The Implementation of a Glove-Based User Interface

Chris Carey

January 26, 2010

Abstract

Multi-touch interfaces have been rising in usage because of how their multiple points of input simplify the execution of complex commands. Depending on the task being performed, multi-touch interfaces can have a strong advantage over standard mouse interfaces. However, they require the user to physically touch the screen. This project explores the glove-based user interface, an interface that goes a step further than the multi-touch interface by providing the utility of a multi-touch interface without the proximity restriction of physically touching the screen. Though the glove interface is not more efficient than the mouse interface if its gestural commands simply match those of a mouse interface, its multiple levels of input allow for the simplification of commands, similar to those seen in multi-touch interfaces, that increase task completion efficiency and provide for a more natural human-computer interaction experience.

1 Introduction

1.1 Purpose

The purpose of this project is to explore the glove-based user interface, an emerging interface that takes place in 3D space and is not restricted to a 2D surface. Applications that deal with complicated data or data in three dimensions need an interface that simplifies its tasks and can provide three dimensional input. It is expected that the glove interface will have advantages in specific applications and not in general computer control, but the prospect of exploring a new phase of human-computer interaction still remains.

1.2 Scope

The goal of this project is to implement a glove-based user interface in order to determine where its advantages and disadvantages lie. A focus on task completion is necessary to evaluate the effectiveness of such an interface [1], and it must be allow for the evaluation of gestures relevant to controlling various applications such as software for geo-spatial imaging, 3D modeling, information visualization, and presentations.

1.3 Background

As the tools and technologies for building alternative user interfaces have become more readily available, alternatives to button and mouse interfaces have emerged. Multi-touch interfaces have been implemented as early as the mid 1980s [2], and have grown in usage both independently and commercially over the past few years as a result of improved accessibility to the required technology [3]. Recent advances in infrared-based multi-touch technologies have been moving towards reducing the need for the user to physically touch the screen, by instead allowing them to hover over the screen [4]. A glove-based user interface, based on nearly identical IR LED technology, would eliminate this restriction all together. And though the idea of a glove-based user interface dates back to the beginnings of virtual reality [5], with proven applications in IR LED sensing now available in todays popular consumer electronics provide the glove-based interface with the potential for wider usage.

2 Implementation

2.1 Hardware Implementation

A modified Logitech USB webcam is used to provide a live video feed of infrared light. The internal IR-blocking filter was removed and an external visible-light blocking filter was created from floppy disk magnetic film.

Fig. 1 IR LED Gloves

Each glove contains three infrared LEDs with a wavelength of 950 nm. The LEDs are located on tips of the thumb, pointer finger, and middle finger. The gloves are wireless and are powered by three 1.5 V AAA batteries.

2.2 Language and Structure

This research project is written in Java using the Java Media Framework in an effort to make the software more accessible and more efficient. A modular architectural framework is utilized in order to add recognizable gestures more easily.

3 Procedure

3.1 LED Detection

Given a captured frame from the webcam, each pixel is identified as either a background pixel, or a foreground pixel representing the IR LED. This binary rasterization is created by automatically determining a threshold value for which each pixel's brightness value is compared to. Pixels with a greater brightness are defined as foreground pixels while pixels with a lesser brightness are defined as background pixels. The optimal threshold value is determined by creating a histogram of pixel brightness values and selecting a value past the peak brightness level that causes only a small percentage of pixels to be considered the foreground, since the IR LEDs only occupy a small portion of the frame [6].

3.2 LED Tracking

The user is required to set the initial state of the LEDs each time all of the user's fingers are off-screen and reappear on-screen. This is accomplished by having the user bring all six LEDs (or three LEDs for only single-handed gestures) into the video frame for about two seconds so that the application can determine which LEDs belong to the left and right hands, which LEDs are used for pointing, which LEDs are used for clicking, and which LEDs are for auxiliary use. An LED object class is created each time an LED is identified and stores information about the LED's classification and previous positions.

