Adjunct Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '16), October 24–26, 2016, Ann Arbor, MI, US.

Development Tool for Rapid Evaluation of Eyes-free In-vehicle Gesture Controls

Jason Sterkenburg
Michigan Technological University
Houghton, MI 49931, USA
jtsterke@mtu.edu

Steven Landry
Michigan Technological University
Houghton, MI 49931, USA
sglandry@mtu.edu

Joshua Johnson
Michigan Technological University
Houghton, MI 49931, USA
jocjohns@mtu.edu

Myounghoon Jeon
Michigan Technological University
Houghton, MI 49931, USA
mjeon@mtu.edu

Automotive'UI 16 Adjunct, October 24-26, 2016, Ann Arbor, MI, USA
ACM 978-1-4503-4654-2/16/10.
http://dx.doi.org/10.1145/3004323.3004357

Abstract
In-vehicle controls, such as navigation systems, radio dials, and climate controls, can be visually demanding and can increase crash risk. We are attempting to use commercially available gesture detection equipment to develop an eyes-free system that can provide access to the same controls without increasing crash risk. We envision a system that can be controlled by intuitive in-air gestures. Information about system status and gesture detection can be provided through an auditory menu display, rather than the visual modality as in touchscreen controls. Here, we describe our motivation to develop the system and describe a tool we have developed to help researchers, even those without programming experience, to configure multiple system designs for rapid usability evaluation and iteration practices.

Author Keywords
In-vehicle controls; eyes-free; tool for rapid evaluation.

ACM Classification Keywords
H.5.2. User interfaces: auditory (non-speech) feedback; input devices and strategies (e.g., mouse, touchscreen); interaction styles (e.g., commands, menus, forms, direct manipulation).

Introduction

Touchscreen use in vehicles has been shown to increase crash risk [1, 2, 3] and has been a subject of concern among driving researchers for many years [4, 5], which has sparked efforts to develop new in-vehicle information systems (IVISs) that reduce visual demands of in-vehicle control use [6, 7]. Recent technological advances have made it possible to cheaply and efficiently measure hand positions of drivers using infrared sensors (e.g., LEAP Motion), or computer vision (e.g., Microsoft Kinect). Some researchers have begun exploring these technologies as tools to develop in-vehicle control systems [8, 9, 10].

We envision a potential in-vehicle gesture control system that can be controlled via in-air movements. Auditory information can be relayed to a user in an open-loop system (i.e., user hears current system status), supplementing or even replacing visual information required to use traditional in-vehicle controls.

There are many questions remaining about the practical implementation of any in-vehicle gesture control system. For example, what type of menu layout is easiest to navigate? What target size optimizes the tradeoff between speed/accuracy? What menu breadth/depth facilitates best driving behavior and secondary task performance? What types of auditory display most effectively convey system status without overloading the driver? Obviously, it is not possible to answer all of these questions in one study. Instead we need to complete successive experiments, testing many iterations of our air gesture prototypes. In order to help us with our rapid prototyping process, we decided to develop a tool that allows for flexible configuration of system design. This tool allows researchers, even those without any programming experience, to rapidly develop and evaluate successive iterations of our gesture control system design. The remainder of this extended abstract describes more basics of the gesture control system, as we have envisioned and designed it so far, and also introduces the tool we have made to facilitate rapid prototyping of air gesture controls.

Eyes-free Gesture Control Prototype

In our initial investigation into the feasibility of eyes-free gesture controls, we were interested in two factors: the arrangement of the grid (2x2 grid vs. 4x4 grid), and the presence of an auditory display, which provides information about the current cursor position in the system [10].

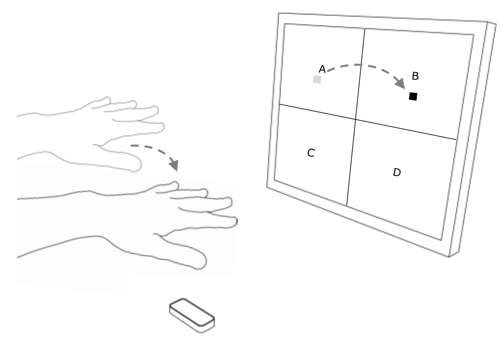

Figure 1: User moving from target A to B with air gesture controls.

System Design

A LEAP Motion, an infrared sensor designed to recognize hand features, was used to detect the hand position of the driver. The LEAP Motion tracks the center of a user's palm, which displays as a "cursor" on a monitor which also displays the menu layout (Figure 1). At any time the user has his/her hand over the LEAP Motion, the monitor will display a cursor indicating the user's hand position. The target which the cursor is currently hovering over, is highlighted (indicated by an alternative background color) (Figure 2). An accompanying auditory display played speech-to-text files stating the name of the target over which the user's cursor is currently hovering. Target names are completely configurable. For our first study we chose target names A, B, C, D, but we changed our target names to be more representative in later studies (Figure 4).

System Navigation
When the user's hand moves, the cursor moves correspondingly within the display (e.g., if the user's hand moves left, the cursor moves left on the monitor). If the cursor moves far enough that it enters the boundary of a new target, the new target will be highlighted, and the old target will return to an un-highlighted state.

The LEAP Motion also counts the number of visible fingers. When the LEAP Motion detects five fingers, it will perform a selection action. When a target is selected, it will flash a bright color, indicating it has been selected. For the auditory display, a selection action is followed by a confirmatory earcon which contains two "raindrop" tones, the first low followed immediately by a second higher frequency note.

This description of how our first system was designed and operated is not intended to be prescriptive in any way, but it does provide helpful context for understanding why we developed our tool for rapid evaluation and what it does.

