
Shiny – An Activity Logging Platform for Google Glass

Shoya Ishimaru

Osaka Prefecture University
Sakai, Japan
ishimaru@m.cs.osakafu-u.ac.jp

Jens Weppner

DFKI
Kaiserslautern, Germany
jens.weppner@dfki.de

Andreas Poxrucker

DFKI
Kaiserslautern, Germany
andreas.poxrucker@dfki.de

Kai Kunze

Osaka Prefecture University
Sakai, Japan
kai.kunze@gmail.com

Paul Lukowicz

DFKI
Kaiserslautern, Germany
paul.lukowicz@dfki.de

Koichi Kise

Osaka Prefecture University
Sakai, Japan
kise@cs.osakafu-u.ac.jp

Abstract

We describe an activity logging platform for Google Glass based on our previous work. We introduce new multi-modal methods for quick non-disturbing interactions for activity logging control and real time ground truth labeling, consisting of swipe gesture, head gesture and laser pointer tagging methods. The methods are evaluated in user studies towards estimating their effectiveness.

Author Keywords

Activity Recognition; Platform; Google Glass;
Ground-truth Labeling

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces, Graphical User Interfaces

Introduction

Moving towards standardized datasets, one of the major problems in activity recognition is ground-truth labeling. Traditionally, this is done offline from the researcher and requires a lot of effort even for controlled datasets with few participants [1, 3, 4]. It is not feasible to do minute labeling for large scale, real-life recordings. We believe that commercially available wearable computers with head-mounted display will enable us to better track and label activities during everyday life. In addition, there is a

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).
UbiComp '14 Adjunct, September 13-17, 2014, Seattle, WA, USA
ACM 978-1-4503-3047-3/14/09.
<http://dx.doi.org/10.1145/2638728.2638798>

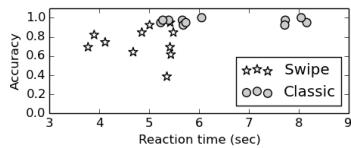


Figure 1: The diagram displays the reaction time vs the correctness of the input. Each symbol depicts one experiment run (40 samples). The star symbol reflects the swipe method and the circle symbol reflects the classic input method.

growing community of people who perform self-tracking using spread sheets and smart phone applications to record what they are doing during the day [2]. Smart phone applications, like reporter¹, pop up reminders 5-10 times a day and ask the user questions about their actions. The granularity of these loggers is too low and it is too disruptive for a lot of people to use these applications on a mobile phone. In turn, we evaluate if it's possible to create a wearable computing interface for HMDs (in our case Google Glass) that gives the user the possibility to quickly select his current activity from a given set without disturbing him too much.

Swipe Gestures

Google Glass and similar devices target micro-interaction which can be performed fast and with minimal interference with other cognitive or physical activity, making them a perfect platform for ground-truth labeling and life-logging. For many types of tasks the default Google Glass interface works very well. An exception is the selection of items from larger lists, in particular when voice interaction is not desired and the user is moving. The default interface involves hitting the right item through a controlled analogue motion (depending on how fast/far the finger moves) on the touchpad which are difficult to perform exactly when the user is moving and require a degree of concentration. As an alternative we propose a "digital" interface that codes the item selection through hierarchical combinations of multitouch gestures (number of fingers, number of swipes, direction of swipe) supported by appropriate representation of the choice on the screen. It makes one dimension of the target size infinite. The user does not need to scroll to the "right" card, but makes his selection over number of fingers and swipe direction.

¹<http://www.reporter-app.com>

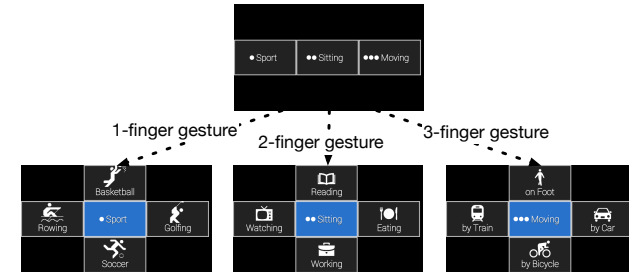


Figure 2: Google Glass swipe gesture input method screenshots. The upper screen displays the categorical overview with hints how to access them (one dot = one finger, ...). As soon as one, two or three fingers touch the touchpad on the right side of Google Glass the corresponding label selection matrix appears. To select a label the user keeps his fingers on the touch pad and swipes to the front, back, up or down.