Whenever an LED moves off-screen and reappears on-screen, it is reclassified using logic based primarily on which group of LEDs (left or right) it is closest to and which LEDs classifications are not in use.

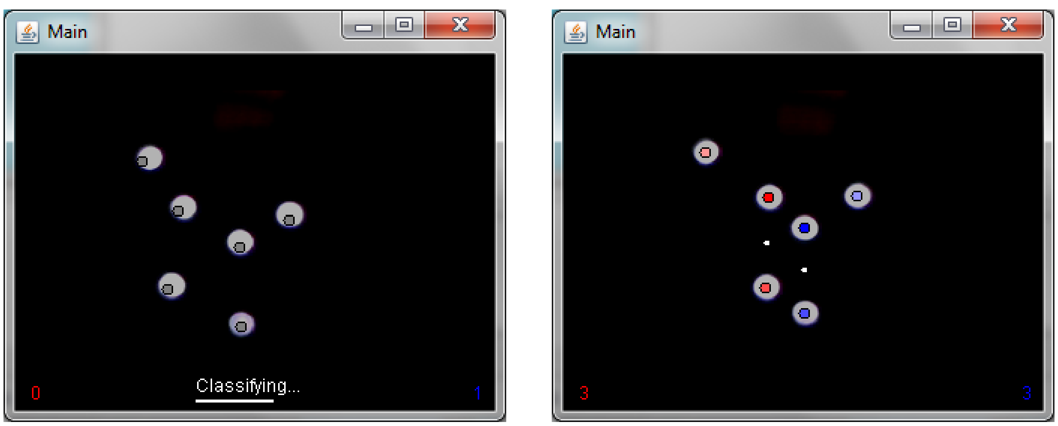

Fig. 2 LED Classification - In this video capture, left hand LEDs are marked as blue and right hand LEDs are marked as red

3.3 Photo Manipulation Application

A simple photo manipulation application was created for testing and demonstration purposes. In this application the user can drag, rescale, and rotate photos using both the mouse interface and the glove interface. Several predefined tasks can be performed, in which the user must manipulate one or more photos to match a final orientation. The final orientation of the photos is displayed as a series of faded-out photos in the background that the user must match the actual photos to.

Fig. 3 Photo Manipulation Application

Cursor and photo position/orientation information is gathered each time step and exported to a CSV (Comma-Separated Values) file whenever a task is completed. From this, relevant space and time data can be evaluated and compared.

3.4 Gesture Recognition and Command Execution

All of the implemented gestures are centered on their usage in the photo manipulation application. Currently, only single-handed gestures have been successfully implemented and evaluated. This allows for the comparison of the mouse interface and the glove interface in simple cursor pointing, clicking, and dragging. Double-handed gestures offer a simpler method of performing these gestures, but have not been successfully implemented and evaluated yet. Furthermore, a balance between single-handed gestures and double-handed gestures will need to be found as most users prefer single-handed gestures over double-handed gestures in multi-touch interfaces [7].

3.4.1 Glove Interface Cursor Control

The cursor's position is based proportionally on the midpoint between the pointer LED and the clicker LED which correspond to the user's thumb and pointer finger. When the LEDs are brought within close proximity to each other, the cursor press command is executed. When the LEDs are moved apart from this state, the cursor release command is executed. The drag gesture of the glove interface is analogous to the drag command of the mouse interface. The user pinches these two LEDs together and moves them to execute the drag command.

3.4.2 Photo Drag

A photo is dragged by dragging anywhere on the image except the marked corners.

3.4.3 Photo Rescale

A photo is rescaled by dragging any of the photo's marked corners towards or away from the center to decrease or increase the photo's size respectively. A photo can also be rescaled by dragging two cursors anywhere on the photo and moving them together or apart; this is an incomplete double-handed gesture.

3.4.4 Photo Rotate

A photo is rotated by dragging any of the photo's corners and rotating it around the photo's center point. A photo can also be rotated by dragging two cursors anywhere on the photo and rotating them around their midpoint; this is an incomplete double-handed gesture.