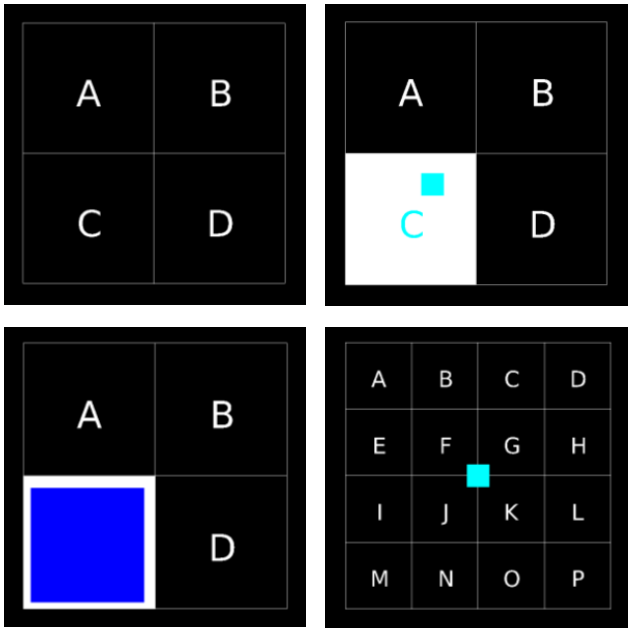

Figure 2: (top left) 2x2 grid, (top right) 2x2 grid with visualization of hand position and highlighting target C, (bottom left) 2x2 grid showing visualization of a selection, (bottom right) 4x4 grid with hand position.

Tool for Rapid Evaluation

We began our prototype design process using PureData, an open source graphical programming language. This software worked well for our purposes, until we started designing experiments with many conditions and we needed to make many prototypes quickly and test them. We decided that it would be more beneficial for us to invest in making a tool to help us easily configure multiple prototypes with a GUI.

Although creating the tool takes more time, when it is completed, it allows for researchers, even those without programming experience, to quickly and easily create many different iterations of a gesture control prototype. Below is a description of the tool we made for that purpose.

Software

Using Eclipse, an Interactive Development Environment (IDE) for Java, we built a Graphical User Interface (GUI) that separates the end-user, researchers, from the code (Figure 3). Within the IDE, window prompts guide users to enter information about how they would like to name their files, and testing block they want to use. Configuration files can be edited easily using a text editor to update target names, change the number of targets in a grid, change the size of the targets, and add or remove the global menu header (i.e., menu which has "home", "back", etc.). Text-to-speech auditory prompts, used to cue participants when and what to select, can be configured as well. Individual wave files or mp3 files containing speech and/or non-speech audio can be played when a user's cursor position is over a target. It is also possible to control the amount of delay before an audio file plays after a user has entered a target. Blocks contain a whole set of unique configurations. Researchers can use blocks to test different configurations against each other in a lab environment. For example, if researchers want to test one version with audio and one without, they can set up two blocks to represent those two conditions. Researchers can make a unique block for every condition they want and they can control the order in which participants see each block system.

The tool allows researchers to:

- Adjust menu depth (e.g., settings > audio)
- Update target text
- Change number of rows and columns
- Change size of targets
- Add or remove menu header
- Add or remove audio files for specific events
- Control delay for audio files
- Change audio cues (e.g., "beep" vs. "hello")
- Design unique blocks of settings

Our software has been uploaded to GitHub (https://github.com/trimlab/IVG-Leap-Grid.git) and can be shared upon request. We are still updating the code and we will make it available to the public when it is complete.

Conclusion

We created a prototype in-vehicle gesture control system with the intention of reducing crash risk associated with in-vehicle control use. We also developed a system that allows for easy configuration and administration of multiple in-vehicle gesture prototypes. This tool serves as a test bed for the early stages of iterative prototyping and rapid evaluation of in-vehicle controls. This tool is especially useful for researchers who do not have experience coding, but still want a number of control over their system designs.

References

1. W. Horrey, and C. Wickens, "In-vehicle glance duration: distributions, tails, and model of crash risk" Transportation Research Record: *Journal of the Transportation Research Board,* vol. 2018, pp. 22-28, 2007.
2. Klauer, Sheila G., et al. "The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data." (2006).
3. Olson, Rebecca L., et al. *Driver distraction in commercial vehicle operations*. No. FMCSA-RRR-09-042. 2009.
4. P. Green, "Crashes induced by driver information systems and what can be done to reduce them," In *Sae Conf. Proc.* SAE; 1999, 2000.
5. Burnett, Gary E., Steve J. Summerskill, and Jack M. Porter. "On-the-move destination entry for vehicle navigation systems: Unsafe by any means?."*Behaviour & Information Technology* 23.4 (2004): 265-272.
6. Sodnik, Jaka, et al. "A user study of auditory versus visual interfaces for use while driving." *International Journal of Human-Computer Studies* 66.5 (2008): 318-332.
7. Riener, Andreas. "Gestural interaction in vehicular applications." *IEEE Computer Society* 4 (2012): 42-47.
8. May, Keenan R., Thomas M. Gable, and Bruce N. Walker. "A multimodal air gesture interface for in vehicle menu navigation." *Adjunct Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 2014.
9. Gable, Thomas M., et al. "Exploring and evaluating the capabilities of Kinect v2 in a driving simulator environment." *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 2015.
10. Sterkenburg, J., Landry, S., Jeon, M., & Johnson, J. Towards an In-vehicle Sonically-enhanced Gesture Control Interface: A Pilot Study. *Proceedings of the 22nd International Conference on Auditory Display*, Canberra, Austria, (2016).