Markuss Davis Skrastins

Formative draft outline of methodology and design

# Direct interaction of a mobile phone and a PC display

# 1. The overall design of the system and methodology

In this project the goal has been for the interaction between the device and the display has been to be as intuitive and smooth as possible. Since the image matching is happening at the same time as when user is interacting with this system, the design of the system must allow the user not to be interrupted or slowed down by the image processing that is happening in the background. Furthermore, even though the core focus in this project has been the accuracy and speed of feature matching, this was not the only problem that had to be solved as the overall system also involved such issues as communication between the computer and the phone.

The overall design of the system is as follows; Both the computer and phone that are participating in this system are hosting local servers through which it is possible to connect to both devices on the local network. This way it is possible to get both the live video stream and the touch input from the phone. Further on, the system is also constantly receiving a live bit stream of the computers screen. From both video feeds the features are extracted and then feature-matching algorithms are used to find the region on the screen, that the device’s camera is pointing to. A separate thread is also listening for the touch inputs, which are then translated to the coordinates of the screen and used to interact with the computer itself.

Since the system built is a real-time system, the use of parallel programming and threading is prevalent throughout the system.

## 1.1. Communication between a phone and a computer

Before different implementations of feature-matching algorithms could be tested, the backbone of the system had to be built. The communication between a phone and a computer is an important issue that had to be addressed carefully, as any bottlenecks in the bit stream between the two would be the major factor in the overall performance of the system. Therefore, there could not be any bottlenecks when getting a live video feed and touch inputs from the phone. It is also important to note, that since the user will only be observing the video feed from the phone itself, the frame limit and the quality of the video can be modified and processed without it affecting the video seen on the phone. However, as mentioned before, since the core focus of this project is the image matching, the communication problem was addressed to be convenient for the system.

For the system to receive the live video feed from the phone, there are multiple different approaches. The approach chosen for receiving the live video feed from the phone was to use an application called “IP Webcam”. This application hosts a live local network server. This can then be accessed by connecting straight to the device’s IP address and OpenCV library can be used to read the bit stream directly from there. If not processed, the original video stream is 60 frames per second and in 1080p quality. These settings can be changed from both the interface that the application used offers or by video processing with OpenCV.

Unfortunately, to process the users input from the device, a different method must be used. From the research done, no pre-built ready-to-use systems already exist that can both transmit the video feed and touch input from the phone to the computer. Therefore, for the touch input, the official “Android Debug Bridge” tools were used. Even though it can be possible to get both the video stream and touch input from these tools, it is more convenient to use the previously mentioned “IP Webcam” for the video stream.

For the touch inputs, the official “Android Debug Bridge” (ADB) package was installed on a linux machine, and a server can be hosted locally on the same machine. After some initial configuration, it is possible for an android device to wirelessly connect to the linux machine, and it is possible to listen for touch inputs by using an ADB library that is written for python. Therefore, it is much easier to combine both video feed and touch input python scripts into a one whole system. It is also worth mentioning that since the ADB tools are using a linux shell to listen to the touch inputs, it is running as a separate thread in the system.

## 1.2. Image matching

The image matching methods can be easily implemented in the previously described system by just changing function called. This allows for easy tests and analysis for the different methods and algorithms of the image-matching.

There are many different algorithms and methods for feature extraction and image matching. In this project, some of the more well-known methods have been used and compared. Furthermore, besides of implementing and fine-tuning known algorithms and methods, a custom deep neural network is also built.

## 1.2.1. Scale-Invariant Feature Transform

Scale-Invariant Feature Transform (SIFT) is a method of detecting multiple key points in an image. This method was developed by David G. Lowe in 1999 and is one of the most popular feature detection methods.