4 Experiment

Three different tasks in the photo manipulation application were performed with both the mouse interface and the glove interface (using single-handed gestures) using the dominant hand in both instances. The first task was to drag a photo 500 pixels from left to right. The second task was to rescale a photo from 25% to 100% of its original size. The third task was to rotate a photo 2.0 radians clockwise. A performance of the gesture that is representative of how that gesture would normally be executed was chosen for analysis.

5 Results

In the first task, the glove spent 42.6% (0.785 s) more time dragging than the mouse. In the second task, the glove spent 12.7% (0.282 s) more time dragging than the mouse. In the third task, the glove spent 72.3% (2.283 s) more time dragging than the mouse.

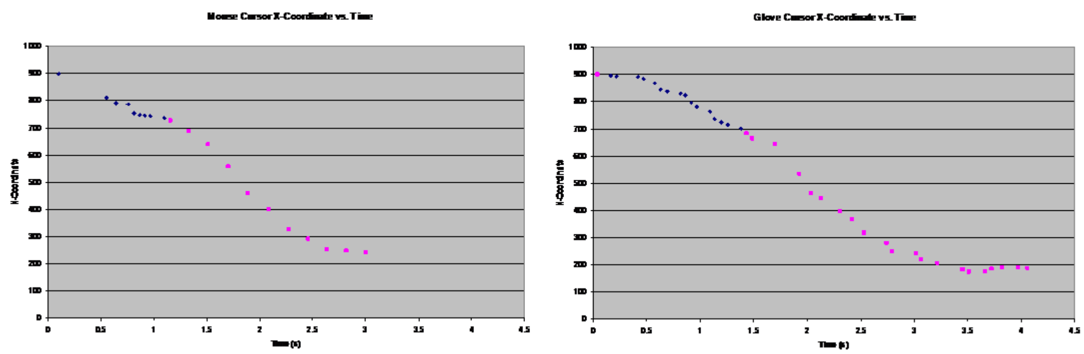

Fig. 4 Representative Results of Task 1 (dragging photo 500 pixels left): Mouse (Left) and Glove (Right) Cursor X-Coordinate vs. Time - pink squares mark when cursor was pressed

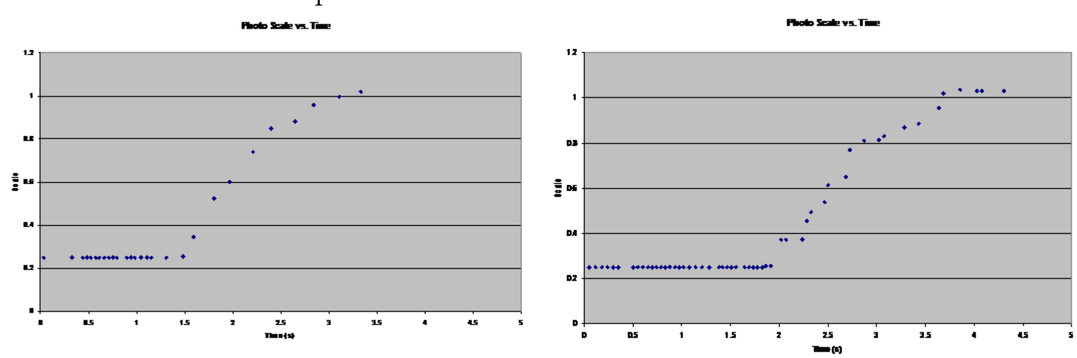

Fig. 5 Representative Results of Task 2 (rescaling photo from 25% to 100%): Photo scale vs. Time - mouse (left) and glove (right) interfaces

