuously monitoring the wearer's physiological state over a Bluetooth connection.



(The start screen of *Zero Shutter Camera*)

Zero Shutter Camera is made open source located at **https://github.com/zshiba/zero-shutter-camera**

How *Zero Shutter Camera* works:

1. Subscribes continuous state classification data.
2. Accumulates at least 10 [sec] of the immediate data.
3. Interprets the quantified confidence value [%] of the wearer being in a high or low workload state by averaging the accumulated data.
4. Takes a picture for the wearer if the wearer is in high workload state with a confidence value above a heuristic threshold; otherwise, keeps monitoring the wearer's state.

Phylter

Phylter is an intermediate software between notification senders and Google Glass to schedule the delivery of notifications by using predictions of the wearer's state as input. *Phylter* assesses the interruptibility of the user based on wearer's current brain state, ultimately deciding if she has the cognitive resources available to handle a notification of a known level of importance.

How *Phylter* works:

1. 2. and 3. Treats the data in exactly the same way as *Zero Shutter Camera*.
4. Whenever a notification comes from a notification sender, inspecting the wearer's current state. If the wearer has high cognitive workload with a heuristic threshold value, then blocks the notification; otherwise, delivers it to Google Glass.

(Google Glass)

Acknowledgments

We thank NSF (IIS-1065154 and IIS-1218170) and Google for supporting this research.

Reference

1. Girouard, et al. Designing a passive brain computer interface using real time classification of functional near–infrared spectroscopy. I*nternational Journal of Autonomous and Adaptive CommunicationsSystems 6*, 1 (2013), 26–44.

ACM USER INTERFACE SOFTWARE AND TECHNOLOGY SYMPOSIUM, UIST 2014, October 5–8, 2014, Honolulu, HI, USA.
http://dx.doi.org/10.1145/2658779.2658790