Remote Touchscreen Control

Sadya Afreen - 2345987

December 17, 2020

1 Status report

1.1 Proposal

1.1.1 Motivation

Hygiene has always been a concern with shared touchscreen devices, but insufficient to prevent the general public from choosing to interact with them. However, hygiene concerns related to shared public use of these touchscreen systems escalated in recent Covid-19 times where direct touch interaction can no longer be perceived as hygienic and safe. Therefore, to compensate for direct finger interaction with public touchscreen, remote indirect interaction using smartphones can be an alternative to address public concerns around touchscreens, during and beyond the Covid-19 pandemic. Thus, this project investigates interaction techniques with remote screens using smartphones, given that smartphones are a personal device that cannot facilitate transmission of infectious diseases and can enable safe interaction with a publicly shared screen. Remote interaction techniques using smartphones have been presented and evaluated before, but have not been directly compared. Thus, this project presents a novel comparative analysis between a set of interaction techniques to evaluate which type of interaction has better efficiency, intuitiveness, effectiveness, and ease of use.

1.1.2 Aims

This project will present a system with four different interaction techniques that allow a smartphone to remotely control the cursor of a public screen, for interaction without direct physical contact. The interaction techniques will be demonstrated via four different prototypes that will enable us to explore the usability and performance of each technique. A remotely-conducted experiment will be used to gather data about interaction performance. Thus, this project will discuss a detailed study between four state-of-the-art input methods for the first time. Finally, an interactive demonstrator application scenario will be presented to show the incorporation of the remote interaction techniques into a practical context, showing the ecological validity of these techniques.

1.2 Progress

- Background research conducted on existing interaction techniques (direct touch and remote).
- Four interaction techniques were implemented using a native Android app to capture user input with the smartphone.

- A client-server application has been implemented in Java so the Android app can remotely control the mouse pointer on a desktop display (using sockets to pass input events between devices).
- All four interaction prototypes are implemented and can remotely control the display cursor.
- A within-subjects experiment has been designed, with a combination of 2D and 3D input tasks.
  1. The 2D tasks are designed and implemented, with appropriate dependent variables
  2. The 3D tasks are designed and have initially been implemented, but need refinement
- Dissertation is in progress (first draft of the first four chapters, with supervisor feedback).

1.3 Problems and risks

1.3.1 Problems

- Continuous streaming of real-time messages from client-server was tricky and required trying out different asynchronous tasks.
  - **Mitigation**: Android Service is used to separate main program thread from socket connection established in the background.
- Translating the cursor updates received from the client to map to the remote screen cursor position required significant fixtures for all the four techniques.
  - **Mitigation**: A sensitivity ratio was applied to the received updates by the server to translate the coordinates and position the remote screen cursor accordingly.
- Implemented Trackpad technique needs to be refined a little to make significant difference between absolute and relative cursor mode.
  - **Mitigation**: Raw x and y coordinates will be refined on the client side application to absolutely position remote screen cursor by taking into account the difference in screen sizes.
- Experiment tasks need to be refined and implemented properly for remote evaluation.
  - **Mitigation**: : Experiment data (appropriate dependent variables) has to be saved in a text file after performing the task to use it for analysis.

1.3.2 Risks

- Latency between Android input app and desktop output app may be discernible, impacting usability. Mitigation: minimize the number of messages that need to be sent, minimize object creation on the Android app, and reduce filter strength for faster response to movement.
- Experiment measurement data may be insufficient to answer research questions. Mitigation: run a pilot test of the experiment to check data validity and formatting.
- Unforeseeable technical issues when deploying the smartphone + desktop app to peers for remote experiment participation. Mitigation: pilot test the experiment with a friend to identify potential technical issues or sticking points; perform a live experiment briefing (e.g., via Zoom, Teams) to help remote participants get the system set up.

1.4 Plan

Semester 2

- Week 1: Complete final refinements and perform a pilot test of the prototypes and experiment.
- Week 2 – 4: Start remote evaluation by deploying the client and server prototypes.
  - **Deliverable**: quantitative and qualitative measures of usability and effectiveness ready for analysis stage.
- Week 5 – 6: Perform analysis on the data to address them in the report.
  - **Deliverable**: Use R or python to analyze data and create visualisations and graphs ready to write the report.
- Week 7-10: Write dissertation report.
  - **Deliverable**: Submit draft of completed report to supervisor two weeks before the deadline.