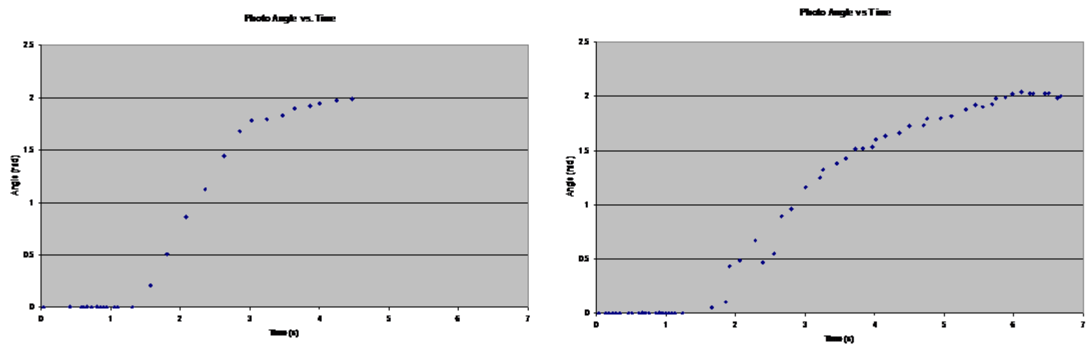

Fig. 6 Representative Results of Task 3 (rotating photo 2.0 radians clockwise): Photo Angle vs. Time - mouse (left) and glove (right) interfaces

It should also be noted that the motion of the glove interface was smooth. However, more correction time was spent when placing a photo into its proper orientation. Overshooting the target was more common when using the glove interface than using the mouse interface.

6 Analysis

In simple single-handed cursor dragging gestures, the glove interface consistently spent more time dragging than the mouse interface. The speed at which the glove cursor was dragged was not significantly slower than that of the mouse interface, but more time was spent placing the photo into its proper orientation. This leads to the conclusion that the glove interface cannot match the accuracy of the mouse interface, at least with its current sensitivity level. A wider camera lens and greater space for interface control would be required to decrease the sensitivity and increase the accuracy. But it should also be noted that the reason a mouse interface would have more accuracy than a glove interface under the same spatial conditions is that the user can rest his or her hand on the mouse allowing gravity to act as a damper. With a glove interface, the user must in fact work against gravity without that damper against small, sudden movements.

7 Conclusion

The glove-based user interface is not more efficient than a mouse-based user interface if it is only used as a replacement with its gestures merely matching those of the mouse interface. In order for the glove interface to be more efficient than the mouse interface, it must take advantage of its multiple degrees of input to simplify tasks that would take more steps to perform with a mouse interface. This confirms the need to design a glove interface with a foundation in task completion and simplification, and is something that will continue to be explored as double-handed gestures will be implemented and evaluated in the near future.

References

[1] Molina, Jose P., et al. 'The Development of Glove-Based Interfaces with the TRES-D Methodology.' *Virtual Reality Software and Technology.*, pp. 216-219, 2006.

[2] Lee, SK, William Buxton, and K. C. Smith. 'A Multi-Touch Three Dimensional Touch-Sensitive Tablet.' *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* San Francisco, pp. 21-25, 1985.

[3] Kim, Ji-Sun, et al. 'iPhone/iPod Touch as Input Devices for Navigation in Immersive Virtual Environments.' *Virtual Reality Conference, 2009. VR 2009. IEEE.*, Lafayette, pp. 261-262, 2009.

[4] Izadi, Shahram, et al. 'ThinSight: A Thin Form-Factor Interactive Surface Technology.' *Communications of the ACM 52.12*, pp. 90-98, 2009.

[5] Sturman, David J., and David Zeltzer. 'A Survey of Glove-based Input.' *Computer Graphics and Applications, IEEE.*, pp. 30-39, 1994.

[6] Baek, SeongHo, et al. 'IRED Gun: Infrared LED Tracking System for Game Interface.' *Lecture Notes in Computer Science 3768/?2005.*, pp 688-699, 2005.

[7] Wobbrock, Jacob O., Meredith Ringel Morris, and Andrew D. Wilson. 'User-Defined Gestures for Surface Computing.' *Proceedings of the 27th*

International Conference on Human Factors in Computing Systems. Boston, pp. 1083-1092, 2009.