

The Implementation of a Glove-Based User Interface

Chris Carey

October 27, 2009

Abstract

As the advantages of multi-touch interfaces are being utilized in numerous applications to improve the user experience, this research project aims to explore a user interface that goes a step further. A glove-based interface provides the utility of a multi-touch interface without the proximity restriction of physically touching the screen. It provides a more natural human-computer interaction which in certain applications also improves efficiency in the performance of complicated tasks.

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 that is responsive, accurate, and efficient. Only when the system meets these goals

can a fair comparison in usability and effectiveness be made to other user interfaces including the mouse interface, the multi-touch interface, and button interfaces of applications. Sophisticated machine vision and gesture recognition techniques have already been employed using real-time video feeds and don't require the user to use any other hardware in order to interact with the computer. However, this project focuses on the development of a gesture-based interface with a focus on accuracy, efficiency, and usability compared to a traditional mouse-based interface. Therefore, a glove-based interface is implemented in order to bypass the difficulties of implementing machine vision techniques on a user in everyday conditions. A focus on task completion is necessary to evaluate the effectiveness of such an interface [1], and in order to investigate how it enhances the user experience, it must be able to manipulate various applications such as software relating to geo-spatial imaging, 3D modeling, information visualization, and presentations.

1.3 Background

As the technologies and tools 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]. And though the idea of a glove-based user interface dates back to the beginnings of virtual reality [4], proven applications in IR LED sensing such as with the Nintendo Wii provide the glove-based interface with the potential 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. 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 two 3V button cell 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

Each captured video frame is evaluated through binary rasterization where each pixel is identified as either a background pixel, or a foreground pixel representing the IR LED. A pixel brightness threshold value is defined to categorize the pixels. 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 [5].

3.2 LED Tracking

An LED object class keeps track of previous positions in order to calculate its velocity and next predicted position based. After the positions of the LEDs in the video frame are located, they are matched with LEDs currently being tracked. If a new LED is detected, a new LED object is created. If an LED no longer appears in the video frame, the corresponding LED object is deleted.

3.3 Gesture Recognition and Command Execution

Since there is no way to determine which LED is connected to which finger, gestures are defined such that they do not depend on which finger performs which action. Gestures are separated into two categories: static gestures and dynamic gestures.

3.3.1 Static Gestures

Static gestures are gestures that (1) do not depend on the absolute location of the LEDs, and (2) are recognized and executed once for each time the gesture is performed. Static gestures correlate to single commands in the user interface. This project currently recognizes one static gesture as a command:

Minimize The minimize command is executed by bringing three LEDs together and then moving them back outwards. It is recognized by checking that the distance between all three LEDs is decreasing above a minimum rate and that a minimum distance between is met before the distances begin increasing. The minimize function uses the `java.awt.Robot` class to execute the keystroke shortcut for minimizing a window.

3.3.2 Dynamic Gestures

Dynamic gestures are gestures that (1) do depend on the absolute location of the LEDs, and (2) are recognized and executed continuously as long as the gesture is being performed. Dynamic gestures correlate to input control, the most prominent dynamic control device being the mouse. This project currently recognizes two dynamic gestures.

Mouse Movement The mouse movement command is the most basic gesture recognized. When no other gestures are being performed, an LED is recognized as the pointing LED. Whether this LED with the greatest y-value or the smallest y-value depends on user-preference in hand position. The mouse is moved to the position on the screen relative to the screen boundaries that correlates to the pointer LEDs position relative to the video frame boundaries.

Drag and Drop This gesture is executed by bringing two LEDs together and keeping them together while moving them. It directly relates to a pinching gesture, which is a more natural way to execute this command. It is recognized when the distance between two LEDs decreases at a minimum rate and until within a minimum distance. The mouse primary button is pressed as long as the LEDs maintain this minimum distance, and is released when they come apart.

4 Results and Analysis

A full analysis of the performance of the glove-based interface will be completed once all LED location and tracking methods have been optimized. For now, only a general overview of the current performance of the glove interface can be presented.

4.1 Speed

The human eye cannot detect a refresh rate faster than 24 frames per second. In other words, video refreshed at 24 frames per second appears no slower to the human eye than real events. The ultimate goal is to reach a refresh rate quicker than 24 frames per second.

Currently this interface reaches an average refresh rate of 21 frames per second; a noticeable lag to the human eye and to the motion of the human body. In order to create a natural user experience, the frame rate must be improved by optimizing algorithms.

4.2 Accuracy

4.2.1 LED Detection

Though no quantitative analysis has been conducted yet, LED detection within the video frame is consistent given that the LED shows up clearly in the video frame. Perhaps the largest obstacle to this project is the clarity in which LEDs appear in the video frame. The range at which LEDs appear in the video frame is only within approximately one foot for satisfactory detection. The LEDs must also be pointed towards the webcam in order for its light to be visible.

The obstacle to improving the range of LED detection is linked to the physical limitation of LEDs. By increasing the current of the LED, the brightness of the LED is increased. However, an LED cannot sustain a high current without physical damage to the LED from heat.

4.2.2 Static Gesture Recognition

The accuracy to which static gestures are recognized can be evaluated by comparing the number of times a static gesture is performed by the user to the number of times a static gesture is recognized by the computer. No

quantitative testing has been conducted yet. The minimize gesture does simplify the task compared to a mouse interface. However, the most common cause of the failure to recognize a static gesture is the failure of the LEDs to appear in clearly the video capture frame.

4.2.3 Dynamic Gesture Recognition

4.3 Current Status

The mouse pointer can be controlled with reasonable accuracy depending on range. The drag and drop function is recognized with poor accuracy. The drop function prematurely activates if an LED changes angles and is not picked up by the webcam.

On the computer being tested, a resolution of 320 by 240 pixels is being mapped to a screen with a resolution of 1280 by 800 pixels. This causes the maximum error between the video capture and the screen output to be 4 pixels. This accuracy limitation must be evaluated and may limit the dexterity through which precise tasks can be performed.

5 Conclusion

The glove-based user interface being implemented in this research project has several obstacles yet to overcome before its ready for full testing, analysis, and specific application usage. LED detection must be improved in order to ensure usability and gesture recognition accuracy. This project has illustrated the potential for a glove-based interface, but has yet to be fully implemented and evaluated.

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] Sturman, David J., and David Zeltzer. 'A Survey of Glove-based Input.' *Computer Graphics and Applications, IEEE.*, pp. 30-39, 1994.
- [5] Baek, SeongHo, et al. 'IRED Gun: Infrared LED Tracking System for Game Interface.' *Lecture Notes in Computer Science 3768/?2005.*, pp 688-699, 2005.