Evaluation

For our initial experiment, we evaluated 11 participants, using both input methods subsequently. We studied the hierarchical list based input method and ours referred to the multi-touch swipe gesture input. We considered the main target of Google Glass: Performing quick look ups and input while underway. The participant was instructed to continuously walk during the experiment to achieve a natural environment with distractions i.e. watching their steps to avoid obstacles. While the participant was in motion the experiment observer communicated one label after another. We repeated this 40 times for each method with random labels. Figure 1 displays the reaction time vs the correctness of the input. For all participants the reaction time is decreased significantly (p-value of 0.04). The reaction time is reduced by 1.7 seconds on average. Yet, the mean accuracy decreases to 75% for the swipe interface versus 95% on average for the classic interface.

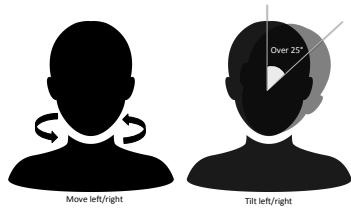


Figure 3: Head movements considered for gestures: moving/tilting left/right.

Head Gestures

For hands-free interactions with the logger we implemented head gestures. We implemented two gesture sets using Google Glass. The first gesture set (referred to as "move" in the following) consists of turning the head left or right to browse through the cards on the HMD. The second gesture set (referred to as "tilt" later on) uses tilting the head over 25° in one direction for browsing. For both sets a double "nod down" is used for selection. The recognition is done using the motion sensor abstraction in Android, simple thresholding on the rotation vector aggregated from the accelerometer, gyroscope and magnetic field sensor over the Android Sensor API. Based on successful initial tests with participants we considered calibration or user dependent classification as unnecessary.

Evaluation

We conducted an initial study to investigate the suitability of the designed head gestures. We compared the standard touch pad provided by Glass (using a hand), the tilt gestures and the move gestures. We measured the time it took after the user left the start screen until he selected the right card and counted how many errors happened during the input (e.g. card interface moving left instead of right). 3 male and 2 female participants (age: 23 - 60) were asked to select 1 out of 11 cards shown on Glass (one start screen and 10 cards with numbers 1 - 10 in order). Participants were asked to select each of the cards 10 times in a randomized order. We disabled Google Glass' standard activation head gesture.

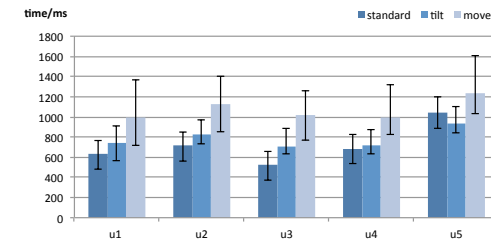


Figure 4: Average time to switch between live cards on the Google Glass display for the different interaction modalities

As seen from the average time to switch between cards in Figure 4, users are not surprisingly fastest on the standard Glass interface with around 710 ms using one hand for selection, the "tilt" interface is with a mean of 782 ms, the "move" interface is worse with 1092 ms average switch time.

Interactive Tagging of Real World Objects

In many situations, the ability to select real world objects can be an important feature of an activity logging platform. We investigated two different live video tagging approaches and implemented prototype versions of them on Glass for an early evaluation. In the first approach, the user selects an object of interest by circling it with an off-the-shelf laser pointer. The software tracks its curve in a sequence of video frames, and recognizes circular movement. The user receives immediate feedback by playing a sound as soon as the spot is detected and tracked. Having recognized the circling gesture, the sub image enclosing the tracking curve is extracted. Figure 5 summarizes the selection procedure. The second interface is based on the touchpad of Glass. As shown in figure 6, the user subsequently adjusts the dimensions of a rectangle overlaid to the video stream to fit the object to select.

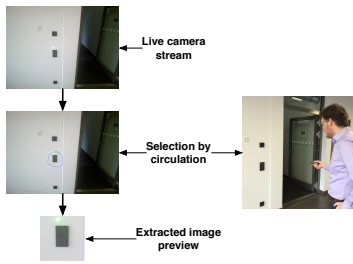


Figure 5: Laser pointer based image selection.

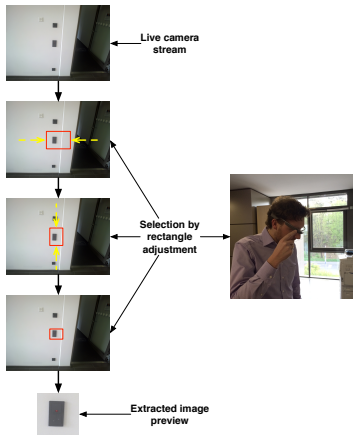


Figure 6: Touchpad based image selection.

Evaluation

We evaluated the functionality, usability, and usefulness of both interfaces in a study involving 11 test persons. All of them were between 20 and 40 years old. Seven did not have a background in computer science, the remaining four persons were from our research department. The participants had to tag a number of objects listed in Figure 7 which were of different size and had to be selected under different illumination conditions. During the experiment, we counted the number of tries as well as the time it took to select an object successfully with each system. Figure 7 summarizes the time/tries of both interfaces for each object averaged over all people.

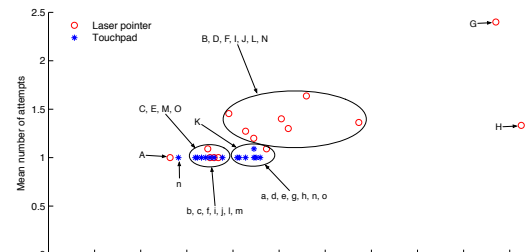


Figure 7: Mean time/mean tries per person for both interfaces. Each point corresponds to a different object averaged over all participants. Letters denote objects as follows: (A, a) Socket, (B, b) Lightswitch (hallway), (C, c) Smoke alarm, (D, d) Door plate, (E, e) Lightswitch (foyer), (F, f) Router (foyer), (G, g) Painting, (H, h) Lamp, (I, i) Router foyer 2, (J, j) Fire alarm, (K, k) Poster logo, (L, l) Router (hallway), (M, m) Handle, (N, n) First aid box, (O, o) Cupboard lock

Sensor Data and Backend Support

The system is compatible to the data format of the CRNT Toolbox (using CSV and json as export format) and the data can also be loaded into our open-sourced labeling tool for offline adjustments.

Related Work

The closest to our framework is Openshades.com, an open framework for Google Glass. Yet, Openshades focuses mostly on computer vision and human computer interaction, not on activity recognition and activity logging.

Conclusion

In this paper we extended our work on activity recognition towards Google Glass. We presented a new input method for quick micro interaction with Google Glass touchpad swipe gestures, a method to navigate through the ContextLogger interface by head gestures, and a method for tagging and picturing of real world objects with a laser pointer.

References

- [1] Bannach, D., Kunze, K., Weppner, J., and Lukowicz, P. Integrated tool chain for recording and handling large, multimodal context recognition data sets. In *Proc. of Ubicomp*, ACM (2010), 357–358.
- [2] Gouveia, R., and Karapanos, E. Footprint tracker: supporting diary studies with lifelogging. In *Proc. of the SIGCHI*, ACM (2013), 2921–2930.
- [3] Ishimaru, S., Kunze, K., Kise, K., Weppner, J., Dengel, A., Lukowicz, P., and Bulling, A. In the blink of an eye - activity recognition with google glass. In *Proc. of AH*, ACM (2014), 150–153.
- [4] Lester, J., Choudhury, T., and Borriello, G. A practical approach to recognizing physical activities. In *Pervasive Computing*. Springer, 2006, 1–16.